東京大学大学院

博士論文

Variant Topological Data Analysis
（バリアント位相データ解析）

チャン　クオック　ホアン

学籍番号 48-167413

指導教員　長谷川　禎彦　准教授

情報理工学系研究科

電子情報学専攻

令和元年12月6日

The University of Tokyo

Doctoral Thesis

Variant Topological Data Analysis

Author:
Quoc Hoan Tran

Supervisor:
Prof. Yoshihiko Hasegawa

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Department of Information and Communication Engineering
Graduate School of Information Science and Technology

December 6, 2019

Declaration of Authorship

I, Quoc Hoan Tran, declare that this thesis titled, "Variant Topological Data Analysis" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date: December 6, 2019

"*I don't know anything, but I do know that everything is interesting if you go into it deeply enough.*"

Richard Feynman

Abstract

Quoc Hoan Tran

Variant Topological Data Analysis

Real-world data exhibit multi-scale and hierarchical structures having patterns, such as in information spreading and epidemic diseases, appearing at various scales. Revealing these patterns to understand the underlying dynamics of data is fundamentally difficult. The major part of this thesis aims to develop a novel set of features to provide insights into the dynamics and reconstruct underlying patterns in the real complex world that the data are supposed to represent. This research lies in the field of algebraic topology, and more specifically, topological data analysis, which is a set of computational topology methods to indicate the "shape" of data.

The most powerful method in topological data analysis is *persistent homology*, which considers data as a finite set of points in a high dimensional space and constructs a geometrical model that is primarily a collection of geometrical shapes depending on the scale to group nearby data. From this geometrical model, data can be modeled as a sequence of shapes along with the change of the scale. The transformation of the topology structure in this sequence is encoded into quantitative features to reveal the hidden structures which are crucial to understanding the underlying structure of data but are sufficiently complicated to be constructed in other standard approaches. Because topology is a qualitative property that is stable under the influence of noise, *persistent homology* has been applied successfully to detect the qualitative changes in material science. However, it remains questions in the existing methods to represent variant-scale and dynamics in other types of data such as time series and complex networks data.

Such research questions are addressed in two separate parts of this thesis. In the first part, we propose *variant topological data analysis*, which is a general framework grounded on topological data analysis to encode complex patterns in data as quantitative features over variable scales. It presents a novel set of features, i.e., variant topological features that directly depict topological information as well as provide insights into the variant-scale and dynamics in data. We show how to integrate these

features into the kernel method to apply in statistical-learning tasks such as classification and change point detection.

The second part of this thesis gathers our contributions in applications of using the variant topological features with time-series and complex networks data. In the statistical-tasks such as classification and transition points detection of time-series and complex networks, the numerical calculations for both synthetic and real-world data demonstrate that the proposed framework is highly effective and presents a unified approach for characterizing the dynamics. Furthermore, we show the robustness of these variant topological features under perturbation applied to the time-series and complex network data. As such, our results suggest that *variant topological data analysis* is a powerful and generic framework for the efficient discovery and analysis of dynamical systems.

The *variant topological data analysis* framework presented in this thesis is evaluated and analyzed on multiple data sets of different dynamical systems to show the robustness of the framework. This thesis presents a unified analysis via the topological method that can be applied to many types of data in dynamical systems.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor Prof. Dr. Yoshihiko Hasegawa, for his amazing dedication in guiding and inspiring me through my Ph.D. His intense passion and collaborative spirit in research make me feel full confidence to build my academic independence. I am honored to have been given a chance to work with him, and to learn from him that it is possible to be a highly professional and successful scientist while keeping both feet on the ground and maintaining modesty, a great sense of humor, and a genuine capability to appreciate others. Every discussion in research with him is delightful and fruitful that can help to clear many vague things in my mind and open new possibilities for me. My gratitude extends way beyond this single paragraph.

I would like to thank Prof. Dr. Shin'ichi Satoh and Prof. Dr. Kenjiro Taura, my advisors during the first two years in my Ph.D., for their support and many fruitful discussions. I am also grateful to Prof. Dr. Hitoshi Iba, Prof. Dr. Yoshimasa Tsuruoka, and Prof. Dr. Junichiro Mori and Prof. Dr. Tetsuya J. Kobayashi for being part of my thesis committee.

Thank you to my wonderful family. My wife, Trang Nguyen, has always been my biggest supporter, no matter what I decided on for my future plans, and my son, Bach Tran, are growing up each day beside me. I never could imagine how my life would change the day my son came and how all my fears and worries would just wash away. My family has provided me strength and motivation to enjoy every moment of my life. I can not thank them enough for their love and support.

I gratefully acknowledge the financial support I received in three years through my Ph.D. by Honjo International Scholarship Foundation. I am also grateful to Ziosoft Inc., represented by president Shusuke Chino for supporting me in the first year of my Ph.D.

I would like to thank members from Hasegawa laboratory. Special thanks to Hiroshi Kera, Tan Van Vu, and Tuan Vo Van for creative collaborations, exciting conversations, and lifelong friendships.

Over the years, I have had the pleasure to receive the continuous support from many organizations and many people. Whatever I did to deserve you, it could not have been enough.

Contents

Declaration of Authorship iii

Abstract vii

Acknowledgements ix

1 Introduction 1

1.1 Topological Data Analysis 1
1.2 Why Topological Data Analysis? 4
1.3 Persistent homology and topological features of data 6
1.3.1 Topological cover and topological features of data 6
1.3.2 Filtration of simplicial complex and topological features 10
1.4 Topological persistence features in characterizing the dynamics in data 14
1.5 Limitation in characterizing the dynamics with variant scales 15
1.6 Aims and contributions of this thesis 19
1.7 Structure of this thesis 22

2 Basic and Background 23

2.1 Some basic concepts in algebraic topology 23
2.2 Simplicial homology 28
2.2.1 Simplicial complexes 28
2.2.2 Homology group 31
2.2.3 Homology calculation 35
2.3 Persistent homology 36
2.3.1 Motivation 36
2.3.2 Filtration 37
2.3.3 Persistence homology group 38
2.3.4 Persistence diagram and topological persistence features 39
2.3.5 Persistent homology calculation 41
2.4 Stability of persistent diagrams 43

3 Variant Topological Data Analysis **45**
3.1 Variant topological features 45
3.2 Distance between variant topological features 46
3.3 Kernel methods for variant topological features 47
3.3.1 Persistence Scale Kernel 48
3.3.2 Persistence Fisher Kernel 49
3.3.3 Selection of the kernel parameters 50
3.3.4 Using kernels with holes of multiple dimensions 51
3.3.5 Kernel Fisher discriminant ratio 51

4 Variant Topological Features in Time-series Analysis **53**
4.1 Time delay embedding of time series 53
4.2 Variant topological features from delay-variant embedding 55
4.3 Stability with respect to the perturbations of time series 55
4.4 Applications 58
4.4.1 Identify the dynamics of biological model 58
4.4.2 Classify real-world time-series data 65
4.4.3 Multiple-time-scale patterns captured by delay-variant embedding 66
4.5 Concluding remarks and discussion 70

5 Variant Topological Features in Complex Networks Analysis **71**
5.1 Variant topological scales in complex networks 71
5.2 Construction of Vietoris–Rips filtration of a network 74
5.3 Variant topological features from scale-variant diffusion embedding . . . 74
5.4 Stability with respect to the perturbations of network structure 77
5.5 Applications 81
5.5.1 Variations of the parameters in network models 81
5.5.2 Identification of network models 84
5.5.3 Classification of real-world network data 87
5.5.4 Detection of transition points in time-evolving networks 90
5.5.5 Considerations on the maximum value of the diffusion timescale 97
5.6 Concluding remarks and discussion 101

6 Conclusion **103**

A Appendix A **105**
A.1 Common measures for a network 105
A.2 Graph kernel methods 105
A.2.1 Random walk kernels 105
A.2.2 ShortestPath kernel 106

A.2.3 Graphlet kernel 106
A.2.4 Weisfeiler–Lehman kernel 107

Bibliography **109**

List of Figures

1.1 The shape of data 5
1.2 Topological covers 7
1.3 Topological covers under noisy condition 7
1.4 Filtration of topological covers 8
1.5 Barcode and persistence diagram 9
1.6 Simplex 11
1.7 Simplicial complex 12
1.8 Definition of holes 13
1.9 Time-delay embedding of Lorentz system 16
1.10 Various topological scales and nondyadic interactions in complex networks 18
1.11 Overview of variant topological data analysis 20

2.1 Example of a k-simplex 29
2.2 Simplicial complex 30
2.3 Definition of holes 34
2.4 Filtration of topological covers 37
2.5 Barcode and persistence diagram of homology classes 40
2.6 Example of a filtration 43

3.1 Variant topological features 46
3.2 Bottleneck distance and kernel method 47

4.1 Delay variant embedding 56
4.2 Schematic of the Hes1 regulatory model 59
4.3 Average power spectra of the protein concentration 60
4.4 Examples of time series of Hes1 model and corresponding persistence diagrams 61
4.5 Average classification accuracies of the delay-variant and single-delay methods over 100 random splits of the Hes1-model data 63
4.6 Average classification accuracies of the delay-variant and single-delay methods over 100 random splits of the Hes1-model data at different embedding dimensions 64

4.7 Time series generated from the frequency-modulated model and its three-dimensional persistence diagram 67
4.8 Projection of the features of synthetic frequency-modulated time series from kernel space to principal components 69

5.1 Various topological scales and interactions between multiple elements in a complex network 72
5.2 Exemplary of Vietoris–Rips filtration constructed from a complex network 75
5.3 Scale-variant topological features of complex networks 77
5.4 Principal components projection from the kernel-mapped feature space of the scale-variant topological features in each network model 82
5.5 Estimated Kernel Fisher discriminant ratio for the series of Girvan–Newman networks and Lancichinetti–Fortunato–Radicchi networks . . . 83
5.6 The kernel principal components projection of the scale-variant topological features, and variation of high-order features of networks 84
5.7 Classification of networks generated from Girvan–Newman, Lancichinetti–Fortunato–Radicchi, and Watts–Strogatz models 86
5.8 Three-dimensional persistence diagrams of one-dimensional holes for the *Drosophila melanogaster* gene regulatory networks 91
5.9 Three-dimensional persistence diagrams of one-dimensional holes for the *Drosophila melanogaster* gene regulatory networks 92
5.10 Three-dimensional persistence diagrams of one-dimensional holes for the *Drosophila melanogaster* gene regulatory networks 93
5.11 Three-dimensional persistence diagrams of one-dimensional holes for the *Drosophila melanogaster* gene regulatory networks 94
5.12 Kernel Fisher discriminant ratio κ calculated from the three-dimensional persistence diagrams of one-dimensional holes for the time-evolving *Drosophila melanogaster* gene regulatory networks 95
5.13 Eigenvalues for Barabási–Albert networks and scale-free configuration networks 98
5.14 Classification of networks generated from the Barabási–Albert model with their configuration networks using scale-variant topological features 99
5.15 Classification of networks generated from the Barabási–Albert model with networks generated from the scale-free configuration model of degree exponent 3 100

List of Tables

4.1 Classification accuracies for real-world time series data sets 66

5.1 Summary statistics of the real-world network datasets 88
5.2 Average and standard deviation of the classification accuracy for social network datasets 89
5.3 Average and standard deviation of the classification accuracy for chemoinformatics and bioinformatics network datasets 89

List of Abbreviations

TDA	**T**opological **D**ata **A**nalysis
VR	**V**ietoris–**R**ips
GN	**G**irvan–**N**ewman
LFR	**L**ancichinetti–**F**ortunato–**R**adicchi
LFR–H	**L**ancichinetti–**F**ortunato–**R**adicchi **H**ierarchical
ER	**E**rdős–**R**ényi
WS	**W**atts–**S**trogatz
SP	**S**ales–**P**ardo

List of Glossaries

b	birth-scale
d	death-scale
h	homology class
n_l	number of l-dimensional simplices
$d^{(\infty)}$	infinity-norm distance
d_B	bottleneck distance
d_H	Hausdorff distance
W_k	k-th Wasserstein distance
$D^{(2)}$	two-dimensional persistence diagram
$D^{(3)}$	three-dimensional persistence diagram
ε	proximity parameter (in Vietoris–Rips construction)
τ	variant scale parameter
σ	simplex
∂	boundary operator
ι	inclusion map
ζ	induced linear map from inclusion map
β_l	number of l-dimensional holes
$\mu_l^{x,y}$	multiplicity of persistence pair (x, y) in l-dimension
$\mathcal{M}$	metric space
$\boldsymbol{X}$	sampled data set
$\boldsymbol{P}$	a set of points (a point cloud)
$\mathbb{P}$	persistent module
$\mathcal{A}$	topology
$(\boldsymbol{X}, \mathcal{A})$	topological space
$T(\boldsymbol{X}, \varepsilon)$	topological cover of $\boldsymbol{X}$ given ε
$V_{\mathrm{R}}(\boldsymbol{X}, \varepsilon)$	Vietoris–Rips simplicial complex of $\boldsymbol{X}$ given ε
$\boldsymbol{B}^L$	unit ball in L-dimensional Euclidean space
$\mathcal{C}_l$	l-chain group of simplicial complex
$\mathcal{Z}_l$	l-th cycle group
$\mathcal{B}_l$	l-th boundary group
$\mathcal{H}_l$	l-th homology group

Chapter 1

Introduction

1.1 Topological Data Analysis

The late twentieth century saw the dawning age of the computer, and with it, the amount of available data has increased dramatically in recent decades. Making sense of the vast amount of data, i.e., extracting the valuable information in the data remains the great challenge of data science, not only because of the sheer size of data but also because of the increase in complexity of data. For data sets obtained in dynamic systems such as time-series signals, genetic sequencing, or genome-wide microarray profiling, data is typically noisy, high-dimensional, or plagued with missing values, making the traditional statistical analysis very difficult.

The rising of machine learning methods in recent years provides useful tools for data analysis. In the machine learning field, the research aims to derive algorithms along with mathematical and statistical models that can make predictions about new data solely from the information contained in collected data sets. These techniques can be mainly applied to the data analysis problems such as *classification* (predicting labels of data), *clustering* (grouping data), and *regression* (approximating function on data). In dealing with high-dimensional data, a good starting point is to reduce the number of dimensions but still preserve the informative patterns which can be effectively interpreted in the statistical models. Along with this line, traditional feature selection methods aim to explicitly select a subset of the explanatory variables based on some domain-specific knowledge. On the other hand, deep learning methods, which represent the data as a nested hierarchy of concepts, i.e., neural networks, aims to extract the suitable features from data for specific tasks implicitly.

For the data such as time-series and complex network data, the true behaviors of the data are often associated with the dynamics of the physical system in which data is obtained. Therefore, it is important to understand from the data the viral factors that govern the dynamics of data. In terms of feature selection and extraction scheme, the domain knowledge should be constructed to obtain the features which

are robust with noise and closely related to the true dynamics behind the data. Recent powerful deep learning methods have also demonstrated success in physical and dynamical systems such as the representation of quantum many-body states [1, 2], the optimization of complex systems [3, 4], or the identification of observables that were not previously obvious. However, these approaches could fail when the input is slightly but deliberately perturbed, resulting in misidentification by a deep neural network. Such perturbed examples can be referred to as adversarial examples and have gained considerable attention recently in the discussion about the robustness of deep learning [5]. Deep learning methods find it difficult to understand why adversarial examples are misclassified because of the black-box nature of the causality between the input data and their underlying model.

Recent mathematical developments are shedding new light on the feature extraction scheme, forging new approaches to better decipher complicated structure in data, which has not been revealed in the other standard approaches [6, 7]. These researches lie in the field of applied algebraic topology, and more specifically, topological data analysis (TDA), which is a set of computational algebraic topology methods to indicate the “shape" of data. Algebraic topology is a sub-field of topology, which is a broad branch with a long history in mathematics that studies the structure of spaces such as the study of smooth manifolds, chaotic dynamic systems, and fractals. Algebraic topology focuses on the global features of a topological space using local data to provide insights on the qualitative properties of geometric structure in the data [8]. These properties are not as sensitive to a choice of coordinates or choice of metric, more precisely, describe properties of a space that remain invariant under stretching and shrinking. The intuitive explanation is that algebraic topology tries to describe an object in how many distinct components does an object have, how many holes on its surface, or how many cavities inside the object. Algebraic topology views two spaces as being “the same" if they can be continuously deformed from one to the other. As a famous example, a coffee mug and a donut are equivalent in algebraic topology; more precisely, algebraic topology does not distinguish between those objects, although they are very different in terms of the exact geometrical shape. If we perform a few transformations on their mouldable clay models without tearing or gluing the object, we can make an object with the mug shape from an object with the donut shape, and vice versa. Both of the coffee mug and the donut are topological equivalent in a sense that they have the same invariant properties, i.e., a single visible cavity and are one complete connected object. Focusing on these invariants provides a perspective that is fundamentally distinct from that offered by purely geometric methods [6].

While algebraic topology is a twentieth-century field of mathematics that can trace its origins and connections back to the mathematics in the 18th century (e.g., the Euler characteristic to study a famous problem in graph theory called the Seven Bridges of

Konigsberg), the applicability of algebraic topology to the other field has been focused on the modern applied mathematics. The first contribution in data science is credited with the discovery of Edelsbrunner and Harer that methods from algebraic topology could be of use when analyzing real-world data sets [9]. In the seminal paper [10], Edelsbrunner et al. begin with a concept called *manifold assumption* that instead of assuming that a given discrete, noisy data set could come from every part of the possible space, it makes more sense to assume the data is sampled from some unknown high-dimensional manifold. In image analysis, for example, when analyzing collection images of animals, typically, we do not see images of white noise or spectrogram of sounds, meaning that "animal" images do not occupy the entire space of possible pixel configurations. Having seen that this assumption is justified, Edelsbrunner et al. marks the beginning of TDA by develop their version of homology theory in algebraic topology to describe the topological structure of unknown manifold from the data.

Given some data set, the first step in TDA is to construct a geometric representation of the data then apply topological algebraic methods to that representation to obtain inferences and data abstractions about the shape and connectivity of the data. In TDA, the most used geometric representation of data is the simplicial complex model, which has the same topological properties as the assumed manifold of data but consists of a finite set of elements. A simplicial complex is a collection of geometrical objects, including points, segments, and higher dimensional elements like triangles and tetrahedra. Algebraic operations are implemented in a vector space defined from simplicial complex, and the topological structures such as gaps or holes in the data manifold, are encompassed and arranged in algebraic structures as homology groups.

There are two major methods in TDA: *Mapper* and *persistent homology*. *Mapper* constructs a simplicial complex (in most cases, just a variation of the Reeb graph) to represent the data after mapping the data space to another domain via a "len", or a filter. *Mapper* is particularly useful for visualization because it can flexibly capture the shape of data in a very different way with other dimensional reduction methods such as principal component analysis (PCA) and multidimensional scaling (MDS). In the present thesis, we only focus on *persistent homology*, which is a method for quantitatively evaluate the topological structures of the underlying data manifold. The simplicial complex, which is built for the representation of data, depends on a parameter ε, i.e., the spatial resolution estimates of neighborhoods of data. Persistent homology studies how the topology structures, i.e., holes in the simplicial complex, change as we vary the spatial resolution ε. The persistent structures, which are detected over a wide range of spatial scales, are deemed more likely to represent the true topological structure of the underlying space rather than artifacts of sampling, noise, or the particular choice of parameters. Moreover, under this scheme, persistent homology provides topological features encoded in a persistence diagram that shows when

the topological structure appears (*birth scale*) and disappears (*death scale*). These features give us an informative overview and qualitative estimation of the topology of the underlying data manifold.

TDA opens new era for understanding a problem in data analysis from a topological perspective. It leads to many possibilities in addressing difficult problems with the wide range of applications such as recognizing effective structures in material science [11–17], detecting the qualitative changes in dynamical systems [18–25], and revealing the notable knowledge in biological contexts including the study of horizontal evolution [26, 27], cancer genomics [28, 29], complex diseases [30, 31], disease spreading [32], gene expression [33, 34], brain networks [35, 36]. In the following sections, we will go deeply into why TDA is needed (Sec. 1.2), what are the differences of TDA with other standard methods, how to obtain toplological features from data (Sec. 1.3), the problem of current TDA methods when applying to the dynamics system (Sec. 1.4), and our contributions to address this kind of problem (Sec. 1.6).

1.2 Why Topological Data Analysis?

The field of TDA emerged in response to various problems of identifying patterns in high-dimensional data analysis. A traditional example is a fitting regression line to the data points whose distribution is presented in Fig. 1.1(a). In this example, a straight line is undoubtedly a shape, and we can consider the regression line as the “shape" of data, i.e., a useful organization to explain the principle of the data set. However, the data does not always fit along a straight line. For example, the data can be grouped into a set of clusters of points that share some measure of similarity [Fig. 1.1(b)], and the data can have a loop that may associate with periodic or recurrent behavior in the dynamics of data [Fig. 1.1(c)]. Another example of the “shape" of data is the “Y" shape in Fig. 1.1(d) with three tendrils (gray points) emanating from a central core (green points). If the data has this shape, we can infer a situation where the core represents the most frequently occurring behaviors, and the tips of the tendrils represent the extreme behaviors in the data. None of the linear regression algorithms or cluster analysis can adequately describe this situation.

The above examples demonstrate that data can take many types of shapes associated with the principle behaviors of data. The first attempt to understand the data would be to assume the shape of data as one of these types and try to fit each type to the data to determine the best representation for data, which can be a straight line, a loop, or an “Y" shape, and so on. However, there is an infinite variety of different possible shapes that occur in real data sets, especially if the data are distributed in a high-dimensional space. In general, noisy and high-dimensional data poses particularly severe challenges to fitting algorithms and clustering analysis [6].

Figure 1.1: There are many types of shapes for real-world data sets, such as (a) straight line, (b) three clusters, (c) loop, and (d) three tendrils emanating from a central core.

The second attempt would be to project high-dimensional data down to a lower-dimensional space that we can identify the essential patterns and representative shape of data. Unfortunately, all of these methods involve transforming the original data in a way that may lose some potentially valuable information, more precisely, do not to preserve the "shape" of the original data. To give an idea for the limitation of the standard dimensional reduction methods such as principal component analysis (PCA) or multi-dimensional scaling (MDS), let us consider a data set sampled from an ellipsoid given by the equation $\sum_{i=1}^{100} \frac{x_i^2}{a_i^2} = 1$ in the 100-dimensional Euclidean space. Here, $a_1 > a_2 > \ldots > a_{100}$ denote the length of semi-axes of the ellipsoid with a situation that one axis of the ellipsoid is much longer than the other ($a_1 \gg a_2$). If we apply PCA or MDS to this data set, because the distances in the 2nd, 3th, ..., 100th axis may be too small to recognize, the data set will appear to a line segment along the first axis direction rather than an ellipse [37].

TDA deals with two difficulties, as mentioned earlier in high-dimensional and noisy data analysis. First, instead of investigating all possible shapes and templates fitting to the data, TDA with the *Mapper* method gives us a way to find a flexible visualization where data can be viewed at many possible shapes (see Refs. [6, 37,

38] for more details). Second, TDA with the *persistent homology* method suggests a general framework to find and quantitatively evaluate interesting patterns in data being able actually to see them via the dimensional reduction of the data. Persistent homology processes the data in the original space to tell us how many clusters and how many loop-like structures there are in the data. Therefore, it can provide knowledge about the structures which play a vital role in the underlying dynamics of the data. In the next section, we will explore in more detail the idea to construct the general framework in persistent homology.

1.3 Persistent homology and topological features of data

In this section, we will describe the primitive strategy for studying data using persistent homology. The mathematical background and preliminaries in detail will be presented in Chapter 2.

1.3.1 Topological cover and topological features of data

We consider a data set $\boldsymbol{X}$ of discrete points sampled from an unknown space as the subspace of the metric space $\mathcal{M} = (\mathbb{X}, d)$ with d denotes the distance metric defined on $\mathbb{X} \times \mathbb{X}$. Persistent homology aims to infer the topological properties of the underlying space of $\boldsymbol{X}$ by focusing on connected components and holes (e.g., loops, tunnels, and cavities, etc.) in the geometric object created from $\boldsymbol{X}$. However, with data set $\boldsymbol{X}$ of $|\boldsymbol{X}|$ discrete points in the space, there are $|\boldsymbol{X}|$ connected components (as discrete points) without any loop, tunnel, or cavity, and therefore no information about the topological properties of the underlying space. The idea of TDA is to not study on the topological properties of $\boldsymbol{X}$ directly, but rather the *topological cover* of $\boldsymbol{X}$. Here, given a nonnegative proximity parameter ε, the ε-scale topological cover $T(\boldsymbol{X}, \varepsilon)$ of $\boldsymbol{X}$ is defined as the set of all points in $\mathbb{X}$ within distance ε from some point in $\boldsymbol{X}$.

In Fig. 1.2, we present two examples of topological cover (in green) for data sets $\boldsymbol{X}_1$ [Fig. 1.2(a)] and $\boldsymbol{X}_2$ [Fig. 1.2(b)] of discrete points (in gray) with corresponding proximity parameters ε_1 and ε_2. Here, with a good sampled data set $\boldsymbol{X}$ and a choice of ε, the topological cover $T(\boldsymbol{X}, \varepsilon)$ can provide important information about the topological properties of underlying space. For instance, the clusters in the underlying space of $\boldsymbol{X}_1$ may correspond to the connected components of $T(\boldsymbol{X}_1, \varepsilon_1)$, and similarly, the loops in the underlying space of $\boldsymbol{X}_2$ may correspond to the hole in $T(\boldsymbol{X}_2, \varepsilon_2)$.

In the practical application, when data is noisy and lies in high-dimensional space, the above strategy must deal with several critical shortcomings. First, the topological properties of $T(X, \varepsilon)$ are not robust to noise in X, such as illustrated in Fig. 1.3. Here, we consider the noise-perturbed data set $\boldsymbol{X}_1'$ of $\boldsymbol{X}$ with the noisy points in

Figure 1.2: The topological covers (a) $T(\boldsymbol{X}_1, \varepsilon_1)$ (in green) of the data set $\boldsymbol{X}_1$ (in gray) with proximity parameter $\varepsilon = \varepsilon_1$, and (b) $T(\boldsymbol{X}_2, \varepsilon_2)$ (in green) of the data set $\boldsymbol{X}_2$ (in gray) with proximity parameter $\varepsilon = \varepsilon_2$. The topological cover $T(\boldsymbol{X}, \varepsilon)$ of a data set $\boldsymbol{X}$ with proximity parameter ε is the set of all points in the space within distance ε from some point in $\boldsymbol{X}$.

Figure 1.3: The topological covers (a) $T(\boldsymbol{X}'_1, \varepsilon_1)$ (in green) of the noise-perturbed data set $\boldsymbol{X}'_1$ (in gray) with proximity parameter $\varepsilon = \varepsilon_1$, and (b) $T(\boldsymbol{X}_2, \varepsilon'_2)$ (in green) of the data set $\boldsymbol{X}_2$ (in gray) with proximity parameter $\varepsilon = \varepsilon'_2$. With noise-perturbed data set and different choices of ε, we obtain totally different topological properties of the space with ones illustrated in Fig. 1.2.

