Data Analysis via Topology and Geometry with Applications to Chemical Engineering

by

Alexander Smith

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Chemical Engineering)

at the

UNIVERSITY OF WISCONSIN-MADISON

2022

Date of final oral examination: 8/16/2022

The dissertation is approved by the following members of the Final Oral Committee:

Victor M. Zavala, Professor, Chemical and Biological Engineering
Reid C. Van Lehn, Associate Professor, Chemical and Biological Engineering
Michael Graham, Professor, Chemical and Biological Engineering
Jessi Cisewski-Kehe, Assistant Professor, Statistics

To my partner in crime, and in life, Devon.

ACKNOWLEDGMENTS

The path leading to the completion of this dissertation has been topologically non-trivial. There are many loops (e.g., cycles) and discontinuities (e.g., connected components) that I have experienced in this journey that would have been impossible to overcome without the help of many individuals that I would like to acknowledge. First and foremost I would like to thank my advisor, Victor Zavala. Without Victor's guidance and encouragement I would have never made it to the end of this journey. Victor gave me the freedom to explore. He gave me the chance to pursue an area of research that I was passionate about and helped me share it with the engineering community. Victor gave me confidence. He encouraged me to push myself and delve deeply into areas of mathematics that I knew almost nothing about, reassuring me the entire time that it would be worthwhile. Victor also gave me trust. Victor allowed me to be independent in my research. For all of these gifts I am incredibly grateful.

I would also like to thank the entire Chemical and Biological Engineering department for their support throughout this entire process. In particular, I want to recognize Prof. Reid Van Lehn for believing in the ideas presented in this dissertation and supporting me through collaborations. These collaborations with Reid and his group helped give life to this area of research in chemical engineering. The same can be said for the group of Prof. Michael Graham who have helped develop these ideas in the area of dynamical systems. I would also like to thank the many (past and present) members of the Zavalab who always supported me and my work. In particular I want to thank Jordan Jalving, Apoorva Sampat, Ranjeet Kumar, Sungho Shin, and Philip Tominac for their mentorship. Joshua Pulsipher and Yue (Ricky) Shao, thank you for acting as sounding boards for my (mostly bad) ideas. Shengli (Bruce) Jiang, thank you for giving me the opportunity to be a mentor and for all of your hard work and dedication that you put into every one of our collaborative efforts. The same is true for all other members I have had the pleasure of getting to know through these years: Weiqi Zhang, Shiyi (Amy) Qin, Jiaze Ma, Leonardo Gonzalez, Dilara Goreke, Blake Lopez, David Cole, Jaron Thompson, and Lisa Je.

I also want to thank those in industry who have supported my research and helped develop my career. First and foremost, Dr. Salvador Garcia was instrumental in guiding my path from industry back to academia. Jason Jelinek for his mentorship when I worked at John Deere and Ivan Castillo and Benjamin Laubach for their support at Dow Chemical.

Finally, I give my deepest gratitude to my parents Doug and Tessa Smith, my brother Hunter Smith, and my wife Devon Smith for their unwavering support. My parents have

always recognized my potential and have always encouraged me to relentlessly pursue my goals. My brother has always been there as someone to talk to and depend upon. He also continuously inspires me to pursue goals outside of my academic work (he is currently a fashion model, a nuclear engineer, and the lead vocalist of a hard core band). Finally, my wife Devon is the reason I began this entire journey. Without her reassurance, guidance, and advice I would not have had the courage to return to academia and pursue what has become one of the most fruitful, rewarding, and exciting journeys of my life.

Alexander Smith
Madison, WI
August 2022

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This dissertation is focused on the introduction, development, and application of topological and geometrical methods for the analysis of data. Topology and geometry allow us to view data as a *shape* and provide us tools to quantify this shape. Abstracting data as a shape captures *intrinsic* characteristics of the data that are independent of the environment and methods used to obtain the data. These representations (e.g., graphs, manifolds, and point clouds) also provide means for integrating *domain knowledge* into data analysis that can strengthen connections between theory and experiment. In this dissertation, we present a thorough review of applied topology and geometry methods in chemical engineering through applications on real data-sets such as: molecular dynamics simulations, dynamical systems, process systems, and soft matter systems.

We first present methods that are centered on the *topology* of data. We provide deep mathematical foundation for two particular areas of topological data analysis: the Euler characteristic and persistence homology. These methods quantify the *topological invariants* of a data shape (e.g., holes, connected components, voids). These invariants describe intrinsic properties of data that are unaffected by continuous transformations of the data such as translation, rotation, stretching, and bending. We also compare the effectiveness and interpretability of models that leverage the topology of data with methods that do not capture it directly (e.g., Fourier transforms).

We then explore methods that exploit the high-dimensional *geometry* of data. In particular, we focus on the *Riemannian geometry* of symmetric, positive-definite (SPD) matrices. An SPD matrix is a versatile data representation that is commonly used in chemical engineering (e.g., covariance/correlation/Hessian matrices and images). A key observation that motivates this work is that SPD matrices live on a Riemannian manifold and that implementing techniques that exploit this basic property can yield significant benefits in data-centric tasks such as classification and dimensionality reduction.

Finally, we present analysis of other data driven methods, such as (convolutional) neural networks, and explore how these methods can be used to understand the high-dimensional structure and patterns found in data. We leverage the pattern identification power of a pre-trained convolutional neural network (VGG16) for the characterization of liquid crystal sensor responses. We also explore the high-dimensional structure of datasets for the catalysis of the water-gas shift reaction. We explore and develop dimensionality reduction methods to understand the structure of the dataset and identify new catalyst formulations.

Chapter 1

INTRODUCTION

In this chapter we present the background, objectives, and motivation of this dissertation. We discuss some of the widely used data science methods and strategies applied in chemical engineering and our research objectives in addressing some of their computational and theoretical challenges through topology and geometry. We also summarize the structure and content of this dissertation. Finally, we introduce some ideas of topology and geometry in the context of chemical engineering research and motivate their application through a connection found between physics, topology, and geometry.

1.1 Current Practices in Data Science & Chemical Engineering

The focus of this dissertation is primarily on data representation and pre-processing, which is a critical aspect of effective data analysis [5]. The representation (mathematical abstraction) of data can simplify the models needed for data analysis and can provide a way for domain knowledge to be incorporated. This section reviews many of the common practices for data representation in Chemical engineering and how they are used for data analysis.

For example, data from a multivariate time series can be represented in many ways as shown in Figure 1.1. A commonly applied method is to compute the covariances between each individual time series and construct a covariance matrix. In the context of chemical

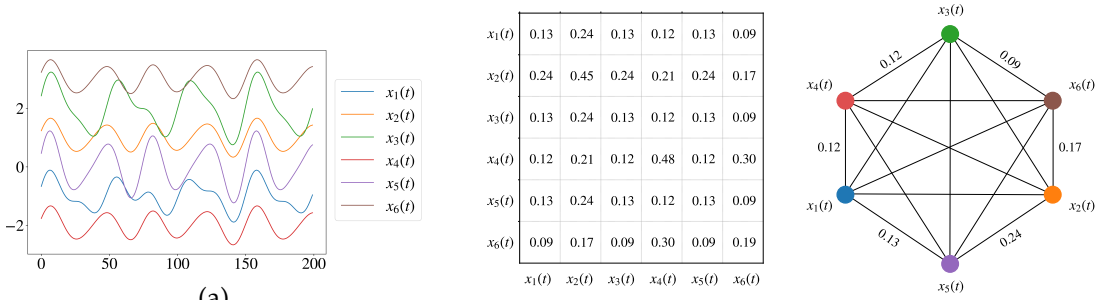


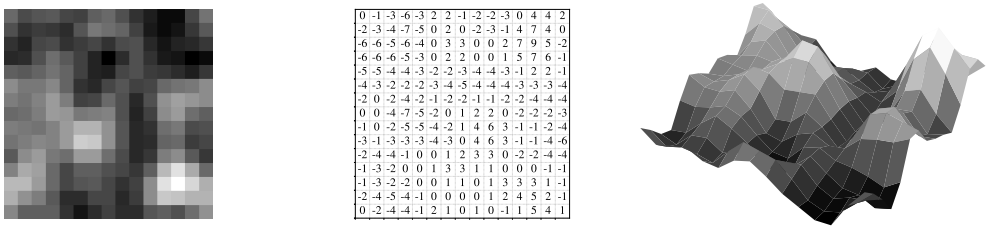
Figure 1.1: Illustration of different representations of a multi-variate time series dataset (a). (b) A covariance matrix representation of the multi-variate time series data in (a). The covariance matrix encodes the covariance relationships between variables $x_i(t), x_j(t)$ and the variance of the individual variables $x_i(t)$. (c) A graphical representation of the covariance matrix. A fully connected graph is constructed and the weights of the edges of the graph are equal to the covariance between each variable. For clarity we only label the edges at the outside of the network, but in the data representation all edges are weighted (see chapter 2 for further detail).

engineering these covariance matrix representations are used in the analysis of data for the control and monitoring of process systems [6, 7, 8, 9, 10]. They are also used in understanding chemical systems through entropy, the dynamics of fluid systems, and in spectroscopy [11, 12, 13, 14].

Analysis of these matrices is often paired with a dimensionality reduction method: Principal Component Analysis (PCA). PCA is a combination of singular value decomposition of a covariance matrix and a projection of data onto the leading eigenvectors [15, 16]. This method is extensively used in this dissertation and explored in detail in chapter 7. PCA has been used in many areas of chemical engineering. In particular it has been used for the analysis and dimensionality reduction of process data [17, 18, 19, 20]. PCA is also applied in the analysis of point cloud data, where data such as individual experiments are represented as points in high-dimensional space or the 3-D positions of molecules in a protein [21, 22]. The structure and shape of these spaces can be understood through PCA, which has been used extensively in areas such as experimental design [23, 24] and molecular dynamics simulation [25, 26, 27, 28]. However, PCA does not account for the *connectivity* of these structures, missing important information encod-

ing potential interactions between data points (e.g., molecules, atoms) [29].

Covariance matrices can also be represented through graphical structures as shown in Figure 1.1. The nodes of the graph represent the individual measured variables. We construct edges between every node of the graph (i.e., a fully connected graph) and weight each edge of the graph with the covariance value between the corresponding variables. Thus, we obtain what is known as an edge weighted graph that captures the information encoded within the covariance matrix. Graphs have been used extensively in chemical engineering research. They are used directly to represent complex networks [30, 31, 32, 33], molecules [34, 35, 36, 31], or are constructed to represent more abstract ideas such as relationships between constraining variables in optimization [37, 38]. More recently graphs have been used as a representation for machine learning based tools such as graphical neural networks (GNN's) [39, 40, 41, 42]. Information can be extracted from graphs in various ways, most common are methods that focus on summarizing the topology of graphs through metrics such as modularity, connectivity, etc [43, 44, 45]. These methods can work well in quantifying a graph structure, but can oversimplify the graphs topology resulting in a loss of information needed to distinguish complex graphs [46].



(a) (b) (c)

Figure 1.2: Different representations of an image (a). (b) A matrix encoding of an image. Each element of the matrix corresponds to the pixel location and intensity within the image. (c) Manifold representation of an image. The image is lifted from 2-dimensional space (e.g., a flat plane), and embedded in 3-dimensional space. Here, the third dimension corresponds with the intensity of the pixel at a given location in the image.

Another commonly analyzed form of data in chemical engineering research is that of images (Figure 1.2). Images are often represented as matrices, making them amenable to

the previously discussed methods. They are also open to more advanced signal processing and machine learning methods such as Fourier transforms and convolutional neural networks (CNNs) due to the spatial organization of the image. Images have become increasingly popular in chemical engineering because they are able to capture more information than simple statistical summaries or individual time series. Signal processing methods, such as Fourier transforms, attempt to extract global structure from an image through a decomposition of the image into independent basis functions or patterns [47]. For example, the Fourier transform has been applied extensively in spectroscropy [48, 49]. It has also been used in the analysis of process systems data where processes are monitored through video/IR imaging [50, 51] and in the analysis of biomedical data [52, 53]. CNNs work on a similar principal, but leverage convolutions as their method of local pattern extraction and perform recursive reductions of the data (e.g., poolings) to extract global patterns [54]. CNNs are covered in greater detail in chapter 6. CNNs provide a method for quantifying complex patterns and structures from data by learning the characteristics of an image that distinguish one class of images (e.g., cats) from another class (e.g., dogs). CNNs are becoming increasingly utilized in chemical engineering research. They have been used in the analysis of complex patterns formed by soft matter systems (e.g., liquid crystals) [55, 56]. They are leveraged in the analysis of hyperspectral imaging and spectroscopy data [57, 55, 58, 59]. They are also applied in the analysis of molecular simulations, and in process fault identification [60, 61, 62, 63]. These methods are often highly accurate and can capture complex patterns/structures in data, but are notoriously difficult to interpret. Another method for summarizing the statistical characteristics of images are spatial correlation functions (e.g., 2-point correlation function) [64]. These functions attempt to summarize the relationships between spatial distance and image (pixel) values. These functions have direct connections to physics and thermodynamics, and are commonly used in the analysis of molecular simulations and for characterizing heterogeneous materials [65, 66, 67, 68, 69]. These methods provide efficient statistical descriptors of data, but do not directly capture topological information (e.g., connectivity)

[70].

We can also interpret images as manifolds with an associated function as shown in Figure 1.2. Manifolds formalize the idea that data can lie in a space that is not governed by Euclidean geometry (i.e., the space is not linear) [10]. Manifolds are covered in detail in chapter 5. Manifold representations of data allow us to capture this non-linear behavior without attempting to constrain or project the data to a linear space. The most prominent example of this methodology is found in the diffusion map method [71]. Here data is assumed to lie in some nonlinear space, and diffusion is simulated in this space in an attempt to reconstruct the non-linear manifold of the data. These methods have been used in chemical engineering data for tasks such as identifying reaction coordinates in dynamical systems and the analysis of molecular dynamics simulations [72, 73, 74, 75]. If the structure of a data manifold is known a priori, it can also be used in tasks such as optimization or system identification where an objective function or solution is constrained to a specified manifold [76, 77, 78, 79].

1.2 Research Objectives

Optimal data representations provide a way to maximize the information and domain knowledge encoded while minimizing the amount of computational resources needed to process and analyze data. A commonality between many of the analysis methods presented in section 1.1 is that they neglect the *topology* and *geometry* of the data. Topology and geometry provide simple and efficient characterizations of data, especially data that is represented as a topological object or shape such as graphs, manifolds, or point clouds which we demonstrate in this dissertation. The objective of this dissertation is to provide researchers with tools and methods from topological and geometrical data analysis (TGDA) and motivate their use through applications to real-world datasets (Figure 1.3).

We focus our applications on a broad range of chemical engineering research areas such as soft matter systems, molecular simulations, process systems, spatio-temporal dy-

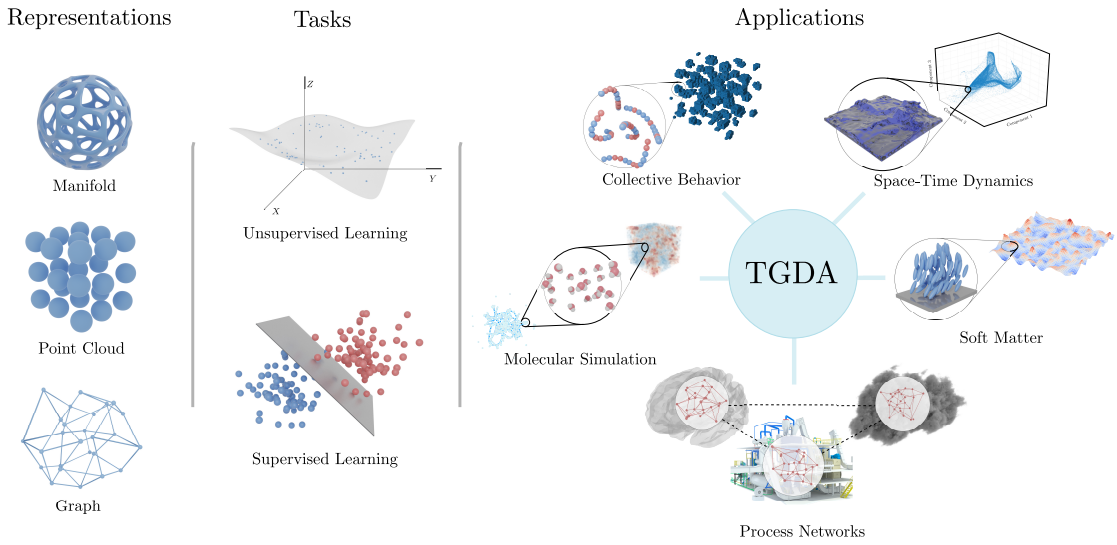


Figure 1.3: Graphical overview of the topological and geometrical data analysis (TGDA) ideas and methodologies presented in this disseration. The goal of TGDA is to identify simple and efficient data representations that capture the physically meaningful characteristics of data which can be integrated in data analysis tasks such as classification and dimensionality reduction.

namical systems, among others. In addition, we completely outline the theoretical and computational aspects of the presented topological and geometrical methods, along with implemented code to reproduce all results in this dissertation.

1.3 Thesis Overview

This dissertation is broken up into three parts. Part I is composed of chapters 2 to 4 and discusses the mathematical foundations and applications of topology based methods for data analysis: the Euler characteristic and persistent homology. Part II contains chapter 5 and covers geometric data analysis methods with a focus on the Riemannian geometry of symmetric, positive-definite matrices (e.g., covariance/correlation matrices). Part III provides a study of data patterns and structure through data driven methods such as (convolutional) neural networks and sparse principal component analysis (chapter 6 and chapter 7). We summarize each chapter below.

In Chapter 2 we study a specific tool known as the Euler characteristic (EC). The EC is a general, low-dimensional, and interpretable descriptor of topological spaces defined by data objects. We review the mathematical foundations of the EC and highlight its connections with statistics, linear algebra, field theory, and graph theory. We discuss advantages offered by the use of the EC in the characterization of complex datasets; to do so, we illustrate its use in different applications of interest in chemical engineering such as process monitoring, flow cytometry, and microscopy. We show that the EC provides a descriptor that effectively reduces complex datasets and that this reduction facilitates tasks such as visualization, regression, classification, and clustering.

Chapter 3 builds upon the work in chapter 2 and provides an in-depth analysis of multiple molecular dynamics (MD) simulation datasets. We show that the Euler characteristic (EC) provides an effective topological descriptor that facilitates MD analysis. We demonstrate the benefits of the proposed approach through case studies that aim to understand and predict the hydrophobicity of self-assembled monolayers and the reactivity of complex solvent environments

In Chapter 4 we introduce a field of research known as Topological Data Analysis (TDA). TDA represents datasets as geometric and topological objects and provides dimensionality reduction techniques that project such objects onto low-dimensional descriptors through algebraic topology. The key properties of these descriptors (also known as topological features) are that they provide multiscale information and that they are stable under perturbations (e.g., noise, translation, and rotation). We also review the key mathematical concepts and methods of TDA and present different applications in chemical engineering.

Chapter 5 explores the use of tools from Riemannian geometry for the analysis of symmetric positive definite (SPD) matrices. An SPD matrix is a versatile data representation that is commonly used in chemical engineering (e.g., covariance/correlation/Hessian matrices and images) and powerful techniques are available for its analysis (e.g., principal component analysis). This work is motivated by the geometry of SPD matrices which live

on a Riemannian manifold. Techniques that exploit the geometry of this manifold can yield significant benefits in data-centric tasks. We demonstrate this via case studies that conduct anomaly detection in the context of process monitoring and image analysis, and an analysis of the spatio-temporal behavior of atmospheric pollutants.

In Chapter 6 we explore a data driven pattern extraction method: convolutional neural networks. Here, we perform a convolutional neural network (CNN) analysis of optical responses of liquid crystals (LCs) when exposed to different chemical environments. Our aim is to identify informative features that can be used to construct automated LC-based chemical sensors and shed some light on the underlying phenomenon that governs and distinguishes LC responses. Our analysis reveals that patterns extracted through the first and second layers of a pre-trained convolutional neural network (VGG16) are sufficient to achieve a perfect classification accuracy of the sensors. We also explore the physical meaning of the extracted patterns and identify color and texture as leading factors in distinguishing LC-based chemical sensor responses.

Finally, in Chapter 7 we present a machine learning framework to explore the predictability limits of catalytic activity from experimental descriptor data (which characterizes catalyst formulations and reaction conditions). We explore the high-dimensional geometry and structure of this dataset through principal component analysis (PCA) and sparse PCA and leverage artificial neural networks to predict reactivity. Furthermore, we propose a constrained-PCA optimization formulation that identifies new experimental points while filtering out regions in the experimental space due to constraints on technology, economics, and expert knowledge. This allows us to navigate the experimental space in a more targeted manner.

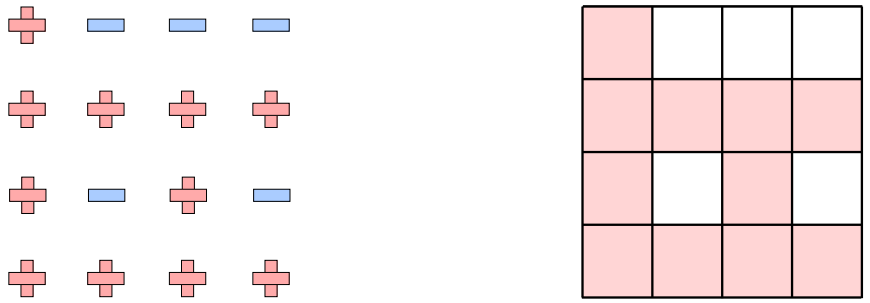
1.4 Motivation: Connecting Physics, Topology, and Geometry

Data analysis in engineering and the physical sciences differentiates itself from data analysis in other fields primarily due to the need for *physical interpretability* of the analysis

results. Therefore, we motivate the research presented in this dissertation through a powerful connection between physics, topology, and geometry, formed through the theorem of 20th century mathematician Hugo Hadwiger known as Hadwiger's Theorem [80]. The theorem provides a means for expressing valuations (e.g., Hamiltonians, energies) of physical systems through the systems geometry and topology [81]. We explore this theorem through a study of the 2-D Ising model [82]. The Ising model, developed in 1920 by Wilhelm Lenz and Ernst Ising, is a simple thermodynamic model of ferromagnetism [83]. The Ising model provides a way to understand the thermodynamic behavior of phase transitions such as symmetry breaking and critical points [82, 84]. In this section we define a direct geometric and topological connection to the 2-dimensional Ising model and understand how topology and geometry can be used to represent the model's thermodynamics.

1.4.1 *A Geometric Ising Model*

We first introduce the original Ising model and propose a geometric interpretation that allows us to study the physical behavior of the model through Hadwiger's theorem.



(a) Lattice Ising Model (b) Geometric Ising Model

Figure 1.4: A comparison of the original Ising model representation (a) and the proposed geometric Ising model representation (b). In the geometric representation positive spin lattice sites are represented by the presence of a square, closed, convex polytope (area = 1) that is centered on the corresponding lattice site. If the spin is negative the square polytope is removed from that lattice point. This representation provides a direct correspondence between the lattice Ising model and the proposed geometric Ising model.

The Ising model consists of discrete variables that represent atomic spin. These vari-

ables σ_i can be in one of two states $\sigma_i \in \{+1, -1\}$. For the 2-dimensional Ising model, these variables are encoded in a square lattice Λ which consists of a set of evenly spaced points $x_i \in \mathbb{Z}^2$ with distance between adjacent points equal to unity. An illustrative example of this model is found in Figure 1.4. Here, each site is represented with a + or - to illustrate the spin state. For a given lattice and set of spin states, we define a spin configuration as:

$$\sigma := (\sigma_i)_{x_i \in \Lambda} \tag{1.1}$$

The Ising model also describes an *energy* function, known as a *Hamiltonian*, that characterizes the total potential and kinetic energy of a given spin configuration σ. The Hamiltonian is given as:

$$H(\sigma) := (-h) \sum_i \sigma_i + (-J) \sum_{\langle i,j \rangle} \sigma_i \sigma_j \tag{1.2}$$

where the notation $\langle i, j \rangle$ indicates that positions x_i, x_j are nearest neighbors on the lattice. J represents the spin-spin interaction of adjacent sites, and h represents the influence of an external magnetic field. The summation over all adjacent lattice sites weighted by J represents the energy associated with interactions between spins, a positive J represents a ferromagnetic system where spins favor alignment. A negative J value represents the opposite. The value h represents the strength of an external magnetic field, where a positive h value shifts the overall spin state to be positive, and vice versa with a negative h value.

We have defined the Ising model along with a Hamiltonian that describes the energy of a given spin state. We now propose a different abstraction of the Ising model that represents the model as a set of square areas s_i in a grid $s_i \in S$, rather than points on a lattice. An illustration of this concept is found in Figure 1.4. Each grid point represents a

unit area square with width and height equal to unity. The grid is formed such that the centroid of each square on the grid is equal to a point $x_i \in \Lambda$. We also extend our binary spin function to this geometric model by considering the grid as a collection of convex polytopes (i.e., filled squares), where a convex square polytope (area = 1) is centered over a lattice point $x_i \in \Lambda$ if the spin of the corresponding lattice site is +1. If the spin of the site is -1, we remove the polytope from that position and consider it an empty space. Thus, for a given lattice Λ and spin configuration σ we obtain a collection of square polytopes. Through this abstraction we can now apply the theorem of Hugo Hadwiger and understand the connections between the geometry and topology of this collection of polytopes and the energy of the Hamiltonian.

1.4.2 *Hadwiger's Theorem*

Hadwiger's theorem is focused on collections of closed and bounded convex sets $\mathcal{K}_i \in \mathbb{K}^n$, where $i = 1, 2, ..., N$, N is the total number of sets in the collection, and $\mathbb{K}^n$ is the collection of all compact convex sets in $\mathbb{R}^n$ (e.g., a collection of convex square polytopes in our geometric Ising model). A given configuration of these convex sets is denoted as $\mathcal{K}^N = \bigcup_{i=1}^{N} \mathcal{K}_i$. Hadwiger's theorem states that if there exists a *continuous*, *rigid motion invariant*, and *additive* valuation $f : \mathcal{K}^N \to \mathbb{R}$, then this valuation can be represented as a weighted sum of the *intrinsic volumes* (also known as *Minkowski functionals*) of the configuration $\mathcal{K}^N$ [85].

A *valuation* is defined as a mapping $f : \mathbb{K}^n \to \mathbb{R}$ such that $f(\emptyset) \to 0$ and for every set $S, T \in \mathbb{K}^n$ that satisfies $S \cup T \in \mathbb{K}^n$ we have that the valuation is additive if:

$$f(S \cup T) = f(S) + f(T) - f(S \cap T) \tag{1.3}$$

Furthermore, Hadwiger's theorem requires that the valuation of the configuration is *rigid motion invariant*, and *continuous*. Rigid motion invariance means that the valuation of

a configuration $f(\mathcal{K}^N) \to \mathbb{R}$ is independent of the configuration's position and orientation in space. If we consider $\mathcal{G}$ to be the group of rigid translational and rotational motions (i.e., the group of Euclidean isometries), such as the set of orthogonal matrices $\mathcal{M} := \{\mathbf{M}, \mathbf{M}^T\mathbf{M} = \mathbf{I}\}$ where $\mathbf{I}$ is the identify matrix, we have that:

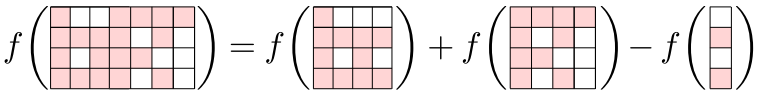
$$f(g\mathcal{K}^N) = f(\mathcal{K}^N) \tag{1.4}$$

where $g \in \mathcal{G}$. Finally, we have that the valuation must be continuous. This means that given a configuration $\mathcal{K} \in \mathbb{K}^n$ and a sequence of configurations $\mathcal{K}_m$ such that the sequence $\mathcal{K}_m \to \mathcal{K}$ as $m \to \infty$ we have that the valuation:

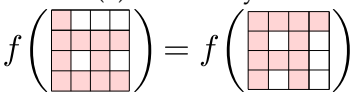
$$f(\mathcal{K}_m) \to f(\mathcal{K}) \tag{1.5}$$

In other words, an approximation of a configuration will reflect an approximation in the valuation. This is especially important when we are considering, for example, image datasets where pixels (convex squares) are used to approximate more complex objects captured in an image. Illustrations of each of these concepts can be found in Figure 1.5.

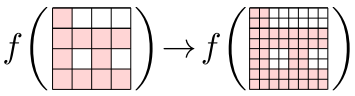
If a valuation $f : \mathcal{K}^N \to \mathbb{R}$ is *additive*, *continuous*, and *ridig motion invariant*, then Hadwiger's theorem states that this valuation can be expressed as the weighted sum of the *intrinsic volumes* of $\mathcal{K}^N$ [86]. Intrinsic volumes (i.e., quermassintegrales, curvature integrals, Minkowski functionals) are concepts from integral geometry and are defined as integrals of curvature over a surface [81]. There are $d + 1$ intrinsic volumes for a d-dimensional surface [81]. Thus, in the case of our geometric 2-dimensional Ising model, we will obtain 3 intrinsic volumes, which we denote M_0, M_1, and M_2. A rigorous definition of these concepts can be found in the following references [81, 87, 86, 88]. Fortunately, for objects of 3 or less dimensions, these intrinsic volumes have a direct relationship to common geometric and topological quantities. The relationships to these quantities for a



(a) Additivity



(b) Rigid motion invariance



(c) Continuity

Figure 1.5: Illustration of the three valuation $f : \mathcal{K}^N \to \mathbb{R}$ requirements for Hadwiger's theorem: (a) additivity, (b) rigid motion invariance, and (c) continuity. If these requirements are met, the valuation can be represented as a weighted summation of the intrinsic volumes of the configuration $\mathcal{K}^N$.

Table 1.1: Intrinsic volumes and their geometric and topological quantities.

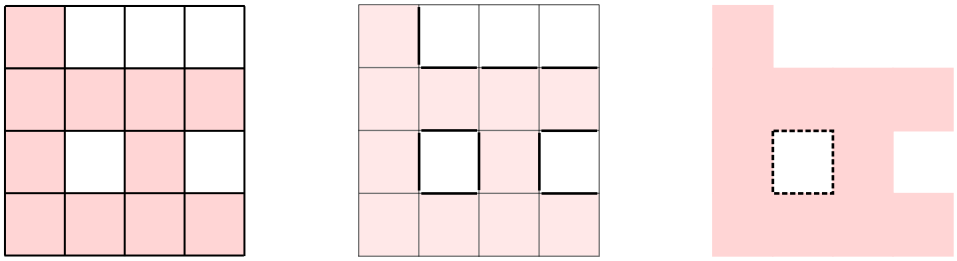
Intrinsic Volume M_i	Geometric/Topological Quantity
$M_0(\mathcal{K}^N) = A(\mathcal{K}^N)$	2-dimensional area A of $\mathcal{K}^N$
$M_1(\mathcal{K}^N) = U(\mathcal{K}^N)$	Boundary length (perimeter) U of $\mathcal{K}^N$
$M_2(\mathcal{K}^N) = \chi(\mathcal{K}^N)$	Euler Characteristic χ of $\mathcal{K}^N$

2-dimensional collection $\mathcal{K}^N \subset \mathbb{R}^2$ can be found in Table 1.1 [70].

M_0 of a collection $\mathcal{K}^N \subset \mathbb{R}^2$ is the area, M_1 is the boundary length (perimeter) of $\mathcal{K}^N$, and the final intrinsic volume M_2 is the Euler characteristic χ of $\mathcal{K}^N$. The M_0 and M_1 intrinsic volumes are geometric characteristics of the collection, where the third volume M_2 is a topological characteristic. The two geometric intrinsic volumes M_0 and M_1 are simple to interpret. The third intrinsic volume M_2, representing the Euler characteristic χ of $\mathcal{K}^N$, is a topological descriptor of the shape formed by $\mathcal{K}^N$. The Euler characteristic for our 2-dimensional shape $\mathcal{K}^N$ is an alternating sum of the number of 0-dimensional topological bases (known as connected components) and the number of 1-dimensional topological bases (known as holes):

$$\chi = \#\text{ Connected Components} - \#\text{ Holes} = \beta_0 - \beta_1 \tag{1.6}$$

where the rank of the first basis is known as the 0-th Betti number $\beta_0 \in \mathbb{Z}_+$, and the rank of the second basis is known as the 1-st Betti number $\beta_1 \in \mathbb{Z}_+$; here, $\mathbb{Z}_+$ denotes the set of all nonnegative integers. These bases represent the *topological invariants* of the collection $\mathcal{K}^N$, which are the characteristics of the collection that are unchanged when the collection is continuously deformed (e.g., stretched, twisted, but not cut or torn). The Euler characteristic is covered in more detail in Chapter 2. We provide an illustration of the computation of the intrinsic volumes for the geometric Ising model configuration (shown in Figure 1.4) in Figure 1.6. We note here that we are not considering boundary conditions for purposes of illustration.



(a) $M_0 = 11$ (Units^2) (b) $M_1 = 11$ (Units) (c) $M_2 = 1 - 1$

Figure 1.6: Illustration of the computation of Area (a), Boundary (b), and Euler characteristic (c) for a given configuration $\mathcal{K}^N \subset \mathbb{R}^2$.

Thus, for our 2-dimensional geometric Ising model we have that a configuration $\mathcal{K}^N$ with a continuous, additive, rigid motion invariant valuation $f : \mathcal{K}^N \to \mathbb{R}$ can be represented as follows:

$$f(\mathcal{K}^N) = c_0 M_0(\mathcal{K}^N) + c_1 M_1(\mathcal{K}^N) + c_2 M_2(\mathcal{K}^N) \tag{1.7}$$

where $c_0, c_1, c_2 \in \mathbb{R}$ are constant values.

1.4.3 *Ising Hamiltonian*

We can now develop a new Hamiltonian $H_G : \mathcal{K}^N \to \mathbb{R}$ for the geometric Ising model that is a valuation defined in terms of a configuration's geometry and topology through the results of Hadwiger's theorem. We follow the convention of the original Ising model and represent the coefficients with a negative value:

$$H_G(\mathcal{K}^N) = \underbrace{(-c_0)M_0(\mathcal{K}^N)}_{\text{Covered Area}} + \underbrace{(-c_1)M_1(\mathcal{K}^N)}_{\text{Total Boundary}} + \underbrace{(-c_2)M_2(\mathcal{K}^N)}_{\text{Euler Characteristic}} \tag{1.8}$$

We can compare this newly defined Hamiltonian to the original Hamiltonian for the Ising model:

$$H(\sigma) = \underbrace{(-h)\sum_i \sigma_i}_{\text{External Field}} + \underbrace{(-J)\sum_{\langle i,j \rangle} \sigma_i \sigma_j}_{\text{Spin-Spin Interaction}} \tag{1.9}$$

We find that there are several similarities between the two models. In particular, there is a direct relationship between the influence of an external field $(-h)\sum_i \sigma_i$ in the $H(\cdot)$ model and the covered area $(-c_0)M_0(\mathcal{K}^N)$ in the $H_G(\cdot)$ model. A positive h value will bias the Ising model to have positive spin and a positive c_0 value will bias the model to have a larger covered area. Vice-versa for negative values in both cases.

There is also a relationship between the energy of spin-spin interactions $(-J)\sum_{\langle i,j \rangle} \sigma_i \sigma_j$ and the total boundary (perimeter) of the geometric model $(-c_1)M_1(\mathcal{K}^N)$. The formation of a boundary in the geometric model represents the presence of adjacent spin states that are different, whereas the absence of a boundary represents adjacent spin states that are identical. This relationship was originally discovered by Rudolph Peierls in his argument for the spontaneous formation of droplets (defects) in the 2-dimensional Ising model [89]. Here, droplets represent connected areas of a particular spin state. Peierls showed that

the energy of these droplets (defects) is exactly related to the boundary (perimeter) of the droplet [90, 91].

We have identified two direct connections between our proposed geometric model Hamiltonian $H_G(\cdot)$ with the original Hamiltonian $H(\cdot)$ proposed by Lenz and Ising. However, we note that there is no direct consideration for the topology of the spin states in the original Ising model Hamiltonian, whereas the proposed geometric model clearly identifies the Euler characteristic as playing an important role in the energy of a given spin configuration. The influence of $(-c_2)M_2(\mathcal{K}^N)$ on the energy of the system is directly related to the dominant topology of the spin configuration. For example, we conduct multiple metropolis monte-carlo simulations of this newly proposed Hamiltonian $H_G(\cdot)$ to illustrate some of the interesting configurations that can be formed through the addition of topological information via the Euler characteristic shown in Figure 1.7. Here, we explore the energy of a simulation as we adjust the parameter c_2 of our Hamiltonian while keeping the temperature constant and c_0, c_1 constant and positive. We see that as the sign and weight of c_2 change, the overall topology of the system becomes dominated with either connected components (c_2 is positive, Euler characteristic is positive) or holes (c_2 is negative, Euler characteristic is negative). We also see that the shape of the resulting plot suggests the presence of a critical point (phase change) associated with the Euler characteristic.

1.4.4 *Concluding Remarks*

The behavior exhibited by the model in Figure 1.7 is similar to that of many physical systems. For example, we can think of the behavior of colloidal systems (e.g. polymers, surfactants) as being modeled by this form of Hamiltonian. At one extreme (c_2 is positive), the colloidal system would exhibit behavior that maximizes connected components and minimizes cycles, this would result in the formation of objects such as micelles [92]. The opposite would occur for the other extreme (c_2 is negative) this would result in the

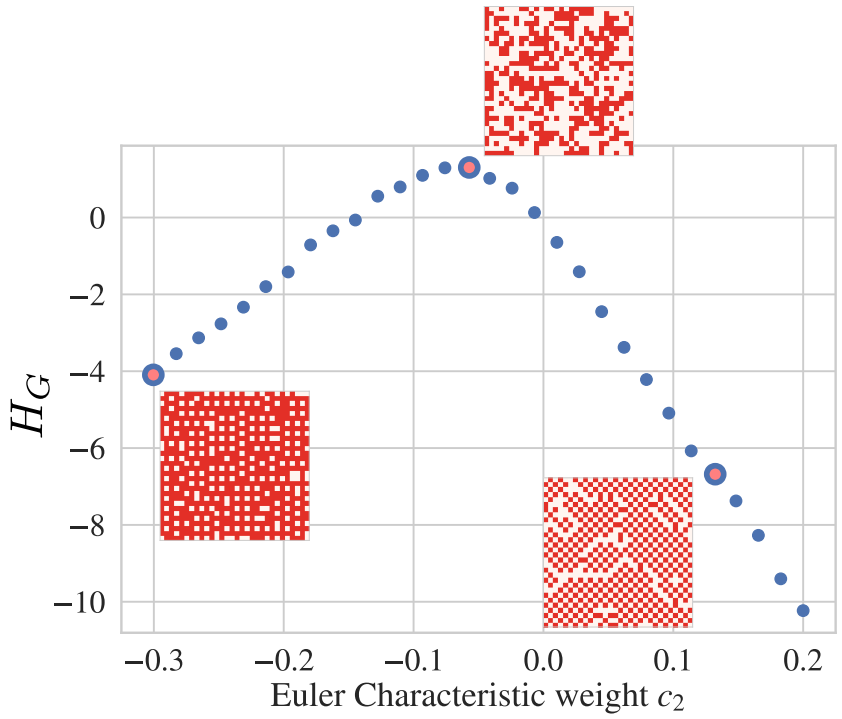


Figure 1.7: Results of the total energy derived from multiple metropolis monte-carlo simulations of the geometric Ising model using the $H_G(\cdot)$ Hamiltonian while varying the magnitude and sign of the coefficient for the Euler characteristic c_2. Final states for the monte-carlo simulation are also shown. The behavior exhibited by the model is similar to that of many physical systems (e.g. colloids, polymers). A negative c_2 induces the formation of a single connected network structure with many holes, similar to the behavior of many gel systems. With a large positive c_2 we see the formation of many connected components, similar to what might be seen in the formation of micelles. The energy profile also illustrates the presence of a critical point and phase change associated with the Euler characteristic.

formation of a single connected network structure with many holes which are found in gel systems [93]. These geometric and topological descriptors of energy have also been applied in characterizing the curvature energy of membranes [94], in quantifying the impact of fluid morphology on capillary condensation [95], and in understanding the large scale structure of the universe [96]. Thus, topology and geometry provide a powerful measure of the physical state of a system and its energy through the connections made by Hadwiger's theorem. The incorporation of topological and geometric measures of data, especially in the field of Chemical Engineering, can provide researchers with new ways to understand the behavior of their systems and uncover relationships and emergent phenomena that would be otherwise impossible to measure or simulate.

Part I

TOPOLOGY

Chapter 2

THE EULER CHARACTERISTIC

The content of this chapter is published in [31]

2.1 Introduction

Datasets are mathematical objects (e.g., point clouds, matrices, graphs, images, field/functions) that have shape [97]. Characterizing the shape (geometrical features) of these objects reduces the dimensionality and complexity of the data while minimizing information loss, but is not always straightforward [98]. Popular tools from statistics, linear algebra, and signal processing (e.g., moments, correlation functions, singular value decomposition, convolutions, Fourier analysis) do not *directly* characterize geometrical features of data objects; instead, such tools are used to characterize other types of features (e.g., variance and frequency content).

Topology is a branch of mathematics that provides powerful tools to characterize the shape of data objects. One such tool is the so-called Euler characteristic (EC); the EC, originally used for the characterization of polyhedra [99], is now broadly used in scientific areas such as random fields [100, 101, 102], cosmology [103, 104, 105], material science [94, 106, 107, 108], thermodynamics [109, 110, 111], and neuro-science [112, 113]. To the best of our knowledge, the EC has seen limited applications in engineering and of these

applications most are focused on the characterisation of the permeability of porous media and modelling geometric states of fluids within porous media [108, 114]. However, a fact that is often overlooked in this literature is that the EC provides a *general* descriptor of different types of topological spaces (this enables the characterization of a much wider range of data objects). This generality arises from the fact that (i) one can use transformations to map a data object into another type of object and (ii) the EC has fundamental connections with statistics, field theory, linear algebra, and graph theory.

In a nutshell, the EC is a descriptor that characterizes geometrical features of a topological space defined by a data object. This characterization is accomplished by performing a decomposition of the space into a set of independent *topological bases*. This decomposition is similar in spirit to an eigen-decomposition of a matrix; here, the matrix object is decomposed into a set of independent basis vectors. The EC is a scalar integer quantity that is defined as the alternating sum of the rank of the topological bases. The EC is often combined with a transformation technique known as *filtration* to characterize the geometry of different objects such as matrices, images, fields/functions, and weighted graphs. This characterization is summarized in the form of what is called an *EC curve* which provides a direct approach to *quantify the topology* of an object.

Topological descriptors such as the EC offer advantages over statistical descriptors [94]. For instance, statistical descriptors such as Moran's I, which measures spatial structure via spatial autocorrelation, or correlation matrices do not directly capture the global structure of the data (thus limiting the ability to characterize geometrical features) [115, 116]. High-order statistical descriptors such as 2-point correlation functions, which have been employed in characterizing the structure of heterogeneous materials, are also limited at capturing spatial and morphological features of the data (especially if the data object is irregular) [117, 70]. However, there exists well-developed theory that connects the EC to the geometry of random fields [101, 102]. Such work establishes that the EC encodes information of simple statistical descriptors such as means and variances and of more complex descriptors (e.g., space-time covariances) [100]. These connections between

topology and statistics are powerful and provide a mechanism to understand the emergence of topological features from physical behavior (e.g., diffusion phenomena). The EC also connects with concepts from linear algebra and graph theory; for example, a matrix or an image can be represented as a weighted graph and the geometry of such graph can be quantified using the EC (a graph is a 2D polyhedron). Establishing these connections is important because data objects encountered in practice are often complex and require the combined use of different characterization techniques. For instance, in brain analysis, one often characterizes a multivariate time series (a collection of time series obtained from different locations in the brain) by constructing a correlation matrix. The topology of this matrix is then reduced to an EC curve through a filtration. The EC curve provides a low-dimensional descriptor that characterizes the spatio-temporal structure of the brain. It is also important to emphasize that the EC curve is a much simpler topological descriptor than the so-called *persistence diagrams* used in most of the topological data analysis (TDA) literature [97]. Persistent diagrams contain more topological information but are more difficult to analyze and interpret.

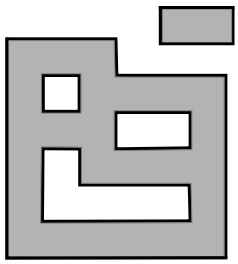
The aim of this chapter is to present an applied perspective on the EC. We briefly discuss the mathematics of the EC and discuss how to use filtration operations to characterize diverse data objects. This discussion will establish connections with field theory, graph theory, and linear algebra. We then bring our focus to applying these concepts to tackle diverse problems arising in science and engineering; in particular, we discuss how the EC can be used in process monitoring by analyzing correlation structures. We also apply the EC in the analysis of both 2D spatial and 3D spatio-temporal fields; these data objects are derived from reaction-diffusion partial differential equations (PDEs), micrographs of liquid crystals, and flow cytometry. In these examples, we show how to use the EC as a data pre-processing (reduction) step that can facilitate machine learning tasks. We also provide scripts and datasets to help the interested reader apply these tools.

2.2 Introduction to the Euler Characteristic

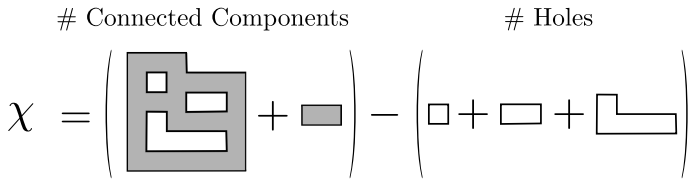
Around 1735, Euler discovered a relationship between the number of vertices, edges, and faces of a convex polyhedron (which is now known as the EC). The study and generalization of this formula, specifically by Cauchy and L'Huilier, is at the origin of topology. The EC is a scalar integer value that summarizes the shape of a topological space (an object). A topological space is a set with a structure defined by continuity and connectedness which also represents the ideas of limits or closeness based on relationships between the sets of the space rather than a specific distance or metric [118] . Topological spaces are a central unifying notion that appears in virtually every branch of modern mathematics (e.g., capture graphs and manifolds). The EC is a topological invariant quantity (e.g., does not change with deformations such rotation, streching, bending). For instance, the topology of a graph is fully defined by its node-edge connectivity (the location of the nodes and edges is irrelevant); as such, graphs that appear to be visually distinct might have the same underlying topology (and thus have the same EC value). In this chapter, we will introduce the EC from the perspective of data objects that can be represented as graphs (a 2D polyhedron) and manifolds (e.g., images and fields/functions). Graphs and manifolds are special types of topological spaces but, as we will see, these spaces are sufficient to represent a vast number of data objects encountered in chemical engineering applications such as the analysis of industrial chemical process sensor data (graph) and soft material experimental images (manifold) .

We begin with a general definition of the EC (which we denote as $\chi \in \mathbb{Z}$) for a 2D manifold (see Figure 2.1). The EC of the manifold is an alternating sum of the number of 0-dimensional topological bases (known as connected components) and the number of 1-dimensional topological bases (known as holes):

$$\chi = \#\,\text{Connected Components} - \#\,\text{Holes} = \beta_0 - \beta_1 \tag{2.1}$$



(a) Example object (2D shape)



(b) EC of example object (χ)

Figure 2.1: Illustration on how topology can be quantified via the Euler characteristic. (a) A 2D shape that has two connected components and three holes. (b) The EC is an alternating sum of the number of connected components and holes and thus $\chi = -1$.

There are only two sets of topological bases (connected components and holes) that describe a 2D object. The rank of the first basis is known as the 0-th Betti number $\beta_0 \in \mathbb{Z}_+$, while the rank of the second basis is known as the 1-st Betti number $\beta_1 \in \mathbb{Z}_+$; here, $\mathbb{Z}_+$ denotes the set of all nonnegative integers. The relationship between the number of topological bases and dimensions holds for n-dimensional shapes; as such, the EC of an $(n+1)$-dimensional shape is given by:

$$\chi = \sum_{i=0}^{n} (-1)^i \beta_i \tag{2.2}$$

To simplify our notation, we will use the generalized topological representation of the *Betti numbers* $\beta_n \in \mathbb{Z}_+$; here, the n-th Betti number β_n is the number of unique n-dimensional topological bases for a given shape [100].

2.3 EC for Graphs

A graph object $G(V, E)$ is a 2D topological space (a 2D polyhedron). The EC of this object is given by:

$$\chi = |V| - |E| = \beta_0 - \beta_1 \tag{2.3}$$

where $|V| \in \mathbb{Z}_+$ is the number of graph vertices (nodes) and $|E| \in \mathbb{Z}_+$ is the number

of graph edges. We define $v(e), v'(e) \in V$ as the support nodes of edge $e \in E$. One can show that $|V| - |E|$ equals the number of connected components of the graph (β_0) minus the number of holes (cycles) of the graph (β_1) [119].

One can use the EC to characterize *edge-weighted graphs* $G(V, E, w_E)$, where each edge $e \in E$ has an associated scalar weight $w_E(e) \in \mathbb{R}$. Similarly, one can use the EC to characterize *node-weighted graphs* $G(V, E, w_V)$, where each node $v \in V$ has an associated scalar weight $w_V(v) \in \mathbb{R}$. This characterization is done via a process known as a *filtration* which leads to the creation of a topological descriptor known as an *EC curve*. The ability to deal with weighted graphs enables analysis of data objects that are represented as *discrete fields*, such as matrices and images. For example, a correlation matrix (a square and symmetric matrix) can be represented as an edge-weighted graph in which the nodes are the random variables, the edges are the connections between variables, and the weights are the degrees of correlation between pairs of random variables (Figure 2.2). A grayscale image can be represented as a node-weighted graph in which the nodes are pixel locations, the weights are the intensity of the pixels, and the edges connect adjacent pixels to form a grid (Figure 2.3). A 2D discrete field (e.g., obtained from a discretized PDE) can also be represented as a node-weighted graph; here, the nodes are locations in the discretization mesh and the weights are values of a variable of interest at such locations (e.g., temperature). It is important to emphasize that graphs are topological spaces that do not live in a Euclidean space; as such, the location of the nodes and edges is irrelevant (topology is fully dictated by the node-edge connectivity). Thus, the EC and associated EC curve are focused on the global topology of the graph during a filtration, and not on specific connectivity information (e.g. the number of edges connected to node x) which it does not track.

To characterize the topology of an edge-weighted graph $G(V, E, w_E)$, we perform a filtration of the graph by eliminating edges that have weights less than or equal to a certain threshold $w_E(e) \leq \ell$ (with $\ell \in \mathbb{R}$). Filtration gives a graph that is sparser than the original graph and that has an associated EC value. We can repeat the filtering process

by progressively increasing the threshold value ℓ and with this obtain new graphs and associated EC values. One stops the process once the threshold reaches the largest weight in the original graph $G(V, E, w_E)$; this gives the original graph itself and and its associated EC value. To formalize the filtration process, we define the following *filtration function* (see Figure 2.2).

Definition 2.3.1. Graph Edge Filtration Function (f_E): For an undirected edge-weighted graph $G := G(V, E, w_E) \in \mathbb{G}$ with scalar edge weight values $\{w_E(e) \in \mathbb{R} : e \in E\}$ the filtration function $f_E : \mathbb{G} \to \mathbb{R}$ is defined such that $f_E(G) = \max_{e \in E} w_E(e)$. The pre-image $f_E^{-1}(\ell)$, with $\ell \in \mathbb{R}$, is given by the graph $G_\ell := G(V, E_\ell, w_{E_\ell})$ where $E_\ell = \{e \in E : w_E(e) \leq \ell\}$.

The pre-image of the filtration function is used to create a set of nested graphs; this is done by defining a sequence of increasing thresholds $\ell_1 < \ell_2 < ... < \ell_m$ with associated graphs:

$$G_{\ell_1} \subseteq G_{\ell_2} \subseteq ... \subseteq G_{\ell_m} \subseteq G \tag{2.4}$$

Here, we note that $G_{\ell_m} = G$ if $\ell_m = \max_{e \in E} w_E(e)$ (the last graph in the filtration is the original graph). We also re-emphasize that the density of the graph (its number of edges) increases with the threshold value; specifically, G_{ℓ_1} is the graph with lowest density (highest sparsity) and G_{ℓ_m} is the graph of highest density (lowest sparsity).

We can define a similar filtration for node-weighted graphs; here, nodes are filtered/eliminated based on their weight values. The *filtration function* for this object (see Figure 2.3) can be defined as follows.

Definition 2.3.2. Graph Node Filtration Function (f_V): For an undirected node-weighted graph $G := G(V, E, w_V) \in \mathbb{G}$ with scalar node weight values $\{w_V(v) \in \mathbb{R} : v \in V\}$ the filtration function $f_V : \mathbb{G} \to \mathbb{R}$ is defined such that $f_V(G) = \max_{v \in V} w_V(v)$. The pre-image $f_V^{-1}(\ell)$ is given by the graph $G_\ell := G(V_\ell, E_\ell, w_{V_\ell})$ where $V_\ell = \{v \in V : w_V(v) \leq \ell\}$ and $E_\ell = \{e \in E : v(e), v'(e) \in V_\ell\}$.

As in the edge-weighted case, the pre-image of the node filtration function is used to create a set of nested graphs:

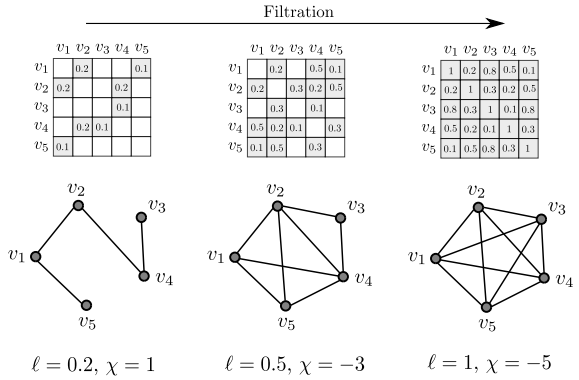
$$G_{\ell_1} \subseteq G_{\ell_2} \subseteq ... \subseteq G_{\ell_m} \subseteq G \tag{2.5}$$

Here, we highlight that node filtration has the effect of eliminating both nodes and edges; in other words, if a node is eliminated, we also eliminate its supported edges. We again note that $G_{\ell_m} = G$ if $\ell_m = \max_{v \in V} w_V(v)$ (the last graph in the filtration is the original graph).

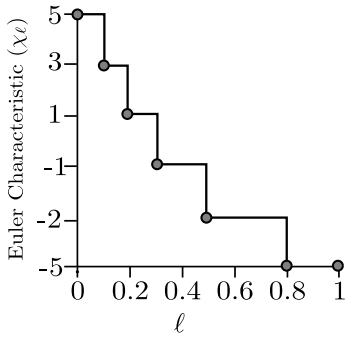
For each graph G_ℓ in the node or edge filtration process, we compute and record its EC value χ_ℓ . This information is used to construct the so-called EC curve, which contains the pairs (ℓ, χ_ℓ). The EC curve thus provides a topological descriptor for edge- or node-weighted graphs. Furthermore, it is important to note that the edge and/or node weights are a critical component of this analysis as this is what allows for the filtration to be performed. Without this information only a single EC value can be computed which is ineffective at distinguishing graphs with different topologies if they have the same ratio of nodes and edges.

2.4 EC for Manifolds

The EC can also be used for characterizing the geometry of topological spaces known as *manifolds* [120, 121, 122]. Specifically, we consider an n-dimensional manifold M that is defined by a *chart* (X, f), where $X \subseteq \mathbb{R}^n$ is an open subset of M (a Euclidean space) and the field/function $f : X \to \mathbb{R}$ is a homeomorphism (a one-to-one, onto, and continuous mapping with an inverse mapping that is also continuous). In simple terms, the manifolds of interest involve an n-dimensional *continuous* domain X and field $f : X \to \mathbb{R}$. We will say that field f is n-dimensional if $X \subseteq \mathbb{R}^n$ and we say that this field is embedded in an $(n+1)$-dimensional manifold (because the dimension of the chart (X, f) is $n+1$). For

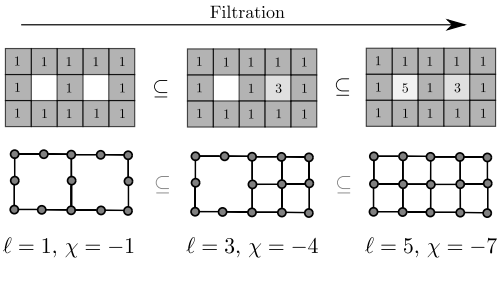


(a) Edge weighted graph filtration.

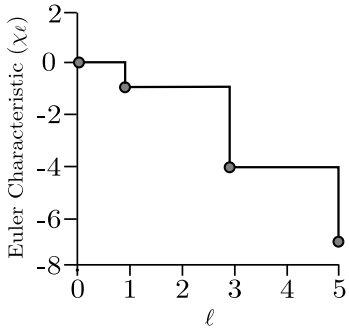


(b) EC Curve for the graph filtration.

Figure 2.2: (a) Filtration of a correlation matrix, which is represented as an edge-weighted graph. The random variables are treated as nodes, and the correlation between the variables are treated as edge weights $w_E(e)$. As the level set of the filtration function f_E increases in value, edges that are below the given threshold are added into the resulting graph f_E^{-1}, revealing the topology of the correlation matrix over multiple correlation thresholds. (b) Resulting EC curve for the filtration of the edge-weighted graph in (a).



(a) Filtration of node-weighted graph (image).



(b) EC Curve for the graph filtration.

Figure 2.3: (a) Filtration of a simple grayscale image, which is represented as a node-weighted graph, with the graph node filtration function f_V. (b) Resulting EC curve for the filtration of the grayscale image in (a).

instance, in Figure 2.4, we show a 1-dimensional (1D) field/function $f : X \to \mathbb{R}$ with domain $X \subseteq \mathbb{R}$; this field is embedded in a 2D manifold M defined by the chart (X, f). In our 1D field example, the horizontal coordinate of the chart is X and the vertical coordinate is f. A 1D field arise, for instance, as the solution of an ordinary differential equation (e.g., the domain is defined by a time coordinate). Another example of a 1D field is the probability density function of a univariate random variable. Similarly, a 2D field $f : X \to \mathbb{R}$ with $X \subseteq \mathbb{R}^2$ is embedded in a 3D manifold M (given by a 3D chart). Formalizing these definitions is important in characterizing the topology of fields. Higher dimensional fields arise, for instance, as solutions of PDEs (e.g., the domain is defined by space-time coordinates) or can be probability density functions of mutivariate random variables. These high-dimensional fields often have complex topology and are notoriously difficult to analyze/characterize.

The chart (X, f) is also often referred to as the *graph of field* f. For $n = 2$ (2D), the chart (X, f) is analogous to the concept of a node-weighted graph $G(V, E, w_V)$ arising in graph theory, in the sense that the domain X captures spatial locations (the nodes V) and the function f captures values at the spatial locations (the node weights w_V). This analogy becomes clearer when the domain X is a 2D box and $G(V, E, w_V)$ has a mesh topology; here, the mesh can be seen as a discrete approximation of the continuous domain X. Mesh topologies arise in images, matrices, and discretized PDEs. We will define the chart associated with a given manifold using the notation $G(X, f)$ and we will refer to this as a *field graph*. This definition introduces some abuse of notation (analogous to $G(V, E, w_V)$) but we do this in order to emphasize connections between field graphs and node-weighted graphs. This will facilitate the explanation of the concept of filtration in a field context. We also emphasize that the concept of a field graph generalizes to arbitrarily high dimensions (while a graph is inherently a 2D object). We also emphasize that a graph does not live in a Euclidean space (while X does). The fact that X is a Euclidean space indicates that there is a notion of order.

A filtration can be applied to continuous fields in the same way that it is applied to

node-weighted graphs (discrete case); however, this now requires a filtration function that is defined over continuous domains. We can define a filtration function via *superlevel sets* of the field [123].

Definition 2.4.1. Superlevel Set: Given a manifold M with field $f : X \to \mathbb{R}$ and domain $X \subseteq \mathbb{R}^n$, the *superlevel set* X_ℓ at a threshold $\ell \in \mathbb{R}$ is defined as:

$$X_\ell = \{x \in X : f(x) \geq \ell\}. \tag{2.6}$$

The super level set is has an associated field graph $G_\ell := G(X_\ell, f_\ell)$ with f_ℓ defined over X_ℓ.

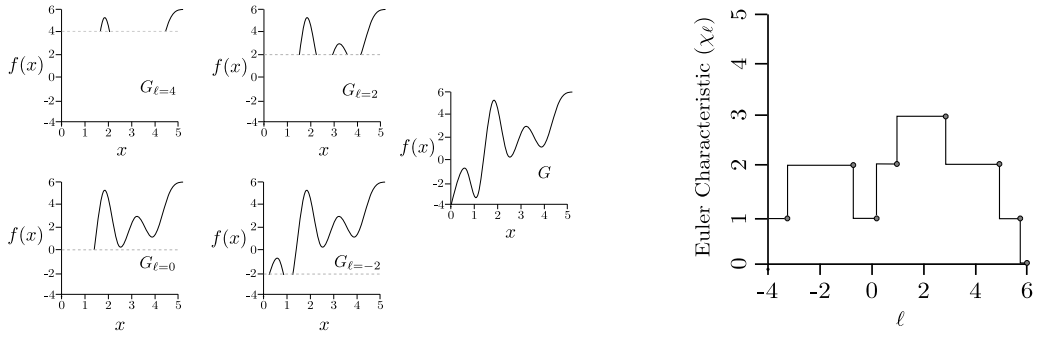
The field graph $G_\ell = (X_\ell, f_\ell)$ contains all points of the manifold M that have a function value greater than or equal to ℓ (it is a filtration of the manifold). Similar to the node-weighted graph case, the filtration creates a nested set of field graphs which are obtained by defining a sequence of decreasing filtration values $\ell_1 > \ell_2 > ... > \ell_m$ with associated field graphs:

$$G_{\ell_1} \subseteq G_{\ell_2} \subseteq ... \subseteq G_{\ell_m} \subseteq G \tag{2.7}$$

Here, the field graphs are sparser with larger threshold values; we also have that $G_{\ell_m} = G(X, f)$ (the original graph) if $\ell_m = \min_{x \in X} \ f(x)$. For each superlevel set we obtain the field graph G_ℓ and we compute and record its EC value χ_ℓ (e.g., we determine the number of connected components and number of holes). This information is used to construct an EC curve, which contains the pairs (ℓ, χ_ℓ).

It is important to highlight that the EC curve provides a topological descriptor for a general n-dimensional field. The types of topological bases change with dimension; for instance, for a 1D field (e.g., a temporal field) we only have connected components ($\chi = \beta_0$) while for a 2D field (e.g., a space-time field) we have connected components and holes ($\chi = \beta_0 - \beta_1$). It is also often convenient to track the evolution of the individual Betti numbers through the filtration; in other words, we keep track of the pairs (ℓ, β_ℓ).

In Figure 2.4, we present multiple superlevel sets of a 1D field and the corresponding EC curve. Since this is a 1D field, the EC of a given superlevel set only involves the number of connected components. Filtration captures topology by revealing a local maximum (connected component is formed) or a local minimum (two components are connected); thus, the EC compactly encodes information about the *critical points* of the field and their relations with respect to the function shape, which are the topologically interesting characteristics of a continuous function or field. *The critical points of fields are key* because they define their topological features.



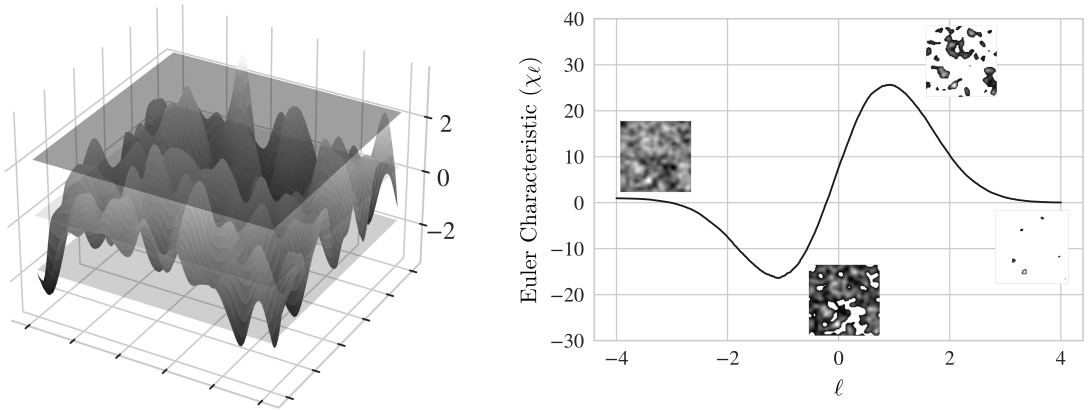
(a) Superlevel set filtration

(b) EC curve

Figure 2.4: (a) Superlevel set filtration of a 1D field (embedded in a 2D manifold). The horizontal lines represent the thresholds of the filtration and these cut through the 2D field graph. The topology of the graphs G_ℓ captures the critical points of the field. When a local maximum is passed, a new connected component is formed (β_0 increases). When a local minimum is passed, two separate components are joined into one component (β_0 decreases). (b) Resulting EC curve for the filtration of the function in (a). The EC curve captures the location of critical points in the function and their relationships via the shape of the function.

Superlevel set filtration is generalizable and extends to higher dimensional fields. Such fields appear in scientific areas such as geophysics [124, 125, 126], climatology [126, 127], astrophysics [128], and medical imaging [129, 130, 131, 132]. Fields can be used to represent many important data objects such as images (2D fields embedded in a 3D manifold), volumes (3D fields embedded in a 4D manifold), and spatio-temporal fields obtained from PDEs (4D fields embedded in a 5D manifold). Theoretical connections between integral geometry, statistics, and topology have shown that the EC provides

a general descriptor to characterize the behavior of these complex data objects, such as identifying higher order statistical characteristics of these data objects (e.g. statistics of derivatives of the data) or capturing global properties of the data, such as the magnitude and frequency of spatial or temporal fluctuations, without the need for assumptions such as isotropy or stationarity [133, 100].



(a) Superlevel set filtration

(b) EC curve

Figure 2.5: (a) Superlevel set filtration of a 2D field (embedded in a 3D manifold). The plane represents the threshold of the filtration and this cuts through the 3D field graph. As the filtration threshold is passed from top to bottom, the EC of the resulting field graph G_ℓ is computed. (b) The EC curve constructed from the filtration of the field. The 2D fields capture the evolution of the topology of the varying superlevel sets during the filtration (note emergence of connected components and holes).

The filtration of a high-dimensional field is analogous to that used in the 1D case. The difference is the number and nature of the topological features captured in higher dimensions. For instance, for a 2D field (embedded in a 3D manifold), we capture both connected components and holes. Specifically, the threshold is a 2D plane that cuts through the 3D graph. When the plane passes through a local maximum we have that connected components are formed, when it passes a saddle point components are joined to form holes, and when a local minimum is passed holes are filled. This reveals that filtration captures incidence of different types of critical points in the field (its topological features) and this information is summarized in the EC curve. This process is illustrated in Figure 2.5; here, we see that the EC curve contains a single minimum and a single maximum.

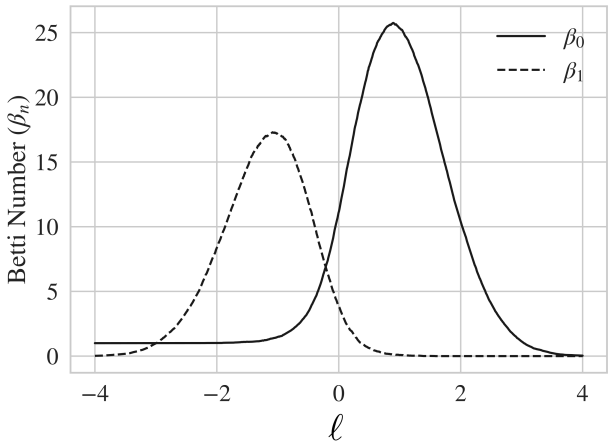


Figure 2.6: Connected components (β_0) and holes (β_1) that make up the EC curve for the 2D field in Figure 2.5. The structure of the EC curve is a direct reflection of the changing topology of the superlevel sets during the filtration. The local maxima of the field represent connected components (β_0) which are eventually joined via saddle points to form holes (β_1) that are filled in once local minima of the field are passed by the filtration.