Figure 1.4: Exemplary of a filtration of topological covers $T(\boldsymbol{X}, \varepsilon)$ constructed from the data set $\boldsymbol{X}$. Topological structure changes are tracked, such as the merging of connected components or clusters, and the emergence and disappearance of loops or circular holes present in the space with increasing ε. For instance, the loop Ω_1 appears at $\varepsilon = \varepsilon_2$ then disappears at $\varepsilon = \varepsilon_4$, whereas the loop Ω_2 appears at $\varepsilon = \varepsilon_3$ then disappears at $\varepsilon = \varepsilon_5$. For each ε, the number of connected components and the number of loops are listed underneath.

red are added to $\boldsymbol{X}$. Even with the same choice of ε as in Fig. 1.2(a) $\varepsilon = \varepsilon_1$, the topological properties of $T(\boldsymbol{X}'_1, \varepsilon_1)$ (with one component and one loop) are totally different from the topological properties of $T(\boldsymbol{X}_1, \varepsilon_1)$ (with three components and zero loops). Second, the selection of ε is a very sensitive problem without a clear reason, especially when the high-dimensional data cannot be visualized directly. For instance, in Fig. 1.3(b), with the same data set $\boldsymbol{X}_2$ as in Fig. 1.2(b), if we choose ε large enough as $\varepsilon = \varepsilon'_2$, we obtain only one component in the space that may not correspond to the true topological properties as the loop illustrated in Fig. 1.2(b).

To deal with the above-mentioned problems, the key idea behind persistent homology is to consider at once the topological properties of the entire family of topological covers $T(\boldsymbol{X}, \varepsilon)$ as ε varies, instead of a single choice of ε. If we consider the proximity parameter ε as the spatial resolution to view data in $\boldsymbol{X}$, the representative topological structures in the underlying space of $\boldsymbol{X}$ should be the structures appear at different spatial resolutions when viewing $\boldsymbol{X}$. We illustrate this idea in Fig. 1.4 when we consider the data set $\boldsymbol{X}$ sampled from the shape of the number "8" in two-dimensional space. We show that we can infer the topological properties of the underlying space of $\boldsymbol{X}$, which has two loops with different perimeters. Now, starting with $\varepsilon = 0$, the topological cover $T(\boldsymbol{X}, \varepsilon = 0)$ contains only the discrete points in $\boldsymbol{X}$. As ε increases, connections exist between components in $T(\boldsymbol{X}, \varepsilon)$, enabling us to obtain a sequence of embedded topological cover called *filtration* in the space. Moreover, if ε becomes considerably large, all the components get connected with each other, we obtain a large and overlapped cover whereby no useful information can be conveyed.

Persistent homology tracks the variation of topological structures, i.e., connected components, and holes over the filtration. For instance, if we focus on the appearance

Figure 1.5: The visualization for the output of persistent homology for the filtration in Fig. 1.4. Each connected component or each loop pattern in the filtration corresponds with a persistence pair (b, d) represented for the values of ε at the emergence (*birth-scale*, $\varepsilon = b$) and the disappearance (*death-scale*, $\varepsilon = d$) of the pattern. These bars (connected components in blue and loops in red) begin at the values of the birth-scale, then end at values of the death-scale in the persistence pairs. The collection of these bars is called a *barcode*, and the representation of all persistence pairs in a two-dimensional diagram (connected components in blue circles and loops in red circles) is called a *persistence diagram*. (a) Barcode and (b) Persistence diagram for the filtration in Fig. 1.4. Note that, points in the diagram are multi-set points.

and disappearance of loop structures in the filtration in Fig. 1.4, we can obtain the information of two loop-like structures Ω_1 and Ω_2. The loop Ω_1 appears at $\varepsilon = \varepsilon_2$ then disappears at $\varepsilon = \varepsilon_4$, whereas the loop Ω_2 appears at $\varepsilon = \varepsilon_3$ then disappears at $\varepsilon = \varepsilon_5$. Note that, in this filtration, with the single choice of ε, only at $\varepsilon = \varepsilon_3$, two loops appear in the topological cover. However, we cannot distinguish the magnitudes of these loops at this choice of ε. On the other hand, we can distinguish loops if we assign to each loop Ω in the filtration a pair of scale called *persistence pair* (b, d), where at $\varepsilon = b$, the loop Ω appears in the topological cover, and at $\varepsilon = d$, Ω disappears in the topological cover. We call b and d as *birth scale* and *death scale* of loop Ω. The difference between *death scale* and *birth scale*, i.e., $|d-b|$, tells us about the magnitude of the loop Ω in the space, or the *persistence* of loop in the filtration. We also obtain *persistence pair* for each of the connected components appear first at $T(\boldsymbol{X}, \varepsilon = 0)$ and merged together along with the filtration.

The output of persistent homology, which we consider as the *topological persistence features* to represent the shape of data set $\boldsymbol{X}$ is the collection of persistence pairs for all connected components, loops, and generally, holes in the constructed filtration of $\boldsymbol{X}$. The lifetime of these patterns are described as blue bars (for connected components) and red bars (for loops) in Fig. 1.5(a). These bars begin at the values of the birth-scale, then end at values of the death-scale in the persistence pairs. The collection of these bars is called a *barcode* of data set $\boldsymbol{X}$ under associated with a filtration construction. Another representation of the topological persistence features is a two-dimensional diagram of multiset points, a called *persistence diagram* [Fig. 1.5(b)]. In this diagram, each point denotes a persistence pair calculated from the filtration.

In the next subsection, we give an example to construct the filtration and present the concepts of *holes*, which is an algebraic generalization for patterns such as connected components, loops, etc.

1.3.2 Filtration of simplicial complex and topological features

In the previous subsection, we have presented the *filtration* as a sequence of topological covers for a data set. However, to put these concepts into computational routine, we need to describe them in the algebraic language. Here, persistent homology employs the use of simplicial complexes, which are complexes of geometric structures called simplices to approximate the topological cover in much more mathematically and computationally tractable representation. Here, an n-simplex is a generalization of the notion of a triangle or tetrahedron to arbitrary dimensions, or more precisely, is geometrically the convex hull of its $n+1$ affinely independent positioned vertices in the space. For example, a 0-simplex is a point, a 1-simplex is a line segment (with two end points as its faces), a 2-simplex is a triangle together with its enclosed area (with

Figure 1.6: A simplex is a generalization of the notion of a triangle or tetrahedron to arbitrary dimensions. For example, a 0-simplex is a point, a 1-simplex is a line segment, a 2-simplex is a triangle together with its enclosed area, and a 3-simplex is a filled tetrahedron, and so on.

three edges and three vertices as its faces), and a 3-simplex is a filled tetrahedron (with triangles, edges, and vertices as its faces), and a 4-simplex is beyond of visualization, but it is a filled shape with tetrahedrons, triangles, edges, and vertices as its faces, and so on (Fig. 1.6). A simplicial complex is a collection of simplices, roughly formed when we "glue" together different simplices with a condition that the common parts of simplices in the simplicial complex must be the faces of both simplices. We call a simplicial complex $\mathcal{K}$ is a n-complex if n is the maximum number such that there is at least one n-simplex in $\mathcal{K}$. For example, Fig. 1.7 illustrates a 1-complex [Fig. 1.7(a)], a 2-complex [Fig. 1.7(b)], a 3-complex [Fig. 1.7(c)], and a shape which is not a simplicial complex since the line segment is connected to the triangle along its edge ant not at one of of its vertices [Fig. 1.7(d)].

The next step is to construct a simplicial complex from data set $\boldsymbol{X}$ to approximate the topological cover of $\boldsymbol{X}$. There are many different types of construction methods with different mathematical definitions and computational standpoints such as the Čech complex, Vietoris–Rips complex [39], Alpha complex, and Witness complex models. We will describe these complex models in Chapter 2, but focus on Vietoris–Rips complex model in this chapter since it is the most practical model from a computational view and mainly used in this thesis.

Given a nonnegative proximity parameter ε, the ε-scale Vietoris–Rips complex $V_{\mathrm{R}}(\boldsymbol{X},\varepsilon)$ is a set of simplices such that every collection of $n+1$ affinely independent points in $\boldsymbol{X}$ forms an n-simplex in $V_{\mathrm{R}}(\boldsymbol{X},\varepsilon)$ if the pairwise distance between the points is less than or equal to ε. Intuitively, we can consider a union of balls of radius $\varepsilon/2$ centered at each point in $\boldsymbol{X}$, then each simplex is built over a subset of points if the balls intersect between every pair of points. In turn, the constructed complex $V_{\mathrm{R}}(\boldsymbol{X},\varepsilon)$ provides information on the topological structure of $\boldsymbol{X}$ associated with ε. Now, starting with $\varepsilon = 0$, the complex contains only the 0-simplices, i.e., the discrete

Figure 1.7: A simplicial complex is a collection of simplices, such that the common parts of simplices in the simplicial complex must be the faces of both simplices. A simplicial complex $\mathcal{K}$ is called a n-complex if n is the maximum number such that there is at least one n-simplex in $\mathcal{K}$. The illustration here depicts (a) a 1-complex, (b) a 2-complex, (c) a 3-complex, and (d) not a simplicial complex.

points. As ε increases, connections exist between the points, enabling us to obtain a sequence of embedded complexes, i.e., a filtration, with edges (1-simplices), and filled triangles (2-simplices) are included into the complexes. Moreover, if ε becomes considerably large, all the points gets connected with each other, whereby no useful information can be conveyed.

We remind that persistent homology tracks the variation of topological structures over the filtration. We refer to the topological structures, i.e., "holes" in high-dimensional data, as connected components, tunnels or loops (e.g., a circle of torus), cavities or voids (e.g., the space enclosed by a sphere), and so on. In persistent homology, a hole is identified via the cycle that surrounds it. In a given manifold, a cycle is a closed submanifold, and a boundary is a cycle that is also the boundary of a submanifold. Holes correspond to cycles that are not themselves boundaries. For instance, a disk is a two-dimensional surface with a one-dimensional boundary (i.e., a circle). If we puncture the disk, we obtain a one-dimensional hole that is enclosed by the circle, which is no longer a boundary [Fig. 1.8(b)(c)]. Similarly, a filled ball is a three-dimensional object with a two-dimensional boundary (i.e., a surface sphere). If we empty the inside of the ball, we obtain a two-dimensional hole that is enclosed by the surface sphere, which is no longer a boundary [Figs. 1.8(d) and 1.8(e)]. Based

Figure 1.8: (a)–(f) Sample manifolds with the number of zero-, one-, and two-dimensional holes listed underneath. (a) The connected component is a zero-dimensional hole. (b)(c) A one-dimensional hole is obtained by puncturing a disk. (d)(e) A two-dimensional hole is obtained by emptying the inside of a ball. (f) Two one-dimensional holes are illustrated as two circles in a torus. (g) Example of a simplicial complex containing 19 points (0-simplices), 24 edges (1-simplices), eight triangular faces (2-simplices), and one filled tetrahedron (3-simplices). There are two one-dimensional holes Ω_1 and Ω_2 in the complex. In this example, all loops $B \to C \to D \to B$, $A \to B \to C \to D \to A$, $B \to E \to C \to D \to B$, and $A \to B \to E \to C \to D \to A$ are 1-cycles because they are closed 1-chains, that is, the closed collection of edges (1-simplices). Each cycle is not a boundary of any 2-chain (collection of triangular faces); thus, it characterizes a one-dimensional hole. Note that these cycles characterize the same hole, Ω_1, because the difference between the two cycles is the boundary of a 2-chain.

on these observations, we can describe and classify holes in the simplicial complex according to the cycles that enclose holes. Given a simplicial complex, we define an n-chain as a collection of n-simplices in the complex. Therefore, in a simplicial complex, we can define an n-cycle as a closed n-chain and an n-boundary as an n-cycle, which is also the boundary of an $(n+1)$-chain. Here, a 0-cycle is a connected component, a 1-cycle is a closed loop, and a 2-cycle is a shell. For instance, in Fig. 1.8(g), all loops $A \to B \to D \to A$, $B \to C \to D \to B$, and $A \to B \to C \to D \to A$ are 1-cycles because they are the closed collection of edges (1-simplices). Furthermore, the loop $A \to B \to D \to A$ is a 1-boundary because it bounds a triangular face (2-simplex). An n-dimensional hole corresponds to an n-cycle that is not a boundary of any $(n+1)$-chain in the simplicial complex. For instance, as illustrated in Fig. 1.8(g), the loops $B \to C \to D \to B$ and $A \to B \to C \to D \to A$ characterize one-dimensional holes because these loops are 1-cycles but are themselves not 1-boundaries. Moreover, two n-cycles characterize the same hole when together they bound an $(n+1)$-chain (i.e., their difference is an n-boundary). Intuitively, the connected components can be considered as zero-dimensional holes, the loops and tunnels as one-dimensional holes, and the cavities and voids as two-dimensional holes.

We consider the emergence and disappearance of holes in the Vietoris–Rips filtration of $\boldsymbol{X}$ as topological persistence features for the data set $\boldsymbol{X}$. Such features can be observed using a barcode, or multi-set points in a two-dimensional persistence diagram.

1.4 Topological persistence features in characterizing the dynamics in data

Because topology is a qualitative property that is stable under the influence of noise, it has been proven to become a useful tool to understand the underlying dynamics of data. Most of the important properties of the attractors of dynamical systems are topological, and their topological structures are essential information to provide insights into behaviors of the dynamical system, such as the detection of bifurcation and chaos. While several data-driven approaches have been proposed for complex system analysis (recurrence plots, Lyapunov exponents, permutation entropy, correlation sum analysis, etc.) to detect the anomalies of qualitative changes in the system behavior, they do not provide any connection between a topological understanding and dynamics. In contrast, TDA, more precisely, persistent homology enables a framework to obtain topological features that can provide useful information about the evolution of system behavior.

The topological persistence features calculated from persistent homology have been used for identifying the qualitative changes in data and successfully applied to the detection of phase transitions in the dynamical system. In Ref. [20], Donato et al. apply persistent homology to the study of equilibrium phase transitions undergone by the so-called mean-field XY model and by the ϕ^4 lattice model. Maletić et al. in Ref. [19] propose a general method for constructing simplicial complex from observed time series of dynamical systems based on the delay coordinate reconstruction procedure. They study the relationship between topological persistence features with the embedding dimensions of the reconstruction process via the lifetime of features and the distribution of features in the persistence diagram. Mittal et al. in Ref. [21] extract the crucial topological summaries such as the number and lifetimes of subcycles to gain insights into system behavior for early detection of bifurcation and chaos. Similar with this research line, Myers et al. in Ref. [25] apply persistent homology for graph construction obtained from the time series, then extract geometric and entropy point summaries from the persistence diagram to distinguish between dynamic states such as periodic and chaotic behavior.

In complex network analysis, the reason for applying persistent homology comes from two specific situations. First, are nondyadic interactions (more than dyadic), and collective behaviors in the network, such as neuron may operate in groups (brain networks), proteins may combine in to complexes (signaling networks), or people may communicate in a community (social networks) [40, 41] [Fig. 1.10(b)]. For such systems, persistent homology with the study of the connection in networks via higher-order geometric models such as clique or simplicial complex can more comprehensively describe the intrinsic structure of the underlying system of networks [18, 42–46]. Second, is the important role of topological hole patterns such as connected components, loops, or cavities in the network. It is easy to image connected components, loops or cycles, which may relate to the community behaviors or oscillation regimes in the network. For cavities, as an example in networks constructed from functional imaging data, one can interpret a cavity as a structure when multiple small groups of regions display correlated activity, but the regional activity profiles are not altogether similar [47].

1.5 Limitation in characterizing the dynamics with variant scales

When applying the topological approach, i.e., persistent homology, to the data obtained from the dynamical system such as time-series data, or complex networks,

Figure 1.9: The observed time series of a single variable from the three-variable Lorentz system is reconstructed by the time-delay embedding scheme. The different values of time delay result in different topological structures in the embedded space, which may lead to an uncertain inference for the behavior of the original system.

because the data is explicitly represented in a high-dimensional space, we must embed the data in a high-dimensional space with a distance metric. This embedding scheme can only focus on a specific scale in the data. However, there are variant and multiple scales in the data with underlying dynamic properties, such as multiple time-scale patterns in a time series, or various topological scales, ranging from individual nodes, through collective patterns, to the global topology in the complex networks. Ideally, if the data is noise-free, a good choice of dynamics scale in the embedding scheme leads to efficiency in utilizing the topological approach. However, in practical situations, noisy data makes the selection of this specific scale becomes an ill-defined and challenging problem.

To illustrate the limitation in characterizing the dynamics having variant scales properties, we give two examples with time-series and complex network data. First, to embed a time series $x(t)$ into a high-dimensional space with a distance metric, we map $x(t)$ to m-dimensional points using delay coordinates on the embedded space as $(x(t), x(t-\tau), \ldots, x(t-(m-1)\tau)$, where a specific τ denotes the predefined time delay and m denotes the embedding dimension. This embedding scheme is based on Takens's theorem, which is one of the theoretical foundations for the analysis of

time series corresponding to nonlinear and deterministic dynamical systems [48]. It states that for a multivariate dynamical system if we measure any single variable with sufficient accuracy for a long time, it is possible to reconstruct the underlying dynamic structure of the entire system from the behavior of that single variable using the above-mentioned embedding procedure. Here, the embedded points from topological structures, such as clusters and loops, and the topological features, which monitor the emergence and disappearance of these structures, can be used to characterize the dynamics of the system. However, since the timescale of the extracted topological features is partially dependent on the time delay, the obtained topological features only reflect patterns on a specific scale of the system. Theoretically, for a noise-free time series of unlimited length, different values of time delay τ are considered to be equivalent in terms of optimal reconstruction. However, because a real-time series is noisy and because of finite length, the selection of τ is considered to be an inherently difficult problem.

In Fig. 1.9, we illustrate the behavior of the attractor in a three-variable Lorentz system with two unstable spiral foci suspended in an attracting surface, and mutually connected in such a way that the outer portion of either spiral is glued toward the side of the other spiral. Here, an evolution of a single variable in the system is observed as a finite-length time series $x(t)$ with the time-delay embedded patterns in a three-dimensional space (the right panel of Fig. 1.9). Note that, the embedded patterns with different time delay $\tau = \tau_1, \tau_2, \tau_3$ results in different behaviors of the attractor. In this specific example, $\tau = \tau_1$ may be an optimal choice for τ to reconstruct the behavior of the system with two spiral foci as two holes in the space. However, we may not obtain exact and useful information about the dynamics of the original system if we set the time delay τ as τ_1, τ_2 and focus on the topological features of the corresponding embedded points. Therefore, the selection of τ depends on the objective of the analysis and is often obtained by trial and error without using any systematic method. This defect limits the power of topological features in time-series data analysis.

Second, we give an example on the characterization of complex networks that are relevant to the dynamics of complex systems. Complex networks can be analyzed at various topological scales, ranging from individual nodes, through collective patterns, to whole networks. Previous studies have focused on the invariants of networks that are constructed from dyadic connections between nodes at a single fixed topological scale [Fig. 1.10(a)]. Nevertheless, their findings have not been so successful to decipher the accurate and complete description of complex systems, especially if there are fluctuations in the systems. By considering the diffusion process at a single specified timescale in a network, one can map the network nodes to a finite set of points in high-dimensional space that contains the topological information of the network at a single scale [49]. If we focus on the topological persistence features at a single timescale,

Figure 1.10: Various topological scales and interactions between multiple elements in a complex network. (a) Complex networks can be analyzed at various topological scales ranging from individual nodes (microscale) to the whole network (macroscale). In between the two scales, there is a mesoscale, where we can observe patterns of collectives, cores, and peripheries. (b) Complex network is a representation of a complex system having dyadic and nondyadic interactions between its elements. The interactions can be represented as simplices such as segments (for dyadic interactions), filled triangles, or filled tetrahedrons (for nondyadic interactions involving three or four elements), and so on.

that is, we can only understand the network at a specific topological scale, therefore, may ignore other important information of the network at other topological scales.

1.6 Aims and contributions of this thesis

This thesis is motivated in the limitations of using topological persistence features to characterize the variant scales. Research in TDA has so far been with a very mathematical perspective. However, in practical applications, it remains questions in the existing methods to represent variant-scale and dynamics in the data, such as in time series and complex network data. This thesis aims to enhance the practical advantages of TDA in data analysis generally, and more precisely, makes TDA accessible and usable in the dynamical analysis. This goal requires an effective representation of topological persistence features to reflect the variant scales in dynamic and a mathematical framework to integrate these features into statistical-inference tasks in real-world problems.

Our contributions are presented in two separate parts of this thesis. First, we propose *variant topological data analysis*, which is a general framework grounded on topological data analysis to encode complex patterns in data as quantitative features over variable scales. It presents a novel set of features, i.e., variant topological features that directly depict topological information as well as provide insights into the variant-scale and dynamics in data. These features are extended from persistent topological features of persistent homology theory but can deal with the limitations of existing topological persistence features in characterizing the dynamics with variant scale. In addition, we provide mathematical supports to guarantee the robustness of these features against noise. We also show how to integrate these features into the kernel method to apply in statistical-learning tasks such as classification and change point detection.

The framework of variant topological data analysis is simply illustrated in Fig. 1.11. Here, data is embedded in a high-dimensional space via a mapping under a scale parameter τ of underlying dynamics, for example, time delay in delay embedding of time series, or time scale in diffusion process on the network. The next step is to use a topological machine, which adopts persistent homology theory to calculate the topological persistence features, i.e., the persistence diagram of the mapped object in the embedded space. Note that these topological persistence features are calculated at each scale τ associated with the mapping of the data. Finally, by considering these features over variable scales τ, we obtain *variant topological features* of the data, i.e., three-dimensional persistence diagram, with the scale τ serving as an additional dimensional in the topological feature space.

After constructing the general framework for variant topological data analysis with solid mathematical supports for variant topological features, the second part of this thesis gathers our contributions in applications of using the variant topological features in time-series and complex network data analysis [24, 34]. In the statistical-tasks

Figure 1.11: The framework of variant topological data analysis. First, data is embedded in a high-dimensional space via a mapping under a scale parameter τ of underlying dynamics. Then, at each scale τ, we use a topological persistence machine, which adopts persistent homology theory to calculate the topological persistence features, i.e., the persistence diagram of the mapped object in the embedded space. Finally, by considering these features over variable scales τ, we obtain *variant topological features* of the data, i.e., a three-dimensional persistence diagram, with the scale τ serving as an additional dimensional in the topological feature space.

such as classification and transition points detection of time-series and complex networks, the numerical calculations for both synthetic and real-world data demonstrate that the proposed framework is highly effective and presents a unified approach for characterizing the dynamics. Furthermore, we show the robustness of these variant topological features under perturbation applied to the time-series and complex network data. As such, our results suggest that *variant topological data analysis* is a powerful and generic framework for the efficient discovery and analysis of dynamical systems.

The *variant topological data analysis* framework presented in this thesis is evaluated and analyzed on different dynamic systems to show the robustness of the framework. This thesis presents a unified analysis via the topological method that can be applied to many types of data in dynamics systems.

This thesis is partially based on several publications by the author, some of which are indicated using brackets in the list above. In comparison to the original publications, this thesis contains a more in-depth discussion of results. A detailed list of peer-reviewed publications in reverse chronological order follows.

- Q. H. Tran, V. T. Vo, and Y. Hasegawa, *Scale-variant topological information for characterizing the structure of complex networks*, **Physical Review E**, vol. 100, 032308 (2019) (selected in Kaleidoscope of Physical Review E).
- Q. H. Tran and Y. Hasegawa, *Topological time-series analysis with delay-variant embedding*, **Physical Review E 99**, vol. 99, 032209 (2019).

The author also contributed to the following peer-reviewed presentations that relate to the topic of this thesis.

- Q. H. Tran, V. T. Vo, and Y. Hasegawa, *Scale-variant topological portraits of complex networks*, Oral Presentation at **NetSci-X: International School and Conference on Network Science**, Tokyo, Japan, Jan. 2020.
- Q. H. Tran and Y. Hasegawa, *Topological time-series analysis with delay-variant embedding*, Oral Presentation at **Conference on Complex Systems**, Singapore, Singapore, Oct. 2019.

1.7 Structure of this thesis

Following the introduction in Chapter 1, Chapter 2 presents an overview of background on TDA, which briefly introduces related topological concepts, such as simplicial complex, holes, and homology groups. It also describes the algorithms to calculate the output of persistent homology, i.e., persistence barcodes and persistence diagrams, which are considered as topological persistence features of the data. The chapter also presents an important property on the stability of topological persistence features under the presence of noise.

Having established the mathematical base and practical utility of persistent homology, Chapter 3 explains the main scope of this thesis, the *variant topological data analysis* framework with our core concept: *variant topological features*. The chapter also introduces the notion to define the distance between variant topological features, and the kernel method for these features to apply in statistical-learning tasks.

Chapter 4 presents the application of utilizing variant topological features in time-series data analysis. First, to apply the topological approach to time series, we have to embed the time-series data into a high-dimensional space using a time-delay parameter. Here, variant topological features, which consider the time-delay as a variable parameter rather than a single fixed value and hence observes the changes in the topological features over different time-scales. We theoretically prove that the proposed variant topological features are robust against noise; we also apply these features to the classification of real time-series data.

Chapter 5 presents the application of using variant topological features to characterize the structure of complex networks. Here, variant topological features are constructed via the observation of a diffusion process on the network along with variable time scales. The variant topological features in our framework can be used as discriminating features to identify networks. Theoretically, we derive an inequality that bounds the maximum deviation of the feature variation under perturbation applied to networks for the first time. We employ these features for identifying synthetic network models, classifying real-world networks, and detecting transition points in time-evolving networks.

The thesis ends with Chapter 6, which briefly summarizes the contributions, discusses them, and points out new areas of research that arise as a result of this work.

Chapter 2

Basic and Background

In this chapter, we discuss the necessary background in algebraic topology, and more precisely, persistent homology which is assumed throughout the rest of this dissertation. Section 2.1 briefly discusses some basic concepts in topology. Then we present the concept of the simplicial complex, which is a useful representation for topological space in the computational routine. Based on the concept of simplicial complexes, we introduce simplicial homology in section 2.2, which is a method for calculating certain invariants of topological spaces. Section 2.3 presents the core concepts of persistent homology. Finally, section 2.4 discusses the stability of persistent homology, i.e., the robustness under noise perturbation applied to the data. This chapter follows great explanations in Refs. [7, 50, 51].

2.1 Some basic concepts in algebraic topology

First, we define topological space as follow.

Definition 1 (Topological space)**.** *A topological space is an ordered pair* $(\boldsymbol{X}, \mathcal{A})$*, where* $\boldsymbol{X}$ *is a set and* $\mathcal{A}$ *is a collection of subsets of* $\boldsymbol{X}$*, satisfying the following properties:*

1. *The empty set and* $\boldsymbol{X}$ *itself belongs to* $\mathcal{A}$*, i.e.,* $\varnothing \in \mathcal{A}, \boldsymbol{X} \in \mathcal{A}$*.*
2. *The union of finite or infinite members of* $\mathcal{A}$ *belongs to* $\mathcal{A}$*, i.e., if* $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n \in \mathcal{A}$ *(n is finite or infinite) then* $\cup_{i=1}^{n} \boldsymbol{X}_i \in \mathcal{A}$*.*
3. *The intersection of any finite number of members belongs to* $\mathcal{A}$*, i.e., if* $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n \in \mathcal{A}$ *(n is finite) then* $\cap_{i=1}^{n} \boldsymbol{X}_i \in \mathcal{A}$*.*

The elements of $\mathcal{A}$ *are called open sets and* $\mathcal{A}$ *is called a topology on* $\boldsymbol{X}$*.*

A simple example for a topological space is a collection of three distinct objects $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$ and $\mathcal{A} = \{\varnothing, \{\boldsymbol{x}_1\}, \{\boldsymbol{x}_2\}, \{\boldsymbol{x}_1, \boldsymbol{x}_2\}, \{\boldsymbol{x}_2, \boldsymbol{x}_3\}, \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}\}$. Another example is a set $\boldsymbol{X}$ of n elements, $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ and $\mathcal{A} = 2^{\boldsymbol{X}}$.

Next, we define a specialization (or canonical) preorder on a topological space.

Definition 2 (Specialization preorder)**.** *The specialization preorder* $\leq$ *on* $(\boldsymbol{X}, \mathcal{A})$ *relates two points of* $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{X}$ *as* $\boldsymbol{x} \leq \boldsymbol{y}$ *(we say that* $\boldsymbol{x}$ *is a specialization of* $\boldsymbol{y}$*) if and only if* $\boldsymbol{y}$ *is contained in all elements of the topology* $\mathcal{A}$ *containing* $\boldsymbol{x}$*.*

In the above-mentioned example,

$$\begin{aligned}\boldsymbol{X} =& \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\},\\ \mathcal{A} =& \{\varnothing, \{\boldsymbol{x}_1\}, \{\boldsymbol{x}_2\}, \{\boldsymbol{x}_1, \boldsymbol{x}_2\}, \{\boldsymbol{x}_2, \boldsymbol{x}_3\}, \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}\}\end{aligned}$$

we can easily check the following relations: $\boldsymbol{x}_1 \nleq \boldsymbol{x}_2, \boldsymbol{x}_1 \nleq \boldsymbol{x}_3, \boldsymbol{x}_1 \nleq \boldsymbol{x}_1, \boldsymbol{x}_2 \nleq \boldsymbol{x}_3, \boldsymbol{x}_3 \nleq \boldsymbol{x}_1$, and $\boldsymbol{x}_3 \leq \boldsymbol{x}_2$.

In many practical scenarios, a data set $\boldsymbol{X}$ is taken to be a finite set of observation vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ in an Euclidean space $\mathbb{R}^L$, where L denotes the dimension of the space. This motivates us to define the metric topological space, which produces an example of topological space with a definite notion distance. Let us now definite a metric space.

Definition 3 (Metric space)**.** *A metric space is an ordered pair* $(\boldsymbol{X}, d)$ *where* $\boldsymbol{X}$ *is a set and* d *is a metric on* $\boldsymbol{X}$*, i.e., a function* $d : \boldsymbol{X} \times \boldsymbol{X} \to \mathbb{R}$ *such that for any elements* $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \boldsymbol{X}$*, the following conditions are satisfied:*

1. $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$
2. $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ *if and only if* $\boldsymbol{x} = \boldsymbol{y}$
3. $d(\boldsymbol{x}, \boldsymbol{z}) \leq d(\boldsymbol{x}, \boldsymbol{y}) + d(\boldsymbol{y}, \boldsymbol{z})$

This definition can be simply rephrased in terms that a metric space is a set paired with a distance function that accept any two elements from that set and returns the metric distance between those elements. One familiar example of metric space is the L-dimensional Euclidean space $\mathbb{R}^L$ ($\boldsymbol{X} = \mathbb{R}^L$), where the distance function between any two points $\boldsymbol{x} = (x_1, x_2, \ldots, x_L)$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_L)$ is defined as $d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_L - y_L)^2}$. In this example, we introduce a natural way to form a topological space. For any $\boldsymbol{x} \in \mathbb{R}^L$ and a given parameter $\varepsilon > 0$, let $B(\boldsymbol{x}, \varepsilon) := \{\boldsymbol{y} \in \mathbb{R}^L \mid d(\boldsymbol{x}, \boldsymbol{y}) < \varepsilon\}$ be the open Euclidean ball around $\boldsymbol{x}$, and $\mathcal{B} = \{B(x, \varepsilon) \mid \boldsymbol{x} \in \mathbb{R}^L, \varepsilon > 0\}$. We will obtain the topology $(\boldsymbol{X} = \mathbb{R}^L, \mathcal{A})$ if $\mathcal{A}$ is set to be the collection (finite or infinite) of union of any subset of open balls from $\mathcal{B}$, as well as the intersection of any finite subset of open balls from $\mathcal{B}$.

Consider a topological space $(\boldsymbol{X}, \mathcal{A})$, and a subset $\boldsymbol{Y}$ of $\boldsymbol{X}$, we introduce the *induced topology* of $\boldsymbol{Y}$ which is inherited from $\mathcal{A}$ as follows:

Definition 4 (Induced topology)**.** *Given a topological space* $(\boldsymbol{X}, \mathcal{A})$*, the induced topology* $\mathcal{A}_{\boldsymbol{Y}}$ *of subset* $\boldsymbol{Y} \subseteq \boldsymbol{X}$ *is defined as the collection of the intersection between open sets in* $\mathcal{A}$ *and* $\boldsymbol{Y}$*.*

The concept of induced topology gives us a convenient when talking about the topology of a subset. In many scenarios, the explicit reference of topology $\mathcal{A}$ can be omitted of the choice of $\mathcal{A}$ is obvious, for example, in the case of induced topology.

Next, we introduce the concept of *continuity* to compare two topological spaces via a map between them. This is an important concept in topology that are relevant to topological invariants such as number of holes in the object. Here, topological invariants are the properties of topological spaces that do not vary with certain types of continuous deformations, Let us remind a famous example with a coffee mug and a donut, if we perform a few transformations on their mouldable clay models such as stretching, bending or compressing without tearing or gluing the object, we can make an object with the mug shape from an object with the donut shape, and vice versa. From a topology standpoint, both of the coffee mug and the donut are topological equivalent in a sense that they have the same invariant properties, i.e., a single visible cavity and are one complete connected object, and we can apply a continuous transformation from the coffee mug to the donut. Interestingly, we can make a consideration about the closeness relationships between points. If we perform continuous transformations from the coffee mug to the donut, then any two points that were "close" before the deformation are still "close" after the deformation. From topological perspective, all the closeness relations between points have been preserved even if the shape looks different from a geometric perspective.

Before moving to the comparison between two topological spaces, we state here the precise definition of continuous function.

Definition 5 (Continuous function)**.** *For two topological spaces* $(\boldsymbol{X}, \mathcal{A}_{\boldsymbol{X}})$ *and* $(\boldsymbol{Y}, \mathcal{A}_{\boldsymbol{Y}})$*, a function* $f : \boldsymbol{X} \to \boldsymbol{Y}$ *is continuous if for any open set* V *in* $\boldsymbol{Y}$*, its preimage* $f^{-1}(V)$ *is also open in* $\boldsymbol{X}$*. A continous function* $f : \boldsymbol{X} \to \boldsymbol{Y}$ *between two topological spaces is also called a map.*

Next, we define the concept of *homeomorphism*, which is the most important concept to study topology.

Definition 6 (Homeomorphism)**.** *Given two topological spaces* $(\boldsymbol{X}, \mathcal{A}_{\boldsymbol{X}})$ *and* $(\boldsymbol{Y}, \mathcal{A}_{\boldsymbol{Y}})$*, a homeomorphism between them is a map* $f : \boldsymbol{X} \to \boldsymbol{Y}$ *such that* f *is bijective and the inverse of* f *is also continuous. Two topological spaces* $(\boldsymbol{X}, \mathcal{A}_{\boldsymbol{X}})$ *and* $(\boldsymbol{Y}, \mathcal{A}_{\boldsymbol{Y}})$ *are homeomorphic, if there is a homeomorphism between them. In this case we denote that* $(\boldsymbol{X}, \mathcal{A}_{\boldsymbol{X}}) \cong (\boldsymbol{Y}, \mathcal{A}_{\boldsymbol{Y}})$*, or* $\boldsymbol{X} \cong \boldsymbol{Y}$ *if we can omit the notion of topology.*

Examples of homeomorphic spaces are an open L-ball and $\mathbb{R}^L$ (with mapping $f(\boldsymbol{x}) = \dfrac{\boldsymbol{x}}{1 - \|\boldsymbol{x}\|}$), two convex body of the same dimension (with mapping as rays from the center that intersect both convex body). Intuitively, if one can deform from either one to the other by bending or stretching, then they are homeomorphic, for example,

a coffee mug and a donut are homeomorphic in this sense. However, this description can be misleading. Some continuous deformations are not homeomorphisms, such as the deformation of a disk into a point. Some homeomorphisms are not continuous deformations, such as the homeomorphism between a trefoil knot and a circle.

Given a topological space $(\boldsymbol{X}, \mathcal{A}_{\boldsymbol{X}})$, we consider its mapping into another target space.

Definition 7. *Given two topological spaces $(\boldsymbol{X}, \mathcal{A}_{\boldsymbol{X}})$ and $(\boldsymbol{Y}, \mathcal{A}_{\boldsymbol{Y}})$ and a mapping $\phi : \boldsymbol{X} \to \boldsymbol{Y}$. ϕ is an immersion if locally for every point $\boldsymbol{x} \in \boldsymbol{X}$, there is a small neighborhood $U(\boldsymbol{x})$ such that ϕ induces an homeomorphism from $U(\boldsymbol{x})$ to $\phi(U(\boldsymbol{x}))$. ϕ is an embedding if it induces a homeomorphism between $\boldsymbol{X}$ and $\phi(\boldsymbol{X})$.*

Homeomorphic is strong condition to compare topological spaces. In very rough speaking, it preserve topological invariants with *certain types* of continuous deformations. It means that there are types of continuous deformations that do not perserve topological invariants, for example, we can deform the disk along radical lines continuously to a single point, although the disk and the point are not homeomorphic. In this case, there is another notion of similarity among topological spaces, i.e., *homotopy equivalence*, that is weaker than homeomorphism, but describe spaces that can be continuously deformed to one another. We first define the concept of *homotopy.*

Definition 8 (Homotopy)**.** *Given two topological spaces $(\boldsymbol{X}, \mathcal{A}_{\boldsymbol{X}})$ and $(\boldsymbol{Y}, \mathcal{A}_{\boldsymbol{Y}})$ and two maps f and g from $\boldsymbol{X}$ to $\boldsymbol{Y}$. A homotopy between them is defined to be a continous function $H : \boldsymbol{X} \times [0,1] \to \boldsymbol{Y}$ such that $H(\boldsymbol{x}, 0) = f(\boldsymbol{x})$ and $H(\boldsymbol{x}, 1) = g(\boldsymbol{x})$ for all $\boldsymbol{x} \in \boldsymbol{X}$. This map H is called a homotopy connecting f and g.*

Intuitively, the second parameter of h can be considered as time that control H to describe a *continous deformation* of f in to g: at time $t = 0$, we have the function f, and at time $t = 1$ we have the function g. Let us look at a few examples

- Consider $\boldsymbol{X} := [0,1]$ and $\boldsymbol{Y}$ is a convex subset of Euclidean space $\mathbb{R}^L$, then there is a linear homotopy given by

$$H : [0,1] \times [0,1] \longrightarrow \boldsymbol{Y} \tag{2.1}$$

$$(s,t) \longmapsto (1-t)f(s) + tg(s). \tag{2.2}$$

- Consider the unit L-disk $\boldsymbol{X} := \boldsymbol{B}^L = \{\boldsymbol{x} \in \mathbb{R}^L \mid \|\boldsymbol{x}\| \leq 1\}$, $f : \boldsymbol{B}^L \to \boldsymbol{B}^L$ is the identity map, and constant function $g : \boldsymbol{B}^L \to \boldsymbol{B}^L$ maps every point in the unit disk to the origin, then f and g are homotopic since we can define the following

homotopy:

$$H : \boldsymbol{B}^L \times [0,1] \longrightarrow \boldsymbol{B}^L \tag{2.3}$$

$$(\boldsymbol{x}, t) \longmapsto (1-t)\boldsymbol{x}. \tag{2.4}$$

We use homotopy to define a homotopy equivalence between two topological spaces.

Definition 9 (Homotopy equivalence)**.** *Two topological spaces* $(\boldsymbol{X}, \mathcal{A}_{\boldsymbol{X}})$ *and* $(\boldsymbol{Y}, \mathcal{A}_{\boldsymbol{Y}})$ *are homotopy equivalence if there exist continuous functions* $f : \boldsymbol{X} \to \boldsymbol{Y}$ *and* $g : \boldsymbol{Y} \to \boldsymbol{X}$ *such that* $f \circ g$ *is homotopic to identity map* $id_{\boldsymbol{Y}}$ *in* $\boldsymbol{Y}$ *and* $g \circ f$ *is homotopic to identity map* $id_{\boldsymbol{X}}$ *in* $\boldsymbol{X}$*. The maps* f *and* g *are called homotopy equivalence in this case. We denote homotopy equivalent relation as* $(\boldsymbol{X}, \mathcal{A}_{\boldsymbol{X}}) \simeq (\boldsymbol{Y}, \mathcal{A}_{\boldsymbol{Y}})$ *or* $\boldsymbol{X} \simeq \boldsymbol{Y}$ *if we can omit the topology notion.*

Intuitively, two spaces $\boldsymbol{X}$ and $\boldsymbol{Y}$ are homotopy equivalent if they can be transformed into one another by bending, shrinking and expanding. Note that, two homeomorphic spaces are also homotopy equivalent, but the converse is not true. More examples of homotopy equivalence are $\mathbb{R}^L$ with a point, a cirle with a plane without a point.

In general, identifying homotopy equivalence via Definition 9 is not easy. We introduce here a special type of homotopy equivalence, that is more intuitive compared to general homotopy equivalence.

Definition 10 (Deformation retraction)**.** *Given a topological space* $(\boldsymbol{X}, \mathcal{A}_{\boldsymbol{X}})$ *and the induced topological space* $(\boldsymbol{Y}, \mathcal{A}_{\boldsymbol{Y}})$ *where* $\boldsymbol{Y}$ *is the subset of* $\boldsymbol{X}$*. A retraction map* r *is a continuous function* $r : \boldsymbol{X} \to \boldsymbol{Y}$ *such that* $r(\boldsymbol{x}) = \boldsymbol{x}$ *for all* $\boldsymbol{x} \in \boldsymbol{Y}$*. A reaction map is called a deformation retraction if it is homotopic to the identity map in* $\boldsymbol{X}$*. In this case,* $\boldsymbol{Y} \subseteq \boldsymbol{X}$ *is a deformation retract of* $\boldsymbol{X}$*.*

Note that a retraction does not preserve much topology, but deformation retraction is a very useful because it implies homotopy equivalence. For example, a circle can be retracted to a point but it is not a deformation retraction. Even though, both homotopy and homeomorphism are abstract concepts with difficulty to examine and compute, especially in high dimensional spaces. To deal with this problem, in section 2.2, a more relaxed, meaningful and computable concept, i.e., simplicial homology will be introduced.

When we study topology, we often mention about the deformation without "tearing a part". If we tear a connected component in the space into two parts, those parts will represent a topological space that is disconnected with different topological invariant than the original topological space of one connected component. To generalize this intuition, we define the connectedness in topology.

Definition 11 (Connectedness)**.** *A topological space* $(\boldsymbol{X}, \mathcal{A})$ *is said to be connected if* $\boldsymbol{X}$ *is not the union of two disjoint nonempty open sets in* $\mathcal{A}$*, and is disconnected if* $\boldsymbol{X}$ *is the union of any two disjoint nonempty open sets in* $\mathcal{A}$*.*

2.2 Simplicial homology

As mentioned in the previous section, it is quite difficult to put the concepts of homotopy and homeomorphism to the computational routine. There is a method, called *homology groups* that is weaker than homotopy but is extremely useful to classify topological spaces. Here, homology is an algebraic object constructed from a topological space that respects to a equivalent relation between two spaces. Two spaces are said equivalent in homology if they can be continuously deformed from one to another. Homology is weaker than homotopy, that is, homotopic spaces will have the same homology groups, but not vice versa. In computational topology, calculating homology means counting the number of "l-dimensional holes" in a space, which can be the number of connected components (0-dimensional holes), number of rings (1-dimensional holes), and number of voids (2-dimensional holes), and so on. Simplicial homology is the construction of homology via a class of spaces called simplicial complexes. Simplicial homology is effectively computable, but requires algebraic set-up, which the subsequent subsections are going to cover.

2.2.1 Simplicial complexes

Before going into the details of homology groups, we start with a discretization but useful representation of an underlying topological space. Specially, we will introduce the concept of *simplicial complexes* in this subsection.

We explain the concept of simplicial complex as a skeleton to extract the topological feature. A simplicial complex is a set of a finite collection of k-simplices (simple pieces) with the conditions presented in Definition 13. Herein, a k-simplex σ built from a set $\mathcal{V} = \{\boldsymbol{v}_0, \ldots, \boldsymbol{v}_k\}$ of $k+1$ affinely independent points is the convex hull of these points, which is denoted as $[\boldsymbol{v}_0, \ldots, \boldsymbol{v}_k]$ (Definition 12). The points of $\mathcal{V}$ are called the vertices of σ, while the l-simplices (with $l < k$) spanned by subset points of $\mathcal{V}$ are called the faces of σ. Intuitively, an k-simplex is a generalization of the notion of a triangle or tetrahedron to arbitrary dimensions. For example, a 0-simplex is a point, a 1-simplex is a line segment (with two end points as its faces), a 2-simplex is a triangle together with its enclosed area (with three edges and three vertices as its faces), and a 3-simplex is a filled tetrahedron (with triangles, edges, and vertices as its faces), and a 4-simplex is beyond of visualization, but it is a filled shape with

Figure 2.1: A simplex is a generalization of the notion of a triangle or tetrahedron to arbitrary dimensions. For example, a 0-simplex is a point, a 1-simplex is a line segment, a 2-simplex is a triangle together with its enclosed area, and a 3-simplex is a filled tetrahedron, and so on.

tetrahedrons, triangles, edges, and vertices as its faces, and so on (Fig. 2.1). For a simplex σ, we define $\dim(\sigma) = k$ if σ is a k-simplex.

Definition 12 (k-simplex)**.** *Consider $\mathcal{V}$ is a set of $k+1$ affinely independent points $\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ in $\mathbb{R}^k$. Then the k-simplex determined by $\mathcal{V}$ is the following set of points in $\mathbb{R}^k$*

$$C = \{\alpha_0 \boldsymbol{v}_0 + \ldots + \alpha_k \boldsymbol{v}_k \mid \sum_{i=0}^{k} \alpha_i = 1 \text{ and } \alpha_i \geq 0 \ \forall i = 0, 1, \ldots, k\}.$$

Definition 13 (Simplicial complex)**.** *The simplicial complex K of dimension k (called as k-complex) is a set of a finite collection of l-simplex ($0 \leq l \leq k$) that satisfying three properties:*

1. *There are k-simplices in K.*
2. *If σ is a simplex in K then any non-empty face of σ also is a simplex in K.*
3. *If σ_1, σ_2 are simplices in K with non-empty intersection, then their intersection is a common face of both σ_1, σ_2.*

Following the Definition 13, a simplicial complex can be roughly formed when we "glue" together different simplices with a condition that the common parts of simplices in the simplicial complex must be the faces of both simplices. For example, Fig. 2.2 illustrates a 1-complex [Fig. 2.2(a)], a 2-complex [Fig. 2.2(b)], a 3-complex [Fig. 2.2(c)], and a shape which is not a simplicial complex since the line segment is connected to the triangle along its edge ant not at one of of its vertices [Fig. 2.2(d)].

In many situations, the input data often is a point cloud, i.e., a set of points that are sampled from some unknown space. Points do not have any interesting topology, and we would like to enforce some connectivity on them and some topology

Figure 2.2: A simplicial complex is a collection of simplices, such that the common parts of simplices in the simplicial complex must be the faces of both simplices. A simplicial complex K is called a n-complex if n is the maximum number such that there is at least one n-simplex in K. The illustration here depicts (a) a 1-complex, (b) a 2-complex, (c) a 3-complex, and (d) not a simplicial complex.

of the underlying space. The well-known simplicial complex spanned by a point cloud are *Vietoris–Rips complex*, *Delaunary complex*, *Alpha complex*, and *Witness complex*, which can be referenced in [7, 51]. One of the popular methods to build simplicial complex from a point cloud (or at least in this thesis) is constructing Vietoris–Rips complex which is an extended version of neighboring graph. Assumed that we have a set of points $\boldsymbol{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$ where $\boldsymbol{p}_i$ is a L-dimensional vector in the Euclidean space $\mathbb{R}^L$ for all $i = 1, 2, \ldots, n$. Given the proximity parameter $\varepsilon > 0$, we consider balls with radius ε around each point $\boldsymbol{x}_i \in \boldsymbol{X}$ as $B_\varepsilon(\boldsymbol{p}_i) = \{\boldsymbol{x} \in \mathbb{R}^L \mid d(\boldsymbol{x}, \boldsymbol{p}_i) \leq \varepsilon\}$, where $d(\boldsymbol{x}, \boldsymbol{y})$ is the Euclidean distance between points $\boldsymbol{x}$ and $\boldsymbol{y}$ in the L-dimensional Euclidean space.

Definition 14 (Vietoris–Rips complex)**.** *A k-dimensional Vietoris–Rips complex $V_{\mathrm{R}}(\boldsymbol{P}, k, \varepsilon)$ is a complex built over $\boldsymbol{P}$ if for every nonnegative integer $l \leq k$, every collection of $l+1$ affinely independent points $\{\boldsymbol{p}_{i_0}, \ldots \boldsymbol{p}_{i_l}\}$ in $\boldsymbol{P}$ forms a l-simplex $[\boldsymbol{p}_{i_0}, \ldots, \boldsymbol{p}_{i_l}]$ in $V_{\mathrm{R}}(\boldsymbol{P}, k, \varepsilon)$ if the balls $B_\varepsilon(\boldsymbol{p}_{i_a})$ and $B_\varepsilon(\boldsymbol{p}_{i_b})$ intersect for every pair $\boldsymbol{p}_{i_a}$ and $\boldsymbol{p}_{i_b}$.*

Next, we define the mathematical concept for holes structure such as connected components and loops, which may be present in the topological spaces created by the simplicial complex. We apply the homology group to quantitatively detecting holes in topological spaces. These holes are quantified by classifying the space that surrounds

them. For example, one measures 1-dimensional holes in a torus by considering loops on its surface. One then classifies them into different types according to whether or not they can be deformed into each other by bending and stretching. In this way, one can distinguish a 2-sphere from a torus by capturing the fact that it is possible to contract any loop on the sphere to a point, whereas there are two distinct loops on the torus surface that cannot be deformed continuously into each other. These loops also cannot be contracted to a point because they surround different holes.

2.2.2 Homology group

We start constructing homology groups by looking at formal sums of simplices. We define $\langle\sigma\rangle$ as the oriented simplex with its orientation is given by an ordering of its vertices with the rule that two orderings define the same orientation if and only if they differ by an even permutation. In this definition, we declare that each oriented simplex is equal to the negative of the simplex with the opposite orientation, for example, $\langle \boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2 \ldots, \boldsymbol{v}_k\rangle = -\langle \boldsymbol{v}_1, \boldsymbol{v}_0, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k\rangle$.

Let K be a simplicial complex. A l-chain is simply a formal sum of l-simplices $\sum_i c_i\sigma_i$, where the coefficients c_i can come from a ring. In practical computations, the coefficients c_i come from a finite field $\mathbb{Z}_p$ (integers modulo p) for a small prime number p. Here, let's take $p = 2$ then we can ignore the orientation of simplex, i.e., we can add two chains of the same dimension to obtain another l-chain such as

$$\gamma = \gamma_1 + \gamma_2 = \sum_i c_i\sigma_i + \sum_i d_i\sigma_i = \sum_i ((c_i + d_i) \mod 2)\sigma_i. \tag{2.5}$$

We define an abstract vector space, called *chain group* $\mathcal{C}_l(K)$ (we abbreviate K later) with its basic consisting of all l-simplices in K, i.e.,

$$\mathcal{C}_l = \left\{ \sum_{\sigma\in K, \dim(\sigma)=l} a_\sigma\sigma \mid a_\sigma \in \mathbb{Z}_2 \right\}, 0 \le l \le k. \tag{2.6}$$

With the above definition, $\mathcal{C}_l$ is a free abelian group as the addition operation is commutative, and its rank is basically the number of elements in a basis, i.e., the number of l-simplicies in K.

To relate the chain groups $\mathcal{C}_l$ with different l, we define a boundary operator.

Definition 15 (Boundary operator)**.** *For each $l \ge 1$, the boundary operator $\partial_l : \mathcal{C}_l \to \mathcal{C}_{l-1}$ is the linear transformation defined by following statements.*

- *The boundary of a l-simplex is the sum of its $(l-1)$-dimensional faces*

$$\partial_l([\boldsymbol{v}_0, \ldots, \boldsymbol{v}_l]) = \sum_{i=0}^{l} (-1)^i [\boldsymbol{v}_0, \ldots, \hat{\boldsymbol{v}}_i, \ldots, \boldsymbol{v}_l], \tag{2.7}$$

where $[\boldsymbol{v}_0, \ldots, \hat{\boldsymbol{v}}_i, \ldots, \boldsymbol{v}_l]$ *is the* $(l-1)$*-simplex obtained from* $[\boldsymbol{v}_0, \ldots, \boldsymbol{v}_l]$ *by removing the vertex* $\boldsymbol{v}_i$*.*

- *The boundary of a* l*-chain is defined as*

$$\partial_l(\sum a_\sigma \sigma) = \sum a_\sigma \partial_l(\sigma). \tag{2.8}$$

We have the following lemma.

Lemma 2.2.1. *The boundary operator satisfies that* $\partial_l \partial_{l+1} = 0$*.*

Proof. Consider a $(l+1)$-simplex $\sigma = [\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{l+1}]$, then

$$\begin{aligned}
\partial_l \partial_{l+1} \sigma &= \partial_l \sum_{i=0}^{l+1} [\boldsymbol{v}_0, \ldots, \hat{\boldsymbol{v}}_i, \ldots, \boldsymbol{v}_{l+1}] = \sum_{i=0}^{l+1} \partial_l [\boldsymbol{v}_0, \ldots, \hat{\boldsymbol{v}}_i, \ldots, \boldsymbol{v}_{l+1}] \\
&= \sum_{i=0}^{l+1} \sum_{j=0}^{i-1} [\boldsymbol{v}_0, \ldots, \hat{\boldsymbol{v}}_j, \ldots, \hat{\boldsymbol{v}}_i, \ldots, \boldsymbol{v}_{l+1}] + \sum_{i=0}^{l+1} \sum_{j=i+1}^{l+1} [\boldsymbol{v}_0, \ldots, \hat{\boldsymbol{v}}_i, \ldots, \hat{\boldsymbol{v}}_j, \ldots, \boldsymbol{v}_{l+1}] \\
&= 0.
\end{aligned} \tag{2.9}$$

It is easy to check that $\partial(\sigma_1 + \sigma_2) = \partial(\sigma_1) + \partial(\sigma_2)$. Therefore $\partial_l : \mathcal{C}_l \to \mathcal{C}_{l+1}$ is a homomorphism, which is a map that preserves additive operation.