The reason for this structure is revealed when we visualize the individual topological bases for our 2D field (see Figure 2.6). Here, we see two distinct areas of the filtration, the first is dominated by connected components (β_0) which causes the maximum and the second is dominated by holes (β_1) which causes the minimum. These topologically distinct components of the filtration represent the presence of local maxima (high β_0 values) and saddle points/local minima (high β_1 values) in the 2D field. Betti numbers reveal differences in the topology of a data object at varying thresholds and have been used in analysis that is complementary to the EC itself [104]. Specifically, analyzing individual Betti numbers can provide additional insight into the appearance/disappearance of specific topological features throughout the filtration process. This approach is at the core of the so-called persistence diagrams (which summarize the appearance/disappearance of topological features). Persistent diagrams, while more informative, are difficult to interpret and analyze primarily because the diagram is constructed from an unordered set of intervals which are not amenable to statistical computations (e.g. means and variances)

[134, 135]. For this reason, it is difficult to analyze persistence diagrams directly or to use them as a data preprocessing step for further analysis in statistical and machine learning methods without further transformation.

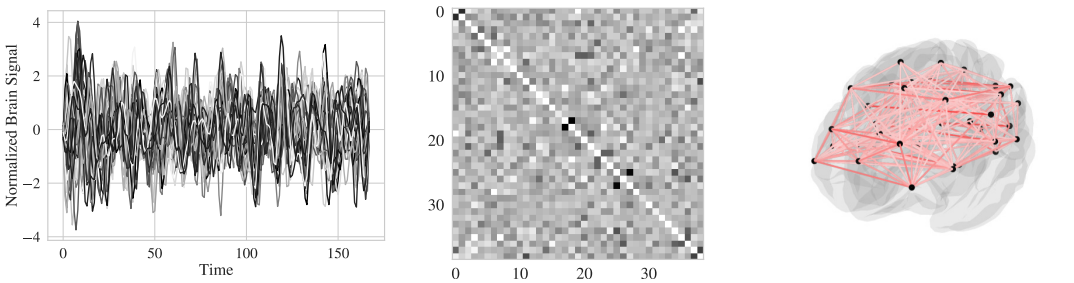
2.5 Case Studies

In this section, we illustrate how to use the EC to characterize diverse datasets arising in applications of interest to chemical and biological engineers . All scripts and data needed to reproduce the results can be found here https://github.com/zavalab/ML/tree/master/ECpaper.

2.5.1 *Brain and Process Monitoring*

One is often interested in characterizing the topology of graphs (such as those arising in brain networks or process networks) as a way to identify abnormal behavior. As an example, we might want to relate the topological structure of the functional connections of brains at different stages of development (adult vs. child) or to identify diseases. Here, we illustrate how to do this using a real dataset taken from the work of Richardson and co-workers available in OpenNeuro (ds000228) [136]. We then show how to use this same approach to identify faults in chemical processes.

In the brain dataset, adults and children watch a short film and the activity in different regions of the brain are measured with functional magnetic resonance imaging (fMRI). Here, the signal in each region $i = 1, ..., n$ is a univariate random variable Y_i and we denote the collection of signals as the multivariate random vector $Y = (Y_1, ..., Y_n)$. We denote the observation of the signals at time $t = 1, ..., m$ as $Y(t) \in \mathbb{R}^n$. This dataset is thus a multivariate time series; the series is used to construct a functional network, which is given by the sample inverse covariance matrix $\text{Cov}[Y]^{-1} \in \mathbb{R}^{n \times n}$ (also known as precision matrix). This procedure is summarized in Figure 2.7. The precision matrix is represented as an edge-weighted graph $G(V, E, w_E)$; here, vertices represent the different



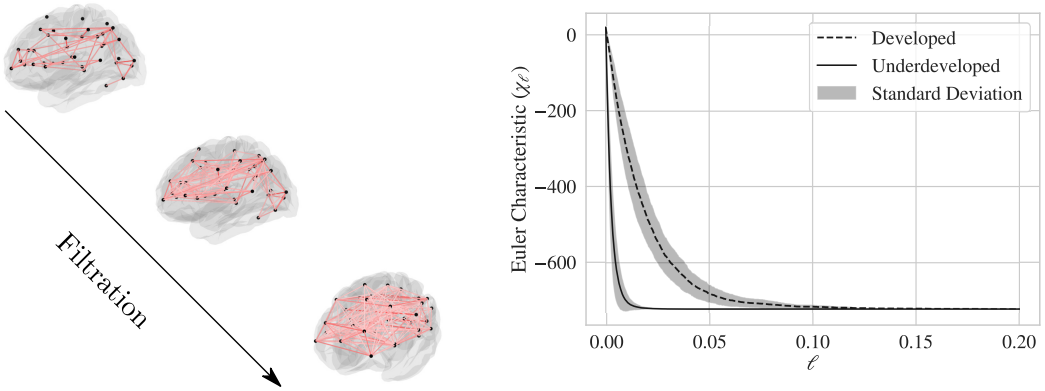
(a) Multivariate time series (b) Precision matrix (c) Functional network

Figure 2.7: (a) Brain signals measured during an fMRI study of a developed brain while watching a film. The signals measure brain activity in different regions of the brain. (b) The precision matrix constructed from the brain signals. (c) Functional network representation of the precision matrix. The width of the edges represent the strength of the partial correlation between the different regions of the brain.

regions of the brain and edge weights $w(e) \in [0,1]$ represent the absolute value of the partial correlation between different regions. From Figure 2.7, we can see that these functional networks have a complex structure, making them difficult to characterize.

We perform a filtration in order to characterize the precision matrix of different brains. This filtration gives an EC curve for each brain that is used to understand topological differences between developed and underdeveloped brains (see Figure 2.8). Here, we can see that there is a perceptible difference between the average EC curves for different brain types. This illustrates the effectiveness of the EC curve in identifying structural differences in complex functional networks [137, 138]. Intuitively, a developed brain has a more widespread correlation structure, and thus the EC curve decays more slowly.

Interestingly, the brain monitoring problem is analogous (from a mathematical viewpoint) to the problem of chemical process monitoring. This is because, fundamentally, any multivariate time series can be represented as a precision matrix. In brain monitoring methods, such as fMRI, the goal is to identify differences in brain activity that may be a result of genetic defects, disease, or different stimuli. In chemical process monitoring, we seek to identify the presence of faults, disturbances, or problems in equipment operation. Brain monitoring typically uses observations on electrical signals, blood flow, or oxygen levels; in a chemical processes we use observations on temperatures, pressures,



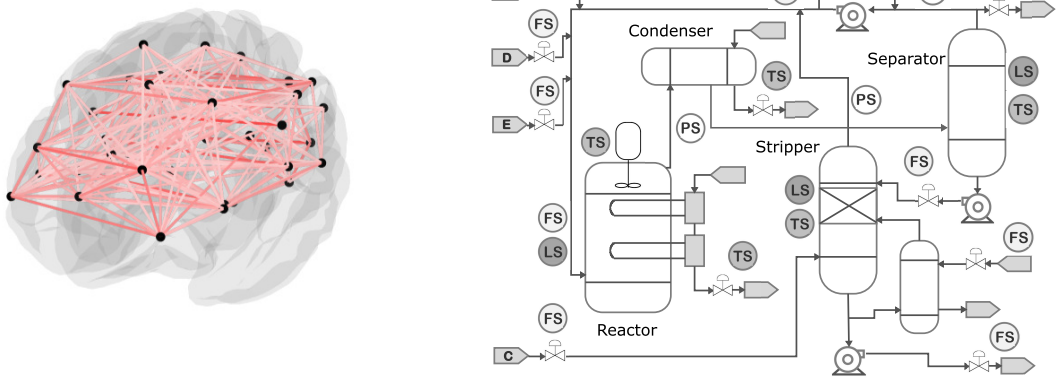
(a) Brain signal time series.

(b) EC curve from precision matrix filtration.

Figure 2.8: (a) A representation of the filtration process showing the addition of new edges as the level set of correlation is increased. The addition of the edges reduces the number of connected components (β_0) and increases the number of cycles (β_1). (b) Comparison of the average EC curves for the brain functional networks of individuals with developed and underdeveloped brains while watching a film. The difference between the two curves demonstrates that the EC, along with the associated filtration, can be an effective tool in differentiating complex networks .

and flows (see Figure 2.9). To highlight this analogy, we provide a fault detection case study for the Tennessee Eastman process dataset [4]. This dataset contains a simulation of a chemical process that is monitored using 52 variables (Figure 2.9). The goal is to use time series for such variables to identify faulty behavior. The variables monitored by the different sensors of a chemical process (in different locations) are Y_i, $i = 1, 2, ..., n$ and the observations are $Y(t) \in \mathbb{R}^n$ for $t = 1, ..., m$. As in the brain example, this multivariate time series is used to construct the precision matrix $\text{Cov}[Y]^{-1} \in \mathbb{R}^{n \times n}$. We again represent this matrix as an edge-weighted graph $G(V, E, w_E)$ and use filtration to determine its EC curve. We compute EC curves for precision matrices obtained from different multivariate time series (containing different types of faults).

The Tennessee Eastman dataset contains multiple simulations of the chemical process; here, some simulations contain faults and others do not. For each simulation, we construct an edge-weighted graph from the precision matrix and leverage the EC curve to identify whether or not the given process is experiencing a fault based upon the topo-

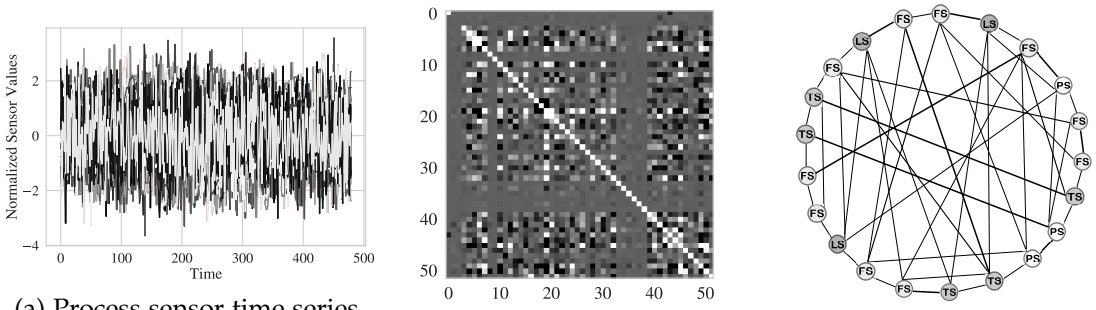


(a) Brain functional network filtration.

(b) Chemical process schematic.

Figure 2.9: (a) Representation of the brain and the different area signals obtained during fMRI. The edges represent interactions between the regions of the brain computed via the precision matrix. (b) A simplified representation of the Tennessee Eastman process. The process is monitored using temperature sensors (TS), pressure sensors (PS), flow sensors (FS) and level sensors (LS) distributed in different regions. While the context of the two systems are vastly different, the type of data produced (i.e. multivariate time series) are identical. This suggests that methods used in the analysis of brain functional networks for the detection of disease can be directly applied to chemical process systems to detect faults or process issues.

logical structure induced by the correlations between the process variables (Figure 2.10). Figure 2.11 shows examples of the precision matrices derived from process simulations that either contain or do not contain faults, demonstrating that the identification of faults from the precision matrix is not a trivial task.



(a) Process sensor time series. (b) Precision matrix. (c) Functional network.

Figure 2.10: (a) Process sensor measurements during the operation of the Tennesee Eastman process system simulation. The measurements represent the output of the various temperature, pressure, flow, and level indicators during the simulation. (b) The precision matrix constructed from the process sensor measurements in (a). This precision matrix is then used to construct an EC filtration. (c) Simplified graphical representation of the precision matrix derived from the process sensor measurements.

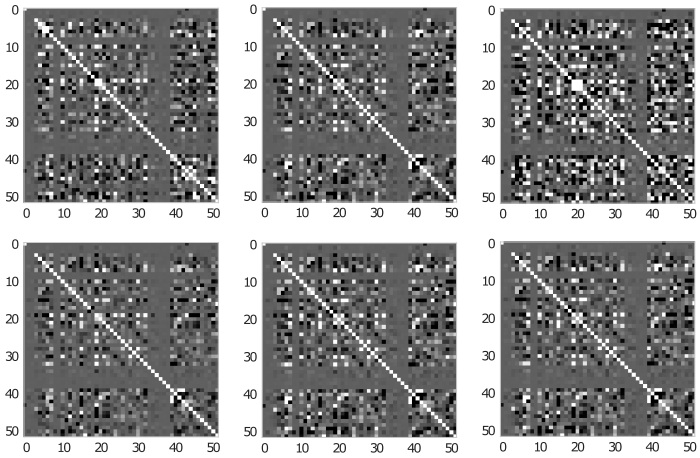
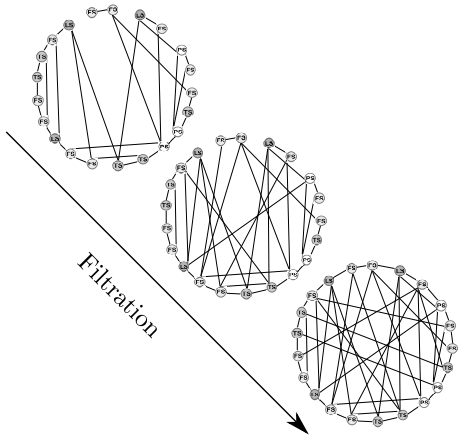


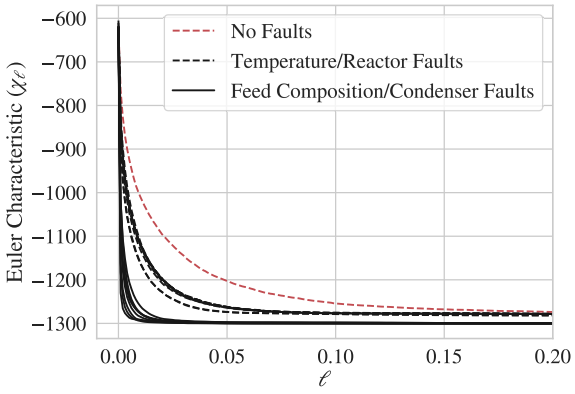
Figure 2.11: Visualization of multiple precision matrices derived from chemical process simulations with and without faults. Distinguishing these matrices is difficult, demonstrating the inherent complexity in identifying faults in a process with the precision matrix alone.

Figure 2.12 demonstrates that there is a quantifiable difference between the process

operating with no faults, and the process operating with faults. We also note that there is a separation into two groups within the faulty systems. The two groups of faults deal with with either feed temperature and reactor faults, or feed composition and condenser faults. The reason for this separation is of interest and will be explored in future work. This demonstrates that the EC can be used to detect faults based purely on the topological structure of the precision matrix. Note that this is a method that accounts for the space-time relationships of the entire process (all-at-once) and does not require statistical assumptions on the data (e.g., independence of observations). The simplicity of (2.3) also ensures that the EC can be rapidly calculated via the number of edges and vertices of a graph. The computation of the EC and the associated EC curve requires a simple thresholding operation (to obtain the number of nodes and edges in the filtered graph) and a few addition and subtraction operations (to sum the number of edges and nodes and compute the EC value) at each point in the filtration, because of this the method scales well with large networks as the required computations are simple and efficient.



(a) Chemical process network filtration.



(b) EC curves from a network filtration.

Figure 2.12: (a) Representation of the filtration process showing the addition of new edges as the level set threshold for correlation is increased. (b) Comparison of the EC curves for the Tennesee Eastman process under faults and no faults. There is also a notable separation of the faults into two groups, one which represents faults primarily associated with feed temperatures or reactor conditions, and the other associated with feed composition changes or condenser faults.

2.5.2 *Spatio-Temporal Data Analysis*

We now use the EC to characterize the spatio-temporal behavior of fields for a reaction diffusion system. Here, the fields are solutions of a PDE model with different diffusion ($D \in \mathbb{R}$) and reaction ($R \in \mathbb{R}$) coefficients. The coefficients can be manipulated to generate fields with different topological features, which capture different mechanistic behavior (e.g., reaction-limited or diffusion-limited). The model is described by the coupled PDEs:

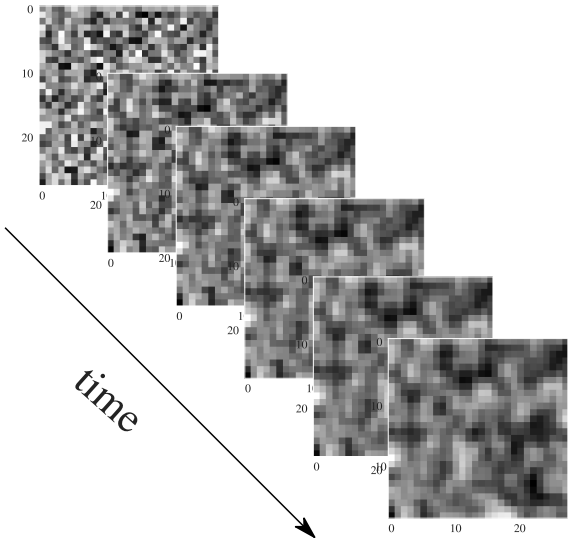
$$\frac{\partial u(x,t)}{\partial t} = D\left(\frac{\partial^2 u(x,t)}{\partial x_1^2} + \frac{\partial^2 u(x,t)}{\partial x_2^2}\right) + R(v(x,t) - u(x,t)) \tag{2.8a}$$

$$\frac{\partial v(x,t)}{\partial t} = D\left(\frac{\partial^2 v(x,t)}{\partial x_1^2} + \frac{\partial^2 v(x,t)}{\partial x_2^2}\right) + R(u(x,t) - v(x,t)) \tag{2.8b}$$

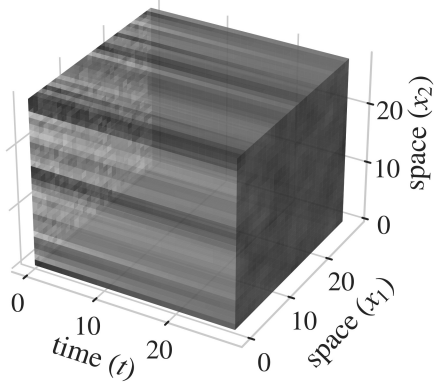
Here, $u(x,t) : \mathcal{D}_x \times \mathcal{D}_t \to \mathbb{R}$ and $v(x,t) : \mathcal{D}_x \times \mathcal{D}_t \to \mathbb{R}$ represent the concentrations for the reactants over space and time. The spatial domain is continuous and given by $\mathcal{D}_x := [0,n] \times [0,n] \subset \mathbb{R}^2$; the temporal domain is $\mathcal{D}_t := [0,T] \subset \mathbb{R}$. We thus have that $u(x,t)$ and $v(x,t)$ are 3D fields (embedded in a 4D manifold). Here, we hypothesize that the topological features of these fields are expected to change with the parameter pair (D,R) (the structure changes with governing mechanism).

An example field generated for a given parameter pair (D,R) is shown in Figure 2.13. Here, we focus on characterizing the topology of $u(x,t)$ for different values of the parameters. To do so, we generate 30 fields (obtained using different random initial conditions) for each of the following combinations: $(D = 3, R = 0.8)$, $(D = 3, R = 0.4)$, and $(D = 6, R = 0.8)$. The goal in the analysis of this dataset is to cluster the realizations into groups that reflect the parameters of the models (e.g., to detect changes in the underlying mechanism). For each simulation, we represent $u(x,t)$ as a spatio-temporal field (Figure 2.13). We construct superlevel sets for each spatio-temporal field and record the EC of the resulting superlevel sets. This process is similar to the examples shown for the 1D and 2D cases (Figures 2.4 and 2.14); however, the filtration performed

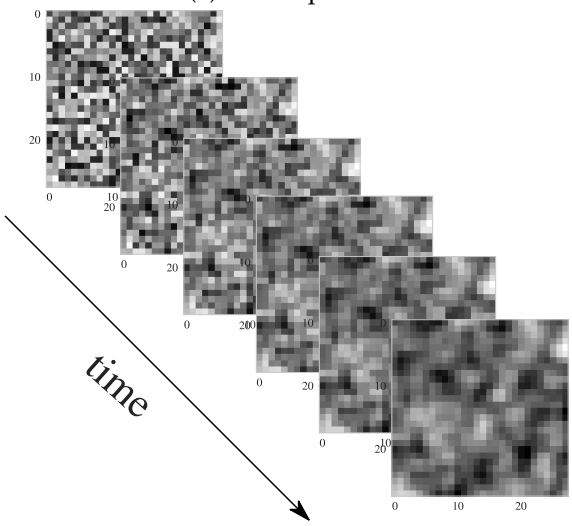
here can no longer be visualized as a simple plane that slices the field. This highlights the versatility of using the EC to characterize datasets over high dimensions.



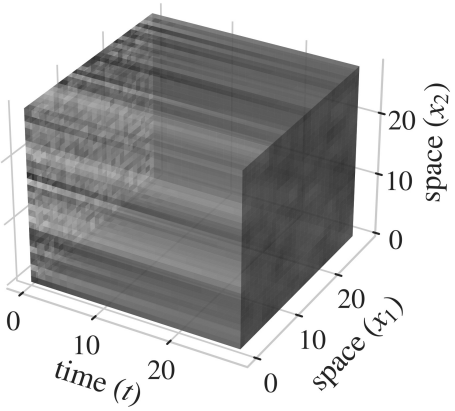
(a) 2D snapshots



(b) 3D space-time field



(c) 2D snapshots

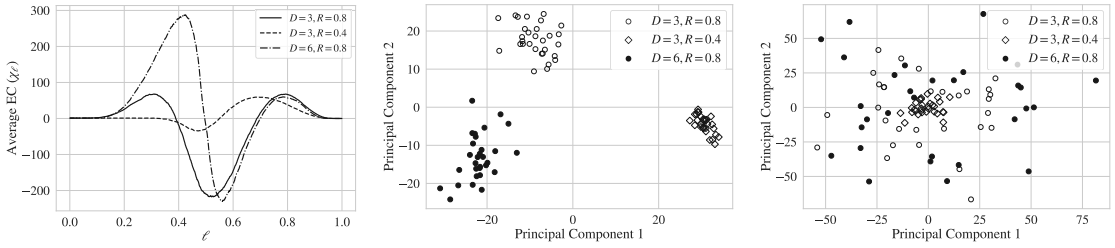


(d) 3D space-time field

Figure 2.13: (a,c) Snapshots obtained during evolution of reaction-diffusion system(2.8a)-(2.8b) with different parameter sets (D, R). (b,c) Collection of snapshots in 3D space-time field. It is difficult to distinguish the realizations of the models with different parameter values, regardless of whether they are viewed via individual snapshots or as 3D space-time fields.

The average EC curves for the three different parameter settings are presented in Figure 2.14. Here, it is clear that the EC reveals a change in the topological structure.

To perform clustering, we represent the EC curve for sample $j = 1, ..., n$ as a vector $\chi_j \in \mathbb{R}^m$; the entries of this vector are the EC value of the level set. Each EC vector can be stacked into a matrix $[\chi_1\ \chi_2\ \chi_3\ ...\ \chi_n]^T \in \mathbb{R}^{n \times m}$. We obtain a matrix for each of the three parameter settings. We apply a singular value decomposition to these matrices and visualize the data projected onto the two leading principal components. Figure 2.14 shows the results; note that there is a distinct clustering of the data into three separate groups. This confirms that the EC curve captures the topological differences of the fields obtained under different parameter settings. For comparison, we compute the Fourier transform of the data to obtain the frequency spectrum and project the spectrum to two dimensions using a singular value decomposition. The results are shown in Figure 2.14; here, we can see that the frequency spectrum of the data does not contain enough information to separate the different types of fields. This indicates that the EC contains information that cannot be captured by the frequency spectrum.



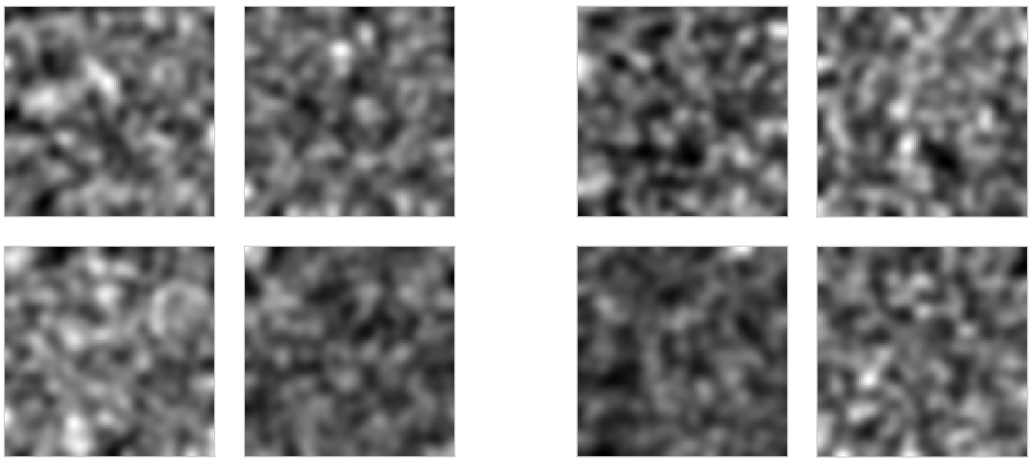
(a) Average ECs (b) SVD of ECs (c) SVD of Fourier spectrum

Figure 2.14: Demonstration of the EC's ability to capture topological differences induced by different reaction-diffusion parameters. (a) Average EC curves for the three different parameter settings. The EC is able to separate the realizations of the reaction-diffusion system. (b) SVD projection of the EC curves onto their two leading principal components, revealing a distinct clustering of the different types of space-time fields. (c) SVD projection of Fourier spectrum of the space-time fields; here, there is no distinct separation between the different fields.

2.5.3 *Image Analysis*

We explore the topological characterization of simulated micrographs for liquid crystal (LC) systems. These micrographs capture their responses to different reactive gaseous

environments [54, 139]. These LC systems start with homeotropic alignment of a thin film on a functionalized surface. When the LC system is exposed to an analyte, the analyte diffuses through the LC film and disrupts the binding between the LC and the surface. This disruption triggers a reorientation of the LC film and forms complex optical patterns and textures simulated via random fields. Figure 2.15 provides images (micrographs) that capture the response of an LC system to a couple of different environments. For this dataset, we want to characterize the topological differences between the textures of the LC systems when exposed to the different envrionments and use this information to classify the datasets. This provides a mechanism to design gas sensors. Such classification tasks have been recently performed succesfully using convolutional neural networks [54]; these machine learning models, however, contain an extremely large number of parameters and are difficult to train.



(a) LC response to environment 1 (b) LC response to environment 2

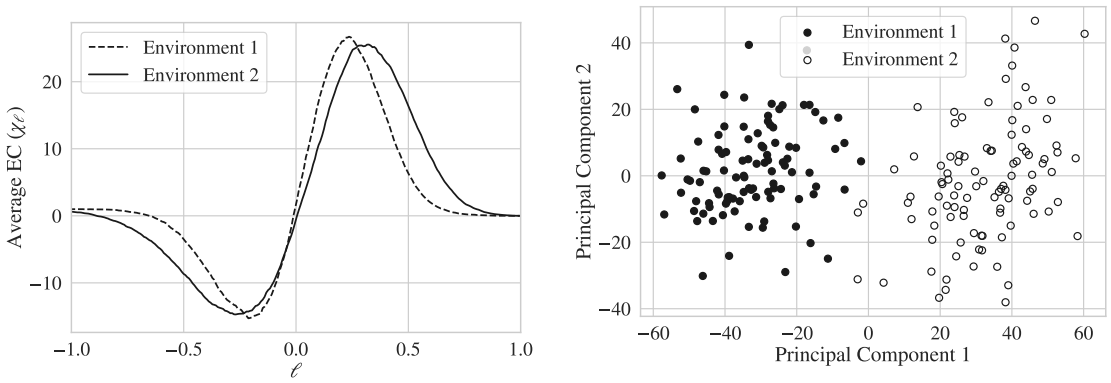
Figure 2.15: Comparison of the visual response of an LC system to two different gaseous environments. We can see that there are perceptible differences between the responses, but these differences are difficult to quantify due to the heterogeneity present. The EC is able to effectively summarize the topological differences and similarities in the images, allowing for an accurate separation of the responses.

The micrographs of the LC systems at their endpoint are discrete fields (matrices), which we represent as node-weighted graphs. The EC curves for such graphs are obtained via filtration. The filtration process seeks to characterize regions of high and low

intensity in the A* channel. We highlight that one can also think of an image as a discrete approximation of a continuous 2D field; this emphasizes that the filtration process for a discrete field is analogous to the filtration process of a node-weighted graph (we perform filtration over weight values instead of over field values).

The average EC curves for the LC systems exposed to the two gaseous environments are presented in Figure 2.16. Here, we also show the projection of the EC curves onto the first two principal components using SVD. We can see that there is a strong separation between the two datasets. As a comparison, we cluster the micrographs by applying SVD directly to the images (as opposed to the EC curves) and by computing the Fourier transform of the images, which decomposes an image into a weighted set of spatial frequency functions. We chose these methods for comparison because they are commonly used in characterizing image textures [141, 142]. We can see these traditional techniques do not provide a clear separation (Figure 2.17). Figure 2.18 shows the distribution of the first principal component of the EC values and compares this with the distribuiton of Moran's I values which capture average spatial autocorrelation in the images by computing spatial autocorrelation in neighborhoods around each pixel in the image and averaging results over the entire image [143]. It is clear that the ECs provide a sharper separation than Moran's I; we conclude that EC is a more effective descriptor. We attribute the limitations encountered with traditional tools (such as SVD, Fourier, and Moran's I) to the spatial heterogeneity of the images (see Figure 2.15) [70].

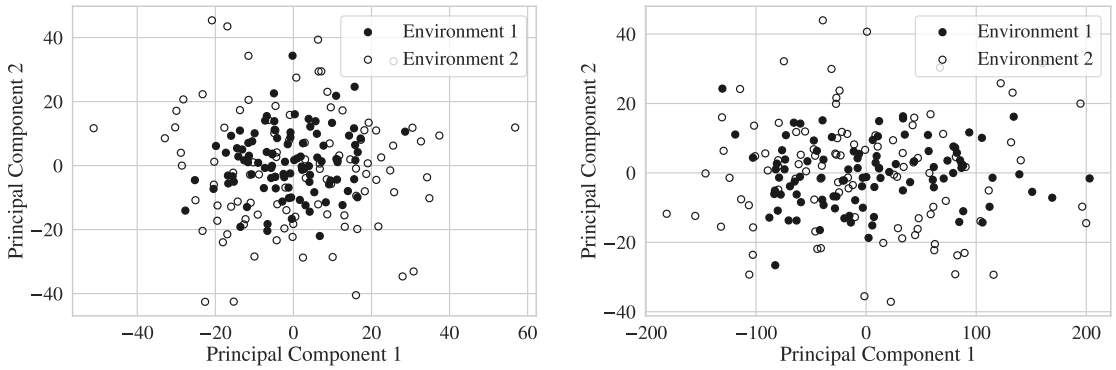
To further demonstrate the usefulness of the EC; we used the EC curve (a vector) as an input to a support vector machine (SVM) for classification of the two datasets. We compare this classification approach against approaches that use SVM with: (i) raw image as input and (ii) Fourier spectrum as input. Using the EC vector as an input, we were able to classify the two datasets with $95 \pm 6\%$ accuracy, compared to $66 \pm 7\%$ and $68 \pm 7\%$ accuracy obtained with raw images and Fourier spectrum. It is particularly remarkable that, after reducing the images to an EC curve, it is possible to separate the datasets using a simple (linear) SVM classifier. This highlights how the EC can be used to pre-process



(a) Average EC for LC system response.

(b) PCA of EC curves from LC system responses.

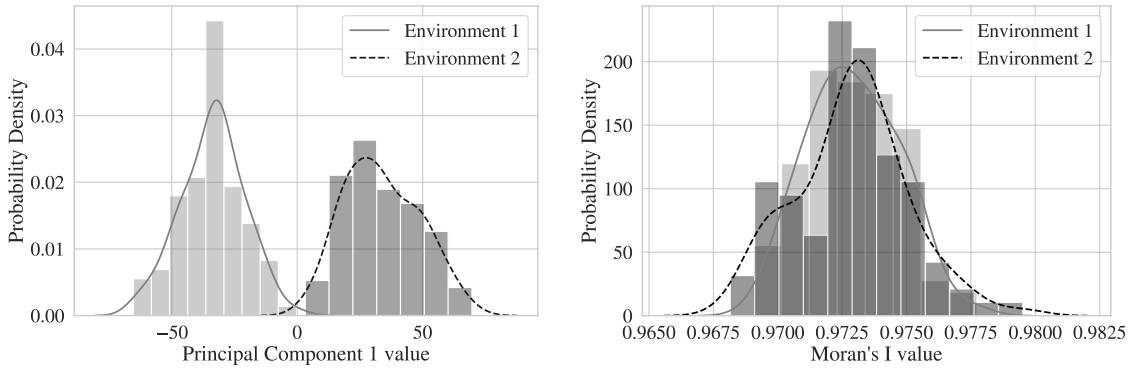
Figure 2.16: (a) The average EC curve of the LC system responses to the two different gaseous environments. (b) SVD performed on the EC curves. This highlights that the EC is able to produce a strong, linear separation of the LC responses.



(a) SVD on images

(b) SVD on Fourier spectrum of images

Figure 2.17: SVD of the LC systems responses using (a) the raw image data and (b) the Fourier spectrum of the images. Under these approaches, there is no obvious separation of the data.



(a) Distribution of 1st PC of EC curves

(b) Distribution of Moran's I values

Figure 2.18: (a) Distribution of the 1st principal components for the EC curves. (b) Distribution of the Moran's I values. We can see that the ECs provide a sharper separation than Moran's I values; as such, ECs are a more informative descriptor of the data.

data and how this can facilitate machine learning tasks. For instance, in this case, it is not necessary to use a more sophisticated machine learning model (e.g., a convolutional neural net) to perform image classification.

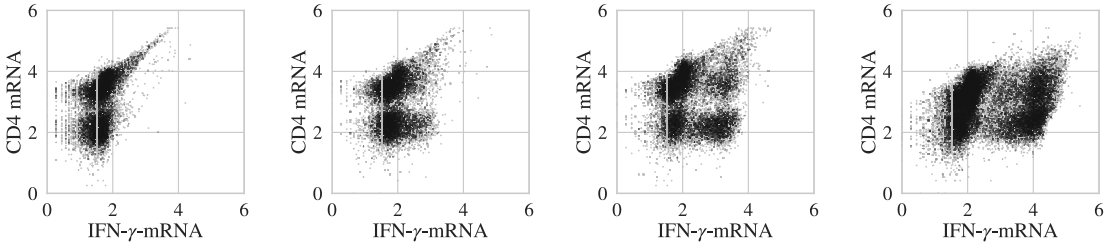
2.5.4 *Point Cloud Analysis*

We now shift our focus to the use of the EC to analyze the structure of point clouds (also known as scatter fields). Here, we consider 2D point clouds that are realizations of a bivariate random variable. As such, the point clouds emanate from a 2D joint probability density function .eps). The analysis of the shape of univariate.epss is typically performed by using summarizing statistics (e.g., moments); analysis in higher dimensions is quite complicated (no good descriptors exist for multi-dimensional joint.epss) [144]. This limitation is particularly relevant when the.eps has a complex shape (e.g., it is non-Gaussian). Here, we will see that one can characterize the complex shapes of a multi-variate.eps by using an EC curve. Moreover, the results that we present highlight that the EC can be used to characterize.epss in higher dimensions (even if it is not possible to visualize them).

We use an experimental flow cytometry dataset to illustrate how this can be done.

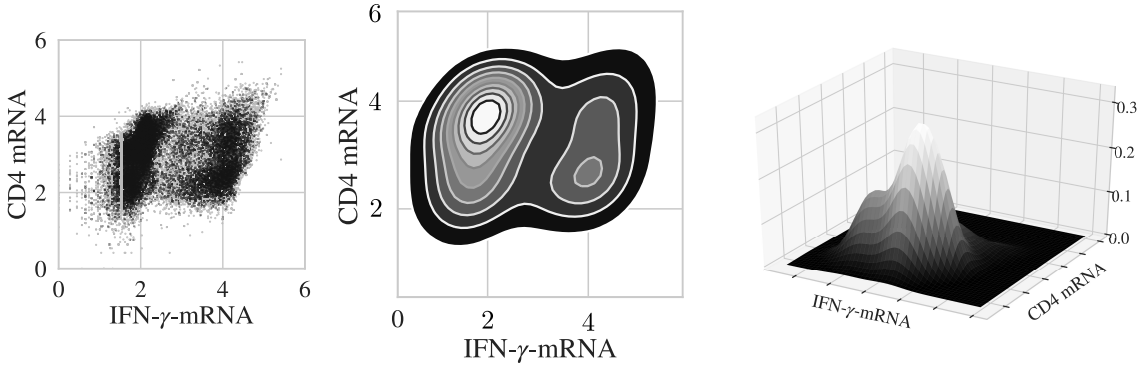
The dataset was obtained through the FlowRepository (Repository ID: FR-FCM-ZZC9) [145]. This dataset is obtained in a study of the kinetics of gene transcription and protein translation within stimulated human blood mononuclear cells through the quantification of proteins (CD4 and IFN-γ) and mRNA (CD4 and IFN-γ) [146]. In our study, we focus on the evolution of the concentration of CD4 mRNA and IFN-γ mRNA in a given cell which is measured via a flow cytometer. At each time point, a number of cells ($\sim 15,000$) are passed through the flow cytometer; each of these cells provides an observation vector $y \in \mathbb{R}^2$ (corresponding to CD4 mRNA and IFN-γ mRNA). These observations are typically visualized as point clouds in a 2D scatter plot. The evolution of the point clouds over time is shown in Figure 4.39. The goal is to characterize how the shape of the point clouds evolves over time; for instance, it is clear that the point cloud progressively separates into two distinct domains.

To characterize the shape of the point clouds, we convert them into a continuous 2D field by applying smoothing. It is important to note that the point clouds are realizations of a bivariate random variable (CD4 mRNA and IFN-γ mRNA); as such, one can obtain a 2D histogram for them (by counting the number of points in a bin). The histogram is an empirical approximation of the joint.eps of CD4 mRNA and IFN-γ mRNA. The continuous 2D field obtained via smoothing is a smooth representation of the 2D histogram. Our approach provides an alternative to traditional heuristic methods such as gating, which are difficult to tune as they are highly sensitive to potential noise and outliers in the data [147].



(a) Time = 0 Minutes. (b) Time = 30 Minutes. (c) Time = 60 Minutes. (d) Time = 90 Minutes.

Figure 2.19: Deformation of a 2D scatter field over time.



(a) Sample distribution. (b) Density estimate. (c) Density surface.

Figure 2.20: Transforming the point cloud to a field. The raw data (a) is smoothed via a Gaussian kernel and then the smoothed 2D field (b) is represented in 3D and processed via a filtration (c).

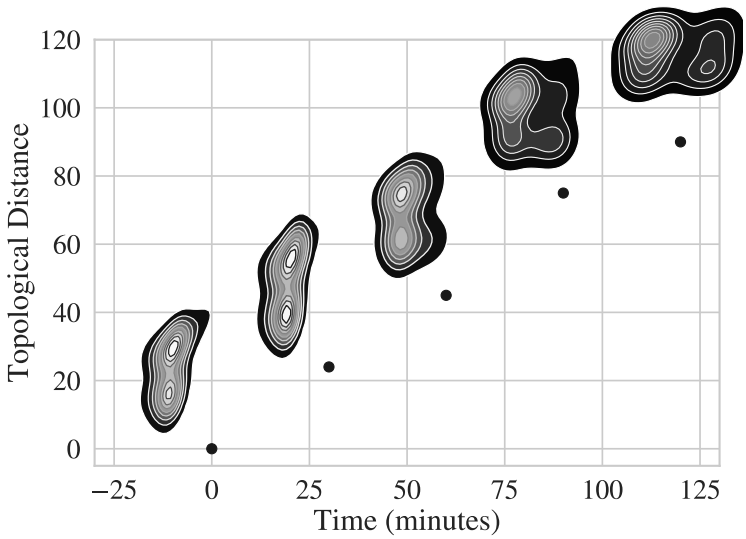


Figure 2.21: Euclidean distance between the EC curves as a function of time. There is a continuous evolution of the distance that characterizes the change in topology.

In order to create the 2D fields, we utilize a Gaussian kernel smoothing [148]. Figure 4.40 shows the smoothing of a scatter plot. The 2D field is a projection of a 3D function (it is embedded in a 3D manifold). This is illustrated in Figure 4.40; as such, we can characterize the field by using a filtration on the values of the.eps. Our goal is thus to compute an EC curve for the.eps at different points in time (to analyze how its shape evolves). To visualize this deformation, we compute the Euclidean norm of the difference

between EC curve at a time point to that at the initial time. From Figure 4.41, we can see that the distance exhibits a strong and continuous dependence with time (indicating that there is a strong change in the shape of the density function). Importantly, this also suggests that there exists a continuous mapping between the EC and time (the topological deformation is continuous with respect to time). For instance, one could construct a dynamical model that predicts the evolution of the shape with time. This indicates that the EC provides a useful descriptor to monitor the evolution of complex shapes. This could help, for instance, to detect times at which the change in shape is fastest/slowest.

Chapter 3

TOPOLOGY & MOLECULAR SIMULATION

The contents of this chapter are published in [29]

3.1 Introduction

This chapter builds upon the work presented in chapter 2 and explores the application of topology in the analysis of molecular simulations. The development of advanced molecular dynamics (MD) simulation methods has provided researchers the ability to rapidly screen for new chemistry, biological interactions, and materials [149, 150, 151]. For example, large-scale molecular simulations have been used in the screening of molecular organic frameworks (MOFs) for hydrogen storage [152]. These techniques are also employed in the study of soft materials such as proteins and polymers [153, 154, 155], and in the design of self-assembled colloidal systems [156]. However, the analysis of MD datasets is challenging due to both their size and complexity; specifically, MD simulations can produce terabytes of data that require computationally efficient and scalable analysis methods, while the complexity of the data requires methods that are generalizable to a broad range of systems and that are robust to data heterogeneity and noise [157].

Quantification and reduction of molecular simulation data has been traditionally con-

ducted via order parameters and summarizing statistics such as radial distribution functions and correlation fields, particularly for condensed-phase systems [158]. These descriptors are usually computationally efficient, physically interpretable, and are derived from principles of physics and statistical mechanics. Moreover, such descriptors usually correlate to emergent properties of interest and thus can be used to construct predictive models. However, order parameters are typically designed for particular applications that meet specific assumptions (e.g., spatial isotropy or crystallinity) and are thus limited in scope [159, 160, 161, 162].

Another approach for quantifying molecular simulation data consists on using machine learning (ML) tools such as convolutional neural networks (CNNs) and autoencoders to extract informative descriptors from data [163, 164, 165, 166, 167]. ML tools are versatile in that they require few assumptions on the application and can be used for processing diverse data formats (e.g., images, tensors, graphs). However, descriptors extracted using ML tools can be difficult to interpret and training predictive models based on such descriptors may require large numbers of parameters and large amounts of labeled MD data, which can be computationally costly to produce.

In this chapter, we investigate the application of tools from topology for the analysis of MD simulation data. Topology focuses on characterizing the global structure (e.g., connectivity, continuity) of shapes and objects and has been gaining attention in diverse scientific and engineering fields [168, 31]. In the context of molecular simulations, connectivity and continuity of molecules such as polymers, proteins, and other molecules have been studied via topology. These topological methods have mostly relied on the use of applied knot theory, which can quantify the entangled structures of molecules and use this information to predict thermodynamic bulk properties. These techniques have also been used for connecting the knotting of molecules with their reactivity and function (e.g., DNA recombinase enzymes) [169, 170, 171]. In these applications, however,

the connection between the molecular data object and its topological representation (i.e., knots) is limited to specific settings. Recent work has also focused on the application of persistence homology to molecular simulation data [172, 168]. Persistent homology has been shown to provide powerful characterizations of the topology and geometry of data [173, 174, 175, 176, 172, 177]. However, the outputs of these methods (e.g., persistence diagrams) can be difficult to directly integrate into data analysis and ML tasks without further transformation (e.g., vectorization and smoothing) and hyperparameter optimization [31].

The main goal of this chapter is to demonstrate that topology can be applied to a broad range of molecular simulation settings. At the core of this approach is the observation that one can represent data as graphs and manifolds, which are versatile topological objects that can be efficiently quantified using a topological descriptor known as the Euler Characteristic (EC) [31, 100]. Graph representations for molecular simulation data have been widely applied and are easy to justify from a physical standpoint [178, 179]. An example arises in the analysis of non-covalent bonding networks (e.g., hydrogen bonding networks in water). Here, the individual water molecules can be considered vertices of a graph with non-covalent bonds (e.g., hydrogen bonds) representing edges between the vertices. Manifold representations for molecular simulations arise when there is a continuous function describing behavior over a space (or surface) of a simulation. An example of a manifold representation arises in the analysis of time-averaged spatial density, where density is computed at each spatial location resulting in a continuous function over the entire simulation space [31, 168]. Manifolds can be extended to more complex spaces, such as the surface of a molecule, nanoparticle, polymer, or protein [180]. In these approaches, the surface is treated as a manifold and physical characteristics (e.g., hydrophobicity, charge, forces, curvature) represent functions on the manifold. However, graph and manifold representations of molecular simulation data can be high-dimensional and not directly amenable to common analysis tasks (e.g. classification, regression, visualization). Thus,

there is need for simple and computationally efficient methods for reducing and quantifying these topological data representations. This characterization can be accomplished by performing a decomposition of the space into a set of independent *topological bases* that capture basic topological features such as holes, connected components, and voids. The EC is a scalar integer quantity that is defined as the alternating sum of the rank of these topological bases of an object. The EC is often combined with a data processing technique known as *filtration*, which enables the characterization of more complex topological objects such as matrices, images, fields/functions, and weighted graphs [31, 138]. The filtration process gives rise to the so-called *EC curve*, which is a function that summarizes how topological features emerge and disappear through the filtration process. Compared to descriptors obtained from persistent homology (e.g., persistence diagrams), the EC curve provides a quantifiable and easy-to-interpret descriptor of complex data objects.

To illustrate the efficiency and effectiveness of the EC, we provide studies arising in a couple of complex molecular simulation systems. The first simulation system is the measurement of hydrophobicity on the surface of 2D self-assembled monolayers (SAMs), and the second system is the analysis of 3D solvation effects on acid-catalyzed reactions systems for biomass processing. Previous work suggests that the topology and geometry of water plays a critical role in understanding and predicting emergent physical and chemical properties for both of these systems [181, 182]. Thus, in both examples, we are focused on quantifying the topological structures and patterns emerging from solvent behavior at the surface of a SAM and around biomass-relevant reactants to quantify hydrophobicity and reactivity, respectively. We also show that the EC can be used to quantify different forms of data representations typically used in these types of MD simulations (hydrogen bond networks and density fields). Specifically, we show that the EC is a descriptor that correlates strongly with emergent properties and this enables the construction of low-dimensional and effective predictive models that are significantly more computationally tractable than recently-developed ML models such as CNNs. We show

that simple regression models that take the EC as input, are able to accurately predict the hydration free energy of simulated 2D SAMs and the change in reactivity due to solvation effects in acid-catalyzed reaction systems. These studies also illustrate the stability of the EC in quantifying noisy MD data and the physical intuition that can be gained through topological analysis. Moreover, we show that the EC can be used to monitor the dynamic evolution of topology in these MD systems which can be used, for instance, to determine when a system has achieved a topological steady-state. We also note that solvent-rich processes (like those studied here) are common in a diverse range systems analyzed through MD, further supporting the general relevance of these methods. All code and data needed for reproducing our results are provided.

3.2 Topology of Graphs and Manifolds

To analyze MD simulation data with the EC, we begin by defining graphs and manifolds in the context of molecular simulation.

A graph is a 2D topological object that consists of an ordered pair $G(V, E)$, where V represents a set of *vertices* and E represents a set of paired vertices known as *edges*. Edges represent relationships (connectivity) between vertices. A graph representation for an atomistic simulation of water is shown in Figure 3.1. Here, the water molecules are represented as vertices and hydrogen bonds between molecules are represented as edges. This graph (network) representation can be used to understand how water is interacting both locally and globally, by quantifying specific features, such as the number of *cycles* and *connected components* of the graph. A cycle represents a path that traverses edges on a graph starting at a particular vertex v_j and ending at that same vertex v_j. Physical examples of graph cycles are found in tetramer, pentamer, and hexamer water structures [183, 184]. A connected component is a subset of a graph $C(V_C, E_C) \subseteq G(V, E)$ in which any vertex $v_i \in V_C$ of the subgraph can reach any other vertex $v_j \in V_C$ by traversing

edges of the subgraph $\{v_i, v_j\} \in E_C$, and is disconnected from all other subsets of the graph. In other words, the number of connected components is the number of connected partitions of a graph. In a hydrogen bonding network for water, connected components help us understand the physical state of the system; for example, in a condensed (or crystalline) state, there will be many hydrogen bonds present, reducing the total number of connected components but increasing the total number of molecules in each connected component. The opposite would hold true for a system acting as an ideal gas; here, no bonds are formed and thus each molecule exists as its own connected component. Data can also be encoded in a graph object (in nodes and edges) using functions $f : V \to \mathbb{R}$ and $f : E \to \mathbb{R}$. Values attached to nodes or edges are typically called weights or features); as such, graphs that encode data are also known as weighted graphs.

Manifolds are also versatile topological data representations that can capture continuous forms of information (e.g., 3D density fields) in high-dimensional spaces. This contrasts with graph representations, which capture discrete characteristics of a 2D data object (e.g., number of bonds, molecules, clusters). A manifold $\mathcal{M}$ is a topological space that *locally* resembles a Euclidean space; this means that the neighborhood of a point $x \in \mathcal{U}$ in an n-dimensional manifold (with $\mathcal{U} \subseteq \mathcal{M}$) can be mapped to $n-$dimensional Euclidean space through a continuous, bijective function. These neighborhoods and associated mappings are also known as *charts*. For example, the surface of the Earth is a 2D manifold and we can map the curved surface of the Earth to a flat Euclidean plane (i.e., a 2D Euclidean space) using a chart in order to measure properties such as distances or areas. The general nature of manifolds allows them to represent a broad range of structures, shapes, and complex geometric objects in molecular simulations (e.g., surface of a protein or a nanoparticle). Manifolds can also have encoded data on them (e.g., Earth surface temperature), which is captured using a continuous function $f : \mathcal{M} \to \mathbb{R}$. In Figure 3.1, we present a manifold representation for a 2D simulation of water. Here, the simulation domain (e.g., a 2D plane) is a 2D manifold and we define a continuous function that captures the time-averaged density of water at each location in the domain.

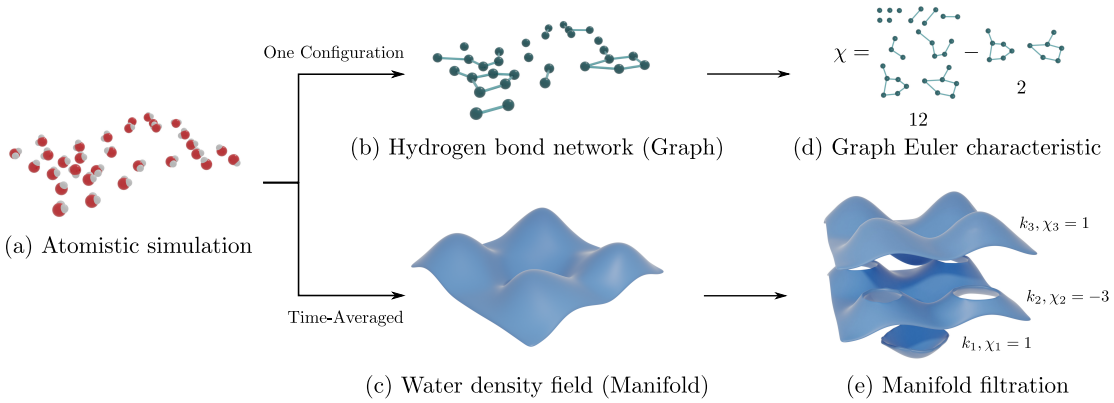


Figure 3.1: Graph and manifold representations of a molecular simulation of water. (a) Snapshot of an atomistic simulation of water (only some molecules are shown). (b) A graphical representation of the hydrogen bonding network formed between water molecules within the simulation, and (c) a density field derived from time-averaging water molecule positions during the simulation. The density field is represented as manifold $\mathcal{M}$, with a continuous function $f : \mathcal{M} \to \mathbb{R}$ that maps each point of the manifold to a corresponding water density value; visualized here by changes in the height of the surface. (d,e) Represents the EC χ quantification of the graph and manifold data representations. (d) The graph is quantified by subtracting the total number of cycles from the total number of connected components in the graph ($\chi = 12 - 2 = 10$). (e) The manifold is quantified through a filtration. At multiple increasing density thresholds $k_i \in \mathbb{R}$, the EC χ_i is computed by subtracting the total number of holes from the total number of connected components in the filtered manifold $x \in \mathcal{M} : f(x) \leq k_i$. We note that filtered manifolds all originate from the same data object and that the vertical layout is meant to illustrate the topological changes as the filtration is performed. The paired values $\{k_i, \chi_i\}$ are used to construct an Euler characteristic curve.

3.2.1 *The Euler Characteristic*

We present a brief introduction to the Euler characteristic (EC) in the context of molecular simulation. A detailed derivation of the EC can be found in Chapter 2. Graph and manifold representations are able to capture both discrete and continuous information within a given simulation and their topology can be directly quantified/summarized using a descriptor known as the Euler characteristic [31]. The EC is denoted as $\chi \in \mathbb{Z}$ and is mathematically defined as the alternating sum of the rank of topological bases for a given space known as Betti numbers $\beta_i \in \mathbb{Z}_+$, where $i \in \mathbb{Z}_+$ represents the dimensionality of the topological basis:

$$\chi := \sum_{i=0}^{n} (-1)^i \beta_i \tag{3.1}$$

Importantly, the topological bases of a space (e.g., connected components, holes, voids) are preserved under deformations such as stretching, twisting, and bending (are topological invariants). For any topological space of n-dimensions, there can only exist topological bases up to that given dimension. For example, a 3D space can only contain β_0 (representing connected components) β_1 (representing holes and cycles), and β_2 (representing voids and cavities) in the space. For example, in Figure 3.1 we represent a hydrogen bonding network for an atomistic MD simulation of water as a graph. A graph is a 2D topological space, and thus has a couple of Betti numbers β_0 and β_1. Figure 3.1 illustrates that, for a simulation snapshot, the number of connected components is 12, the number of cycles is 2, and thus the EC is $\chi = 12 - 2 = 10$.

3.2.2 *Manifold Filtrations*

Analysis of data represented as manifolds (or weighted graphs) requires an added processing step known as a *filtration*. A filtration quantifies the topology of *sublevel sets* of the

manifold. Given an n-dimensional manifold $\mathcal{M}$ and a continuous function $f : \mathcal{M} \to \mathbb{R}$, a *sublevel set* of the manifold is defined as $\mathcal{M}_{k_i}$ that contains points $\{x \in \mathcal{M} : f(x) \leq k_i\}$, where $k_i \in \mathbb{R}$ represents our *filtration threshold*. Hence, we can construct nested sublevel sets at increasing filtration thresholds for the manifold:

$$\mathcal{M}_{k_1} \subseteq \mathcal{M}_{k_2} \subseteq ... \subseteq \mathcal{M}_{k_n} \subseteq \mathcal{M} \tag{3.2}$$

where $k_1 < k_2 < ... < k_n$ represent our filtration thresholds, and $\mathcal{M}$ represents the original manifold. We can measure/quantify the topology of these nested sublevel sets with the EC at each filtration threshold $\{\chi_1, \chi_2, ..., \chi_n\}$. We ultimately obtain an ordered pair of values $\{k_i, \chi_i\}$, which characterize the topology of the manifold and its associated function. An illustration of the filtration process is found in Figure 3.1 for a 2D atomistic simulation, where time-averaged water density is analyzed over the space of the simulation. We have selected three different filtration values $k_1 \leq k_2 \leq k_3$ corresponding to the three sublevel sets. Similar to the graph example, we compute the EC by counting the total number of n-dimensional topological bases (β_0, β_1 for a 2D manifold). The bottom most sublevel set at filtration value k_1 represents a single connected component, capturing a local minima in the function f, and resulting in $\chi_1 = 1 - 0 = 1$. As the filtration threshold increases to k_2, a single connected component remains but four holes (i.e., cycles) are formed indicating the presence of local maxima of the function f, which results in an EC value of $\chi_2 = 1 - 4 = -3$. The final filtration threshold k_3 returns the original manifold, which is a single connected component: $\chi_3 = 1 - 0 = 1$ (further filtration of the space will not change the manifold topology). The filtration of a weighted graph is conducted in an analogous manner (by eliminating nodes or edges in which the data is below a certain threshold value). We note that filtration operations are easy to conduct and are thus scalable.

3.3 Applications of Topology in MD Simulations

The EC of graphs and manifolds provides a topological descriptor that quantifies complex structures and patterns that arise in MD simulations. The EC can be used to conduct a wide variety of ML and data analysis tasks such as visualization, clustering, regression, and classification. Here, we demonstrate that the EC of a molecular system correlate strongly to emergent physical and chemical characteristics. As such, we show that the EC can be used as an informative descriptor to predict emergent behavior. Moreover, we show that such predictions can be conducted using simple linear regression models, which contrasts with existing approaches based on CNNs.

The first set of simulations studied involve self-assembled monolayers (SAMs); here, we use the EC to predict the hydration free energy of the 2D SAM surface. The second set of simulations aims to predict the reactivity of a molecule based on the topology of a solvent environment composed of water and a cosolvent. These examples were specifically chosen because they were previously studied using advanced CNNs and thus have a frame of reference [181, 63]. We also highlight that these molecular systems are solvent-dominated; as such, their emergent properties are known to be influenced by the spatial structure and correlations of the solvent environment [185, 186]. This information will be quantified directly using the EC of graph and manifold representations of such environments.

Implementation details for both case studies can be found in the Supplementary Information. All code and data needed to reproduce the results can be found in https://github.com/zavalab/ML/tree/master/MD_Euler.

3.3.1 *Hydrophobicity on the Surface of Self-Assembled Monolayers*

We study the surfaces of SAMs using an MD simulation dataset obtained from recent work of Kelkar and co-workers [181]. The SAM structures are built from a planar array of alkanethiol ligands with hydroxyl, amine, or amide end groups. Each simulation consists of a single SAM solvated by bulk water. A simulation snapshot can be found in Figure 3.2a. A total of 50 different SAMs were created (22 having hydroxyl groups, 14 with amine end groups, and 14 with amide end groups). The partial charges of the end groups are modulated using a scaling factor that simulates changes in the polarity of the SAM surface. Additional details on the MD simulation methodology and parameters are available in [181].

Our goal is to study the topology of water in a thin interfacial layer located at the SAM surface. To do so, we leverage the topological structures formed by water at the SAM-water interface to directly predict the hydration free energy (HFE) of the SAM through linear regression. The HFE is a property that captures surface hydrophobicity behavior and is key in understanding protein adsorption [187, 188]. A common method for computing the HFE in molecular simulations is indirect umbrella sampling (INDUS). This method is highly accurate but it is computationally expensive, as it requires sampling of a low-probability event [189]. The dataset developed in previous work leveraged INDUS to create a set of SAM simulations with computed HFE values. Such simulations were used to train and test a CNN that directly predicts HFE from the SAM structure (represented as a 2D water density field). Here, we instead develop a linear regression model using the EC of the SAM structure.

To quantify the topology of the SAM structure, we subsample each SAM simulation using non-overlapping sets of 200 simulation time steps. We then compute a time-averaged EC value for the hydrogen bonding network within the interfacial layer denoted

as $\langle\chi\rangle$ for each subsample. The presence (or absence) of hydrogen bonds between water molecules was computed using the Luzar-Chandler criterion [190, 191]. For an interval of simulation time points $t \in [a, b)$ where $b, a \in \mathbb{Z}_+$ and $b - a = 200$, we compute the EC value of the hydrogen bonding network χ_t at each simulation time point. The time-averaged hydrogen bonding EC value is computed as:

$$\langle\chi\rangle := \frac{1}{(b-a)} \sum_{t=a}^{b} \chi_t \tag{3.3}$$

In other words, $\langle\chi\rangle$ captures the time-average topology of the system. The details outlining the practical computation of the EC for these simulations can be found in the supporting information and code shared in this manuscript.

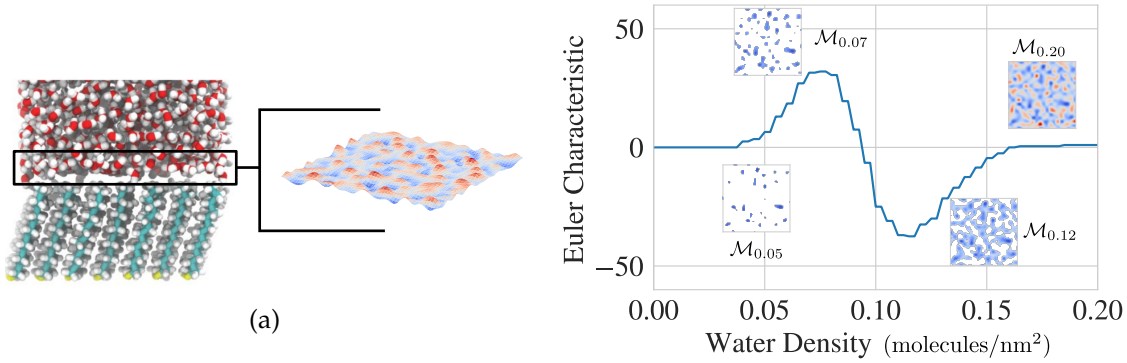
We also represent the SAM surface as a 2D manifold $\mathcal{M}$ and treat the time-averaged water density at the interfacial layer as a continuous function on the manifold $f : \mathcal{M} \to \mathbb{R}$ as shown in Figure 3.2. The manifolds are constructed by binning water molecule positions at the interfacial layer in a 20×20 grid, where each grid point accounts for 0.1 nm^2 area on the surface of SAM over the period of 200 simulation time steps. The accumulated bin data is then averaged to obtain a continuous water density function for our surface manifold $\mathcal{M}$. This representation matches the representation of Kelkar and co-workers used as input to a CNN to allow for direct model comparison [181]. We performed a manifold filtration to quantify spatial density fluctuations in water at the SAM surface with an EC curve. We recall that the EC curve is the set of paired filtration thresholds k_i and EC values χ_i of the filtered sublevel sets $\mathcal{M}_{k_i}$. The filtration thresholds k_i represent water densities given in units of molecules/nm^2.

A visualization of the EC curve for a time-averaged water density field derived from a SAM simulation (HFE = $33k_BT$) is shown in Figure 3.2. We illustrate the topological

changes in the manifold as we perform our filtration: $\mathcal{M}_{0.05} \subseteq \mathcal{M}_{0.07} \subseteq \mathcal{M}_{0.12} \subseteq \mathcal{M}_{0.20}$. We see in the first sublevel set $\mathcal{M}_{0.05}$ that connected components start to form. These are a direct result of local minima (e.g., areas of low water density) on the SAM surface, and result in a positive EC value. As the filtration threshold increases, there is an increase in the number of connected components representing areas of low water density and an increased EC value ($\mathcal{M}_{0.07}$). As the filtration threshold increases, we begin to pass saddle points in the density function where individual components merge, resulting in a single connected component with many holes (representing local maxima of the water density) and a corresponding negative EC value $\mathcal{M}_{0.12}$. The filtration then reaches a threshold in which the topology of the sublevel set is equal to the topology of the original manifold $\mathcal{M}_{0.20} = \mathcal{M}$. In this case, the original manifold $\mathcal{M}$ is a single connected component with an EC value $\chi = 1$. Details on the practical computation of the EC for sublevel sets can be found in the supporting information and code.

Figure 3.3 illustrates the input for a regression model derived from subsets of a SAM simulation (HFE = $33k_BT$ as labeled by INDUS). Figure 3.3 also reveals that topological representations are invariant to different types of deformation of the data [192]. Specifically, we see that both the hydrogen bond network and water density manifold topology varies significantly over time, but their corresponding topologies (captured by $\langle\chi\rangle$ and EC curves) are similar.

Before building a linear regression model that predicts HFE from the SAM structure, we first determined if there was indeed a relationship between the topology of the SAM interfacial water structure and the HFE. In Figure 3.4, we find that there is a strong correlation between the topology of the time-averaged water density field at the SAM surface and its emergent HFE. Specifically, the local minima and maxima (i.e., critical points) of the density change in both shape and magnitude as the HFE for the SAM is changed. These topological changes are captured effectively using the EC curve. At low HFE values (e.g., HFE = $33k_BT$) we see that there are many critical points with relatively low



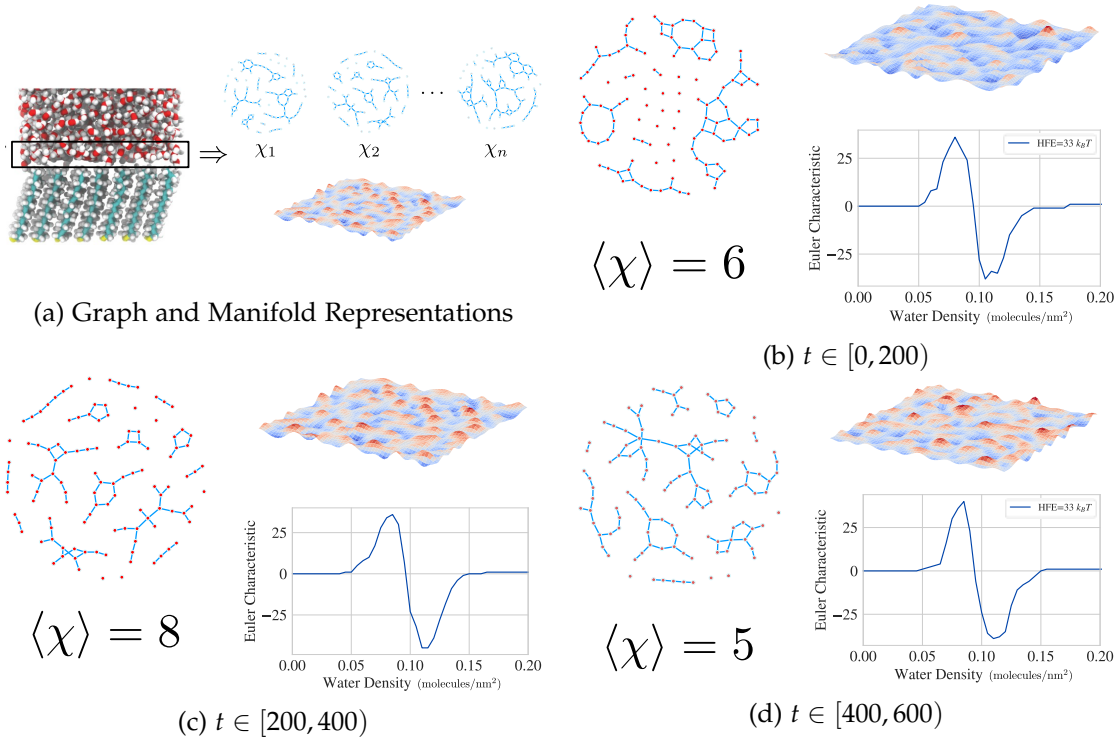
(a)

(b)

Figure 3.2: (a) Interfacial water density field derived from a SAM simulation. The 2D density field is represented as a manifold $\mathcal{M}$ with a continuous function $f : \mathcal{M} \to \mathbb{R}$ that maps each manifold location to its corresponding water density value visualized here by color (blue = low, red = high) and surface height. (b) The EC curve obtained from level set filtration of the density field. The EC curve is created by thresholding the density field function/manifold, creating multiple nested submanifolds $\mathcal{M}_{k_i}$, and then computing the EC of each submanifold. The EC curve is constructed from the paired values $\{k_i, \chi_i\}$. We visualize the corresponding submanifolds as the density threshold increases from $k_0 = 0$ to $k_n = 0.2$.

magnitude; as HFE is increased (e.g., HFE = $100k_BT$) we see fewer critical points but the corresponding magnitude of these critical points is increased, which we see is reflected in the EC curves.