Boundary operator connects the chain groups $\mathcal{C}_l$ into a *chain of complex*,

$$\ldots \xrightarrow{\partial_{l+1}} \mathcal{C}_l \xrightarrow{\partial_l} \mathcal{C}_{l-1} \to \ldots \xrightarrow{\partial_2} \mathcal{C}_1 \xrightarrow{\partial_1} \mathcal{C}_0$$

Let's consider the following two subspaces of $\mathcal{C}_l$, which are determined by the kernel and the image of the boundary map.

Definition 16 (Cycles)**.** *A* l*-cycle is* l*-chain* γ *such that* $\partial\gamma = 0$*. The collection of* l*-cycles, denoted by* $\mathcal{Z}_l$ *forms a subgroup of* $\mathcal{C}_l$*, is called the* l*-th cycle group.*

$$\mathcal{Z}_l = \ker(\partial_l : \mathcal{C}_l \to \mathcal{C}_{l-1}) \tag{2.10}$$

Definition 17 (Boundaries)**.** *A* l*-boundary is a* l*-chain* γ *that is the boundary of a* $(l+1)$*-chain, i.e., there exist* $\sigma \in \mathcal{C}_{l+1}$ *such that* $\gamma = \partial_{l+1}\sigma$*. The collection of* l*-boundaries, denoted by* $\mathcal{B}_l$*, forms a subgroup of* $\mathcal{C}_l$*, is called the* l*-th boundary group.*

$$\mathcal{B}_l = \mathrm{Im}(\partial_{l+1} : \mathcal{C}_{l+1} \to \mathcal{C}_l). \tag{2.11}$$

From the lemma 2.2.1, it is easy to check that $\mathcal{B}_l \subseteq \mathcal{Z}_l \subseteq \mathcal{C}_l$.

Homology tends to exclude cycles that also boundaries. Two cycles $z_1, z_2 \in \mathcal{Z}_l$ have an equivalence relation if they are *homologous*, meaning that they differ by a

boundary ($z_1 - z_2 \in \mathcal{B}_l$). The equivalence relation divide the l-cycles $\mathcal{Z}_l$ into a union of disjoint subsets, called homology classes. The dimension of a set of l-th homology class is described in the form of the lth Betti number β_l.

Definition 18. *Let $[z]$ denote the homology class of $z \in \mathcal{Z}_l$. The l-th homology group which represents for l-dimensional holes is defined by*

$$\mathcal{H}_l = \mathcal{Z}_l/\mathcal{B}_l = \{[z] \mid z \in \mathcal{Z}_l\}. \tag{2.12}$$

The l-th Betti number, β_l is defined as the dimension

$$\beta_l = \dim(\mathcal{H}_l) = \dim(\mathcal{Z}_l) - \dim(\mathcal{B}_l). \tag{2.13}$$

Intuitively, β_0 refers to the number of connected components (zero-dimensional holes), β_1 represents the number of loops or enclosed regions (one-dimensional holes), β_2 equals the number of empty void in the space (two-dimensional holes), and so on. Betti number is a topological invariant, in terms that the homology of a topological space does not depend on its triangulation. Another useful topological invariant is *Euler characteristics*, which is defined as follow.

Definition 19 (Euler characteristic)**.** *Give a topological space $(\boldsymbol{X}, \mathcal{A})$, let β_l denotes the l-th Betti number of $(\boldsymbol{X}, \mathcal{A})$. The Euler characteristics of $\boldsymbol{X}$ (we ommit $\mathcal{A}$) is*

$$\chi(\boldsymbol{X}) := \sum_{l=0} (-1)^l \beta_l. \tag{2.14}$$

Lemma 2.2.2 (Euler-Poincare formula)**.** *Given a d-dimensional simplicial complex K with n_l denoting the number of l-simplices in K, then the Euler characteristics can be computed as the alternating sum of the number of simplices in each dimension.*

$$\chi(K) := \sum_{l=0}^{d} (-1)^l n_l. \tag{2.15}$$

We present here in the Fig. 2.3 intuitive examples of holes. A hole is identified via the cycle that surrounds it. In a given manifold, a cycle is a closed submanifold, and a boundary is a cycle that is also the boundary of a submanifold. Holes correspond to cycles that are not themselves boundaries. For instance, a disk is a two-dimensional surface with a one-dimensional boundary (i.e., a circle). If we puncture the disk, we obtain a one-dimensional hole that is enclosed by the circle, which is no longer a boundary [Fig. 2.3(b)(c)]. Similarly, a filled ball is a three-dimensional object with a two-dimensional boundary (i.e., a surface sphere). If we empty the inside of the ball, we obtain a two-dimensional hole that is enclosed by the surface sphere, which is no longer a boundary [Figs. 2.3(d) and 2.3(e)]. Based on these observations, we

Figure 2.3: (a)–(f) Sample manifolds with the number of zero-, one-, and two-dimensional holes listed underneath. (a) The connected component is a zero-dimensional hole. (b)(c) A one-dimensional hole is obtained by puncturing a disk. (d)(e) A two-dimensional hole is obtained by emptying the inside of a ball. (f) Two one-dimensional holes are illustrated as two circles in a torus. (g) Example of a simplicial complex containing 19 points (0-simplices), 24 edges (1-simplices), eight triangular faces (2-simplices), and one filled tetrahedron (3-simplices). There are two one-dimensional holes Ω_1 and Ω_2 in the complex. In this example, all loops $B \to C \to D \to B$, $A \to B \to C \to D \to A$, $B \to E \to C \to D \to B$, and $A \to B \to E \to C \to D \to A$ are 1-cycles because they are closed 1-chains, that is, the closed collection of edges (1-simplices). Each cycle is not a boundary of any 2-chain (collection of triangular faces); thus, it characterizes a one-dimensional hole. Note that these cycles characterize the same hole, Ω_1, because the difference between the two cycles is the boundary of a 2-chain.

can describe and classify holes in the simplicial complex according to the cycles that enclose holes. Given a simplicial complex, we define an l-chain as a collection of l-simplices in the complex. Therefore, in a simplicial complex, we can define an n-cycle as a closed l-chain and an l-boundary as an l-cycle, which is also the boundary of an $(l+1)$-chain. Here, a 0-cycle is a connected component, a 1-cycle is a closed loop, and a 2-cycle is a shell. For instance, in Fig. 2.3(g), all loops $A \to B \to D \to A$, $B \to C \to D \to B$, and $A \to B \to C \to D \to A$ are 1-cycles because they are the closed collection of edges (1-simplices). Furthermore, the loop $A \to B \to D \to A$ is a 1-boundary because it bounds a triangular face (2-simplex). An l-dimensional hole corresponds to an l-cycle that is not a boundary of any $(l+1)$-chain in the simplicial complex. For instance, as illustrated in Fig. 2.3(g), the loops $B \to C \to D \to B$ and $A \to B \to C \to D \to A$ characterize one-dimensional holes because these loops are 1-cycles but are themselves not 1-boundaries. Moreover, two l-cycles characterize the same hole when together they bound an $(l+1)$-chain (i.e., their difference is an l-boundary). Intuitively, the connected components can be considered as zero-dimensional holes, the loops and tunnels as one-dimensional holes, and the cavities and voids as two-dimensional holes.

2.2.3 Homology calculation

The advantage of homology is its convenience in computational routine. We illustrate this fact in this subsection by presenting here the traditional method to compute the homology group information such as the Betti number of a simplicial complex. Note that from the relation between *ker* and *Im* function in linear algebra, we have following relation.

$$\mathcal{C}_l \cong \ker(\partial_l) \oplus \mathrm{Im}(\partial_l) \Rightarrow \mathcal{C}_l \cong \mathcal{Z}_l \oplus \mathcal{B}_{l-1}. \tag{2.16}$$

For our convenient, we denote $n_l = \dim(\mathcal{C}_l)$, $z_l = \dim(\mathcal{Z}_l)$ and $b_l = \dim(\mathcal{B}_l)$. The above relation let us have the following equation.

$$n_l = z_l + b_{l-1}, \qquad \beta_l = z_l - b_l. \tag{2.17}$$

First, we associate the homology group information to a matrix, so-called *boundary matrix* that stores information about the faces of every simplex.

Definition 20 (Boundary matrix)**.** *Let $\mathcal{C}_l = \{\alpha_1, \alpha_2, \ldots, \alpha_{n_l}\}$ and $\mathcal{C}_{l-1} = \{\gamma_1, \gamma_2, \ldots, \gamma_{n_{l-1}}\}$. The boundary operator ∂_l is represented as the $n_l \times n_{l-1}$ matrix B_l such that $B_l[i,j] = 1$ if and only if γ_j is a face of α_i, i.e., $\gamma_j \in \partial_l(\alpha_i)$, and $B_l[i,j] = 0$ otherwise.*

By using Gaussian elimination that maintains the rank of matrix in column operations, we can compute $b_{l-1} = \text{rank}(B_l)$ in $O(n_l^2 n_{l-1})$ time, and thus we can compute z_l by $z_l = n_l - b_{l-1}$. The computation of β_l is straightforward.

2.3 Persistent homology

2.3.1 Motivation

We consider a point cloud $\boldsymbol{X}$, i.e., a set of points sampled from an unknown space $\mathbb{X}$ associated with a distance metric defined on $\mathbb{X} \times \mathbb{X}$. The idea of topological data analysis is to not study on the topological properties of $\boldsymbol{X}$ directly, but rather via the topological cover of $\boldsymbol{X}$. Here, given a parameter ε, the ε-scale topological cover $T(\boldsymbol{X}, \varepsilon)$ of $\boldsymbol{X}$ is defined as the set of all points in $\mathbb{X}$ within distance ε from some point in $\boldsymbol{X}$. With a good sampled data set $\boldsymbol{X}$ and a good choice of ε, the topological cover $T(\boldsymbol{X}, \varepsilon)$ can provide important information about the topological properties of the underlying space. However, the topological properties of $T(\boldsymbol{X}, \varepsilon)$ are not robust to noise in $\boldsymbol{X}$, and the selection of ε is a sensitive problem, especially when the high-dimensional data cannot be visualized directly.

To deal with the above-mentioned problems, the key idea behind persistent homology is to consider the topological properties of the entire family of topological covers $T(\boldsymbol{X}, \varepsilon)$ as ε varies, instead of a single choice of ε. If we consider ε as the spatial resolution to view shape of data $\boldsymbol{X}$, there are many scenarios where the representative topological structures such as holes in the underlying space of $\boldsymbol{X}$ should be the structures appear at different ε. We consider here the Vietoris–Rips construction of simplicial complexes. We can image to center a ball on each points and increase gradually the radius of these balls. We connect an edge between points if their corresponding balls intersect, and filled a triangle between three points when their pairwise balls intersect, and so on. We illustrate this idea in Fig. 2.4 where we consider the data set $\boldsymbol{X}$ sampled from a figure-of-eight shape in two-dimensional space. Starting with $\varepsilon = 0$, $T(\boldsymbol{X}, \varepsilon = 0)$ contains only the discrete points in $\boldsymbol{X}$. As ε increases, connections exist between components in $T(\boldsymbol{X}, \varepsilon)$, enabling us to obtain a sequence of embedded topological cover, i.e., the filtration in the space. Moreover, if ε becomes considerably large, all the components gets connected with each other, we obtain a large and overlapped cover whereby no useful information can be conveyed.

Persistent homology tracks the variation of topological structures such as connected components, loops, and cavities over the filtration. For instance, if we focus on the appearance and disappearance of loop-like structures in the filtration in Fig. 2.4, we can obtain the information of two loops Ω_1 and Ω_2. The loop Ω_1 appears at $\varepsilon = \varepsilon_2$ then disappears at $\varepsilon = \varepsilon_4$, whereas the loop Ω_2 appears at $\varepsilon = \varepsilon_3$ then disappears at

Figure 2.4: The data set $\boldsymbol{X}$ sampled from true space $\mathbb{X}$ is transformed into a filtration of topological covers $T(\boldsymbol{X}, \varepsilon)$. Topological structure changes along the filtration are tracked, such as the merging of clusters, and the emergence and disappearance of loops or circular holes present in the space with increasing ε. For instance, the loop Ω_1 appears at $\varepsilon = \varepsilon_1$ then disappears at $\varepsilon = \varepsilon_3$, whereas the loop Ω_2 appears at $\varepsilon = \varepsilon_2$ then disappears at $\varepsilon = \varepsilon_4$.

$\varepsilon = \varepsilon_5$. Note that, in this filtration, with the single choice of ε, only at $\varepsilon = \varepsilon_3$, two loops appear in the topological cover. However, we cannot distinguish the magnitudes of these loops at this choice of ε. On the other hand, we can distinguish Ω_1 and Ω_2 if we assign to each loop in the filtration a pair of scale called *persistence pair* (b, d), where the loop appears at $\varepsilon = b$ and disappears at $\varepsilon = d$ in the filtration. We call b and d as *birth-scale* and *death-scale* of loop Ω. We also obtain *persistence pair* for each of connected components appear first at $T(\boldsymbol{X}, \varepsilon = 0)$ and merged together along the filtration. The output of persistent homology, which we consider as the *topological persistence features* to represent the shape of $\boldsymbol{X}$ is the collection of persistence pairs for all connected components, loops, and generally, holes in the constructed filtration of $\boldsymbol{X}$. Another representation for the output of persistent homology is a two-dimensional diagram of multiset points, called *persistence diagram*, where each point denotes a persistence pair calculated from the filtration.

In the next subsections, we will go in details the concept of filtration, topological persistence features, and the calculation method.

2.3.2 Filtration

Given a simplicial complex K, a subcomplex of K is a subset of simplices in K that is closed under the face relation in Definition 13. A *filtration* $\mathcal{K}$ of K is a finite sequence of subcomplexes of K that starts with the empty complex and ends with the complete

complex:

$$\mathcal{K} : \varnothing = K_0 \subseteq \ldots \subseteq K_t \subseteq \ldots \subseteq K_n = K_{n+1} = \ldots = K_N = K. \tag{2.18}$$

From the filtration one can define inclusion maps $\iota^j : K_j \to K_{j+1}$ $(0 \leq j < N)$. In fact, for any $i < j$, we have $\iota^{i,j} : K_i \to K_j$ which is defined as $\iota^{i,j} = \iota^{j-1} \circ \ldots \circ \iota_j$.

Note that, if $\iota : X \to Y$ is an inclusion map, it will map a cycle (a boundary) of X to a cycle (a boundary) in Y. Therefore it will map homologous cycles to homologous cycles. Hence, the inclusion maps induce linear maps

$$\zeta_l : \mathcal{H}_l(X) \longrightarrow \mathcal{H}_l(Y), \tag{2.19}$$

$$[\gamma] \longmapsto [\iota(\gamma)], \tag{2.20}$$

where $\mathcal{H}_l(X), \mathcal{H}_l(Y)$ are the l-th homology of simplicial complexes X, Y. The inclusion ζ_l induces a sequence of homomorphism between the homology groups as:

$$\varnothing \to \mathcal{H}_l(K_1) \to \mathcal{H}_l(K_2) \to \ldots \to \mathcal{H}_l(K_n) = \ldots = \mathcal{H}_l(K_N). \tag{2.21}$$

The above sequence associated with the set of homomorphism between them is called a *persistent module*:

$$\mathbb{P} := \{\mathcal{H}_*(K_i) \mid 0 \leq i \leq N\}_{\zeta_*^{i,j} : \mathcal{H}_*(K_i) \to \mathcal{H}_*(K_j)} \tag{2.22}$$

When the choice of filtration and maps are clear, one can write a persistent module as $\mathbb{P}(\{K_i\}_\zeta)$ or simply $\mathbb{P}$.

2.3.3 Persistence homology group

One is interested in how a homological class evolves along the sequence in Eq. (2.21), i.e., determining when a homology class first appears and when it disappears.

Definition 21 (Birth and death of a homolology class)**.** *A homology class $h \in \mathcal{H}_l(K_N)$ is born at K_b if $h \in \mathcal{H}_l(K_b)$ but is not in the image of the map $\zeta_l^{b-1} : \mathcal{H}_l(K_{b-1}) \to \mathcal{H}_l(K_b)$.*

Then, h dies entering K_d if the image of the map $\zeta_l^{b-1,d-1} : \mathcal{H}_l(K_{b-1}) \to \mathcal{H}_l(K_{d-1})$ does not contain the image of h, but the image of the map $\zeta_l^{b-1,d} : \mathcal{H}_l(K_{b-1}) \to \mathcal{H}_l(K_d)$ does.

The birth-scale, death-scale, and the persistence of h are $b, d, d-b$, respectively. (b, d) is also called as a persistence pair to represent h.

In some scenarios where the filtration is created by sliding a parameter, for example, the parameter ε in Vietoris–Rips construction of a pointcloud, the birth-scale,

death-scale, and the persistence of h can be considered as $\varepsilon_b, \varepsilon_d$, and $\varepsilon_d - \varepsilon_b$, respectively.

To understand the birth and death of homology class in the filtration, we consider a simple case of the filtration $\varnothing = K_0 \subseteq K_1 \subseteq \ldots \subseteq K_N$ where the simplices in K_N are added consequently in the filtration. If we order the simplices in K_N as $\sigma_1, \sigma_2, \ldots, \sigma_N$, then $K_{i+1} \backslash K_i = \sigma_i$, i.e., we add the simplex σ_i to K_i to obtain K_{i+1}.

We denote $l = \dim(\sigma_i)$, then adding σ_i only changes the l-th chain group with following cases:

1. σ_i **create a new l-cycle.**
   This l-cycle contains σ_i but no $(l+1)$-chain can have σ_i on the boundary. Then, the new l-cycle must belong to a new homology class. Notice that a family of a new cycles may be added to the filtration, but they are homologous, i.e., belong to a new l-homology class. In this case, a *birth* event happens, and β_l++, z_l++, while other ranks remain the same. Thus, there is a homology class $h \in \mathcal{H}_l(K_{i+1})$ such that it does not have pre-image under ζ_l^i in $\mathcal{H}(K_i)$.

2. σ_i **does not create a new l-cycle.**
   The boundary of σ_i was a non-trivial and filled $(l-1)$-cycle in K_{i+1}, and now it is filled. In this case, a *death* event happens, and $\beta_{l-1}--, b_{l-1}++$, while other ranks remain the same. Thus, there is a non-trivial homology class $h \in \mathcal{H}_l(K_i)$ such that its image $\zeta_l^i(h)$ is zero in $\mathcal{H}_l(K_{i+1})$.

Intuitive, persistence pairing (i, j) tells us that, the addition of σ_j will kill a homology class which is created when adding σ_i. In other words, there is a l-cycle containing σ_i that is filled only when adding σ_i but not before.

Next, we define *persistent homology group.*

Definition 22 (Persistent homology groups)**.** *The l-th persistent homology groups is the image of the homomorphisms induced by inclusion, that is $\mathcal{H}_l^{i,j} := \mathrm{Im}(\zeta_l^{i,j})$ for any $0 \le i < j \le N$. The l-th persistence Betti numbers are the ranks of these groups: $\beta_l^{i,j} = rank(\mathcal{H}_l^{i,j})$.*

2.3.4 Persistence diagram and topological persistence features

Persistent homology group $\mathcal{H}_l^{i,j}$ is the set of homology classes that was born in $\mathcal{H}_l(K_i)$ and appears in the filtration until $\mathcal{H}_l(K_j)$. The persistence Betti numbers which are the ranks of such classes, indicate the number of homologous independent l-cycles that survive. We need one more step to explore the "lifetime" of these cycles.

We can visualize persistent homology groups through a barcode plot which records the intervals that homology classes are born and die, or through multi-set points in the

Figure 2.5: The visualization for the output of persistent homology of 0-dimensional and 1-dimensional homology classes. Each homology class in the filtration corresponds with a persistence pair (b, d) represented for the values of ε at the emergence (*birth-scale*, $\varepsilon = b$) and the disappearance (*death-scale*, $\varepsilon = d$) of the class. These bars (0-dimensional homology classes $\mathcal{H}_0$ in blue and 1-dimensional homology classes $\mathcal{H}_1$ in red) begin at the values of the birth-scale, then end at values of the death-scale in the persistence pairs. The collection of these bars is called a *barcode*, and the representation of all persistence pairs in a two-dimensional diagram ($\mathcal{H}_0$ in blue circles and $\mathcal{H}_1$ in red circles) is called a *persistence diagram*. Note that, points in the diagram are multi-set points.

two-dimensional Cartesian plane (called *persistence diagram*) (2.5). In the Vietoris–Rips filtration, the birth-scale ε_b and the death-scale ε_d of a homology class can be represented as a point in the diagram with coordinate is $(\varepsilon_b, \varepsilon_d)$. In the persistence diagram, points far from the diagonal generally correspond to robust features, which are persistent over a long time, whereas those near the diagonal are regarded as noise in the data.

Definition 23 (Persistence diagram)**.** *We consider the quantity* $\mu_l^{i,j} := (\beta_l^{i,j-1} - \beta_l^{i,j}) - (\beta_l^{i-1,j-1} - \beta_l^{i-1,j})$*. The* l*-the persistence diagram* $D(\mathbb{P})$ *of a persistence module* $\mathbb{P}$ *is the multiset of points in two-dimensional plane where* $(x, y) \in D(\mathbb{P})$ *if* $\mu_l^{x,y} \neq 0$*, and the multiplicity of* (x, y) *is* $\mu_l^{x,y}$*. Each point* $(x, y) \in D(\mathbb{P})$ *is called a persistence pair with the persistence defined as* $|y - x|$*, which is the lifetime of the corresponding homology class.*

In this thesis, we define topological persistence features as persistence pairs in the persistence diagram.

2.3.5 Persistent homology calculation

The algorithm to calculate persistent homology takes the input as the compatible ordering simplices $\sigma_1, \ldots, \sigma_n$ induced from the filtration $\mathcal{K} : \varnothing = K_0 \subseteq K_1 \subseteq \ldots \subseteq K_N = K$. Here, the compatible means that the simplies in each complex K_i in the filtration precede the ones in $K \backslash K_i$, and the faces of a simplex precede the simplex. The output of the algorithm is the set of all persistence pairs, which is so-called the persistence diagram. Note that if we need to calculate l-th persistent homology groups, it is sufficient to list the simplices until $(l+1)$ dimension.

It turns out that we can extend the homology calculation to the setting of persistent homology. In fact, we build a boundary matrix for the full filtration in the same way we built individual boundary matrices In the homology calculation, we set up the boundary matrices to compute number of l-th homology classes by setting the columns to represent the l-simplices in the l-chain group and the rows to represent the $(l-1)$-simplices in the $(l-1)$-chain group. Here, we extend the construction of the boundary matrix to an entire filtration of complex, that is a square matrix where the columns and rows represent the simplices in the filtration in their total order. Specially, we construct a square matrix B of size $n \times n$ such that $B[i,j] = 1$ if the simplex σ_i is a face of simplex σ_j of codimension 1; and $B[i,j] = 0$ for otherwise. We will do the computations on the boundary matrix B.

Note that, the boundary matrix stores information about linear maps from a k-chain group to the $(k-1)$-chain group $(1 \leq k \leq l+1)$. Therefore, the result of multiplying the boundary matrix to any element in k-chain will be an element in $(k-1)$-chain. To find the bases that form cycles and properties, we reduce the matrix B into Smith normal form or the reduction form, that is finding two matrices S, T such that the matrix $R = SBT$ is a matrix with its nonzero values $(= 1)$ only appear on the diagonal. The reduction form of boundary matrix B contains the information about when certain homology class at each dimension are born or die in the filtration.

Let $low(j)$ be the row number of the lowest non-zero entry in column j. If the entire column in zero, we set $low(j) = 0$ if all elements in column j are zero. B is called *reduced* if each row has at most entry that is the lowest value $(= 1)$ for a column. The following standard algorithm reduces B in $O(n^3)$ time by operating on the columns of B. One can refer to Ref. [7, 51] for more details concerning the algorithms and their improvement versions to compute persistent homology.

Once the boundary matrix is reduced, we can read off the topological features, i.e., the persistence pairs in the following way:

- If $low(j) = i$, then the simplex σ_j is paired with σ_i, and the entrance of σ_i causes the birth of a homology class that dies with the entrance of σ_j.

Algorithm 1 The standard reduction algorithm

```
procedure RIGHT-REDUCTION(B)                    ▷ The reduction algorithm of B
    R ← B_l
    for j = 1 → n do
        while there exists i < j with low(i) = low(j) do
            add column i to column j
        end while
    end for
    return R                                    ▷ The reduction form of B is R
end procedure
```

- If $low(j) = 0$ then the entrance of σ_j causes the birth of a homology class and we look to row j to see whether it is paired, i.e., simplex σ_k such that $low(k) = j$. If there is no k then σ_j is unpaired.

If the simplex σ_i is paired with a simplex σ_j, we can define the persistence pair as $(ph(\sigma_i), ph(\sigma_j))$, where for a simplex $\sigma \in K$, $ph(\sigma)$ is the smallest index s such that $\sigma \in K_s$. An unpaired simplex σ_k gives the persistence pair with infinite element $(ph(\sigma_k), \infty)$.

We illustrate the computation of persistent homology here with the filtration and the total order of simplices in Fig. 2.6. The boundary matrix is

$$B = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{2.23}$$

By applying the standard reduction algorithm (one first adds column 5 to column 6, and then column 4 to column 6), we obtain the reduction matrix:

$$R = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{2.24}$$

The persistence pairs can be obtained from the reduction matrix R as follow:

Figure 2.6: Example of a filtration with a total order on the simplices where σ_i denotes the i-th simplex.

- $low(1) = 0$ then σ_1 is emergent but unpaired; this gives the persistence pair $(1, \infty)$ in $\mathcal{H}_0$.
- $low(4) = 2$ then σ_2 is emergent and paired with σ_4 but both of them enter at the same time in the filtration; this gives no persistence pair.
- $low(5) = 3$ then σ_3 is emergent and paired with σ_5; this gives the persistence pair $(2, 3)$ in $\mathcal{H}_0$.
- $low(7) = 6$ then σ_6 is emergent and paired with σ_7; this gives the persistence pair $(3, 4)$ in $\mathcal{H}_1$.

2.4 Stability of persistent diagrams

One of the main advantages of persistent homology is its output, i.e., the persistence diagrams are proven to be stable with respect to the data being perturbed by noise. To evaluate the stability of the persistence diagram, we introduce the concept of *k-th* Wasserstein distance as a metric structure for comparing the persistence diagrams.

Given two persistence diagrams D_1 and D_2, consider all matchings ψ such that a point on one diagram can be matched either to a point on the other diagram or to its projection on the diagonal line $b = d$. For each pair $(\boldsymbol{p}, \boldsymbol{q}) \in \psi$ for which $\boldsymbol{p} = (b_1, d_1)$ and $\boldsymbol{q} = (b_2, d_2)$, we denote the distance metric on $\mathbb{R}^2$ between $\boldsymbol{p}$ and $\boldsymbol{q}$ as $d(\boldsymbol{p}, \boldsymbol{q})$.

Definition 24 (The Wasserstein distance)**.** *The k-th Wasserstein distance ($k \geq 1$) between two diagrams D_1 and D_2 is defined as*

$$W_k(D_1, D_2) := \inf_{\psi} \left[\sum_{(\boldsymbol{p},\boldsymbol{q}) \in \psi} d^k(\boldsymbol{p}, \boldsymbol{q}) \right]^{1/k} \tag{2.25}$$

For $k = \infty$ and $d(\boldsymbol{p}, \boldsymbol{q})$ is the infinity-norm distance $d^{(\infty)}(\boldsymbol{p}, \boldsymbol{q}) = \max(|b_1 - b_2|, |d_1 - d_2|)$, the Wasserstein distance becomes the bottleneck distance which is defineds as the infimum of the longest matched infinity-norm distance over all matchings ψ.

Definition 25 (The bottleneck distance)**.** *The bottleneck distance between two diagrams D_1 and D_2 is defined as*

$$d_B(D_1, D_2) := \inf_{\psi} \max_{(\boldsymbol{p},\boldsymbol{q})\in\psi} d^{(\infty)}(\boldsymbol{p}, \boldsymbol{q}) \tag{2.26}$$

The research in the stability of persistent homology, i.e., the robustness against the small perturbations in the input data, is referred profoundly in Ref. [52]. It is proven that under favorable conditions, the bottleneck distance of persistent diagrams are bounded with respect to suitable distance functions on the filtration. Here, we introduce the simple form of stability theorem for persistent homology of points embedded in the Euclidean space, which is used in our thesis.

Theorem 2.4.1 (Stability theorem)**.** *Let X and Y be finite sets of points embedded in the Euclidean space $\mathbb{R}^n$. Denote their two-dimensional persistent diagrams as $D(X)$ and $D(Y)$, respectively. Then,*

$$d_B(D(X), D(Y)) \leq 2d_H(X, Y), \tag{2.27}$$

where $d_H(X, Y)$ is the Hausdorff distance given by

$$d_H(X, Y) = \max\left\{\max_{\boldsymbol{x}\in X}\min_{\boldsymbol{y}\in Y} d(\boldsymbol{x}, \boldsymbol{y}), \max_{\boldsymbol{y}\in Y}\min_{\boldsymbol{x}\in X} d(\boldsymbol{x}, \boldsymbol{y})\right\}.$$

Here, $d(\boldsymbol{x}, \boldsymbol{y})$ is the Euclidean distance metric between $\boldsymbol{x}, \boldsymbol{y}$.

Chapter 3

Variant Topological Data Analysis

This chapter presents the main contribution of our thesis, *variant topological data analysis*, which is a general framework grounded on topological data analysis to encode complex patterns in data as quantitative features over variable scales. It is motivated by the existence of variant-scale in the dynamics of the data while other standard approaches encounter difficulty to represent. Specially, we present a novel set of features, i.e., variant topological features that are relevant with topological information of the original space of sampled data, as well as provide insights into the variant-scale and dynamics in data. We briefly define the pipeline to compute variant topological features in section 3.1 and the distance to compare these features in section 3.2. Then we show in section 3.3 how to integrate these features into the kernel method of machine learning techniques to apply in statistical-learning tasks, such as classification and change point detection.

3.1 Variant topological features

The framework of variant topological data analysis is simply illustrated in Fig. 3.1. Here, data is embedded in a high-dimensional space via a mapping under a scale parameter τ of underlying dynamics, for example, time delay in delay embedding of time series, or time scale in diffusion process on the network. The next step is to use a topological machine, which adopts persistent homology theory to calculate the topological persistence features, i.e., the persistence diagram of the mapped object in the embedded space (Chapter 2). Note that these topological persistence features are calculated at each scale τ associated with the mapping of the data. Finally, by considering these features over variable scales τ, we obtain *variant topological features* of the data, i.e., three-dimensional persistence diagram, with the scale τ serving as an additional dimensional in the topological feature space.

Definition 26 (Variant topological features)**.** *We consider a mapping of data $\boldsymbol{X}$ into a set $P_\tau(\boldsymbol{X})$ of discrete points in high-dimensional space with distance metric defined*

Figure 3.1: The framework of variant topological data analysis. First, data is embedded in a high-dimensional space via a mapping under a scale parameter τ of underlying dynamics. Then, at each scale τ, we use a topological persistence machine, which adopts persistent homology theory to calculate the topological persistence features, i.e., the persistence diagram of the mapped object in the embedded space. Finally, by considering these features over variable scales τ, we obtain *variant topological features* of the data, i.e., a three-dimensional persistence diagram, with the scale τ serving as an additional dimensional in the topological feature space.

on this space, for example, a L-dimensional Euclidean space $\mathbb{R}^L$ ($P_\tau(\boldsymbol{X}) \in \mathbb{R}^L$). Here, τ is the scale parameter to represent the variant-scale of underlying dynamics in $\boldsymbol{X}$. We denote $D_\tau^{(2)}(\boldsymbol{X})$ as the (two-dimensional) persistence diagram calculated from a filtration built on $P_\tau(\boldsymbol{X})$. Given a domain $\mathcal{T}$ of τ, we define the three-dimensional persistence diagram, which is called variant topological features as follow:

$$D^{(3)}(\boldsymbol{X}) := \{(b, d, \tau) \mid (b, d) \in D_\tau^{(2)}(\boldsymbol{X}), \tau \in \mathcal{T}\}. \tag{3.1}$$

3.2 Distance between variant topological features

To demonstrate the useful usage of variant topological features in applications, we show that the three-dimensional persistence diagram is stable with respect to the data being perturbed by noise. To evaluate the stability of the three-dimensional persistence diagram, we extend the concept of *bottleneck distance* that is mentioned in Chapter 2 as a metric structure for comparing the persistence diagrams. Given two three-dimensional persistence diagrams $D_1^{(3)}$ and $D_2^{(3)}$, consider all matchings ψ such that a point on one diagram can be matched either to a point on the other diagram or to its projection on the diagonal plane $\mathcal{W}^{(3)} = \{(b, b, \tau) \mid b, \tau \in \mathbb{R}\}$ (Fig. 3.2(a)). For each pair $(\boldsymbol{p}, \boldsymbol{q}) \in \psi$ for which $\boldsymbol{p} = (b_1, d_1, \tau_1)$ and $\boldsymbol{q} = (b_2, d_2, \tau_2)$, we define the *relative infinity-norm distance* between $\boldsymbol{p}$ and $\boldsymbol{q}$ as $d_\xi^{(\infty)}(\boldsymbol{p}, \boldsymbol{q}) = \max(|b_1 - b_2|, |d_1 - d_2|, \xi|\tau_1 - \tau_2|)$,

Figure 3.2: (a) Bottleneck distance between two diagrams $D_1^{(3)}$ (green points) and $D_2^{(3)}$ (red points). The τ axis is rescaled with the positive coefficient ξ to adjust the scale difference between the point-wise distance and time. A green (red) point is matched to either a red (green) point or its projection on the diagonal plane, $\mathcal{W}^{(3)}$. The distance between two matched points is the longest edge of their differences along any coordinate dimension. The bottleneck distance is defined as the longest distance between two matched points that is minimal compared to all other matchings. (b) Kernel between $D_1^{(3)}$ (green points) and $D_2^{(3)}$ (dark red points). The faint red points are symmetric to the dark red points with respect to $\mathcal{W}^{(3)}$. Points near $\mathcal{W}^{(3)}$, i.e., $\boldsymbol{p}_1, \boldsymbol{q}_2$, exhibit less influence, whereas the pairs of points far away from $\mathcal{W}^{(3)}$, i.e., $(\boldsymbol{p}_2, \boldsymbol{q}_1), (\boldsymbol{p}_2, \boldsymbol{q}_3), (\boldsymbol{p}_3, \boldsymbol{q}_1), (\boldsymbol{p}_3, \boldsymbol{q}_3)$, make essential contributions to the kernel.

where ξ is a positive rescaling coefficient introduced to adjust the scale difference between the point-wise distance and time. The bottleneck distance $d_{B,\xi}^{(3)}(D_1^{(3)}, D_2^{(3)})$ can be defined as the infimum of the longest matched relative infinity-norm distance over all matchings ψ:

$$d_{B,\xi}^{(3)}(D_1^{(3)}, D_2^{(3)}) = \inf_{\psi} \max_{(\boldsymbol{p},\boldsymbol{q})\in\psi} d_{\xi}^{(\infty)}(\boldsymbol{p}, \boldsymbol{q}). \tag{3.2}$$

3.3 Kernel methods for variant topological features

In the statistical-learning tasks, many learning algorithms require an inner product between the data in the vector form. Because the space of three-dimensional persistence diagrams is not a vector space, we deem it not straightforward to use the scale-variant topological features in the statistical-learning tasks. This problem can be mitigated through the use of a feature map Φ from the positive-definite kernel, which maps the scale-variant topological features to a space called *kernel-mapped feature space* $\mathscr{H}_b$ where we can define the inner product [24]. In general, choosing the explicit form of mapping a persistence diagram E to Φ_E in the kernel-mapped feature

space is not discernible. Nonetheless, we can use a kernel function to compute the inner product in the kernel-mapped feature space, leaving the mapping function and the kernel-mapped feature space completely implicit. Here, based on previous works on the kernel of two-dimensional persistence diagrams [53, 54], we present here two methods as *persistence scale kernel* [53], and *persistence fisher kernel* [54] to construct the kernel for three-dimensional persistence diagrams.

3.3.1 Persistence Scale Kernel

Given a positive bandwidth ς and a positive rescaling coefficient ξ introduced to adjust the scale difference between the point-wise distance and time, based on Refs. [24, 53], we define the kernel $k_{\varsigma,\xi}$ between two three-dimensional persistence diagrams, $D_1^{(3)}$ and $D_2^{(3)}$, as

$$k_{\varsigma,\xi}(D_1^{(3)}, D_2^{(3)}) = \frac{1}{\varsigma\sqrt{2\pi}} \sum_{\substack{\boldsymbol{q}^{(1)}\in E \\ \boldsymbol{q}^{(2)}\in F}} \left(e^{-\frac{d_\xi^2(\boldsymbol{q}^{(1)},\boldsymbol{q}^{(2)})}{2\varsigma^2}} - e^{-\frac{d_\xi^2(\boldsymbol{q}^{(1)},\bar{\boldsymbol{q}}^{(2)})}{2\varsigma^2}} \right), \tag{3.3}$$

where $d_\xi^2(\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}) = |b_1 - b_2|^2 + |d_1 - d_2|^2 + \xi^2|\tau_1 - \tau_2|^2$, $d_\xi^2(\boldsymbol{q}^{(1)}, \bar{\boldsymbol{q}}^{(2)}) = |b_1 - d_2|^2 + |d_1 - b_2|^2 + \xi^2|\tau_1 - \tau_2|^2$, with $\boldsymbol{q}^{(1)} = (b_1, d_1, \tau_1)$ and $\boldsymbol{q}^{(2)} = (b_2, d_2, \tau_2)$, $\bar{\boldsymbol{q}}^{(2)} = (d_2, b_2, \tau_2)$. In our experiments, we use the normalized version of the kernel, which is calculated as

$$k_{\varsigma,\xi}(D_1^{(3)}, D_2^{(3)}) \leftarrow k_{\varsigma,\xi}(D_1^{(3)}, D_2^{(3)}) / \sqrt{k_{\varsigma,\xi}(D_1^{(3)}, D_1^{(3)}) k_{\varsigma,\xi}(D_2^{(3)}, D_2^{(3)})}. \tag{3.4}$$

Because Eq. (3.3) and Eq. (3.4) define the positive-definite kernels in the set of three-dimensional persistence diagrams [24], according to Moore–Aronszajn's theorem [55], there exists a mapping function Φ such that the inner product $\langle \Phi_E, \Phi_F \rangle_{\mathscr{H}_b}$ between $\Phi_{D_1^{(3)}}$ and $\Phi_{D_2^{(3)}}$ in the kernel-mapped feature space $\mathscr{H}_b$ is $k_{\varsigma,\xi}(D_1^{(3)}, D_2^{(3)})$. Therefore, we can use the explicit form of inner product $\langle \Phi_{D_1^{(3)}}, \Phi_{D_2^{(3)}} \rangle_{\mathscr{H}_b}$ in the statistical-learning tasks.

We prove that the proposed kernel for the three-dimensional persistence diagrams is positive-definite. We prove for the case when the positive rescaling coefficient $\xi = 1$. The proof is straightforward for other positive values of ξ.

For each parameter ς and a three-dimensional persistence diagram $D^{(3)}$, we define the following feature mapping $\Phi_\varsigma : \mathcal{D}^{(3)} \to \mathcal{L}^2(\Omega)$, where $\mathcal{D}^{(3)}$ is the space of three-dimensional persistence diagrams, $\Omega = \{\boldsymbol{x} = (x_1, x_2, x_3) \mid x_1, x_2, x_3 \in \mathbb{R}, x_1 \le x_2\}$ and $\mathcal{L}^2(\Omega)$ is the Hilbert space of square-integrable $\mathcal{L}^2$-functions defined on the domain Ω:

$$\Phi_\varsigma(D^{(3)})(\boldsymbol{x}) = \frac{1}{\sqrt{\kappa}} \sum_{\boldsymbol{p}\in D^{(3)}} \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{p}\|^2}{\varsigma^2}\right) - \exp\left(-\frac{\|\boldsymbol{x} - \bar{\boldsymbol{p}}\|^2}{\varsigma^2}\right), \tag{3.5}$$

where $\bar{\boldsymbol{p}}$ is a symmetric point of $\boldsymbol{p}$ with respect to diagonal plane $\mathcal{W}^{(3)}$ on $\mathbb{R}^3$ and κ is a positive value depending on ς, which we will show later.

We show that the kernel defined in Eq. (3.3) is the inner product of Φ_ς on $\mathcal{L}^2(\Omega)$ as

$$k_{\varsigma,\xi}(D_1^{(3)}, D_2^{(3)}) = \langle \Phi_\varsigma(D_1^{(3)}), \Phi_\varsigma(D_2^{(3)}) \rangle_{\mathcal{L}^2(\Omega)} \tag{3.6}$$

$$= \int_\Omega \Phi_\varsigma(D_1^{(3)})(\boldsymbol{x}) \Phi_\varsigma(D_2^{(3)})(\boldsymbol{x}) d\boldsymbol{x}. \tag{3.7}$$

We extend the domain of function $\Phi_\varsigma(D^{(3)})(\boldsymbol{x})$ from Ω to $\mathbb{R}^3$ to obtain a function that is symmetric with respect to the diagonal plane $\mathcal{W}^{(3)}$ (because $\|\boldsymbol{x}-\boldsymbol{p}\| = \|\bar{\boldsymbol{x}}-\bar{\boldsymbol{p}}\|$ and $\|\boldsymbol{x}-\bar{\boldsymbol{p}}\| = \|\bar{\boldsymbol{x}}-\boldsymbol{p}\|$). Then we have

$$\begin{aligned}
&\int_{\Omega^{(3)}} \Phi_\varsigma(D_1^{(3)})(\boldsymbol{x}) \Phi_\varsigma(D_2^{(3)})(\boldsymbol{x}) d\boldsymbol{x} \\
&= \frac{1}{2} \int_{\mathbb{R}^3} \Phi_\varsigma(D_1^{(3)})(\boldsymbol{x}) \Phi_\varsigma(D_2^{(3)})(\boldsymbol{x}) d\boldsymbol{x} \\
&= \frac{1}{2\kappa} \sum_{\substack{\boldsymbol{p} \in D_1^{(3)} \\ \boldsymbol{q} \in D_2^{(3)}}} \left[A(\boldsymbol{p},\boldsymbol{q}) + A(\bar{\boldsymbol{p}},\bar{\boldsymbol{q}}) - A(\bar{\boldsymbol{p}},\boldsymbol{q}) - A(\boldsymbol{p},\bar{\boldsymbol{q}}) \right],
\end{aligned} \tag{3.8}$$

where $A(\boldsymbol{c},\boldsymbol{d})$ is defined as

$$\begin{aligned}
A(\boldsymbol{c},\boldsymbol{d}) &= \int_{\mathbb{R}^3} \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}\|^2 + \|\boldsymbol{x}-\boldsymbol{d}\|^2}{\varsigma^2} \right) d\boldsymbol{x} \\
&= \frac{\varsigma^3 (2\pi)^{3/2}}{8} \exp\left(-\frac{\|\boldsymbol{c}-\boldsymbol{d}\|^2}{2\varsigma^2} \right).
\end{aligned} \tag{3.9}$$

Here, $\boldsymbol{c},\boldsymbol{d} \in \mathbb{R}^3$. Since $\|\boldsymbol{p}-\boldsymbol{q}\|^2 = \|\bar{\boldsymbol{p}}-\bar{\boldsymbol{q}}\|^2, \|\bar{\boldsymbol{p}}-\boldsymbol{q}\|^2 = \|\boldsymbol{p}-\bar{\boldsymbol{q}}\|^2$, and from Eqs. (3.8) and (3.9) we have the closed form of the kernel where $\kappa = \dfrac{\pi^2\varsigma^4}{2}$.

The kernel is positive-definite due to the inner-product nature of the feature mapping. Consider the three-dimensional persistence diagrams $D_1^{(3)}, \ldots, D_N^{(3)}$ of the l-dimensional holes that are needed to compute the kernel. The Gram matrix for these diagrams is defined as $\mathbf{K}_{N\times N}^{(l)} = [k_{ij}]$, whose element k_{ij} is $k_{ij} = k_{\varsigma,\xi}(D_i^{(3)}, D_j^{(3)})$ with $i = 1, \ldots, N$ and $j = 1, \ldots, N$.

3.3.2 Persistence Fisher Kernel

Let $D_{i\Delta}$ and $D_{j\Delta}$ be the point sets obtained by projecting two three-dimensional persistence diagrams D_i and D_j, respectively, on the diagonal plane $b = d$ in the three-dimensional space. Instead of comparing D_i and D_j, Persistence Fisher kernel compares two extended persistence diagrams $D_i' = D_i \cup D_{i\Delta}$ and $D_j' = D_j \cup D_{j\Delta}$ which

have the same number of points. From D'_i, the summation of normal distributions can be defined as

$$\rho_{D'_i} = \frac{1}{Z} \sum_{\boldsymbol{\mu} \in D'_i} \mathcal{N}(\boldsymbol{\mu}, \nu \boldsymbol{I}), \tag{3.10}$$

where $\mathcal{N}$ is a Gaussian function with bandwidth ν, $\boldsymbol{I}$ is an identity matrix, and the normalization constant with the integral calculated on $\boldsymbol{\Omega} = D_i \cup D_{i\Delta} \cup D_j \cup D_{j\Delta}$ is

$$Z = \int_{\boldsymbol{\Omega}} \sum_{\boldsymbol{\mu} \in D'_i} \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \nu \boldsymbol{I}) d\boldsymbol{x}. \tag{3.11}$$

The similar formula is defined for $\rho_{D'_j}$. The Persistence Fisher kernel is defined as

$$\mathrm{k_{PF}}(D_i, D_j) = \exp(-\alpha d_{\mathrm{FIM}}(D_i, D_j)), \tag{3.12}$$

where α is a positive scalar. Herein, the Fisher information metric $d_{\mathrm{FIM}}(D_i, D_j)$ between $\rho_{D'_i}$ and $\rho_{D'_j}$ is defined as

$$d_{\mathrm{FIM}}(D_i, D_j) = \arccos\left(\int_{\boldsymbol{\Omega}} \sqrt{\rho_{D'_i}(\boldsymbol{x}) \rho_{D'_j}(\boldsymbol{x})} d\boldsymbol{x} \right). \tag{3.13}$$

Note that the Persistence Fisher kernel $\mathrm{k_{PF}}$ is also positive definite kernel of a reproducing kernel Hilbert space with $0 < \mathrm{k_{PF}}(D_i, D_j) \leq 1$, and if $D_i = D_j$ then $\mathrm{k_{PF}}(D_i, D_j) = 1$. Therefore, we can define the squared distance $d^2_{\mathrm{k_{PF}}}(D_i, D_j)$ induced by $\mathrm{k_{PF}}$ between two diagrams D_i and D_j as

$$\begin{aligned} d^2_{\mathrm{k_{PF}}}(D_i, D_j) &= \mathrm{k_{PF}}(D_i, D_i) + \mathrm{k_{PF}}(D_j, D_j) - 2\,\mathrm{k_{PF}}(D_i, D_j) \\ &= 2(1 - \mathrm{k_{PF}}(D_i, D_j)). \end{aligned} \tag{3.14}$$

3.3.3 Selection of the kernel parameters

In our experiments, we compute the kernel by taking values of τ as discrete values with sampling interval 1 and set the rescaling coefficient to $\xi = \varsigma$. The selection of kernel bandwidth ς can be chosen by cross-validation; however, as proposed in [56], we present here a heuristic way to select ς. Consider the three-dimensional persistence diagrams $D^{(3)}_1, D^{(3)}_2, \ldots, D^{(3)}_N$ that are required to compute the kernel. We denote $\varsigma^2_s = \mathrm{median}\{(b_i - b_j)^2 + (d_i - d_j)^2 \mid (b_i, d_i, \tau_i), (b_j, d_j, \tau_j) \in D^{(3)}_s\}$ with $s = 1, 2, \ldots, N$. ς is set as $\varsigma^2 = \frac{1}{2}\mathrm{median}\{\varsigma^2_s \mid s = 1, \ldots, N\}$, such that $2\varsigma^2$ takes values close to many $(b_i - b_j)^2 + (d_i - d_j)^2$ values.

3.3.4 Using kernels with holes of multiple dimensions

For all l-dimensional holes, we obtain the persistence diagrams and compute their kernels as k_{ς_l,ξ_l}, where ς_l is the bandwidth of this kernel and ξ_l is the positive rescaling coefficient corresponding with l-dimensional holes. To use persistence diagrams for different dimensions of holes, we can combine the kernels at various dimensions through linear combinations. In our experiments, we only consider the topological features of zero-dimensional holes (connected components) and one-dimensional holes (loops). Thus, the combined Gram matrix of N data can be defined as

$$\mathbf{K}_{N\times N} = \alpha_0 \mathbf{K}^{(0)}_{N\times N} + \alpha_1 \mathbf{K}^{(1)}_{N\times N}, \tag{3.15}$$

where $0 \le \alpha_0, \alpha_1 \le 1, \alpha_0 + \alpha_1 = 1$, and $\mathbf{K}^{(0)}_{N\times N}$ and $\mathbf{K}^{(1)}_{N\times N}$ are the Gram matrices of N persistence diagrams at the zero-dimensional and one-dimensional holes, respectively. We can choose α_0, α_1 by cross-validation.

3.3.5 Kernel Fisher discriminant ratio

We can use the above-defined kernels to estimate the transition points with respect to the series of diagrams $\mathcal{D} = \{D^{(3)}_1, D^{(3)}_2, \ldots, D^{(3)}_N\}$. Here, we define the transitions as they abruptly change at given unknown instants (change-points) in $\mathcal{D}$. We use the kernel change-point detection method [57] to solve the change-point regression problem with $\Phi_{D^{(3)}_1}, \Phi_{D^{(3)}_2}, \ldots, \Phi_{D^{(3)}_N}$. Given an index s $(1 < s \le N)$, we calculate the kernel Fisher discriminant ratio $\kappa_{N,s}(\mathcal{D})$, which is a statistical quantity to measure the dissimilarity between two classes assumptively defined by two sets of diagrams having index before and from s. Here, the index s achieving the maximum of $\kappa_{M,s}(\mathcal{D})$ corresponds to the estimated transition point.

The corresponding empirical mean and covariance functions in $\mathscr{H}_b$ associated with the data in $\mathcal{D}$ having index before and from s are defined as

$$\hat{\mu}_1 = \frac{1}{s-1}\sum_{i=1}^{s-1} \Phi_{D^{(3)}_i}, \tag{3.16}$$

$$\hat{\Sigma}_1 = \frac{1}{s-1}\sum_{i=1}^{s-1} \left\{\Phi_{D^{(3)}_i} - \hat{\mu}_1\right\} \otimes \left\{\Phi_{D^{(3)}_i} - \hat{\mu}_1\right\}, \tag{3.17}$$

$$\hat{\mu}_2 = \frac{1}{M-s+1}\sum_{i=s}^{M} \Phi_{D^{(3)}_i}, \tag{3.18}$$

$$\hat{\Sigma}_2 = \frac{1}{M-s+1}\sum_{i=s}^{M} \left\{\Phi_{D^{(3)}_i} - \hat{\mu}_2\right\} \otimes \left\{\Phi_{D^{(3)}_i} - \hat{\mu}_2\right\}. \tag{3.19}$$

Here, $f \otimes g$ for two functions $f, g \in \mathscr{H}_b$ is defined for all functions $h \in \mathscr{H}_b$ as $(f \otimes g)h = \langle g, h\rangle_{\mathscr{H}_b} f$.

The kernel Fisher discriminant ratio $\kappa_{M,s}(\mathcal{D})$ is defined as

$$\kappa_{M,s}(\mathcal{D}) = \frac{(s-1)(M-s+1)}{M} \left\langle \hat{\mu}_2 - \hat{\mu}_1, (\hat{\Sigma} + \eta \boldsymbol{I})^{-1}(\hat{\mu}_2 - \hat{\mu}_1) \right\rangle_{\mathscr{H}_b}, \tag{3.20}$$

where η is a regularization parameter and $\hat{\Sigma} = \frac{s-1}{M}\hat{\Sigma}_1 + \frac{M-s+1}{M}\hat{\Sigma}_2$. Here, the index s achieving the maximum of $\kappa_{M,s}(\mathcal{D})$ corresponds to the estimated transition point.

Chapter 4

Variant Topological Features in Time-series Analysis

Identifying the qualitative changes in time-series data provides insights into the dynamics associated with such data. Such qualitative changes can be detected through topological approaches, which first embed the data into a high-dimensional space using a time-delay parameter and subsequently extract topological features describing the shape of the data from the embedded points. However, the essential topological features that are extracted using a single time delay are considered to be insufficient for evaluating the aforementioned qualitative changes, even when a well-selected time delay is used. We therefore propose a delay-variant embedding method that constructs the variant topological features by considering the time delay as a variable parameter instead of considering it as a single fixed value. This delay-variant embedding method reveals multiple-time-scale patterns in a time series by allowing the observation of the variations in topological features, with the time delay serving as an additional dimension in the topological feature space. We theoretically prove that the constructed topological features are robust when the time series is perturbed by noise. Furthermore, we combine these features with the kernel technique in machine learning algorithms to classify the general time-series data. We demonstrate the effectiveness of our method for classifying the synthetic noisy biological and real time-series data. Our method outperforms a method that is based on a single time delay and, surprisingly, achieves the highest classification accuracy on an average among the standard time-series analysis techniques.

4.1 Time delay embedding of time series

Time-series data can undergo qualitative changes such as transitioning from the quiescent state to oscillatory dynamics through a bifurcation. Identifying such changes enables deep understanding of the underlying dynamics; however, this is challenging when the data are subject to noise. In material science, topological features,

which indicate the "shape" of the data, can be used to detect the qualitative changes, i.e., phase transitions [20, 58] or transitions in morphological and hierarchical structures [12, 14–16]. Because topology is a qualitative property that is stable under the influence of noise, the topological features of time-series data are expected to reflect the qualitative changes in dynamics. These features are constructed through delay embedding, in which a time series $x(t)$ is mapped to m-dimensional points using delay coordinates $[x(t), x(t-\tau), ..., x(t-(m-1)\tau)]$ on the embedded space, where τ denotes the predefined time delay and m denotes the embedding dimension. Further, the embedded points form geometric features, such as clusters and loops, and the topological features, which monitor the emergence and disappearance of geometric features, can be used to characterize the dynamics of the system [19, 21].