We further highlight the relationship between the SAM topology and the HFE by performing principal component analysis (PCA) on all 1600 sample EC curves derived from 40 simulations with precomputed HFE values. The EC curve of each sample is represented as a vector $x_j \in \mathbb{R}^m$ where $m = 20$ is our number of filtration thresholds and the entries of x_j are the EC values of the sublevel sets. Each vector is stacked into a matrix $[x_1\ x_2\ \cdots\ x_n]^T \in \mathbb{R}^{n\times m}$. We apply a singular value decomposition to this matrix and visualize the data projected onto the two leading principal components; these components capture the low-dimensional structure of the EC and show that this is highly correlated with the HFE. These results confirm that the topology of the SAM affects the



(a) Graph and Manifold Representations

(b) $t \in [0, 200)$

(c) $t \in [200, 400)$

(d) $t \in [400, 600)$

Figure 3.3: (a) Illustration of the graph and manifold data representations derived from the SAM simulation shown in Figure 3.2 at HFE = $33k_BT$,. (b),(c) and (d) Representative hydrogen bond networks (graphs) and time-averaged density fields (manifolds) derived from the interfacial water in SAM molecular simulations over different time frames of the same simulation. They also contain the time-averaged graph EC $\langle\chi\rangle$ and the EC curve created from a filtration of the associated density field. Each SAM simulation is split into multiple subsets of a single simulation $t \in [a, b)$, for which a corresponding density field EC curve and time-averaged graph EC is computed. We note the stability of both the time-averaged graph EC and the EC curve. The density fields and graphs are visually very different, but the topological measures of the graphs and density fields are almost identical throughout the simulation. This demonstrates the robustness of these topological descriptors in capturing the underlying characteristics of molecular simulations.

HFE and that such topology can be captured using the EC.

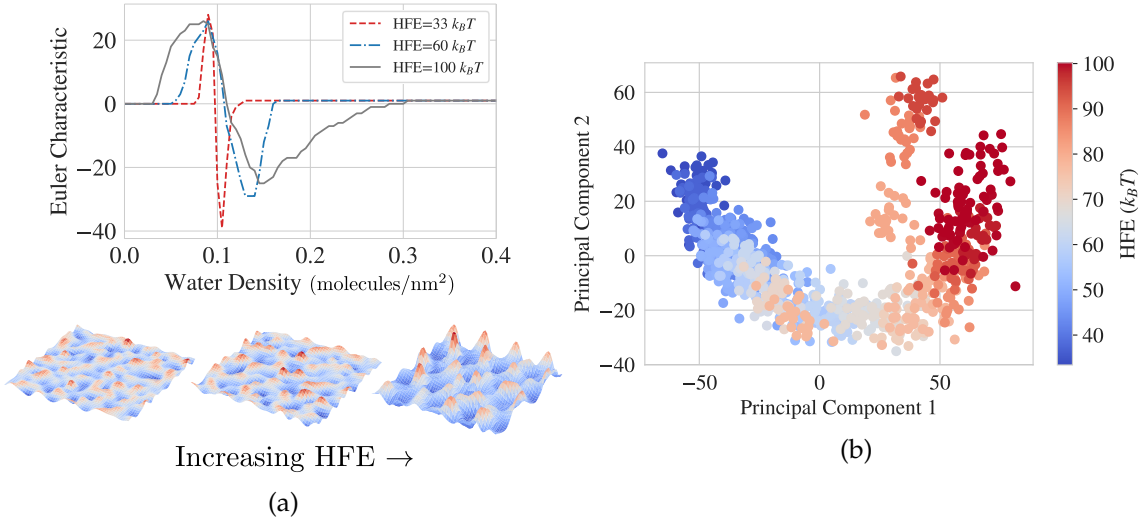


Figure 3.4: (a) EC curves for multiple density fields taken from simulations of SAMs with increasing HFE values. The impact of increased HFE on the SAM interfacial water layer is directly correlated with changes in the resulting EC curve. (b) Principal Component Analysis (PCA) is conducted on the EC curves for each density field from the set of SAM simulations. From (a) we see a continuous change in the EC curve that correlates with the HFE of the given SAM, in (b) we capture this continuous change and visualize a data structure that correlates directly with the HFE of the simulations through the first two principal components.

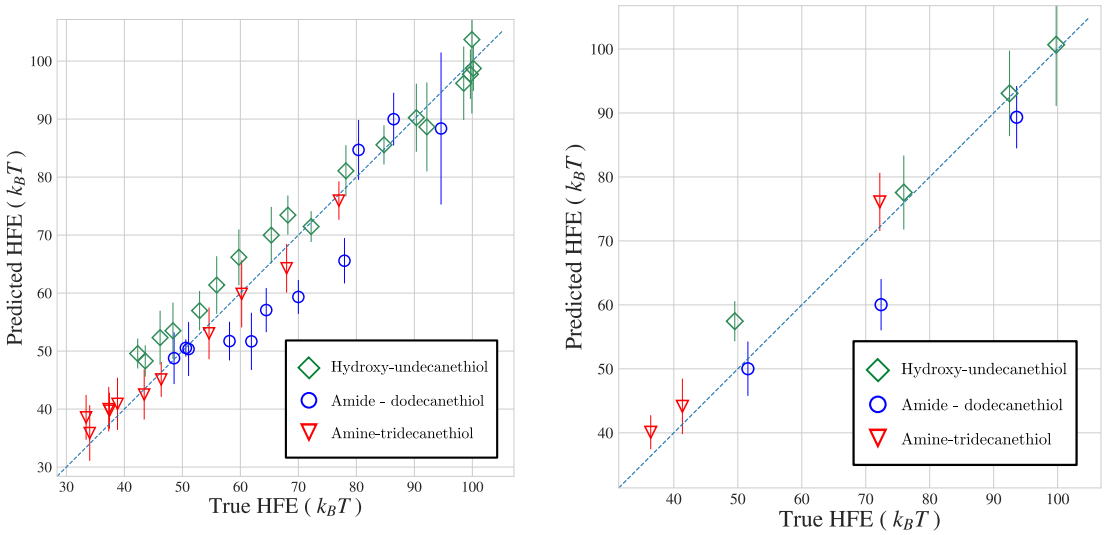
We next develop a linear regression model by using the water density EC curves and hydrogen bonding network EC values $\langle\chi\rangle$ as model inputs. The model chosen is a linear support vector machine (SVM) model taken from the `LIBSVM` library [193]. We train the linear model using a set of 40 simulations with precomputed HFE values via the INDUS method (1600 samples). Once the model has been trained, we test its prediction accuracy on a completely separate set of 10 SAM simulations (400 subsampled points). Our goal is to accurately predict the true HFE (computed via INDUS) for this separate set of simulations. The regression results for both model training and model testing are shown in Figure 3.5. We see that the linear model is able to predict the HFE for the testing set of simulations with little error (RMSE = 2.2 k_BT). Moreover, we have found that this model improves substantially the results of a previously-developed 2D CNN (RMSE = 5.8k_BT) for identical training/testing splits [181]. We also note that the computations required

to both train and predict with the linear model are minimal compared to INDUS and to machine learning models. We thus conclude that the topological approach can be used in the analysis of high-throughput simulations or in screening for surfaces with optimal chemical or physical properties. We also note that the incorporation of both hydrogen bond network and surface water density are critical in the performance of the model.

We note that the linear prediction model uses topological information obtained from both graph and manifold representations of the SAM. We have found that, when either representation (density field or hydrogen bond network) is used independently for developing a prediction model, the resulting prediction RMSE values increase. Moreover, we have found that such models lead to severe underprediction of HFE for amide-dodecanethiol simulations. This may be explained by the tendency for amide end groups to form substantial hydrogen bonds with other amide end groups rather than water, which is unique for the surfaces studied [194, 195, 196]. This added complexity is captured effectively when combining topological information from both graph and manifold representations and highlights how such representations can provide complementary information.

3.3.2 *Solvent-Mediated Reactivity in Acid-Catalyzed Reactions*

We now use the EC for understanding and predicting solvent-mediated reactivity of acid-catalyzed reactions based on the topology of water and cosolvent mixtures. Previous work has demonstrated that varying the cosolvent type and concentration in a cosolvent/water mixture impacts the relative reactivity of acid-catalyzed reactions for biomass conversion [182, 63]. Walker and co-workers analyzed the influence of solvation towards reactivity by studying the structure of water in molecular simulations of a single reactant molecule in different cosolvent/water mixtures (snapshot shown in Figure 3.6). We demonstrate that the EC can be used to predict the solvent-mediated changes in reac-



(a) Training.

(b) Testing.

Figure 3.5: (a) Training data parity plot of predicted versus INDUS derived HFE. Linear regression is conducted using the corresponding EC curve and averaged graph EC $\langle\chi\rangle$ as inputs. The training data set is split into two portions, one for model training and the other for model validation. Predictions on the validation dataset are very accurate and suggest the linear model can obtain high accuracy in HFE prediction (RMSE = 2.3 k_BT). (b) Testing data parity plot of predicted versus INDUS derived HFE. The testing data consists of a completely separate set of SAM simulations not used in model training. For each simulation EC curves and $\langle\chi\rangle$ are measured. The trained linear model is then used to predict the HFE for the separate set of SAM simulations. The results demonstrate a high level of accuracy and low prediction error (RMSE = 2.2 k_BT), which is comparable to training set accuracy as expected. Error bars in both plots represent a single standard deviation from the mean.

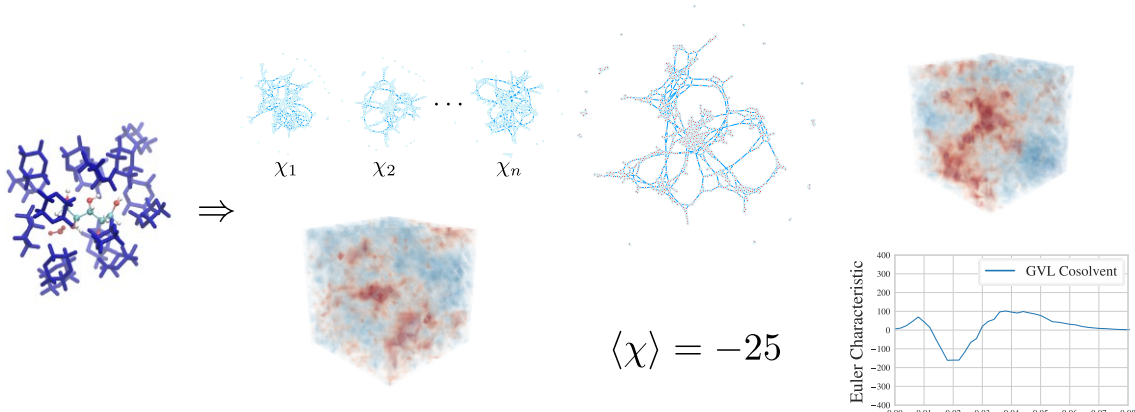
tivity as the concentration and type of cosolvent are varied, a task that previously used 3D CNNs [63]. The organic, polar aprotic cosolvents modeled in this study are dioxane (DIO), γ-valerolactone (GVL), tetrahydrofuran (THF), dimethyl sulfoxide (DMSO), acetonitrile (ACN), and acetone (ACE). Biomass-derived reactants modeled in this study are ethyl tert-butyl ether (ETBE), tert-butanol (TBA), cellobiose (CEL), glucose (GLU), levoglucosan (LGA), 1,2-propanediol (PDO), fructose (FRU), and xylitol (XYL); further details about the dataset can be found in the supporting information and work of Chew and co-workers [63].

We again represent the MD simulation data as both graphs and manifolds and assess whether topological descriptors alone can predict solvent-mediated reaction rates (Figure 3.6). We propose this method of analysis because there is an established history of water enriched structures playing an important role in understanding and predicting reactivity [197, 182, 198, 199]. These structures can be quantified through the EC and EC filtrations and used directly in prediction. The simulations contain a single reactant molecule centered in a 4 nm^3 cube surrounded by water and cosolvent in specified weight percentages. We subsample the 2 ns simulations in sets of 200 picoseconds (i.e. 20 frames) and produce both graph (hydrogen bonding) and manifold (water density) representations. To understand the topology of the hydrogen bonding networks in the simulations, we consider the EC for the water-water hydrogen bonding network χ_{ww}, the cosolvent-reactant hydrogen bonding network χ_{cr}, and the water-reactant hydrogen bonding network (χ_{wr}). Hydrogen bonds in each case were computed using the Luzar-Chandler criterion. For each of these networks, we construct a time-averaged EC value for the subsampled simulations $\langle\chi\rangle$. The manifold $\mathcal{M}$ for this system is now the entire simulation space (versus the surface of the SAMs described previously), which consists of a 3D cube with a continuous function $f : \mathcal{M} \to \mathbb{R}$ representing the time-averaged density of water in each simulation subsample. We ignore the structure of the corresponding cosolvent topology because it is directly related to the water topology (high density water implies low-density cosolvent).

We construct our manifold and function representation by placing a $20 \times 20 \times 20$ grid centered on the reactant molecule. We then accumulate and bin water positions within each grid point (each representing a 0.2 nm^3 volume) over the 200 picoseconds of simulation time. The accumulated data is then averaged over the time frame and represents a continuous density function over the simulation space. The EC curves in Figure 3.6 look slightly different to those for our previous 2D system. This system, now in 3D, has a 3rd Betti number (β_2), which quantifies the number of voids/cavities that appear during filtration. These voids are associated with pockets of high water density (e.g., local maxima) within our 3D manifold and result in a second peak in the EC curve $\chi = \beta_0 - \beta_1 + \beta_2$. Details for the practical computation of the EC for these 3D manifolds can be found in the supporting information and code.

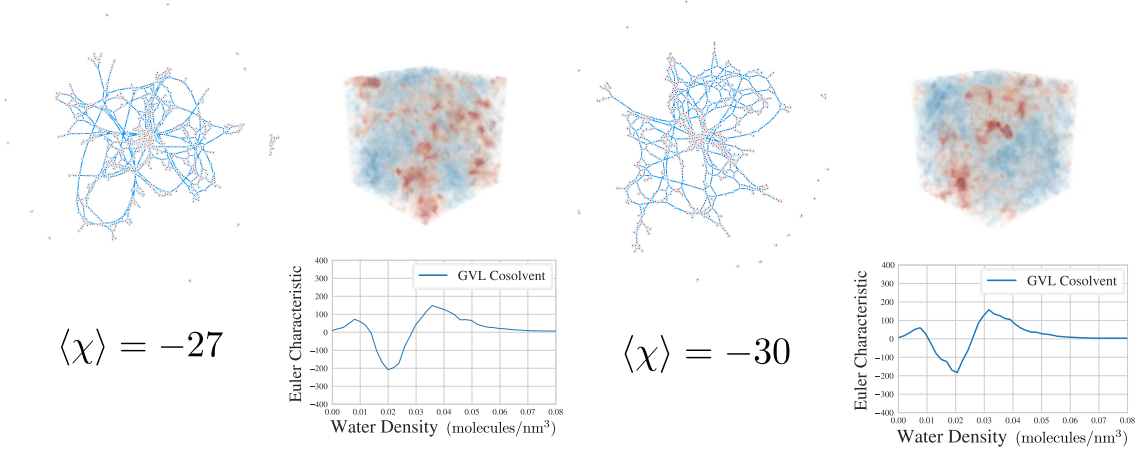
Figure 3.6 provides another demonstration of the robustness of these topological methods. We illustrate the changes in the water density and hydrogen bonding network $\langle \chi_{ww} \rangle$ during the course of a single MD simulation. Visually, these graphs and manifold functions appear to be distinct but, when quantified and compared through the EC, they are almost identical (indicating that they are topologically close). This result is of practical relevance, because it shows that the EC can be used to monitor the dynamics of topology (e.g., to determine when the system is undergoing a topological transition or has reached steady-state). From the ECs of Figure 3.6, for instance, it appears that the system quickly reaches a topological steady-state and thus the MD simulation can be terminated early to reduce computational time.

We developed a linear regression model that takes as input the corresponding EC curves and hydrogen bonding EC values ($\langle \chi_{ww} \rangle, \langle \chi_{cr} \rangle, \langle \chi_{wr} \rangle$) and outputs the experimentally determined change in reaction rate $\sigma = \log_{10}(k_{org}/k_{H_2O})$ where k_{org} represents the reaction rate in a cosolvent/water mixture and k_{H_2O} represents the reaction rate in pure water [182]. We train the linear regression model on a set of 76 cosolvent and reactant combinations (760 subsampled points) and test our model on a set of 32 different reactant and solvent combinations (320 subsampled points), which is the same data



(a) Graph and Manifold Representations

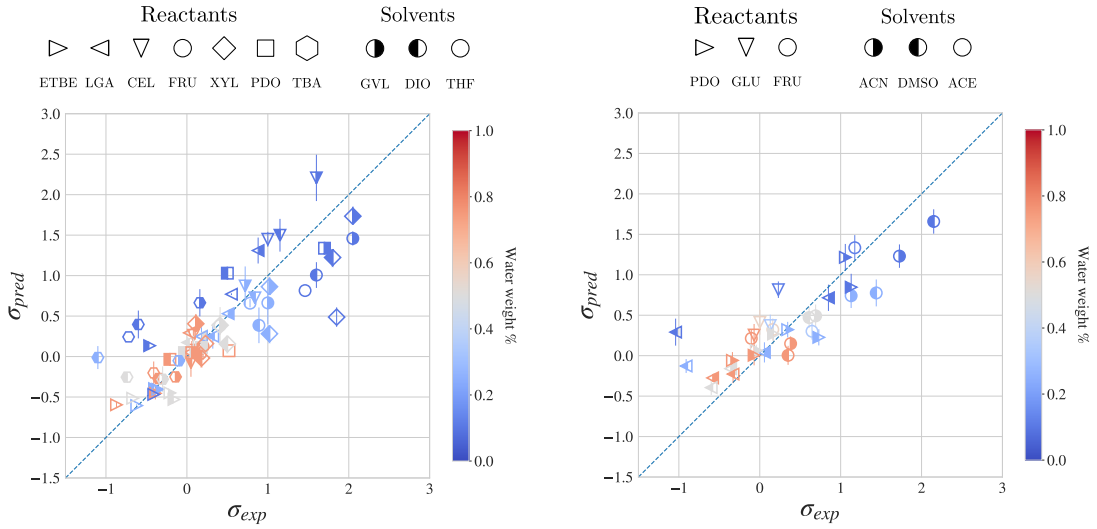
(b) $t \in [0, 200)$



(c) $t \in [200, 400)$

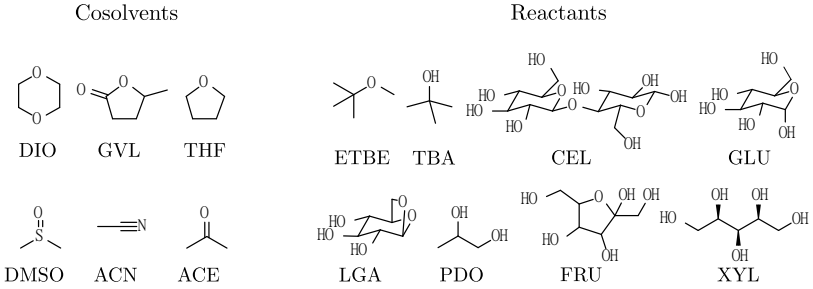
(d) $t \in [400, 600)$

Figure 3.6: (a) Illustration of the graph and manifold data representations derived from the acid-catalyzed reaction simulations. (b),(c), and (d) Representative water-water hydrogen bond networks (graphs) and time-averaged density fields (manifolds) derived from the water in the molecular simulations over different time frames. They also contain the time-averaged graph EC $\langle\chi_{ww}\rangle$ and the EC curve created from a filtration of the water density field. The density fields and graphs are visually different, but the EC values are similar throughout the simulation. These results demonstrate the robustness of topological descriptors.



(a) Training Data Regression.

(b) Testing Data Regression.



(c) Structures of reactants and co-solvents.

Figure 3.7: (a) Training data parity plot of predicted versus experimental σ (change in reaction rate). The EC curve and averaged graph EC values ($\langle \chi_{ww} \rangle$, $\langle \chi_{cr} \rangle$, $\langle \chi_{wr} \rangle$) are used as inputs to a linear model. Predictions on the training dataset are accurate (RMSE = 0.39) and suggest a linear model could be used to obtain high accuracy in the prediction of reactivity trends (σ). (b) Testing data parity plot of predicted versus experimental σ. An unseen test set of acid-catalyzed reaction simulations are created for different cosolvents and solutes. From this data, the corresponding EC curve and graph EC values are computed. The trained linear model is used to predict the experimentally verified reactivity increase for the separate set of acid-catalyzed reaction simulations. The results demonstrate a high level of test set accuracy with low prediction error (RMSE = 0.42). Error bars in both plots represent a single standard deviation from the mean.

training/testing split used to evaluate the CNN developed in the work of Chew and co-workers [63]. Figure 5.1 lists the different potential combinations of reactant, cosolvent, and cosolvent/water ratios that are used in both training and testing. Figure 5.1 also shows the accuracy of the linear model in both training (RMSE = 0.39) and testing (RMSE = 0.42), from which we can conclude that the simple linear model is able to accurately predict the change in reactivity for these chemical systems. We can compare these results directly to the work of Chew and co-workers, where the authors used 3D CNNs that contain up to ∼172,417 parameters, compared to the 23 parameters used in our linear model. The topological approach achieves accuracy superior to the trained 3D CNN on the same testing set (RMSE = 0.48). We also note significant improvements in the computational resources needed for training the models. Our linear model takes approximately 2 minutes to train (this time includes computation of ECs), while the 3D CNN can take up to 2 hours. Furthermore, our linear model does not require a search for optimal hyper-parameters or 3D CNN architecture, further reducing the needed computational resources. These results highlight the desirable scalability of topological characterizations based on the EC.

An added benefit of linear models and our topological characterization of MD simulation data is interpretability, which in the physical sciences is often as important as prediction accuracy [200]. Figure 3.8 contains an analysis of simulations of fructose in varying cosolvent/water mixtures, all at the same cosolvent/water weight ratio (90%/10%). Figure 3.8 illustrates the differences in water topology that occurs when the chemistry of the cosolvent is altered. We focus on two particular cosolvents: THF and DMSO. For fructose, the change in reaction rate is highest when in a DMSO/water mixture ($\sigma = 1.7$) and lowest when in a THF/water mixture ($\sigma = 0.8$). We note that this corresponds with large changes in the topology of the water density. In THF, water is agglomerated in large clusters near the reactant, which reduces the total number of topological features in the water density function (e.g., connected component, holes, voids) and dampens the magnitude of the peaks and valleys in the EC curve, which is consistent with the find-

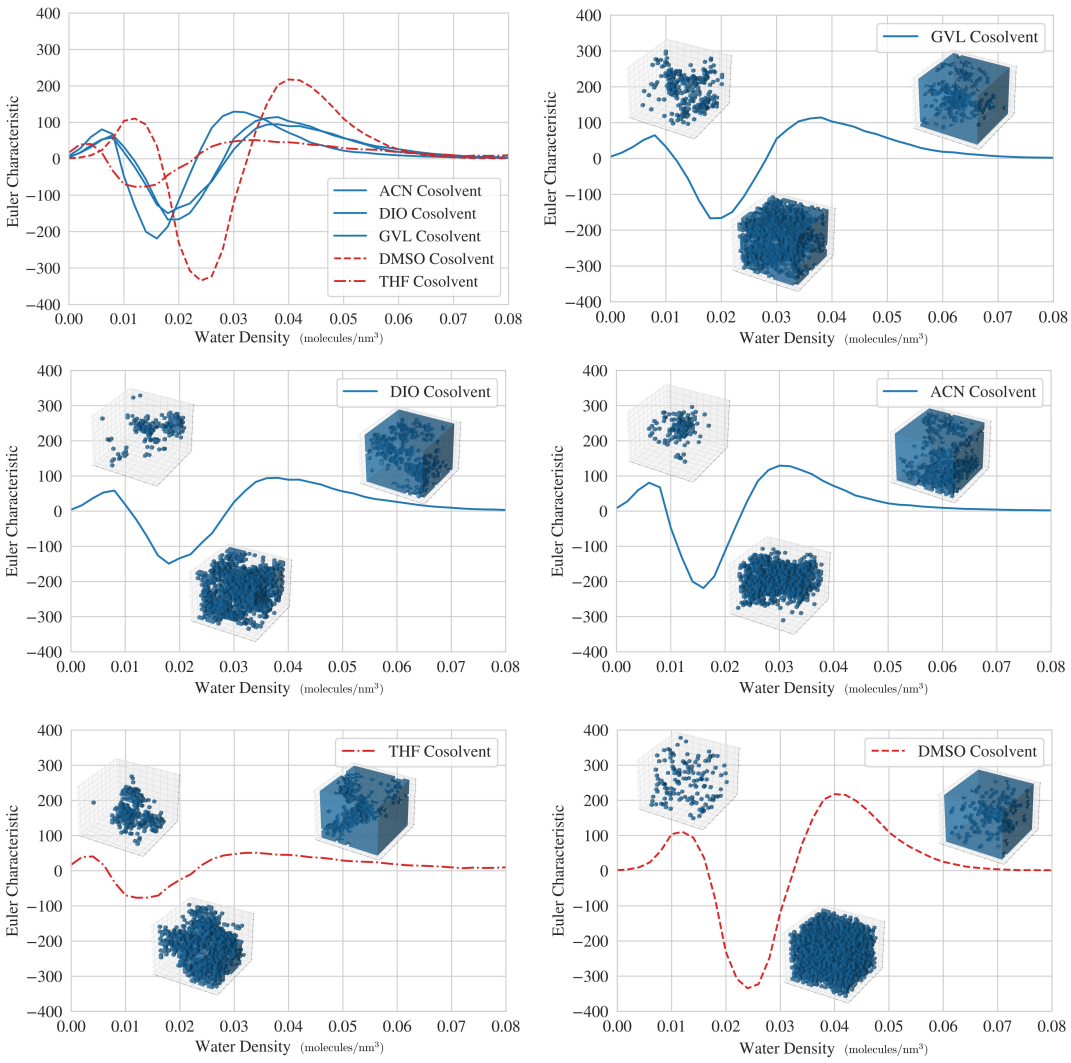


Figure 3.8: EC curves for fructose solvated in different cosolvent/water mixtures along with the representative submanifolds during various points in the filtration. Each of the EC curves are computed with a 90 wt% cosolvent, 10 wt% water solution. Many of the simulations (ACN, DIO, GVL) behave similarly from a topological perspective, and each have a similar impact on the reactivity of fructose. The EC curves for THF and DMSO differ from the EC curves of the previously mentioned solvents. DMSO interrupts water-water interactions resulting in a larger number of high and low density areas, which manifests in an increased number of connected components, holes, and voids in the density field captured by the EC curve. The opposite occurs with THF where we see fewer, but larger, clusters of high and low density water; this suggests that THF increases the interactions between water molecules and causes larger clusters of water to form.

ings of Chew and co-workers [201]. The sublevel sets at points in the filtration are also illustrated (Figure 3.8), further confirming this result. The opposite holds true for water in DMSO, here we see a larger number of high-density and low-density areas on our manifold increasing the peak and valley magnitude of the EC curve. This indicates that DMSO is interrupting the interactions between water molecules and reducing the total amount of water molecules near fructose. This behavior can increase selectivity of the acid-catalyzed reaction of fructose, where the shielding of subsequent products (e.g., 5-hydroxymethylfurfural) inhibits the formation of undesired products (e.g., levulinic acid) as shown in the findings by Mushrif and co-workers [197].

Chapter 4

TOPOLOGICAL DATA ANALYSIS & PERSISTENCE HOMOLOGY

The contents of this chapter is published in [97]

4.1 Introduction

This chapter provides a rigorous introduction to, and example applications of, some of the advanced data analysis methods from the field of Topological Data Analysis. Statistical and signal processing techniques are the dominant paradigms used to analyze data. Unfortunately, these techniques provide limited capabilities to analyze certain types of datasets. Some interesting examples that highlight this limitation are the Anscombe quartet and the *Datasaurus dozen* datasets [202, 203]. These datasets are *visually* distinct but they have the same descriptive statistics (e.g., *mean, standard deviation* and *correlation*). An illustration of this issue is provided in Figure 4.1; here, the two datasets have the same mean and standard deviation along both dimensions and have the same correlation between dimensions. However, it is clear that these datasets define objects with different geometric features (shape).

The recent application of algebraic and computational topology to data science has led to the development of a new field known as Topological Data Analysis (TDA) [204]. TDA techniques are based on the observation that data (e.g., a set of points in a Euclidean

space) can be interpreted as elements of a geometric object. As the name suggests, TDA utilizes techniques from computational topology to quantify the shape of data [205]. Fundamentally, topology studies geometric and spatial relations that persist (are stable) in the face of continuous deformations of an object (e.g., stretching, twisting, and bending).This perspective brings a number of advantages over other data analysis techniques [204, 192]:

- Topology studies data in a manner that is independent of the chosen coordinates.
- Topology studies data in a way that minimizes sensitivity to the choice of metric.
- Topology generalizes known graph theory techniques to high-dimensional spaces.
- Topology is robust to large quantities of noise.

The main focus of this chapter is a technique in the field of TDA known as *persistence homology* [206, 207]. The goal of persistent homology is to *identify and quantify* topologically dominant features within the data in the form of basic (low-dimensional) topological features such as connected components, holes, voids, and their generalizations. This information can then be used by statistical and machine learning techniques to perform regression, classification, hypothesis testing, and clustering tasks [208, 209, 210, 211, 212, 213]. The TDA methodology is summarized in Figure 4.2. It is important to emphasize that TDA is a dimensionality reduction technique that maps data from its original high-dimensional space to a low-dimensional space that it is easier to understand and visualize. This is similar in spirit to principal component analysis (PCA), which is a statistical technique that projects data into a low dimensional space by extracting latent variables (principal components) that contain maximum information in terms of variance.

TDA has been used in different scientific and engineering domains. In the medical imaging field, persistent homology has been used in studying brain dendrograms [214] and in identifying brain networks in children with ADHD and autism [113]. In material science, these techniques have been used to characterize complex craze formations [215], to analyze hierarchical structures in glasses [216], and in materials informatics [217].

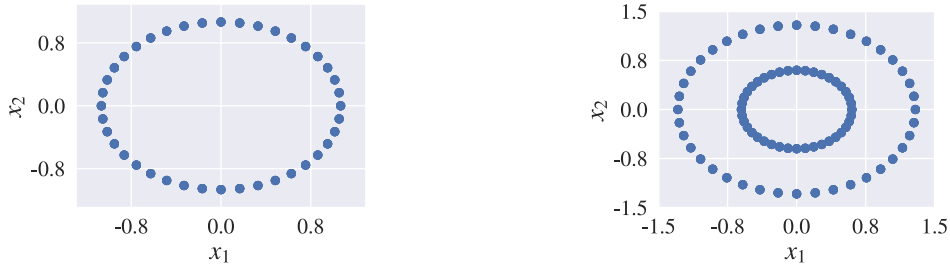


Figure 4.1: Datasets with the same mean along the x_1, x_2 dimensions (zero), the same standard deviation along both dimensions (1/2) and the same correlation between dimensions (zero). While the statistical descriptor values (i.e., first and second moments of a 2D Gaussian ellipse) are identical, the geometric objects that they define are different.

They have also been used in high throughput screening of nanoporous materials, such as zeolites [175]. TDA has been used in the analysis of dynamical systems and time series [218, 219] to study gene expression [220] and the dynamics of Kolmogorov flows and Raleigh-Bernard convection [221]. TDA has also been used in the analysis of time-varying functional networks [222]. In chemistry and biochemistry, persistent homology has been used to characterize protein structure, flexibility, and folding [223]; as well as a metric for understanding membrane fusion [224], and used to predict fullerene stability [225].

In this chapter, we provide a concise summary of relevant concepts and computational methods of TDA from the perspective of chemical engineering applications. We show how to apply persistent homology techniques to analyze datasets described by point clouds and functions in high dimensions and we discuss fundamental stability results of topological features in the face of perturbations. We present multiple case studies with complex synthetic and real experimental datasets to demonstrate the benefits of TDA. Specifically, we show that TDA extracts informative features from complex datasets that correlate strongly with emerging features of practical interest. For instance, we show that the topological features of a 3D solvent environment explains reactivity in such an environment and that topological features of liquid crystals explain composition of its environment. These two results are presented for the first time in this chapter. Moreover, we show how to characterize topological features of scatter fields from flow cytometry

experiments. Our work seeks to open new research directions and applications of TDA in chemical engineering.

4.2 TDA Basics

The first part of this chapter develops the mathematical basis for TDA (see Sections 4.3-4.5) while the second part explores the application of TDA to different chemical engineering problems. We develop this section as a simplified overview of TDA in order to equip the reader with the knowledge necessary to immediately review TDA applications of interest. The focus of TDA is to capture and record the evolution of the topology of a dataset at different scales which is measured through a filtration (see Section 4.5). Figure 4.2 is an example of applying TDA to two simple point clouds. On point clouds, a filtration is done by expanding balls of radius (ϵ) around each data point, and connecting those points for which the given balls overlap. This changes the topology of the data representation as the expansion proceeds, resulting in the appearance and disappearance of topological features such as connected components (represented by H_0) and holes (represented by H_1). A similar filtration can also be constructed on continous functions (see Section 4.5.3) or on discrete representations of continuous functions such as images (see Section 4.5.4). Regardless of how the filtration is performed, the time of appearance (birth) and the time of disappearance (death) of the topological features during the filtration constructs the persistence diagram. The persistence diagram is a scatter plot for which the x axis is birth and the y axis is death. All topological features that appear and disappear during the filtration are recorded as points with coordinates x = birth and y = death and persistence ($y - x$) within their representative group (e.g., H_0, H_1). This scatter plot encodes (reduces) the high-dimensional structure of the dataset during the filtration and can be vectorized (see Section 4.5.2) for use in statistical and machine learning models and methods.

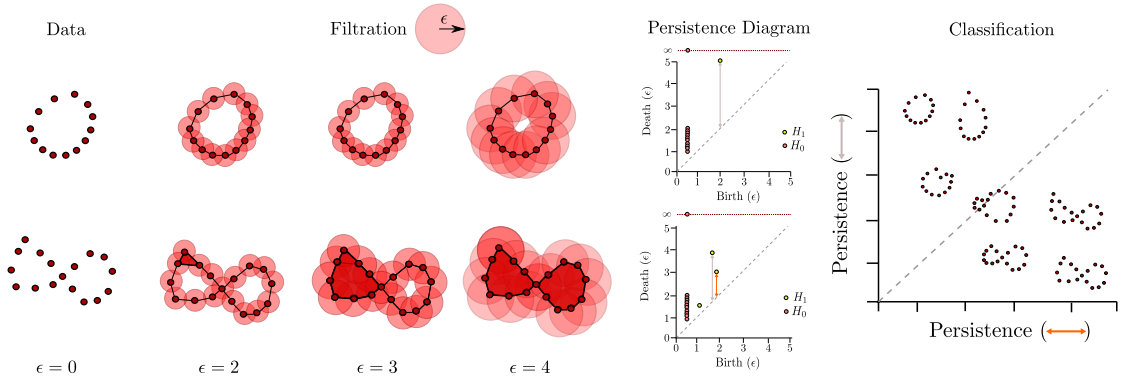


Figure 4.2: Persistence homology methodology for point clouds: each point cloud is converted into a geometric object via a filtration where the topology is measured at each point in the filtration. At certain points in the filtration topological features, such as the holes above, appear and are eventually filled. The ϵ value at the appearance and disappearance of these features are recorded as birth and death in the filtration. The birth and death of the topological features are represented as points in a persistence diagram, with x=birth and y=death and persistence defined as $(y - x)$. The persistence diagram encodes the topological evolution of the data during the filtration and can be used directly to separate point clouds of different shape and cluster those of similar shape. In this illustration we create a representative classification plot that demonstrates the separation of example point clouds based upon the persistence of the largest and second largest (which is zero in some cases) hole(s) that appear and disappear during the filtration.

4.3 Fundamental Concepts of TDA

We discuss fundamental concepts and computational methods of TDA. We first introduce the notion of simplicial and cubical complexes, which are the basic geometric constructs used to represent data objects. The representation of data objects as a simplicial or cubical complex enables the use of methods from *simplicial and cubical homology* to quantify the shape of the data in terms of connectedness and topologically important features [226, 227]. In the following discussion, we use $\mathbb{R}$ to denote the set of real numbers and $\mathbb{Z}$ to denote the set of integer numbers..

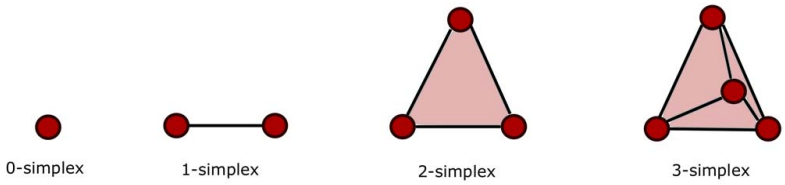


Figure 4.3: Examples of k-dimensional simplices for $k = 0, 1, 2, 3$. A simplex is a generalization of a triangle in high dimensions. 0 simplices are vertices (points), 1-simplices are edges, 2-simplices are triangles, and 3-simplices are tetrahedra.

4.3.1 *Simplicies and Simplicial Complexes*

A simplex is a generalization of a triangle from 2D to other dimensions (e.g., a tetrahedra is a 3-dimensional simplex). Simplices spanning dimension $k = 0$ to dimension $k = 3$ are shown in Figure 4.3. The formal definition of a simplex is as follows.

Definition 4.3.1. k-simplex: A *k-simplex* is a convex hull spanned by k+1 affinely independent points $v \in \mathbb{R}^m$ and is denoted as:

$$\sigma = [v_0, v_2, ..., v_k] \tag{4.1}$$

Some interesting properties of simplices are:

1. An m-face of simplex σ is the convex hull of any of its nonempty subsets.
2. The m-face is a simplex.
3. A 0-face is a **vertex**.
4. A 1-face is an **edge**.

A *simplicial complex* (denoted as $\mathcal{K}$) is obtained by connecting (glueing) simplices, as shown in Figure 4.4a. We denote the dimension of a given simplex or simplicial complex as $\dim(\cdot)$.

Definition 4.3.2. Simplicial Complex: A simplicial complex $\mathcal{K} \subset R^m$ is a collection of simplices that satisfies the following properties:

1. Every face of an elemnt of $\mathcal{K}$ is also in $\mathcal{K}$
2. A nonempty intersection of simplices $\sigma_1, \sigma_2 \in \mathcal{K}$ is a face of σ_1 and σ_2
3. The dimension of $\mathcal{K}$ is the highest dimension of its simplices: $\dim(\mathcal{K}) = \max(\dim(\sigma) : \sigma \in \mathcal{K})$

A simplicial complex is used to represent the topological characteristics of objects. One example of this is found in finite element analysis where simplicial complexes (also known as triangulations) are used to represent domains over which partial differential equations are solved [228]. In Figure 4.4b we represent a geometric object (a ring) as a simplicial complex. We see that the central hole of the ring, which is its main topological feature, is preserved in its representation as the simplicial complex. These objects are thus said to be homotopy equivalent (represented by the notation $\simeq$). Homotopy equivalence identifies spaces which can be deformed continuously into one another without cutting or tearing (e.g., via stretching) [229]. The flexibility of simplicial complexes allows us to create a homotopically equivalent representation of any shape encountered in practice. As we will see, algebraic calculations can be applied to simplicial complexes to quantify the features of the original object [226].

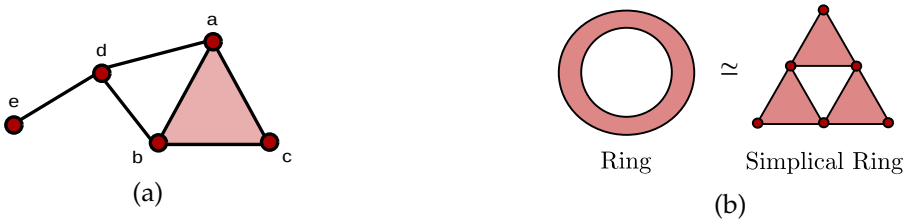
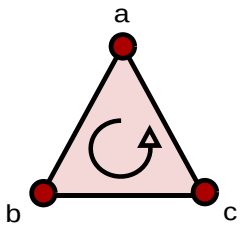
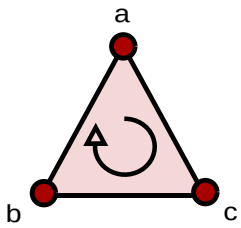


Figure 4.4: (a) A simplicial 2-complex created by connecting a 2-simplex (a triangle) and multiple 1-simplices (edges). This simplicial complex contains one hole. (b) A geometric object (a ring) and its simplicial complex representation using 2-simplices. Both shapes contain an empty hole and are homotopy equivalent.



(a) Simplex represented by ordered set $\{a, b, c\}$



(b) Simplex represented by ordered set $\{b, a, c\}$

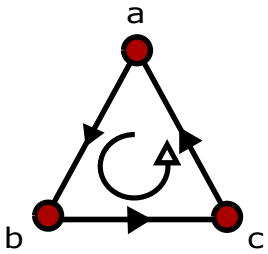
Figure 4.5: Possible orientations of a 2-simplex.

4.3.2 *Simplicial Homology*

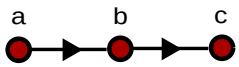
Simplicial homology provides computational techniques to study topological spaces that are represented as simplicial complexes. We make some basic definitions that are necessary to explain the working principles of these techniques.

Definition 4.3.3. Simplex Orientation: The orientation of a k-simplex is given by the ordering of the vertices in the simplex $[v_1, v_2, ..., v_{k+1}]$. Two orderings can define the same orientation if and only if they differ by an even permutation; thus, there are only two allowable orientations of a simplex.

An example of the possible orientations of a 2-simplex is shown in Figure 4.5. We can see that only two orientations are possible for this simplex. Here, each oriented simplex is equal to the negative of the simplex with opposite orientation; mathematically, this is stated as $[a, b, c] = -[b, a, c]$.



(a) 1-Complex ($\mathcal{K}_1$)



(b) 1-Complex ($\mathcal{K}_2$)

Figure 4.6: Examples of 1-simplicial complexes with the same number of vertices; $\mathcal{K}_1$ contains a hole while $\mathcal{K}_2$ does not.

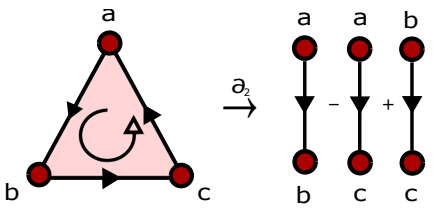


Figure 4.7: Visualization of the boundary operation (∂_2) applied to a 2-simplex. The boundary operator maps the 2-simplex onto its bounding 1-simplices. The $[a, c]$ simplex is inverted to retain lexicographical ordering.

4.3.3 *Cycles, Holes, and Homology Groups*

In simplicial homology, we want to identify cycles in a given object and we want to know whether a given cycle is bounding a collection of higher dimensional simplices (if it does not, then the cycles is a hole). An example of this concept is presented in Figure 4.6; we see that complex $\mathcal{K}_1$ is a cycle that bounds an empty space which constitutes a hole, while the complex $\mathcal{K}_2$ represents a line with no holes. We now proceed to explore concepts that will allow us to systematically identify the presence of holes and cycles.

A simplicial k-chain on a complex $\mathcal{K}$ is used in the identification of holes in a simplicial complex, and is defined as follows.

Definition 4.3.4. Simplicial k-chains: A k-chain is a finite weighted sum defined on all k-simplicies within a complex $\mathcal{K}$:

$$\sum_{i=1}^{N} c_i \sigma_i \tag{4.2}$$

where $c_i \in \mathbb{Z}$ and N is the number of k-dimensional simplices. The set of k-chains on $\mathcal{K}$ is written as C_k. Typically, the coefficients are given by $c_i \in \{-1, 0, 1\}$ and we recall that a value of -1 inverts the simplex.

The *boundary operator* is a linear operator that maps the k-chains of a complex to its boundaries. The boundaries are the associated $(k-1)$-chains that make up the higher dimensional k-chain. A visualization of the boundary operator is presented in Figure 4.7.

Definition 4.3.5. Boundary Operator: For a set of k-chains (denoted as C_k) we define the

boundary operator ∂ as the mapping:

$$\partial : C_k \rightarrow C_{k-1} \tag{4.3}$$

The boundary operation on a general simplex σ with vertices $[v_0, v_2, ..., v_k]$ is shown in (4.4), where the vertex $\hat{v_i}$ is removed from the set of vertices in the summation. The boundary operation on a k-simplex maps the simplex to a summation of its $k-1$ faces.

$$\partial([v_0, v_1, ..., v_k) = \sum_{i=0}^{k+1} (-1)^i [v_0, ..., \hat{v_i}, ..., v_k] \tag{4.4}$$

We use the short-hand notation $\partial(C_k) = \partial_k$ to represent a boundary operation. We show an example for a complex $\mathcal{K}$ with $dim(\mathcal{K}) = k$ in (4.5); here, we see that the boundary operator is a mapping from the chains of a higher dimension to chains of lower dimension within the simplicial complex. We also note that the k-chain for dimensions greater than k and less than 0 are zero and that the boundary maps ∂_{k+1} and ∂_0 are zero maps.

$$0 \xrightarrow{\partial_{k+1}} C_k(\mathcal{K}) \xrightarrow{\partial_k} C_{k-1}(\mathcal{K}) \xrightarrow{\partial_{k-1}} ... \xrightarrow{\partial_1} C_0(\mathcal{K}) \xrightarrow{\partial_0} 0 \tag{4.5}$$

As an example, we apply the boundary operator to the simplicial chains in the complexes in Figure 4.6. We note that these complexes are built from sets of 1-simplices; thus, when we apply the boundary operator ∂_1, we obtain a set of vertices (0-simplices). The result of the operation on complex $\mathcal{K}_1$ is found in 4.6, where we invert the orientation of the $[a, c]$ boundary in order to retain lexicographical ordering:

$$\partial_1(\mathcal{K}_1) = \partial([a, b]) + \partial([b, c]) - \partial([a, c]) = (a - b) + (b - c) - (a - c) = 0 \tag{4.6}$$

and the operation on the 1-chain in the complex $\mathcal{K}_2$ is:

$$\partial_1(\mathcal{K}_2) = \partial([a, b]) + \partial([b, c]) = (a - b) + (b - c) = a - c \tag{4.7}$$

This example illustrates how simplicial homology uses basic boundary operations to identify cycles in a complex. We can see that the cycle formed by $\mathcal{K}_1$ is mapped to zero. In simplicial homology, a cycle is defined as a chain that is mapped to the *null space* or *kernel* of the boundary operator (denoted as $ker(\partial)$) and which is zero. With these basic definitions, we can now formally define *cycles* and *boundaries*.

Definition 4.3.6. Cycles: The k-dimensional *cycles* are given by:

$$Z_k = \ker(\partial_k) \tag{4.8}$$

where $\ker(\partial_k)$ is the kernel of the operator ∂_k.

Definition 4.3.7. Boundaries: The *boundaries* are given by:

$$B_k = \text{im}(\partial_{k+1}) \tag{4.9}$$

where $\text{im}(\partial_{k+1})$ is the image of the operator ∂_{k+1}.

In summary, the information contained in Z_k gives us the cycles of dimension k within a given complex and the information in B_k tells us whether or not a cycle is the boundary of a collection of higher dimensional simplices. It is also important to note that B_k is a subgroup of Z_k. Also, if a cycle is not a boundary, then it is known as a *hole*. We can summarize this information for any complex by defining the k-homology group H_k and the Betti number β_k. In simple terms, the H_k group contains the unique k-holes within a complex and β_k counts the number of unique k-holes in a complex.

Definition 4.3.8. k-Homology Group: The k-homology group H_k is given by the quotient group:

$$H_k = Z_k / B_k. \tag{4.10}$$

We illustrate H_k in Figure 4.8; here, we have two simplicial representations z and b, where z is a cycle that does not bound a higher dimensional simplex (we have $z \in Z_1$ and

$z \notin B_1$) while b does bound a higher dimensional simplex (we have $b \in B_1$ and $b \in Z_1$ as $B_1 \subseteq Z_1$). We can also see that both z and $z + b$ contain the same hole and thus they are homologically equivalent (their difference is a boundary). The homology group H_k formally defines this concept and states that, if a cycle $z_1 = z_2 + B_k$ and $z_1, z_2 \in Z_k$, then the two cycles are equivalent $z_1 \simeq z_2$ and are not independent elements of the group (H_k). This logic prevents counting the same topological feature multiple times.

Figure 4.8: An illustration of the homology group H_1. (a) z represents a cycle $z \in Z_1$ that is not a boundary ($z \notin B_1$), whereas $b \in Z_1$ is a cycle that bounds a 2-simplex ($b \in B_1$). (b) The cycle z is homotopy equivalent to $z + b$ ($z \simeq z + b$) and should not be counted as a separate hole.

Before we can formally define Betti numbers, we must define the *rank* of a group.

Definition 4.3.9. Group Rank: We define the $\text{rank}(Z)$ of a group Z as:

$$\text{rank}(Z) = \min\{|Y| : Y \subseteq Z, \langle Y \rangle = Z\} \tag{4.11}$$

where $|Y|$ represents the cardinality of set Y and $\langle Y \rangle$ represents the subgroup of Z generated by element of Y.

We will see that the rank of a group is analogous to the notion of the rank of a matrix (or to the dimension of a vector space). Specifically, it identifies the number of independent basis elements (known as generators) of a group. With this in mind, we define the k^{th}-Betti number as follows.

Definition 4.3.10. k^{th}-Betti Number: The k^{th}-Betti number β_k is the rank of H_k and is

given by:

$$\beta_k = \text{rank}(H_k) = \text{rank}(Z_k) - \text{rank}(B_k) \tag{4.12}$$

We illustrate these concepts using the complex $\mathcal{K}_1$ presented in Figure 4.6. This complex presents a hole; performing the necessary computations to obtain Z_1 and B_1 for $\mathcal{K}_1$ we find the that:

$$Z_1(\mathcal{K}_1) = \{(a,b) + (b,c) - (a,c)\} \tag{4.13a}$$

$$B_1(\mathcal{K}_1) = \{0\} \tag{4.13b}$$

We now perform the quotient operation to obtain H_1 and β_1 for $\mathcal{K}_1$:

$$H_1(\mathcal{K}_1) = \{(a,b) + (b,c) - (a,c)\}/\{0\} = \{(a,b) + (b,c) - (a,c)\} \tag{4.14a}$$

$$\beta_1(\mathcal{K}_1) = \text{rank}(\{(a,b) + (b,c) - (a,c)\}) = 1 \tag{4.14b}$$

We see that the H_1 group identifies the 1-dimensional holes within the complex, and β_1 counts the number of 1-dimensional holes in the complex. The same is true for all other dimensions $k \geq 0$ as long as $\dim(\mathcal{K}) \geq k$ because $H_k = 0$ for all $k > \dim(\mathcal{K})$. This concept becomes familiar when viewed from the perspective of linear algebra and matrices. One can consider the H_k group similar to the basis vectors for a given matrix, where in this case H_k defines the *topological* basis for a given shape and the Betti numbers (β_k) correspond to the rank of this bases and can be seen as the total number of unique topological features. The main difference is that for a given shape there can be multiple sets of topological bases for each dimension of the shape. The goal of TDA is to identify and compare the topological bases for each shape, similar to how one might compare the structure of matrices based upon their basis vectors and corresponding rank.

The 0^{th} Homology Group (H_0)

The 0^{th} homology group H_0 plays an important role in topological analysis. H_0 is the measure of the number of *connected components* in a complex. A *component* (or *subcomplex*) is defined as follows.

Definition 4.3.11. Subcomplex: A subcomplex is a subset $\mathcal{S}$ of a complex $\mathcal{K}$ such that $\mathcal{S}$ is also a complex.

Definition 4.3.12. Connected complex: A complex is connected if there exists a path made of 1-simplices from any vertex of the complex to any other vertex.

The group H_0 describes how many disconnected subcomplexes $\mathcal{S}$ there are within a given complex $\mathcal{K}$. A simple example is shown in Figure 4.9; here, we note that the first complex $\mathcal{K}_3$ has two disjoint subcomplexes, and the second complex $\mathcal{K}_4$ has a single connected component. The calculations for $\mathcal{K}_3$ are given by:

$$H_0(\mathcal{K}_3) = \{a, b, c, d\} / \{(a - b), (c - d)\} \tag{4.15a}$$

$$\beta_0(\mathcal{K}_3) = 2 \tag{4.15b}$$

and for $\mathcal{K}_3$ are:

$$H_0(\mathcal{K}_4) = \{a, b, c, d\} / \{(a - b), (b - c), (c - d)\} \tag{4.16a}$$

$$\beta_0(\mathcal{K}_4) = 1. \tag{4.16b}$$

4.4 Computational Methods for TDA

Now that we have developed a basic understanding of simplicial homology, we can begin to streamline computations through the tools of numerical linear algebra. In order to simplify our discussion, we will define a new k-chain where the coefficients $c_i \in \mathbb{Z}_2$

(a) 1-complex $\mathcal{K}_3$ (b) 1-complex $\mathcal{K}_4$

Figure 4.9: Example complexes with different number of connected components. $\mathcal{K}_3$ has two connected components and $\mathcal{K}_4$ has a single connected component.

where $\mathbb{Z}_2$ is the set of binaries $\{0, 1\}$ (rather than $c_i \in \mathbb{Z}$), for which $1 + 1 = 0$. With this new definition, we can remove the need for defining an orientation on a simplicial complex. In (4.17a)-(4.17c), we see that computational results are the same as those for the example provided in (4.6).

$$C_1(\mathcal{K}_1) = [a, b] + [b, c] + [a, c] \tag{4.17a}$$

$$\partial_1(\mathcal{K}_1) = \partial([a, b]) + \partial([b, c]) + \partial([a, c]) = (a + b) + (b + c) + (a + c) \tag{4.17b}$$

$$\partial_1(\mathcal{K}_1) = (1 + 1)a + (1 + 1)b + (1 + 1)c = 0 + 0 + 0 = 0 \tag{4.17c}$$

We can now couple the newly defined k-chain with the boundary operator to create what is known as a *boundary matrix* $\mathbf{B} \in \mathbb{Z}_2^{n \times m}$ where n represents the simplices of dimension $(k - 1)$ in $\mathcal{K}$ and m represents the simplices of dimension k in $\mathcal{K}$. The boundary matrix is built based on the following rules:

- The face of a simplex precedes the simplex in column (row) index.
- An entry of one is placed in position (j, i) if σ_i is a face of σ_j, otherwise it is zero.
- If two simplices are of the same dimension, lexicographic ordering is used (i.e., $[a, b] < [a, c] < [b, c]$).

For example, we take complex in Figure 4.4a and compute the H_1 group. We show matrix $\mathbf{B}$ for ∂_0 in Table 4.1, for ∂_1 in Table 4.2, and for ∂_2 in Table 4.3. Again, the boundary

matrix associated with ∂_1 is mapping the 1-simplices of the complex to its boundaries and similarly for ∂_2.

∂_0	[a]	[b]	[c]	[d]	[e]
[o]	0	0	0	0	0

Table 4.1: Boundary matrix $\mathbf{B_0}$ for ∂_0 of complex in Figure 4.4a

∂_1	[a,b]	[a,c]	[a,d]	[b,c]	[b,d]	[d,e]
[a]	1	1	1	0	0	0
[b]	1	0	0	1	1	0
[c]	0	1	0	1	0	0
[d]	0	0	1	0	1	1
[e]	0	0	0	0	0	1

Table 4.2: Boundary matrix $\mathbf{B_1}$ for ∂_1 of complex in Figure 4.4a

∂_2	[a,b,c]
[a,b]	1
[a,c]	1
[b,c]	1
[b,d]	0
[d,e]	0

Table 4.3: Boundary matrix $\mathbf{B_2}$ for ∂_2 of complex in Figure 4.4a

In order to compute rank(Z_j) and rank(B_j) and the number of k-holes in the complex (the Betti number β_k), we must first reduce the matrices to a canonical form known as the *Smith normal form* (SNF).

Definition 4.4.1. Smith Normal Form: A matrix $\mathbf{M} \in \mathbb{Z}_2^{n\times m}$ is in *Smith normal form* if it is diagonal and if it can be obtained by multiplying $\mathbf{M}$ by invertible matrices $\mathbf{S} \in \mathbb{Z}_2^{n\times n}$ and $\mathbf{T} \in \mathbb{Z}_2^{m\times m}$ as $\mathbf{M_{SNF}} = \mathbf{SMT}$.

The reduced matrix $\mathbf{B_{1_{SNF}}}$ is shown in Table 4.4 and $\mathbf{B_{2_{SNF}}}$ is shown in Table 4.5. The ∂_0 matrix cannot be further reduced and thus it is not shown. The SNF matrices contain all the information required to find rank(Z_k) and rank(B_k) for $k \in \{1, 2\}$:

$$\text{rank}(Z_k) = m - \text{rank}(\mathbf{B}_{(\mathbf{k})_{\mathbf{SNF}}}) \tag{4.18a}$$

$$\text{rank}(B_k) = \text{rank}(\mathbf{B}_{(\mathbf{k+1})_{\mathbf{SNF}}}) \tag{4.18b}$$

These simple calculations show us that $\text{rank}(Z_1) = 2$ and $\text{rank}(B_1) = 1$. From this we can see that the number of 1-holes in our complex is $\beta_1 = \text{rank}(Z_1) - \text{rank}(B_1) = 1$, as expected.

∂_1	[a,b]	[a,c]	[a,d]	[b,c]	[b,d]	[d,e]
[a]	1	0	0	0	0	0
[b]	0	1	0	0	0	0
[c]	0	0	1	0	0	0
[d]	0	0	0	0	0	1
[e]	0	0	0	0	0	0

Table 4.4: Reduced boundary matrix $\mathbf{B}_{\mathbf{1}_{\mathbf{SNF}}}$ for ∂_1 of the simplicial complex in Figure 4.4a. We note that the matrix is not diagonal in order to retain lexicoraphical ordering, but can be easily made diagonal.

∂_2	[a,b,c]
[a,b]	1
[a,c]	0
[b,c]	0
[b,d]	0
[d,e]	0

Table 4.5: Reduced boundary matrix $\mathbf{B}_{\mathbf{2}_{\mathbf{SNF}}}$ for ∂_2 of the simplicial complex in Figure 4.4a.

4.5 Persistent Homology

Persistent homology is a methodology originally proposed by Edelsbrunner, Letscher, and Zomorodian and further developed by many others for extracting and quantifying topological information from data [230]. This methodology is discussed in detail in [231, 232, 233, 204, 234, 206]. The homology of data provides deep insight into the structure of the data and quantification capabilities of their geometric features [206, 234].

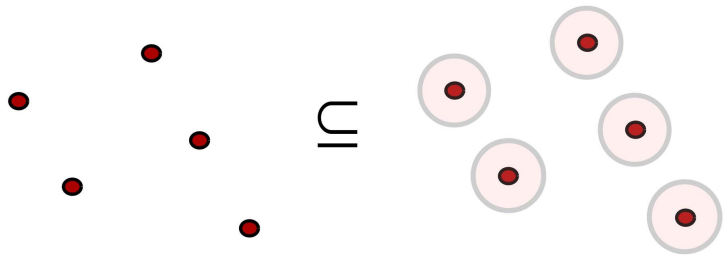


Figure 4.10: A cover of points $x_i \in \mathcal{X}$ is defined by a set of balls $B(x_i, \epsilon)$ expanded around each point.

4.5.1 *Building Simplicial Complexes from Data*

Direct computation of the homology of an arbitrary set defined by the space $\mathcal{X} \subset \mathbb{R}^n$ is a complex task. To simplify this task, we leverage simplicial homology; here, we identify a simplicial complex $\mathcal{K}$ such that its homology is the same or similar to that of $\mathcal{X}$. We can define such a complex $\mathcal{K}$ by creating a geometric object known as the *cover* ($\mathcal{U}$) of $\mathcal{X}$.

Definition 4.5.1. Cover: $\mathcal{U} = \{U_i\}_{i \in I}$ is a cover of a metric space $\mathcal{X}$ if $\mathcal{X} \subseteq \bigcup_{i \in I} U_i$.

If we define our metric space ($\mathcal{X}$) to be a finite set of points in $\mathbb{R}^n$, then we can imagine each set U_i to be a ball $\{B(x_i, \epsilon) : x_i \in \mathcal{X}, \epsilon \in \mathbb{R}^+\}$ centered around each point $x_i \in \mathcal{X}$, where ϵ is the ball's radius. An example of such a cover is presented in Figure 4.10. We now utilize this cover to develop a simplicial complex, known as a *Čech complex*.

Definition 4.5.2. Čech Complex: The Čech Complex (Č) is a simplicial complex built from k-simplicies that are the non-empty intersection of k+1 sets of a cover $\mathcal{U}$.

A Čech complex is also known as the "nerve" of the cover $\mathcal{U}$.

Definition 4.5.3. Nerve: The nerve of collection $\mathcal{U} = \cup_{i \in I}\{U_i\}$ is the simplicial complex with vertices I and k-simplices built from $\{i_0, i_1, ..., i_k\}$ if and only if $U_{i_0} \cap U_{i_1} \cap ... \cap U_{i_k} \neq \varnothing$.

By using the so-called "Nerve Theorem" construction, we can build a simplicial complex $\mathcal{K}$ for space $\mathcal{X} = \cup_{i \in I}\{U_i\}$. Under certain assumptions, the simplicial complex $\mathcal{K}$ and

the space $\mathcal{X}$ are homotopy equivalent [235, 236]. With this, we can apply the calculations and analysis of simplicial homology directly to our data. However, there is one caveat that is important to note, which is the selection of the distance ϵ. An example of the the nerve of a dataset with varying levels of ϵ is shown in Figure 4.11. We can see that, as we adjust ϵ of the cover $\mathcal{U}$, we obtain different Čech complexes, each with a different homology. We want to ensure that the homology captures the most interesting features of the data. In the complex shown in Figure 4.11, these features are the two clusters of points; here, one cluster forms a loop and the other does not. It is easy to see in this example what range of ϵ values would be most effective at capturing this information. However, if our dataset is of much higher dimension, finding the correct ϵ is much more difficult. Consequently, we characterize the dataset for multiple values of ϵ. This information is captured by a *filtered simplicial complex* [204].

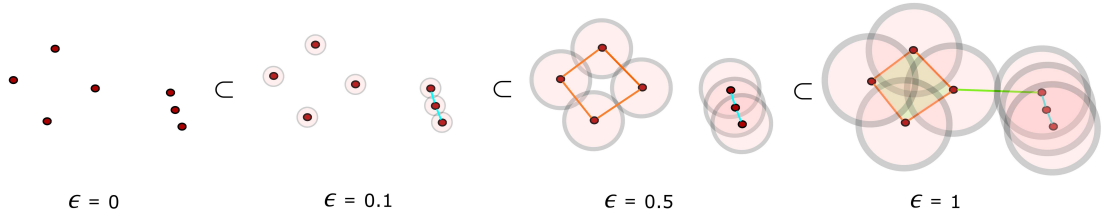


Figure 4.11: Filtration of points $x_i \in \mathcal{X}$ by a set of balls $B(x_i, \epsilon)$ with expanding ϵ. As ϵ is increased the topology of the Čech complex evolves and this introduces holes and higher dimensional simplices. This filtration builds a filtered simplicial complex ($\mathcal{K}_{\epsilon=0} \subset \mathcal{K}_{\epsilon=0.1} \subset \mathcal{K}_{\epsilon=0.5} \subset \mathcal{K}_{\epsilon=1}$).

Definition 4.5.4. Filtered Simplicial Complex: A filtered simplicial complex $\mathcal{K} \in \mathbb{R}^m$ is a simplicial complex for which there is a series of nested simplicial subcomplexes $\mathcal{K}_\epsilon \in \mathbb{R}^m$ such that:

$$\mathcal{K}_{\epsilon_0} \subset \mathcal{K}_{\epsilon_1} \subset ... \subset \mathcal{K}_{\epsilon_n} \tag{4.19}$$

where $\epsilon_0 < \epsilon_1 < \cdots < \epsilon_n$.

Referring back to the example in which we define the cover $\mathcal{U}$ as a set of balls $B(x_i, \epsilon)$ of radius ϵ, we can view the filtered complex as the set of nerve complexes that are formed

as we expand ϵ. Figure 4.11 demonstrates that the filtration of nerve complexes $\mathcal{K}_{\epsilon=0.1} \subset \mathcal{K}_{\epsilon=0.3} \subset \mathcal{K}_{\epsilon=0.5} \subset \mathcal{K}_{\epsilon=1}$. Note that the interesting features of the data (the two clusters and one loop) are present in the homology of the subcomplexes and persist during a large portion of the filtration. The main goal of this analysis is to identify *persistence intervals* in the complex filtered by ϵ. Given a topological feature present in a filtration, we identify the value of ϵ where the feature is *born* (appears) and *dies* (disappears).

Definition 4.5.5. Birth: For a filtered complex $\mathcal{K}$ and subcomplexes $\mathcal{K}_i, \mathcal{K}_j$ where $i < j$. A topological feature $x \in H_p(\mathcal{K}_j)$ is *born* at j if $x \notin H_p(\mathcal{K}_i)$.

Definition 4.5.6. Death: For a filtered complex $\mathcal{K}$ and subcomplexes $\mathcal{K}_i, \mathcal{K}_j$ where $i < j$. A topological feature $x \in H_p(\mathcal{K}_i)$ *dies* at j if $x \notin H_p(\mathcal{K}_j)$. A feature will also die if the feature merges with a feature born earlier in the filtration, this is known as the *elder rule*.

Definition 4.5.7. Persistence Interval: For a given topological feature x with birth point i and death point j, the *persistence interval* (Int) for the feature is given by:

$$\text{Int} = [i, j) : i, j \in \bar{\mathbb{R}} \tag{4.20}$$

If $j = \infty$ then the component does not die during the filtration (persists forever).

With a filtration we are identifying the appearance and disappearance of topologically interesting features in our dataset. The filtered complex in Figure 4.11 demonstrates this concept. We can see that the hole in the dataset is born at $\epsilon = 0.5$ and is completely filled in at $\epsilon = 1$, persisting for a majority of the filtration. We can also see that the individual points (at $\epsilon = 0$) become two connected components at $\epsilon = 0.5$ and then become a single component a $\epsilon = 1$. Thus, the longest persistence intervals in both H_1 and H_0 capture the defining topological characteristics of the filtered complex.

4.5.2 *Persistence Diagrams*

The information about topological features contained within a filtered complex is summarized into what is known as a *persistence diagram* (PD) [230]. This can be computed via an extension of the matrix methods presented in Section 4.4 [205]. The PD is a visual method that represents the *birth* (x) and *death* (y) of topological features as a set of points in $\mathbb{R}^2$. The persistence diagram associated with the filtration in Figure 4.11 is shown in Figure 4.12. This diagram represents the *birth* and *death* of the features of the H_0 and H_1 homology groups for the filtered complex. The *persistence interval* associated with each feature is the vertical line segment between the persitence point and the diagonal. This information allows for a direct visual understanding of the topology of the dataset.