Theoretically, for a noise-free time series of unlimited length, embeddings with the same value of m but different values of time delay τ are considered to be equivalent in terms of "optimal" reconstruction [48]. Herein, the reconstruction that preserves the invariants of dynamics, such as the fractal dimension and the Lyapunov exponent, when the time series is embedded can be referred to as optimal reconstruction. However, because a real time series is noisy and because of finite length, the selection of τ is considered to be an inherently difficult problem. Instead of being based on mathematically rigorous criteria, majority of the methods that are required for the selection of τ are observed to be based on heuristics, i.e., they involve a tradeoff between *redundance* and *irrelevance* in case of successive delay coordinates [59]. Traditional methods determine τ as the first time scale that maximizes the indices of independence such as linear independence and nonlinear independent information [60], and the correlation sum [61]. Meanwhile, the geometry-based strategies that are used for estimating τ examine the measures that are required for expanding the attractor in the reconstructed phase space [62–64]. However, the optimal value of τ exhibits no definite theoretical properties because τ has no theoretical relevance in the mathematical framework of delay embedding [65]. Furthermore, there is no universal strategy for selecting τ, and the value of τ that works well for one application may not work well for another application even though the same data may be used [66]. Therefore, the optimal selection of τ depends on the objective of the analysis and is obtained by trial and error without using any systematic method. This defect limits the power of topological features in time-series data analysis.

4.2 Variant topological features from delay-variant embedding

Herein, we propose a *delay-variant embedding* method in which the topological features are constructed by considering τ as the variable parameter; using this method, the topological changes are monitored in the embedded space. Embedding with a single value of τ is sensitive to noise; further, considering a range of τ can provide useful information with which the dynamics can be understood. We theoretically prove the stability of constructed features against noise and apply these features to classification of several time-series datasets. We demonstrate that our method outperforms a method that is based on a single value of τ in classifying the oscillatory activity of synthetic noisy biological data. Surprisingly, in classifying real time-series data, our approach demonstrates a higher accuracy on an average when compared to several standard techniques that are used for performing time-series analysis. The source code used in our experiments is publicly available on GitHub [67].

For the observed time series $x(t)$, we use $D^{(2)}_{(m),\tau}(x(t))$ to denote the two-dimensional persistence diagram calculated from the embedded points with time delay τ and embedding dimension m. We consider τ in a predefined set $\mathcal{T} = \{\tau_1, \tau_2, ..., \tau_K\}$, where $\tau_1 < \tau_2 < \cdots < \tau_K$ (K denotes the size of the set). The variant topological features, i.e., three-dimensional persistence diagram for time series $x(t)$ can be defined as

$$D^{(3)}_{(m)}(x(t)) = \left\{(b, d, \tau) \mid (b, d) \in D^{(2)}_{(m),\tau}(x(t)), \tau \in \mathcal{T}\right\}. \tag{4.1}$$

Figure 4.1 shows a schematic of the three-dimensional persistence diagram $D^{(3)}_{(m)}(x(t))$ for loops and tunnels (one-dimensional holes), wherein the embedded points exhibit different shapes and topological features for different values of τ. In the middle panel of Fig. 4.1, there are two big loops for τ_1, whereas loops survive for short times with different distributions of birth and death scales (right panel) for τ_2 and τ_3. This information cannot be obtained using a single value of τ in the embedding.

4.3 Stability with respect to the perturbations of time series

We show that three-dimensional persistence diagrams are stable with respect to the bottleneck distance under perturbation applied to the time series. Given two time series $x(t)$ and $y(t)$ of the same length, let $D^{(3)}_{(m)}(x)$ and $D^{(3)}_{(m)}(y)$ be their three-dimensional persistence diagrams respectively, as calculated for the embedding dimension m. Based on the stability properties of two-dimensional persistence diagrams [68],

Figure 4.1: Time series $x(t)$ is embedded in an m-dimensional space (in this illustration, $m = 3$) through delay embedding (left panel). The embedded points at different values of time delay τ have different geometric features, such as clusters and loops (middle panel). We extract the topological features such as the emergence (birth scale) and disappearance (death scale) of these geometric features. The birth and death scales at each τ are represented as points in a two-dimensional persistence diagram. By observing the manner in which the topological features vary with τ serving as an additional dimension, we obtain the three-dimensional persistence diagram (right panel), which can be considered to be a typical feature of the time series.

we can prove the following stability theorem of three-dimensional persistence diagrams obtained from delay-variant embedding:

Definition 27 (Stability theorem from delay-variant embedding)**.**

$$d^{(3)}_{\mathrm{B},\xi}(D^{(3)}_{(m)}(x), D^{(3)}_{(m)}(y)) \leq 2\sqrt{m}\max_t |x(t) - y(t)| \tag{4.2}$$

for an arbitrary positive ξ*.*

If we identify $y(t)$ as the perturbed data obtained by adding noise to $x(t)$, Eq. (4.2) shows that the upper limit of the bottleneck distance between $D^{(3)}_{(m)}(x)$ and $D^{(3)}_{(m)}(y)$ is governed by the magnitude of the noise. Thus, the inequality of Eq. (4.2) states that our three-dimensional persistence diagrams are robust with respect to the time-series data being perturbed by noise.

We prove the result in Eq. (4.2). First, for our convenience, let us mention again the definition in Chapter 2 of the bottleneck distance $d^{(2)}_B$ between two two-dimensional persistence diagrams $D^{(2)}_1$ and $D^{(2)}_2$ as

$$d^{(2)}_B(D^{(2)}_1, D^{(2)}_2) = \inf_{\gamma} \max_{(\boldsymbol{p},\boldsymbol{q})\in\gamma} \|\boldsymbol{p} - \boldsymbol{q}\|_\infty, \tag{4.3}$$

where γ is a matching between $D_1^{(2)}$ and $D_2^{(2)}$ such that a point on a diagram is matched to a point on the other or to its projection on the diagonal line $\mathcal{W}^{(2)} = \{(b,b) \mid b \in \mathbb{R}\}$. Here the distance $\|\boldsymbol{z}\|_\infty$ is defined as $\|\boldsymbol{z}\|_\infty = \max\{|z_1|, \ldots, |z_n|\}$ for $\boldsymbol{z} = (z_1, \ldots, z_n) \in \mathbb{R}^n$.

The bottleneck distance between the two-dimensional persistence diagrams satisfies the following property [68]:

Proposition. *Let X and Y be finite sets of points embedded in the Euclidean space $\mathbb{R}^n$. Denote their two-dimensional persistent diagrams as $D^{(2)}(X)$ and $D^{(2)}(Y)$, respectively. Then,*

$$d_B^{(2)}(D^{(2)}(X), D^{(2)}(Y)) \leq 2d_H(X,Y), \tag{4.4}$$

where $d_H(X,Y)$ is the Hausdorff distance given by

$$d_H(X,Y) = \max\left\{\max_{\boldsymbol{x}\in X}\min_{\boldsymbol{y}\in Y} d(\boldsymbol{x},\boldsymbol{y}), \max_{\boldsymbol{y}\in Y}\min_{\boldsymbol{x}\in X} d(\boldsymbol{x},\boldsymbol{y})\right\}.$$

Here, $d(\boldsymbol{x},\boldsymbol{y})$ is the Euclidean distance between $\boldsymbol{x},\boldsymbol{y}$.

For each $\tau \in \mathcal{T} = \{\tau_1, \tau_2, ..., \tau_K\}$, we denote $X_{(m)}^\tau$ and $Y_{(m)}^\tau$ are the embedded points of $x(t)$ and $y(t)$, respectively, in an embedding space with dimension m and time delay τ. Consider two two-dimensional persistence diagrams $D^{(2)}(X_{(m)}^\tau)$ and $D^{(2)}(Y_{(m)}^\tau)$ calculated from $X_{(m)}^\tau$ and $Y_{(m)}^\tau$, respectively. Let Γ_τ be the set of matchings defined in Eq. (4.3) between $D^{(2)}(X_{(m)}^\tau)$ and $D^{(2)}(Y_{(m)}^\tau)$. For each collection $\Lambda = \{\gamma_1, \gamma_2, \ldots, \gamma_K \mid \gamma_i \in \Gamma_{\tau_i}, i = 1, 2, \ldots, K\}$, we construct a matching ψ between two three-dimensional persistence diagrams $D_{(m)}^{(3)}(x)$ and $D_{(m)}^{(3)}(y)$, such that, for each $(\boldsymbol{p},\boldsymbol{q}) \in \psi$, then $\boldsymbol{p} = (b_1, d_1, \tau)$, $\boldsymbol{q} = (b_2, d_2, \tau)$, and $(\boldsymbol{p}_\gamma, \boldsymbol{q}_\gamma) \in \gamma$, where $\boldsymbol{p}_\gamma = (b_1, d_1)$, $\boldsymbol{q}_\gamma = (b_2, d_2)$, and $\gamma \in \Lambda \cap \Gamma_\tau$. Let Γ be the set of all matchings ψ constructed this way. From the definition of bottleneck distance, we have the following inequality:

$$d_{B,\xi}^{(3)}(D_{(m)}^{(3)}(x), D_{(m)}^{(3)}(y)) \leq \inf_{\psi\in\Gamma}\max_{(\boldsymbol{p},\boldsymbol{q})\in\psi} d_\xi^{(\infty)}(\boldsymbol{p},\boldsymbol{q}). \tag{4.5}$$

For $(\boldsymbol{p},\boldsymbol{q}) \in \psi$, we have

$$\begin{aligned} d_\xi^{(\infty)}(\boldsymbol{p},\boldsymbol{q}) &= \max\{|b_1 - b_2|, |d_1 - d_2|, \xi|\tau - \tau|\} && (4.6)\\ &= \max\{|b_1 - b_2|, |d_1 - d_2|\} && (4.7)\\ &= \|\boldsymbol{p}_\gamma - \boldsymbol{q}_\gamma\|_\infty, && (4.8) \end{aligned}$$

and Eq. (4.5) becomes

$$
\begin{aligned}
& d_{B,\xi}^{(3)}(D_{(m)}^{(3)}(x), D_{(m)}^{(3)}(y)) \\
& \quad \leq \max_{\tau \in \mathcal{T}} \inf_{\gamma \in \Gamma_\tau} \max_{(\boldsymbol{p}_\gamma, \boldsymbol{q}_\gamma) \in \gamma} \|\boldsymbol{p}_\gamma - \boldsymbol{q}_\gamma\|_\infty \qquad (4.9) \\
& \quad = \max_{\tau \in \mathcal{T}} d_B^{(2)}(D^{(2)}(X_{(m)}^\tau), D^{(2)}(Y_{(m)}^\tau)). \qquad (4.10)
\end{aligned}
$$

From Eq. (4.10) and Proposition A1, we have

$$
d_{B,\xi}^{(3)}(D_{(m)}^{(3)}(x), D_{(m)}^{(3)}(y)) \leq 2 \max_{\tau \in \mathcal{T}} d_H(X_{(m)}^\tau, Y_{(m)}^\tau). \qquad (4.11)
$$

At each time point t_0, we consider $\mathcal{X}_{(m)}^\tau(t_0) = (x(t_0), x(t_0 - \tau), \ldots, x(t_0 - (m-1)\tau)) \in X_{(m)}^\tau$ and $\mathcal{Y}_{(m)}^\tau(t_0) = (y(t_0), y(t_0 - \tau), \ldots, y(t_0 - (m-1)\tau)) \in Y_{(m)}^\tau$. From the definition of Euclidean distance, we have

$$
\begin{aligned}
d(\mathcal{X}_{(m)}^\tau(t_0), \mathcal{Y}_{(m)}^\tau(t_0)) &= \sqrt{\sum_{i=0}^{m-1} [x(t_0 + i\tau) - y(t_0 + i\tau)]^2} \\
&\leq \sqrt{m} \max_t |x(t) - y(t)|. \qquad (4.12)
\end{aligned}
$$

From Eq. (4.12) and the definition of Hausdorff distance, we have the following inequality:

$$
\begin{aligned}
d_H(X_{(m)}^\tau, Y_{(m)}^\tau) &= \max \left\{ \max_{t_1} \min_{t_2} d(\mathcal{X}_{(m)}^\tau(t_1), \mathcal{Y}_{(m)}^\tau(t_2)), \max_{t_2} \min_{t_1} d(\mathcal{X}_{(m)}^\tau(t_1), \mathcal{Y}_{(m)}^\tau(t_2)) \right\} \\
&\leq \max \left\{ \max_{t_1} d(\mathcal{X}_{(m)}^\tau(t_1), \mathcal{Y}_{(m)}^\tau(t_1)), \max_{t_2} d(\mathcal{X}_{(m)}^\tau(t_2), \mathcal{Y}_{(m)}^\tau(t_2)) \right\} \\
&\leq \sqrt{m} \max_t |x(t) - y(t)|. \qquad (4.13)
\end{aligned}
$$

Equation (4.2) is obtained by taking the maximum of Eq. (4.13) over $\tau \in \mathcal{T}$ and applying it to Eq. (4.11).

4.4 Applications

4.4.1 Identify the dynamics of biological model

Initially, the proposed method is applied to classify periodic and aperiodic time series using synthetic single-cell data. This is challenging because of the difficulty associated with discriminating between an oscillation containing noise and a mere noisy fluctuation [69]. We generate synthetic mRNA and protein time-series data from a stochastic model of the *Hes1* genetic oscillator [70, 71] exhibiting negative autoregulation with delay. We use the delayed version of the Gillespie algorithm [72, 73] to generate data

Figure 4.2: Schematic of the Hes1 regulatory model. mRNA molecules are produced by the transcription of DNA. The production of Hes1 protein occurs by translation of *hes1* mRNA with delay time ζ. The Hes1 protein represses its own expression by directly binding to its promoter as negative feedback.

from 1,000 cells in both the oscillatory and non-oscillatory parameter regimes [71, 74]. We measure the protein levels after every ν (= 64, 32, 16, 8) min for 4,096 min. We normalize the time series to have zero mean and unit variance; further, we add Gaussian white noise with variance $\sigma_n^2 = 0.1$ to assess the robustness of the method.

We generate data from Hes1 regulatory model, which is a stochastic model of the *Hes1* genetic oscillator exhibiting negative autoregulation with delay [70, 71]. The model describes the concentrations and interactions of two types of particles: *hes*1 mRNA molecules, denoted by M, and Hes1 protein molecules, denoted by P. The stochastic dynamics are defined by the following reactions:

$$M \xrightarrow{\mu_m} \varnothing; \tag{4.14}$$

$$P \xrightarrow{\mu_p} \varnothing; \tag{4.15}$$

$$M \xrightarrow{\beta_p} M + P; \tag{4.16}$$

$$\varnothing \overset{g(n_p),K(\zeta)}{\Longrightarrow} M. \tag{4.17}$$

The degradations of mRNA and protein are described in reactions (4.14) and (4.15), respectively, where the rates of these degradation are μ_m and μ_p, respectively. mRNA molecules are translated into protein via reaction (4.16) by the translation-rate parameter β_p. The final reaction, (4.17) with a double arrow describes the transcription process for producing mRNA which is accompanied by time delay. The rate of *hes*1 mRNA production depends on the concentration of Hes1 protein molecules through

Figure 4.3: Average power spectra of the protein concentrations calculated from Gillespie simulation of 1,000 independent non-oscillating data points (a) and 1,000 oscillating data points (b). The measurements start at $t = 5,000$min for the system to equilibrate. Parameters for (a) are $P_0 = 300, h = 1, \zeta = 0, \beta_m = \beta_p = 1, \mu_m = \mu_p = 0.07, \Theta = 20$, and for (b) are $P_0 = 100, h = 3, \zeta = 18, \beta_m = \beta_p = 1, \mu_m = \mu_p = 0.03, \Theta = 20$. The time series are generated with measurements every 1 min over a period of $t = 9,096$min.

a negative-feedback mechanism, as described by the function

$$g(n_p) = \beta_m \left[1 + [n_p/(P_0\Theta)]^h\right]^{-1}. \tag{4.18}$$

Here, n_p is the number of protein molecules in the system, β_m, P_0, and h are constants, and Θ is the size of the system. This transcription process is associated with a time delay ζ drawn from a distribution $K(\zeta)$; that is, the protein concentration at time $t - \zeta$ only affects the production of mRNA at time t.

In the simulations, the protein levels were measured after every ν (= 64, 32, 16, 8) min for 4,096 min. The measurements start at $t = 5,000$ min for the system to equilibrate. The parameters for the oscillatory regime are $P_0 = 300, h = 1, \zeta = 0, \beta_m = \beta_p = 1, \mu_m = \mu_p = 0.07, \Theta = 20$, and for the non-oscillatory regime are $P_0 = 100, h = 3, \zeta = 18, \beta_m = \beta_p = 1, \mu_m = \mu_p = 0.03, \Theta = 20$.

To confirm that the parameter regimes used in the simulation are oscillatory or non-oscillatory, we apply the Fourier transform to the generated time-series data at the measured interval $\nu = 1$min. Although the power spectrum cannot be used to distinguish two regimes with individual time series, the average power spectrum of

Figure 4.4: (a)(d) Examples of two time series generated from the (a) non-oscillatory and (d) oscillatory regimes of the Hes1 model measured after every 16 min. (b)(e) Two-dimensional persistence diagrams of the loop patterns calculated with embedding dimension $m = 3$ and a single value of τ selected according to [60] for the (b) non-oscillatory ($\tau = 4$) and (e) oscillatory ($\tau = 2$) time series. (c)(f) Three-dimensional persistence diagrams of loop patterns calculated with $m = 3$ and 10 values of τ given by $\tau_1 = 1, \tau_2 = 2, \ldots, \tau_{10} = 10$ for the (c) non-oscillatory and (f) oscillatory time series.

1,000 data points in each parameter regime shows clear oscillatory or non-oscillatory behavior (Fig. 4.3). In the oscillatory regime, there is a clear peak at a frequency of 0.5 hours^{-1}, whereas the average power spectrum in the non-oscillatory regime shows no peak.

Figure 4.4 depicts the examples of the time series and persistence diagrams for the two regimes with an embedding dimension of $m = 3$. Figures 4.4(a) and (d) show the time series generated by the measurements in the non-oscillatory and oscillatory regimes, respectively, after every $\nu = 16$ min, and Figs. 4.4(b) and (e) denote their respective two-dimensional persistence diagrams computed from the single-delay embedding. In Figs. 4.4(b) and (e), τ is selected by the mutual-information method to maximize the statistical measure of nonlinear independence in the delay coordinates of the embedded points [60]. Figures 4.4(c) and (f) depict the three-dimensional persistence diagrams obtained using Fig. 4.4(a) and (d), respectively, with $K = 10$ time-delay values as $\tau_1 = 1, \tau_2 = 2, \ldots, \tau_{10} = 10$. In these examples, it is difficult to distinguish between the two regimes using either the original time series or the two-dimensional diagrams, whereas there is a distinct pattern in the three-dimensional diagrams because all the topological variations are considered when τ changes. In the

three-dimensional diagram of the non-oscillatory data, the points, which represent the loops in the embedded space, are extensively distributed along the birth and death scales, corresponding to the random fluctuation in the time series. In contrast, in the oscillatory data, points are densely distributed along the birth and death scales; this manifests the appearance of repeated loop patterns in the embedded space, thereby indicating periodic patterns in the time series.

Further, we assess the effectiveness of the proposed topological features by classifying the data obtained from the Hes1 model, which can be randomly split with equal probability to belong to the training and test sets. We use the support vector machine [75] to perform classification in the kernel space. In the delay-variant method, we consider $K = 10$ time-delay values as $\tau_1 = 1, \tau_2 = 2, \ldots, \tau_{10} = 10$. In the single-delay method, we choose a single value of τ for each value from $\{\tau_1, \ldots, \tau_{10}\}$. For each embedding dimension of the single-delay method, we use the maximum accuracy in the test dataset over all the values of τ for performing comparison with the delay-variant method. Figure 4.5 depicts the average classification accuracy obtained using 100 random splits at different embedding dimensions m and different measurement intervals $\nu = 8$ (Fig. 4.5(a)), $\nu = 16$ (Fig. 4.5(c)), $\nu = 32$ (Fig. 4.5(e)), and $\nu = 64$ (Fig. 4.5(g)). When the same embedding dimension m is being used, the total dimension for the delay-variant method (mK) is observed to be higher than that for the single-delay method (m). In the classification task, the high-dimensional features tend to achieve high classification accuracies. Therefore, to ensure that a fair comparison between the single-delay method and the delay-variant method can be performed, we define the effective embedding dimension as $M = mK$ and further compare the accuracy for different values of M with $\nu = 8$ (Fig. 4.5(b)), $\nu = 16$ (Fig. 4.5(d)), $\nu = 32$ (Fig. 4.5(f)), and $\nu = 64$ (Fig. 4.5(h)). For $\nu = 64$ (Fig. 4.5(h)), we do not consider embeddings with $M \geq 60$ because the length of the time series is 64.

When the comparison is performed over the embedding dimension m, Fig. 4.5 depicts that both the methods will achieve almost the same accuracy when the time series is long enough ($\nu = 8, 16$) at $m \geq 4$. In the single-delay method, τ should be selected by cross-validation for performing a fair comparison; however, even with trial and error to obtain the maximum accuracy in the test dataset in the single-delay method, our delay-variant method still performs better. When ν is increased, the time series is reduced, thereby increasing the difficulty of classification; regardless, the delay-variant method attains a higher accuracy than the single-delay method. For $\nu = 32, 64$, the delay-variant method outperforms the single-delay method over a wide range of m (4–10) in terms of the accuracy. This observation indicates that the single-delay method degrades quickly with a shorter time series because it is highly sensitive to the choice of τ. However, by considering a range of τ with delay embedding, this sensitivity can be reduced, thereby achieving stable performance.

Figure 4.5: Average classification accuracies of the delay-variant (circles) and single-delay (squares) methods over 100 random splits of the Hes1-model data at different (a)(c)(e)(g) embedding dimensions m and different (b)(d)(f)(h) effective embedding dimensions M. The time-series data were obtained at different measurement intervals (a)(b) $\nu = 8$, (c)(d) $\nu = 16$, (e)(f) $\nu = 32$, and (g)(h) $\nu = 64$. The insets in (a) ($\nu = 8$) and (c) ($\nu = 16$) highlight the accuracy over the embedding dimension m=6–10.

Figure 4.6: Average classification accuracies of the delay-variant (circles) and single-delay (squares) methods over 100 random splits of the Hes1-model data at different embedding dimensions m. The time-series data were obtained by adding the Gaussian white noise with variance (a) $\sigma_n^2 = 0.1$, (b) $\sigma_n^2 = 0.3$, (c) $\sigma_n^2 = 0.5$, and (d) $\sigma_n^2 = 0.7$ to the time series generated with $\nu = 16$ min. The inset in (a) ($\sigma_n^2 = 0.1$) depicts the accuracy over the embedding dimension $m = 6$–10

While performing the comparison over the effective embedding dimension M, the accuracy of the delay-variant method does not change over a wide range of M (40–100), indicating that this method is more reliable than the single-delay method (Fig. 4.5(b), (d), (f), and (h)). For a given effective dimension, the embedded points of single-delay method become highly folded and excessively sparse in a high-dimensional space because the embedding dimension that is used in the single-delay method is considerably higher than that used in the delay-variant method. Furthermore, selecting a considerably high embedding dimension increases the computational complexity of the embedding and causes the high impact of noise acting in a high proportion of elements in delay coordinates. Therefore, selecting a considerably high embedding dimension makes it disadvantageous to use topological features for characterizing the time series, thereby degrading the classification performance.

To demonstrate that the delay-variant method is more robust than the single-delay method when the time series is being perturbed by noise, we compare the two aforementioned methods based on the values of the noise variance σ_n^2 of 0.1, 0.3, 0.5, and 0.7. Figure 4.6 depicts the average classification accuracy versus the embedding dimension m over 100 random splits at a measurement interval of $\nu = 16$ min when the noise variance $\sigma_n^2 = 0.1$ (Fig. 4.6(a)), $\sigma_n^2 = 0.3$ (Fig. 4.6(b)), $\sigma_n^2 = 0.5$ (Fig. 4.6(c)), and $\sigma_n^2 = 0.7$ (Fig. 4.6(d)). As σ_n^2 is increased, the classification becomes more difficult because the time-series data become noisier. Even so, the delay-variant method still

outperforms the single-delay method over m. Furthermore, the single-delay method degrades more quickly as σ_n^2 is increased, thereby indicating that our delay-variant method is more robust and stable in noisy conditions than the single-delay method.

4.4.2 Classify real-world time-series data

Further, we evaluate the performance of the delay-variant method in classifying different heartbeat-signal patterns based on six real electrocardiogram (ECG) datasets in [76], namely ECG200 (separate normal and myocardial infarction heartbeats; 200 time series of length 96), ECG5000 (separate five levels of congestive heart failure; 5,000 time series of length 140), ECGFiveDays (separate records from two different days for the same patient; 884 time series of length 136), TwoLeadECG (separate records from two different leads; 1,162 time series of length 82), and Non-InvasiveFetalECGThorax1 and Non-InvasiveFetalECGThorax2 (separate records from the left and right thorax with expert labeling of 42 classes of non-invasive fetal ECG; 3,765 time series of length 750 in each dataset). We use the delay-variant method to classify the Caenorhabditis elegans roundworms from EigenWorms dataset as either wild or mutant based on their movements [77, 78]. The movement trajectories are processed as 259 time series of length 900. Finally, we denote that our method can be used to diagnose whether a certain symptom can be observed in an automotive subsystem using the time-series data associated with the engine noise with the FordB dataset for 4,446 time series of length 500 [76]. Here, the training data in the FordB dataset were collected under typical operating conditions, whereas the test data were collected under noisy conditions. For these datasets, we employ the train-test split provided in [76]. For the Non-InvasiveFetalECGThorax1, Non-InvasiveFetalECGThorax2, EigenWorms, and FordB datasets, we downsampled the time series with sampling rates as 2, 2, 3, and 2, respectively, before computing the persistence diagrams.

In the delay-variant method, we use $K = 10$ time-delay values of $\tau_1 = 1, \tau_2 = 2, \ldots, \tau_{10} = 10$ with embedding dimensions of $m = 2, 3, \ldots, 10$. In the single-delay method, we choose single τ for each value from $\{\tau_1, \ldots, \tau_{10}\}$. For each embedding dimension of the single-delay method, we use the maximum accuracy in the test dataset over all values of τ to compare with the delay-variant method. In both the methods, we use a linear combination of normalized kernels for the zero-dimensional and one-dimensional holes. The combination weights and the embedding dimension m are selected by cross-validation. Further, we compare the delay-variant and single-delay methods along with some alternative standard approaches. Most of the previous research on time-series classification focused on finding appropriate similarity measures

Table 4.1: Classification accuracies (%) for the ECG200, ECG5000, Non-InvasiveFetalECGThorax1 (Thorax1), Non-InvasiveFetalECGThorax2 (Thorax2), ECGFiveDays (FiveDays), TwoLeadECG (TwoLead), EigenWorms (Worms), and FordB datasets. For each dataset, the best and second-best scores are colored in dark pink and light pink, respectively. The notations are 1st-nearest neighbor classifier (NN), Euclidean distance (E), dynamic time warping distance (D), autocorrelation function (AC), power spectrum (PS), elastic ensemble (EE), and learned-shapelets method (LS).