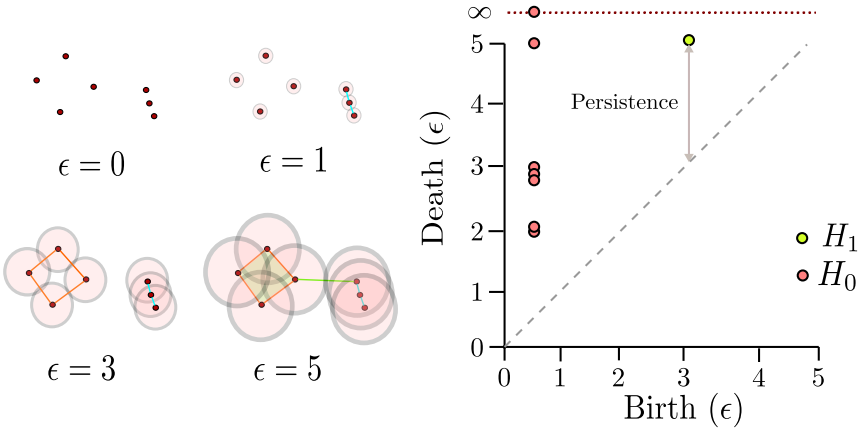


Figure 4.12: Filtration of points $x_i \in \mathcal{X}$ by a set of balls $B(x_i, \epsilon)$ with expanding ϵ and its corresponding persistence diagram. The PD records the ϵ value at which topological features are born and the ϵ value of their death during the filtration. For example, the cycle born at $\epsilon = 3$ ($x = 3$) dies at $\epsilon = 5$ ($y = 5$) when it is filled in, with a total persistence of $5 - 3 = 2$, which is seen in the PD.

An important property of PDs is that they can be *vectorized* to enable quantification and these vectors can be used to perform tasks such as regression, classification, or clustering. For instance, one can apply PCA to the vectorized PDs to identify clusters defined by different topological features.

Definition 4.5.8. Vectorization: A persistence diagram PD_{X_i} of a dataset X_i is vectorized by mapping the PD_{X_i} to a vector $\overrightarrow{PD}_{X_i} \in \mathbb{R}^q$.

There are multiple ways to vectorize a PD, the most widely used mappings are the *persistence landscape* [210, 208] and the *persistence image* [212] (we will focus on the latter). The *persistence image* is a smoothed representation of the points (x, y) in a persistence diagram $(x, y) \in PD$. Typically, the smoothing is done by applying a Gaussian kernel (with mean u and variance σ^2) to each of the points $\{x, y\} \in PD$:

$$\phi(x, y) = \frac{1}{2\pi\sigma^2} \exp - \frac{(x - u_x)^2 + (y - u_y)^2}{\sigma^2} \tag{4.21}$$

Definition 4.5.9. Persistence Surface: The *persistence surface* is a scalar mapping $\rho : \mathbb{R}^2 \to \mathbb{R}$ with weighting function $w : \mathbb{R}^2 \to \mathbb{R}$ and is defined as:

$$\rho(PD) = \sum_{u \in PD} w(u)\phi(u) \tag{4.22}$$

A *persistence image* is the discretization of the *persistence surface*.

Definition 4.5.10. Persistence Image: For a given PD, a *persistence image* of size n by m is a collection of pixels supported in a rectangle $R = [a, b] \times [c, d]$. The (i, j) pixel $p_{i,j}$, is the area $[a + \frac{i-1}{n}(b - a), a + \frac{i}{n}(b - a)] \times [c + \frac{j-1}{m}(d - c), c + \frac{j}{m}(d - c)]$.

$$\overrightarrow{\rho}[i, j] = \iint_{p_{i,j}} \rho dx dy \tag{4.23}$$

The ultimate goal of vectorization methods is to create a *stable* representation of the PD.

Definition 4.5.11. Stability: A vector representation $\overrightarrow{PD}_{X_i}$ of a persistence diagram PD_{X_i} is said to be *stable* if small perturbations in PD_{X_i} (represented as PD'_{X_i}) results in a bounded change in $\overrightarrow{PD}'_{X_i}$.

Mathematically, we establish the stability property as:

$$\text{dist}(\overrightarrow{PD}_{X_i}, \overrightarrow{PD}'_{X_i}) \leq L \cdot \text{dist}(PD_{X_i}, PD'_{X_i}) \tag{4.24}$$

where $dist(\cdot, \cdot)$ represents a distance metric and L represents a scalar constant. The distance between PDs is commonly measured using the *Wasserstein distance or the bottleneck distance. Whereas the distance between the vectorized PDs ($\overrightarrow{PD}$) can be expressed in terms of l_p norms.*

Definition 4.5.12. Wasserstein distance: The p^{th}-*Wasserstein distance* between persistence diagrams PD_1 and PD_2 is defined as:

$$d_{W_p}(PD_1, PD_2) = \left(\inf_{\gamma} \sum_{x \in PD_1} \|x - \gamma(x)\|_{\infty}^{p} \right)^{1/p} \tag{4.25}$$

where γ ranges over all possible bijections from PD_1 to PD_2. By convention, we add an infinite number of points in the diagonal to allow for bijections between PD's containing difference numbers of points.

Definition 4.5.13. Bottleneck distance: The *bottleneck distance* between persistence diagrams, PD_1 and PD_2, is defined as:

$$d_B(PD_1, PD_2) = \inf_{\gamma} \sup_{x} \|x - \gamma(x)\|_{\infty} \tag{4.26}$$

where γ ranges over all possible bijections from PD_1 to PD_2 with the same considerations made in the Wasserstein distance definition.

Notably, it has been proven that *persistence landscapes* and *persistence images* are stable under the appropriate distances [210, 208, 212]. Intuitively, stability indicates that topology changes in a *continuous* manner under perturbations. This makes them excellent representations of PD and amenable to use in diverse tasks such as regression and classification.

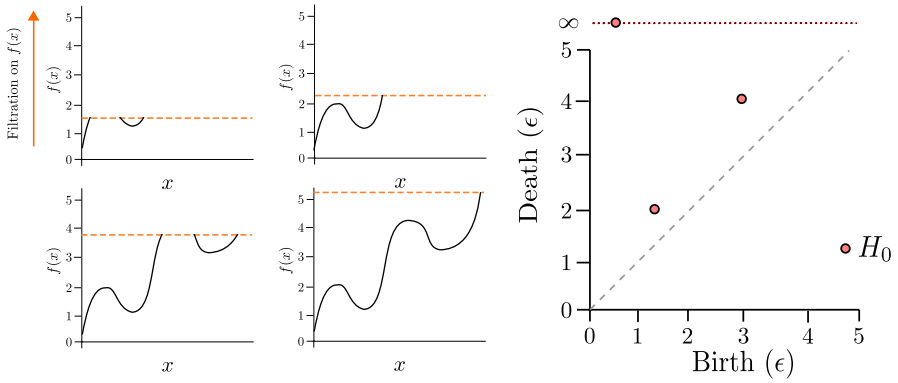


Figure 4.13: Morse filtration of a function $f : \mathbb{R} \to \mathbb{R}$. Consider a sublevel set $f^{-1}(-\infty, b)$ for increasing values of b. The topology of these sublevel sets changes at the critical points of f. As b increases the value of the sublevel set, the persistence diagram records the topological changes in the function.

4.5.3 *Topology of Continuous Functions*

In the previous sections we focus on understanding the topology of datasets that are made of point clouds. We now discuss how to quantify the topology of continuous functions. An example of this type of object is the scalar function shown Figure 4.13. Topologically, the interesting features of this function are its critical points (min and max points). In order to characterize these critical points we utilize a new form of filtration known as a *Morse filtration* or *sublevel set* filtration [237]. The Morse filtration is derived from ideas of Morse theory, which is the study of the topology of manifolds through differential functions and the analysis of the critical points of these functions [238]. Here, we consider the graph of a continuous function as a differentiable manifold [123].

Definition 4.5.14. Level Set: Given a differentiable manifold M and function $f : M \to \mathbb{R}$, the *level set* M^a at a point a is defined as the pre-image:

$$M^a = f^{-1}(a) = \{x \in M : f(x) = a\}. \tag{4.27}$$

The level set contains all points of the manifold that have the same function value. In

order to create a filtration, we use a *sublevel set* defined below.

Definition 4.5.15. Sublevel Set: A *sublevel set* $M^{(a,b)}$, where $a, b \in \bar{R}$ and $a = -\infty$ is defined as:

$$M^{(-\infty,b)} = \{x \in M : f(x) \leq b\}. \tag{4.28}$$

As we pass through the Morse filtration and build the sublevel set of the function, we are creating a well-defined filtered complex. The topology of the function will change as the filtration passes through critical points in the manifold [237]. These topological changes are quantified in a persistence diagram which is subsequently vectorized for analysis. An example of this type of filtration and the corresponding persistence diagram are shown in Figure 4.13. The Morse filtration is a good choice for functions that are continuous or that can be approximated as piece-wise linear functions (e.g., a time series). The method can be expanded to k-dimensional functions [239]. This makes it a powerful approach to characterize complex surfaces (landscapes) that have many minima/maxima. We demonstrate this technique on 2D and 3D functions.

Stability of Persistence Diagrams for Functions

Persistence diagrams of real valued functions are also *stable* representations of data. The following Theorem 1, established by Cohen-Steiner and co-workers, highlights this result [234].

Theorem 1. *Given real valued functions f, g with finitely many critical points and their corresponding persistence diagrams PD_f, PD_g, we have that:*

$$d_B(PD_f, PD_g) \leq \|f - g\|_\infty \tag{4.29}$$

where $\|\cdot\|_\infty$ represents the l_∞ distance between two functions $f, g : X \to \mathbb{R}$:

$$\|f - g\|_\infty = \sup\{\|f(x) - g(x)\| : x \in X\}. \tag{4.30}$$

An illustration of the stability of persistence diagrams is shown in Figure 4.14. Here, we can see that the persistence diagram of the two functions are similar. The presence of the strong critical points, with large persistence, are well captured in both diagrams. We can also see that the persistence diagram captures the structure of the weak critical points (arising from noise), which have short persistence.

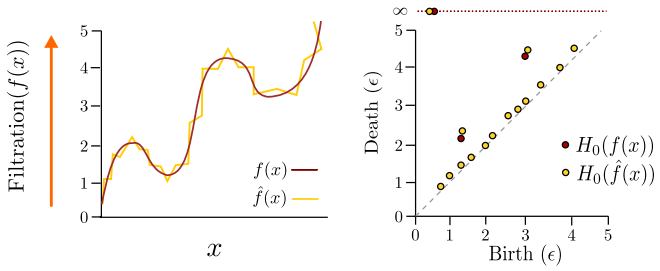


Figure 4.14: Filtration of functions $f(x)$ and $\hat{f}(x)$. The strong critical points of both functions are captured in the persistence diagram analysis while the weak critical points (arising from noise) remain close to the diagonal as they have minimal persistence.

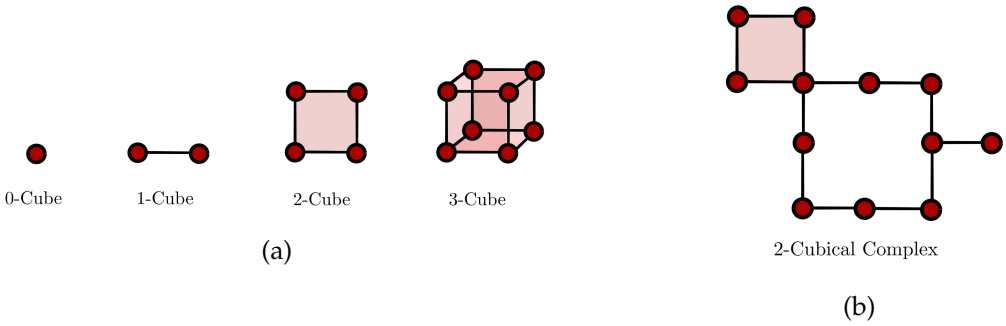


Figure 4.15: (a) Representations of the elementary k-cubes of dimension 0 to dimension 3. (b) A 2-cubical complex that contains a 2-cube, along with a hole created by 1-cubes.

4.5.4 *Cubical Complexes and Images*

We briefly discuss cubical complexes and cubical homology as these are important in understanding data over rectangular domains such as images (represented by pixels in 2D or voxels in 3D) and are used primarily in Morse filtrations [240, 241]. Cubical homology

is similar to simplicial homology but uses different basis shapes. This highlights the fact that one can choose different basis shapes in topological analysis. The basis shapes for cubical complexes are k-cubes (hypercubes) or elementary cubes built from sides of equal length. In Figure A.1 we show k-cubes of dimension 0 through 3. An example cubical complex is found in Figure 4.15b.

Filtered cubical complexes can be developed from data and we can perform persistence homology calculations on them, similar to filtered simplicial complexes. An important application of cubical analysis is the analysis of images. An image can be viewed as a 2D surface embedded in three dimensions where two dimensions are the coordinates of each pixel, and the third dimension is the scalar value associated with each pixel (e.g., intensity). We can also view an image as an approximation of a continuous object over which a Morse filtration can be performed. We demonstrate the concept of filtration on the image in Figure 4.16a. We filter through the level sets of this image and develop the filtered complex through the sublevel set. This filtration is demonstrated in Figure 4.16b, where $\mathcal{C}_{f\leq 1} \subseteq \mathcal{C}_{f\leq 3} \subseteq \mathcal{C}_{f\leq 5}$. The persistence diagram is then represented in Figure 4.17. The PD is able to capture the dominant topological features of our image such as the presence of the two critical points in the image and the fact that there is only a single connected component. The persistence diagram generated from a cubical complex filtration and a simplicial complex filtration have the same properties and can be vectorized in the same way. Thus, we can extend our analysis from discrete point data to that of images or other high-dimensional continuous objects.

4.5.5 *Inverse Analysis*

The algorithm to compute persistence homology allows us, for each point in a persistence diagram, to identify representative features in the filtration. The intuition behind this idea is presented in Figure 4.18. However, before we discuss this procedure along with an appropriate post-processing step, a few issues need to be clarified. The persistence

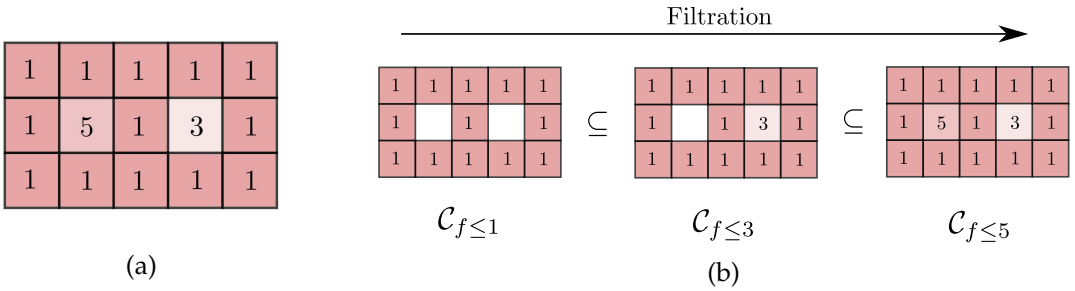


Figure 4.16: (a) Representation of an image; the values within each pixel represent intensity. (b) The representative filtration on the image itself and the corresponding sublevel sets.

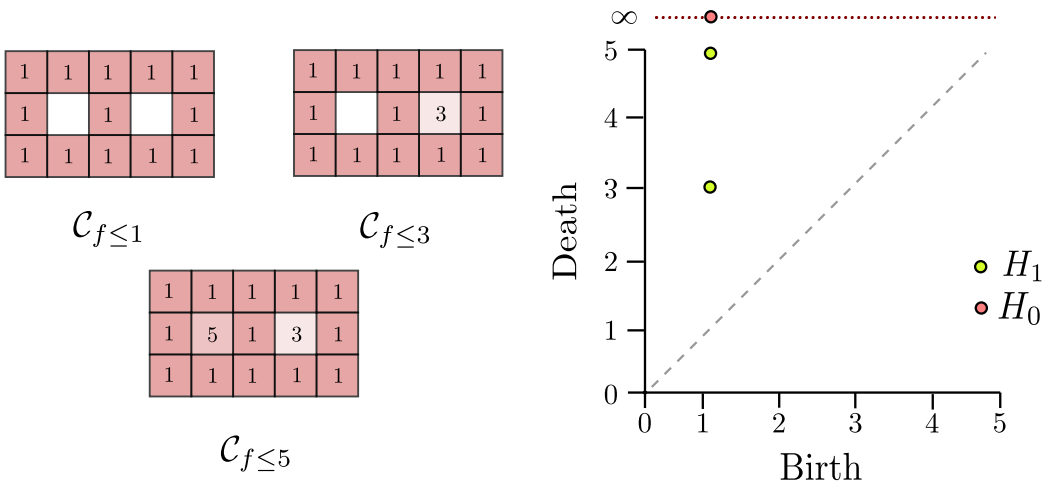


Figure 4.17: Persistence diagram for the filtration of the image in Figure 4.16a. The PD reveals the structure of the image,which changes at the critical points of the image (i.e., $f = 3$ and $f = 5$.)

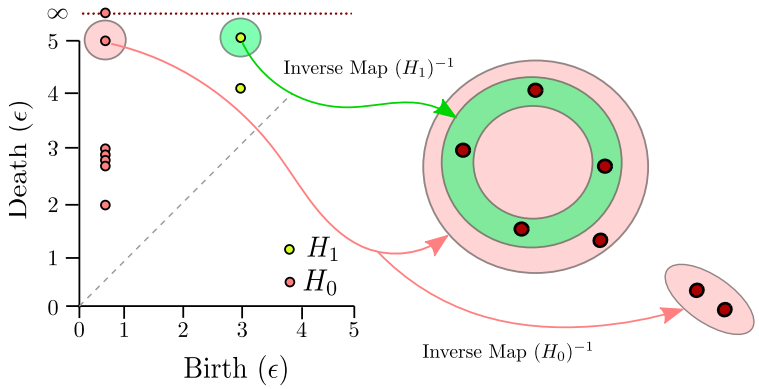


Figure 4.18: Inverse mapping of the features of persistence diagram to the features dataset in Figure 4.12. The features of interest are picked up within the persistence diagram; In this case they are the highly persistent hole and connected component. The inverse mapping for the selected point in one dimensional persistence identifies the dominant loop in the dataset and the inverse mapping for the selected point in zero dimensional persistence identifies the two most distinct clusters of the data.

algorithm will provide a set of cycles that *generate* a persistent homology group. However, it will not necessarily be the set that is close to what we may call *geometrically optimal generators*. A similar problem can occur when identifying an optimal basis for a vector space. For example, let us consider the $\mathbb{R}^2$ space and various possible bases for this space presented below.

$$B_1 = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}, B_2 = \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 10 \end{bmatrix} \right\}, B_3 = \left\{ \begin{bmatrix} 123 \\ 21 \end{bmatrix}, \begin{bmatrix} 2121 \\ 1243 \end{bmatrix} \right\}.$$

B_1, B_2 and B_3 are all valid basis of $\mathbb{R}^2$ and they can be mapped to each other via multiplication by a non-singular matrix. Yet, only B_1 can be called *geometrically optimal* as it the natural representation of the space.

We now illustrate, with a simple example, how this issue manifests itself when selecting a basis for a persistent homology group. The cubical complex presented in Figure 4.19 contains two holes (h_1, h_2). Cycles g_1 and g_2 surrounding h_1 and h_2 are generators (they form a basis) for the first homology group of this shape. They are the optimal geometric representation of these holes. Let $\mathbf{G_{b1}}$ be the basis made up of these two cycles. The

second basis $\mathbf{G_{b2}}$ contains cycles g_1' and g_2'. They are a perturbed version of cycles g_1 and g_2, as $g_1 \simeq g_1'$ and $g_2 \simeq g_2'$, but are not geometrically optimal.

Lastly, let us consider a basis $\mathbf{G_{b3}}$. It generates the first homology group of our shape, but it is a non-optimal basis for the holes because there is no unique correspondence between cycles and holes. $\mathbf{G_{b3}}$ and $\mathbf{G_{b1}}$ can be transformed into each other by multiplication of a nonsingular matrix (change of basis).

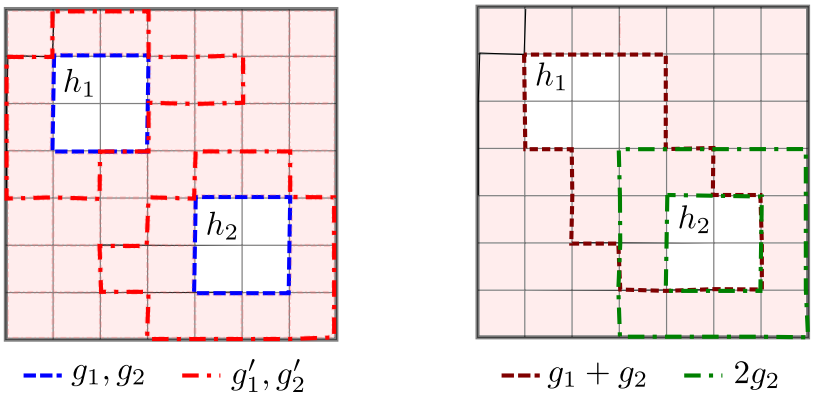


Figure 4.19: A representation of a cubical complex with two holes. (left) Cycles g_1 and g_2 are an optimal representation of holes (h_1, h_2) as they trace each hole. Cycles g_1' and g_2' also represent these holes, but are not geometrically optimal. (right) Another set of generators for the holes (h_1, h_2) that are not optimal as they represent linear combinations of the cycles g_1 and g_2.

$$\mathbf{G_{b1}} = \begin{bmatrix} g1 & 0 \\ 0 & g2 \end{bmatrix}, \mathbf{G_{b2}} = \begin{bmatrix} g1' & 0 \\ 0 & g2' \end{bmatrix}, \mathbf{G_{b3}} = \begin{bmatrix} g1 & 0 \\ g2 & 2g2 \end{bmatrix}$$

The persistent homology algorithm may return non-optimal bases, like $\mathbf{G_{b3}}$. There is not much that can be done to fix this issue in general. However, there is a way of finding the most optimal generators within their homology class. This will allow, for instance, to simplify $\mathbf{G_{b2}}$ into $\mathbf{G_{b1}}$. In order to address the problem of non-optimal generators, we construct an integer optimization problem to obtain the sparsest representation of a given generator. We begin this optimization with the representative cycle c obtained from the persistence algorithm and identify the sparsest chain of simplices from the homology class of c. For example, in Figure 4.18, we wish to identify the basis (generator) for

the persistent cycle identified in green. We construct a simplified representation for the dataset in Figure 4.18 where $\mathcal{K}_\epsilon$ represents the associated simplicial complexes with the specified level of filtration. Because we know the cycle is born at $\epsilon = 3$ and dies at $\epsilon = 5$ we only focus on the portion of the filtration up to the level $\epsilon = 5$. Given the simplices below that filtration level, we wish to identify the sparsest set of 1-simplices $z = \sum_i \sigma_i$, where $\sigma_i \in \mathcal{K}$, that is homologous to the generator c. We denote $||z||_0$ as the cardinality of z.

With this information, we construct an integer optimization formulation, found in 4.31, where we seek to identify the sparsest set of simplices $z \in \mathcal{K}_{\epsilon<5}$ that is homologous to the cycle c obtained from persistent homology computations. In Figure 4.20 we identify multiple cycles ($z_1, z_2 \in \mathcal{K}_{\epsilon<5}$), and note that $z_1 = z$ because $||z_1||_0 < ||z_2||_0$.

$$z = \text{argmin}\, ||z||_0, \text{ subject to } z \simeq c \tag{4.31}$$

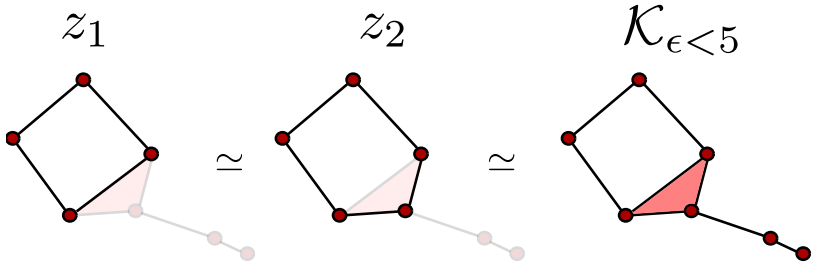


Figure 4.20: A representation of potential simplicial complexes that are homotopy equivalent ($\simeq$) to the filtration complex ($\mathcal{K}$). The complex z_1 contains the fewest number of 1-simplices, thus is the optimal representation of the cycle contained in $\mathcal{K}_{\epsilon<5}$.

Inverse analysis can be used to understand dominant features that drive classification and regression results. For instance, the defining characteristics of a given class within a dataset can be identified via the weights of regression/classification models in the space of PDs. These defining characteristics can then be mapped back to the original data set which can be used to gain physical understanding of differences between datasets. Applications of these methods in material science are found in [217, 216, 215]. Here, inverse analysis is used to identify fracture or degradation sites in materials and to identify pore

configurations in granular crystallization.

4.6 Case Studies

We now proceed to demonstrate how these methods can be applied to different types of datasets. To do so, we use a couple of illustrative examples and real datasets derived from soft materials and molecular dynamics simulations. All the calculations presented were conducted in `Python` [242] using the TDA package `GUDHI` [243]. All scripts and data needed to reproduce these results can be found in https://github.com/zavalab/ML/tree/master/TDApaper.

4.6.1 *Topology of Point Clouds*

We illustrate how to use TDA to analyze point clouds; specifically, we seek to extract topological features from the data to perform binary classification of point clouds. The clouds used here are collections of points in two dimensions x_1, x_2 (for visualization purposes). In actual applications, one can conduct analysis on a point cloud of any dimension. We represent each cloud as X_i where each cloud can belong to two different types of classes (Class 1 or Class 2). Our goal is to take each point cloud X_i as an input, project this data to their respective H_1 persistence diagram PD_{X_i} through an epsilon ball filtration (i.e., extract the topological features), vectorize the PDs ($\overrightarrow{PD}_{X_i} \in \mathbb{R}^q$), and perform classification of the cloud based on the vectorized topological features.

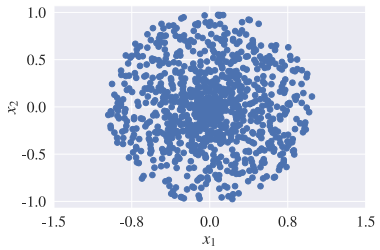
The point cloud classes are shown in Figure 4.21 and the H_1 PDs are found in Figure 4.22. Note how the point clouds of Class 1 define a simple object (ellipse) while those of Class 2 define a more complex object (overlapping ellipses). We can see that the persistence diagrams are visually distinct; specifically, point clouds of Class 2 have features that persist over a longer range of the filtration.

We utilize the persistence image method to vectorize the PDs. We apply principal component analysis (PCA) to the vectorized PDs to verify if there is a separation between

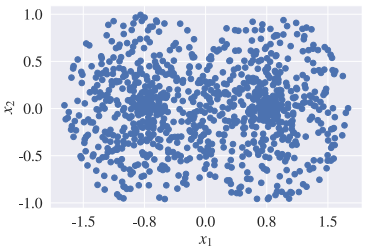
Class 1 and Class 2. We emphasize that the PCA projection is not applied to the original datasets X_i but to the transformed datasets $\overrightarrow{PD}_{X_i}$ obtained from TDA. The PCA projection on first and second principal components is shown in Figure 4.22c. This shows an obvious separation between Class 1 and Class 2, which means that the topological features extracted from the data (contained in H_1) are informative. This also suggests that a simple linear classifier using vectorized PDs as features should work well. To test this hypothesis, we apply a linear support vector machine (SVM) classifier using the $\overrightarrow{PD}_{X_i}$ as features; we find that we can *perfectly classify* the datasets (we use a 5-fold cross validation scheme). This again indicates that the topological features extracted with TDA are highly informative.

An advantage of utilizing a linear classifier is the ability to extract which features are the ones driving classification. Specifically, the magnitude of the weights of the SVM classifier $w \in \mathbb{R}^q$ can be directly associated with the importance of each feature of the vectorized PDs [54]. Weights with a large negative value are characteristic of point clouds in Class 1 and weights with a large positive value are characteristic of point clouds in Class 2. A visualization of the weights is shown in Figure 4.23a. From this representation, we can see that Class 1 is characterized by 1-D holes that are born and die in the early stages of the filtration suggesting a higher number of small radius holes. Class 2 is characterized by 1-D holes that persist over multiple stages of the filtration, suggesting the presence of holes with large radius. These results highlight how one can exploit topological information obtained with TDA to perform statistical (PCA) or machine learning tasks (SVM classification).

Work conducted in [244] has led to the development a set of techniques that are useful in the interpretation of PDs. These techniques allow for the identification of *volume-optimal* cycles, which are cycles that correspond to the sparsest representation of the topological features identified in a PD. This technique has been implemented in the `Homcloud` software, which can be used to identify the 1D holes that are responsible for differences between the classes. Inverse analysis identifies features of the data corresponding to areas

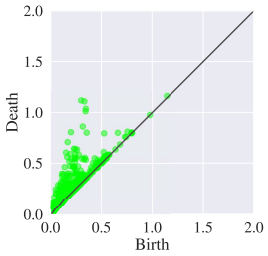


(a) Sample point cloud of Class 1.

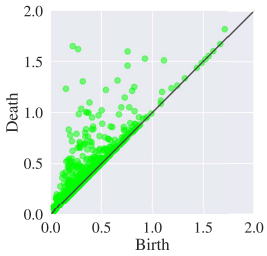


(b) Sample point cloud of Class 2.

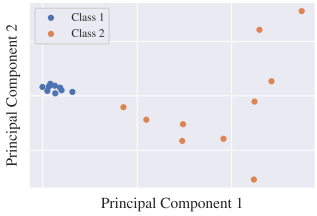
Figure 4.21: Types of point clouds analyzed using epsilon ball filtration.



(a)

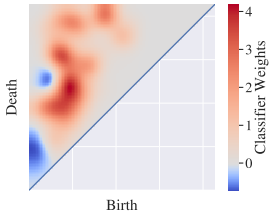


(b)

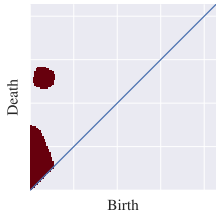


(c)

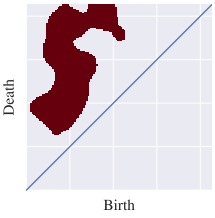
Figure 4.22: Persistence diagrams for Class 1 and Class 2 point clouds and the corresponding PCA analysis on the set of persistence images created from Class 1 and Class 2. (a) H_1 persistence diagram for Class 1. (b) H_1 persistence diagram for Class 2. (c) Principal components of persistence images for Class 1 and Class 2 datasets. It is clear that there is separation of the persistence diagrams.



(a)

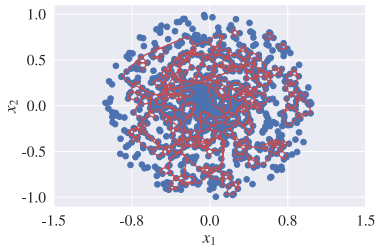


(b)

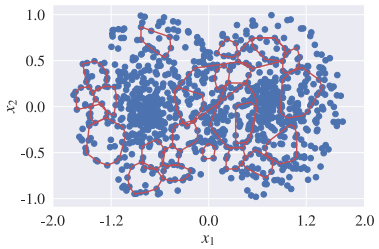


(c)

Figure 4.23: Masks highlight the areas of the PD that are important in distinguishing Class 1 from Class 2. We perform inverse analysis on these areas to visualize what features of the original data distinguishing classes. (a) Weights from SVM classification in the space of PDs. The areas of the diagram that distinguish Class 2 are in red and the areas of the diagram that distinguish Class 1 are in blue. (b) PD mask for Class 1. (c) PD mask for Class 2.



(a) Inverse H_1 analysis for Class 1.



(b) Inverse H_1 analysis for Class 2.

Figure 4.24: Inverse analysis based on classification weights for Class 1 and Class 2. The analysis reveals that the classifier is separating the classes based on the presence of large cycles in Class 2 and the larger number of smaller cycles in Class 1.

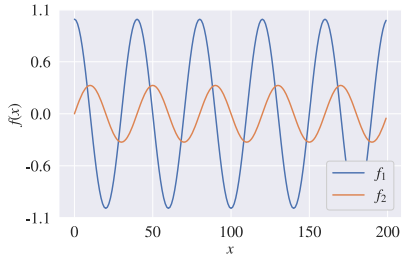
of the PD associated with the masks found in Figure 4.23. The inverse analysis for a sample of Class 1 and Class 2 is shown in Figure 4.24. For Class 2 we see larger separation between points and larger holes that persist for a longer period of the filtration. In Class 1 we see that the separation between points is smaller, resulting in smaller loops that are formed early and die quickly.

4.6.2 *Topology of Time-Series and Phase-Planes*

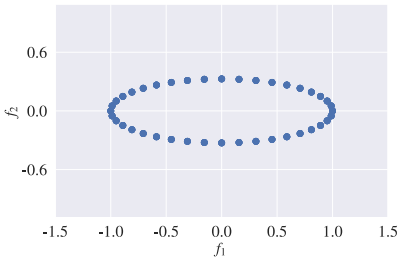
Persistence homology has seen applications in the area of time series analysis [218, 245, 246, 247, 248]. A simple example of the application of persistence homology is the analysis of the topology of phase-planes generated by a dynamical system. An example for two state variables f_1, f_2 is shown in Figure 4.25. The phase plane for functions f_1 and f_2 is created by plotting the two functions against each other (Figure 4.25b). The phase plane for this periodic system defines an ellipse, which is easy to characterize (e.g., in terms of its axes). We can add complexity to the topology of the phase plane by perturbing the dynamical system. For example, by adding a perturbation, we change the phase-plane to that shown in Figure 4.26. The topology of the new plane cannot be fully characterized using simple ellipsoids.

The analysis of the phase plane topology through an epsilon ball filtration allows us to differentiate the dynamics of the perturbed and unperturbed systems. We compare their

PDs in Figure 4.27 and 4.28; the unperturbed system contains a single highly persistent cycle while the perturbed system contains four cycles that are less persistent.

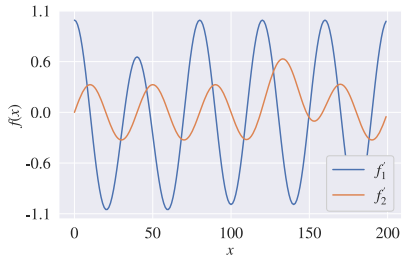


(a) Time series for functions f_1 and f_2.

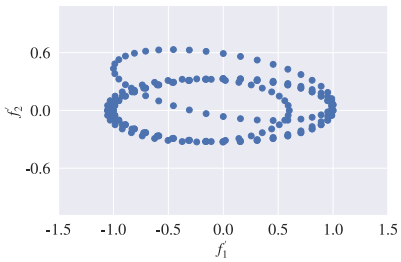


(b) Phase plane for functions f_1 and f_2

Figure 4.25: Phase plane for periodic orbit with two states. The plane is represented as cloud points from the edge of an ellipse and is ideal for a geometric analysis.

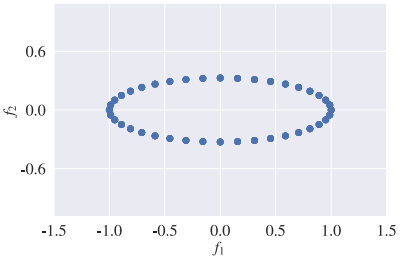


(a) Timer series for functions f_1' and f_2'.

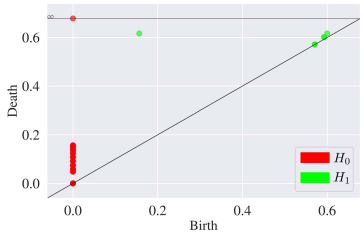


(b) Phase plane for functions f_1' and f_2'

Figure 4.26: Phase plane for perturbed periodic orbit with two states. The geometry of the plane can no longer be represented as an ellipse, but still represents a sampling from a more complex geometric object.



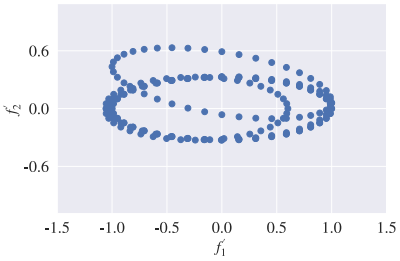
(a) Phase plane of functions f_1 and f_2



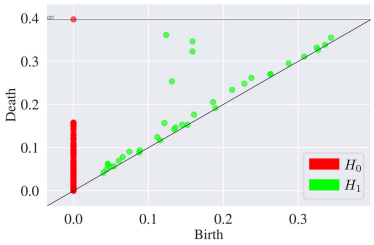
(b) Persistence Diagram of the phase plane.

Figure 4.27: PD for the phase plane reflects the presence of a single, persistent loop in the diagram. The persistence diagram captures the important geometric aspects of the data.

We can also use persistent homology within a *sliding windowing method* to detect when



(a) Phase plane of functions f_1' and f_2'



(b) Persistence diagram of the phase plane.

Figure 4.28: PD for the phase plane reflects the presence of four loops. The persistence diagram captures the important geometric aspects of the phase plane without having to fit a complex geometric model to the data.

a change has occurred in the dynamics of a system (e.g., for fault detection). This concept is illustrated in Figure 4.29 where we only show 4 windows for illustrative purposes; here we can see that the first, second, and fourth windows have similar topology while the third window has a different topology. We compare the PDs of each window in Figure 4.29; the PDs clearly reveal that the phase plane of the third window is different. This example demonstrates that the geometry of time series are highly informative and can capture the behavior of the system with minimal information. The small windows used in this example immediately demonstrate the cyclic/periodic nature of the time series and show a large difference in shape when a perturbation occurs. Many current methods require a large amount of information to model the given system via statistics or machine learning methods [249, 250, 251].

Another common perturbation of a system is *noise*; an important observation here is that noise usually has the effect of introducing local effects to a trajectory but does not distort the overall topology of the trajectory. This is illustrated in Figure 4.30, where we can see that the ellipse shape of the phase plane is retained. In other words, the topology of the phase plane is stable in the presence of noisy perturbations. In a real system, one may wish to characterize whether noise-free and noisy systems exhibit similar dynamics. One way to do this is to perform persistent homology calculations on both phase planes and to compare the resulting PDs. We show the results of this analysis in Figure 4.31; the

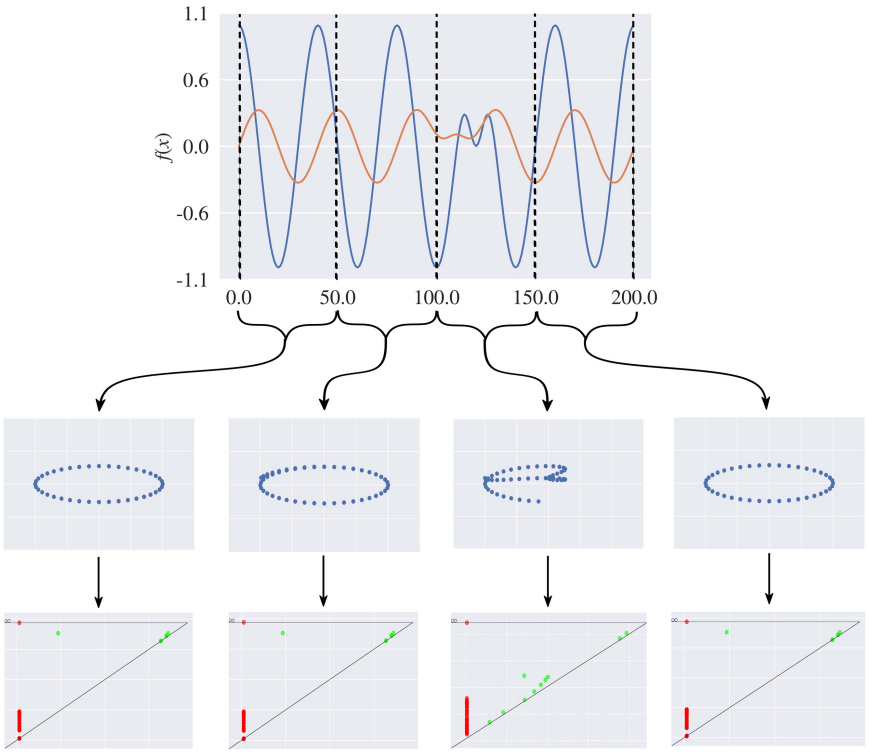
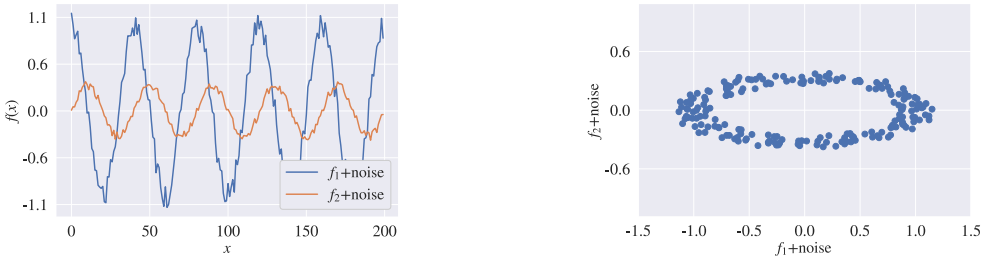


Figure 4.29: Sliding window method applied to time series. The first, second, and fourth windows have phase planes with similar topologies while the third window has an obvious shift (which introduces a change in the topology of its plane). This change in topology is captured by the persistence diagrams.

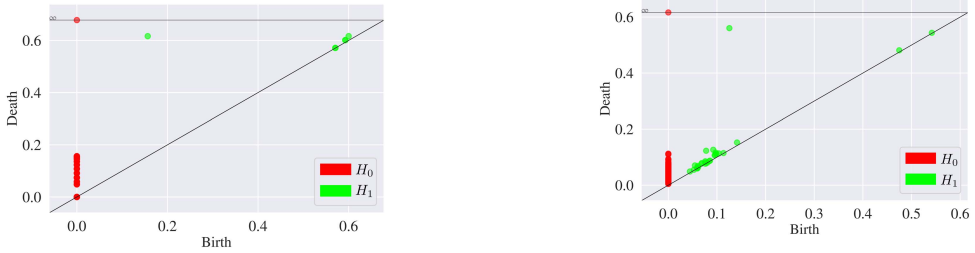
PDs reveal that both phase planes contain a persistent cycle (which indicates that they have phase planes with similar topologies). Features created by noise are shown at the bottom of the diagonal (these features have short persistence). These results demonstrate the stability of PDs [252], which is an important concept in TDA. These results also highlight that TDA is a powerful tool for the classification of time series [245], identification of periodic orbits and shifts [219], and for change point detection [253]. The paper of Perea [247] provides an excellent overview of TDA in signal processing.



(a) Time series with added Gaussian noise.

(b) Phase plane representation.

Figure 4.30: Phase plane for noisy f_1 and f_2 shows a similar topology to the noiseless counterparts.



(a) PD for functions f_1 and f_2

(b) PD for functions f_1 and f_2 with added noise

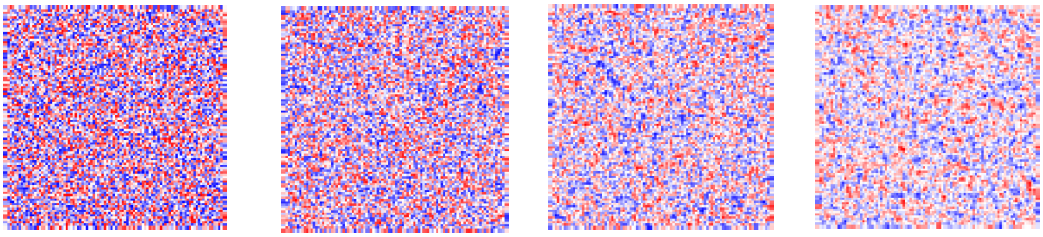
Figure 4.31: Persistence diagrams for noisy and noiseless functions. Note that the dominant feature (cycle) persists.

4.6.3 *Topology of 2D Scalar Fields*

In this example we use TDA to analyze the topology of 2D scalar fields over a discrete n by n domain U. We generate data by applying propagating a random field $\{u_{i,j}$:

$u_{i,j} \in \mathcal{U}(0,1)\}$ using the dynamic 2D diffusion equation (4.32) and we obtain the final steady-state field. We generate fields with different textures by using different diffusion coefficients D and independent, random initializations. The resulting fields are used as the datasets; illustrative examples for different diffusion coefficients are shown in Figure 4.32. Here, the blue color are points of small intensity (small values of $u_{i,j}$) while the red color are points of high intensity. We see that small coefficients generate textures that are more granular while large coefficients generate smoother textures. Our goal is to characterize the geometry of the fields to investigate if their underlying structure can be correlated to the diffusion coefficient. In our analysis, we represent the scalar field as a function (in 3D), as shown in Figure 4.33. This functional representation of the field reveals that low and high intensity points define critical points of the function. This also reveals that the field has complex topological features (many critical points with no obvious patterns are present).

$$\frac{u_{i,j}^{(n+1)} - u_{i,j}^{(n)}}{\Delta t} = D \left[\frac{u_{i+1,j}^{(n)} - 2u_{i,j}^{(n)} + u_{i-1,j}^{(n)}}{(\Delta x)^2} + \frac{u_{i,j+1}^{(n)} - 2u_{i,j}^{(n)} + u_{i,j-1}^{(n)}}{(\Delta y)^2} \right] \tag{4.32}$$



$D = 0.1$ $D = 0.3$ $D = 0.5$ $D = 0.6$

Figure 4.32: Samples from 2D scalar fields with their corresponding diffusion coefficient (D) value, where red represents high intensity values and blue represents small intensity values.

To illustrate the benefits of using TDA against other techniques, we investigate whether the structure of the dataset can be revealed by direct application of PCA [254] and diffusion maps [71]. This is a simple, naive comparison but will demonstrate that the datasets have nontrivial structure. The projection of the data onto the first two principal compo-

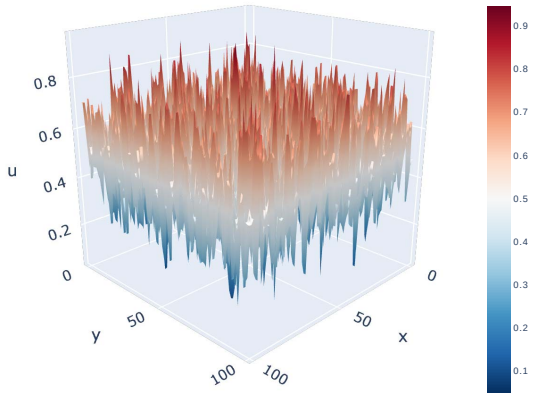
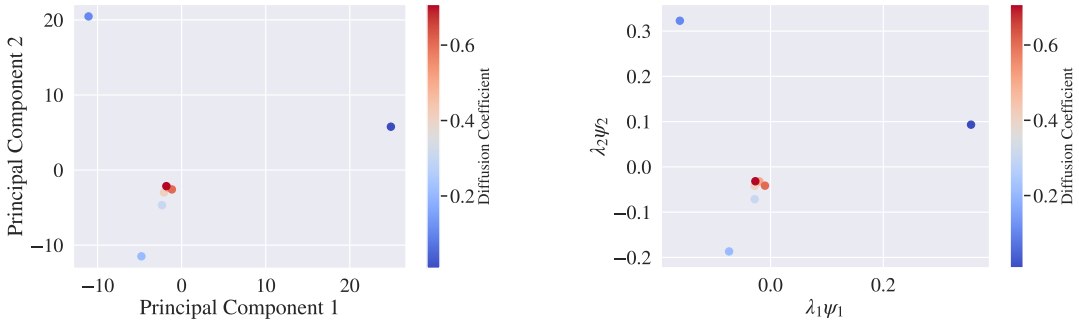


Figure 4.33: 3D-functional representation of a scatter field with diffusion coefficient $D = 0.6$. The function is treated as a cubical complex and the filtration is performed over the scalar value.

nents is shown in Figure 4.34a. Here, we highlight the points based on the associated diffusion coefficient. We also apply the diffusion maps, which is a nonlinear dimensionality reduction technique, and we obtain similar results (see Figure 4.34b). From these results we see that the features extracted by PCA and diffusion maps do not correlate to the diffusion coefficient.



(a) Dominant principal components from PCA. (b) Dominant features from diffusion map.

Figure 4.34: Dimensionality reduction for the 2D dimensional scalar fields using PCA and diffusion maps.

We now apply TDA to the 3D field functions and extract persistence diagrams. Example persistence diagrams for H_0 and H_1 are shown in Figure 4.35. We see that there is

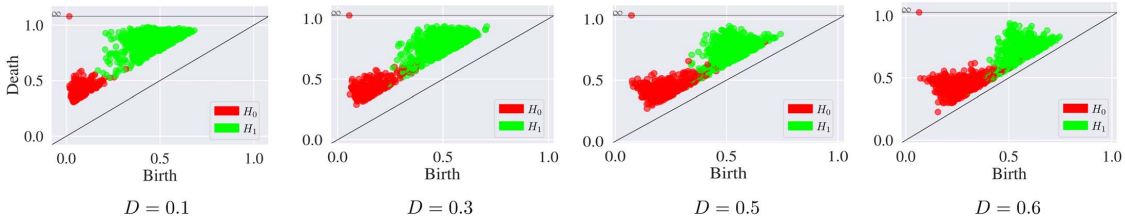


Figure 4.35: Evolution of PDs with the diffusion coefficient. A dependence of the topology with the diffusion coefficient emerges.

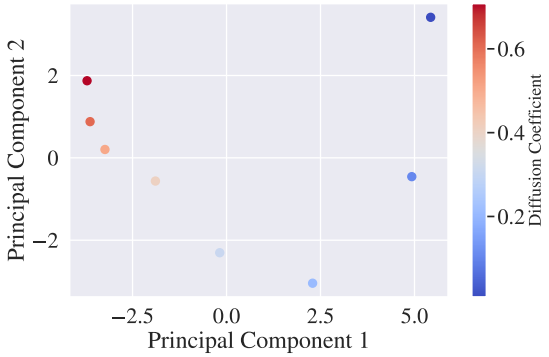


Figure 4.36: PCA performed on the PD for two-dimensional scalar fields. This reveals that the geometry of the dataset is directly related to the diffusion coefficient.

a visual shift in the PDs as we increase the diffusion coefficient. This seems to indicate that the PDs vary continuously with the diffusion coefficient. To verify this, we vectorize the PDs and apply PCA to the vectorized diagrams. The projection of the vectorized PDs onto the dominant principal components is shown in Figure 4.36. It is clear that there is a continuous dependence of the PD on the diffusion coefficient (it forms a continuous manifold). This result provides another demonstration of the stability of persistence diagrams and on how topology varies continuously under perturbations. Specifically, stability indicates that small changes in a given function (f, g) results in bounded changes in the associated persistence diagrams (PD_f, PD_g). Thus, because our perturbations to each function are based on changes in D, we can guarantee that the distance between PDs is bounded by the size in the perturbation in the diffusion coefficient (i.e., the distance is not arbitrary).

4.6.4 *Topology of Images*

We now illustrate how to use TDA to analyze images. Specifically, we analyze the optical response of liquid crystal sensors to air contaminants, in particular dimethyl methylphosphonate (DMMP) [54, 255]. We analyze the spatial response of a sensor in the presence of DMMP and in the presence of humid nitrogen (nitrogen + water), which we will refer to as water. The sensor responds to both DMMP and water, but there are subtle spatial differences in the optical response of the sensor to these different environments. The working principle of these sensors relies on a change in orientation of liquid crystal molecules in a film when exposed to an air contaminant (analyte). The change in orientation results in optical fields with different spatial and color features (see Figure 4.37). We use TDA to investigate if the topological features of these patterns present a dependence on the air environment. Such information can be used to design sensors (i.e., we can calibrate the sensor by correlating the optical response to the presence of DMMP).

Each optical micrograph is an image, taken from the endpoint of the sensor response,

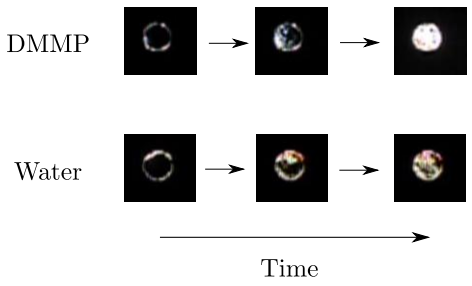
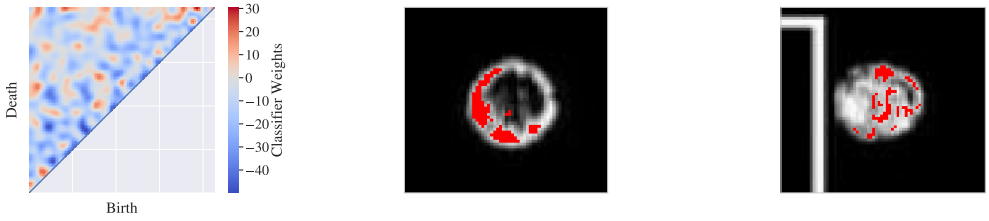


Figure 4.37: Optical patterns for a liquid crystal sensor when exposed to DMMP or water.



(a) (b) (c)

Figure 4.38: (a) Areas corresponding to optimal weights from linear SVM classification. The areas of the PD that distinguish DMMP are shown in red and the ones that distinguish water are shown in blue. Inverse analysis based on SVM weights for (b) DMMP and (c) water responses. Note that the camera artifact in (c) has no highlighted areas, demonstrating that the extracted features are physically relevant.

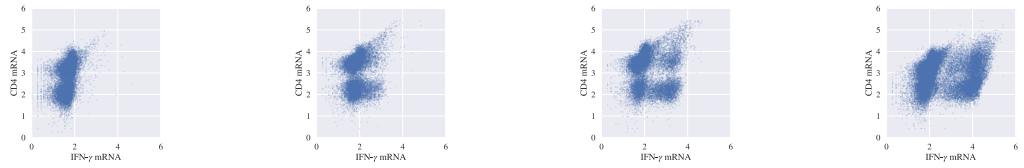
with three channels (*Red*, *Green*, and *Blue*). We project these three channels onto a single grayscale channel by computing the total intensity of each pixel in the image. The conversion of the image to a single channel allows us to treat the image as a cubical complex over which we can perform a simple Morse filtration (level sets defined in terms of intensity, as done in the previous diffusion field example). A grayscale image was used to simplify the computations; however, more complicated approaches could be taken to deal with the three color channels. In order to understand the important of the information contained in the Morse filtration analysis, we apply a linear SVM to the vectorized persistence diagrams associated to our images. We find that the topological features of the images gives us a classification accuracy of $85 \pm 2\%$ (for a dataset of more than 1,000 images). In order to identify the characteristic features for the responses at high and low concentration, we utilize the classification weights of the linear SVM model. The classification weights are

visualized in Figure 4.38. We apply a masking method to identify the portions of the persistence diagram that are critical for defining whether a response pattern is a result of high or low concentration. We can use these masked areas to identify the features of the images that separate the high concentration patterns from the low concentration patterns. To visualize the geometric differences between the patterns associated with DMMP and water, we again utilize the `Homcloud` software to perform the inverse analysis via volume optimal cycles. Here, we focus on inverse analysis for the H_1 homology group. The results of this analysis for a couple of sample images is found in Figure 4.38. Inverse analysis reveals that, when the sensors are exposed to water, the pattern exhibits many small distinct clusters; in contrast, when the sensor is exposed to DMMP, there are few large clusters. This shows how inverse analysis allows us to pinpoint topological features of the image that drive classification. The ability to classify as well as extract meaningful information from the topology of the images provides an important advantage over machine learning methods. The work of Smith, Cao, and co-workers demonstrate that machine learning models can be used to classify the responses of these sensors with higher accuracy, but provide minimal interpretability. Interpretability is needed to understand the physics of these sensors which can improve the sensitivity, and broaden the applicability, of the sensors [54, 56].

4.6.5 *Topology of Probability Density Functions*

The analysis of the shape of probability density functions is typically done using summarizing statistics (e.g., moments such as skewness and kurtosis) or through parametric techniques (fitting a parametric model such as a Gaussian mixture to the data) [144]. These models are powerful in their simplicity but might not be flexible enough to capture complex features of density functions (particularly in high dimensions). In this example we explore the shape of complex density functions by using topological techniques. We use an experimental flow cytometry dataset to illustrate how this can be done. The flow

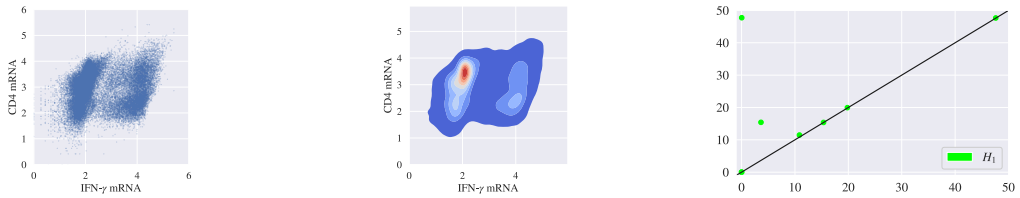
cytometry dataset was obtained through the FlowRepository [256] (Repository ID: FR-FCM-ZZC9). This dataset represents a temporal study of the kinetics of gene transcription and protein translation within stimulated human blood mononuclear cells through the quantification of proteins (CD_4 and IFN-γ) and mRNA (CD_4 and IFN-γ) [146]. In our study, we focus on the evolution of the concentration of CD_4 mRNA and IFN-γ mRNA in a given cell which is measured via a flow cytometer. At each time point in the study, a number of cells ($\sim 15,000$) are passed through the flow cytometer, each one of these cells provides a vector of scalar values $x \in \mathbb{R}^n$ corresponding to each measurable variable. In this case we set $n = 2$ as we are only utilizing the scalar values that represent the measure of both CD_4 mRNA and IFN-γ mRNA, samples of these distributions are found in Figure 4.39. From this we obtain a 2D scatter field. We have restricted ourselves to two dimensions for illustrative purposes, but this same analysis can be conducted for a point cloud of higher dimensions, accounting for all variables measured by a flow cytometer. The goal in this analysis is to use TDA to quantify the temporal evolution of the shape of the scatter field during the kinetic response of human blood mononuclear cells during stimulation. This approach provides an alternative to traditional parametric or heuristic methods such as gating, which are difficult to tune as they are highly sensitive to potential noise and outliers in the data. The gate selection may also require complicated multivariate mixture models to identify the correct gating values [147].



(a) Time = 0 Minutes (b) Time = 30 Minutes (c) Time = 60 Minutes (d) Time = 90 Minutes

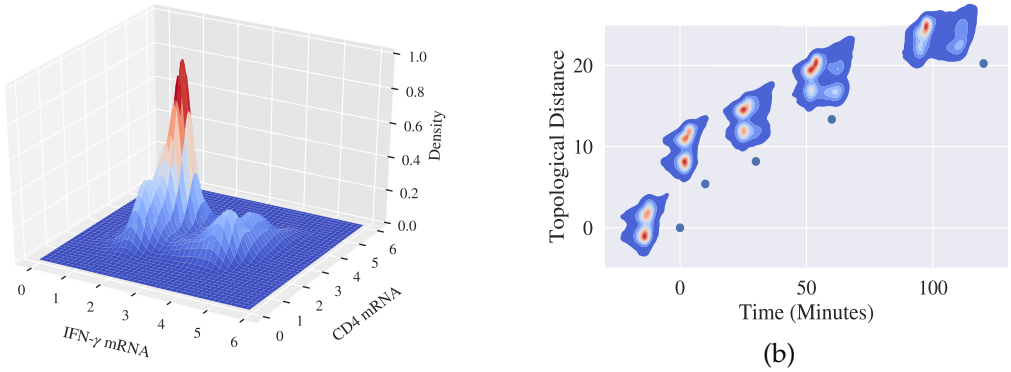
Figure 4.39: Deformation of a 2D scatter field over time.

In order to analyze the topology of the scatter fields, we utilize Gaussian kernel smoothing [148]. The work of [257] demonstrates that provided a large enough sample, the homology of the Gaussian kernel density estimate derived from a sample is equivalent



(a) Sample Distribution. (b) Gaussian Density Estimate (c) Persistence Diagram

Figure 4.40: Processing of the flow cytometry scatter field. The raw data is first smoothed via a Gaussian kernel and then the smoothed diagram is processed via a Morse filtration.



(a) (b)

Figure 4.41: (a) Scatter field as a probability density function in 3D; Morse filtration is performed using level sets of the probability density. (b) Wasserstein distance between the persistence diagrams as a function of time. There is a clear continuous evolution of the Wasserstein distance that characterizes the change in topology.

to the homology of the true density. Figure 4.40 shows the Gaussian kernel smoothing of a flow cytometry scatter plot. An example persistence diagram for the function is shown in Figure 4.40. As in the case of the diffusion example, in 3D we can represent the scatter field as a continuous function (in this case a probability density function). This probability density function (Figure 4.41a) can be analyzed using Morse filtration. Our goal in this analysis is to quantify the time evolution of the probability density functions during stimulation. Our strategy consists of computing the Wasserstein distance (w_d) between the persistence diagram of a given time point to the persistence diagram of the sample at time zero [258]. Specifically, given flow cytometry samples X_1, X_2 and the time zero sample X_0 as well as their corresponding persistence diagrams PD_1, PD_2, PD_0 and time points $T_0 \leq T_1 \leq T_2$, we observe that:

$$w_d(PD_1, PD_0) \leq w_d(PD_2, PD_0) \tag{4.33}$$

From Figure 4.41b, we can see that the distance exhibits a strong dependence on time. This suggests that there exists a continuous mapping between the persistent diagram and time (the topological deformation is continuous with respect to time). This again reveals the continuity of the persistence diagrams (of topology) to perturbations. These results highlight how topological data analysis provides a quantifiable approach to characterize complex probability density functions and their evolution over time.

4.6.6 *Topology of 3D Fields*

We now illustrate how to use TDA to analyze the topology induced by 3D point clouds. Specifically, we study datsets generated by molecular dynamics (MD) simulations [259, 182]. The dataset under study analyzes the influence of the 3D liquid-phase environment formed by molecules of a solvent, co-solvent, and a reactant on reactivity [259]. The reactivity is quantified via a *kinetic solvent parameter* σ that is obtained from experiments. Experiments suggest that reactivity is influenced by hydrophilicity of the solvent. The

main hypothesis is that, as the solvent concentration is increased, the water in the system is concentrated around the solvent molecule, and that molecules with high hydrophilicity are able to take advantage of this effect. In order to study this hypothesis, molecular dynamics computations were performed in [259]. The data output of an MD simulation has both spatial and temporal dimensions. Each simulation gives atomic positions $X_t \in \mathbb{R}^{M\times 3}$ (M is the number of species) at multiple times t (measured in nanoseconds). In our analysis, we utilize a 3D point cloud of water molecule positions that result from a time average of 100 nanoseconds. An example of this point cloud (visualized as a field) is provided in Figure 4.42. Each density field is labeled with a reactivity σ obtained from experiments. Recent work by Chew and co-workers has analyzed the 3D density by using 3D convolutional neural networks (CNNs) and has shown that the features extracted from the CNN are strongly correlated to reactivity [259]. CNNs are highly effective tools but require a large number of parameters ($\sim$160,000 in this case) and are difficult to interpret. Our goal is to study if the topology of the 3D density can be characterized in a more straightforward manner and to explore whether topology changes in correlation with reactivity.

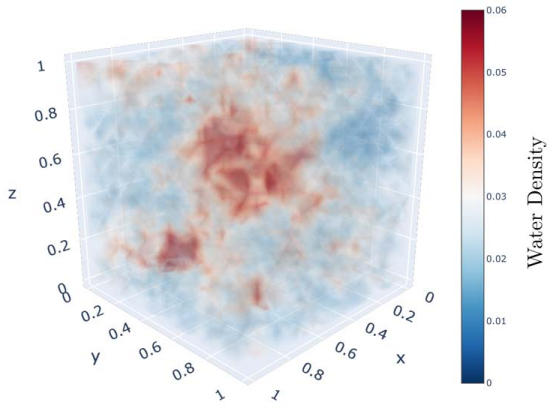
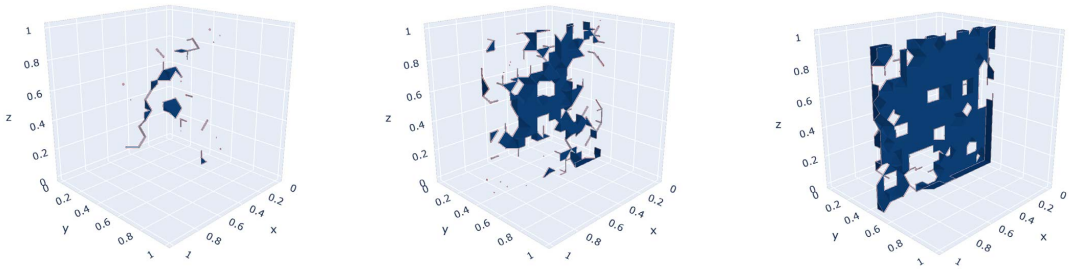


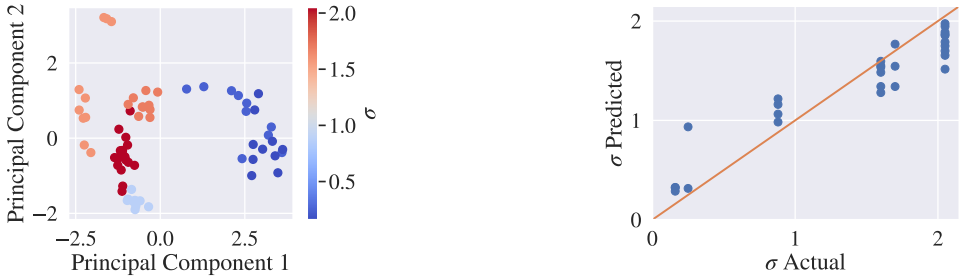
Figure 4.42: Visualization of 3D water density field generated by MD simulation.

To perform our analysis, we treat the 3D point cloud as a continuous field (function). Here, we perform a Morse filtration and treat the data as a 3D cubical complex (a voxel).



(a) Density Level Set of 0.01 (b) Density Level Set of 0.02 (c) Density Level Set of 0.03

Figure 4.43: Slices of 3D water density field as filtration proceeds for different density values. The filtration reveals the presence of voids in the data associated with high concentrations of water molecules.



(a) PCA analysis (b) SVM Regression

Figure 4.44: (a) PCA analysis on persistence diagrams for MD simulations. The analysis reveals strong dependence of reactivity σ. (b) Regression plot for SVM with a radial basis function. The predictions over 5-fold cross validation yield an MSE of 0.07 ± 0.003.

The filtration will be done by exploring levels sets for the water density. Note that this is a filtration in a higher dimension that our previous examples for the diffusion field and for liquid crystal sensors. The main focus of this approach is to capture the clustering of water near hydrophilic molecules, and the lack of clustering near non-hydrophilic molecules. We visualize the water density filtration in Figure 4.43 via a 2-dimensional slice. We can see that voids in the data are generated as we increase the filtration value. The voids represent areas of high water density which is precisely what we wish to quantify. From these filtrations, we produce persistence diagrams focused on H_2 (since this homology group quantifies these voids). The persistent diagrams are then vectorized via the persistence image and we use PCA reduction to visualize them (see Figure 4.44a). We use SVM regression with a radial basis kernel to predict reactivity as a function of the persistence diagrams (see Figure 4.44b).

The PCA projection reveals that there is strong dependence of the reactivity on the topology. This suggests that the information gained via persistent homology extracts informative features of the 3D field that explain reactivity. This also suggests that a simple regression method (as opposed to a complex neural network) would be effective at predicting the reactivity. In order to test this hypothesis, the experimental dataset with 70 points is split into train/test sets with 49/21, points to build the SVM regression model. A 5-fold cross validation is performed to estimate the performance of the SVM regression. We found that this model captures the trend of reactivity well; moreover, this simple model yields a mean square error of $0.07 \pm .003$ (Figure 4.44). These results are relevant because they indicate that it is possible to predict experimental reactivity directly from MD simulations.

Part II

GEOMETRY

Chapter 5

RIEMANNIAN GEOMETRY

The contents of this chapter are published in [10]

5.1 Introduction

The assumption that data lies in a Euclidean space is pervasive throughout science and engineering and is the basis of diverse data analysis techniques used in these domains. Making this blanket assumption, however, is not always appropriate and can affect the accuracy/interpretability of such techniques or even break fundamental physical laws.

Recognizing that data can live in spaces that are governed by *non-Euclidean* geometry is critical to appropriately representing, manipulating, and analyzing certain data objects. A simple example of this arises when computing travel paths between a couple of points that are located on the surface of the Earth; when computing the distance between such points, the *elliptic* geometry of the Earth surface must be taken into consideration. If Euclidean geometry is assumed, travel paths between antipodal locations (e.g., United States and China) can require traversing the Earth (not physically-realizable paths).

Elliptic geometry is a non-Euclidean geometry in which one of the postulates of Eu-

cliean geometry (sum of interior angles of a triangle is equal to 180^0) no longer holds, because of the presence of positive curvature in a surface [260]. Elliptic geometry was first proposed by Bernhard Riemann in the 19th century, and he further developed these ideas into the field that is now known as Riemannian geometry [261].

An important example in which assuming Euclidean geometry can lead to spurious results is in the analysis of *symmetric positive definite* (SPD) matrices (e.g., covariance/correlation matrices). SPD matrices lie on a high-dimensional space which is governed by Riemannian geometry (known as a Riemannian manifold) [262]. Standard techniques for the analysis of SPD matrices (e.g., PCA or basic matrix norms) do not take this property into consideration and can lead to misleading results. For instance, computing the distance between SPD matrices in Euclidean geometry (e.g., via the Frobenius norm) ignores the fact that such matrices live on a Riemannian manifold, and this can yield misleading results [263, 264, 265]. Specifically, the so-called *swelling effect* can occur when applying operations in Euclidean geometry to SPD matrices [264]. This effect introduces spurious results by inflating the determinants of SPD matrices and can also distort the results of commonly used methods [265]. Computing interpolations and averages of SPD matrices, which is key in understanding physical systems (e.g. Brownian motion), can also break physical conservation laws if performed under Euclidean geometry [266].

In this chapter, we focus our attention on the use of techniques from Riemannian geometry for the analysis of data objects that can be represented as SPD matrices. An SPD matrix is a simple but versatile data representation that is widely used in multivariate analysis techniques such as PCA [267, 268, 269]. SPD representations are also used in process control, monitoring, and anomaly detection [270, 271, 272, 273, 274, 31], in the study of functional brain networks [275, 276, 277, 278, 31], in object detection [279, 280, 281], in biomedical image analysis [266, 282], in the analysis of Laplacian matrices in graph theory, and in the analysis of Hessian matrices in optimization [283]. In applications such

as image analysis, an SPD representation can be obtained by applying transformations to a raw data object (e.g., smoothing via kernel functions and/or combination of image features). The defining feature of an SPD matrix is that all its eigenvalues are real and positive. In the context of optimization, it is well-known that an SPD Hessian matrix defines positive curvature of a multi-dimensional quadratic function and is key in defining the geometry of objective and constraint functions (e.g., convex or non-convex). In the context of statistics, it is well-known that an SPD covariance matrix defines a multi-dimensional ellipsoid (a surface with positive curvature) and that the level sets of a multivariate Gaussian probability density function are ellipsoids.

This chapter provides a practical introduction to the Riemannian geometry of SPD matrices and demonstrates applications of interest to the process systems engineering community. Specifically, we illustrate the benefits of exploiting the Riemannian geometry of SPD matrices and discuss how these tools can be incorporated into common dimensionality reduction and classification workflows. We also provide example applications of interest in science and engineering. The first application focuses on analysis of covariance matrices derived from multivariate time series, which is a common task in process monitoring. Our application focuses on the so-called Tennessee Eastman Process (TEP) [4]. The TEP is a process where anomalies/faults are systematically introduced which shifts the relationships between the measured variables. Covariance matrices encode these changing relationships and are then used to predict what type of anomaly the process is experiencing. The second application is in defect/anomaly detection of textiles taken from the MVTEC AD dataset [1]. Here, grayscale images of textiles are represented as covariance matrices by characterizing the relationships between the original image and multiple transformations of the image (e.g., smoothing with different kernels). The image transformations emphasize different features of the original image (e.g., edges and fibers). Subsequently, when an anomaly/defect is introduced (e.g., a cut or discoloration of the textile) these relationships will change, impacting the covariance matrix and allow-

ing us to detect anomalies. This type of analysis can also be applied to other relevant image/field datasets such as those arising in microscopy and flow cytometry [31, 168]. All data and scripts needed to conduct such analyses is provided as open-source code in https://github.com/zavalab/ML/tree/master/RiemannianSPD. With this, we aim to provide a concise and easy introduction to non-experts to the field of Riemannian geometry.

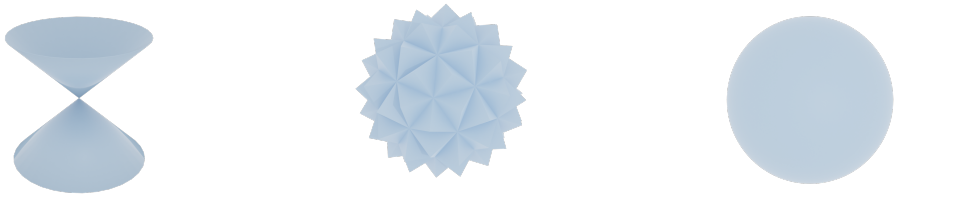
5.2 Riemannian Manifolds

The key observation driving this work is that SPD matrices lie on a Riemannian manifold; we thus begin our discussion by characterizing such manifolds. In this section, we aim to provide an intuitive understanding of manifolds and equip the reader with knowledge of their key properties.

An n-dimensional manifold is a topological space (a space where closeness and connectedness are defined but not directly measurable) that *locally* resembles an n-dimensional Euclidean space. Specifically, an n-dimensional manifold M is a set where every point $p \in M$ has an open neighborhood $U \subset M$ (known as a *chart*) that can be mapped to an open set of n-dimensional Euclidean space $V \subset \mathbb{R}^n$ via a one-to-one, onto, and continuous mapping $f : U \to V$ (i.e., a homeomorphism). The set of charts whose union covers the manifold is known as an *atlas* $\bigcup_{i=1}^{n} U_i = M$. The chart/atlas nomenclature is derived from navigation along the surface of the Earth (a 2D manifold); here, charts are flat (Euclidean) maps of the Earth that are collected in an atlas.

In Figure 5.1 we illustrate multiple topological spaces embedded in a 3D Euclidean space. The spiked (b) and smooth (c) hollow spheres are examples of 2D manifolds. The space in (a) represents a couple of cones that intersect at a single point and is not manifold; this is because a neighborhood drawn around the intersecting point will look

like a smaller version of the intersecting cones, which cannot be mapped to 2D Euclidean space through a homeomorphism.



(a) Non-Manifold (b) Non-Differentiable Manifold (c) Differentiable Manifold

Figure 5.1: (a) Space composed of a couple of cones intersecting at a single point. This is a non-manifold space because any neighborhood formed around the intersecting point is not homeomorphic to 2D Euclidean space (the neighborhood is a smaller version of the two intersecting cones). (b) Represents a 2D manifold (all points and associated neighborhoods can be mapped to 2D Euclidean space) but is not a differentiable manifold because of the cusps occurring at the edges of the manifold (differential is not defined everywhere). (c) A smooth sphere is a 2D manifold that is also differentiable (curves on the surface can be differentiated everywhere).

Manifolds can also be endowed with geometric structure; for our analysis, we are particularly interested in whether or not a given manifold is *differentiable*. In simple terms, a differentiable manifold is a manifold for which calculus (e.g., computing derivatives and integrals) can be performed on the charts that make up the manifold atlas [284]. This also means that *curves* on the surface of the manifold can be analyzed from a geometric perspective using calculus. A curve is defined as a continuous function $\gamma : [a, b] \rightarrow M$ mapping the interval $[a, b] \in \mathbb{R}$ to the manifold M. We will not cover the specific mathematical requirements that make a manifold differentiable in general, but refer interested readers to the following reference for details [284]. Examples of a differentiable and non-differentiable manifold are shown in Figure 5.1. We can see that the spiked sphere in (b) has multiple cusps where a derivative cannot be defined for a curve, making it non-differentiable. For the smooth sphere in (c), a derivative can be taken anywhere, allowing for more complex operations/transformations to be performed on the manifold [285].

We now restrict our attention to a special class of manifolds known as *Riemannian*

manifolds. A Riemannian manifold is a differentiable manifold equipped with a defined *tangent space* at each point in the manifold $p \in M$, denoted as T_pM [262]. The tangent space is the set of tangent vectors of all curves passing through point $p \in M$. The tangent space T_pM is a vector (linear) space that is of the same dimension as the manifold itself. An illustration of a tangent space is shown in Figure 5.2 for a smooth sphere (a 2D Riemannian manifold). For a Riemannian manifold, the tangent space T_pM is equipped with an inner product $g_p : T_pM \times T_pM \to \mathbb{R}$, along with a norm metric $|\cdot|_p : T_pM \to \mathbb{R}$ defined by $|v|_p = \sqrt{g_p(v,v)}$ for any vector $v \in T_pM$. These properties allow us to define the *length* of a curve on the manifold surface. A differentiable curve $\gamma : [a,b] \to M$ assigns to each $t \in (a,b)$ a tangent vector $\gamma'(t) \in T_{\gamma(t)}M$; thus, to obtain the length of the curve $L(\gamma)$, we integrate the norm of the tangent vectors along the curve (i.e., arc length):

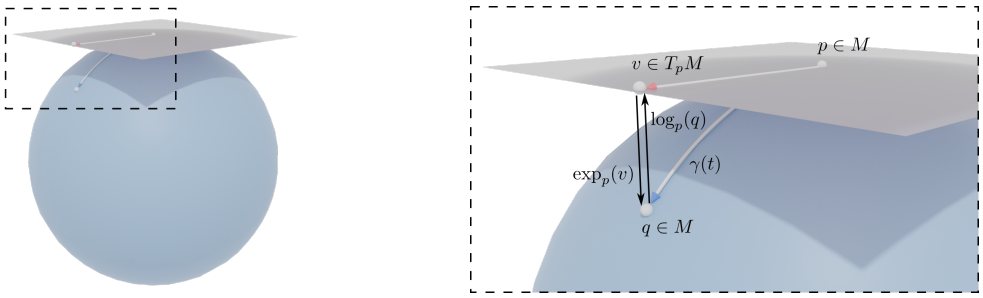
$$L(\gamma) := \int_a^b |\gamma'(t)|_{\gamma(t)}\, dt \tag{5.1}$$

In our analysis, we are primarily interested in measuring the *shortest* curve between a couple points on the manifold (known as a *geodesic*). Given a couple of points on a Riemannian manifold $p, q \in M$ and the set of all curves $\gamma : [a,b] \to M$ such that $\gamma(a) = p$ and $\gamma(b) = q$, the geodesic $\bar{\gamma}$ is the curve with the shortest total length $L(\bar{\gamma})$:

$$L(\bar{\gamma}) := \inf\{L(\gamma) \mid \gamma : [a,b] \to M, \text{with } \gamma(a) = p, \gamma(b) = q\} \tag{5.2}$$

An illustration of the geodesic between a couple of points on the smooth sphere is shown in Figure 5.2. Geodesics are a powerful tool for quantifying the relationship between points on a manifold surface and can be used to compute summarizing statistics for points on the surface (such as means and variances) [285].

There are direct relationships between the tangent space of a Riemannian manifold and geodesics, such as the *exponential map* and the *logarithmic map*. For a tangent vector $v \in T_pM$ constructed at point $p \in M$, there exists a unique geodesic $\gamma : [0,1] \to M$ such that $\gamma(0) = p$ and $\gamma'(0) = v$. The vector $v \in T_pM$ is mapped to the endpoint of the



(a) (b)

Figure 5.2: (a) Illustration of a Riemannian manifold (M) and the associated tangent space at a point $p \in M$. (b) Illustration of a geodesic $\gamma(t)$ constructed between two points $p, q \in M$, along with the associated tangent space vector $v \in T_pM$. The exponential map ($\exp_p(v) : v \to q$) and the logarithmic map ($\log_p(q) : q \to v$) are also shown.

geodesic $\gamma(1) \in M$ through the exponential map (see Figure 5.2):

$$\exp_p(v) = \gamma(1). \tag{5.3}$$

The inverse of the exponential map is the logarithmic map, which maps the point $\gamma(1)$ in the neighborhood of $p \in M$ to a vector in the tangent space $v \in T_pM$ (see Figure 5.2):

$$\log_p(\gamma(1)) = v \tag{5.4}$$

These functions provide a mapping from the surface of the manifold to the tangent space of a given point. The tangent space, which is a (linear) vector space, can be analyzed using standard techniques designed for Euclidean spaces (e.g., classification, regression, and dimensionality reduction) while properly capturing the relationships between points defined by the geometry of the manifold (e.g., geodesics). For instance, we can use these constructs to analyze and operate on the space of SPD matrices, as we discuss next.

5.3 Riemannian Geometry of SPD Matrices

This section will provide the mathematical background needed to understand the geometry of the space defined by SPD matrices. We first introduce the needed notation and define the properties of different matrix spaces that are used in our analysis. We then demonstrate how the properties of these matrix spaces can be used to quantify the geometry and structure of the space defined by SPD matrices. The main message is that SPD matrices lie on a Riemannian manifold and that important computations (e.g., matrix operations, summarizing statistics, classification, regression, and dimensionality reduction) can be performed by respecting the geometry of this manifold by conducting these on the tangent space; respecting such properties can lead to important improvements in efficiency and interpretability.

5.3.1 *Matrix Spaces, Properties, and Notation*

We define spaces and properties of matrices that reflect the structure of SPD matrices. We denote $\mathcal{S}(n) := \{\mathbf{S} \in \mathcal{M}(n), \mathbf{S} = \mathbf{S}^T\}$ as the set of symmetric $n \times n$ matrices in the space of square, real matrices $\mathcal{M}(n)$ and the set $\mathcal{P}(n) := \{\mathbf{P} \in \mathcal{S}(n), u^T\mathbf{P}u > 0, \forall\ u \in \mathbb{R}^n\}$ as the set of all $n \times n$ SPD matrices.

We also define the exponential and logarithmic mappings used in our analysis. The matrix exponential $\exp(\mathbf{P})$, where $\mathbf{P} \in \mathcal{S}(n)$, is defined as:

$$\exp(\mathbf{P}) := \mathbf{U}\ \mathrm{diag}(\exp(\lambda_1), ..., \exp(\lambda_n))\ \mathbf{U}^T \tag{5.5}$$

where $\mathbf{U}$ represents the matrix of eigenvectors of $\mathbf{P}$ and $\lambda_1 > ... > \lambda_n$ represent the eigenvalues of $\mathbf{P}$ (also denoted as $\lambda_i(\mathbf{P})$). We also define the inverse operation; the matrix logarithm $\log(\mathbf{P})$ as:

$$\log(\mathbf{P}) := \mathbf{U}\ \mathrm{diag}(\log(\lambda_1), ..., \log(\lambda_n))\ \mathbf{U}^T. \tag{5.6}$$

The following properties should also be considered in the analysis [286]:

- $\forall\ \mathbf{P} \in \mathcal{P}(n)$ we have that $\det(\mathbf{P}) > 0$
- $\forall\ \mathbf{P} \in \mathcal{P}(n)$ we have that $\mathbf{P}^{-1} \in \mathcal{P}(n)$
- $\forall\ \mathbf{P} \in \mathcal{P}(n)$ we have that $\log(\mathbf{P}) \in \mathcal{S}(n)$
- $\forall\ \mathbf{S} \in \mathcal{S}(n)$ we have that $\exp(\mathbf{S}) \in \mathcal{P}(n)$

We also define the Frobenius inner product for matrices $\mathbf{A}$ and $\mathbf{B}$ as:

$$\langle \mathbf{A}, \mathbf{B} \rangle_F := \mathrm{Tr}(\mathbf{A}^T\mathbf{B}), \tag{5.7}$$

where $\mathrm{Tr}(\cdot)$ represents the matrix trace operator. The Frobenius norm for a matrix $\mathbf{A}$ is:

$$||\mathbf{A}||_F = \sqrt{\mathrm{Tr}(\mathbf{A}^T\mathbf{A})} = \sqrt{\sum_{i=1}^{n} \lambda_i(\mathbf{A})}. \tag{5.8}$$

5.3.2 *Manifold of SPD Matrices*

To understand the Riemannian geometry of the space of SPD Matrices ($\mathcal{P}(n)$), we first need to construct a Riemannian *metric*, which will allow us to compute distances and other relationships between points on a manifold. The metric we consider for Riemannian manifolds is known as the *Affine Invariant Riemannian Metric (AIRM)*; a detailed derivation of the metric is found in the work of Bhatia [287]. We aim to provide an intuitive understanding of this metric, and its meaning.

Given SPD matrices $\mathbf{A} \in \mathcal{P}(n)$ and $\mathbf{B} \in \mathcal{P}(n)$, we construct a geodesic $\gamma(t) : [0, 1] \rightarrow \mathcal{P}(n)$ parameterized as:

$$\gamma(t) = \exp\big((1-t)\cdot\log(\mathbf{A}) + t\cdot\log(\mathbf{B})\big) \tag{5.9}$$

where we have log maps $\mathcal{P}(n) \rightarrow \mathcal{S}(n)$ and exp maps $\mathcal{S}(n) \rightarrow \mathcal{P}(n)$. Here, we are using the $n \times n$ identity matrix $\mathbf{I} \in \mathcal{P}(n)$ as our tangent space basis. Informally, we are mapping our matrices from our SPD manifold $\mathcal{P}(n)$ to the tangent vector space $\mathcal{S}(n)$ via the logarithmic map, constructing a line between these points in the tangent space, and then projecting the constructed line back to the manifold via the exponential map. This is guaranteed to be the shortest length path between the points in $\mathcal{P}(n)$. The corresponding geodesic distance between matrices $\mathbf{A}$ and $\mathbf{B}$ is given by:

$$d_g(\mathbf{A},\mathbf{B}) := ||\log(\mathbf{A}) - \log(\mathbf{B})||_F. \tag{5.10}$$

Here, note that we are simply projecting the matrices from the Riemannian manifold to the tangent vector space prior to measuring their distance using the Frobenius norm [287]. However, an important consideration must be made when using the $n \times n$ identity matrix $\mathbf{I}$ as the tangent space basis. In many applications, the data may lie within a particular neighborhood that is far from $\mathbf{I}$ on the manifold. Thus, projections to the tangent space $T_I M$ can result in distortions of the data [285]. Intuitively, one can think of this as similar to an analysis of a projection of the Earth surface onto a plane tangent to the 0′ latitude and 0′ longitude point (as many maps are represented). In this projection, landmasses near the edge of the projection are highly distorted, whereas points near (0′,0′) have almost no distortion. Thus, our aim is to identify a distance with respect to a tangent space defined by the matrices of concern. This distance can be constructed using a critical property of this Riemannian manifold and metric: *congruence invariance*.

Congruence invariance states that, for any $n \times n$ invertible matrix $\mathbf{X}$ and matrices

$\mathbf{A}, \mathbf{B} \in \mathcal{P}(n)$:

$$d_g(\mathbf{X}^T\mathbf{A}\mathbf{X}, \mathbf{X}^T\mathbf{B}\mathbf{X}) = d_g(\mathbf{A}, \mathbf{B}) \tag{5.11}$$

We thus have that linear transformations of the given matrices do not impact the geodesic distance on the manifold. This property allows us to redefine the geodesic distance as:

$$\begin{aligned} d_g(\mathbf{A}, \mathbf{B}) &= d_g(\mathbf{I}, \mathbf{A}^{-1/2}\mathbf{B}\mathbf{A}^{-1/2}) && \text{(5.12a)} \\ &= ||\log(\mathbf{I}) - \log(\mathbf{A}^{-1/2}\mathbf{B}\mathbf{A}^{-1/2})||_F && \text{(5.12b)} \\ &= ||\log(\mathbf{A}^{-1/2}\mathbf{B}\mathbf{A}^{-1/2})||_F && \text{(5.12c)} \\ &= \left(\sum_{i=1}^{n} \log^2 \lambda_i(\mathbf{A}^{-1}\mathbf{B}) \right)^{1/2} && \text{(5.12d)} \end{aligned}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix, $\mathbf{A} = \mathbf{A}^{1/2}\mathbf{A}^{1/2}$, and $\lambda_i(\mathbf{A}^{-1}\mathbf{B})$ are the eigenvalues of $\mathbf{A}^{-1}\mathbf{B}$.

We can simplify the geodesic distance further; we have that the geodesic between matrices $\mathbf{I}$ and $\mathbf{A}^{-1/2}\mathbf{B}\mathbf{A}^{-1/2}$ is:

$$\gamma_0(t) = \exp\big(\log(\mathbf{A}^{-1/2}\mathbf{B}\mathbf{A}^{-1/2})t\big) = (\mathbf{A}^{-1/2}\mathbf{B}\mathbf{A}^{-1/2})^t \tag{5.13}$$

We can then leverage congruence invariance to shift this geodesic to our matrices of interest as:

$$\gamma(t) = \mathbf{A}^{1/2}\big(\gamma_0(t)\big)\mathbf{A}^{1/2} = \mathbf{A}^{1/2}(\mathbf{A}^{-1/2}\mathbf{B}\mathbf{A}^{-1/2})^t\mathbf{A}^{1/2} \tag{5.14}$$

where $\gamma(0) = \mathbf{A}$ and $\gamma(1) = \mathbf{B}$, which provides a geodesic that is independent of the tangent space basis. Essentially, we are leveraging congruence invariance to translate our points to a neighborhood of $\mathbf{I}$, which allows us to compute distances with minimal distortion, and then translate the points back through these linear transformations. We can apply this same logic to the exponential map (and the logarithmic map); for matrices $\mathbf{A}, \mathbf{B} \in \mathcal{P}(n)$ and $\mathbf{T}_B \in T_A M$ where $T_A M \subset \mathcal{S}(n)$:

$$\mathbf{B} = \exp_A(\mathbf{T_B}) = \mathbf{A}^{1/2}\exp(\mathbf{A}^{-1/2}\mathbf{T_B}\mathbf{A}^{-1/2})\mathbf{A}^{1/2} \tag{5.15}$$

$$\mathbf{T_B} = \log_A(\mathbf{B}) = \mathbf{A}^{1/2}\log(\mathbf{A}^{-1/2}\mathbf{B}\mathbf{A}^{-1/2})\mathbf{A}^{1/2} \tag{5.16}$$

Here, we are mapping the matrix $\mathbf{B} \in \mathcal{P}(n)$ to the tangent space vector $\mathbf{T}_B \in T_A M$ which is centered at $\mathbf{A} \in \mathcal{P}(n)$ via $\log_A(\mathbf{B})$ and inversely through $\exp_A(\mathbf{B})$. Thus, with these newly defined geodesics and mappings, we are able to compute relationships between SPD matrices with minimal distortions in the tangent space. However, when given a large dataset with multiple SPD matrices, the choice of a tangent space basis may not be immediately clear. In this case, the geometric mean of the matrices on the manifold is typically identified and used as reference point.

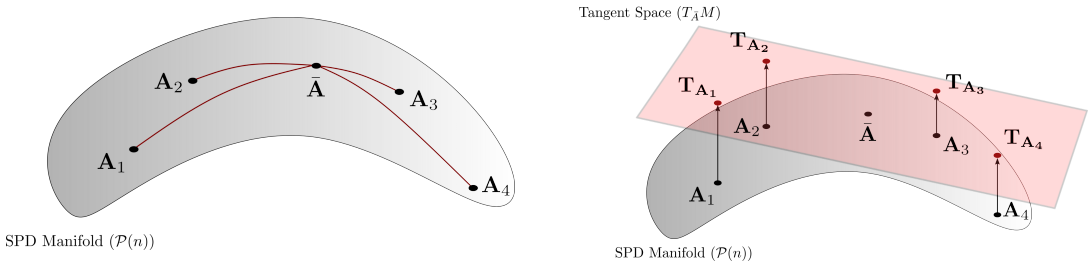
5.3.3 *SPD Matrix Means and Tangent Spaces*

In the analysis of a set of SPD matrices, we often need to identify a center point on the SPD manifold that will minimize the distortion of all geometric relationships between the matrices of the dataset when mapped to the tangent space. This matrix is the (Riemannian) geometric mean of the matrices [288]. For a set of SPD matrices $\mathbf{A}_i$, the geometric mean (see Figure 5.3) is the matrix $\bar{\mathbf{A}}$ that minimizes the sum of squared geodesic distances to

all other matrices in the set:

$$\bar{\mathbf{A}} := \underset{\mathbf{M}}{\operatorname{argmin}} \sum_{i=1}^{n} ||\log(\mathbf{M}^{-1/2}\mathbf{A}_i\mathbf{M}^{-1/2})||_F \tag{5.17}$$

We can see that the geometric mean is obtained by solving a matrix optimization problem. For the SPD manifold this problem is geodesically convex (similar to Euclidean convexity) [289]. This optimization problem can be solved by using classical optimization algorithms that have been adapted to geometric setting (e.g., gradient descent) [289]. A detailed review on these approaches can be found in the work by Absil, Mahony, and Sepulchre [289].



(a) Geometric mean.

(b) Tangent space projection.

Figure 5.3: (a) Illustration of the geometric mean $\bar{\mathbf{A}}$ of a set of matrices $\{\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4\} \in \mathcal{P}(n)$. The geometric mean represents a point on the manifold that minimizes the geodesic to all other matrices in the set. (b) Representation of the tangent space $T_{\bar{A}}M$ at the geometric mean. The set of matrices $\mathbf{A}_i$ are projected (through the logarithmic map) onto this tangent space with minimal geometric distortion $\mathbf{T}_{\mathbf{A}_i} \in T_{\bar{A}}M$.

Given a set of SPD matrices $\mathbf{A}_i$ and a geometric mean $\bar{\mathbf{A}}$, we can construct a tangent space at the geometric mean $T_{\bar{A}}M$, and project the SPD matrices onto the tangent space $\mathbf{T}_{\mathbf{A}_i} = \log_{\bar{A}}(\mathbf{A}_i)$, as shown in Figure 5.3. The matrices are now represented in a vector (linear) space that reflects the geometry of the SPD manifold. Projecting the data into a vector space allows us to apply common matrix analysis methods such as PCA.

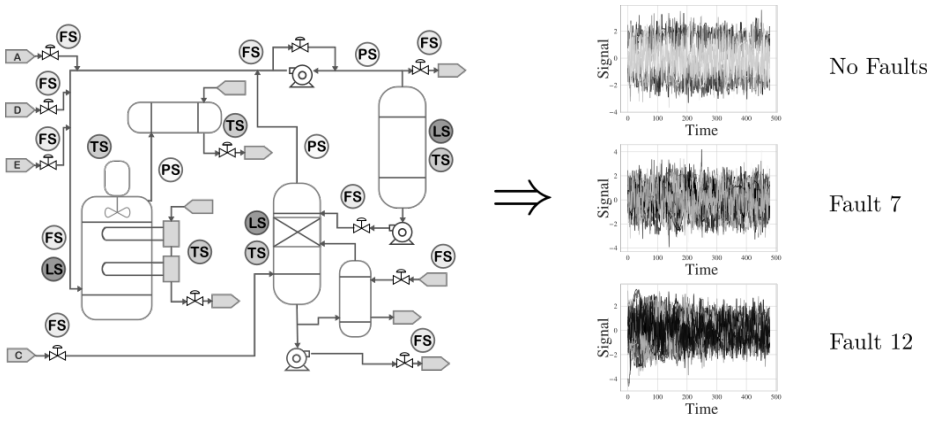


Figure 5.4: Simplified illustration of the Tennessee Eastman Process (TEP) and resulting multivariate time series sensor data. Process sensors measure values such as temperature, pressure, flow, and level. From the multivariate time series data it is difficult to distinguish whether there is a fault occurring, or what type of fault may be occurring. Thus, a simplified and informative representation of the data is needed to be able to distinguish when the process is behaving normally or is experiencing a particular fault.

5.4 Case Study - Process Monitoring

We focus on data obtained from a simulated industrial process known as the Tennessee Eastman Process (TEP) [4]. This dataset is a widely used benchmark dataset for testing and comparing various anomaly (i.e. fault) detection methods [290, 291, 19]. Figure 5.4 provides a high-level illustration of the process along with the multivariate time series data that is produced by the sensors monitoring the process. The process has a total of 52 measurements, 41 are process variables, 11 are manipulated variables. There are 20 different potential faults, which are defined in Table 5.1 (for further details see Appendix A in [4]). Our aim is to use geometric methods to detect and classify the presence and type of fault using only the multivariate sensor data.

The difficulty in distinguishing potential faults in the TEP is illustrated in Figure 5.4, where multivariate time series sensor signals for the 52 monitored variables are shown for the TEP without faults, and in the presence of fault 7 (step change in the component "C" header pressure) and fault 12 (random variation in the condenser cooling water inlet). The complexity of the multivariate process dynamics and of the number of sensor mea-

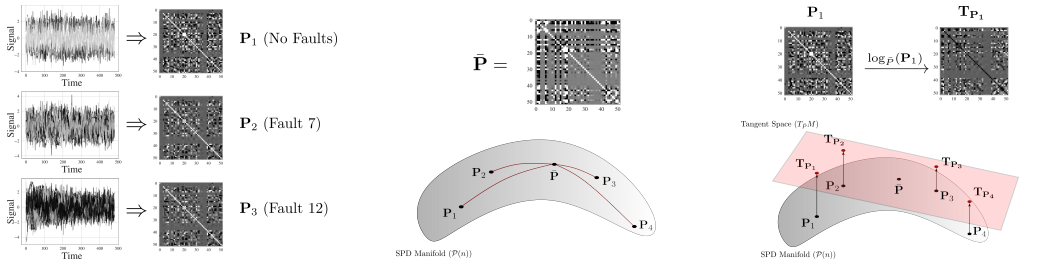
Table 5.1: Types of Faults for Tennessee Eastman Process [4].

Fault ID	Fault Name	Type
Fault 1	A/C feed ratio, B composition constant (stream 4)	Step
Fault 2	B composition, A/C ratio constant (stream 4)	Step
Fault 3	D feed temperature (stream 2)	Step
Fault 4	Reactor cooling water inlet temperature	Step
Fault 5	Condenser cooling water inlet temperature	Step
Fault 6	A feed loss (stream 1)	Step
Fault 7	C header pressure loss - reduced availability (stream 4)	Step
Fault 8	A, B, C feed composition (stream 4)	Random variation
Fault 9	D feed temperature (stream 2)	Random variation
Fault 10	C feed temperature (stream 4)	Random variation
Fault 11	Reactor cooling water inlet temperature	Random variation
Fault 12	Condenser cooling water inlet temperature	Random variation
Fault 13	Reaction kinetics	Slow drift
Fault 14	Reactor cooling water valve	Sticking
Fault 15	Condenser cooling water valve	Sticking
Fault 16-20	Unknown	Unknown

surements make it difficult to reliably identify if a fault is occurring and to distinguish between fault types. Our method focuses on simplifying the data by quantifying the relationships between the 52 measured variables through covariance matrices and leveraging the geometry of the covariance matrices to detect and distinguish faults in the TEP.

5.4.1 *Data Pre-Processing*

To begin our analysis of the TEP, we must first pre-process the TEP data into multiple covariance matrices. To accomplish this, we represent each of the 52 measured variables as a univariate random variable x_i where $i = 1, 2, ..., 52$ and we denote the collection of signals as a multivariate random vector $\mathbf{X} = (x_1, ..., x_n)$ where $n = 52$. We denote the observations of each signal at time $t = 1, 2, .., m$ as $x_i(t) \in \mathbb{R}^m$. We use this representation to construct the sample covariance matrix for the process data $\mathbf{P} \in \mathcal{P}(n)$ as:



(a) Covariance matrices. (b) Geometric mean. (c) Tangent space projection.

Figure 5.5: Representation of the data pre-processing workflow for the geometric analysis of the TEP data. (a) Covariance matrices are constructed from the sensor data multivariate time series forming a set of SPD matrices ($\mathbf{P}_i \in \mathcal{P}(n)$). (b) The geometric mean of the matrices ($\bar{\mathbf{P}} \in \mathcal{P}(n)$) is identified as the point that minimizes the squared geodesic distance to all other points. (c) All derived covariance matrices are mapped to the tangent space $T_{\bar{P}}M$ through the logarithmic mapping. This maps the matrices into a vector space that reflects the manifold geometry. The mapped data can then be analyzed with commonly defined dimensionality reduction and classification/regression methods.

$$\mathbf{P} := \frac{1}{m-1}\mathbf{X}\mathbf{X}^T \tag{5.18}$$

The TEP dataset consists of multiple separate simulations of the process, both with and without faults. Thus, for each simulation we construct a sample covariance matrix $\mathbf{P}_i$. Our goal is to pair each simulation sample covariance matrix with the fault occurring in the simulation. Figure 5.5 provides examples of the sample covariance matrices that are constructed from simulations containing no faults, fault 7, and fault 12. We note that there are no obvious differences between the covariance matrices that would identify a given fault. These covariance matrices lie on the SPD manifold, and can be integrated into our geometric framework. An illustration of this computational workflow is found in Figure 5.5. We compute the geometric mean $\bar{\mathbf{P}}$ for our set of covariance matrices $\mathbf{P}_i \in \mathcal{P}(n)$, the matrices are then projected to the tangent space $T_{\bar{P}}M$ centered at the geometric mean via the logarithmic mapping $\mathbf{T}_{\mathbf{P}_i} = \log_{\bar{P}}(\mathbf{P}_i)$. The data is now projected into a vector space that retains the geometric characteristics of the SPD matrices with minimal distortion, and can be integrated in dimensionality reduction and classification algorithms to perform

analysis.

5.4.2 *Principal Geodesic Analysis*

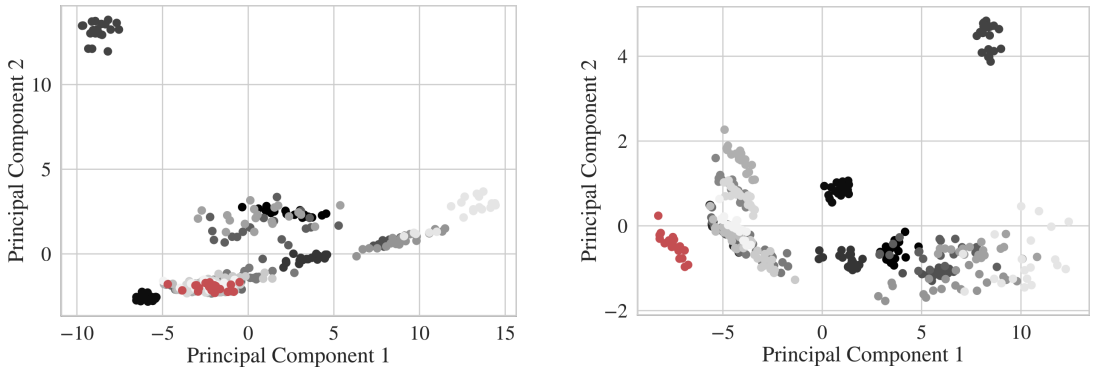
Mapping the process data covariance matrices to the tangent space provides an avenue for the application of common dimensionality reduction techniques. Here, we apply PCA to the matrices mapped to the tangent (vector) space. PCA applied on the tangent space of the SPD manifold is commonly known as Principal Geodesic Analysis (PGA), as it identifies the geodesics that capture the most variance in the data [292]. An example comparison of PGA versus PCA (directly on the covariance matrices) is presented in Figure 5.6. The simulations with no faults are colored in red and the faulty simulations are represented by different grayscale values. We can see that using only the first two components in PGA, we are able to perfectly separate the faulty and non-faulty simulations; on the other hand, when applying PCA directly on the covariance matrices we can see that there is significant overlap between the faulty and non-faulty simulations. The comparison of these projections demonstrates that capturing the geometry of the Riemannian manifold in the analysis of covariance matrices can improve the performance with minimal added complexity.

This improvement in separation of the data through the geometric approach is due in part to the congruence invariance of our defined metric on the Riemannian manifold. As previously stated, congruence invariance means that any $n \times n$ invertable matrix $\mathbf{X}$ applied to a set of covariance matrices $\mathbf{P}_i \in \mathcal{P}(n)$ does not impact the geodesic distance between the two matrices:

$$d_g(\mathbf{X}^T\mathbf{P}_i\mathbf{X}, \mathbf{X}^T\mathbf{P}_j\mathbf{X}) = d_g(\mathbf{P}_i, \mathbf{P}_j) \tag{5.19}$$

Operations such as re-scaling and normalization, which can be represented algebraically as invertable square matrices, have no impact on the geodesic distance between

the matrices [293]. Therefore, there is no need to select specific scaling or normalization strategies when applying our geometry based analysis of the data (e.g., PGA). However, this is not true when ignoring the data geometry, making methods such as PCA susceptible to the chosen framework (or lack of) for normalization/scaling. We perform no scaling prior to PCA in this case to ensure a direct comparison between PCA and PGA.



(a) Principal component analysis.

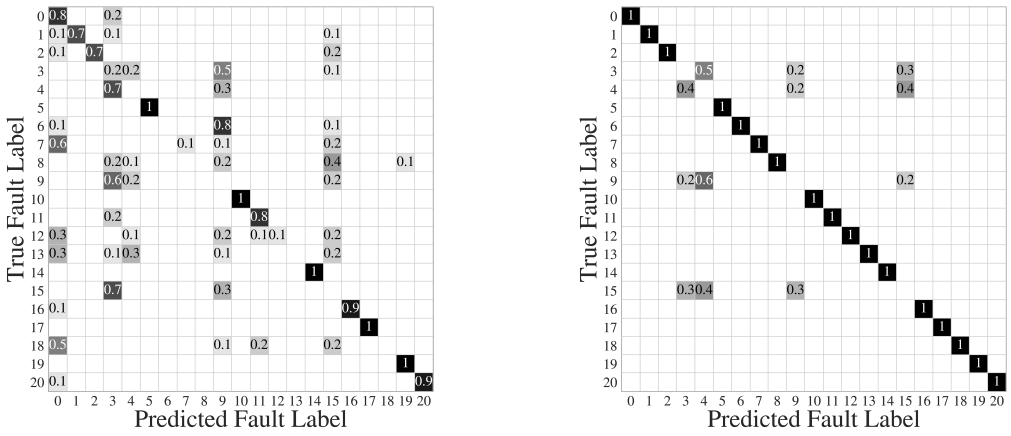
(b) Principal geodesic analysis.

Figure 5.6: Comparison of PCA applied to the raw covariance matrices and PCA applied to data mapped onto the tangent space (PGA). The red points represent the simulations where no fault is occurring and the grayscale points represent simulations with different faults. (a) PCA on the raw covariance matrices shows minimal separation in the data; faultless simulations are overlapped with faulty simulations. (b) PCA performed in the tangent space provides perfect separation between the faulty and faultless simulations, and also shows clustering of the faulty systems into separate groups. This demonstrates that simple considerations for the geometry of the data can yield improved results.

5.4.3 *Classification and Clustering Results*

PGA analysis of the covariance matrices also reveals that there is definite clustering of the data with respect to the different fault types within the TEP dataset. This suggests that classification of the different fault types can be done directly using the tangent space of the SPD manifold. To investigate this we use a simple linear (ridge) classifier. We compare the prediction accuracy of the linear classifier using coefficients of the tangent space projected matrices versus the coefficients of the non-transformed covariance matrices as input. In the analysis, we perform a simple train-test split of the data, where 30% of the data

is used for testing and 70% of the data is use for training. Figure 5.7 illustrates the dramatic increase in accuracy when the model incorporates geometric information, which is reflected in the normalized confusion matrices. Here, a value $x \in [0, 1]$ on the diagonal indicates an accuracy of $x * 100\%$ when classifying a particular fault. All values in the off diagonal (e.g., row i, column j, where $i \neq j$) represent the percentage of covariance matrices associated with fault i that have been incorrectly labeled as experiencing fault j. When the SPD manifold is accounted for via the tangent space projection, there is perfect classification of the data (with the exception of faults $3, 4, 9$ and 15). When the manifold geometry is ignored, there are few instances where high classification accuracy is achieved. The faults $3, 4, 9$ and 15 have been shown in prior work to be difficult to classify [291]. We also note that these faults are only misclassified within their group (are never classified as having no fault), which suggests that there is limited quantifiable difference in the covariance matrices for these faults. The inclusion of more information around these particular faults may correct this issue and further increase accuracy.



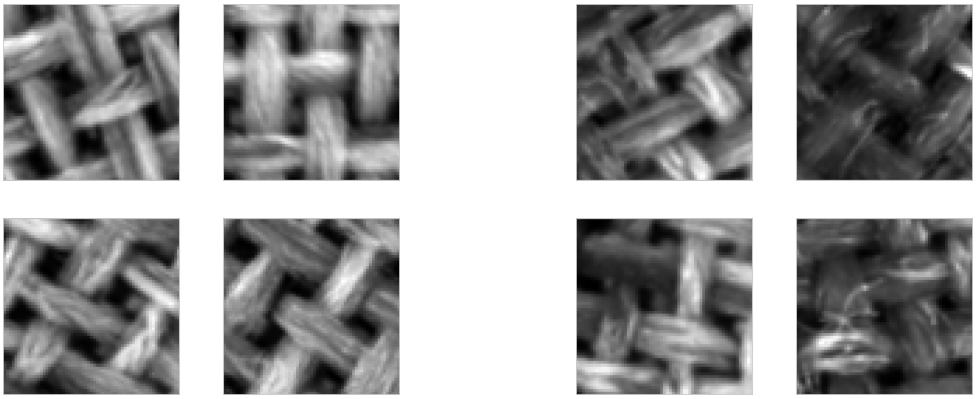
(a) Classification raw covariance matrices.

(b) Classification in tangent space ($T_{\bar{P}}M$).

Figure 5.7: Comparison of linear classification on the raw covariance matrices versus matrices mapped to the tangent space $T_{\bar{P}}M$. (a) Classification of the covariance matrices without regard for the data geometry results in poor classification accuracy. (b) Simple mapping of the data to the appropriate tangent space provides a dramatic improvement in classification accuracy.

5.5 Case Study - Image Analysis

Another important application of covariance matrices is in the analysis of images [294, 295, 296]. Here, we focus on an analysis of real images from textile manufacturing that contain non-defective and defective woven textiles taken. The images were obtained from the public MVTEC AD dataset [1]. Example images of non-defective and defective textiles are found in Figure 5.8. Our goal is to classify textiles using a linear classifier and Riemannian geometry.



(a) Normal textiles. (b) Defective textiles.

Figure 5.8: Example images from the woven textile dataset [1]. (a) Representative sample of woven textiles with no defects. (b) Representative sample of woven textiles that are considered defective.

5.5.1 *Data Pre-Processing*

Covariance matrices are useful data representations for images because they are invariant to translations and rotations [279]. These matrices can also be used to combine multiple image features that can be quantified through filters and kernel methods [297]. Here, we select nine image features, as shown in Figure 5.9. We implement both the Frangi and Hessian filters, which are designed to detect edges and fiber structures of the woven textures [298]. We apply these filters to the image; multiple transformed images that are smoothed through a Gaussian filter of varying strength are shown in Figure 5.9. We

use Gaussian filters to emphasize features of different scale within the image [299]. This yields nine total feature images (including the original image), which we use to construct a sample covariance matrix. We do this by taking each feature image (which we treat as a 64×64 matrix) and vectorizing each image $x_i \in \mathbb{R}^m$, where $i = 1, 2, ..., 9$ and $m = 4096$. Each feature image can therefore be represented as a realization of a multivariate random vector $\mathbf{X} = [x_1, x_2, ..., x_9]$. We use this observation to construct a sample covariance matrix $\mathbf{P} \in \mathcal{P}(n)$ (see Figure 5.9) where $n = 9$:

$$\mathbf{P} := \frac{1}{m-1}\mathbf{X}\mathbf{X}^T. \tag{5.20}$$

The covariance matrices are SPD, and can be directly integrated into our geometric analysis framework.

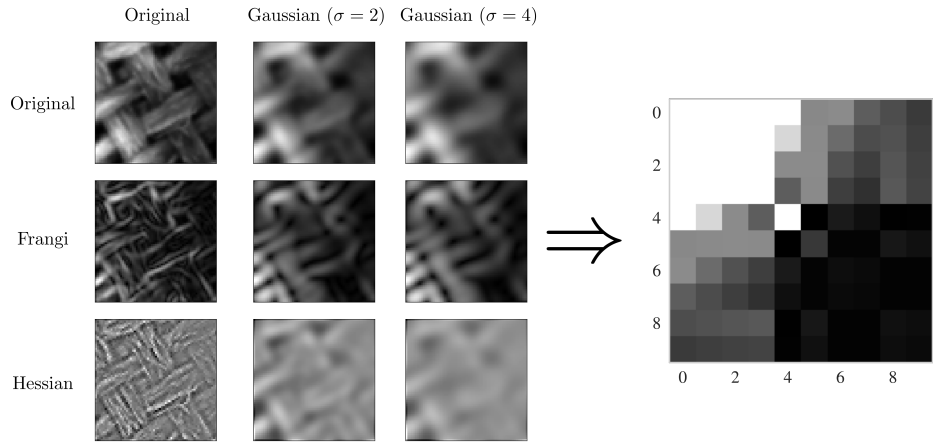
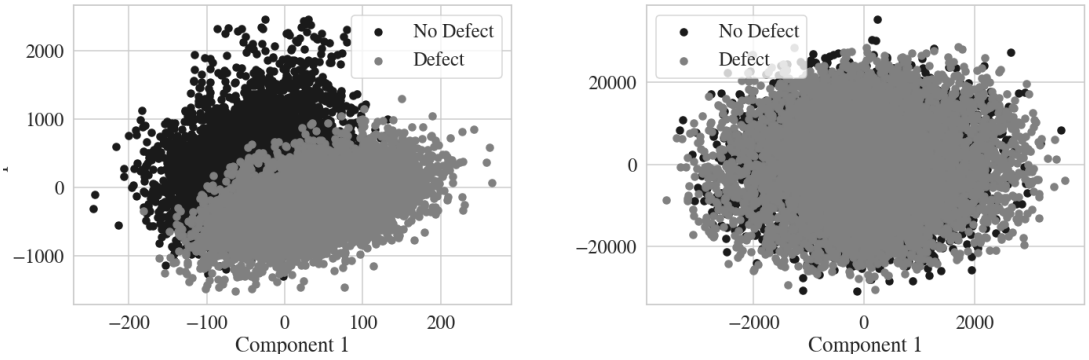


Figure 5.9: Workflow for the construction of an image covariance matrix. (left) Image filters and transformations emphasize specific characteristics of an image. Gaussian filters emphasize features of different scale within an image, and the Frangi and Hessian filters capture important fiber and edge features of an image. (right) The covariance between each image representation can be computed and used to form a covariance matrix. Importantly, the covariance matrix representation is invariant to transformations such as rotation and translation which are present in the textile images.

5.5.2 *Classification Results*

We first apply PGA to the image covariance matrices (see Figure 5.10). PGA reveals a distinct clustering and separation of the data with only two components. The PGA suggests that a simple classifier can be used to separate the defective and non-defective textile samples (after projecting to the tangent space). Thus, we construct a simple linear support vector machine classifier, which we train with 70% of the image data and test our trained model on the remaining 30% of the data. The covariance representation of the images provides a simple characterization of the data and its various features, while also imbuing the data with the inherent geometry of SPD matrices. This results in the trained model being able to separate the defective and non-defective samples in the testing data with 92% accuracy. We also compare our PGA analysis of the covariance matrix representation to PCA applied to the raw image data in Figure 5.10. The PCA analysis reveals almost no separation of the data, and results in poor classification accuracy when used as input to a linear SVM (50 % accuracy). This is likely due to sensitivity of the analysis on the raw images to rotations and translations of the textiles.



(a) PGA on covariance matrices.

(b) PCA on raw images.

Figure 5.10: (a) PGA of the covariance matrices derived from the set of woven textile images. The analysis reveals clustering of the data into groups representing the defective and non-defective textiles. A simple linear SVM classifier is able to separate the defective and non-defective textile images with 92% accuracy. (b) PCA analysis of the raw images; without the covariance matrix representation and considerations for the data geometry, we see there is almost no separation in the data.

5.6 Case Study - Atmospheric Data Analysis

We explore the benefits of using Riemannian geometry in the analysis of multi-site, multi-pollutant atmospheric monitoring data. Our approach uses covariance matrices to encode spatio-temporal variability and correlations of multiple pollutants at different sites and times. A key property of the covariance matrix representation is that it lies on a Riemannian manifold and one can exploit this property to facilitate dimensionality reduction, outlier detection, and spatial interpolation. We demonstrate the benefits of this approach by analyzing real atmospheric data collected from monitoring stations in Beijing, China.

Air pollution damages human health and impacts the environment (e.g., climate change) [300, 301, 302]. A key factor in developing pollution mitigation policies, technological solutions, and improving public awareness, is the monitoring and modeling of atmospheric pollutant behavior [303]. Air pollution is traditionally measured within distributed monitoring stations. These stations provide accurate measurements of multiple atmospheric pollutants (e.g., O_3, NO_x, $PM_{2.5}$) at high temporal resolution. Historically, air pollution research and policy has focused on the control of individual pollutants due to the complexities that arise in the analysis, modeling, and interpretation of multi-pollutant data [304]. However, the need for air quality management tools and methods that integrate multi-pollutant data has been recognized by government agencies, such as the Environmental Protection Agency (EPA) [304, 305, 306, 307]. Furthermore, the dynamic relationships between different pollutants encoded in multi-pollutant time series data can provide insight into the chemical and physical interactions between pollutants. For example, chemical interactions between NO_x and O_3 can be captured by observing temporal correlations between their atmospheric concentrations. For instance, a positive correlation between NO_x and O_3 is commonly present due to the formation of O_3 through the photolysis of NO_2 [308]. Whereas a negative correlation suggests the depression of O_3 concentration due to NO_x titration.

We apply the introduced methods in an analysis of multi-pollutant data taken from

34 air quality monitoring sites in Beijing, China [309]. The data is made available through the Beijing Municpal Monitoring Center (bjmemc.com.cn). For each site we record hourly concentrations of six atmospheric pollutants: CO, NO_2, O_3,PM_{10}, $PM_{2.5}$ and SO_2. For each of the 34 sites we obtain a stochastic multivariate time series. We can compute the pairwise covariance between each of the time series and construct a covariance matrix for each site. Our analysis is focused on quantifying and understanding the spatial and temporal behavior of these covariance matrices through dimensionality reduction and spatial interpolation. Figure 5.11 illustrates the basic workflow used in pre-processing the atmospheric data and the subsequent analysis on the Riemannian SPD manifold.

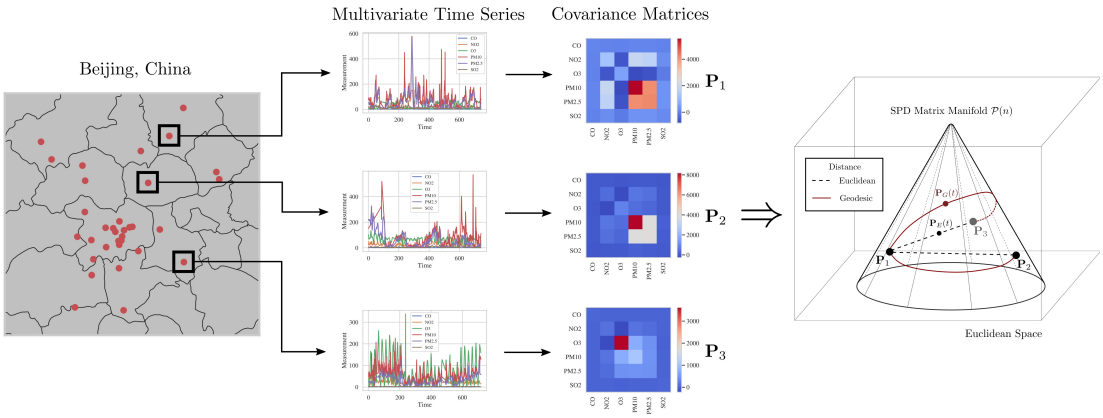
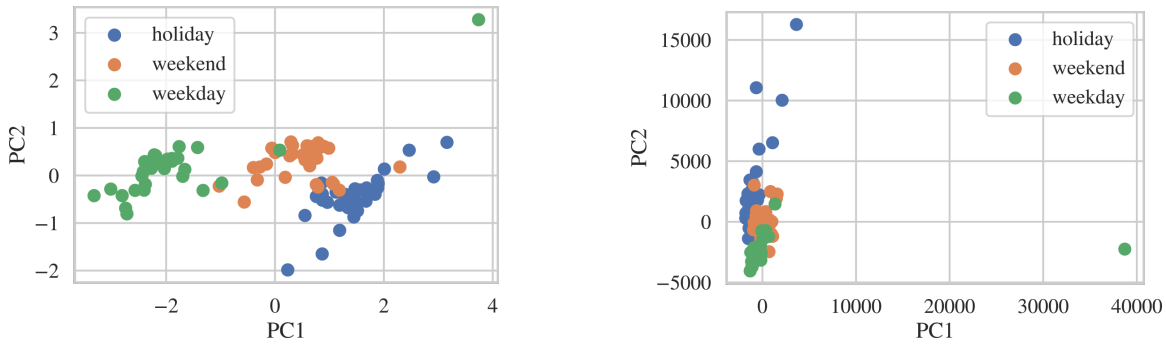


Figure 5.11: Illustration of the workflow used in the pre-processing and analysis of atmospheric data measured at multiple locations with Beijing, China. Multivariate measurements of six atmospheric pollutants: CO, NO_2, O_3,PM_{10}, $PM_{2.5}$ and SO_2 are taken over time at each monitoring site. These multivariate time series can be represented as covariance matrices, which are SPD for this dataset. The covariance matrices can then be analyzed through Riemannian geometry and geodesic distances.

5.6.1 *Dimensionality Reduction*

We first analyse the multi-pollutant data collected during weekdays, weekends, and holidays for each of the 34 sites. For each site we obtain 3 covariance matrices, one representing dynamics observed during the weekday, one during the weekend, and a final during holidays. We can then perform the proposed Riemannian geometric method for the data



(a) Geodesic PCA

(b) Euclidean PCA

Figure 5.12: A comparison of dimensionality reduction methods for the multi-pollutant covariance matrices derived from holiday, weekend, and weekday data across the 34 sites in Beijing, China. (a) Principal component analysis of covariance matrices that have been mapped to the tangent space at the geometric mean $\mathcal{T}_{\bar{\mathbf{P}}}\mathcal{P}$. (b) Principal component analysis where the covariance matrices have been assumed to be governed by Euclidean geometry. The geodesic PCA method is able to capture distinct separation between pollutant dynamics on weekdays, weekends, and holidays. The Euclidean PCA method is dominated by outliers and is impacted by the scale of the variables, and does not show clear separation between the three sample groups.

by first identifying the geometric mean of the matrices $\bar{\mathbf{P}}$, and then mapping the data from the Riemannian manifold $\mathcal{P}(n)$ to the tangent space at the geometric mean $\mathcal{T}_{\bar{\mathbf{P}}}\mathcal{P}$ through the logarithmic map $\log_{\bar{\mathbf{P}}}(\cdot)$. Our matrices are now in a (linear) vector space, and can be analysed through principal component analysis (PCA). Here, each mapped covariance matrix $\mathbf{T}_{\mathbf{P}_i} \in \mathcal{T}_{\bar{\mathbf{P}}}\mathcal{P}$ can be can be vectorized $T_{P_i} := \text{vec}(\mathbf{T}_{\mathbf{P}_i}) \in \mathbb{R}^{n^2}$, where $n^2 = 36$. We can then construct a matrix $\mathbf{M} = \left[T_{P_1}^T, T_{P_2}^T, ..., T_{P_j}^T\right]^T \in \mathbb{R}^{j \times 36}$, where $j = 32 * 3$. The matrix $\mathbf{M}$ contains the 3 transformed covariance matrices for each of the 32 sites. We then perform a singular value decomposition of the matrix $\mathbf{M}$ and project the data onto the leading eigenvectors. The results of this analysis for the first two leading eigenvectors is found in Figure 5.12a. We compare these results to a similar analysis that assumes the covariance matrices are governed by Euclidean, rather than Riemannian, geometry. In this second approach we perform the same type of the analysis, but do not project the covariance matrices to the tangent space at the geometric mean. The results of this Euclidean based approach are found in Figure 5.12b.

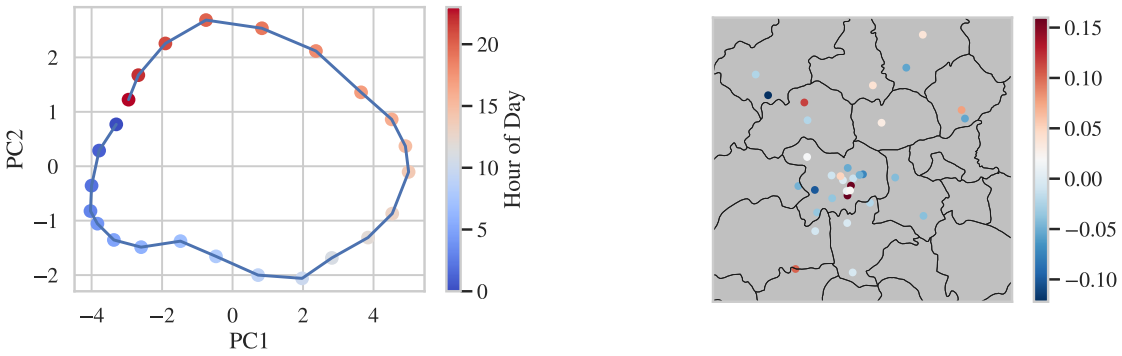
The output of PCA that considers the Riemannian geometry of the covariance matrices

provides a much clearer result that demonstrates clustering of the behavior of the sites into weekday, weekend, and holiday groupings. The Euclidean method does not capture this same information and is distorted by the presence of potential outliers in the data. One of the reasons for this distortion is the need for normalization or scaling of the data prior to PCA. This is not a challenge for the Riemannian approach due to a powerful property known as *congruence invariance*. Congruence invariance means that any $n \times n$ invertable matrix $\mathbf{X}$ applied to a set of covariance matrices $\mathbf{P}_i \in \mathcal{P}(n)$ does not impact the geodesic distance between the two matrices:

$$d_g(\mathbf{X}^T\mathbf{P}_i\mathbf{X}, \mathbf{X}^T\mathbf{P}_j\mathbf{X}) = d_g(\mathbf{P}_i, \mathbf{P}_j) \tag{5.21}$$

Operations such as re-scaling and normalization, which can be represented algebraically as invertable square matrices, have no impact on the geodesic distance between the matrices [293]. Therefore, there is no need to select specific scaling or normalization strategies when applying our geometry based analysis of the data, it is done naturally through the geometry of the manifold. However, this is not true when ignoring the data geometry, making Euclidean methods susceptible to the chosen framework (or lack of) for normalization/scaling. We perform no scaling prior to PCA in this case to ensure a direct comparison between the Riemannian geometric and Euclidean methods.

We illustrate another application of dimensionality reduction in the analysis of multi-site, rather than multi-pollutant, dynamics. Here, we measure the dynamics of NO_2 at each site over the entire year. For a given site we obtain a univariate stochastic time series $x_i \in \mathbb{R}^m$, where $i = 1, 2, ..., 34$ and $m = 24 * 365$ because NO_2 is measured in hourly intervals. For each site i we split the time series data into subsets $x_i^h \in \mathbb{R}^p$ where $p = 365$ and $h = 1, 2, ..., 24$. We take each subset and form a multi-variate time series matrix $\mathbf{X}^h = [x_1^h, x_2^h, ..., x_34^h]$ for each hour of the day $h = 1, 2, ..., 24$. We can then construct a



(a) Geodesic PCA

(b) Coefficients of the leading eigenvector

Figure 5.13: An analysis of the behavior of NO_2 across multiple sites during each hour of the day. (a) Illustration of the clearly cyclic behavior of NO_2 dynamics observed during each hour of the day across the different Beijing sites. (b) An analysis of the coefficients of the first eigenvector of PCA. We find that certain sites have a large influence on the behavior of NO_2 dynamics during the day (positive coefficients - red color) and sites that have a larger influence during the night (negative coefficients - blue color)

covariance matrix from this data as:

$$\mathbf{P}^h := \frac{1}{365-1}\mathbf{X}\mathbf{X}^T. \tag{5.22}$$

which results in a total of 24 covariance matrices of shape $\mathbf{P}_i^h \in \mathbb{R}^{n^2}$, where $n = 34$. We follow the same procedure as the previous example. The results of the Riemannian Geometric analysis are found in Figure 5.13. Using the Riemannian geometric approach, we see a data structure that indicates a clear cyclic behavior in NO_2 dynamics throughout the day. We can also understand what sites are impacting the dynamics of NO_2 by observing the values of the eigenvectors associated with each principal component. We visualize this in Figure 5.13 where we color each of the 34 sites with the coefficients of the leading eigenvector associated with the variance of each site (i.e. the diagonal values of each covariance matrix). From this analysis we can see which sites have a larger influence on the behavior of NO_2 dynamics during the day (positive coefficients - red color) and those that have a larger influence during the night (negative coefficients - blue color).

5.6.2 *Spatial Interpolation*

Consideration for the Riemannian geometry of covariance matrices is crucial when performing spatial interpolation of covariance matrix values [310]. As shown previously in Figure 5.11, interpolation between matrices with an assumption of Euclidean geometry can result in matrices that do not lie on the covariance matrix manifold $\mathcal{P}(n)$. Whereas interpolation through geodesics provides assurance that the resulting interpolated matrix will lie on the Riemannian manifold [310]. Euclidean interpolation of covariance matrices can also introduce a *swelling effect* on the interpolated matrices [264, 10]. The swelling effect causes an increase in the generalized variance (i.e., determinant) of the interpolated covariance matrices. This introduces a spurious increase in the variance of the atmospheric pollutant data dynamics creating results that are not physically consistent. An example of this effect is shown in Figure 5.14. Here, we compare the generalized variance (i.e., determinant) of matrices from Euclidean and geodesic interpolation between two sample covariance matrices from our dataset. From Figure 5.14 we see the swelling of the data variance with the Euclidean interpolation method, given by the function:

$$\mathbf{P}_E(t) := \mathbf{P}_1(1-t) + \mathbf{P}_2(t) \tag{5.23}$$

where $t \in [0,1]$ and $P_E(0) = \mathbf{P}_i$, $P_E(1) = \mathbf{P}_j$. At $t = 0.85$ the generalized variance of the interpolated matrix is double that of either the sample covariance matrices, falsely indicating the presence of a pollution source or new pollutant interactions. This is not the case with the geodesic interpolation which reflects a natural evolution of the generalized variance between the two samples, given by the function:

$$\mathbf{P}_G(t) := \mathbf{P}_1^{1/2} \left(\mathbf{P}_1^{-1/2} \mathbf{P}_2 \mathbf{P}_1^{-1/2} \right)^t \mathbf{P}_1^{1/2} \tag{5.24}$$

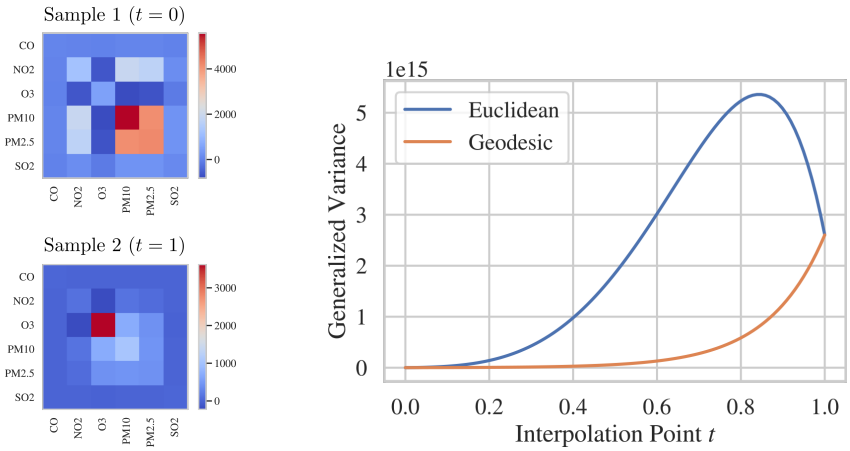


Figure 5.14: Visualization of the *swelling effect*. We compare the Euclidean and geodesic interpolations between covariance matrices labeled Sample 1 and Sample 2. If Euclidean geometry is assumed, the interpolated matrices have inflated generalized variance (i.e., determinants) that is almost double that of either Sample 1 or Sample 2. In a spatial interpolation method (e.g., Kriging) this will result in a false increase in variance, potentially indicating sources of pollution that are non-existent. However, If interpolation is done along a geodesic there is no swelling and the generalized variance evolves in a way that is natural to the data.

where $t \in [0,1]$, $P_G(0) = \mathbf{P}_1$, $P_G(1) = \mathbf{P}_2$, and $\mathbf{P}_i = \mathbf{P}_i^{1/2}\mathbf{P}_i^{1/2}$.

Another aspect of spatial interpolation methods is the modelling of spatial dependence between observed data points. This is often modeled through the use of the empirical variogram [311]. Given a set of $k \in \mathbb{Z}_+$ sample covariance matrices $\mathbf{P}_i \in \mathcal{P}(n)$ measured at k spatial locations $s_i \in \mathbb{R}^2$ we compute the squared distance between each sample covariance matrix and the spatial lag distance between each location $||s_i - s_j||_2$. This information is spatially binned and averaged. For the geodesic method we compute the empirical variogram as:

$$\hat{\gamma}(h \pm \delta)_G := \frac{1}{2|N(h \pm \delta)|} \sum_{(s_i,s_j) \in N(h \pm \delta)} ||\log(\mathbf{P}_i) - \log(\mathbf{P}_j)||_F \tag{5.25}$$

where $h, \delta \in \mathbb{R}$ represents the spatial lag bin center and width, $N(h \pm \delta) := \{(s_i, s_j) : ||s_i - s_j||_2 \in h \pm \delta\}$, and $|N(h \pm \delta)| \in \mathbb{Z}$ represents number of elements contained in

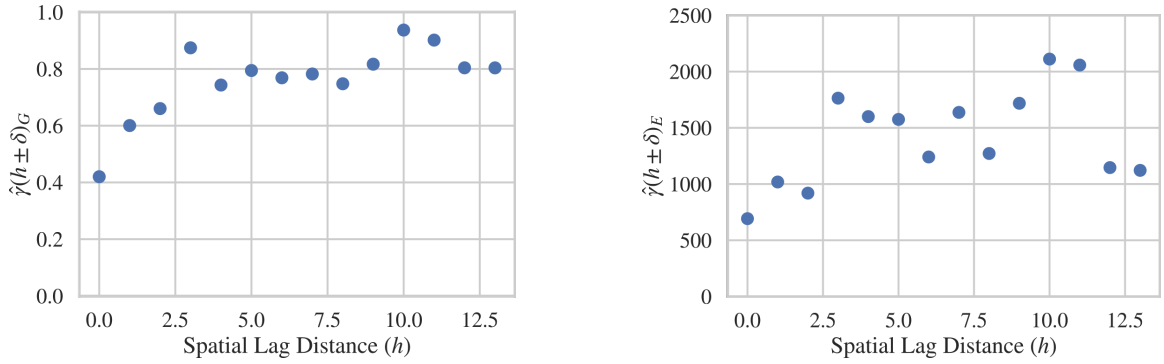


Figure 5.15: Empirical variograms constructed using geodesic and Euclidean distances during the weekdays of January to March. The variogram constructed using geodesic distances $\hat{\gamma}(h \pm \delta)_G$ reveals an exponential behavior, whereas the Euclidean variogram shows almost no structure. The incorporation of the manifold geometry into the analysis of spatial depence can reveal information that is missed if Euclidean geometry is assumed.