Data	**Delay variant**	Single delay	NN (E)	NN (D)	NN (AC)	NN (PS)	EE	LS
ECG200	90.0	87.0	88.0	88.0	82.0	86.0	88.0	88.0
ECG5000	93.6	92.1	92.5	92.5	91.0	93.6	93.9	93.2
Thorax1	91.8	78.2	82.9	82.9	72.1	87.5	84.6	25.9
Thorax2	93.0	83.6	88.0	87.0	75.2	88.4	91.4	77.0
FiveDays	99.9	92.0	79.7	79.7	98.1	100.0	82.0	100.0
TwoLead	99.4	94.0	74.7	86.8	80.4	96.1	97.1	99.6
Worms	83.1	83.1	61.0	58.4	76.6	81.8	68.8	72.7
FordB	90.8	78.2	60.6	59.9	78.0	79.0	66.2	91.7

for the 1st-nearest neighbor (NN) classifier. For ensuring similarity in the time domain, we consider either the Euclidean (E) or dynamic time warping (D) distance. For ensuring similarity in the frequency domain and for similarity in the autocorrelation, we consider the power spectrum (PS) and the autocorrelation function (AC). We also examine the elastic ensemble (EE) [79] as a combination of nearest neighbor classifiers using multiple distance measures in the time domain. Finally, we perform the learned-shapelets (LS) method, which classifies time series by learning the representative *shapelets* (i.e., short discriminant time-series subsequences) [80]. Instead of implementing these standard algorithms, we use the results from Ref. [81, 82]. The test results are presented in Table 4.1, wherein the best and second-best accuracy scores of each dataset are colored in dark pink and light pink, respectively. The delay-variant method shows better results than the single-delay method for all the datasets and outperforms all the other algorithms on an average. Further, the delay-variant method offers the best results for four of the eight datasets and the second-best results for the remaining four, suggesting that our method is an effective mechanism to classify the time series.

4.4.3 Multiple-time-scale patterns captured by delay-variant embedding

Subsequently, we investigate the types of features that can be effectively captured using the delay-variant method. Because the time scale of the extracted pattern

Figure 4.7: (a) Time series generated from the frequency-modulated model. (b) Three-dimensional persistence diagram of the loop patterns calculated with $m = 6$ and 20 values of τ given by $\tau_1 = 1, \tau_2 = 2, \ldots, \tau_{20} = 20$. The multiple-time-scale components in the time series corresponding to the limit-cycle oscillations of the dynamics are revealed from the points in the diagram over different values of τ.

is partially dependent on the time delay, the delay-variant method should be able to extract the patterns that comprise multiple different time scales. To better understand this ability, we analyze the synthetic noisy time series obtained from a frequency-modulated model. Given the original signal $s_m(t) = A_m \sin(2\pi f_m t)$ with a carrier signal $s_c(t) = A_c \sin(2\pi f_c t)$, we consider the modulated signal to be $s(t) = A_c \sin(2\pi f_c t + A_m \sin(2\pi f_m t))$, where $t \in [0, 0.1]$, $A_m = 10$, $A_c = 1.0$, and $f_m = 25$. Further, we consider the frequency f_c to be $f_c = 5f$ with $f = 1$–20 to represent patterns with multiple time scales in the modulated signal. For each value of f_c, we generate 20 noisy discrete time series $z_l(n) = s(0.0002n) + w_l(n)$, where $l = 1, 2, \ldots, 20$, $n = 0, 1, \ldots, 500$, and $w_l(n)$ denote the Gaussian white noise with a variance of 0.1.

In the delay-variant method, we use $K = 20$ time-delay values of $\tau_1 = 1, \tau_2 = 2, \ldots, \tau_{20} = 20$. Figure 4.7 depicts a time series with $f_c = 25$ and its three-dimensional persistence diagram for the loops or cycles at an embedding dimension of $m = 6$. In the persistence diagram, the points near the diagonal plane $\mathcal{W}^{(3)} = \{(b, b, \tau) \mid b, \tau \in \mathbb{R}\}$ represent the noise in the time series, whereas the points that are away from $\mathcal{W}^{(3)}$ represent significant cycles in the embedded space. Therefore, the points in the diagram that has been plotted for different values of τ reveal the multiple-time-scale components in the time series corresponding to the limit-cycle oscillations of the dynamics.

An example of the principal component projection from the kernel space is depicted in Fig. 4.8 for (a) the delay-variant method with embedding dimension $m = 6$, (b) the single-delay method with $m = 6$ and $\tau = 1, 2, \ldots, 20$, and (c) the single-delay method with $m = 120$ (to compare with the delay-variant method at the same effective

embedding dimension $M = 120$). Because the length of the time series is 501, in the single-delay approach, τ can only take values from $\{1, 2, 3, 4\}$ when $m = 120$. Different colors represent the data generated using different values of f_c. Figure 4.8 depicts that the delay-variant method results in more distinct regions corresponding to different values of f_c than that obtained using the single-delay method. Note that the points obtained using the single-delay method cannot be distinct, regardless of the angle. These results denote that the delay-variant method is superior to the single-delay method with respect to the ability to identify changes in the multiple-time-scale patterns in time series.

Figure 4.8: Projection of the features of synthetic frequency-modulated time series from kernel space to principal components for (a) the delay-variant method with embedding dimension $m = 6$, (b) the single-delay method with $m = 6$ and $\tau = 1, 2, \ldots, 20$, and (c) the single-delay method with $m = 120$ (for performing comparison with the delay-variant method at the same effective embedding dimension $M = 120$) and $\tau = 1, 2, 3, 4$. Different colors represent the data generated using different values of the carrier frequency f_c.

4.5 Concluding remarks and discussion

We have demonstrated that the topological features that are constructed using delay-variant embedding can capture the topological variation in a time series when the time-delay value changes. Therefore, these features can be used to discriminate between different time series. Theoretically, we have mathematically denoted that these features are robust when the time series is being perturbed by noise. Our method outperformed the standard time-series analysis techniques while classifying both synthetic and real time-series data. These results indicate that the topological features that are deduced with delay-variant embedding can be used to reveal the representative features of the original time series.

In general, the mathematical model of a time series denotes a stochastic process. It is important to understand the factor that distinguishes this model from other models with respect to the dynamics. Recent powerful deep learning methods have focused on accurately classifying time series with predefined labels without identifying the essential behavior of the model. Unfortunately, these approaches could fail when the input is slightly but deliberately perturbed, resulting in misclassification by a neural network. Such perturbed examples can be referred to as *adversarial examples* and have gained considerable attention recently in the discussion about the robustness of deep learning [5]. Machine learning tools, such as deep learning methods, find it difficult to understand why adversarial examples are misclassified because of the black-box nature of the causality between the input data and their underlying model. From another viewpoint, topological data analysis provides a systematic methodology for understanding the true behavior of the data, which will eventually become significant in characterizing the behavior of the model.

We expect that our study will lead to a unified analysis mechanism of the time-series data. This study paves several opportunities for the application of topological tools to produce effective algorithms for analyzing the time-series data. For instance, given a time series, we could use delay-variant embedding to predict its future values or to detect anomalous values or outliers therein.

Chapter 5

Variant Topological Features in Complex Networks Analysis

The structure of real-world networks is usually difficult to characterize owing to the variation of topological scales, the nondyadic complex interactions, and the fluctuations in the network. We aim to address these problems by introducing a general framework using a method based on topological data analysis. By considering the diffusion process at a single specified timescale in a network, we map the network nodes to a finite set of points that contains the topological information of the network at a single scale. Subsequently, we study the shape of these point sets over variable timescales that provide scale-variant topological information, to understand the varying topological scales and the complex interactions in the network. We conduct experiments on synthetic and real-world data to demonstrate the effectiveness of the proposed framework in identifying network models, classifying real-world networks, and detecting transition points in time-evolving networks. Overall, our study presents a unified analysis that can be applied to more complex network structures, as in the case of multilayer and multiplex networks.

5.1 Variant topological scales in complex networks

Characterizing the structure of complex networks is the most fundamental challenge in deciphering network dynamics. The anatomy of a network is quite relevant to phenomena occurring in networks, such as the spread of information, epidemic disease, or robustness under attack. Moreover, it has attracted considerable research interest given the numerous applications including controlling and predicting patterns of dynamics in networks [18, 83, 84], evaluating the structural and functional similarities of biological networks [85–87], and detecting transition points in time-evolving networks [88–90]. In a technical sense, the structure of real-world networks is inherently difficult to characterize, firstly, because these networks have complex

Figure 5.1: Various topological scales and interactions between multiple elements in a complex network. (a) Complex networks can be analyzed at various topological scales ranging from individual nodes (microscale) to the whole network (macroscale). In between the two scales, there is a mesoscale, where we can observe patterns of collectives, cores, and peripheries. (b) Complex network is a representation of a complex system having dyadic and nondyadic interactions between its elements. The interactions can be represented as simplices such as segments (for dyadic interactions), filled triangles, or filled tetrahedrons (for nondyadic interactions involving three or four elements), and so on.

patterns that can reflect various topological scales ranging from microscale (individual nodes) to mesocale (community, cores, and peripheries), to macroscale (the whole network) [91–93] [Fig. 5.1(a)]. For demonstrating these patterns, the conventional statistical measures [94, 95] and methods [91, 96–99] are limited when representing the varying topological scales. Secondly, real-world networks represent complex systems that have dyadic and nondyadic interactions [40, 41, 44] [Fig. 5.1(b)]. The majority of the current methods used for characterizing complex networks focus only on the dyadic interactions, such as detecting the existence of pairwise edges or paths connected by successive edges. Thirdly, real-world networks often suffer from fluctuations caused by external factors [100]. Consequently, the quest for unifying the principles underlying the topology of networks emerges only in simple, idealized models [101, 102].

Herein, we propose a general framework for characterizing the structure of complex

networks, mainly based on the topological data analysis of a diffusion process viewed at variable timescales. We consider a diffusion process in which a random walker moves randomly between nodes in continuous time at the transition rate proportional to the edge weights. The interaction between the nodes via the diffusion process can reflect the structure of the network at different topological scales. For example, a microscale structure is revealed with a small diffusion timescale τ. Increasing τ will increase the ranges of interactions to reflect the mesoscale decomposition of the network, until the macroscale structure is finally captured. By considering the diffusion process at a single specified timescale τ, we can map the network nodes to a finite set of points known as a *point cloud* in a high dimensional space. In the point cloud, a group of close points represents the unit of interacted nodes in the diffusion process. The shape of this point cloud contains the topological information of the network at a single topological scale.

Based on a topological data analysis method that provides insight into the shape of data [6], we build a geometrical model that is primarily a collection of geometrical shapes to reveal the underlying structure of the point cloud. In this geometrical model, two points in the point cloud are connected if their distance is less than or equal to a given threshold. If the threshold is considerably small, only points appear in the geometrical model, and no connections are created between points. As the threshold is gradually increased, more pairwise connections are created, and geometrical shapes as line segments, triangles, tetrahedrons, and so on, are added to the geometrical model. In the case where the threshold becomes considerably large, all pairs of points in the point cloud will be connected, and only a giant overlapped geometrical shape remains in the space. To obtain information regarding the "shape" of the point cloud, we focus on the changes of topological structures, such as the merging of connected components, and the emergence and disappearance of loops in the geometrical model as the threshold is increased. Therefore, at each timescale τ, we construct the topological features to monitor the emergence and disappearance of the topological structures. We can consider such features as a representation for the network at a single topological scale (τ-scale). Further, we extend these features by considering the timescale τ as a variable parameter instead of a single fixed value. The extended features, referred to as scale-variant topological features, can reflect the varying topological scales in the complex network.

The scale-variant topological features are proven to be robust under perturbation applied to the network, and thus, can serve as discriminative features for characterizing the networks. We input these features in the kernel technique in machine learning algorithms to apply to statistical-learning tasks, such as classification and transition points detection. We show that the proposed framework can characterize the parameters that are used to generate the networks through an analysis of several

network models. Furthermore, we can classify both synthetic and real-world networks with more effective results when compared with other conventional approaches. We further apply the proposed framework to detect the transition points with respect to the topological structure in the time-evolving gene regulatory networks of *Drosophila melanogaster*. Interestingly, these transition points agree well with the transition points relative to the dynamics obtained from the experimental results on the profiling.

5.2 Construction of Vietoris–Rips filtration of a network

We define and describe in Fig. 5.2 the process of extracting topological features of a complex network at each specific timescale τ. At each τ, we calculate the diffusion distance matrix $\mathbf{\Delta}_\tau$ of size $N \times N$, whose element Δ_{ij} is the Euclidean distance between points $\boldsymbol{p}_{\mathcal{G}}(\tau|i)$ and $\boldsymbol{p}_{\mathcal{G}}(\tau|j)$ [Fig. 5.2(a)]. If $\varepsilon = 0$, the nodes of the network can be considered discrete points. As we increase ε, new pairwise connections and simplices may appear when ε meets each value of Δ_{ij}. We obtain a filtration as a sequence of embedded simplicial complexes. Hole patterns such as connected components (zero-dimensional holes) or loops (one-dimensional holes) can appear or disappear over this filtration. For instance, in Fig. 5.2(b), at $\varepsilon = 0$, we have six separated nodes considered as six separated connected components, but at $\varepsilon = 0.407$, three nodes are connected with each other; thus, two connected components disappear at this scale. We can describe these patterns as two blue bars started at scale 0 and ended at scale 0.407. The same explanation with the red bar started at scale 0.428 and ended at scale 0.430, which represents the emergence of loop pattern $(v_1 \to v_2 \to v_3 \to v_5 \to v_1)$ at $\varepsilon = 0.428$ and the disappearance at $\varepsilon = 0.430$. Figure 5.2(c) illustrates the corresponding persistence diagrams for zero-dimensional holes and one-dimensional holes, where the birth-scale and the death-scale are represented for the values of ε at the emergence and the disappearance of the holes.

5.3 Variant topological features from scale-variant diffusion embedding

Let $\mathcal{G}$ be an undirected weighted network with N nodes, $v_1, \ldots, v_N$, and assume that there is a single random walker moving randomly between the nodes in continuous time. When the walker is located at v_i, we assume the walker to move to the neighboring node v_j at a transition rate w_{ij}/W_i, where $w_{ij} \geq 0$ represents the weight of the edge from v_i to v_j $(i, j \in \{1, 2, \ldots, N\})$ and $W_i = \sum_{j=1}^{N} w_{ij}$. Herein, if there is no edge

	$\boldsymbol{p}_\mathcal{G}(\tau\|1)$	$\boldsymbol{p}_\mathcal{G}(\tau\|2)$	$\boldsymbol{p}_\mathcal{G}(\tau\|3)$	$\boldsymbol{p}_\mathcal{G}(\tau\|4)$	$\boldsymbol{p}_\mathcal{G}(\tau\|5)$	$\boldsymbol{p}_\mathcal{G}(\tau\|6)$
$\boldsymbol{p}_\mathcal{G}(\tau\|1)$	0.000	0.407	0.430	0.565	0.407	0.565
$\boldsymbol{p}_\mathcal{G}(\tau\|2)$		0.000	0.428	0.445	0.433	0.540
$\boldsymbol{p}_\mathcal{G}(\tau\|3)$			0.000	0.425	0.428	0.425
$\boldsymbol{p}_\mathcal{G}(\tau\|4)$				0.000	0.540	0.410
$\boldsymbol{p}_\mathcal{G}(\tau\|5)$					0.000	0.445
$\boldsymbol{p}_\mathcal{G}(\tau\|6)$						0.000

Figure 5.2: An exemplary of Vietoris–Rips filtration constructed from a complex network $\mathcal{G}$ at a specific timescale τ ($\tau = 1$ in this example). We map nodes $v_1, \ldots, v_N$ of the network $\mathcal{G}$ to a point cloud of N points $\boldsymbol{p}_\mathcal{G}(\tau|1), \ldots, \boldsymbol{p}_\mathcal{G}(\tau|N)$ through a diffusion dynamics described by the random walk Laplacian $\boldsymbol{L}_\mathcal{G}^{\mathrm{rw}}$. (a) Diffusion distance matrix $\boldsymbol{\Delta}_\tau$ of size $N \times N$, whose element Δ_{ij} is the Euclidean distance between points $\boldsymbol{p}_\mathcal{G}(\tau|i)$ and $\boldsymbol{p}_\mathcal{G}(\tau|j)$. (b) A complex is built over a set of points if the pairwise distances between them are less than or equal to a threshold parameter ε. If $\varepsilon = 0$, we have the discrete points. As ε takes the increasing sequence values of diffusion distance Δ_{ij}, the hole patterns such as connected components (zero-dimensional hole) or loops (one-dimensional hole) can appear or disappear over this filtration. The lifetime of these hole patterns are described as blue bars (for zero-dimensional holes) and red bars (for one-dimensional holes). These bars begin at the values of ε when the holes appear, then end at values when the holes disappear. (c) The corresponding persistence diagrams for zero-dimensional and one-dimensional holes. The birth-scale and the death-scale are represented for the values of ε at the emergence and the disappearance of the holes.

between v_i and v_j, then $w_{ij} = 0$. Now, let $p_{\mathcal{G},k}(\tau|i)$ denote the probability of a random walker on v_k at time τ that starts from v_i. The probability distribution vector, $\boldsymbol{p}_{\mathcal{G}}(\tau|i) = [p_{\mathcal{G},1}(\tau|i), \ldots, p_{\mathcal{G},N}(\tau|i)]$, is given based on the solution of the Kolmogorov forward equation [49]:

$$\frac{d\boldsymbol{p}_{\mathcal{G}}(\tau|i)}{d\tau} = -\boldsymbol{p}_{\mathcal{G}}(\tau|i)\boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}}. \tag{5.1}$$

Here, $\boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}}$ is the random walk Laplacian whose components l_{ij} $(i, j \in \{1, 2, \ldots, N\})$ are given by,

$$l_{ij} = \begin{cases} 1 & \text{if } i = j \text{ and } W_i \neq 0 \\ -w_{ij}/W_i & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

The solution for Eq. (5.1) is $\boldsymbol{p}_{\mathcal{G}}(\tau|i) = \boldsymbol{\mu}_i \exp(-\tau\boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}})$, where $\boldsymbol{\mu}_i = [0, \ldots, 0, 1, 0, \ldots, 0]$ with its i-th element being equal to 1; the others are equal to 0 $(i \in \{1, 2, \ldots, N\})$.

At each timescale τ, we consider mapping χ_τ from the set $V_{\mathcal{G}} = \{v_1, v_2, \ldots, v_N\}$ of nodes in $\mathcal{G}$ to the Euclidean space $\mathbb{R}^N$ such that,

$$\begin{aligned} \chi_\tau : \quad V_{\mathcal{G}} &\longrightarrow \mathbb{R}^N \\ v_i &\longmapsto \boldsymbol{p}_{\mathcal{G}}(\tau|i) \quad (i = 1, 2, \ldots, N). \end{aligned} \tag{5.3}$$

The mapped point $\boldsymbol{p}_{\mathcal{G}}(\tau|i)$ of nodes v_i represents the probability on all nodes at time τ of a random walker that starts from v_i. Therefore, $\boldsymbol{p}_{\mathcal{G}}(\tau|i)$ can reflect the interaction between v_i and other nodes at τ-scale, and characterize the structural role of node v_i with multi-resolutions when τ varies. The shape of the point cloud $P_{\mathcal{G}}(\tau) = \{\boldsymbol{p}_{\mathcal{G}}(\tau|1), \ldots, \boldsymbol{p}_{\mathcal{G}}(\tau|N)\}$ provides valuable insights into the dyadic and nondyadic interactions between nodes, and into the structural property of $\mathcal{G}$ at τ-scale. Moreover, the distance between two mapped points in $P_{\mathcal{G}}(\tau)$ is relatively small if there are many paths connecting two original nodes in $\mathcal{G}$. The nodes that belong to the same community or cluster in the network tend to form a group of close points in $P_{\mathcal{G}}(\tau)$.

In Fig. 5.3(a), we consider an undirected network that comprises four clusters with more intra-cluster connections than inter-cluster ones. Pairwise distances of the mapped points of the nodes belonging to the same clusters are smaller than the distances between the nodes belonging to different clusters. These distances decrease as values of τ increase [Fig. 5.3(b)]. In the point cloud, the hole patterns appear with different sizes in different groups of points as τ varies. We obtain the scale-variant topological features that reflect the variation of topological scales by considering the two-dimensional persistence diagrams with the varying τ. Consider τ in a set $\mathcal{T}$ =

Figure 5.3: (a) An undirected network comprising four clusters with more connections within intra-clusters than between inter-clusters. (b) For each τ, the nodes are mapped onto a point cloud such that the distances of the mapped points of the nodes in the same clusters are smaller than those between the nodes belonging to different clusters. These distances decrease as τ increases with $\tau_1 < \tau_2 < \tau_3$. (c) The topological features at each τ characterize the shape of the point cloud. These features are displayed as a two-dimensional persistence diagram at each τ. (d) The scale-variant topological features, i.e., the three-dimensional persistence diagram, are obtained by integrating two-dimensional diagrams at varying τ. The birth-scale and death-scale axes of the diagrams are represented at the logarithmic scale.

$\{\tau_1, \tau_2, ..., \tau_K\}$, where $0 < \tau_1 < \tau_2 < \cdots < \tau_K$ are predefined or sampled values from the continuous domain of timescales. The scale-variant topological features, i.e., the three-dimensional persistence diagram of l-dimensional holes for network $\mathcal{G}$, are defined by $D^{(3)}_{(l)}(\mathcal{G}) = \{(b, d, \tau) \mid (b, d) \in D^{(2)}_{(l),\tau}(\mathcal{G}), \tau \in \mathcal{T}\}$ [Fig. 5.3(d)].

5.4 Stability with respect to the perturbations of network structure

We show that the scale-variant topological features are robust with respect to some perturbations of the network. To describe this robustness, we use the bottleneck distance, $d^{(3)}_{\mathrm{B},\xi}$, a metric structure introduced in Chapter 2 for comparing three-dimensional persistence diagrams. Herein, ξ is a positive rescaling coefficient introduced to adjust the scale difference between the pointwise distance and time. We consider two undirected networks $\mathcal{G}$ and $\mathcal{H}$ with the same number of nodes. Based on Refs. [68, 103], we can prove that the upper limit of the bottleneck distance between $D^{(3)}_{(l)}(\mathcal{G})$ and

$D^{(3)}_{(l)}(\mathcal{H})$ is governed by the matrix 2-norm of the difference between $\boldsymbol{L}^{\mathrm{rw}}_{\mathcal{G}}$ and $\boldsymbol{L}^{\mathrm{rw}}_{\mathcal{H}}$:

$$d^{(3)}_{\mathrm{B},\xi}(D^{(3)}_{(l)}(\mathcal{G}), D^{(3)}_{(l)}(\mathcal{H})) \leq 2\tau_K \|\boldsymbol{L}^{\mathrm{rw}}_{\mathcal{G}} - \boldsymbol{L}^{\mathrm{rw}}_{\mathcal{H}}\|_2. \tag{5.4}$$

Herein, $\|\boldsymbol{A}\|_2$ denotes the matrix 2-norm of matrix $\boldsymbol{A}$. The inequality of Eq. (5.4) indicates that our scale-variant topological features are robust with respect to the perturbations applied to the random walk Laplacian matrix. Therefore, these features can be used as discriminative features for characterizing networks.

We prove the result in Eq. (5.4). First, for our convenience, let us mention again the concept of the bottleneck distance between two two-dimensional diagrams. Let X and Y be finite sets of points embedded in the Euclidean space $\mathbb{R}^n$. Denote their *two*-dimensional persistence diagrams for l-dimensional holes as $\mathcal{S}^{(2)}_{(l)}(X)$ and $\mathcal{S}^{(2)}_{(l)}(Y)$, respectively. We consider all matchings, γ, such that a point on one diagram is matched to a point on the other diagram or to its projection on the line $b = d$ in two-dimensional space. The bottleneck distance $d^{(2)}_{\mathrm{B}}$ between $\mathcal{S}^{(2)}_{(l)}(X)$ and $\mathcal{S}^{(2)}_{(l)}(Y)$ is defined as the infimum of the longest matched infinity-norm distance over all matchings, γ:

$$d^{(2)}_{\mathrm{B}}(\mathcal{S}^{(2)}_{(l)}(X), \mathcal{S}^{(2)}_{(l)}(Y)) = \inf_{\gamma} \max_{(\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}) \in \gamma} \|\boldsymbol{q}^{(1)} - \boldsymbol{q}^{(2)}\|_\infty. \tag{5.5}$$

Here, $\|\boldsymbol{q}^{(1)} - \boldsymbol{q}^{(2)}\|_\infty = \max(|b_1 - b_2|, |d_1 - d_2|)$ for which $\boldsymbol{q}^{(1)} = (b_1, d_1)$ and $\boldsymbol{q}^{(2)} = (b_2, d_2)$.

The bottleneck distance between the two-dimensional persistence diagrams satisfies the following inequality [68]:

$$d^{(2)}_{\mathrm{B}}(\mathcal{S}^{(2)}_{(l)}(X), \mathcal{S}^{(2)}_{(l)}(Y)) \leq 2d_{\mathrm{H}}(X, Y), \tag{5.6}$$

where $d_{\mathrm{H}}(X, Y)$ is the Hausdorff distance given as

$$d_{\mathrm{H}}(X, Y) = \max\left\{\max_{\boldsymbol{x} \in X} \min_{\boldsymbol{y} \in Y} d(\boldsymbol{x}, \boldsymbol{y}), \max_{\boldsymbol{y} \in Y} \min_{\boldsymbol{x} \in X} d(\boldsymbol{x}, \boldsymbol{y})\right\}. \tag{5.7}$$

Here, $d(\boldsymbol{x}, \boldsymbol{y})$ is the Euclidean distance between two points $\boldsymbol{x}, \boldsymbol{y}$ in $\mathbb{R}^n$.