$N(h \pm \delta)$. For the Euclidean method the empirical variogram is given as:

$$\hat{\gamma}(h \pm \delta)_E := \frac{1}{2|N(h \pm \delta)|} \sum_{(s_i, s_j) \in N(h \pm \delta)} ||\mathbf{P}_i - \mathbf{P}_j||_F \tag{5.26}$$

We compare the empirical variogram for the Euclidean and geodesic methods constructed from the Beijing data observed during weekdays in the months of January to March. The empirical variograms $\hat{\gamma}(h \pm \delta)_G, \hat{\gamma}(h \pm \delta)_E$ are found in Figure 5.15. The variogram constructed with the geodesic distance reveals a clear exponential behavior, whereas the variogram constructed with a Euclidean distance does not show a clear behavior that fits known variogram models [312]. The incorporation of the Riemannian manifold geometry can reveal spatial relationships between the covariance matrices that are missed when Euclidean geometry is assumed.

Part III

DATA DRIVEN METHODS FOR PATTERN AND STRUCTURE ANALYSIS

Chapter 6

CONVOLUTIONAL NEURAL NETWORKS & LIQUID CRYSTAL SENSORS

The contents of this chapter are published in [54]

6.1 Introduction

This chapter focuses on the application of a data driven method, the convolutional neural network, for pattern and structure quantification in data. We apply these methods in the analysis of liquid crystal sensor data. Liquid crystals (LCs) provide a versatile platform for sensing of air contaminants (chemical sensing) [255, 313] and for sensing of heat transfer and shear stress (mechanical sensing) [314]. In the context of chemical sensing, LC sensors can be designed to change their orientational ordering and optical birefringence upon exposure of the LC to a certain targeted chemical environment. For instance, an LC sensor can be prepared by supporting a thin LC film (thickness of micrometers) on a chemically functionalized surface. Typically, the molecules within the LC film (the mesogen) bind to the surface and assume a homeotropic (perpendicular) orientation that provides an initial optical signal. Subsequent exposure of the LC film to an analyte leads to diffusive transport of the analyte through the LC phase and displacement of the mesogen at the surface, triggering rich space-time optical responses (Figure 6.1). The response

time of the spatial-average brightness of the optical signal has been shown to be strongly correlated to the differential binding energy between the analyte and mesogen to the surface. The physicochemical principles of LC chemical sensors are explained in detail elsewhere[255].

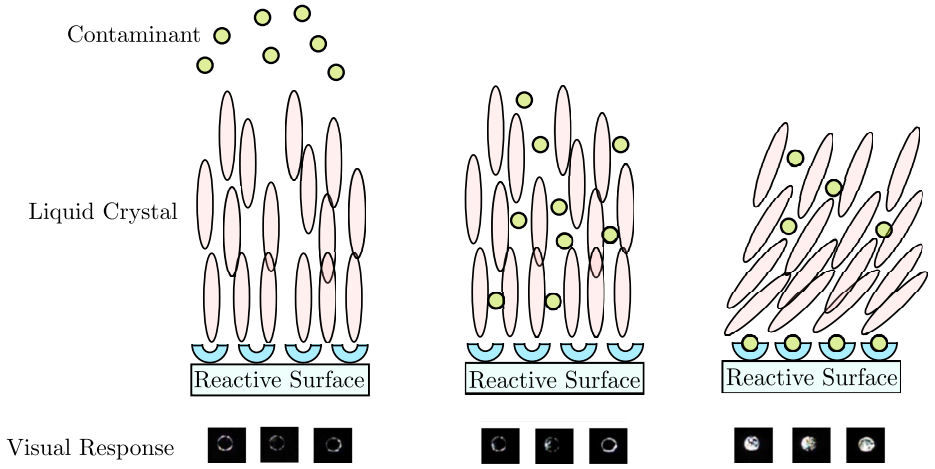


Figure 6.1: Working design principles of a liquid crystal chemical sensor.

A primary challenge for the development of LC sensors (as in other sensing technologies) is their potential sensitivity to *interfering* chemical species. For instance, LC sensors designed for detection of dimethyl methylphosphonate (DMMP), $CH_3PO(OCH_3)_2$, might exhibit similar optical responses when exposed to humid nitrogen [315]. Moreover, LC responses can also be slow, as these require diffusion of the air contaminant through the LC film and displacement of the mesogen at the surface. Sluggish responses limit the applicability of the LC sensor (e.g., when detecting highly toxic chemicals). These issues are illustrated in the experimental responses shown in Figure 6.2. Although the selectivity of LC sensors can be optimized by chemical design to largely eliminate the effects of humidity, a natural step is to determine whether or not one can unravel *hidden patterns* in the optical responses that can help discern between chemical species. The identification of such patterns can also help reduce detection times and simplify the design of LC sensors.

Machine learning techniques are actively being used for pattern recognition in diverse branches of science and engineering. Specifically, convolutional neural networks

(CNNs) have been used for brain tumor and skin lesion classification [316, 317]. The goal of a classification strategy is to separate different images by using numerical features (descriptors) that characterize such images. Features are projections of the original image into an information space that seek to best summarize/describe an image (features are characteristic patterns of the image). Certain features can be strongly correlated to physical phenomena that govern a system; for instance, image features such as textures are often correlated to structural properties of materials [318]. Interestingly, informative features that capture multi-scale spatial patterns can be extracted from CNNs that have been pre-trained using generic images (that are not directly related with the application at hand). Such features can then be used in an external classification engine such as a fully connected network, logistic regression, or support vector machine. For instance, in the work of [317], the pre-trained CNN `Alexnet` [319] was used to classify skin lesions. In the work of [320], textures extracted from the pre-trained CNN `VGG16` [321] are used to predict material properties. The principle behind the exploitation of pre-trained CNNs is known as *transfer learning* [322].

Cao and coworkers recently used `Alexnet` to characterize optical LC responses (as grayscale images) and demonstrated that spatial features of the LC response can be used to discern the chemical environment [323]. Specifically, the authors demonstrated that spatial features extracted from the deep layers of `AlexNet` can be used to achieve classification accuracy levels of 99%. Notably, they also observed that snapshots taken *within three seconds* of exposing the LC are sufficient to classify the environment (either DMMP or humid nitrogen). Unfortunately, reaching such high levels of accuracy required an extremely large number of features (on the order of thousands), which resulted in computational issues and clouded the physical interpretability of the dominant features. In particular, features extracted from deep CNN layers, while informative, are difficult to interpret.

In this chapter, we extend the results of Cao and coworkers by analyzing LC response features extracted from `VGG16`, which is a CNN that embeds a smaller set of convolutional

filters than `Alexnet`. Moreover, in the current study, we use RGB (color) images directly (in previous work grayscale images were used). Our findings demonstrate that features extracted from the first and second convolutional layers of `VGG16` allow for a *perfect* classification accuracy for the same dataset studied by Cao and co-workers, while reducing the number of features to approximately one hundred. We demonstrate that the number of features can be further reduced to ten via recursive feature elimination with minimal losses in sensor accuracy. This feature reduction procedure reveals that complex *spatial color patterns* are developed within seconds in the LC response, which leads us to hypothesize that differences in spontaneous fluctuations in LC tilt orientation (angle) play a key role in sensor selectivity and responsiveness. Our analysis also reveals that *hue distributions* provide an effective set of features to characterize LC responses.

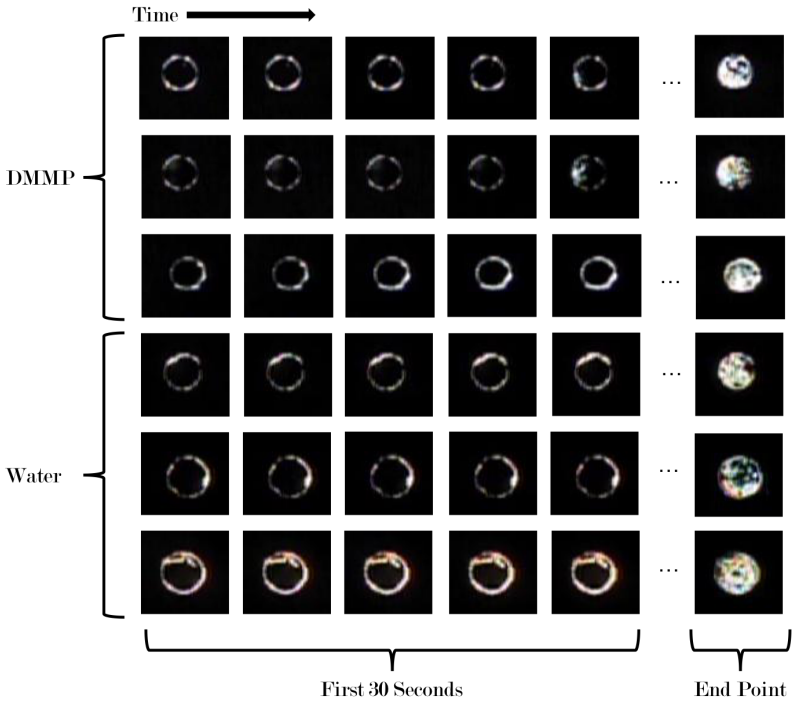


Figure 6.2: Optical responses of liquid crystals under gaseous N_2-water (30% relative humidity) and N_2-DMMP (10 PPM) environments. LCs were deposited into microwells with a diameter of 3mm to enable high-throughput data collection. LC responses were recorded at room temprature.

6.2 Methods

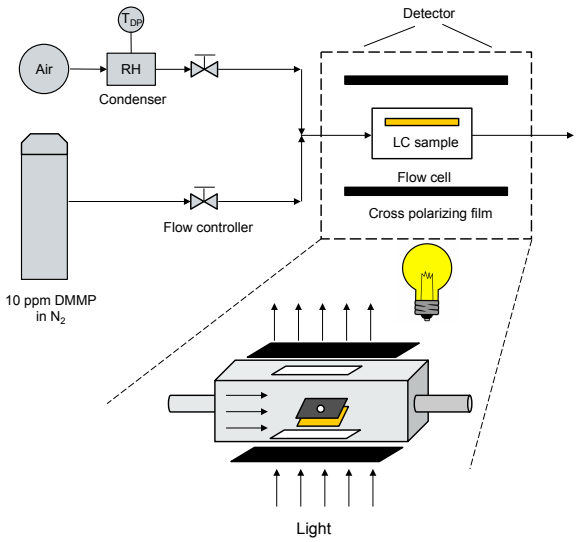


Figure 6.3: Sketch of experimental system used for collecting LC response data.

6.2.1 *Experimental Methods*

We recorded six videos that show the response of LCs to DMMP-N_2 at 10 ppm (the length of each video ranges from 4 - 13 minutes) and six videos that show the response of LCs to water-N_2 (the length of each video ranges from 7 - 30 minutes). The experimental system is sketched in Figure 6.3. Each video tracks the dynamic evolution of multiple independent micro-wells (the total number of micro-wells recorded was 391). We captured a frame (micrograph) from each video every 3.3 seconds. We split each frame into several images, each containing a single micro-well at a specific time. The total number of micro-well images (snaptshots) generated was 75,081 and each image is resized to 60 x 60 pixels (see Figure 6.2 for some example micrographs). The experimental procedure followed to obtain the LC response data was the following:

- *Formation of thin films of LC supported on metal-salt-decorated surfaces*: 50 μL of 10mM aluminum perchlorate salts in ethanolic solution were deposited by spin-coating

(3000 rpm for 30s) onto the glass surfaces at the bottom of the polymeric micro-wells. Next, 2 μL of 5CB (4-cyano-4'-pentylbiphenyl) were deposited into the polymeric micro-wells [324] with a depth of 5 μm using a micropipette. The excess LC was removed from the array by wicking into a microcapillary.

- *Optical characterization of LC films*: The optical appearance of the LC was characterized by using an Olympus BX-60 polarizing light microscope in transmission mode (Olympus, Japan). Conoscopic imaging of the LC films was performed by inserting a Bertran lens into the optical path of a polarized-light microscope to confrim the homeotropic orientation [325].

- *Ordering transitions induced by DMMP and humid N_2*: The LC-filled micro-wells were exposed to a stream of dry N_2 containing DMMP (10 ppmv) within a flow cell [326] with glass windows that permitted characterization of the optical appearance of the LC using a polarized optical microscope. The gas containing DMMP was delivered to the flow cell at 300 mL/min by using a rotameter (Aalborg Instruments and Control, Orangeburg, NY). For experiments performed to evaluate the response of the LCs to water vapor, nitrogen containing 30% relative humidity was delivered to the flow cell at 300 mL/min with the same rotameter (we call these mixture N_2-Water). The optical appearance of the LC film was recorded using an Olympus camera (Olympus C2040Zoom, Melville, NY) and WinTV software (Hauppauge, NY).

6.2.2 *Computational Methods*

In this section, we summarize the machine learning methods used to analyze optical micrographs of LCs. We focus on classifying whether an LC sensor has been exposed to DMMP or humid air (we call this water, for convenience). In other words, our framework is focused on binary classification. We use the same dataset reported by Cao and coworkers [323] but focus on patterns developed within the first 30 seconds of the LC response.

Details regarding the experimental system and data preparation methods can be found in [323].

In summary, the dataset analyzed was obtained from six videos that show the response of LCs to a gaseous stream of N_2 containing 10 ppm DMMP and six videos that show the response of LCs to a gaseous stream of N_2 containing 30% relative humidity (both at room temperature). Each video tracks the dynamic evolution of multiple independent microwells (the total number of microwells recorded was 391). We captured a frame (a micrograph) from each video every 3.3 seconds. We split each frame into several images, each containing a single microwell at a specific time. The total number of microwell snapshots generated was 75,081 (the dataset analyzed is extensive).

Examples of snapshot sequences collected during the microwell responses are presented in Figure 6.2. Our machine learning analysis treats snapshots as time-independent; this type of analysis is more challenging than analysis of time-dependent sequences and it is more desirable from a sensor design perspective because we want to detect a contaminant by ignoring its response history. Specifically, our aim is to show that machine learning techniques can detect a contaminant by just looking at a snapshot at any time (by exploiting the spatial pattern of the response).

Classification

In an ideal setting in which an image can be characterized using highly informative features, classification can be performed using a *linear* hyperplane, where the dimension of the hyperplane is equal to the number of features minus one. For instance, if an image can be characterized using two features, the hyperplane will be a line. This hyperplane provides a decision boundary under which every image on one side is considered a member of one class and every image on the opposite side is considered a member of the contrasting class. In most settings, these classes are provided a numerical label of +1 or -1. In our setting, water is considered the +1 class and DMMP is considered the -1 class.

The classification engine used for the LC dataset is a linear support vector machine

(LSVM), which is trained using image features extracted from the CNN `VGG16`. An illustration of the LSVM method is presented in Figure 6.4. LSVM is a classification method that builds a linear decision boundary between observations. This is done by finding a hyperplane that maximizes the margin between the set of closest images to the hyperplane (known as the support vectors) and the hyperplane itself. The hyperplane is a weighted linear combination of all the CNN features representing each observation. The magnitude of each feature weight represents its relative importance (a proxy for information content); in other words, a feature that is highly informative (explains differences in the images well) will tend to have a large weight while a non-informative feature will tend to have a small weight. The images that are closest to the margin are the most difficult to classify (difficult to distinguish) while the ones that are farthest away from the margin can be easily classified (easy to distinguish). The support vectors are the images that define the separation boundary.

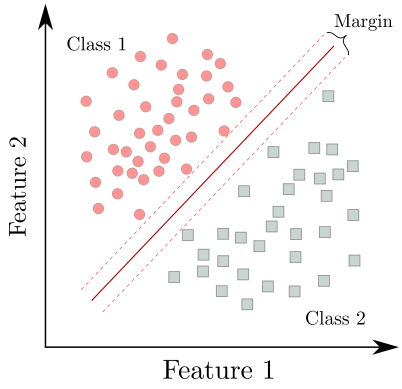


Figure 6.4: Illustration of a linear support vector machine.

The identification of relevant features can be achieved by penalizing the l_1 norm of the weights of the LSVM classifier. This penalization term seeks to *sparsify* the weight vector (have few nonzero entries). Consequently, a penalized LSVM classifier is tasked with not only finding a separating hyperplane that best classifies the images but is also required to do so with a minimal number of features (this set of features are interpreted as the ones

that provide most information). The mathematical formulation of the LSVM problem is:

$$\min_{w_0,w} \sum_{i=1}^{n} \left[1 - y_i \left(w_0 + \sum_{j=1}^{q} w_j x_i \right) \right] + \lambda ||w||_1 \tag{6.1}$$

Here, n is the number of images (observations), m is the image feature dimension, $w \in \mathbb{R}^m$ represents the feature weights, $x_i \in \mathbb{R}^m$ are the features of observation i, $y_i \in \{-1, +1\}$ represents the label for observation i, and $\lambda \in \mathbb{R}_+$ is a hyperparameter for the penalization of the l_1 norm [327]. The solution of problem (6.1) is often called the training phase and the images used for its solution are often called the training set. Once the classifier has been trained, one uses the optimal hyperplane weights w^* identified in the training phase to predict the label of a new image that is not in the original training set. The new images are known as the test (validation) dataset. This process is repeated five times, each time with a new training and validation set (five-fold validation). This allows for a robust testing of the effectiveness of the classification model on the entire dataset.

Feature Extraction

In order to train the LSVM classifier, we first need to identify features that best explain each image. Cao and coworkers previously used `Alexnet` to conduct feature extraction from LC micrographs. `Alexnet` is a CNN that has been pre-trained using the `Imagenet` database [328]. This database is a collection of millions of images that contains over 1,000 categories. The original goal of `Alexnet` was to work as a classifier [319]; however, one can also use features extracted by `Alexnet` to train an external classifier such as an LSVM (transfer learning). This approach avoids re-training the CNN, which can be highly computationally expensive. Cao and co-workers demonstrated that the transfer learning approach can be effectively used to classify optical micrograps of LCs using LSVM; their analysis, however, used over 5,000 features to explain each micrograph. Moreover, in their approach, the micrographs were transformed into grayscale images; as we will see, this transformation leads to *significant losses of information* and hides physical LC behavior.

In this work, we consider a different pre-trained CNN that we hypothesize may be better suited to our given application. We sought to merge our understanding of working principles of CNNs with our knowledge of physical behavior of LCs. A primary consideration is the length scale that characterizes the LC reponse. Specifically, we know that nematic ordering and interfacial interactions within LCs give rise to optical patterns of orientation on the micrometer-scale; because of this, the patterns created by the LCs need to be captured with a small observation lense. Moreover, the interference colors created by the LCs are an indicator of their tilt angles (orientation) and thus the CNN selected should be trained using RGB images directly (as opposed to grayscale images). A CNN that fits these requirements is `VGG16`, which has been pre-trained by the visual geometry group at Oxford [321]. The `VGG16` CNN has been trained on the `Imagenet` database. The structure and optimal weight values for the trained `VGG16` network are freely available through the `Keras` software and are what is used during feature extraction [329]. `VGG16` utilizes the smallest possible convolutional filter size (3x3), which should be best for capturing small-scale structural patterns in images. Moreover, `VGG16` is a much shallower CNN than `AlexNet` and thus its features are easier to interpret. A simplified representation of the `VGG16` architecture is shown in Figure 6.5.

The basic idea behind feature extraction using a CNN such as `VGG16` is to reduce a given input image into a small set of numerical values that can be used to best summarize and classify the image. Each image is represented by a set of input channels, each expressed as a two-dimensional pixel field (a matrix). The input channels are typically the red, green, and blue (RGB) channels of an image. Image reduction is performed through a sequence of *matrix convolution* operations in which spatial information is extracted from the image using filters (matrices with specific patterns). Subsequent convolutions compress this image to the point where a decision (e.g., classification or regression) can be made, which is represented by the decision block in Figure 6.5. More details on this procedure can be found in [321].

A convolution is a manipulation of an image matrix with a filter matrix. Specifically,

a convolution is the process of finding the extent to which a given pattern (defined by a convolutional filter) is present within a neighborhood of an image (and repeating the process by spanning all neighborhoods of the image). In other words, the convolution seeks to identify to what extent a specific spatial morphology and/or correlation structure (defined by the filter matrix) is present in the image. An example of applying a convolution filter to an image is illustrated in Figure 6.7. Convolutional filters provide a quantifiable approach for identifying multiple spatial structures within a given neighborhood (different filters identify different types of patterns). The larger the value of the filter output, the more similar the given neighborhood is to the pattern that the filter is attempting to find. Optimal filter matrices that best classify a set of images (optimal patterns) can be found by training the CNN directly on the dataset. Specifically, the training process aims to compute the entries of the filter matrices that best separate the images). Matrices are high-dimensional objects and, as such, training a CNN involves a highly computationally expensive procedure. Filters extracted from training over a given set, however, can also be *reused* to seek for similar patterns in a different dataset. In other words, pre-trained filters (pre-identified spatial patterns) can be used on a different image set with the sole purpose of obtaining feature information. While the filters are not optimal for the new dataset, this procedure is often effective at detecting general patterns in images and the obtained feature information can be used in an external classifier such as LSVM.

In the example provided in Figure 6.7, we see that the convolutional filter is seeking to match the neighborhood to a cross pattern and thus the top neighborhood has a higher output (perfect match) than the lower output (imperfect match). In the CNN, the matching is applied to every pixel in the image, and thus there is a convolution value for every pixel neighborhood (resulting in a matrix of filter outputs). In our approach, the entire set of outputs for each filter are averaged and utilized as a feature for the LSVM classification. This is done in order to ensure that the features are spatially-invariant. Spatial invariance allows for images that are not of a uniform size or perfectly centered to be treated as similar as possible. This practice also forces the classifier to seek meaningful and general-

izable features associated with the sensors rather than arbitrary features based upon the location of the sensor in the given frame (thus leading to more consistency in the results).

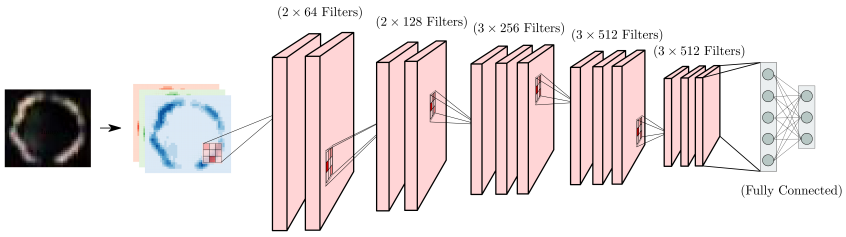


Figure 6.5: Schematic of `VGG16` architecture.

The next decision to be made is what feature information should be extracted from `VGG16`. The `VGG16` network has been trained to classify highly complex images and the deepest layers have been carefully tuned to differentiate such images. The early layers of the network, however, are the most general and are easier to interpret (they are less evolved). Accordingly, in our approach, we use the outputs of the first and second convolutional blocks to inform features for LSVM classification. Feature extraction is conducted by feeding a given image into `VGG16`. We modified the network so that the only output it provides is information extracted from the first and second convolutional blocks. This information is extracted in the form of convolutional filter activations via convolutions.

In summary, the CNN used here provides a number of features equal to the number of convolutional filters used for each image. In our case, the total number of features reaches 192 (64 for the first block and 128 for the second block). Note that the number of features increases with the depth of the layer, which precisely reinforces our desire to focus on the first layers. A visual representation of this process for the first and second convolutional blocks is provided in Figure 6.6. Feature extraction and network modification were performed using `Keras` [329] and `Tensorflow` [330]. The `VGG16` network and trained weights are made available in the `Keras` software, which allows for easy manipulation of the `VGG16` network so that this process may be completed for any number of image sets. With the extraction of the features from the first two layers of the `VGG16` network, analysis of the classification may be conducted.

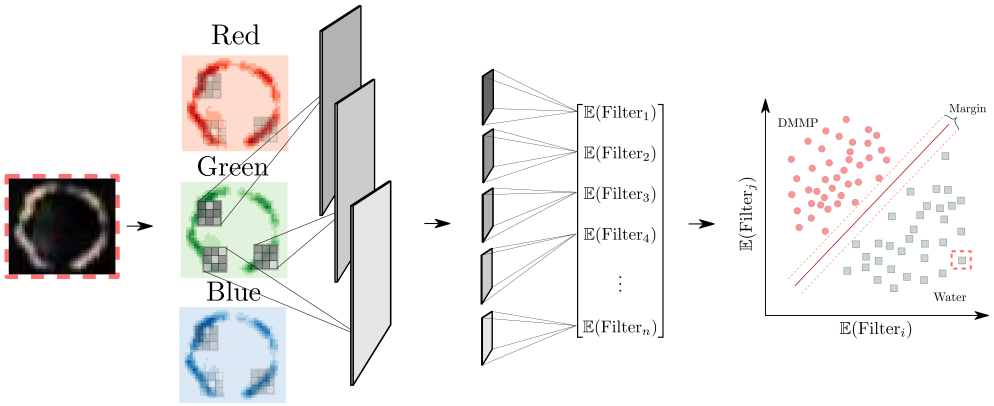


Figure 6.6: Schematic of feature extraction and classification framework ($\mathbb{E}(\cdot)$ represents spatial average).

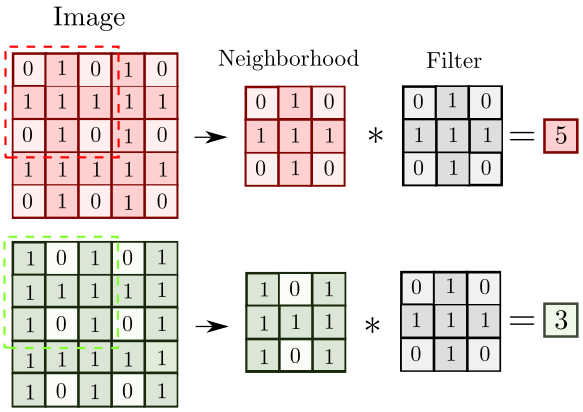


Figure 6.7: Illustration of the application of a convolution filter to a neighborhood of two different images.

6.3 Results and Discussion

We now describe our findings when applying CNN techniques to analyze LC micrographs. All scripts and data needed to reproduce the results are available in https://github.com/zavalab/ML/tree/master/LC_CNN_Color.

6.3.1 *Classification and Feature Reduction*

Our ML framework using `VGG16` features and LSVM was able to classify water and DMMP micrographs with *100% accuracy*. Notably, these results were obtained for micrographs

collected within 30 seconds of exposing the LCs to the chemical environments. This result was achieved when using all of the 128 features of the second convolutional layer. Table 6.1 reports the results for a five-fold cross-validation; here, we can see that an accuracy of 98% is obtained when we use the 64 features of the first convolutional layer. These results indicate that LC features developed *early in the sensor response* are highly informative and sufficient to discriminate among chemical environments.

From Table 6.1 we also see that it is possible to drastically reduce the feature set (this is done by selecting the features with the largest LSVM weights) while retaining a high accuracy level of 90-93%. The fact that we can obtain such high levels of accuracy with a reduced feature set can be attributed to the fact that the `VGG16` network was pre-trained using highly complex images, which suggests that many of the features extracted may be redundant or unnecessary (i.e., optical LC micrographs are simpler images than those use in generic databases such as `ImageNet`). In Table 6.2 we observe that the performance of the classifier is independent of the time at which the samples are collected. This reinforces our observation that *differences in LC features develop early in the response and they seem to persist*. Our results achieve a reduction in the number of required features reported in previous work by *two orders of magnitude*. This reduction facilitates the physical interpretation of the LC features.

Layer	**Features**	**Accuracy**	**Std.Dev.**
2^{nd} Conv.	128	100 %	$\pm$ 0 %
2^{nd} Conv.	10	93 %	$\pm$ 2 %
1^{st} Conv.	64	98 %	$\pm$ 1 %
1^{st} Conv.	10	90 %	$\pm$ 3 %

Table 6.1: Five-fold cross validation of SVM classification using `VGG16` features.

Time	2^{nd} Conv.	Std.Dev.	1^{st} Conv	Std.Dev.
3 seconds	100 %	± 0 %	96 %	± 2 %
6 Seconds	100 %	± 0 %	95 %	± 2 %
9 Seconds	100 %	± 0 %	94 %	± 2 %
12 Seconds	100 %	± 0 %	96 %	± 1 %
15 Seconds	100 %	± 0 %	94 %	± 2 %
18 Seconds	100 %	± 0 %	95 %	± 1 %
21 Seconds	100 %	± 0 %	95 %	± 2 %
24 Seconds	100 %	± 0 %	96 %	± 2 %
27 Seconds	100 %	± 0 %	96 %	± 2 %

Table 6.2: Five-fold cross validation of select time SVM classification using `VGG16` features.

To validate the classification results of our ML framework, we compared our results against the classification achieved with principal component analysis (PCA). Here, we use PCA to project the 128 dimensional feature space of the second layer into two dimensions [331]. The results of the projection are visualized in Figure 6.8. The clustering and separation of the water and DMMP features indicates that there exist perceptible differences in the CNN features of water and DMMP. These PCA results indicate that the features extracted from the CNN are indeed highly informative but the existence of a significant overlapping region also highlights that an accurate classification between micrographs requires more than two features.

The highly classification accuracy achieved, while having high importance from a sensor design stand-point, is not the only goal of our analysis. Specifically, we are interested in assigning physical interpretation of the extracted features. To do so, we analyzed the features extracted from the first convolutional layer of `VGG16` (visualized in Figure 6.6). These features are basic, highly informative, and do not depend on previous layers of convolution. Consequently, the features of the first layer are generalizable and more suit-

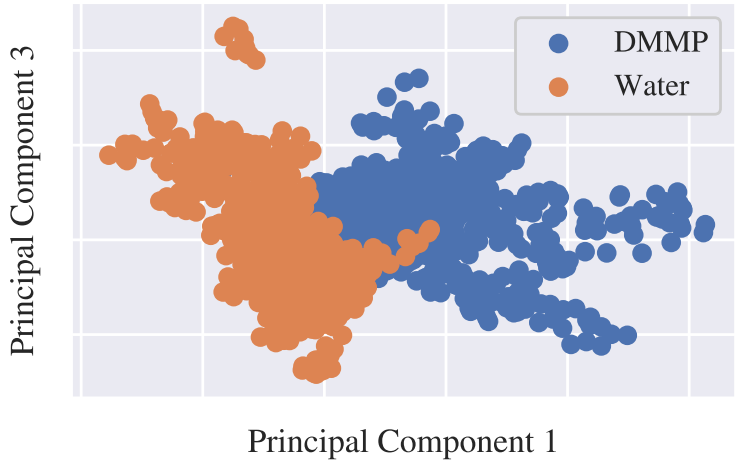


Figure 6.8: Classification using principal component analysis of `VGG16` features.

able for physical analysis. We also recall that the features extracted from the first layer of `VGG16` are the average outputs of 64 different filters. The LSVM hyperplane feature weights (shown in Table 6.3) help us identify which of these 64 filters are most dominant. Here, we see that Filter 8, 4, 52, and 38 are the most dominant ones.

Filter Number	**Filter Weight Percent**	**Filter Association**
Filter 8	16.8 %	Water
Filter 4	16.5%	Water
Filter 52	14.3%	DMMP
Filter 38	14.2%	DMMP
Filter 17	12.3%	Water
Filter 18	9.3%	DMMP
Filter 6	8.0%	DMMP
Filter 37	5.0%	Water
Filter 43	3.6%	Water
Filter 10	0.1%	DMMP

Table 6.3: Optimal LSVM weight vector obtained from training set (using ten features from first CNN layer).

6.3.2 *Maximally Activating Textures*

To obtain some insight into the spatial patterns (textures) that the most dominant `VGG16` filters (identified in Table 6.3) are capturing, we generated synthetic textures and identified the ones that maximized the average output for the different filters. This was done by feeding white noise images into `VGG16` and modifying the image to maximize the output of each filter. We refer to these textures as the maximally activating textures. A visualization of this process is seen in Figure 6.9. Visualizations of the top five maximally activating textures for water are presented in Figure 6.10 and for DMMP are presented in Figure 6.11. Here, we also show the activation fields on the input image associated with each filter. Two important aspects to consider when evaluating maximally activating textures are color and the texture (spatial pattern). The hue color is of particular interest in the analysis of LCs as different hues are a result of different orientations of the :Cs within the film [325, 332] (assuming that the LC film thickness is relatively uniform). Moreover, hue covers a spectrum of color, so it is preferred over RGB channels. In other words, a hue value captures the three values of RGB associated to a color. The maximally activating textures in Figures 6.11 and 6.10, reveal that DMMP and water have a distinct set of hues. From this observation, we conclude that *hue plays an important role in characterizing both water and DMMP responses*.

6.3.3 *Hue Analysis*

In order to understand the importance of hue in the characterization of the DMMP and water responses, we developed a simple (but interpretable) feature set for each image. Specifically, we analyzed the normalized distribution of the image hues. This distribution, which is split into 100 bins, captures the distribution of hue within each sample image (the distribution of color). Each image is then represented as a 100-dimensional H vector in which each element h_i represents the probability (frequency) of finding a pixel in a given point of the hue spectrum.

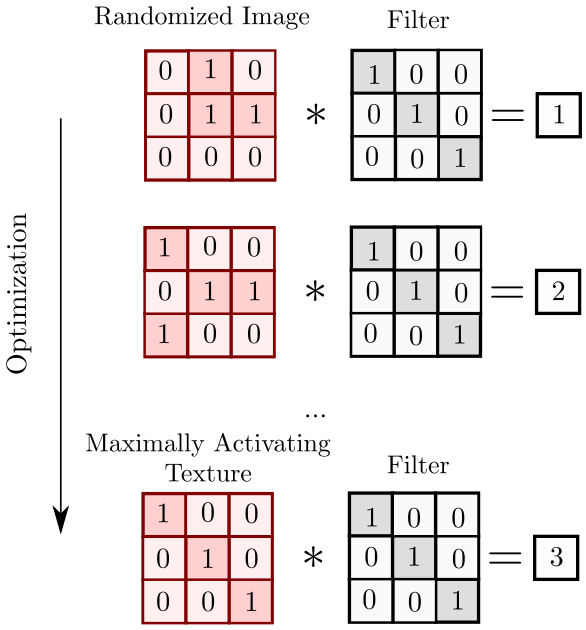


Figure 6.9: Finding maximally activating textures. To find the spatial pattern that is being maximized by a given filter, we feed different synthetic patterns and identify the one that maximizes the output.

An example hue distribution for water and DMMP are shown in Figure 6.12, along with their cumulative distribution functions (CDFs) in 6.13. From the hue distributions we see that the intensity peak at a hue value of 65 (yellow to orange) is much stronger for DMMP than for water. The CDF comparison reveals that DMMP exhibits no activity in the hue range of 20-60 (blue to yellow), while water does. The CDFs also indicate that *water micrographs have a more homogeneous coverage of the hue spectrum* (reflected as a smoother CDF curve) while DMMP micrographs have a more heterogenous coverage of the hue spectrum.

We used LSVM and hue distribution information to understand the efficacy of using hue in differentiating a water and DMMP responses. In Table 6.4 we can see that an accuracy of nearly 88% can be achieved by using hue distributions of the images alone. These results reveal that *hue (color) is an informative feature for classification*. Moreover, this result suggests that water and DMMP contain different hue distributions, which is most likely a result of differing LC orientations within the sensor film. Moreover, our

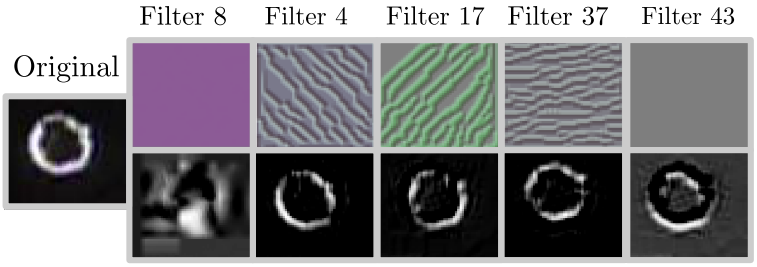


Figure 6.10: Maximally activating textures (top) and activations (bottom) for top water filters.

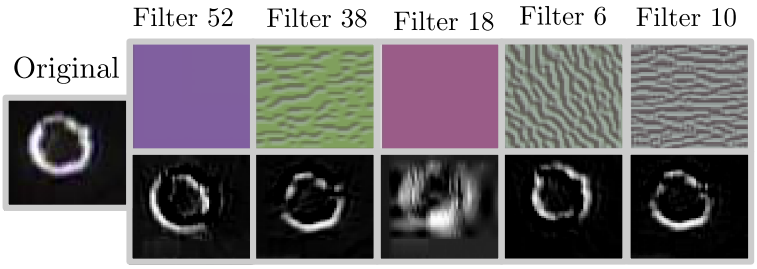


Figure 6.11: Maximally activating textures (top) and activations (bottom) for top DMMP filters.

results suggest that differences in color develop early in the response. These results make sense because the optical properties of liquid crystals are known to be highly sensitive to stimuli. The lower classification accuracy obtained with hue distributions (compared with CNN features) are attributed to the fact that hue distributions do not capture spatial pattern (correlation) information (while CNN features do).

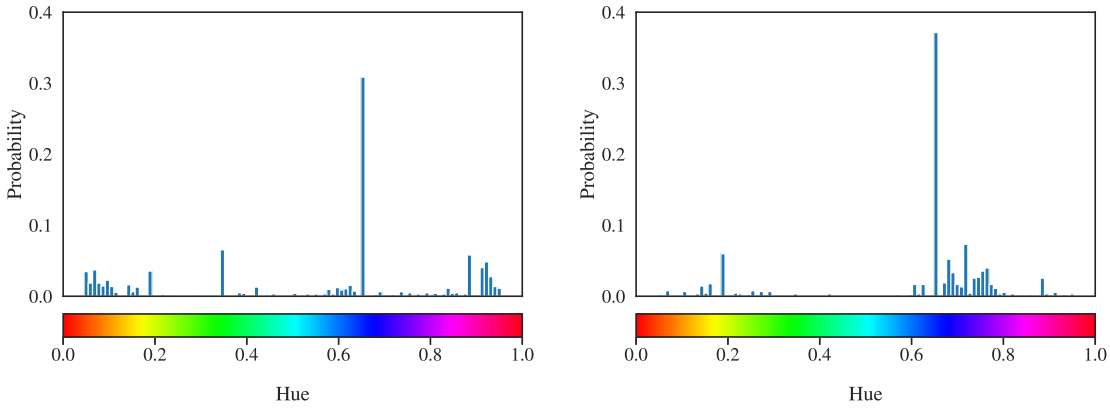


Figure 6.12: Hue distributions for representative water (top) and DMMP (bottom) micrographs.

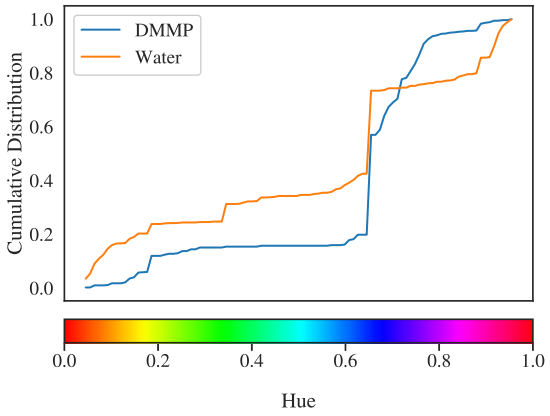


Figure 6.13: Comparison of the hue cumulative distributions for water and DMMP.

Feature Type	**Features**	**Accuracy**	**Std.Dev.**
Hue Distribution	100	88 %	± 8 %

Table 6.4: Five-fold cross validation of LSVM classification using hue distribution.

Layer	Features (Grayscale)	Accuracy	Std.Dev.
2nd Conv.	128	94 %	± 2 %
2nd Conv.	10	75 %	± 3 %
1st Conv.	64	87 %	± 3 %
1st Conv.	10	83 %	± 3 %

Table 6.5: Five-fold cross validation of LSVM classification using grayscale `VGG16` features.

6.3.4 *Grayscale Analysis*

To understand the information content that can be attributed to color and to pure spatial patterns, we used `VGG16` feature information extracted from *grayscale* images (ignoring color). From this analysis, we found that the classification accuracy was reduced by 6-12%. This further supports the observation that color is an important source of information but also that the spatial patterns found within the filters cannot be ignored. In order to analyze the grayscale patterns, we created a single texture that is a linear combination of the maximally activating textures. The linear combination was created by using the hyperplane weights obtained with LSVM. The linear combination is shown in (6.2) and the coefficients are taken from Table 6.3. The linear combinations of the grayscale patterns for DMMP and water are shown in Figure 6.14.

$$\text{DMMP Texture} = \text{Filter } 52\left(\frac{0.143}{\text{Total Weight} = 0.459}\right) + \text{Filter } 38\left(\frac{0.142}{0.459}\right) + \text{Filter } 18\left(\frac{0.093}{.459}\right) + \ldots$$

$$\text{Water Texture} = \text{Filter } 8\left(\frac{0.168}{\text{Total Weight} = 0.541}\right) + \text{Filter } 4\left(\frac{0.165}{0.541}\right) + \text{Filter } 17\left(\frac{0.123}{.541}\right) + \ldots \tag{6.2}$$

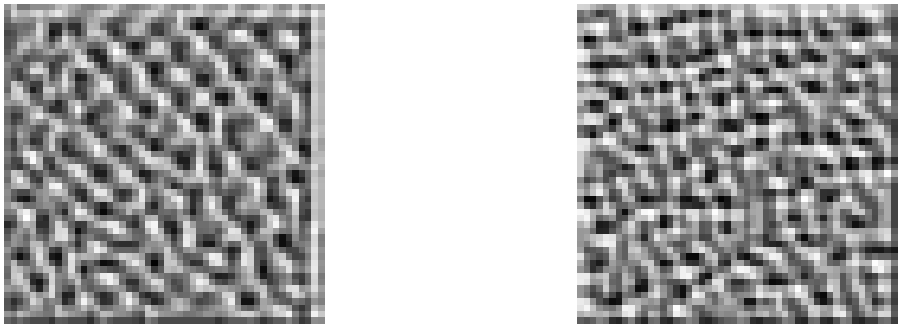


Figure 6.14: Textures for water (top) and DMMP (bottom). Textures are linear combinations of maximally activating filters.

The representative textures for both DMMP and water are used to summarize and understand differences in spatial patterns. The water texture posses a larger spatial correlation between the light and dark pixels, while the DMMP texture appears more randomized. We confirmed this observation quantitatively by analyzing the spatial autocorrelation of the textures. This is done by using Moran's I coefficient, which is a measure of global spatial autocorrelation, and is given by:

$$\text{Moran's I} = \frac{N}{\sum_i \sum_j w_{ij}} \frac{\sum_i \sum_j w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\sum_i (x_i - \bar{x})^2} \tag{6.3}$$

Here, N represents the size of the neighborhood being analyzed, x_i represents the intensity of pixel i, $\bar{x}$ represents the average intensity in neighborhood N, and w_{ij} represents the inverse distance weighting matrix in neighborhood N.

Texture	**Moran's I**	**P Analysis**
Water	0.54	$P < 0.00001$
DMMP	0.40	$P < 0.00001$

Table 6.6: Global Moran's I coefficient values.

The Moran's I coefficients reveal that both DMMP and water patterns have positive spatial autocorrelation with high confidence (Table 6.6) but that autocorrelation in water is of longer range. This result may be further validated by calculating the local Moran's

I coefficient values for every pixel in the image in a 3x3 pixel neighborhood. The resulting correlation fields, shown in Figure 6.15, indicate that the DMMP texture has higher variance and that areas of both positive and negative autocorrelation are clustered. For the water texture, on the other hand, we see a more uniform autocorrelation and with higher overall magnitude (confirming the observations obtained with the global Moran's I coefficient).

Our analysis indicates that `VGG16` is capable of unraveling spatial patterns that result from exposure of the LC sensor to either DMMP or water. Moreover, we conclude that perceptible changes in spatial patterns are sufficient for the LSVM to discern between two chemical environments with high accuracy. We hypothesize that the differences in correlation length of the LC textures detected by `VGG16` with DMMP and water reflect differences in the anchoring energy of the LC on the surface of the sensor. Specifically, a high anchoring energy will suppress LC orientational fluctuations and lead to a small correlation length. This result suggests that one key influence of water on the LC is to lower the anchoring energy at the metal salt-coated surface used in the LC sensor. The result also suggests that macroscopic orientational transitions may not be necessary in order to detect targeted chemical species using LCs, but that characterization of fluctuations in orientation by using `VGG16` may be a useful future strategy to explore in experiments.

Overall, analysis of both the grayscale spatial patterns and hues provide new insight into possible physical mechanisms that underlie the ability of `VGG16` to differentiate the response of the LC sensors to water and DMMP. Moreover, an additional important finding of our study is that perceptible changes in both color and spatial patterns can be detected with `VGG16` within seconds of exposure of the LC film to the chemical environments (a thin bright ring is only perceptible by human vision early in the response).

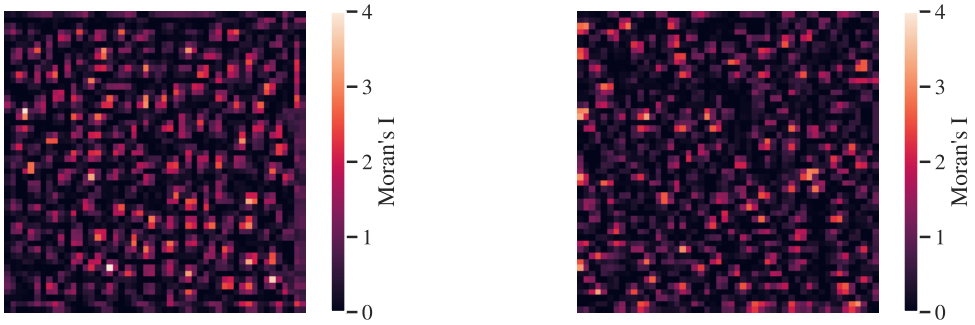


Figure 6.15: Local Moran's I analysis for water (top) and DMMP (bottom).

Chapter 7

HIGH-DIMENSIONAL DATA ANALYSIS - CATALYSIS

The contents of this chapter are published in [23]

7.1 Introduction

Machine learning (ML) provides a powerful set of techniques that facilitates the analysis of the high-dimensional structure of data sets and the construction of predictive models from such data sets [333]. ML techniques have been recently utilized in a wide variety of catalysis applications; an excellent review is provided in [334]. For instance, ML techniques have been used to uncover likely reaction mechanisms and to accelerate screening of different catalyst formulations. For instance, in [335], ML techniques are used to construct surrogate density functional theory (DFT) models that enable accelerated predictions of adsorption energies based on group additivity fingerprints and to identify rate limiting steps. A similar method is demonstrated for identifying electrocatalysts for CO_2 reduction and H_2 evolution. Here, active learning is used for identifying candidate active surfaces [336]. Surrogate DFT models have also been used to accelerate the search of new compounds and crystal structures with promising catalytic properties [337]. Artificial neural networks (ANN) are used in [338] to develop predictive models that use first-principles *ab initio* data on adsorption energies and electronic fingerprints

of idealized bimetallic surfaces as descriptors to predict surface reactivity of metal alloys. It is shown that complex nonlinear interactions of adsorbates on multimetallics can be captured by the ANN model.

In this chapter, we present a ML framework to explore the predictability limits of catalytic activity from different sources of experimental descriptor data that characterize catalyst formulations and reaction conditions. Our framework uses ANNs to fuse experimental descriptor data to construct predictive models. The framework also uses principal component analysis (PCA) and sparse PCA to identify experimental points and descriptors that contain large amounts of information and that have strongest impact on prediction accuracy of the ANN. Specifically, PCA is used to project the physical experimental design space into an information space and this allows us to identify regions under which the ANN predictions are likely to be more accurate. We also propose a constrained-PCA formulation that identifies experiments in different regions of the information space while factoring in constraints from of physical, economical, and expert nature. We illustrate that the framework can be used to uncover catalyst formulations and operating conditions that can help improve the predictive accuracy of the ANN model.

The ML modeling approach proposed contrasts with classical approaches such as microkinetic modeling (in which mechanistic insights are used to predict kinetic behavior). These classical approaches have been widely applied to the water-gas shift reaction (WGSR) [339, 340, 341, 342] and provide deep and generalizable insights that can inform decisions on reactor design, reaction conditions, and support/promoter selection. However, these models are limited in the types of information that they can incorporate and need to be adjusted for different reaction settings (e.g., different catalyst formulations). As a result, these modeling approaches are time-consuming. Scaling relations and Brønsted-Evans-Polanyi relations have been developed to address some of these limitations [343] but these are only effective within limited interpolation ranges. ML models can capture more general ranges and nonlinear behavior and are also more flexible in the information that they can incorporate but these models do not provide mechanistic insight and are

less generalizable. The primary goal of a ML framework is thus not to replace classical methods but instead help analyze large experimental data sets and with this unravel non-obvious trends that ultimately help develop mechanistic insights and models [344].

The proposed ML framework is applied to a comprehensive data set for the low-temperature WGSR, which is one of the most widely studied reactions in the literature and for which vast amounts of experimental data are available. Our framework analyzes 2,228 experimental data points reported in the literature in over a decade [345]. We also propose a comprenhensive set of 27 descriptors that characterize catalyst formulations and reaction conditons in experiments reported. The reaction rate is used as a decriptor of catalytic activity and we trained ANNs to predict activity using the experimental descriptor data. We show that ANNs can effectively fuse comprehensive sets of descriptor data (that would be difficult to incorporate in classical kinetic models). Moreover, we find that accurate predictions can be obtained with only 30% of the data points available. However, our analysis also reveals that the ANN model will be of limited use for predicting activity for new catalyst formulations (not contained in the existing experimental data set). PCA analysis reveals that this is because catalyst formulations and reaction conditions explored in the literature are highly clustered in narrow regions of the information space. Specifically, we find that nearly 90% of available experimental information arises from changes in the reaction conditions (only 10% arises from changes in formulations). PCA analysis also reveals that traditional descriptors for catalyst formulations do not provide much information to the ANN model. These results suggest that predictability limitations can be addressed with more systematic data collection procedures that explicitly quantify information content of experimental points. Moreover, results suggest that new sources of descriptors (as those obtained from DFT calculations for catalyst screening and characterization) are needed [346, 347, 348].

7.2 Data Collection and Preparation Methods

The data set collected by Odabasi and co-workers was used as the basis of this work [345]. Each *experimental data point* is characterized by a set of *descriptors* for the catalyst formulation and for the reaction conditions. In our framework, we propose a set of 27 descriptors (see Tables 7.1 and 7.2). The logarithm of the reaction rate (denoted as k) was used as a measure of catalytic activity. The primary metals, support, and promoters considered in the data set are summarized in Table 7.4. The entire data set studied can be found in the supplementary material.

Descriptor Name	Range	Units
Loading - Weight Percent (Primary Metal)	0.1 to 39.8	Weight %
Binding Energy of Carbon (Primary Metal)	-6.46 to -3.06	eV
First Ionization Energy (Promoter)	0 to 13.58	eV
Electronegativity (Promoter)	0 to 1.91	
Covalent Radius (Promoter)	0 to 2.44	Angstroms
(Z) Ionic Radius (Promoter)	0 to 1.522	1/pm
Lowest Oxidation State (Promoter)	-4 to 2	
Highest Oxidation State (Promoter)	0 to 7	
Redox Potential (Promoter)	-3.04 to 0.3	E^o (V)
MW (Promoter)	0 to 186.21	
Loading - Weight Percent (Promoter)	0 to 78.7	Weight %
Redox Potential (Support)	-3.8 to 0.8570	E^o (V)
First Ionization Energy (Support Metal)	0.5344 to 0.7865	kJ/mol/1000
Electronegativity (Support)	1.10 to 1.90	
Highest Oxidation State (Support)	3 to 7	
Lowest Oxidation State (Support)	-4 to 1	
Molecular Weight (Support Element)	26.98 to 232.04	
Molecular Weight (Entire Support)	56.07 to 325.82	

Table 7.1: Descriptors for catalyst formulations

The original data set reported in [345] contains 4,360 experimental points. This set was filtered such that only reaction mechanisms and pathways of the *low-temperature* WGSR are explored (different mechanisms and pathways arise at high temperatures). The data set was further filtered by removing experimental points with a thermodynamic driver β outside the range [0,0.8] or points in which the thermodynamic driver cannot

Descriptor Name	Range	Units
Calcination Time	0 to 10	hours
Calcination Temperature	25 to 650	^{o}C
Reaction Temperature	423.15 to 623.15	K
H_2 feed	0 to 60	volume %
H_2O feed	2 to 60	volume %
CO_2 feed	0 to 15	volume %
CO feed	0.2 to 12	volume %
Time on Stream	0 to 5808	min
F/W	0.028 to 173	mL_{total}/minute/mg cat

Table 7.2: Descriptors for reaction conditions

be computed. Experimental points that use zeolite, hydroxyapatite, activated carbon, and mixed supports were removed because these supports are difficult to characterize based on descriptors reported in the literature (i.e., advanced first-principles descriptors are needed for this). Experimental points that use CH_4 and O_2 co-feeds in the reaction conditions were also removed. These data filtering steps resulted in a data set comprising 2,228 points. Table 7.3 summarizes the criteria and points removed from the original set. The filtered data was standardized using a *z-score* transformation. This form of standardization ensures that descriptors have zero mean and that their covariance values are comparable. Standarization also ensures that the use of different measurement units does not impact dimensionality reduction analysis and facilitates training of predictive ANN models [349].

Criterion	Removed Points	Remaining Points
Original Data	–	4360
Zeolite Support	66	4294
Hydroxyapatite Support	58	4236
Activated Carbon Support	25	4211
YSZ Promoter	21	4190
CH_4 Co-feed	223	3967
O_2 Co-feed	32	3935
$\beta > 0.8$	265	3670
$\beta < 0$	66	3604
β not computable	176	3428
Temperature (^{o}C) < 150	118	3310
Temperature (^{o}C) > 350	440	2870
Mixed Supports	642	2228

Table 7.3: Summary of data filtering steps

Catalyst Material	Variations
Primary Metal	Au, Cu, Pt, Pd, Ir , Rh , Ru
Promoter	Li, Na, K, Rb, Cs, Mg, Ca
	Sr, Y, La, Ce, Nd, Sm, Gd
	Ho, Er, Tm, Yb, Ti, Zr, V
	Cr, Mn, Re, Fe, Co, Ni , Zn
Support	Al_2O_3, MgO, CeO_2, TiO_2, MnO_3, Y_2O_3, Tb_4O_7
	HfO_2, La_2O_3, Co_3O_4, ThO_2, SiO_2, Fe_2O_3, Sm_2O_3
	Gd_2O_3, Yb_2O_3, CaO, ZrO_2

Table 7.4: Primal metals, support, and promoter considered in WGSR data set.

7.3 Computational Methods

It is important to highlight that the experimental descriptor space for the WGSR is high-dimensional. In particular, discretizing each of the 27 descriptor dimensions in 3 points results in a total of $3^{27} = 7.6 \times 10^{12}$ possible experimental points. As a result, it is impossible to explore the entire space and thus systematic techniques are needed to explore

such space efficiently. In this section we describe the elements of a machine learning framework that seeks to enable this. The framework includes a principal component analysis (PCA) component to project the descriptor space into a low-dimensional information space, a neural network component to predict catalytic activity from descriptor data, and a constrained-PCA component to identify experimental points in information-rich regions while filtering out regions that are unreachable due to economic, physical, or technical reasons (provided by an expert user). A scheme of the proposed framework is presented in Figure 7.1.

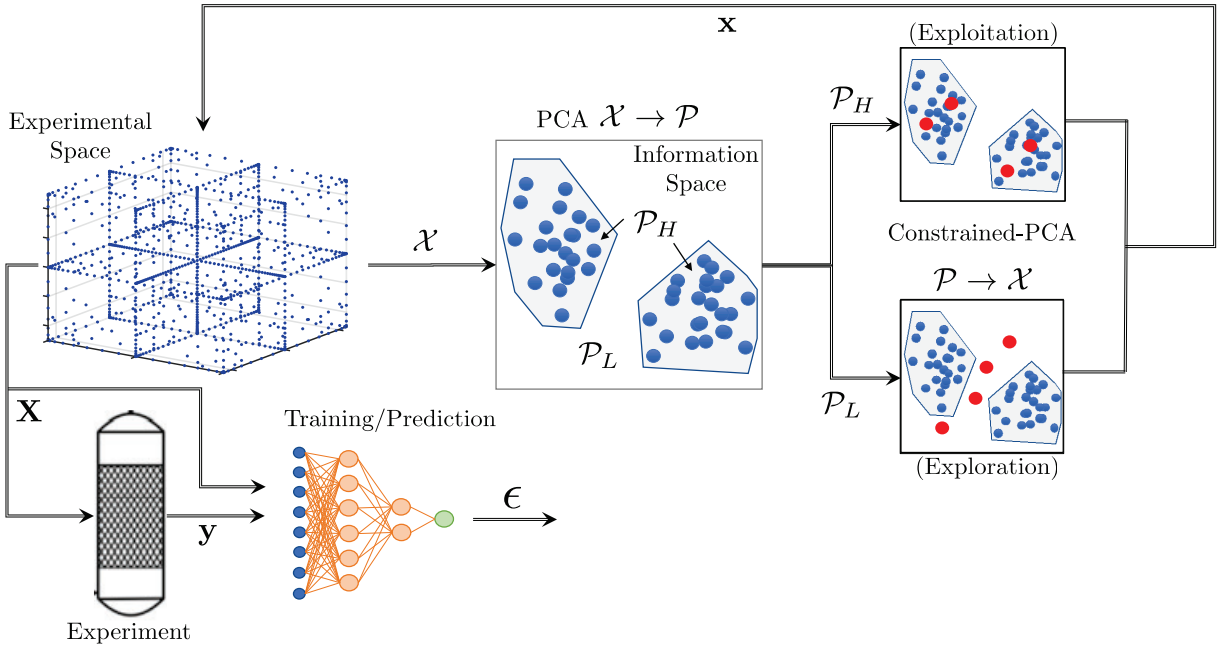


Figure 7.1: Scheme of machine learning framework for prediction of catalytic activity.

7.3.1 *Principal Component Analysis*

Principal component analysis (PCA) is a popular technique that is used to reduce the dimensionality of experimental data sets while retaining as much as information present in the data as possible [350]. PCA can be interpreted as a procedure that transforms

data from an original (physical) space into an information content space. This is done by rotating a given data point from its original space into the new space in which directions of variance (information) are decoupled. The directions are known as the principal components and correspond to the eigenvectors of the covariance matrix for the experimental data set. The eigenvalues associated with the eigenvectors provide a measure of information content in each principal component.

To explain the basic principles of PCA, we consider a experimental data set matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ with n rows (experimental points) and p columns (descriptors). We denote the i-th data point (row) of $\mathbf{X}$ as $\mathbf{x}_i \in \mathbb{R}^p$ for $i = 1, ..., n$. Each of these points is a descriptor vector that lies in the experimental data space $\mathcal{X}$ that represents the entire set of possible experimental points. The first principal component $\mathbf{P}_{i,1} \in \mathbb{R}$ associated with experimental data point $\mathbf{x}_i$ is defined as the linear combination of the elements of $\mathbf{x}_i$ with coefficient vector $\mathbf{v}_1 \in \mathbb{R}^p$ and takes the form $\mathbf{P}_{i,1} = \mathbf{v}_1^T \mathbf{x}_i$. PCA seeks to find a eigenvector $\mathbf{v}_1$ of unit length such that the variance of the first principal components $\mathbf{P}_{i,1}$, $i = 1, ..., n$ is maximized. Recognizing that the covariance matrix $\Sigma \in \mathbb{R}^{p \times p}$ of the data points $\mathbf{x}_i$ is $\Sigma = \mathbf{X}^T \mathbf{X}$, we have that $\mathbf{v}_1^T \Sigma \mathbf{v}_1$ is the variance of $\mathbf{P}_{i,1}$. Consequently, finding the coefficient vector $\mathbf{v}_1$ that maximizes the variance subject to $\|\mathbf{v}_1\|^2 = 1$ yields the largest eigenvalue of Σ (denoted by λ_1) and the associated eigenvector $\mathbf{v}_1$. The next eigenvector $\mathbf{v}_2$ is found by maximizing the variance of the second principal component $\mathbf{P}_{i,2}$, subject to the constraint that the eigenvector is orthogonal to $\mathbf{v}_1$ (i.e., $\mathbf{v}_1^T \mathbf{v}_2 = 0$). This reveals that the maximum variance is the second largest eigenvalue λ_2 and $\mathbf{v}_2$ is the corresponding eigenvector. Orthogonality ensures that the first principal component is uncorrelated from the second component. This process is repeated to obtain p eigenvectors $\mathbf{v}_j$, $j = 1, ..., p$ that generate p principal components for each data point $\mathbf{x}_i$. The principal component information is collected in a matrix $\mathbf{P} \in \mathbb{R}^{n \times p}$ and the eigenvectors in a matrix $\mathbf{V} \in \mathbb{R}^{p \times p}$. We denote each row of principal components (corresponding to experimental point $\mathbf{x}_i$) as $\mathbf{p}_i$. The principal component vector $\mathbf{p}_i$ is a point that lies in an information space (that we denote as $\mathcal{P}$). The covariance matrix can also be expressed as an expansion of the form $\Sigma = \sum_{j=1}^{p} \lambda_j \mathbf{v}_j \mathbf{v}_j^T$.

Typically, most information of Σ is contained in the first few largest eigenvalues. This is often visualized using an spectral graph, which shows the cumulative sum of the largest eigenvalues.

Principal components are descriptors in the information space that are linear combinations of descriptors in the physical experimental space. The coefficients of such linear combinations are given by the eigenvectors. Principal components are obtained by projecting a data point from the physical space $\mathcal{X}$ into an information space $\mathcal{P}$ by using the projection matrix $\mathbf{V}$. Understanding this synergy between physical and information spaces is important because it allows us to quantify the information content of experimental data points and of different descritors. For instance, experimental points that are far apart in the physical space might in fact be close in the information space (and thus provide a similar amount of information). This can occur, for instance, if the experimental points are highly correlated. Similarly, experimental points that are close together in the physical space might in fact be far apart in the larger information space if they are uncorrelated. These types of insights cannot be directly obtained from inspection of the experimental data because of its dimensionality and because of the presence of complex multivariable interactions. PCA thus provides a powerful tool for understanding the underlying structure of a high-dimensional experimental data set that can inform the selection of experiments and descriptors. Carlson and co-workers provide a detailed overview on the application of PCA to experimental design [351].

The interpretation of eigenvectors obtained in PCA is complicated by the fact that the principal components are functions of the entire set of descriptors (the eigenvectors are dense). *Sparse PCA* is a method that attempts to address this issue by generating sparse eigenvectors (containing only a few nonzero entries). Sparsification allows for a *sharper separation* of the most relevant descriptors and thus enhances interpretability. Diverse sparse PCA approaches have been explored in the literature [352, 353, 354]. Here, we use the sparse regression approach proposed in [352]. Under this approach, the i-th column

of the sparse eigenvector matrix $\hat{\mathbf{V}}$ (denoted by $\hat{\mathbf{V}}_i$) is given by:

$$\hat{\mathbf{V}}_i = \frac{\hat{\mathbf{v}}_i}{||\hat{\mathbf{v}}_i||} \tag{7.1}$$

where $\hat{\mathbf{v}}_i \in \mathbb{R}^p$ is the solution vector of the optimization problem:

$$\hat{\mathbf{v}}_i = \underset{\mathbf{v}}{\operatorname{argmin}} \, ||\mathbf{p}_i - \mathbf{X}\mathbf{v}||_2^2 + \kappa_2||\mathbf{v}||_2^2 + \kappa_1||\mathbf{v}||_1. \tag{7.2}$$

Here, $\hat{\mathbf{v}}_i$ is the sparse eigenvector that best matches the i-th column of the principal component matrix $\mathbf{P}$. Sparsification of the eigenvector is induced by adding ℓ_1 and ℓ_2 penalties to the regression problem (with corresponding positive weights κ_1 and κ_2). The sparse eigenvectors are used to construct the sparse matrix $\hat{\mathbf{V}}$, which is a sparse approximation of $\mathbf{V}$. Consequently, the variance contained in the sparse eigenvectors is less than that contained in the original *dense* eigenvector (but the interpretation of the sparse counterparts is enhanced). The optimization formulation (7.2) is known as the *elastic net* and was solved using the `LARS-EN` algorithm [355]. The *SpaSM* toolbox [356] was utilized for obtaining the sparse PCs.

7.3.2 *Neural Network Model*

The proposed model is an ANN that seeks to predict catalyst activity (reaction rate) as a function of the descriptors for catalyst formulations and reaction conditions. We use this model to assess the predictability limits of reaction rates and the impact of different experiments and descriptor information on prediction accuracy. The ANN computes the prediction mapping $\mathbf{X} \rightarrow \hat{\mathbf{y}}$ and we defined the associated prediction error as $\epsilon = \mathbf{y} - \hat{\mathbf{y}}$. Here, $\mathbf{y} \in \mathbb{R}^n$ is an output vector containing the experimental reaction rates corresponding to the input experiment matrix $\mathbf{X}$ and $\hat{\mathbf{y}} \in \mathbb{R}^n$ is an output vector containing the associated predicted rates.

ANNs are complex networks composed of simple processing elements (called *neu-*

rons) operating on local data and communicating with other elements [357]. The ANN structure chosen in this work to analyze the WGSR data set is a multi-layer feedforward neural network. These types of ANNs are made up of layers of neurons, the first layer of neurons is the input to the ANN and the last layer is the output of the ANN. The signals emanating from our layer are fixed and mapped to another function using activation functions. The parameters that capture the mixing of signals between layers and the parameters of the activation functions are determined by solving a nonconvex optimization problem that seeks to minimize the mean squared errors (MSEs) of the predicted outputs. ANNs are a powerful tool to construct predictive models because they can capture complex multivariable and nonlinear relationships between descriptors by mixing signals across multiple layers. This ability is clearly stated in the work of Cybenko et al. [358], in which it is shown that any continuous function of real variables can be approximated by an ANN.

PCA can aid the development of predictive ANN models because it will reveal areas that are sparse or dense in information and with this reveal regions under which we can trust the model predictions. To capture this, we split the information space $\mathcal{P}$ into high-information (densely covered) space $\mathcal{P}_H$ and low-information (sparsely covered) space $\mathcal{P}_L$. These spaces are associated to the dual goals of the predictive model (exploitation and exploration). In a high-information space, the model can be trusted more and it can thus be exploited to identify new catalyst formulations or reactions conditions that will maximize activity. In a low-information space, the model is less trusted and thus one needs to explore this space using new experiments to reinforce the model. PCA can also reveal descriptors that contain high information or low information because PCA will reveal descriptors that are strongly correlated (e.g., a descriptor that is a linear combination of other descriptors).

7.3.3 *Constrained-PCA*

PCA provides a *bridge* between physical and information spaces that can be used to guide experiments. Here, we propose an optimization formulation that seeks to identify new experimental conditions that achieve a desired target in the information space while factoring in *constraints* in the physical space of technical or economic nature (imposed by an expert).

Having a projection matrix $\mathbf{V}$ (or $\hat{\mathbf{V}}$) obtained from PCA (or sparse PCA) for a given experimental data set, we can use such matrix to project an experimental point from the physical space *mathcal*X to the information space $\mathcal{P}$. Our objective is to use the projection matrix to determine a new experimental point $\mathbf{x}$ such that its associated principal components $\mathbf{p} = \mathbf{V}\mathbf{x}$ reach a desired target point in the information space $\bar{\mathbf{p}}$. The target point $\bar{\mathbf{p}}$ can be selected in the high-information space $\mathcal{P}_H$ to conduct exploitation (e.g., find a catalyst formulation and reaction conditions that improves activity) or in the low-information space $\mathcal{P}_L$ to conduct exploitation (e.g., find a catalyst formulation and reaction conditions that seeks to improve the predictive accuracy of the ANN model).

As we search for an experiment that reaches the target $\bar{\mathbf{p}}$, we also seek to restrict the experimental conditions to lie withing a range of the form $\mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U$. Moreover, we can impose general constraints; for instance, we can seek that the experiment is such that its cost $\mathbf{c}^T\mathbf{x}$ does not exceed a certain derived value $\bar{c}$ (e.g., $\mathbf{c}^T\mathbf{x} \leq \bar{c}$). Here, $\mathbf{c} \in \mathbb{R}^p$ is a cost coefficient vector that captures the individual cost of given descriptors (e.g., higher temperature requires higher energy and some metals are more expensive than others). The constraints can be expressed in general form as $\mathbf{M}\mathbf{x} \leq \mathbf{r}$, where $\mathbf{M}$ is a coefficient matrix and $\mathbf{r}$ is a coefficient vector. Constraints can also be used to express combinations of catalyst formulations and reaction conditions that are not compatible (e.g., a certain metal might only be compatible with certainty supports or promoters). Constraints thus help an *expert* express logic and convey prior knowledge that helps filter out large regions of the experimental space. This can help navigate this space in a more scalable manner

(which is extremely high-dimensional).

In summary, the new experiment $\mathbf{x}$ is found by solving the constrained-PCA problem:

$$\min_{\mathbf{x}} \|\mathbf{p} - \bar{\mathbf{p}}\| \tag{7.3a}$$

$$\text{s.t. } \mathbf{p} = \mathbf{V}\mathbf{x} \tag{7.3b}$$

$$\mathbf{M}\mathbf{x} \leq \mathbf{r}. \tag{7.3c}$$

Here, $\|\cdot\|$ is a a suitable norm. In summary, the above formulation seeks to find an experiment $\mathbf{x}$ that lies in a desired information region and while satisfying the constraints.

7.4 Results and Analysis

In this section we use the proposed ML framework to analyze predictability of catalytic activity for the WGSR data set.

7.4.1 *PCA and Sparse PCA*

Application of PCA to the WGSR data set reveals that the first principal component contains 30% of the total variance, the second component contains 15%, and that 90% of the variance is contained in the first ten components. This is illustrated in the spectral graph presented in Figure 7.2. This reveals that the experimental input matrix $\mathbf{X}$ can be compressed by a factor of three and thus there is a significant amount of redundancy in the data. This is seen in Figure 7.3, where we present the projection of the experimental data into the information space (we only show the first two principal components). This reveals highly clustered experimental points along linear planes in the information space. We have found that experimental points along such planes result from variations of reaction conditions (temperatures and flow rates). This can be more clearly seen in Figure 7.4a, where we categorize the data points by temperature. From this we can conclude

that a handful of variations in temperature conditions are actually needed to cover the information space (e.g., the extremes of the temperature ranges in each cluster) and the rest are redundant (provide little information). These results also seem to suggest that more information is obtained by varying the catalyst formulation compared to varying the reaction conditions. To confirm this observation, we conducted PCA using only descriptor data for the catalyst formulation (descriptors listed in Table 7.1). In other words, we eliminated all descriptors associated with reaction conditions. Figure 7.5 shows that, by doing so, the overall structure of the data is preserved. This indicates that variations of reaction conditions do not provide significant information. This is an issue because a very large fraction of the data available in the literature (nearly 90%) arises from changes in the reaction conditions (while the rest 10% varies the catalyst formulation). In fact, there are only 187 unique catalyst formulations (combinations of primary metals, promoters, and supports) in the experimental data set.

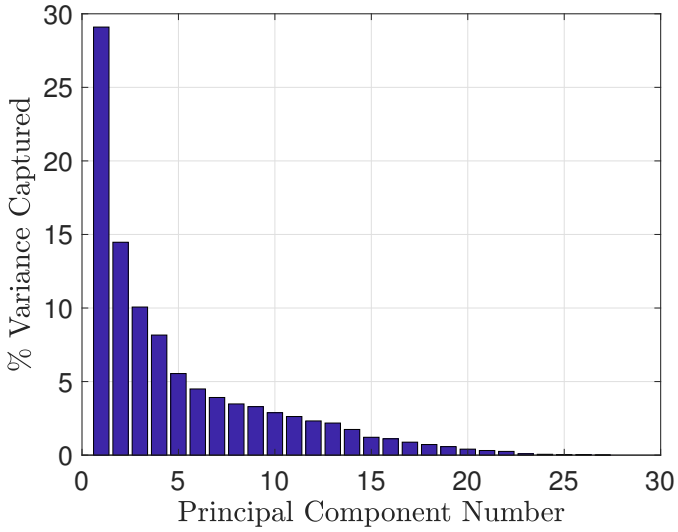
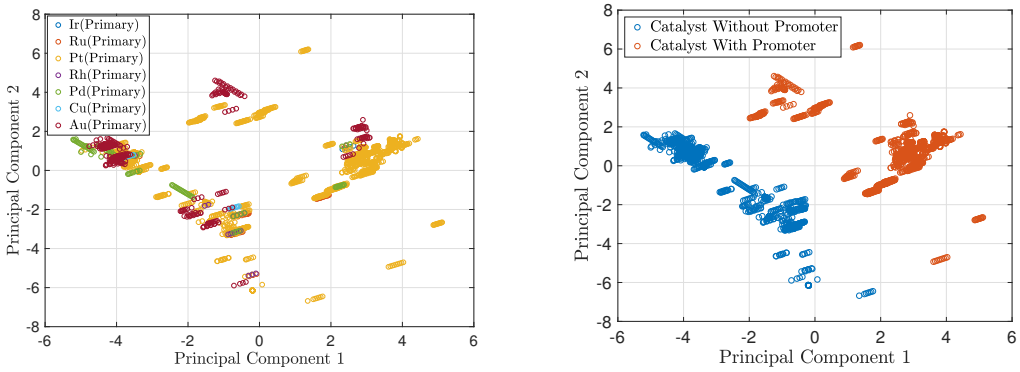


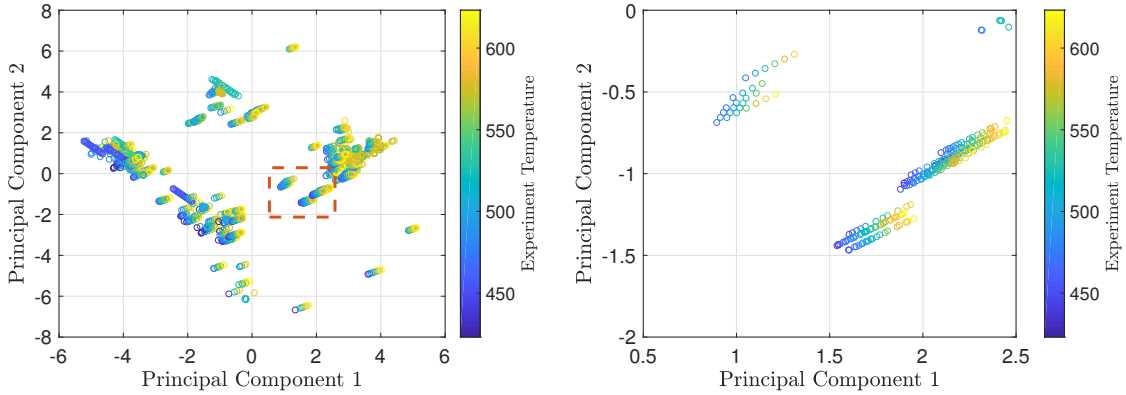
Figure 7.2: Spectral graph of WGSR data set

The experimental data points are categorized in terms of the primary metal of the catalyst formulation in Figure 7.3a(b). Here, we can see that platinum experiments span a much wider region of the information space. The data points are categorized in terms



(a) PCs categorized by primary metals (b) PCs categorized by promoter

Figure 7.3: PCA projection of WGSR data set into information space



(a) PCs categorized by temperature (b) PCs categorized by temperature (zoomed)

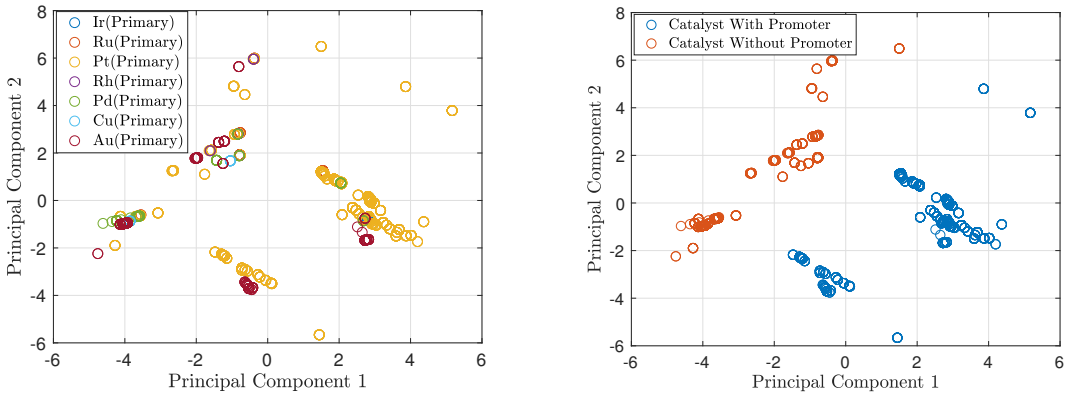
Figure 7.4: PCA projection of WGSR data set (categorized by temperature).

of catalysts with and without promoters in Figure 7.3b. Here, it is clear that there is a separation in the information space. This indicates that certain regions in the information space might not be accessible with (or without) the use of a promoter. In other words, without the use of a promoter, it will not be technically possible to design an experiment that accesses the top-right region in the information space. This highlights that there exist technical constraints that limit the exploration of the information space and the predictability of the model.

Figure 7.5 reveals that high clustering is also observed when changing the catalyst formulations (suggesting that the different formulations explored in the data set do not

provide much information). Figure 7.7b shows that the split between catalyst formulations with and without promoters is still present without the features in Table 7.1. We thus conclude that large areas in the information space have not been explored in the literature. Novel and more informative catalyst formulations are thus needed.

PCA analysis also reveals interesting outliers in the experimental data set. In particular, Figure 7.6 shows that a separate cluster of experimental points associated with Au(CeO_2) and Pt(CeO_2) catalysts exists. These catalysts are efficient for the low-temperature WGSR and utilize a rather unique reaction mechanism [359].



(a) PCs categorized by primary metal

(b) PCs categorized by promoter

Figure 7.5: PCA projection of WGSR data set (using only catalyst descriptors).

Dense and sparse eigenvectors for the WGSR data set are presented in Table 7.5. We found that the sparse eigenvector of the first principal component contains descriptors related to the promoter, while the sparse eigenvector for the second component contains descriptors related to the support and the primary metal. This indicates that the promoter descriptors are more informative than those of the primary metal and of the support. This makes sense, since the data sets only contain two descriptors for the primary metal (binding energy and loading weight percent) while they contain nine descriptors for the promoter. In other words, the promoter is better characterized. The sparse principal components also indicate that certain descriptors of the primary metal (loading weight percent), of the promoter (e.g., charge low, redox potential), and of the support (e.g.,

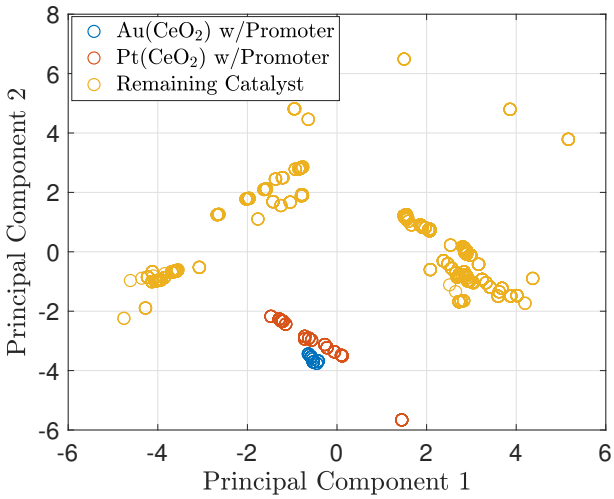
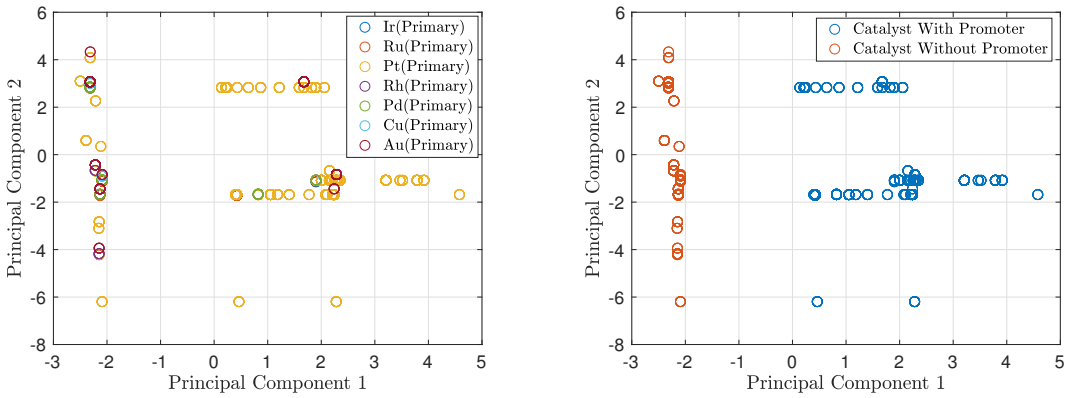


Figure 7.6: Location of $Au(CeO_2)$ and $Pt(CeO_2)$ clusters.



(a) Sparse PCs categorized by primary metal

(b) Sparse PCs categorized by promoter

Figure 7.7: Sparse PCA projection of WGSR data set.

Descriptor	Sparse PC 1	Sparse PC 2	Dense PC 1	Dense PC 2
Binding Energy of Carbon (Primary Metal)	0	.1	-.1	-.1
Loading - Weight Percent (Primary Metal)	0	0	-0.1	-.03
MW (Promoter)	0.2	0	0.3	-0.2
First Ionization Energy (Promoter)	0.5	0	0.3	-0.2
Covalent Radius (Promoter)	0.4	0	0.3	-0.2
Charge Low (Promoter)	0	0	-.1	0.03
Charge High (Promoter)	0.5	0	0.3	-0.2
(Z) Ionic Radius (Promoter)	0.3	0	0.3	-0.2
Electronegativity (Promoter)	0.5	0	0.3	-0.2
Redox Potential (Promoter)	0	0	-0.2	0.1
Loading - Weight Percent (Promoter)	0	0	0.1	-0.2
Molecular Weight (Entire Support)	0	0.4	-0.3	-0.3
Molecular Weight (Support Metal)	-0.1	0.3	-0.3	-0.2
First Ionization Energy (Support)	0	-0.5	0.2	0.4
Redox Potential (Support)	0	-0.4	0.2	0.36
Highest Oxidation State (Support)	0	0.4	-0.3	-0.3
Lowest Oxidation State (Support)	0	0	-0.1	0.2
Electronegativity (Support)	0	-0.4	0.3	0.3

Table 7.5: Sparse and dense PCA eigenvectors. Zeros represent exclusion, and postive/negative values represent the influence of the feature on the direction of the Principal Component.

lowest oxidation state) are not informative (they have an eigenvector entry of zero).

Sparse PCA results also indicate that the separation in the descriptors is sharp (few descriptors appearing in the first principal component appear in the second component and viceversa). These results contrast with those obtained with standard (dense) PCA, where we note that all descriptors appear in the first and second descriptors. We can thus see that the sparse PCA results are easier to interpret.

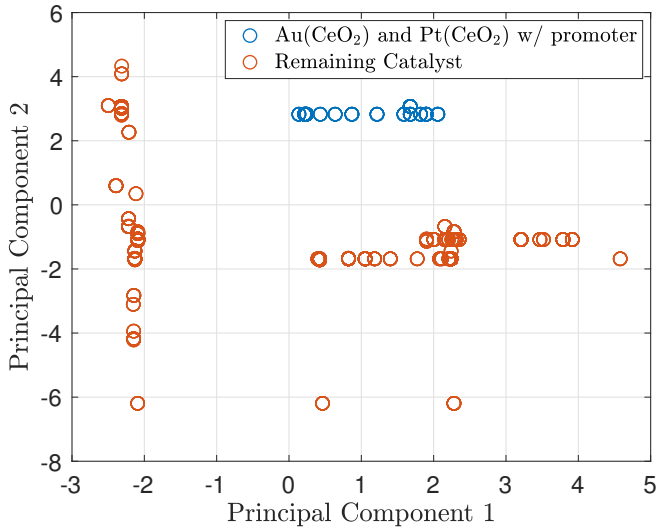


Figure 7.8: Separation of $Au(CeO_2)$ and $Pt(CeO_2)$ data sets using sparse PCA.

Figure 7.7 illustrates that the sparse PCA information space is also not well explored and also reveals that a clear separation between catalysts with and without promoter exist. Figure 7.8 demonstrates a clear separation between experiments that use $Au(CeO_2)$ and $Pt(CeO_2)$ catalysts. We thus conclude that sparse PCA maintains the structure of the information space of standard PCA (but a sharper separation in obtained).

7.4.2 *Neural Network Predictions*

The structure of the ANN utilized for the WGSR is depicted in Figure 7.9. The ANN structure used in this work has two hidden layers (one with six nodes and the other with two nodes). The 27 experimental descriptors are used as inputs and the only output is the

reaction rate. This optimized structure was identified by assessing the prediction accuracy for different structures. We use the mean squared error (MSE) of the reaction rate as a measure of prediction accuracy of the ANN. Having a fixed optimized ANN structure, five ANNs were trained simultaneously on a randomly selected portion of the WGSR data set. Five ANNs were trained simultaneously in order to overcome entrapment in local minima and overfitting. The trained ANNs were then used to predict the reaction rate for 425 *randomly* selected experiments.

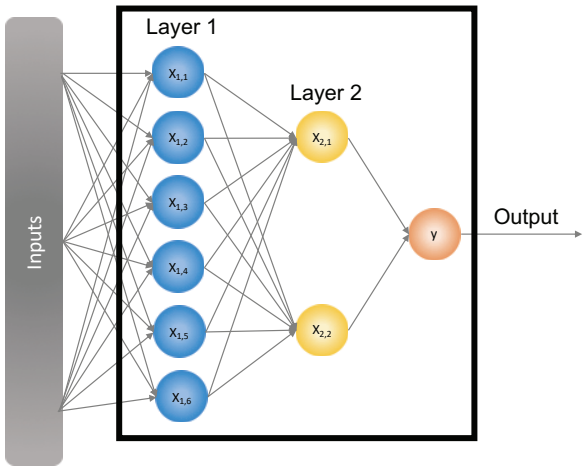


Figure 7.9: Structure of neural network used for the prediction of catalytic activity.

A learning curve for the ANN is presented in Figure 7.10. Here, it is shown that the accuracy of the ANN settles after using 30% of the data points for training (chosen randomly). This indicates that the ANN is indeed quite effective at capturing relationships between the multiple descriptors to predict catalytic activity. Figure 7.11 presents a regression plot for the ANN trained with only 30% percent of the available data for training. We can see that the ANN exhibits high accuracy over the entire span of the space covered by the experimental data.

7.4.3 *Selection of Formulations using ANNs*

We propose a couple of experiments to demonstrate the practical application of a trained ANN. The first experiment is to select a set of primary metals, supports, and promoters

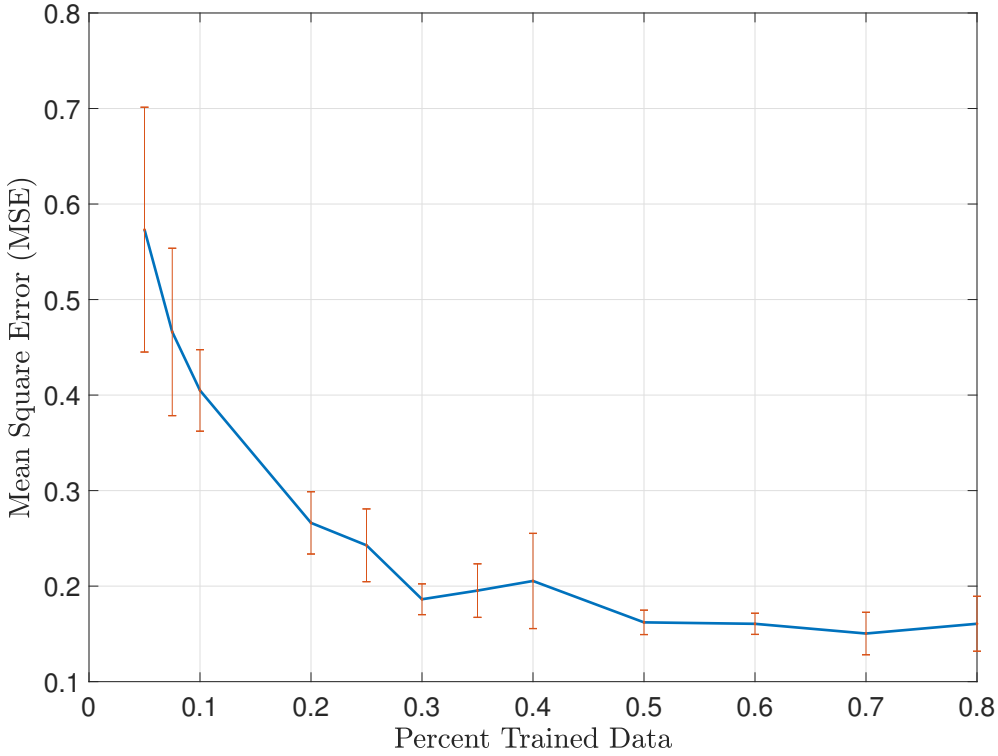


Figure 7.10: Learning curve for neural network for WGSR data set.

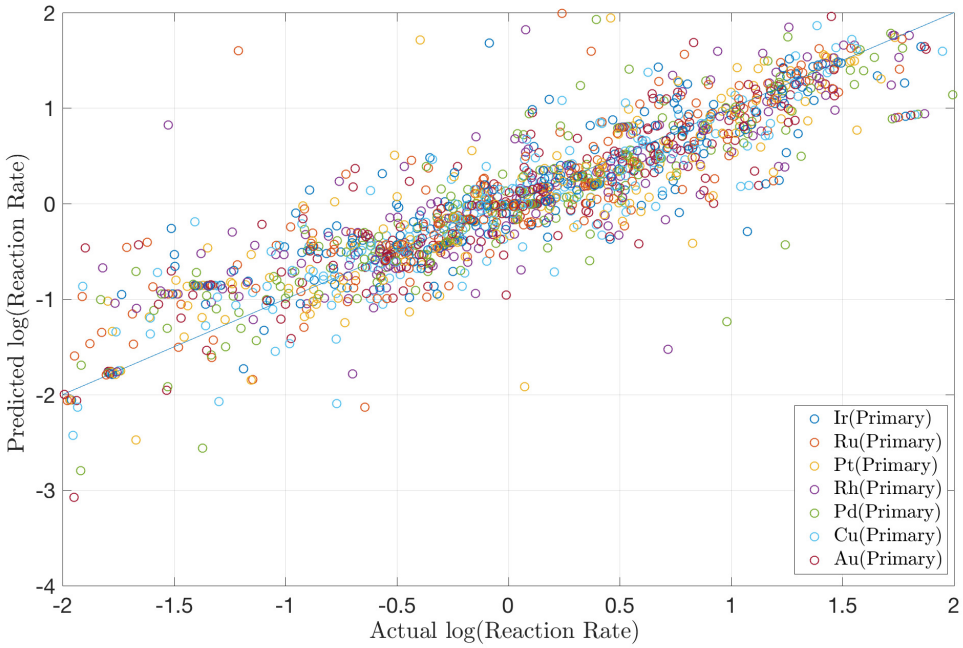


Figure 7.11: Regression plot for neural network for WGSR data set (30% of data set used for training).

that provides the highest reaction rate at different temperatures. In this case, we focus on formulations that are already present within the data set. The ANN has been provided information for different catalyst formulations and predicts the reaction rate $\log(k)$ from which we identify the top predicted performers. The primary metal, support, and promoters are found in Table 7.6. The top performers identified with the ANN and the true top performers are found in Table 7.7. In both cases, the network is able to identify the top formulations. In Table 7.8 we present the predicted $\log(k)$ for the top formulations against the true $\log(k)$. We observe that the ANN is able to capture overall trends on catalyst performance (key to select suitable formulations) but the actual predicted values for $\log(k)$ show some significant errors.

Catalyst Material	Variations
Primary Metal	Au, Cu, Pt, Pd, Rh , Ru, Ir
Promoter	Na, Cr, Ni, Y, Sr, La, Ce, Nd, Re
Support	Al_2O_3, SiO_2, TiO_2,MnO_3, Fe_2O_3, ZrO_2, CeO_2,HfO_2

Table 7.6: Primary metals, supports, and promoter considered in ANN experimental data set.

Temp (K)	Rank	ANN (Prime,Sup,Prom)	Experimental (Prime,Sup,Prom)
473.15	1	Pt, CeO_2, (None)	Pt, CeO_2, (None)
473.15	2	Au, CeO_2, (None)	Au, CeO_2, (None)
473.15	3	Pt, ZrO_2, (None)	Au, ZrO_2, (None)
623.15	1	Pt, CeO_2, (None)	Pt, CeO_2, (None)
623.15	2	Pt, ZrO_2, (None)	Au, CeO_2, (None)
623.15	3	Au, CeO_2, (None)	Pt, ZrO_2, (None)

Table 7.7: Prediction results for first ANN experiment (formulation selection).

Temperature (K)	Rank	Predicted $\log(k)$	True $\log(k)$
473.15	1	2.67	1.24
473.15	2	0.57	1.21
473.15	3	0.56	0.599
623.15	1	3.056	2.91
623.15	2	2.84	2.03
623.15	3	2.46	2.81

Table 7.8: Prediction results for first ANN experiment (reaction rate).

A second computational experiment was conducted to identify effective promoters for Au supported on CeO_2 that are not contained within the current data set. The selected promoters are found in Table 7.9. The trained ANN was provided different primary metal,

promoter, and support formulations at a given set of reaction conditions ($T = 473K$) to predict $\log(k)$ and with this identify the top promoters. The top three promoters identified are shown in Table 7.10 along with their predicted $\log(k)$. Interestingly, none of these suggested promoters are contained in the current data set and do not seem to have been explored in the literature. Consequently, these would be interesting candidates to explore experimentally.

Catalyst Material	Variations
Primary Metal	Au
Promoter	Li, Na, K, Rb, Cs, Mg, Ca, Sr, Y, La, Ce, Nd, Sm, Gd
	Ho, Er, Tm, Yb, Ti, Zr, V, Cr, Mn, Re, Fe, Co, Ni , Zn
Support	CeO_2

Table 7.9: Primary metals, supports, and promoter considered in ANN experimental data set.

Temperature (K)	Promoter	Predicted $\log(k)$
473.15	Li	5.56
473.15	Na	4.49
473.15	Mg	4.42

Table 7.10: Results for second ANN experiment (promoter selection).

7.4.4 *Neural Network Analysis*

In order to understand the *impact of individual descriptors* on the prediction accuracy of the model, we conducted a *leave-one-out* analysis. Here, the ANN was trained using 50% of the data but a given descriptor was omitted each time. This procedure was repeated five times for each omitted descriptor to ensure consistency in the ANN predictions. Tables 7.11 and 7.12 summarize the results. Here, we recall that the best MSE value obtained is 0.14 (using the entire set of descriptors). The analysis reveals that *temperature* is the most

important descriptor (removing this descriptor increases the MSE by 128%). Temperature is followed by the flow per gram of catalyst and the volumetric percentages of H_2, CO, and H_2O along with the binding energy of carbon and the loading of the primary metal. Calcination time and temperature are important because they impact physical structure, pore volume, and surface area. We also found that descriptors for promoters and and support metals have the least impact on the MSE. We note, however, that these low-impact descriptors cannot be removed from the data set (as they collectively embed information). To highlight this point, we trained the ANN without using the descriptors shown in Table 7.12. From Table 7.13 we see that the MSE increases by 135%. From Figure 7.12 we can see that prediction accuracy is decreased (compare against Figure 7.11, obtained with the entire set of descriptors).

Descriptor left out	MSE
Temperature (K)	0.32
CO_2 feed volume %	0.19
Loading - Weight Percent (primary metal)	0.19
Binding Energy of Carbon (primary metal)	0.17
Calcination Time (hours)	0.16
Calcination Temperature (C^o)	0.16
H_2 feed volume %	0.16
H_2O feed volume %	0.16
F/W (mL_{total}/minute/mg Catalyst)	0.16

Table 7.11: Leave-one-out analysis (descriptors with highest impact on MSE).

Exploration of the information space obtained with PCA can be used to anticipate and analyze the prediction performance of the ANN. For example, our previous PCA analysis reveals that data points corresponding to Au-Cerium(Oxide) and Pt-Cerium(Oxide) form a well-defined and isolated clusters in the information space (see Figure 7.13a). When the ANN is trained without these points, predictability is significantly affected (see Figure 7.13b). This is because the space under which the ANN has been trained is not well covered (and thus requires significant extrapolation). This indicates that ANN is not fully capable of predicting activity across catalyst formulations. This point is reinforced in Figure 7.14, which shows that the prediction accuracy when descriptors for different

Descriptor left out	MSE
CO feed volume %	0.15
Time on Stream (min)	0.14
First Ionization Energy (Promoter)	0.14
Electronegativity (Promoter)	0.14
Covalent Radius (Promoter)	0.14
(Z) Ionic Radius (Promoter)	0.14
Charge Low (Promoter)	0.14
Charge High (Promoter)	0.14
Redox Potential (Promoter)	0.14
MW (Promoter)	0.14
Loading - Weight Percent (Promoter)	0.14
Redox Potential (Support)	0.14
First Ionization Energy (Support Metal)	0.14
Electronegativity (Support)	0.14
Highest Oxidation State (Support)	0.14
Lowest Oxidation State (Support)	0.14
Molecular Weight (Support Metal)	0.14
Molecular Weight (Whole Support)	0.14

Table 7.12: Leave-one-out analysis (descriptors with lowest impact on MSE).

Descriptors Included	MSE
Table 7.11, 7.2, and 7.12 (Base Case)	0.14
Table 7.11 and 7.2	0.33

Table 7.13: Impact of sets of descriptors on prediction accuracy.

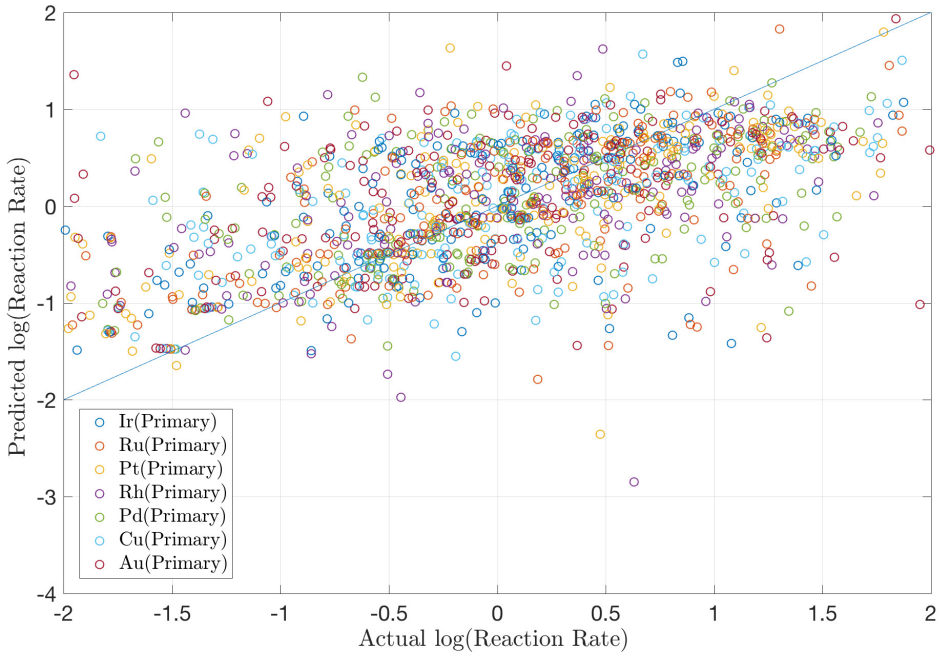
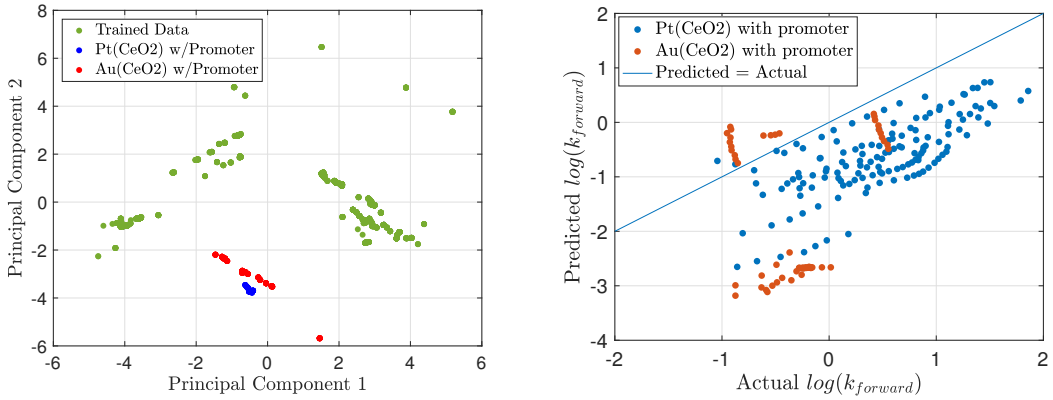


Figure 7.12: Prediction accuracy without descriptors of Table 7.12.

primary metals have been excluded from the training data set. In particular, the MSE increases by at least one order of magnitude. These results also reinforce the observation that descriptors of primary metals have a strong impact on prediction accuracy.



(a) $Au(CeO_2)$ and $Pt(CeO_2)$ points in information space.

(b) Regression plot for $Au(CeO_2)$ and $Pt(CeO_2)$ points.

Figure 7.13: Impact of removing $Au(CeO_2)$ and $Pt(CeO_2)$ data from training on prediction accuracy.

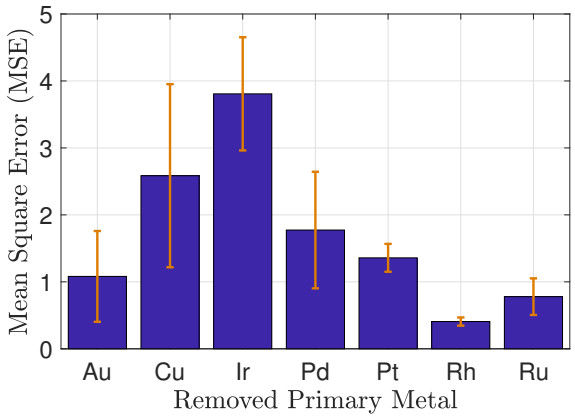


Figure 7.14: Mean squared errors of predictions upon removal of different primary metal data points.

From our analysis we conclude that WGSR data reported in the literature covers a wide range of experimental conditions but such data in fact has limited information content. In particular, we have seen that many data points cover only small portions in the information space. PCA analysis can be utilized to select experiments that better cover the information space. In particular, loading information can be used to identify a new descriptor vector that targets a specific point in the information space. This can potentially lead to the discovery of new catalysts and novel reaction mechanisms. It is also possible, however, that certain regions of the information space are not reachable due to physical or technical reasons. In the next section we discuss an approach to identify such points.

We also conclude that information content of experimental data reported in the literature needs to be enriched with additional (first-principles) descriptors. In particular, descriptor data currently reported can effectively predict variations in reaction conditions but cannot fully anticipate activity for formulations that are not included in the training set. In other words, the ANN has limited extrapolation capabilities. As a result, the ANN is of limited used when searching for new catalyst formulations. Descriptors from DFT can be used to enrich information and enable prediction accross formulations. Promising results are reported in [346, 347, 348]. In particular, Greeley and co-workers

[346] utilize DFT to develop search criteria for new catalysts for the hydrogen evolution reaction. Descriptors are used to characterize phenomena such as catalyst segregation and island formation. Ras and co-workers. The work in [348] uses descriptors such as adsorptive gaps and molecular adsorptive volumes. Goeltl and co-workers [347] explored structural descriptors, such as stability of the active site, the center of mass of the unoccupied orbitals, and the energetic center of d orbitals, to understand adsorption strength to transition metal sites. The combination of PCA, ANNs, and experimental/DFT descriptor data can enable more powerful predictive capabilities.

7.4.5 *Constrained-PCA*

The constrained-PCA formulation is used to provide insight into the unexplored space associated with possible catalyst formulations and reaction conditions. Figure 7.15 provides a visualization of all possible catalyst formulation combinations for primary metals, promoters, and supports in the information space and contrasts this to the explored space currently reported in the literature. All combinations found are reported in supplementary information. Clearly, there exist many formulations that have not been reported in the literature. This may be due to technical infeasibility or simply lack of time and budget.

To further illustrate the capabilities provided by the constrained-PCA framework, we seek a catalyst formulation that is close to the Au(CeO_2) and Pt(CeO_2) formulations in the information space but that is also *less expensive*. Our framework reveals that a catalyst that achieves this consists of a Cu primary metal, a Co promoter, and CeO_2 support. Figure 7.16 shows the target point $\bar{\mathbf{p}}$ in the information space and the most cost effective catalyst formulation that is close to it. Interestingly, the catalyst formulation suggested by constrained-PCA *is not included in the WGSR data set* studied here. However, work conducted by Li and co-workers [360] report that a Cu(CeO_2) catalyst retains high WGS activity up to 600°C, similar to that achieved by Au(CeO_2) and Pt(CeO_2) catalysts.

We have also found that another cost effective catalyst that is close to the target in

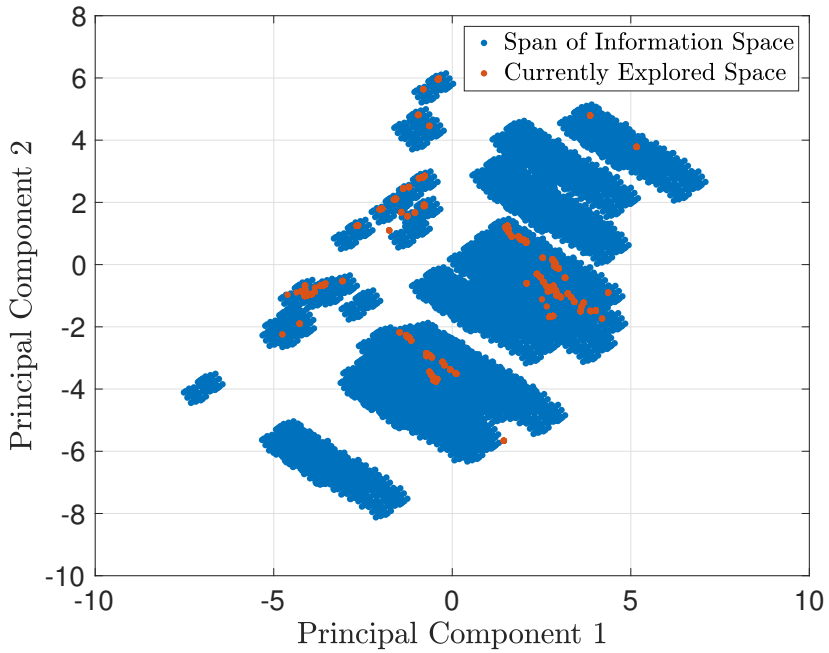


Figure 7.15: Full span of information space identified by constrained-PCA and space explored in the literature.

the information space is one that uses Y_2O_3 as support. This formulation is again not included in the WGSR data set studied here but work by Yusheng and co-workers [361] explored the addition of Y to $Cu(CeO_2)$ catalysts and suggest that the addition of Y may facilitate the formation of oxygen vacancies on the Ce support. These results illustrate how a ML framework can help navigate the information space in a more effective manner.

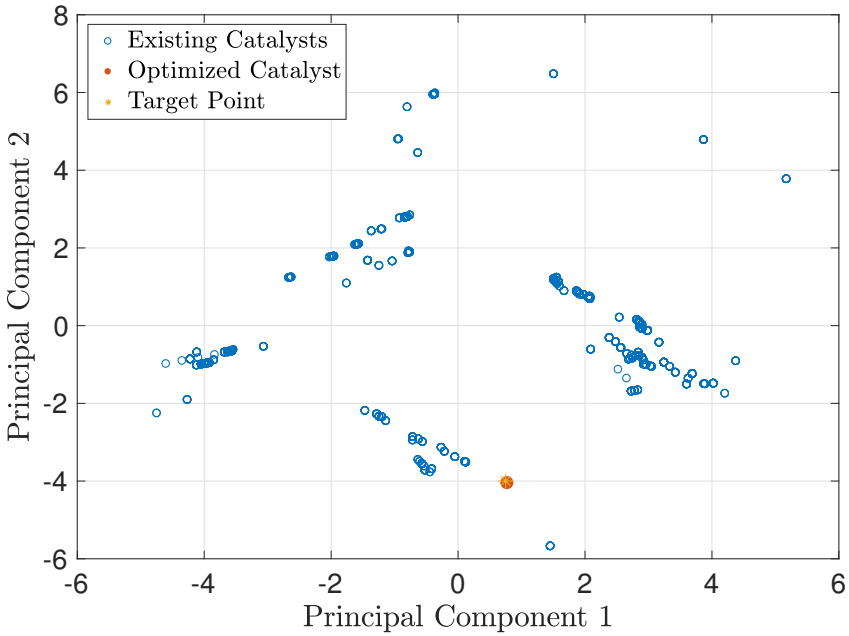


Figure 7.16: Identification of $Cu(CeO_2)$ catalyst formulation using constrained-PCA.

Part IV

FINAL THOUGHTS

Chapter 8

CONCLUSIONS AND FUTURE DIRECTIONS

The incorporation of topology, geometry and other data driven methods in the quantification of the *shape* of data yields data representations that are simple to compute, stable, and physically interepretable. We have also shown that characterizations of data that account for the topology and geometry of data can both simplify the models needed for data analysis while also improving the effectiveness of many tasks such as classification, regression, and dimensionality reduction. The objective of this dissertation has been to provide scientists and engineers a rigorous introduction to the mathematics of applied geometry and topology, supply software and tools to implement these ideas, and to illustrate motivating applications of these methods on real-world datasets. We now conclude this dissertation by summarizing the key findings/contributions made through the work presented in Chapters 2 - 7.

8.1 Contributions

The Euler Characteristic

In Chapter 2 we define and apply the the Euler characteristic (EC) which is a powerful tool for the characterization of complex data objects such as point clouds, graphs, matrices, images, and functions/fields. The EC summarizes the topological characteristics of

such data objects. We have demonstrated that the EC can be used in a wide variety of applications through creative data transformations (e.g., point clouds to fields and multivariate time series to correlation matrices). We have also shown that the EC provides an effective descriptor and can be used as a pre-processing step that simplifies visualization, clustering, and classification tasks.

The Topology of Molecular Simulations

In Chapter 3 we expand upon the application of the topological methods in Chapter 2 to molecular simulations. Manifold and graph representations of molecular simulation data provide a flexible avenue for capturing both discrete and continuous information sources. In this chapter, we analyze the topology associated with simulations of self-assembled monolayers and acid-catalyzed reaction systems. We characterize these topological representations with the Euler characteristic curve. We show that this method results in improvements in computational efficiency, generalizability, and simplicity which can be leveraged to reduce the complexity of models needed for analysis of molecular simulation data (e.g., 3D convolutional neural networks are reduced to linear regression models). These improvements also reduce the reliance on large labeled datasets, which are needed for ML model training, and provide physical intuition. This provides opportunities to improve the information gained from the analysis of high-throughput or large-scale simulation data, which can be used in screening for new materials and chemistry or in optimizing physical and chemical characteristics of existing systems.

Topological Data Analysis and Persistence Homology

Chapter 4 introduces the area of Topological Data Analysis (TDA) and a particular method known as persistence homology. Topological data analysis (TDA) provides a set of powerful methods and tools for understanding the underlying topology and geometry of data. These techniques represent data such as point clouds and functions, as geometric objects and explores these objects in terms of basic geometric and topological features. We show that TDA offers a number of important theoretical properties (such as stability), offers flexibility to extract features from different types of data, and how these extracted

features can be exploited using statistical and machine learning techniques.

Riemannian Geometry

Chapter 5 presents an introduction to concepts of Riemannian geometry for symmetric positive definite (SPD) matrices and shows how these concepts can be used in applications of interest to chemical engineers. Specifically, we discussed approaches to capture the geometry of the SPD manifold (a Riemannian manifold) in dimensionality reduction and classification tasks. Through a couple of case studies, we demonstrated that capturing such geometry can lead to significant improvements in accuracy and interpretability

Convolutional Neural Networks and Liquid Crystals

Chapter 6 explores convolutional neural networks as a data driven method for pattern and structure quantification. We developed a machine learning framework to obtain high classification accuracies for optical micrographs of LC-based sensors. The features used for classification are outputs of the convolutional filters over a given image, which were extracted from the first and second layers of the VGG16 network. these features were analyzed through the creation of a linear combination that represented LC responses to various environments. Analysis of these spatial patterns indicates that the liquid crystal sensor response has perceptible differences in spatial correlation and hue (color).

High-Dimensional Structure of Catalysis Datasets

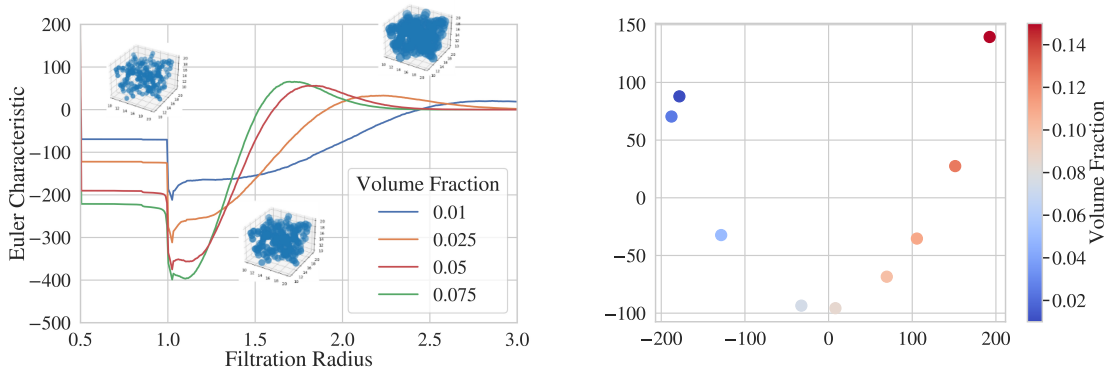
In Chapter 7 we review and develop dimensionality reduction methods such as principal component analysis (PCA) and sparse PCA to characterize the high-dimensional structure of a large experimental database. We presented a data driven framework to explore the predictability limits of catalytic activity based on experimental descriptor data that characterizes catalyst formulations and reaction conditions. The framework is applied to a comprehensive data set for the water–gas shift reaction which comprises descriptor data for diverse catalyst formulations and reaction conditions. We also demonstrate that the use of dimensionality reduction techniques, such as PCA, can allow for a deeper understanding of large catalysis datasets and can also provide a method for identification of new and unique catalyst formulations via constrained-PCA.

8.2 Future Research Directions

This section focuses on areas of future research at the intersection of topology, geometry, and chemical engineering. Our future research will provide *concise*, *interpretable*, and *scalable* characterizations of data that can be integrated in *machine learning* and *statistical analysis* tasks (e.g., classification and principal component analysis). The aim of this future research is to identify unifying and rigorous principles to elucidate a wide range of phenomena through topology, geometry, and data science.

Topology and Geometry of Complex Materials

A grand challenge in characterizing and designing functional soft materials (e.g., polymers, proteins) is the topological and chemical complexity of the resulting macromolecules. The resulting material properties, such as large-scale morphologies or aggregation dynamics, are difficult to understand using collective variables derived from domain expertise. TGDA provides a direct approach for the analysis of these complex morphologies and aggregations. For example, in a collaboration with *Prof. Emanuela Del Gado* (Georgetown) we are beginning to characterize colloidal gel structures through topology and geometry [93]. Figure 8.1 presents an analysis of the topology of gel simulations at varying volume fractions through the Euler characteristic. TGDA is used to pre-process the data and principal component analysis (PCA) is used to reduce the data dimensionality. We identify a low-dimensional, continuous data manifold that provides connections between the gel structure and the volume fraction. This research will seek to form fundamental connections between the microscale properties of functional materials and their macroscale behavior. The ability to effectively quantify macroscale structure through TGDA will provide useful first-principles connections across time and length scales in these complex materials, aiding our experimental collaborators in understanding, designing, and optimizing new state-of-the-art materials.



(a) EC curve of different gel structures

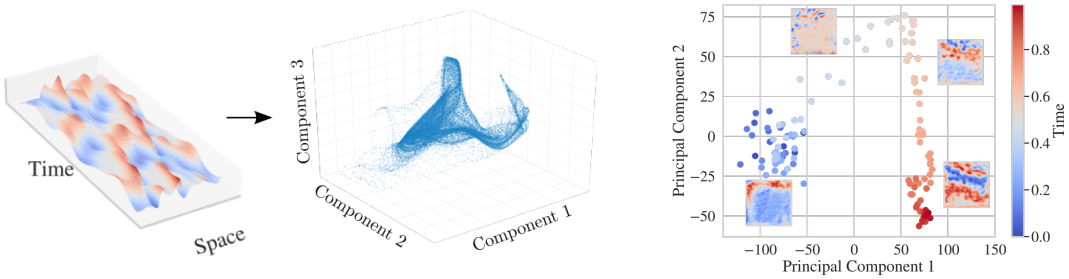
(b) PCA of EC curves

Figure 8.1: TGDA coupled with simple dimensionality reduction techniques (e.g., PCA) can be used to understand the complex geometry and topology of self assembled colloidal systems such as gels. Here, we see that there is a continuous relationship between the topology of a gel and its volume fraction.

Spatio-Temporal Analysis.

Spatio-temporal datasets are found in a wide range of engineering research areas such as fluid dynamics, molecular simulation, thermodynamics, materials science, biology, and many others. Spatio-temporal data, produced through experiments and simulations, encodes a large amount of useful information about the properties of these systems. However, extracting this information is incredibly difficult due to the myriad of challenges associated with spatio-temporal data, such as the high-dimensionality and heterogeneity of the data (i.e., behavior in space differs from behavior in time) [100]. Due to these complexities, spatio-temporal data is often averaged over space or time in order to simplify the needed analysis, resulting in information loss. TGDA is able to tackle data that is heterogeneous and high-dimensional without restrictive assumptions such as stationarity or isotropy [168, 31]. For example, in collaboration with the group of *Prof. Michael Graham* (UW - Madison) we have begun exploring the topology of chaotic, spatio-temporal dynamical systems like the Kuramoto-Sivashinsky (KS) equation found in Figure 8.2. TGDA is leveraged to simplify realizations of the KS equation, yielding an optimal low-dimensional representation of the resulting data. The structure of this low-dimensional manifold aids in the identification of laminar and turbulent behavior in the data. In a

collaboration with *Prof. Nicholas Abbott* we characterize the space-time evolution of the optical response of a liquid crystal (LC) sensor after exposure to an analyte (Figure 8.2). The characterization yields a clarified, low-dimensional representation of the response, distilling the information encoded in the spatio-temporal dynamics of the LC system and helping distinguish responses to different analytes. Our research in this area will focus on expanding the role of TGDA in spatio-temporal data analysis through the development and advancement of TGDA methods, and will aid our collaborators in understanding the spatio-temporal behavior of their studied systems while extracting physically meaningful information from these complex datasets.



(a) TGDA of the Kuramoto-Sivashinsky equation.

(b) TGDA of LC response.

Figure 8.2: TGDA coupled with simple dimensionality reduction techniques (e.g., PCA) can be used to elucidate and summarize complex dynamics found in spatio-temporal datasets.

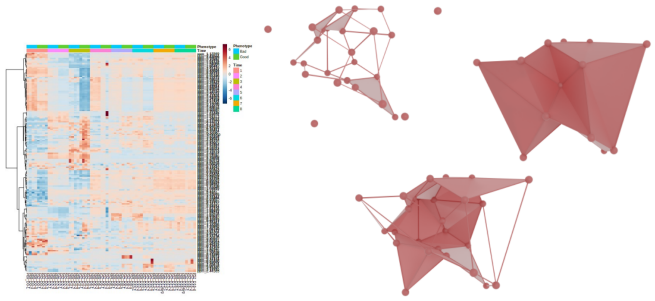


Figure 8.3: Multi-omics data can be represented as a high-dimensional simplicial complex.

Topology of High-Order Network Structures.

Graph-based abstractions of data have found a broad set of applications ranging from optimization to chemical and biological networks [30]. These representations encode interactions between variables that can be useful for the analysis of complex systems (e.g., large biological networks). For example, in collaboration with *Prof. Sean Palecek* (UW-Madison), we are exploring multi-omics data for pluripotent stem cells and how this information can be used to predict differentiation of cardiomyocytes. In the analysis of multi-omics data, the integration of biological network structure is crucial, because it establishes relationships between variables that reflect cell biology. Unfortunately, graph-based representations are restricted to *pairwise* interactions between variables and cannot capture higher-order relationship which are found in these biological networks (e.g., proteins A, B, and C must be expressed before D can be expressed). Abstractions such as simplicial complexes and hypergraphs can more readily encode these higher-order interactions and can be be quantified through TGDA [362]. The collaboration with Prof. Palecek is focused on integrating multi-omics data with simplicial complexes and hypergraph structures (Figure 8.3). TGDA and higher order networks have also found a large number of applications in understanding, diagnosing, and treating *cancer* which we will build upon. For example, TGDA has been applied to identify genetic alterations in cancer, for predicting treatment responses, and in tumor segmentation [363, 364]. Importantly, these high-order structures are not limited to biological networks, these also exist in chemical reaction networks, supply chains, industrial chemical processes, electrical grids, and many other systems [365]. The focus of this research will be to exploit these high-order topological representations in modeling chemical and biological systems, while leveraging TGDA for their analysis.

Connecting Physics, Statistics, Topology, and Geometry.

Current TGDA methods provide a large collection of topological and geometrical descriptors for complex systems and datasets. The ability to confirm topological differences or similarities is key in many applications; unfortunately, many of the descriptors available

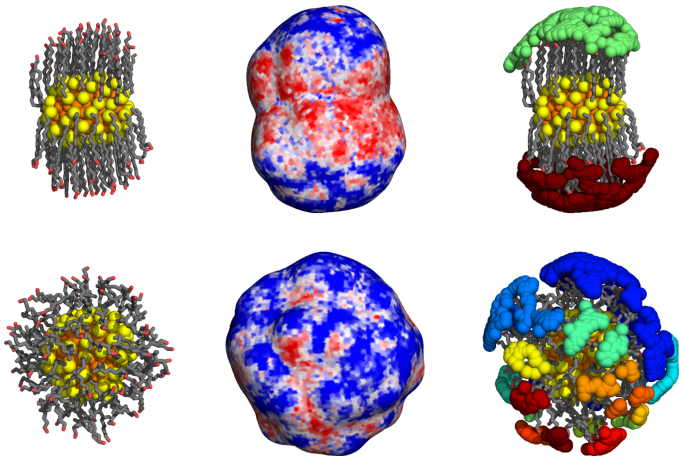


Figure 8.4: Representations of monolayer-protected gold nanoparticles from the work of Chew and co-workers [2, 3].

are not directly amenable for such tasks. To address this issue, we will leverage theoretical connections between topology, geometry, and statistics through *Random Field Theory* (RFT) [101]. Random fields are a generalization of stochastic processes to higher dimensions and the topology of their statistics can be characterized using TGDA. For example, in a collaboration with *Prof. Reid Van Lehn* (UW-Madison), we are exploring the properties of *monolayer-protected gold nanoparticles* through TGDA (Figure 8.4) [2, 3]. In this research, we are seeking to understand the behavior of these nanoparticles in various environments and how they interact with complex molecules (e.g., proteins). An important aspect of this research is to determine statistically-significant differences in how systems interact with various environments [2]. Here, we will apply ideas from TGDA to compare both the structure of the nanoparticles and their interactions with an environment (e.g., fluctuations at the water-nanoparticle interface) and adapt ideas from RFT to identify statistically significant differences between the nanoparticles based on their TGDA descriptors. Our broader research goal will be focused on consolidating connections between random field theory, statistics, topology, and geometry; establishing TGDA as a viable method for data analysis in the engineering community.

Appendix A

SUPPLEMENTARY INFORMATION

This appendix presents supplementary information from select sections of this dissertation.

A.1 Topology & Molecular Simulation

A.1.1 *Molecular dynamics simulation details*

All molecular dynamics (MD) trajectories analyzed in this work were previously generated and released publically. Data for the simulations of self-assembled monolayers (SAMs) were taken from [181] and data for the simulations of mixed-solvent environments were taken from [63]. Complete methodological details on the generation of these data sets are included in the source publications; here, we briefly summarize key details on the simulation procedures.

A.1.2 *Shared MD Parameters*

For both data sets, classical MD simulations were performed with a leapfrog integrator with a 2-fs timestep using the Gromacs 2016 simulation package [366]. Ligands (in the SAM systems) and reactants/cosolvents (in the mixed-solvent systems) were modeled

using the CGenFF/CHARMM36 force fields [367, 368]. Verlet lists were generated using a 1.2 nm neighbor list cutoff. Van der Waals interactions were modeled with a Lennard-Jones potential that was smoothly shifted to zero between 1.0 nm and 1.2 nm. Electrostatic interactions were calculated using the smooth Particle Mesh Ewald method with a short-range cutoff of 1.2 nm, grid spacing of 0.12 nm, and fourth-order interpolation. Bonds were constrained using the LINCS algorithm. Details on the conversion of MD trajectory output to hydrogen bond networks and density fields are included in the main text. To create hydrogen bond networks, hydrogen bonds between species were defined using standard Luzar-Chandler criteria as implemented in *gmx hbond* [190].

Simulations of Self-Assembled Monolayers

Previously generated MD trajectories for 50 SAMs (40 for training, 10 for testing) were obtained from the data set published in Ref. [181]. Each idealized SAM consisted of 64 alkanethiol ligands with either a hydroxyl (22 SAMs), amine (14 SAMs), or amide (14 SAMs) end group and with a backbone containing either 11, 12, or 13 methylene groups for the hydroxyl, amide, and amine end groups respectively. The partial charges of all ligand end group atoms were multiplied by a scaling factor, k, between 0 (most hydrophobic) to 1 (most hydrophilic) for each of the end groups. Each SAM was constructed by positioning the 64 ligands in the x-y plane to mimic self-assembly onto a gold (111) lattice with a grafting density of 21.6 $Å^2$/ligand [369]. Ligands were oriented with the end groups pointing in the positive z-direction. A 5-nm thick water layer was placed above the SAM and a 3-nm thick buffering vacuum layer was then added above the top of the water layer. Harmonic restraints with a spring constant of 50,000 kJ/mol/nm^2 were applied to the sulfur atoms to hold the SAM in place (gold atoms were not included). Water was modeled using the TIP4P/2005 water model. NVT production simulations were performed for 40 ns with configurations output every 1 ps. The temperature was maintained at 300 K using a velocity-rescale thermostat with a time constant of 0.1 ps. SAMs were labeled with hydration free energies (HFEs) computed using indirect umbrella sampling

(INDUS). All INDUS calculations used a 2.0×2.0×0.3 nm^3 cavity positioned with its base on a constant water density isosurface. INDUS simulations were performed using GROMACS 2016 patched with the PLUMED plugin (version 2.5.1) using the same runtime parameters as the unbiased simulations [189, 370]. Each system was equilibrated for 5 ns, then NVT INDUS simulations were performed using ≈13 windows per SAM. Each window was simulated for 5 ns, with the first 2 ns discarded as equilibration. Additional details are included in Ref. [181].

Simulations of Mixed-Solvent Environments

Previously generated MD trajectories for 108 reactant-solvent combinations (76 for training, 32 for testing) were obtained from the data set published in Ref. [63]. The initial simulation box dimensions were set to $(6\ nm)^3$ in all simulations, and water and cosolvent molecules were added in the desired proportions. Water was modeled using the Single Point Charge/Extended (SPC/E) model. The solvent system was equilibrated in a NPT simulation for 5 ns at T = 300 K and P = 1 bar with a velocity-rescale thermostat and Berendsen barostat. A single reactant molecule was added to the system and equilibrated with the same barostat and thermostat for 500 ps. NPT production simulations were then performed at the reaction temperature and P = 1 bar using a Nose-Hoover thermostat and Parrinello-Rahman barostat. All thermostats used a 1.0 ps time constant and all barostats used a 5.0 ps time constant with an isothermal compressibility of 5.0×10^{-5} bar^{-1}. Simulations used to generate the data in this work were performed for 2 ns with configurations output every 10 ps.

A.1.3 *Euler Characteristic Computation - Hydrogen Bonding Networks*

Hydrogen bonding networks can be represented as graphs $G(V, E)$, with vertices $v \in V$ and edges $\{v_i, v_j\} \in E$. Vertices represent individual molecules (v_i) in the simulation, and edges represent the presence of hydrogen bonds between two molecules $\{v_i, v_j\}$. A graph

is a two dimensional object. Thus, the Euler characteristic (EC) for a graph is given by the following equation:

$$\chi := \beta_0 - \beta_1 \tag{A.1}$$

where β_0 represents the number of connected components of the graph, and β_1 represents the number of cycles in the graph. Fortunately, the EC can also be computed for graphs using the following equation:

$$\chi = \beta_0 - \beta_1 = |V| - |E| \tag{A.2}$$

where $|V|$ represents the number of vertices in a graph $G(V, E)$ and $|E|$ represents the number of edges [371]. Thus, the computation of the EC for a given graphical representation of molecular simulation data is done by summing the total number of molecules within the simulation, and subtracting the total number of hydrogen bonds. This computation is illustrated in our supplementary code.

A.1.4 *Euler Characteristic Computation - Water Density Manifolds*

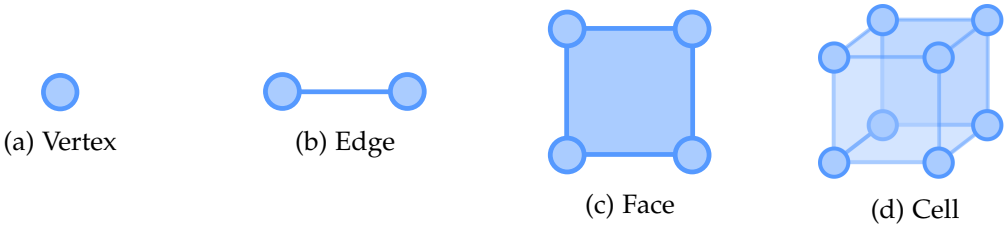
We construct water density manifolds directly from simulation data by averaging particle positions over a period of time within the simulation. From a computational standpoint, these manifolds are represented as 2-dimensional histograms in the Self-Assembled Monolayer (SAM) simulations and as 3-dimensional histograms in the acid-catalyzed reaction simulations. We focus on water particle positions during each of these simulations. Water particle positions are determined by the center-of-mass of the water molecule. In the SAM simulations the 2-dimensional histogram represents a 20×20 grid that covers the SAM surface interfacing with bulk water. Each grid point represents an area of 0.1

nm² on the SAM surface. For each simulation snapshot t we count the number of water molecules with centers of mass within each grid point $n^t_{i,j}$. We compute the time averaged water density value within each grid point ($\rho_{i,j}$) over 200 snapshots as follows:

$$\rho_{i,j} = \frac{1}{N} \sum_{t=1}^{200} n^t_{i,j} \tag{A.3}$$

where N represents the total number of water molecules accumulated over all bins in the 200 snapshots. Each point of the 2D grid is assigned a value $\rho_{i,j}$ which now represents a density field.

In order to perform a filtration and ultimately obtain an EC curve for this density field, we must first represent it as a *cubical complex*. A cubical complex is a set composed of n-dimensional cubes. In this work we are focused on vertices (0-dimensional), edges (1-dimensional), faces (2-dimensional) and cells (3-dimensional). Examples of these objects are found in Figure A.1. We note that all cubes of ($n > 0$)-dimension are built from lower-dimensional cubes (e.g., a face has four edges and each edge is supported at both ends by vertices).



(a) Vertex (b) Edge (c) Face (d) Cell

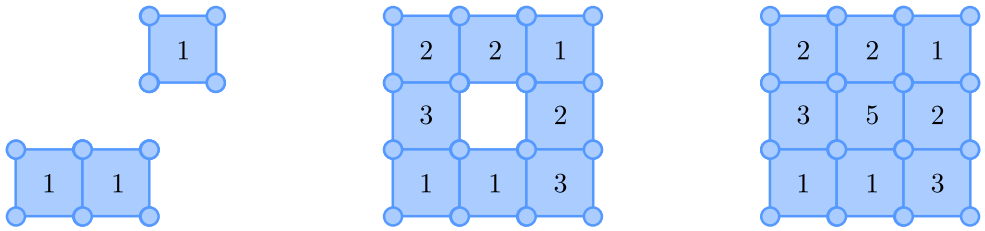
Figure A.1: Cubes of dimension 0 (vertex) to 3 (cell). These cubes can be combined to represent larger objects such as a density field histogram.

To perform a topological analysis of this density field we represent the field as a 20×20 grid of faces, where each face has an associated density value (similar to pixels in an image). The field is now a cubical complex built from 0, 1, and 2-dimensional cubes which can be quantified through the EC. We then construct a filtration where we treat the cubical complex as our manifold $\mathcal{M}$, and the density values assigned to each face as our

function $f : \mathcal{M} \to \mathbb{R}$. Through our filtration we can obtain sublevel sets $\mathcal{M}_l$ containing all faces with density value $\rho_{i,j} \leq l$ where $l \in \mathbb{R}$. There are two main ways to compute the EC for this cubical complex, the first leverages algebraic topology and computes the boundaries (i.e., cycles) within a cubical complex that are empty (e.g., they form a hole). Details of this method are intensive and require background in algebraic topology. This method is completely outlined, along with all supporting information, in our previous work [168]. Another, simpler method for this 2-dimensional cubical complex is similar to the graph example. We treat the cubical complex as a 2-dimensional polytope with vertices V, edges E, and faces F. From this, we can compute the EC as follows:

$$\chi = \beta_0 - \beta_1 = |V| - |E| + |F| \tag{A.4}$$

Where $|V|$ represents the number of vertices, $|E|$ represents the number of edges, and $|F|$ represents the number of faces [80]. Thus, at each point in our filtration we can simply count the number of vertices, edges, and faces, within the resulting sublevel sets $\mathcal{M}_l$ and obtain the EC (χ) directly. An example of this process is illustrated in Figure A.2 for a simple 3×3 density field represented as a cubical complex.



(a) $\chi(\mathcal{M}_1) = 10 - 11 + 3 = 2$ (b) $\chi(\mathcal{M}_3) = 16 - 24 + 8 = 0$ (c) $\chi(\mathcal{M}_5) = 16 - 24 + 9 = 1$

Figure A.2: Example filtration of a density field represented as a 2-dimensional cubical complex. At each sublevel set $\mathcal{M}_l$ we can compute the EC directly through the alternating sum of the number of vertices, edges, and faces.

We can follow the natural extension of this logic for our 3-dimensional density fields. These fields are constructed in the same way as our 2-dimensional density fields (e.g.,

counting and averaging particle positions within grid points), but with the primary difference being our field now has a third dimension. We segment our 3-dimensional volume into a $20 \times 20 \times 20$ grid with each grid point representing a cell with 0.2 nm^3 volume. We compute a density value at each grid point $\rho_{i,j,k}$ over 200 simulation snapshots t as follows:

$$\rho_{i,j,k} = \frac{1}{N} \sum_{t=1}^{200} n_{i,j,k}^{t} \tag{A.5}$$

where N represents the total number of water molecules accumulated over all bins in 20 snapshots. We can then represent this as a 3-dimensional cubical complex with associated density function. We treat the complex as a 3-dimensional polytope to compute the EC values during a filtration. The EC computation for a 3-dimensional polytope, where $|V|$ represents the number of vertices, $|E|$ represents the number of edges, $|F|$ represents the number of faces, and $|C|$ represents the number of cells, is as follows [80]:

$$\chi = \beta_0 - \beta_1 + \beta_2 = |V| - |E| + |F| - |C| \tag{A.6}$$

These computations can be made for our data representations directly and are used to construct the associated EC curves in our topological analysis of multiple MD simulations. This is outlined in the main text and demonstrated in the supporting code, in which we leverage the `GUDHI` (Geometry Understanding in Higher Dimensions) software [243].

BIBLIOGRAPHY

[1] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD–A comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9592–9600, 2019.

[2] Alex K Chew and Reid C Van Lehn. Effect of core morphology on the structural asymmetry of alkanethiol monolayer-protected gold nanoparticles. *The Journal of Physical Chemistry C*, 122(45):26288–26297, 2018.

[3] Alex K Chew, Bradley C Dallin, and Reid C Van Lehn. The interplay of ligand properties and core size dictates the hydrophobicity of monolayer-protected gold nanoparticles. *ACS nano*, 15(3):4534–4545, 2021.

[4] James J Downs and Ernest F Vogel. A plant-wide industrial process control problem. *Computers & chemical engineering*, 17(3):245–255, 1993.

[5] Elliot W Eisner. The promise and perils of alternative forms of data representation. *Educational researcher*, 26(6):4–10, 1997.

[6] Covariance Matrix, Marion R Reynolds Jr, and Gyo-Young Cho. Multivariate control charts for monitoring the mean vector and covariance matrix. *Journal of Quality Technology*, 38(3):230–253, 2006.

[7] Arthur B Yeh, Dennis KJ Lin, and Richard N McGrath. Multivariate control charts

for monitoring covariance matrix: a review. *Quality Technology & Quantitative Management*, 3(4):415–436, 2006.

[8] Anthony Hotz and Robert E Skelton. Covariance control theory. *International Journal of Control*, 46(1):13–32, 1987.

[9] RSH Mah and AC Tamhane. Detection of gross errors in process data. *AIChE Journal*, 28(5):828–830, 1982.

[10] Alexander Smith, Benjamin Laubach, Ivan Castillo, and Victor M Zavala. Data analysis using riemannian geometry and applications to chemical engineering. *arXiv preprint arXiv:2203.12471*, 2022.

[11] Ioan Andricioaei and Martin Karplus. On the calculation of entropy from covariance matrices of the atomic fluctuations. *The Journal of Chemical Physics*, 115(14):6289–6292, 2001.

[12] Steven Brunton, Bernd Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *arXiv preprint arXiv:1905.11075*, 2019.

[13] WJ Krzanowski, Philip Jonathan, WV McCarthy, and MR Thomas. Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 44(1):101–115, 1995.

[14] Jürgen Schlitter. Estimation of absolute and relative entropies of macromolecules using the covariance matrix. *Chemical physics letters*, 215(6):617–621, 1993.

[15] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[16] Ian Jolliffe. Principal component analysis. *Encyclopedia of statistics in behavioral science*, 2005.

[17] Manish Misra, H Henry Yue, S Joe Qin, and Cheng Ling. Multivariate process monitoring and fault diagnosis by multi-scale pca. *Computers & Chemical Engineering*, 26(9):1281–1293, 2002.

[18] Junghui Chen and Kun-Chih Liu. On-line batch process monitoring using dynamic pca and dynamic pls models. *Chemical Engineering Science*, 57(1):63–75, 2002.

[19] Leo H Chiang, Evan L Russell, and Richard D Braatz. Fault diagnosis in chemical processes using fisher discriminant analysis, discriminant partial least squares, and principal component analysis. *Chemometrics and intelligent laboratory systems*, 50(2):243–252, 2000.

[20] Seongkyu Yoon and John F MacGregor. Principal-component analysis of multiscale data for process monitoring and fault diagnosis. *AIChE Journal*, 50(11):2891–2903, 2004.

[21] Charles C David and Donald J Jacobs. Principal component analysis: a method for determining the essential dynamics of proteins. In *Protein dynamics*, pages 193–226. Springer, 2014.

[22] Tomokazu Konishi. Principal component analysis for designed experiments. *BMC bioinformatics*, 16(18):1–9, 2015.

[23] Alexander Smith, Andrea Keane, James A Dumesic, George W Huber, and Victor M Zavala. A machine learning framework for the analysis and prediction of catalytic activity from experimental data. *Applied Catalysis B: Environmental*, 263:118257, 2020.

[24] Isao Sasaki, Hiroshi Tsuchiya, Masateru Nishioka, Masayoshi Sadakata, and Tatsuya Okubo. Gas sensing with zeolite-coated quartz crystal microbalances—principal component analysis approach. *Sensors and Actuators B: Chemical*, 86(1):26–33, 2002.

[25] Alexandros Altis, Phuong H Nguyen, Rainer Hegger, and Gerhard Stock. Dihedral

angle principal component analysis of molecular dynamics simulations. *The Journal of chemical physics*, 126(24):244111, 2007.

[26] Elena Papaleo, Paolo Mereghetti, Piercarlo Fantucci, Rita Grandori, and Luca De Gioia. Free-energy landscape, principal component analysis, and structural clustering to identify representative conformations from molecular dynamics simulations: the myoglobin case. *Journal of molecular graphics and modelling*, 27(8):889–899, 2009.

[27] Sarah A Mueller Stein, Anne E Loccisano, Steven M Firestine, and Jeffrey D Evanseck. Principal components analysis: a review of its application on molecular dynamics data. *Annual Reports in Computational Chemistry*, 2:233–261, 2006.

[28] Florian Sittel, Abhinav Jain, and Gerhard Stock. Principal component analysis of molecular dynamics: On the use of cartesian vs. internal coordinates. *The Journal of Chemical Physics*, 141(1):07B605_1, 2014.

[29] Alexander Smith, Spencer Runde, Alex Chew, Atharva Kelkar, Utkarsh Maheshwari, Reid Van Lehn, and Victor Zavala. Topological analysis of molecular dynamics simulations using the euler characteristic. 2022.

[30] Jordan Jalving, Yankai Cao, and Victor M Zavala. Graph-based modeling and simulation of complex systems. *Computers & Chemical Engineering*, 125:134–154, 2019.

[31] Alexander D Smith and Victor M Zavala. The Euler characteristic: A general topological descriptor for complex data. *Computers & Chemical Engineering*, 154:107463, 2021.

[32] Jordan Jalving, Sungho Shin, and Victor M Zavala. A graph-based modeling abstraction for optimization: Concepts and implementation in plasmo. jl. *arXiv preprint arXiv:2006.05378*, 2020.

[33] Sungho Shin, Mihai Anitescu, and Victor M Zavala. Exponential decay of sensitivity in graph-structured nonlinear programs. *SIAM Journal on Optimization*, 32(2):1156–1183, 2022.

[34] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

[35] Ivan Gutman and Ernesto Estrada. Topological indices based on the line graph of the molecular graph. *Journal of chemical information and computer sciences*, 36(3):541–543, 1996.

[36] Xiaofeng Wang, Zhen Li, Mingjian Jiang, Shuang Wang, Shugang Zhang, and Zhiqiang Wei. Molecule property prediction based on spatial graph embedding. *Journal of chemical information and modeling*, 59(9):3817–3828, 2019.

[37] Prodromos Daoutidis, Wentao Tang, and Andrew Allman. Decomposition of control and optimization problems by network structure: Concepts, methods, and inspirations from biology. *AIChE Journal*, 65(10):e16708, 2019.

[38] Wentao Tang, Andrew Allman, Davood Babaei Pourkargar, and Prodromos Daoutidis. Optimal decomposition for distributed optimization in nonlinear model predictive control through community detection. *Computers & Chemical Engineering*, 111:43–54, 2018.

[39] Shiyi Qin, Tianyi Jin, Reid C Van Lehn, and Victor M Zavala. Predicting critical micelle concentrations for surfactants using graph convolutional neural networks. *The Journal of Physical Chemistry B*, 125(37):10610–10620, 2021.

[40] Shiyi Qin, Shengli Jiang, Jianping Li, Prasanna Balaprakash, Reid Van Lehn, and Victor Zavala. Capturing molecular interactions in graph neural networks: A case study in multi-component phase equilibrium. 2022.

[41] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[42] Connor W Coley, Wengong Jin, Luke Rogers, Timothy F Jamison, Tommi S Jaakkola, William H Green, Regina Barzilay, and Klavs F Jensen. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical science*, 10(2):370–377, 2019.

[43] Vincenzo Nicosia, John Tang, Cecilia Mascolo, Mirco Musolesi, Giovanni Russo, and Vito Latora. Graph metrics for temporal networks. In *Temporal networks*, pages 15–40. Springer, 2013.

[44] DJ Klein. Graph geometry, graph metrics and wiener. *MATCH Commun. Math. Comput. Chem*, 35(7):27, 1997.

[45] Peter Wills and François G Meyer. Metrics for graph comparison: a practitioner's guide. *Plos one*, 15(2):e0228728, 2020.

[46] Pavel Chebotarev. Studying new classes of graph metrics. In *International Conference on Geometric Science of Information*, pages 207–214. Springer, 2013.

[47] Ronald Newbold Bracewell and Ronald N Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York, 1986.

[48] Robert Bell. *Introductory Fourier transform spectroscopy*. Elsevier, 2012.

[49] M Carbonaro and A Nucara. Secondary structure of food proteins by fourier transform spectroscopy in the mid-infrared region. *Amino acids*, 38(3):679–690, 2010.

[50] Gerald Steiner and Edmund Koch. Trends in fourier transform infrared spectroscopic imaging. *Analytical and bioanalytical chemistry*, 394(3):671–678, 2009.

[51] M Olga Guerrero-Pérez and Gregory S Patience. Experimental methods in chemical engineering: Fourier transform infrared spectroscopy—ftir. *The Canadian Journal of Chemical Engineering*, 98(1):25–33, 2020.

[52] Terry M Peters and Jacqueline C Williams. *The fourier transform in biomedical engineering*. Springer Science & Business Media, 1998.

[53] Ajit P Yoganathan, Ramesh Gupta, and William H Corcoran. Fast fourier transform in the analysis of biomedical data. *Medical and biological engineering*, 14(2):239–245, 1976.

[54] Alexander D Smith, Nicholas Abbott, and Victor M Zavala. Convolutional network analysis of optical micrographs for liquid crystal sensors. *The Journal of Physical Chemistry C*, 124(28):15152–15161, 2020.

[55] Shengli Jiang, Zhuo Xu, Medhavi Kamran, Stas Zinchik, Sidike Paheding, Armando G McDonald, Ezra Bar-Ziv, and Victor M Zavala. Using atr-ftir spectra and convolutional neural networks for characterizing mixed plastic waste. *Computers & Chemical Engineering*, 155:107547, 2021.

[56] Yankai Cao, Huaizhe Yu, Nicholas L Abbott, and Victor M Zavala. Machine learning algorithms for liquid crystal-based sensors. *ACS sensors*, 3(11):2237–2245, 2018.

[57] Stas Zinchik, Shengli Jiang, Søren Friis, Fei Long, Lasse Høgstedt, Victor M Zavala, and Ezra Bar-Ziv. Accurate characterization of mixed plastic waste using machine learning and fast infrared spectroscopy. *ACS Sustainable Chemistry & Engineering*, 9(42):14143–14151, 2021.

[58] Yushi Chen, Lin Zhu, Pedram Ghamisi, Xiuping Jia, Guoyu Li, and Liang Tang. Hyperspectral images classification with gabor filtering and convolutional neural network. *IEEE Geoscience and Remote Sensing Letters*, 14(12):2355–2359, 2017.

[59] Wei Hu, Yangyu Huang, Li Wei, Fan Zhang, and Hengchao Li. Deep convolutional neural networks for hyperspectral image classification. *Journal of Sensors*, 2015, 2015.

[60] Zhuo Cao, Yabo Dan, Zheng Xiong, Chengcheng Niu, Xiang Li, Songrong Qian, and Jianjun Hu. Convolutional neural networks for crystal material property prediction using hybrid orbital-field matrix and magpie descriptors. *Crystals*, 9(4):191, 2019.

[61] Matthew Ragoza, Joshua Hochuli, Elisa Idrobo, Jocelyn Sunseri, and David Ryan Koes. Protein–ligand scoring with convolutional neural networks. *Journal of chemical information and modeling*, 57(4):942–957, 2017.

[62] Shengli Jiang and Victor M Zavala. Convolutional neural nets in chemical engineering: Foundations, computations, and applications. *AIChE Journal*, 67(9):e17282, 2021.

[63] Alex K Chew, Shengli Jiang, Weiqi Zhang, Victor M Zavala, and Reid C Van Lehn. Fast predictions of liquid-phase acid-catalyzed reaction rates using molecular dynamics simulations and convolutional neural networks. *Chemical science*, 11(46):12464–12476, 2020.

[64] Salvatore Torquato and HW Haslach Jr. Random heterogeneous materials: microstructure and macroscopic properties. *Appl. Mech. Rev.*, 55(4):B62–B63, 2002.

[65] Peter I Ravikovitch, Aleksey Vishnyakov, and Alexander V Neimark. Density functional theories and molecular simulations of adsorption and phase transitions in nanopores. *Physical Review E*, 64(1):011602, 2001.

[66] James G Berryman and Stephen C Blair. Use of digital image analysis to estimate fluid permeability of porous materials: Application of two-point correlation functions. *Journal of applied Physics*, 60(6):1930–1938, 1986.

[67] M Baniassadi, S Ahzi, H Garmestani, D Ruch, and Y Remond. New approximate

solution for n-point correlation functions for heterogeneous materials. *Journal of the Mechanics and Physics of Solids*, 60(1):104–119, 2012.

[68] Peter B Corson. Correlation functions for predicting properties of heterogeneous materials. i. experimental measurement of spatial correlation functions in multiphase solids. *Journal of applied Physics*, 45(7):3159–3164, 1974.

[69] G Saheli, H Garmestani, and BL Adams. Microstructure design of a two phase composite using two-point correlation functions. *Journal of computer-aided materials design*, 11(2):103–115, 2004.

[70] Hubert Mantz, Karin Jacobs, and Klaus Mecke. Utilizing minkowski functionals for image analysis: a marching square algorithm. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(12):P12015, 2008.

[71] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

[72] Boaz Nadler, Stéphane Lafon, Ronald R Coifman, and Ioannis G Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis*, 21(1):113–127, 2006.

[73] Andrew L Ferguson, Athanassios Z Panagiotopoulos, Pablo G Debenedetti, and Ioannis G Kevrekidis. Systematic determination of order parameters for chain dynamics using diffusion maps. *Proceedings of the National Academy of Sciences*, 107(31):13597–13602, 2010.

[74] Andrew L Ferguson, Athanassios Z Panagiotopoulos, Ioannis G Kevrekidis, and Pablo G Debenedetti. Nonlinear dimensionality reduction in molecular simulation: The diffusion map approach. *Chemical Physics Letters*, 509(1-3):1–11, 2011.

[75] Sang Beom Kim, Carmeline J Dsilva, Ioannis G Kevrekidis, and Pablo G Debenedetti. Systematic characterization of protein folding pathways using dif-

fusion maps: Application to trp-cage miniprotein. *The Journal of chemical physics*, 142(8):02B613_1, 2015.

[76] Jiang Hu, Xin Liu, Zai-Wen Wen, and Ya-Xiang Yuan. A brief introduction to manifold optimization. *Journal of the Operations Research Society of China*, 8(2):199–248, 2020.

[77] Suvrit Sra and Reshad Hosseini. Conic geometric optimization on the manifold of positive definite matrices. *SIAM Journal on Optimization*, 25(1):713–739, 2015.

[78] Steven T Smith. Optimization techniques on riemannian manifolds. *Fields institute communications*, 3(3):113–135, 1994.

[79] Constantin Udriste. *Convex functions and optimization methods on Riemannian manifolds*, volume 297. Springer Science & Business Media, 2013.

[80] Daniel A Klain and Gian-Carlo Rota. *Introduction to geometric probability*. Cambridge University Press, 1997.

[81] Klaus R Mecke. Additivity, convexity, and beyond: applications of minkowski functionals in statistical physics. In *Statistical Physics and Spatial Statistics*, pages 111–184. Springer, 2000.

[82] Barry M McCoy and Tai Tsun Wu. *The two-dimensional Ising model*. Courier Corporation, 2014.

[83] Stephen G Brush. History of the lenz-ising model. *Reviews of modern physics*, 39(4):883, 1967.

[84] Barry A Cipra. An introduction to the ising model. *The American Mathematical Monthly*, 94(10):937–959, 1987.

[85] P-M König, R Roth, and KR Mecke. Morphological thermodynamics of fluids: shape dependence of free energies. *Physical Review Letters*, 93(16):160601, 2004.

[86] Daniel A Klain. A short proof of hadwiger's characterization theorem. *Mathematika*, 42(2):329–339, 1995.

[87] Klaus R Mecke. A morphological model for complex fluids. *Journal of Physics: Condensed Matter*, 8(47):9663, 1996.

[88] Tristan Needham. *Visual Differential Geometry and Forms: A Mathematical Drama in Five Acts*. Princeton University Press, 2021.

[89] Claudio Bonati. The peierls argument for higher dimensional ising models. *European Journal of Physics*, 35(3):035002, 2014.

[90] Roberto Fernández, Pablo A Ferrari, and Nancy L Garcia. Loss network representation of peierls contours. *Annals of Probability*, pages 902–937, 2001.

[91] Andreas Wipf. Peierls argument and duality transformations. In *Statistical Approach to Quantum Field Theory*, pages 231–258. Springer, 2021.

[92] Yoshikiyo Moroi. *Micelles: theoretical and applied aspects*. Springer Science & Business Media, 1992.

[93] Mehdi Bouzid and Emanuela Del Gado. Network topology in soft gels: Hardening and softening materials. *Langmuir*, 34(3):773–781, 2018.

[94] Klaus R Mecke and Dietrich Stoyan. *Statistical physics and spatial statistics: the art of analyzing and modeling spatial structures and pattern formation*, volume 554. Springer Science & Business Media, 2000.

[95] Klaus Mecke and CH Arns. Fluids in porous media: a morphometric approach. *Journal of Physics: Condensed Matter*, 17(9):S503, 2005.

[96] Klaus R Mecke, Thomas Buchert, and Herbert Wagner. Robust morphological measures for large-scale structure in the universe. *arXiv preprint astro-ph/9312028*, 1993.

[97] Alexander D Smith, Paweł Dłotko, and Victor M Zavala. Topological data analysis: Concepts, computation, and applications in chemical engineering. *Computers & Chemical Engineering*, page 107202, 2020.

[98] Elizabeth Munch. A user's guide to topological data analysis. *Journal of Learning Analytics*, 4(2):47–61, 2017.

[99] Leonhard Euler. Elementa doctrinae solidorum. *Novi commentarii academiae scientiarum Petropolitanae*, pages 109–140, 1758.

[100] Robert J Adler. Some new random field tools for spatial analysis. *Stochastic Environmental Research and Risk Assessment*, 22(6):809–822, 2008.

[101] Robert J Adler. *The geometry of random fields*. SIAM, 2010.

[102] Robert J Adler and Jonathan E Taylor. *Random fields and geometry*. Springer Science & Business Media, 2009.

[103] Jens Schmalzing and Krzysztof M Górski. Minkowski functionals used in the morphological analysis of cosmic microwave background anisotropy maps. *Monthly Notices of the Royal Astronomical Society*, 297(2):355–365, 1998.

[104] Pratyush Pranav, Rien Van de Weygaert, Gert Vegter, Bernard JT Jones, Robert J Adler, Job Feldbrugge, Changbom Park, Thomas Buchert, and Michael Kerber. Topology and geometry of gaussian random fields i: on betti numbers, euler characteristic, and minkowski functionals. *Monthly Notices of the Royal Astronomical Society*, 485(3):4167–4208, 2019.

[105] M Kerscher, K Mecke, J Schmalzing, C Beisbart, Th Buchert, and H Wagner. Morphological fluctuations of large-scale structure: The pscz survey. *Astronomy & Astrophysics*, 373(1):1–11, 2001.

[106] Christoph H Arns, Mark A Knackstedt, and KR Mecke. Reconstructing complex materials via effective grain shapes. *Physical Review Letters*, 91(21):215506, 2003.

[107] Sung Nok Chiu, Dietrich Stoyan, Wilfrid S Kendall, and Joseph Mecke. *Stochastic geometry and its applications*. John Wiley & Sons, 2013.

[108] Christian Scholz, Frank Wirner, Jan Götz, Ulrich Rüde, Gerd E Schröder-Turk, Klaus Mecke, and Clemens Bechinger. Permeability of porous materials determined from the euler characteristic. *Physical review letters*, 109(26):264504, 2012.

[109] KR Mecke. Morphological thermodynamics of composite media. *Fluid Phase Equilibria*, 150:591–598, 1998.

[110] Hamid Hosseinzade Khanamiri, Carl Fredrik Berg, Per Arne Slotte, Steffen Schlüter, and Ole Torsæter. Description of free energy for immiscible two-fluid flow in porous media by integral geometry and thermodynamics. *Water Resources Research*, 54(11):9045–9059, 2018.

[111] Hendrik Hansen-Goos, Roland Roth, Klaus Mecke, and S Dietrich. Solvation of proteins: linking thermodynamics to geometry. *Physical review letters*, 99(12):128101, 2007.

[112] James M Kilner, Stefan J Kiebel, and Karl J Friston. Applications of random field theory to electrophysiology. *Neuroscience letters*, 374(3):174–178, 2005.

[113] Hyekyoung Lee, Moo K Chung, Hyejin Kang, Bung-Nyun Kim, and Dong Soo Lee. Discriminative persistent homology of brain networks. In *2011 IEEE international symposium on biomedical imaging: from nano to macro*, pages 841–844. IEEE, 2011.

[114] James E McClure, Thomas Ramstad, Zhe Li, Ryan T Armstrong, and Steffen Berg. Modeling geometric state for fluids in porous media: Evolution of the euler characteristic. *Transport in Porous Media*, 133:229–250, 2020.

[115] Eitan Richardson and Michael Werman. Efficient classification using the euler characteristic. *Pattern Recognition Letters*, 49:99–106, 2014.

[116] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169–2178. IEEE, 2006.

[117] Yang Jiao, FH Stillinger, and S Torquato. Modeling heterogeneous materials via two-point correlation functions: Basic principles. *Physical Review E*, 76(3):031110, 2007.

[118] James Munkres. *Topology*. Pearson Education, 2014.

[119] RJ Adler, O Bobrowski, M Borman, E Subag, and S Weinberger. Borrowing strength: Theory powering applications–a festschrift for lawrence d. *Brown. Beachwood: Institute of Mathematical Statistics*, pages 124–143, 2010.

[120] Jonathan E Taylor, Robert J Adler, et al. Euler characteristics for gaussian fields on manifolds. *Annals of Probability*, 31(2):533–563, 2003.

[121] Ruth Charney and Michael Davis. The euler characteristic of a nonpositively curved, piecewise euclidean manifold. *Pacific Journal of Mathematics*, 171(1):117–137, 1995.

[122] Thomas Letendre. Expected volume and euler characteristic of random submanifolds. *Journal of Functional Analysis*, 270(8):3047–3110, 2016.

[123] Henri Poincaré. *Analysis situs*. Gauthier-Villars Paris, France, 1895.

[124] Merab Menabde, Alan Seed, Daniel Harris, and Geoff Austin. Self-similar random fields and rainfall simulation. *Journal of Geophysical Research: Atmospheres*, 102(D12):13509–13515, 1997.

[125] Gordon A Fenton. Random field modeling of cpt data. *Journal of geotechnical and geoenvironmental engineering*, 125(6):486–498, 1999.

[126] George Christakos. *Random field models in earth sciences*. Courier Corporation, 2012.

[127] Ilya Lyashenko, Vladimir Pugach, and Sergiy Skuratovskyi. Modelling of phase distortion in earth atmosphere with a correlated random field. In *2020 IEEE Ukrainian Microwave Week (UkrMW)*, pages 751–756. IEEE, 2020.

[128] Keith J Worsley. Boundary corrections for the expected euler characteristic of excursion sets of random fields, with an application to astrophysics. *Advances in Applied Probability*, pages 943–959, 1995.

[129] Thomas E Nichols. Multiple testing corrections, nonparametric methods, and random field theory. *Neuroimage*, 62(2):811–815, 2012.

[130] Matthew Brett, Will Penny, and Stefan Kiebel. Introduction to random field theory. *Human brain function*, 2, 2003.

[131] K Worsley. Random field theory. *Statistical parametric mapping: the analysis of functional brain images*, pages 232–245, 2007.

[132] Keith J Worsley, Jonathan E Taylor, Francesco Tomaiuolo, and Jason Lerch. Unified univariate and multivariate random field theory. *Neuroimage*, 23:S189–S195, 2004.

[133] Robert J Adler, Jonathan E Taylor, Keith J Worsley, and Keith Worsley. Applications of random fields and geometry: Foundations and case studies. In *In preparation, available on R. Adler's home*. Citeseer, 2007.

[134] Katharine Turner, Yuriy Mileyko, Sayan Mukherjee, and John Harer. Fréchet means for distributions of persistence diagrams. *Discrete & Computational Geometry*, 52(1):44–70, 2014.

[135] Christoph D Hofer, Roland Kwitt, and Marc Niethammer. Learning representations of persistence barcodes. *Journal of Machine Learning Research*, 20(126):1–45, 2019.

[136] Hilary Richardson, Grace Lisandrelli, Alexa Riobueno-Naylor, and Rebecca Saxe. "mri data of 3-12 year old children and adults during viewing of a short animated film". 2018.

[137] Moo K Chung, Hyekyoung Lee, Alex DiChristofano, Hernando Ombao, and Victor Solo. Exact topological inference of the resting-state brain networks in twins. *Network Neuroscience*, 3(3):674–694, 2019.

[138] Moo K Chung, Alexander Smith, and Gary Shiu. Reviews: Topological distances and losses for brain networks. *arXiv preprint arXiv:2102.08623*, 2021.

[139] Tibor Szilvási, Nanqi Bao, Karthik Nayani, Huaizhe Yu, Prabin Rai, Robert J Twieg, Manos Mavrikakis, and Nicholas L Abbott. Redox-triggered orientational responses of liquid crystals to chlorine gas. *Angewandte Chemie*, 130(31):9813–9817, 2018.

[140] Shani L Levit, Jimmy Nguyen, Nicholas P Hattrup, Briget E Rabatin, Ratib Stwodah, Christopher L Vasey, Michael P Zeevi, McKenna Gillard, Paola A D'Angelo, Kathleen W Swana, et al. Color space transformation-based algorithm for evaluation of thermochromic behavior of cholesteric liquid crystals using polarized light microscopy. *ACS omega*, 5(13):7149–7157, 2020.

[141] Fei Ye, Zhiping Shi, and Zhongzhi Shi. A comparative study of pca, lda and kernel lda for image classification. In *2009 International Symposium on Ubiquitous Virtual Reality*, pages 51–54. IEEE, 2009.

[142] Jae S Lim. Two-dimensional signal and image processing. *Englewood Cliffs*, 1990.

[143] Hongfei Li, Catherine A Calder, and Noel Cressie. Beyond moran's i: testing for spatial dependence based on the spatial autoregressive model. *Geographical Analysis*, 39(4):357–375, 2007.

[144] Eric Walter and Luc Pronzato. Identification of parametric models. *Communications and control engineering*, 8, 1997.

[145] Josef Spidlen, Karin Breuer, Chad Rosenberg, Nikesh Kotecha, and Ryan R Brinkman. Flowrepository: A resource of annotated flow cytometry datasets associated with peer-reviewed publications. *Cytometry Part A*, 81(9):727–731, 2012.

[146] Dennis Van Hoof, Woodrow Lomas, Mary Beth Hanley, and Emily Park. Simultaneous flow cytometric analysis of ifn-γ and cd4 mrna and protein expression kinetics in human peripheral blood mononuclear cells during activation. *Cytometry Part A*, 85(10):894–900, 2014.

[147] Kenneth Lo, Ryan Remy Brinkman, and Raphael Gottardo. Automated gating of flow cytometry data via robust model-based clustering. *Cytometry Part A: the journal of the International Society for Analytical Cytology*, 73(4):321–332, 2008.

[148] Simon J Sheather. Density estimation. *Statistical science*, pages 588–597, 2004.

[149] Sander Pronk, Szilárd Páll, Roland Schulz, Per Larsson, Pär Bjelkmar, Rossen Apostolov, Michael R Shirts, Jeremy C Smith, Peter M Kasson, David Van Der Spoel, et al. Gromacs 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics*, 29(7):845–854, 2013.

[150] S Doerr, MJ Harvey, Frank Noé, and G De Fabritiis. Htmd: high-throughput molecular dynamics for molecular discovery. *Journal of chemical theory and computation*, 12(4):1845–1852, 2016.

[151] Matthew J Harvey and Gianni De Fabritiis. High-throughput molecular dynamics: the powerful new tool for drug discovery. *Drug discovery today*, 17(19-20):1059–1062, 2012.

[152] N Scott Bobbitt and Randall Q Snurr. Molecular modelling and machine learning for high-throughput screening of metal-organic frameworks for hydrogen storage. *Molecular Simulation*, 45(14-15):1069–1081, 2019.

[153] Christine Peter and Kurt Kremer. Multiscale simulation of soft matter systems. *Faraday discussions*, 144:9–24, 2010.

[154] Norbert Attig, Kurt Binder, Helmut Grubmüller, and Kurt Kremer. Computational soft matter: from synthetic polymers to proteins. *John von Neumann Institute for Computing (NIC), Juelich*, 2004.

[155] Harold A Scheraga, Mey Khalili, and Adam Liwo. Protein-folding dynamics: overview of molecular simulation techniques. *Annu. Rev. Phys. Chem.*, 58:57–83, 2007.

[156] Mario Orsi. Molecular simulation of self-assembly. In *Self-assembling Biomaterials*, pages 305–318. Elsevier, 2018.

[157] Lisa Je, George W Huber, Reid C Van Lehn, and Victor M Zavala. On the integration of molecular dynamics, data science, and experiments for studying solvent effects on catalysis. *Current Opinion in Chemical Engineering*, 36:100796, 2022.

[158] Robert T McGibbon, Kyle A Beauchamp, Matthew P Harrigan, Christoph Klein, Jason M Swails, Carlos X Hernández, Christian R Schwantes, Lee-Ping Wang, Thomas J Lane, and Vijay S Pande. Mdtraj: a modern open library for the analysis of molecular dynamics trajectories. *Biophysical journal*, 109(8):1528–1532, 2015.

[159] Wolfgang Lechner and Christoph Dellago. Accurate determination of crystal structures based on averaged local bond order parameters. *The Journal of chemical physics*, 129(11):114707, 2008.

[160] Wilfred F van Gunsteren, Jožica Dolenc, and Alan E Mark. Molecular simulation as an aid to experimentalists. *Current opinion in structural biology*, 18(2):149–153, 2008.

[161] Jamshed Anwar and Dirk Zahn. Uncovering molecular processes in crystal nucleation and growth by using molecular simulation. *Angewandte Chemie International Edition*, 50(9):1996–2013, 2011.

[162] Michael P Allen. Molecular simulation and theory of the isotropic–nematic interface. *The Journal of Chemical Physics*, 112(12):5447–5453, 2000.

[163] Frank Noé, Alexandre Tkatchenko, Klaus-Robert Müller, and Cecilia Clementi. Machine learning for molecular simulation. *Annual review of physical chemistry*, 71:361–390, 2020.

[164] Wen Torng and Russ B Altman. High precision protein functional site detection using 3d convolutional neural networks. *Bioinformatics*, 35(9):1503–1512, 2019.

[165] José Jiménez, Stefan Doerr, Gerard Martínez-Rosell, Alexander S Rose, and Gianni De Fabritiis. Deepsite: protein-binding site predictor using 3d-convolutional neural networks. *Bioinformatics*, 33(19):3036–3042, 2017.

[166] José Jiménez, Miha Skalic, Gerard Martinez-Rosell, and Gianni De Fabritiis. K deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *Journal of chemical information and modeling*, 58(2):287–296, 2018.

[167] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

[168] Alexander D Smith, Paweł Dłotko, and Victor M Zavala. Topological data analysis: concepts, computation, and applications in chemical engineering. *Computers & Chemical Engineering*, 146:107202, 2021.

[169] Kate E Horner, Mark A Miller, Jonathan W Steed, and Paul M Sutcliffe. Knot theory in modern chemistry. *Chemical Society Reviews*, 45(23):6432–6448, 2016.

[170] DW Sumners. The knot theory of molecules. *Journal of mathematical chemistry*, 1(1):1–14, 1987.

[171] Chengzhi Liang and Kurt Mislow. Knots in proteins. *Journal of the American Chemical Society*, 116(24):11189–11190, 1994.

[172] Søren S Sørensen, Christophe AN Biscio, Mathieu Bauchy, Lisbeth Fajstrup, and Morten M Smedskjaer. Revealing hidden medium-range order in amorphous materials using topological data analysis. *Science Advances*, 6(37):eabc2320, 2020.

[173] Lee Steinberg, John Russo, and Jeremy Frey. A new topological descriptor for water network structure. *Journal of cheminformatics*, 11(1):1–11, 2019.

[174] Wai Shing Tang, Gabriel Monteiro da Silva, Henry Kirveslahti, Erin Skeens, Bibo Feng, Timothy Sudijono, Kevin K Yang, Sayan Mukherjee, Brenda Rubenstein, and Lorin Crawford. A topological data analytic approach for discovering biophysical signatures in protein dynamics. *PLoS computational biology*, 18(5):e1010045, 2022.

[175] Yongjin Lee, Senja D Barthel, Paweł Dłotko, Seyed Mohamad Moosavi, Kathryn Hess, and Berend Smit. High-throughput screening approach for nanoporous materials genome using topological data analysis: application to zeolites. *Journal of chemical theory and computation*, 14(8):4427–4437, 2018.

[176] Mariam Pirashvili, Lee Steinberg, Francisco Belchi Guillamon, Mahesan Niranjan, Jeremy G Frey, and Jacek Brodzki. Improved understanding of aqueous solubility modeling through topological data analysis. *Journal of cheminformatics*, 10(1):1–14, 2018.

[177] Chad M Topaz, Lori Ziegelmeier, and Tom Halverson. Topological data analysis of biological aggregation models. *PloS one*, 10(5):e0126383, 2015.

[178] Joel Bernstein, Raymond E Davis, Liat Shimoni, and Ning-Leh Chang. Patterns in hydrogen bonding: functionality and graph set analysis in crystals. *Angewandte Chemie International Edition in English*, 34(15):1555–1573, 1995.

[179] George A Jeffrey. Hydrogen-bonding: an update. *Crystallography Reviews*, 9(2-3):135–176, 2003.

[180] Pablo Gainza, Freyr Sverrisson, Frederico Monti, Emanuele Rodola, D Boscaini, MM Bronstein, and BE Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020.

[181] Atharva S Kelkar, Bradley C Dallin, and Reid C Van Lehn. Predicting hydrophobicity by learning spatiotemporal features of interfacial water structure: Combining molecular dynamics simulations with convolutional neural networks. *The Journal of Physical Chemistry B*, 124(41):9103–9114, 2020.

[182] Theodore W Walker, Alex K Chew, Huixiang Li, Benginur Demir, Z Conrad Zhang, George W Huber, Reid C Van Lehn, and James A Dumesic. Universal kinetic solvent effects in acid-catalyzed reactions of biomass-derived oxygenates. *Energy & Environmental Science*, 11(3):617–628, 2018.

[183] Shannon McDonald, Lars Ojamäe, and Sherwin J Singer. Graph theoretical generation and analysis of hydrogen-bonded structures with applications to the neutral and protonated water cube and dodecahedral clusters. *The Journal of Physical Chemistry A*, 102(17):2824–2832, 1998.

[184] TP Radhakrishnan and William C Herndon. Graph theoretical analysis of water clusters. *The Journal of Physical Chemistry*, 95(26):10609–10617, 1991.

[185] VM Gun'ko, VV Turov, VM Bogatyrev, VI Zarko, R Leboda, EV Goncharuk, AA Novza, AV Turov, and AA Chuiko. Unusual properties of water at hydrophilic/hydrophobic interfaces. *Advances in Colloid and Interface Science*, 118(1-3):125–172, 2005.

[186] Peter G Kusalik and Igor M Svishchev. The spatial structure in liquid water. *Science*, 265(5176):1219–1221, 1994.

[187] Jayendran C Rasaiah, Shekhar Garde, and Gerhard Hummer. Water in nonpolar confinement: From nanotubes to proteins and beyond. *Annu. Rev. Phys. Chem.*, 59:713–740, 2008.

[188] Nicholas B Rego and Amish J Patel. Understanding hydrophobic effects: Insights from water density fluctuations. *Annual Review of Condensed Matter Physics*, 13:303–324, 2022.

[189] Amish J Patel, Patrick Varilly, David Chandler, and Shekhar Garde. Quantifying density fluctuations in volumes of all shapes and sizes using indirect umbrella sampling. *Journal of statistical physics*, 145(2):265–275, 2011.

[190] Alenka Luzar and David Chandler. Hydrogen-bond kinetics in liquid water. *Nature*, 379(6560):55–57, 1996.

[191] Natalia Shenogina, Pawel Keblinski, and Shekhar Garde. Strong frequency dependence of dynamical coupling between protein and water. *The Journal of chemical physics*, 129(15):10B614, 2008.

[192] Afra Zomorodian. Topological data analysis. *Advances in applied and computational topology*, 70:1–39, 2012.

[193] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[194] Ramūnas Valiokas, Mattias Östblom, Sofia Svedhem, Stefan CT Svensson, and Bo Liedberg. Thermal stability of self-assembled monolayers: Influence of lateral hydrogen bonding. *The Journal of Physical Chemistry B*, 106(40):10401–10409, 2002.

[195] Penelope A Lewis, Rachel K Smith, Kevin F Kelly, Lloyd A Bumm, Scott M Reed, Robert S Clegg, John D Gunderson, James E Hutchison, and Paul S Weiss. The role of buried hydrogen bonds in self-assembled mixed composition thiols on au $\{111\}$. *The Journal of Physical Chemistry B*, 105(43):10630–10636, 2001.

[196] Robert S Clegg, Scott M Reed, and James E Hutchison. Self-assembled monolayers stabilized by three-dimensional networks of hydrogen bonds. *Journal of the American Chemical Society*, 120(10):2486–2487, 1998.

[197] Samir H Mushrif, Stavros Caratzoulas, and Dionisios G Vlachos. Understanding solvent effects in the selective conversion of fructose to 5-hydroxymethyl-furfural: a molecular dynamics investigation. *Physical Chemistry Chemical Physics*, 14(8):2637–2644, 2012.

[198] Jithin John Varghese and Samir H Mushrif. Origins of complex solvent effects on chemical reactivity and computational tools to investigate them: a review. *Reaction Chemistry & Engineering*, 4(2):165–206, 2019.

[199] Gengnan Li, Bin Wang, and Daniel E Resasco. Water-mediated heterogeneously catalyzed reactions. *Acs Catalysis*, 10(2):1294–1309, 2019.

[200] George EP Box. Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799, 1976.

[201] Alex K Chew, Theodore W Walker, Zhizhang Shen, Benginur Demir, Liam Witteman, Jack Euclide, George W Huber, James A Dumesic, and Reid C Van Lehn. Effect of mixed-solvent environments on the selectivity of acid-catalyzed dehydration reactions. *ACS Catalysis*, 10(3):1679–1691, 2019.

[202] Francis J Anscombe. Graphs in statistical analysis. *The american statistician*, 27(1):17–21, 1973.

[203] Justin Matejka and George Fitzmaurice. Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1290–1294, 2017.

[204] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[205] Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[206] Robert Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

[207] Gunnar Carlsson, Afra Zomorodian, Anne Collins, and Leonidas J Guibas. Persistence barcodes for shapes. *International Journal of Shape Modeling*, 11(02):149–187, 2005.

[208] Peter Bubenik. Statistical topology using persistence landscapes. *arXiv preprint arXiv:1207.6437*, 3, 2012.

[209] Peter Bubenik, Gunnar Carlsson, Peter T Kim, and Zhi-Ming Luo. Statistical topology via morse theory persistence and nonparametric estimation. *Algebraic methods in statistics and probability II*, 516:75–92, 2010.

[210] Peter Bubenik and Paweł Dłotko. A persistence landscapes toolbox for topological statistics. *Journal of Symbolic Computation*, 78:91–114, 2017.

[211] Andrew J Blumberg, Itamar Gal, Michael A Mandell, and Matthew Pancia. Robust statistics, hypothesis testing, and confidence intervals for persistent homology on metric measure spaces. *Foundations of Computational Mathematics*, 14(4):745–789, 2014.

[212] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *The Journal of Machine Learning Research*, 18(1):218–252, 2017.

[213] Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A stable multi-scale kernel for topological machine learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4741–4748, 2015.

[214] Hyekyoung Lee, Hyejin Kang, Moo K Chung, Bung-Nyun Kim, and Dong Soo Lee. Persistent brain network homology from the perspective of dendrogram. *IEEE transactions on medical imaging*, 31(12):2267–2277, 2012.

[215] Takashi Ichinomiya, Ippei Obayashi, and Yasuaki Hiraoka. Persistent homology analysis of craze formation. *Physical Review E*, 95(1):012504, 2017.

[216] Takenobu Nakamura, Yasuaki Hiraoka, Akihiko Hirata, Emerson G Escolar, and Yasumasa Nishiura. Persistent homology and many-body atomic structure for medium-range order in the glass. *Nanotechnology*, 26(30):304001, 2015.

[217] Mickaël Buchet, Yasuaki Hiraoka, and Ippei Obayashi. Persistent homology and materials informatics. In *Nanoinformatics*, pages 75–95. Springer, Singapore, 2018.

[218] Lee M Seversky, Shelby Davis, and Matthew Berger. On time-series topological data analysis: New data and opportunities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 59–67, 2016.

[219] Jose A Perea and John Harer. Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics*, 15(3):799–838, 2015.

[220] Jose A Perea, Anastasia Deckard, Steve B Haase, and John Harer. Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. *BMC bioinformatics*, 16(1):257, 2015.

[221] Miroslav Kramár, Rachel Levanger, Jeffrey Tithof, Balachandra Suri, Mu Xu, Mark Paul, Michael F Schatz, and Konstantin Mischaikow. Analysis of kolmogorov flow

and rayleigh–bénard convection using persistent homology. *Physica D: Nonlinear Phenomena*, 334:82–98, 2016.

[222] Bernadette J Stolz, Heather A Harrington, and Mason A Porter. Persistent homology of time-dependent functional networks constructed from coupled time series. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(4):047410, 2017.

[223] Kelin Xia and Guo-Wei Wei. Persistent homology analysis of protein structure, flexibility, and folding. *International journal for numerical methods in biomedical engineering*, 30(8):814–844, 2014.

[224] Peter M Kasson, Afra Zomorodian, Sanghyun Park, Nina Singhal, Leonidas J Guibas, and Vijay S Pande. Persistent voids: a new structural metric for membrane fusion. *Bioinformatics*, 23(14):1753–1759, 2007.

[225] Kelin Xia, Xin Feng, Yiying Tong, and Guo Wei Wei. Persistent homology for the quantitative prediction of fullerene stability. *Journal of computational chemistry*, 36(6):408–422, 2015.

[226] Allen Hatcher. *Algebraic topology*. 2005.

[227] Daniel M Kan. Abstract homotopy. *Proceedings of the National Academy of Sciences of the United States of America*, 41(12):1092, 1955.

[228] Robert D Cook et al. *Concepts and applications of finite element analysis*. John wiley & sons, 2007.

[229] James R Munkres. *Elements of algebraic topology*. CRC Press, 2018.

[230] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st annual symposium on foundations of computer science*, pages 454–463. IEEE, 2000.

[231] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

[232] Ulrich Bauer and Michael Lesnick. Induced matchings of barcodes and the algebraic stability of persistence. In *Proceedings of the thirtieth annual symposium on Computational geometry*, pages 355–364, 2014.

[233] Frédéric Chazal, Vin De Silva, Marc Glisse, and Steve Oudot. The structure and stability of persistence modules. *arXiv preprint arXiv:1207.3674*, 21, 2012.

[234] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

[235] Paul Alexandroff. Über den allgemeinen dimensionsbegriff und seine beziehungen zur elementaren geometrischen anschauung. *Mathematische Annalen*, 98(1):617–635, 1928.

[236] Robert W Ghrist. *Elementary applied topology*, volume 1. Createspace Seattle, 2014.

[237] Konstantin Mischaikow and Vidit Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete & Computational Geometry*, 50(2):330–353, 2013.

[238] John Willard Milnor, Michael Spivak, and Robert Wells. *Morse theory*, volume 1. Princeton university press Princeton, 1969.

[239] David Günther, Jan Reininghaus, Hubert Wagner, and Ingrid Hotz. Efficient computation of 3d morse–smale complexes and persistent homology using discrete morse theory. *The Visual Computer*, 28(10):959–969, 2012.

[240] Madjid Allili, Konstantin Mischaikow, and Allen Tannenbaum. Cubical homology and the topological classification of 2d and 3d imagery. In *Proceedings 2001 international conference on image processing (Cat. No. 01CH37205)*, volume 2, pages 173–176. IEEE, 2001.

[241] Marc Niethammer, Andrew N Stein, William D Kalies, Pawel Pilarczyk, Konstantin Mischaikow, and Allen Tannenbaum. Analysis of blood vessel topology by cubical

homology. In *Proceedings. International Conference on Image Processing*, volume 2, pages II–II. IEEE, 2002.

[242] Travis E Oliphant. Python for scientific computing. *Computing in Science & Engineering*, 9(3):10–20, 2007.

[243] Clément Maria, Jean-Daniel Boissonnat, Marc Glisse, and Mariette Yvinec. The gudhi library: Simplicial complexes and persistent homology. In *International congress on mathematical software*, pages 167–174. Springer, 2014.

[244] Ippei Obayashi. Volume-optimal cycle: Tightest representative cycle of a generator in persistent homology. *SIAM Journal on Applied Algebra and Geometry*, 2(4):508–534, 2018.

[245] Yuhei Umeda. Time series classification via topological data analysis. *Information and Media Technologies*, 12:228–239, 2017.

[246] Yuan Wang, Hernando Ombao, and Moo K Chung. Topological data analysis of single-trial electroencephalographic signals. *The annals of applied statistics*, 12(3):1506, 2018.

[247] Jose A Perea. Topological time series analysis. *Notices of the American Mathematical Society*, 66(5), 2019.

[248] Firas A Khasawneh, Elizabeth Munch, and Jose A Perea. Chatter classification in turning using machine learning and topological data analysis. *IFAC-PapersOnLine*, 51(14):195–200, 2018.

[249] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, Puneet Agarwal, et al. Long short term memory networks for anomaly detection in time series. In *Proceedings*, volume 89, pages 89–94, 2015.

[250] Nikolay Laptev, Saeed Amizadeh, and Ian Flint. Generic and scalable framework for

automated time-series anomaly detection. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1939–1947, 2015.

[251] Philip K Chan and Matthew V Mahoney. Modeling multiple time series for anomaly detection. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8–pp. IEEE, 2005.

[252] Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J Guibas, and Steve Y Oudot. Proximity of persistence modules and their diagrams. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*, pages 237–246, 2009.

[253] Marian Gidea and Yuri Katz. Topological data analysis of financial time series: Landscapes of crashes. *Physica A: Statistical Mechanics and its Applications*, 491:820–834, 2018.

[254] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[255] Rahul R Shah and Nicholas L Abbott. Principles for measurement of chemical exposure based on recognition-driven anchoring transitions in liquid crystals. *Science*, 293(5533):1296–1299, 2001.

[256] J Spidlen, K Breuer, C Rosenberg, N Kotecha, and RR Brinkman. A resource of annotated flow cytometry datasets associated with peer-reviewed publications. *Cytometry Part A*, 81(9):727–731, 2012.

[257] Omer Bobrowski, Sayan Mukherjee, Jonathan E Taylor, et al. Topological consistency via kernel estimation. *Bernoulli*, 23(1):288–328, 2017.

[258] Mathieu Carriere, Marco Cuturi, and Steve Oudot. Sliced wasserstein kernel for persistence diagrams. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 664–673. JMLR. org, 2017.

[259] Alex K Chew, Shengli Jiang, Weiqi Zhang, Victor M Zavala, and Reid C Van Lehn. Fast predictions of liquid-phase acid-catalyzed reaction rates using molecular dynamics simulations and convolutional neural networks. 2019.

[260] Isaac Chavel. *Riemannian geometry: a modern introduction*, volume 98. Cambridge university press, 2006.

[261] Bernhard Riemann. *On the Hypotheses which lie at the Bases of Geometry*. Birkhäuser, 2016.

[262] John M Lee. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media, 2006.

[263] P Thomas Fletcher, Conglin Lu, Stephen M Pizer, and Sarang Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE transactions on medical imaging*, 23(8):995–1005, 2004.

[264] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM journal on matrix analysis and applications*, 29(1):328–347, 2007.

[265] Stefan Sommer, François Lauze, Søren Hauberg, and Mads Nielsen. Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximations. In *European conference on computer vision*, pages 43–56. Springer, 2010.

[266] Xavier Pennec. Manifold-valued image processing with SPD matrices. In *Riemannian geometric statistics in medical image analysis*, pages 75–134. Elsevier, 2020.

[267] Gregory C Reinsel. *Elements of multivariate time series analysis*. Springer Science & Business Media, 2003.

[268] Larry Wasserman. *All of statistics: a concise course in statistical inference*, volume 26. Springer, 2004.

[269] Jianfeng Yao, Shurong Zheng, and ZD Bai. *Sample covariance matrices and high-dimensional data analysis*. Cambridge University Press Cambridge, 2015.

[270] Bo Feng, Mengyin Fu, Hongbin Ma, Yuanqing Xia, and Bo Wang. Kalman filter with recursive covariance estimation—sequentially estimating process noise covariance. *IEEE Transactions on Industrial Electronics*, 61(11):6253–6263, 2014.

[271] M Willjuice Iruthayarajan and S Baskar. Covariance matrix adaptation evolution strategy based design of centralized PID controller. *Expert systems with Applications*, 37(8):5775–5781, 2010.

[272] M Mansouri, MZ Sheriff, R Baklouti, M Nounou, H Nounou, A Ben Hamida, and N Karim. Statistical fault detection of chemical process-comparative studies. *Journal of Chemical Engineering & Process Technology*, 7(1):282–291, 2016.

[273] Barry M Wise and Neal B Gallagher. The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control*, 6(6):329–348, 1996.

[274] Evan L Russell, Leo H Chiang, and Richard D Braatz. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemometrics and intelligent laboratory systems*, 51(1):81–93, 2000.

[275] Anqi Qiu, Annie Lee, Mingzhen Tan, and Moo K Chung. Manifold learning on brain functional networks in aging. *Medical image analysis*, 20(1):52–60, 2015.

[276] Olaf Sporns. Network analysis, complexity, and brain function. *Complexity*, 8(1):56–60, 2002.

[277] Gaël Varoquaux, Flore Baronnet, Andreas Kleinschmidt, Pierre Fillard, and Bertrand Thirion. Detection of brain functional-connectivity difference in post-stroke patients using group-level covariance modeling. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 200–208. Springer, 2010.

[278] Joaquín Goñi, Martijn P Van Den Heuvel, Andrea Avena-Koenigsberger, Nieves Velez De Mendizabal, Richard F Betzel, Alessandra Griffa, Patric Hagmann, Bernat Corominas-Murtra, Jean-Philippe Thiran, and Olaf Sporns. Resting-brain functional connectivity predicted by analytic measures of network communication. *Proceedings of the National Academy of Sciences*, 111(2):833–838, 2014.

[279] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In *European conference on computer vision*, pages 589–600. Springer, 2006.

[280] Fatih Porikli. Achieving real-time object detection and tracking under extreme conditions. *Journal of Real-Time Image Processing*, 1(1):33–40, 2006.

[281] Yang Xu, Zebin Wu, Jun Li, Antonio Plaza, and Zhihui Wei. Anomaly detection in hyperspectral images based on low-rank and sparse representation. *IEEE Transactions on Geoscience and Remote Sensing*, 54(4):1990–2000, 2015.

[282] Nara M Portela, George DC Cavalcanti, and Tsang Ing Ren. Semi-supervised clustering for MR brain image segmentation. *Expert Systems with Applications*, 41(4):1492–1497, 2014.

[283] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[284] Serge Lang. *Fundamentals of differential geometry*, volume 191. Springer Science & Business Media, 2012.

[285] Xavier Pennec. Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127–154, 2006.

[286] Maher Moakher and Philipp G Batchelor. Symmetric positive-definite matrices: From geometry to applications and visualization. In *Visualization and Processing of Tensor Fields*, pages 285–298. Springer, 2006.

[287] Rajendra Bhatia. *Positive definite matrices*. Princeton university press, 2009.

[288] Kisung You and Hae-Jeong Park. Re-visiting Riemannian geometry of symmetric positive definite matrices for the analysis of functional connectivity. *Neuroimage*, 225:117464, 2021.

[289] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[290] Wenfu Ku, Robert H Storer, and Christos Georgakis. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and intelligent laboratory systems*, 30(1):179–196, 1995.

[291] Shen Yin, Steven X Ding, Adel Haghani, Haiyang Hao, and Ping Zhang. A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process. *Journal of process control*, 22(9):1567–1581, 2012.

[292] Mehrtash T Harandi, Mathieu Salzmann, and Richard Hartley. From manifold to manifold: Geometry-aware dimensionality reduction for SPD matrices. In *European conference on computer vision*, pages 17–32. Springer, 2014.

[293] Alexandre Barachant and Marco Congedo. A plug&play p300 bci using information geometry. *arXiv preprint arXiv:1409.0107*, 2014.

[294] Fatih Porikli and Oncel Tuzel. Fast construction of covariance matrices for arbitrary size image windows. In *2006 International Conference on Image Processing*, pages 1581–1584. IEEE, 2006.

[295] Ian L Dryden, Alexey Koloydenko, and Diwei Zhou. Non-Euclidean statistics for covariance matrices, with applications to diffusion tensor imaging. *The Annals of Applied Statistics*, 3(3):1102–1123, 2009.

[296] Charlene E Caefer, Jerry Silverman, Oded Orthal, Dani Antonelli, Yaron Sharoni, and Stanley R Rotman. Improved covariance matrices for point target detection in hyperspectral data. *Optical Engineering*, 47(7):076402, 2008.

[297] Oncel Tuzel, Fatih Porikli, and Peter Meer. Pedestrian detection via classification on Riemannian manifolds. *IEEE transactions on pattern analysis and machine intelligence*, 30(10):1713–1727, 2008.

[298] Alejandro F Frangi, Wiro J Niessen, Koen L Vincken, and Max A Viergever. Multiscale vessel enhancement filtering. In *International conference on medical image computing and computer-assisted intervention*, pages 130–137. Springer, 1998.

[299] Ian T Young and Lucas J Van Vliet. Recursive implementation of the Gaussian filter. *Signal processing*, 44(2):139–151, 1995.

[300] Marilena Kampa and Elias Castanas. Human health effects of air pollution. *Environmental pollution*, 151(2):362–367, 2008.

[301] Helmut Mayer. Air pollution in cities. *Atmospheric environment*, 33(24-25):4029–4037, 1999.

[302] Veerabhadran Ramanathan and Yan Feng. Air pollution, greenhouse gases and climate change: Global and regional perspectives. *Atmospheric environment*, 43(1):37–50, 2009.

[303] Emily G Snyder, Timothy H Watkins, Paul A Solomon, Eben D Thoma, Ronald W Williams, Gayle SW Hagler, David Shelow, David A Hindin, Vasu J Kilaru, and Peter W Preuss. The changing paradigm of air pollution monitoring. *Environmental science & technology*, 47(20):11369–11377, 2013.

[304] Francesca Dominici, Roger D Peng, Christopher D Barr, and Michelle L Bell. Protecting human health from air pollution: shifting from a single-pollutant to a multipollutant approach. *Epidemiology (Cambridge, Mass.)*, 21(2):187, 2010.

[305] Antonella Zanobetti, Elena Austin, Brent A Coull, Joel Schwartz, and Petros Koutrakis. Health effects of multi-pollutant profiles. *Environment international*, 71:13–19, 2014.

[306] US EPA. The multi-pollutant report: Technical concepts and examples, 2007.

[307] Joe L Mauderly. The national environmental respiratory center (nerc) experiment in multi-pollutant air quality health research: I. background, experimental strategy and critique. *Inhalation Toxicology*, 26(11):643–650, 2014.

[308] Sanford Sillman and Dongyang He. Some theoretical results concerning 03-nox-voc chemistry and nox-voc indicators. *Journal of Geophysical Research: Atmospheres*, 107(D22):ACH–26, 2002.

[309] Jinxi Hua, Yuanxun Zhang, Benjamin de Foy, Xiaodong Mei, Jing Shang, and Chuan Feng. Competing pm2. 5 and no2 holiday effects in the beijing area vary locally due to differences in residential coal burning and traffic patterns. *Science of The Total Environment*, 750:141575, 2021.

[310] Davide Pigoli, Alessandra Menafoglio, and Piercesare Secchi. Kriging prediction for manifold-valued random fields. *Journal of Multivariate Analysis*, 145:117–131, 2016.

[311] Noel Cressie. *Statistics for spatial data*. John Wiley & Sons, 2015.

[312] Noel Cressie and Douglas M Hawkins. Robust estimation of the variogram: I. *Journal of the international Association for Mathematical Geology*, 12(2):115–125, 1980.

[313] DJ Mulder, APHJ Schenning, and CWM Bastiaansen. Chiral-nematic liquid crystals as one dimensional photonic materials in optical sensors. *Journal of Materials Chemistry C*, 2(33):6695–6705, 2014.

[314] PT Ireland and TV Jones. Liquid crystal measurements of heat transfer and surface shear stress. *Measurement Science and Technology*, 11(7):969, 2000.

[315] Kun-Lin Yang, Katie Cadwell, and Nicholas L Abbott. Use of self-assembled monolayers, metal ions and smectic liquid crystals to detect organophosphonates. *Sensors and Actuators B: Chemical*, 104(1):50–56, 2005.

[316] Evangelia I Zacharaki, Sumei Wang, Sanjeev Chawla, Dong Soo Yoo, Ronald Wolf, Elias R Melhem, and Christos Davatzikos. Classification of brain tumor type and grade using mri texture and shape in a machine learning scheme. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 62(6):1609–1618, 2009.

[317] Jeremy Kawahara, Aicha BenTaieb, and Ghassan Hamarneh. Deep features to classify skin lesions. In *Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on*, pages 1397–1400. IEEE, 2016.

[318] Peter C Collins, Santhosh Koduri, Brian Welk, Jaimie Tiley, and Hamish L Fraser. Neural networks relating alloy composition, microstructure, and tensile properties of α/β-processed timetal 6-4. *Metallurgical and Materials Transactions A*, 44(3):1441–1453, 2013.

[319] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[320] Julia Ling, Maxwell Hutchinson, Erin Antono, Brian DeCost, Elizabeth A Holm, and Bryce Meredig. Building data-driven models with microstructural images: Generalization and interpretability. *Materials Discovery*, 10:19–28, 2017.

[321] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014.

[322] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[323] Yankai Cao, Huaizhe Yu, Nicholas L Abbott, and Victor M Zavala. Machine learning algorithms for liquid crystal-based sensors. *ACS sensors*, 3(11):2237–2245, 2018.

[324] Marco A Bedolla Pantoja and Nicholas L Abbott. Surface-controlled orientational transitions in elastically strained films of liquid crystal that are triggered by vapors of toluene. *ACS applied materials & interfaces*, 8(20):13114–13122, 2016.

[325] Daniel S Miller, Rebecca J Carlton, Peter C Mushenheim, and Nicholas L Abbott. Introduction to optical methods for characterizing liquid crystals at interfaces. *Langmuir*, 29(10):3154–3169, 2013.

[326] Jacob T Hunter and Nicholas L Abbott. Dynamics of the chemo-optical response of supported films of nematic liquid crystals. *Sensors and Actuators B: Chemical*, 183:71–80, 2013.

[327] Ji Zhu, Saharon Rosset, Robert Tibshirani, and Trevor J Hastie. 1-norm support vector machines. In *Advances in neural information processing systems*, pages 49–56, 2004.

[328] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[329] François Chollet et al. Keras. https://keras.io, 2015.

[330] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[331] Ian Jolliffe. Principal component analysis. In *International encyclopedia of statistical science*, pages 1094–1096. Springer, 2011.

[332] Victor Belyaev, Alexey Solomatin, and Denis Chausov. Phase retardation vs. pretilt

angle in liquid crystal cells with homogeneous and inhomogeneous lc director configuration. *Optics Express*, 21(4):4244–4249, 2013.

[333] M Bishop Christopher. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2016.

[334] John R Kitchin. Machine learning in catalysis. *Nature Catalysis*, 1(4):230, 2018.

[335] Zachary W. Ulissi, Andrew J. Medford, Thomas Bligaard, and Jens K. Nørskov. To address surface reaction network complexity using scaling relations machine learning and dft calculations. *Nature Communications*, 8:14621 EP –, 03 2017.

[336] Kevin Tran and Zachary W Ulissi. Active learning across intermetallics to guide discovery of electrocatalysts for co 2 reduction and h 2 evolution. *Nature Catalysis*, 1(9):696–703, 2018.

[337] Geoffroy Hautier, Christopher C Fischer, Anubhav Jain, Tim Mueller, and Gerbrand Ceder. Finding nature's missing ternary oxide compounds using machine learning and density functional theory. *Chemistry of Materials*, 22(12):3762–3767, 2010.

[338] Xianfeng Ma, Zheng Li, Luke EK Achenie, and Hongliang Xin. Machine-learning-augmented chemisorption model for co2 electroreduction catalyst screening. *The journal of physical chemistry letters*, 6(18):3528–3533, 2015.

[339] Amit A Gokhale, James A Dumesic, and Manos Mavrikakis. On the mechanism of low-temperature water gas shift reaction on copper. *Journal of the American Chemical Society*, 130(4):1402–1414, 2008.

[340] Yongtaek Choi and Harvey G Stenger. Water gas shift reaction kinetics and reactor modeling for fuel cell grade hydrogen. *Journal of Power Sources*, 124(2):432–439, 2003.

[341] KM Vanden Bussche and GF Froment. A steady-state kinetic model for methanol synthesis and the water gas shift reaction on a commercial cu/zno/al2o3catalyst. *Journal of Catalysis*, 161(1):1–10, 1996.

[342] CV Ovesen, P Stoltze, JK Nørskov, and CT Campbell. A kinetic model of the water gas shift reaction. *Journal of catalysis*, 134(2):445–468, 1992.

[343] Ali Hussain Motagamwala, Madelyn R Ball, and James A Dumesic. Microkinetic analysis and scaling relations for catalyst design. *Annual review of chemical and biomolecular engineering*, 9:413–450, 2018.

[344] Andrew J Medford, M Ross Kunz, Sarah M Ewing, Tammie Borders, and Rebecca Fushimi. Extracting knowledge from data through catalysis informatics. *ACS Catalysis*, 8(8):7403–7429, 2018.

[345] Çağla Odabaşı, M Erdem Günay, and Ramazan Yıldırım. Knowledge extraction for water gas shift reaction over noble metal catalysts from publications in the literature between 2002 and 2012. *International Journal of Hydrogen Energy*, 39(11):5733–5746, 2014.

[346] Jeff Greeley, Thomas F Jaramillo, Jacob Bonde, IB Chorkendorff, and Jens K Nørskov. Computational high-throughput screening of electrocatalytic materials for hydrogen evolution. *Nature materials*, 5(11):909, 2006.

[347] Florian Goltl, Philipp Moller, Pajean Uchupalanun, Philippe Sautet, and Ive Hermans. Developing a descriptor-based approach for co and no adsorption strength to transition metal sites in zeolites. *Chemistry of Materials*, 29(15):6434–6444, 2017.

[348] Erik-Jan Ras, Manuel J Louwerse, Marjo C Mittelmeijer-Hazeleger, and Gadi Rothenberg. Predicting adsorption on metals: simple yet effective descriptors for surface catalysis. *Physical Chemistry Chemical Physics*, 15(12):4436–4443, 2013.

[349] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 1998.

[350] Jolliffe. Principal component analysis. Springer, 2002.

[351] Rolf Carlson and Johan E Carlson. *Design and optimization in organic synthesis*, volume 24. Elsevier, 2005.

[352] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.

[353] Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11(Feb):517–553, 2010.

[354] Alexandre d'Aspremont, Laurent E Ghaoui, Michael I Jordan, and Gert R Lanckriet. A direct formulation for sparse pca using semidefinite programming. In *Advances in neural information processing systems*, pages 41–48, 2005.

[355] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[356] Karl Sjöstrand, Line Harder Clemmensen, Rasmus Larsen, and Bjarne Ersbøll. Spasm: A matlab toolbox for sparse statistical modeling. *Journal of Statistical Software Accepted for publication*, 2012.

[357] Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. Introduction to multilayer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 39(1):43–62, 1997.

[358] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[359] Qi Fu, Howard Saltsburg, and Maria Flytzani-Stephanopoulos. Active nonmetallic au and pt species on ceria-based water-gas shift catalysts. *Science*, 301(5635):935–938, 2003.

[360] Yue Li, Qi Fu, and Maria Flytzani-Stephanopoulos. Low-temperature water-gas shift reaction over cu-and ni-loaded cerium oxide catalysts. *Applied Catalysis B: Environmental*, 27(3):179–191, 2000.

[361] SHE Yusheng, Li Lei, Zhan Yingying, LIN Xingyi, Qi ZHENG, and WEI Kemei. Effect of yttrium addition on water-gas shift reaction over cuo/ceo2 catalysts. *Journal of Rare Earths*, 27(3):411–417, 2009.

[362] Vsevolod Salnikov, Daniele Cassese, and Renaud Lambiotte. Simplicial complexes and complex systems. *European Journal of Physics*, 40(1):014001, 2018.

[363] Raúl Rabadán, Yamina Mohamedi, Udi Rubin, Tim Chu, Adam N Alghalith, Oliver Elliott, Luis Arnés, Santiago Cal, Álvaro J Obaya, Arnold J Levine, et al. Identification of relevant genetic alterations in cancer using topological data analysis. *Nature communications*, 11(1):1–10, 2020.

[364] Anuraag Bukkuri, Noemi Andor, and Isabel K Darcy. Applications of topological data analysis in oncology. *Frontiers in Artificial Intelligence*, 4:38, 2021.

[365] Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. Directed hypergraphs and applications. *Discrete applied mathematics*, 42(2-3):177–201, 1993.

[366] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C Smith, Berk Hess, and Erik Lindahl. Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1:19–25, 2015.

[367] Robert B Best, Xiao Zhu, Jihyun Shim, Pedro EM Lopes, Jeetain Mittal, Michael Feig, and Alexander D MacKerell Jr. Optimization of the additive charmm all-atom protein force field targeting improved sampling of the backbone ϕ, ψ and side-chain $\chi 1$ and $\chi 2$ dihedral angles. *Journal of chemical theory and computation*, 8(9):3257–3273, 2012.

[368] Kenno Vanommeslaeghe, Elizabeth Hatcher, Chayan Acharya, Sibsankar Kundu, Shijun Zhong, Jihyun Shim, Eva Darian, Olgun Guvench, P Lopes, Igor Vorobyov, et al. Charmm general force field: A force field for drug-like molecules compatible with the charmm all-atom additive biological force fields. *Journal of computational chemistry*, 31(4):671–690, 2010.

[369] J Christopher Love, Lara A Estroff, Jennah K Kriebel, Ralph G Nuzzo, and George M Whitesides. Self-assembled monolayers of thiolates on metals as a form of nanotechnology. *Chemical reviews*, 105(4):1103–1170, 2005.

[370] Amish J Patel, Patrick Varilly, and David Chandler. Fluctuations of water near extended hydrophobic and hydrophilic surfaces. *The journal of physical chemistry B*, 114(4):1632–1637, 2010.

[371] Robert J Adler, Omer Bobrowski, Matthew S Borman, Eliran Subag, and Shmuel Weinberger. Persistent homology for random fields and complexes. In *Borrowing strength: theory powering applications–a Festschrift for Lawrence D. Brown*, pages 124–143. Institute of Mathematical Statistics, 2010.