Given two three-dimensional persistence diagrams as D_1 and D_2, consider all matchings ψ such that a point on one diagram is matched to a point on the other diagram or to its projection on the plane $b = d$. For each pair $(\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}) \in \psi$ for which $\boldsymbol{q}^{(1)} = (b_1, d_1, \tau_1)$ and $\boldsymbol{q}^{(2)} = (b_2, d_2, \tau_2)$, we define *the relative infinity-norm distance* between $\boldsymbol{q}^{(1)}$ and $\boldsymbol{q}^{(2)}$ as $d^{(\infty)}_{\xi}(\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}) = \max(|b_1 - b_2|, |d_1 - d_2|, \xi|\tau_1 - \tau_2|)$, where ξ is a positive rescaling coefficient introduced to adjust the scale difference between the point-wise distance and time. The bottleneck distance, $d^{(3)}_{\mathrm{B},\xi}(D_1, D_2)$, is defined as the

infimum of the longest matched relative infinity-norm distance over all matchings ψ:

$$d_{\mathrm{B},\xi}^{(3)}(D_1, D_2) = \inf_{\psi} \max_{(\boldsymbol{q}^{(1)},\boldsymbol{q}^{(2)})\in\psi} d_{\xi}^{(\infty)}(\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}). \tag{5.8}$$

For each $\tau \in \mathcal{T}$ and two networks $\mathcal{G}, \mathcal{H}$ with the same number N of nodes, we first prove the following inequality:

$$d_{\mathrm{B}}^{(2)}(D_{(l),\tau}^{(2)}(\mathcal{G}), D_{(l),\tau}^{(2)}(\mathcal{H})) \leq 2\tau \|\boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}} - \boldsymbol{L}_{\mathcal{H}}^{\mathrm{rw}}\|_2. \tag{5.9}$$

Here, two two-dimensional persistence diagrams $D_{l,\tau}^{(2)}(\mathcal{G})$ and $D_{l,\tau}^{(2)}(\mathcal{H})$ are calculated for l-dimensional holes from two point clouds $P_{\mathcal{G}}(\tau) = \{\boldsymbol{p}_{\mathcal{G}}(\tau|1), \ldots, \boldsymbol{p}_{\mathcal{G}}(\tau|N)\}$ and $P_{\mathcal{H}}(\tau) = \{\boldsymbol{p}_{\mathcal{H}}(\tau|1), \ldots, \boldsymbol{p}_{\mathcal{H}}(\tau|N)\}$, respectively.

Since $D_{(l),\tau}^{(2)}(\mathcal{G}) = \mathcal{S}_{(l)}(P_{\mathcal{G}}(\tau))$ and $D_{(l),\tau}^{(2)}(\mathcal{H}) = \mathcal{S}_{(l)}(P_{\mathcal{H}}(\tau))$, we apply Eq. (5.6) to have

$$\begin{aligned} d_{\mathrm{B}}^{(2)}(D_{(l),\tau}^{(2)}(\mathcal{G}), D_{(l),\tau}^{(2)}(\mathcal{H})) &= d_{\mathrm{B}}^{(2)}(\mathcal{S}_{(l)}(P_{\mathcal{G}}(\tau)), \mathcal{S}_{(l)}(P_{\mathcal{H}}(\tau))) \\ &\leq 2d_{\mathrm{H}}(P_{\mathcal{G}}(\tau), P_{\mathcal{H}}(\tau)). \end{aligned} \tag{5.10}$$

From the definition of the Hausdorff distance, we have

$$d_{\mathrm{H}}(P_{\mathcal{G}}(\tau), P_{\mathcal{H}}(\tau)) = \max\left\{\max_i \min_j d(\boldsymbol{p}_{\mathcal{G}}(\tau|i), \boldsymbol{p}_{\mathcal{H}}(\tau|j)), \max_j \min_i d(\boldsymbol{p}_{\mathcal{G}}(\tau|i), \boldsymbol{p}_{\mathcal{H}}(\tau|j))\right\} \tag{5.11}$$

$$\leq \max\left\{\max_i d(\boldsymbol{p}_{\mathcal{G}}(\tau|i), \boldsymbol{p}_{\mathcal{H}}(\tau|i)), \max_j d(\boldsymbol{p}_{\mathcal{G}}(\tau|j), \boldsymbol{p}_{\mathcal{H}}(\tau|j))\right\}$$

$$= \max_i d(\boldsymbol{p}_{\mathcal{G}}(\tau|i), \boldsymbol{p}_{\mathcal{H}}(\tau|i)). \tag{5.12}$$

Since $\boldsymbol{p}_{\mathcal{G}}(\tau|i) = \boldsymbol{\mu}_i \exp(-\tau \boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}})$ and $\boldsymbol{p}_{\mathcal{H}}(\tau|i) = \boldsymbol{\mu}_i \exp(-\tau \boldsymbol{L}_{\mathcal{H}}^{\mathrm{rw}})$, we have

$$d(\boldsymbol{p}_{\mathcal{G}}(\tau|i), \boldsymbol{p}_{\mathcal{H}}(\tau|i)) = \|\boldsymbol{\mu}_i(e^{-\tau \boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}}} - e^{-\tau \boldsymbol{L}_{\mathcal{H}}^{\mathrm{rw}}})\|_2 \tag{5.13}$$

$$\leq \|\boldsymbol{\mu}_i\|_2 \|e^{-\tau \boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}}} - e^{-\tau \boldsymbol{L}_{\mathcal{H}}^{\mathrm{rw}}}\|_2 \tag{5.14}$$

$$= \|e^{-\tau \boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}}} - e^{-\tau \boldsymbol{L}_{\mathcal{H}}^{\mathrm{rw}}}\|_2. \tag{5.15}$$

We write the difference of the matrix exponential in terms of an integral [103],

$$\|e^{-\tau \boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}}} - e^{-\tau \boldsymbol{L}_{\mathcal{H}}^{\mathrm{rw}}}\|_2 = \left\| \int_0^{\tau} e^{-\boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}}(\tau-t)} \boldsymbol{E} e^{-\boldsymbol{L}_{\mathcal{H}}^{\mathrm{rw}} t} dt \right\|_2 \tag{5.16}$$

$$\leq \int_0^{\tau} \|e^{-\boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}}(\tau-t)} \boldsymbol{E} e^{-\boldsymbol{L}_{\mathcal{H}}^{\mathrm{rw}} t}\|_2 dt \tag{5.17}$$

$$\leq \|\boldsymbol{E}\|_2 \int_0^{\tau} \|e^{-\boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}}(\tau-t)}\|_2 \|e^{-\boldsymbol{L}_{\mathcal{H}}^{\mathrm{rw}} t}\|_2 dt, \tag{5.18}$$

where $\boldsymbol{E} = \boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}} - \boldsymbol{L}_{\mathcal{H}}^{\mathrm{rw}}$. We know that $-\boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}}$ is a negative semi-definite matrix with the largest eigenvalue equal to 0. It implies that the largest eigenvalue of $e^{-\boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}}(\tau-t)}$ is equal to 1, and hence $\|e^{-\boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}}(\tau-t)}\|_2 = 1$. We obtain the same result with $-\boldsymbol{L}_{\mathcal{H}}^{\mathrm{rw}}$, i.e., $\|e^{-\boldsymbol{L}_{\mathcal{H}}^{\mathrm{rw}} t}\|_2 = 1$. Then, from Eq. (5.15) and Eq. (5.18), we have

$$d(\boldsymbol{p}_{\mathcal{G}}(\tau|i), \boldsymbol{p}_{\mathcal{H}}(\tau|i)) \leq \|\boldsymbol{E}\|_2 \int_0^{\tau} 1 dt \tag{5.19}$$

$$= \tau \|\boldsymbol{E}\|_2 = \tau \|\boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}} - \boldsymbol{L}_{\mathcal{H}}^{\mathrm{rw}}\|_2. \tag{5.20}$$

From Eq. (5.10), Eq. (5.12) and Eq. (5.20), we have the result in Eq. (5.9).

Let Γ_τ be the set of matchings defined in Eq. (5.5) between two two-dimensional persistence diagrams $D_{(l),\tau}^{(2)}(\mathcal{G})$ and $D_{(l),\tau}^{(2)}(\mathcal{H})$. For each collection $\Lambda = \{\gamma_1, \gamma_2, \ldots, \gamma_K \mid \gamma_i \in \Gamma_{\tau_i}, i = 1, 2, \ldots, K\}$, we construct the matching ψ between two three-dimensional persistence diagrams $D_{(l)}^{(3)}(\mathcal{G})$ and $D_{(l)}^{(3)}(\mathcal{H})$, such that, for each $(\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}) \in \psi$, then $\boldsymbol{q}^{(1)} = (b_1, d_1, \tau)$, $\boldsymbol{q}^{(2)} = (b_2, d_2, \tau)$, and $(\boldsymbol{q}_\gamma^{(1)}, \boldsymbol{q}_\gamma^{(2)}) \in \gamma$, where $\boldsymbol{q}_\gamma^{(1)} = (b_1, d_1)$, $\boldsymbol{q}_\gamma^{(2)} = (b_2, d_2)$ and $\gamma \in \Lambda \cap \Gamma_\tau$. Let Γ be a set of all matchings ψ constructed this way. From the definition of the bottleneck distance, we have the following inequality:

$$d_{\mathrm{B},\xi}^{(3)}(D_{(l)}^{(3)}(\mathcal{G}), D_{(l)}^{(3)}(\mathcal{H})) \leq \inf_{\psi \in \Gamma} \max_{(\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}) \in \psi} d_\xi^{(\infty)}(\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}). \tag{5.21}$$

For $(\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}) \in \psi$, we have

$$d_\xi^{(\infty)}(\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}) = \max\{|b_1 - b_2|, |d_1 - d_2|, \xi|\tau - \tau|\} \tag{5.22}$$

$$= \max\{|b_1 - b_2|, |d_1 - d_2|\} \tag{5.23}$$

$$= \|\boldsymbol{q}_\gamma^{(1)} - \boldsymbol{q}_\gamma^{(2)}\|_\infty, \tag{5.24}$$

and Eq. (5.21) becomes

$$d_{\mathrm{B},\xi}^{(3)}(D_{(l)}^{(3)}(\mathcal{G}), D_{(l)}^{(3)}(\mathcal{H})) \leq \max_{\tau \in \mathcal{T}} \inf_{\gamma \in \Gamma_\tau} \max_{(\boldsymbol{q}_\gamma^{(1)}, \boldsymbol{q}_\gamma^{(2)}) \in \gamma} \|\boldsymbol{q}_\gamma^{(1)} - \boldsymbol{q}_\gamma^{(2)}\|_\infty \tag{5.25}$$

$$= \max_{\tau \in \mathcal{T}} d_{\mathrm{B}}^{(2)}(D_{(l),\tau}^{(2)}(\mathcal{G}), D_{(l),\tau}^{(2)}(\mathcal{H})). \tag{5.26}$$

From Eq. (5.9) and Eq. (5.26), we obtain Eq. (5.4) in the main text, which is the stability property of our scale-variant features.

5.5 Applications

5.5.1 Variations of the parameters in network models

We now investigate how the scale-variant topological features can reflect variations of the parameters of network models. We generate networks using Girvan–Newman (GN) [104], Lancichinetti–Fortunato–Radicchi (LFR) [105, 106], Watts–Strogatz (WS) [107], Erdős–Rényi (ER) [108], Lancichinetti–Fortunato–Radicchi with hierarchical structure (LFR–H) [98], and Sales–Pardo (SP) [97] models. We focus on the model parameters that represent the topological scale of these networks, such as the ratio r between the probability of inter- (p_{out}) and intra-community links (p_{in}) (GN), mixing rate μ (LFR), rewiring probability β (WS), pair-link probability p_{link} (ER), mixing rate μ_{macro} for macrocommunities (LFR–H), and ρ, which estimates the separations between topological scales in the SP model. The model parameters are varied as $r = p_{\mathrm{out}}/p_{\mathrm{in}} = 0.01, 0.02, \ldots, 1.0$; $\mu = 0.01, 0.02, \ldots, 1.0$; $\beta = 0.00, 0.01, \ldots, 1.0$; $p_{\mathrm{link}} = 0.020, 0.021, \ldots, 0.1$; $\mu_{\mathrm{macro}} = 0.01, 0.02, \ldots, 0.2$ and; $\rho = 0.05, 0.10, \ldots, 2.0$. We generate 10 network realizations for each of the models GN, LFR, WS, ER, and SP, and 20 network realizations for the LFR–H model at each value of the corresponding model parameter. There are 128 nodes in the GN, LFR, WS, and ER networks, 300 nodes in each LFR–H network, and 640 nodes in each SP network.

We compute three-dimensional persistence diagrams for one-dimensional holes with $\tau_1 = 1, \tau_2 = 2, \ldots, \tau_{100} = 100$, and then calculate the kernel defined in Eq. (3.3) for the collection of generated networks in each model. Figure 5.4 shows the principal components projections from the kernel-mapped feature space of each model, at which the points with different colors represent the networks generated from different values of the model parameters. In WS, ER, LFR–H, and SP models, the scale-variant topological features reflect a variation of the parameters associated with the topological scales mainly that the points located at different positions have different colors [Figs. 5.4(c)–(f)]. In GN and LFR models, there are variations in the topological scales of the network as r and μ vary from 0 (four separate groups) to 1 (a purely random graph). Using the kernel Fisher discriminant ratio calculated for the series of persistence diagrams, we obtain the transition with respect to the topological structure at $r = 0.12$ and $\mu = 0.26$ for the series of networks obtained at increasing r and μ (Fig. 5.5). These values correspond to the boundaries between the identifiable phases, where parameters can be identified from the kernel-mapped feature space and the non-identifiable phases [Figs. 5.4(a) and 5.4(b)].

Figure 5.4: Principal components projection from the kernel-mapped feature space of the scale-variant topological features in each network model. Points with different colors represent networks generated from different values of the model parameters. Networks are generated from (a) Girvan–Newman (GN), (b) Lancichinetti–Fortunato–Radicchi (LFR), (c) Watts–Strogatz (WS), (d) Erdős–Rényi (ER), (e) Lancichinetti–Fortunato–Radicchi hiearchical (LFR–H), and (f) Sales–Pardo (SP) models. Parameters for these models vary as follows: $r = p_{\text{out}}/p_{\text{in}} = 0.01, 0.02, \ldots, 1.0$ (GN); $\mu = 0.01, 0.02, \ldots, 1.0$ (LFR); $\beta = 0.00, 0.01, \ldots, 1.0$ (WS); $p_{\text{link}} = 0.020, 0.021, \ldots, 0.1$ (ER); $\mu_{\text{macro}} = 0.01, 0.02, \ldots, 0.2$ (LFR–H) and; $\rho = 0.05, 0.10, \ldots, 2.0$ (SP).

Figure 5.5: Kernel Fisher discriminant ratio κ estimated for the series of (a) Girvan–Newman networks and (b) Lancichinetti–Fortunato–Radicchi networks. The transition point is detected with respect to the topological structure of networks from the series of persistence diagrams for one-dimensional holes obtained when $r = p_{\text{out}}/p_{\text{in}}$ is increased as $r_1 = 0.01, r_2 = 0.02, \ldots, r_{100} = 1.0$ (for Girvan–Newman networks), and μ is increased as $\mu_1 = 0.01, \mu_2 = 0.02, \ldots, \mu_{100} = 1.0$ (for Lancichinetti–Fortunato–Radicchi networks). The maximum value of κ is marked with the orange point of the dashed line. The transition point is the value of the parameter that achieves the maximum value of κ. The transition points are obtained as $r_c = 0.12$ (for Girvan–Newman networks) and $\mu_c = 0.26$ (for Lancichinetti–Fortunato–Radicchi networks).

Figure 5.6: Networks from Girvan–Newman, Lancichinetti–Fortunato–Radicchi, and Watts–Strogatz models are generated with labels denoted by GN-org, LFR-org, and WS-org, respectively; their corresponding configuration networks labels are denoted by GN-conf, LFR-conf, and WS-conf. (a) The kernel principal components projection of the scale-variant topological features for these networks. (b)(c) Variation of high-order features, i.e., degree assortativity coefficient, maximum modularity, and average clustering coefficient for (b) all generated networks, and (c) configuration networks. The different colors represent the networks generated from different models.

5.5.2 Identification of network models

Here we show that the scale-variant topological features can classify the networks generated from different models, even if they have similar global statistical measures. We study the configuration model in Ref. [109], which generates random networks (known as *configuration networks*) having the same sequences of node degrees as a given network. The labels of the networks generated from GN, LFR, and WS models are denoted by GN-org, LFR-org, and WS-org, respectively, while their corresponding configuration networks labels are denoted by GN-conf, LFR-conf, and WS-conf. We compute the three-dimensional persistence diagrams for one-dimensional holes of these networks with timescale values $\tau_1 = 1, \tau_2 = 2, \ldots, \tau_{100} = 100$. Accordingly, we calculate the kernel for these diagrams, then perform three-dimensional projections of the principal components from the kernel-mapped feature space [Fig. 5.6(a)]. Here, points with different colors represent networks generated from different models. In Fig. 5.6(a), the points appear to be distinguishable by their colors, thus, we can conclude that the scale-variant topological features can characterize the differences with respect to the topological structure between networks, and even between configuration networks generated from different models.

While the node degree distribution in a configuration network is the same as the given network, the topological correlations between the nodes are destroyed. Therefore, we investigate conventional higher-order features of the network, such as the degree assortativity coefficient, the average clustering coefficient, and the maximum

modularity obtained via Louvain heuristic [104, 110]. Figure 5.6(b) highlights the variation of these features in our generated networks. Specifically in Fig. 5.6(b), the points with corresponding labels GN-org, LFR-org, and WS-org appear to be distinguishable from others; thus it becomes easy to distinguish between networks generated from different models and between a given network with its corresponding configuration network. However, if we look at the variation of these features for configuration networks [Fig. 5.6(c)], we note that the conventional higher-order features of the network cannot capture the apparent differences in topological structure between the configuration networks, even when their corresponding original networks are generated from different mechanics models. In contrast with this observation and as highlighted in Fig. 5.6(a), the scale-variant topological features can provide a better representation of the topological structure of networks.

Accordingly, we quantify to what extent the scale-variant topological features identify the networks generated from different models. We employ the scale-variant method, which uses the scale-variant topological features to classify networks into six labels, namely, GN-org, LFR-org, WS-org, GN-conf, LFR-conf, and WS-conf. We randomly split 10 networks generated at each value of the model parameters into two, i.e., five networks for training and five for testing, and apply the support vector machine [111] for classification in the kernel-mapped feature space. Figure 5.7(a) depicts the average normalized confusion matrix over 100 random splits, where the row and column labels are the predicted and true labels, respectively. Figure 5.7(a) shows a reasonably high accuracy for identifying the networks generated from different models with the following labels: GN-org (99.2%), LFR-org (99.2%), WS-org (99.4%), GN-conf (94.8%), LFR-conf (99.4%), and WS-conf (96.6%). This result demonstrates that the scale-variant topological features can reflect well on the behaviors of these network models.

To highlight the benefits of the scale-variant method, we compare it with the other conventional methods using common network measures [94, 112–115], well-recognized graph kernels [116], and topological features calculated at an average fixed topological scale. We use the common network measures in Appendix A as well as the graph kernels that are based on random walks (KStepRW, GeometricRW, ExponentialRW) [117, 118], paths (ShortestPath) [119], limited-sized subgraphs (Graphlet) [120], and subtree patterns (Weisfeiler–Lehman [121]) in Appendix A. Moreover, we consider two variations of topological features evaluated at an average fixed topological scale to show the advantages of using variable timescales. Also, instead of using a particular timescale, we use the scale-average and the scale-norm-average methods to preserve the geometrical persistence of the point cloud. The former uses the topological features extracted from the average distance matrix $\mathbf{\Delta}_{\text{avg}} = (1/K)\sum_{i=1}^{K}\mathbf{\Delta}_{\tau_i}$, whereas the latter uses the features from the average

Figure 5.7: Classification of networks generated from Girvan–Newman, Lancichinetti–Fortunato–Radicchi, and Watts–Strogatz models, with labels denoted by GN-org, LFR-org, and WS-org, respectively; their corresponding configuration networks labels are denoted by GN-conf, LFR-conf, and WS-conf. (a) Average normalized confusion matrix of the scale-variant method over 100 random train-test splits of the data. The 10 networks generated at each value of the model parameters are split into two, with five networks for training and the other five for testing. (b) Average accuracies (%) of the classification methods over 100 random train-test splits at each proportion of the training data (bold lines). The shaded areas indicate the confidence intervals of one standard deviation calculated using the same ensemble of runs.

normalized distance matrix $\tilde{\mathbf{\Delta}}_{\text{avg}} = (1/K)\sum_{i=1}^{K}\tilde{\mathbf{\Delta}}_{\tau_i}$ [49]. Herein, $\mathbf{\Delta}_{\tau_i}$ denotes the distance matrix of pairwise Euclidean distances between points in $P_{\mathcal{G}}(\tau_i)$, whereas $\tilde{\mathbf{\Delta}}_{\tau_i}$ is obtained by dividing $\mathbf{\Delta}_{\tau_i}$ by its maximum element. We randomly split the 10 networks generated at each value of the model parameters into proportions for training and for testing, and employ the support vector machine as the classifier to both the common network measures and the kernel-mapped feature space. We compute the average classification accuracy over 100 random splits at different proportions of the training data. Figure 5.7(b) depicts the performance of the methods with accuracies greater than 70%, mainly illustrating that the scale-variant method outperforms the other methods in terms of classification accuracy. Moreover, the scale-variant method is shown to achieve approximately 97% of accuracy, even with a small size of the training dataset, e.g., only 10% of all the data, whereas the other methods yielded accuracies of at most 84%. These results validate the effectiveness and the reliability of our scale-variant method in capturing the differences between network structures. The source code used in our experiments is publicly available on GitHub [122].

5.5.3 Classification of real-world network data

Next, we apply the scale-variant topological features to the classification of chemoinformatics network datasets (MUTAG, BZR, COX2, DHFR, FRANKENSTEIN, NCI1, NCI109), bioinformatics dataset (PROTEIN), and large real-world social network datasets, such as movie collaboration networks (IMDB–BINARY, IMDB–MULTI), scientific collaboration networks (COLLAB), and networks obtained from online discussion threads on Reddit (REDDIT–BINARY, REDDIT–MULTI–5K) [123–130]. The aggregate statistics for these datasets is provided in Table 5.1. We compute three-dimensional persistence diagrams with $\tau_1 = 1, \ldots, \tau_{50} = 50$, and use the multiple kernel learning method [131] to learn the linear combination of the normalized kernels for zero-dimensional and one-dimensional holes. Subsequently, we compare our scale-variant method with methods employing the common network measures and the scale-average and scale-norm-average methods. Likewise, we compare the scale-variant method with many state-of-the-art algorithms in classifying graphs and networks as follows: (i) random walk kernels based on matching walks in two graphs (KStepRW, GeometricRW, ExponentialRW) [117, 118], (ii) the shortest path kernel (ShortestPath) [119], (iii) the graphlet count kernel (Graphlet) [120], (iv) the Weisfeiler–Lehman subtree kernel (Weisfeiler–Lehman) [121], (v) the deep graph kernel (DGK) [129], (vi) the PATCHY-SAN convolutional neural network (PSCN) [132], and (vii) the graph kernel based on return probabilities of random walks (RetGK) [133]. Here, in order to make a fair comparison with these methods, as presented in the literature [133], we apply the support vector machine [111] as the classifier in the

Table 5.1: Summary statistics of the real-world network datasets.

Dataset	Type of networks	Num. of networks	Num. of classes	Num. of networks in each class	Avg. num. of nodes	Avg. num. of edges
MUTAG	Chemoinformatics	188	2	(63,125)	17.93	19.79
BZR	Chemoinformatics	405	2	(319, 86)	35.75	38.36
COX2	Chemoinformatics	467	2	(365, 102)	41.22	43.45
DHFR	Chemoinformatics	756	2	(295, 461)	42.43	44.54
FRANKENSTEIN	Chemoinformatics	4337	2	(2401, 1936)	16.90	17.88
NCI1	Chemoinformatics	4110	2	(2053, 2057)	29.87	32.30
NCI109	Chemoinformatics	4127	2	(2048, 2079)	29.68	32.13
PROTEINS	Bioinformatics	1113	2	(663, 450)	39.06	72.82
IMDB–BINARY	Social	1000	2	(500, 500)	19.77	96.53
IMDB–MULTI	Social	1500	3	(500, 500, 500)	13.00	65.94
COLLAB	Social	5000	3	(2600, 775, 1625)	74.49	2457.78
REDDIT–BINARY	Social	2000	2	(1000, 1000)	429.63	497.75
REDDIT–MULTI –5K	Social	4999	5	(1000, 1000, 1000, 1000, 999)	508.52	594.87

kernel-mapped feature space. Moreover, we perform 10-fold cross-validations, where a single 10-fold is created by randomly shuffling the dataset, and then splitting it into 10 different parts (folds) of equal size. In every single 10-fold, we use nine folds for training and one for testing and averaging of the classification accuracy of the test set obtained throughout the folds. To reduce the variance of the accuracy due to the splitting of data, we repeat the whole process of cross-validation for 10 times, and then report the average and standard deviation of the classification accuracies.

The social network datasets contain networks that do not have information, such as labels and attributes of nodes. For movie collaboration datasets (IMDB–BINARY, IMDB–MULTI), collaboration ego-networks are generated for each actor (actress). In each network, two nodes representing the actors or actresses are connected when they appear in the same movie. The task is to identify whether a given ego-network of an actor (actress) belongs to one of the predefined movie genres. For scientific collaboration dataset (COLLAB), collaboration ego-networks are generated for different researchers, with the objective of determining whether the collaboration network of a researcher belongs to one of the research fields as High Energy Physics, Condensed Matter Physics, or Astro Physics. For Reddit datasets, each network is generated from an online discussion thread where nodes correspond to users, and edges correspond to the responses between users. Here, the task is to identify whether a given network belongs to a *question/answer*-based community or a *discussion*-based community (REDDIT–BINARY), or one of five predefined subreddits (REDDIT–MULTI–5K). Table 5.2 presents the average accuracies along with their standard deviations over ten 10-folds. The results for Weisfeiler–Lehman kernel, DGK kernel, PSCN, and RetGK kernel are taken from Ref. [133]. Specifically in Table 5.2, the best and the

Table 5.2: Average and standard deviation (mean±sd) of the classification accuracy (%) for social network datasets IMDB–BINARY, IMDB–MULTI, COLLAB, REDDIT–BINARY, and REDDIT–MULTI–5K. These social network datasets contain networks that do not have information such as labels and attributes of nodes. In each dataset, the best and the second-best scores are colored in dark pink and light pink, respectively. The notation (∗) indicates that the kernel computation with the implementation in [116] is not completed after 72h.

Method	IMDB–BINARY	IMDB–MULTI	COLLAB	REDDIT–BINARY	REDDIT–MULTI–5K
Scale-variant	74.2 ± 0.9	49.9 ± 0.3	79.6 ± 0.3	87.8 ± 0.3	53.1 ± 0.2
Scale-average	67.7 ± 0.8	44.9 ± 0.4	71.4 ± 0.1	79.8 ± 0.3	51.5 ± 0.2
Scale-norm-average	70.2 ± 0.7	44.9 ± 0.4	62.6 ± 0.1	73.9 ± 0.2	48.7 ± 0.3
CommonMeasures	72.0 ± 0.2	44.9 ± 0.3	75.2 ± 0.1	85.7 ± 0.3	56.6 ± 0.2
KStepRW	60.0 ± 0.8	43.8 ± 0.7	(∗)	(∗)	(∗)
GeometricRW	67.0 ± 0.8	45.2 ± 0.4	(∗)	(∗)	(∗)
ExponentialRW	65.2 ± 1.1	43.1 ± 0.4	(∗)	(∗)	(∗)
ShortestPath	58.2 ± 1.0	42.0 ± 0.6	58.5 ± 0.2	81.9 ± 0.1	49.0 ± 0.1
Graphlet	65.9 ± 1.0	43.9 ± 0.4	72.8 ± 0.3	77.3 ± 0.2	41.0 ± 0.2
Weisfeiler–Lehman	70.8 ± 0.5	49.8 ± 0.5	74.8 ± 0.2	68.2 ± 0.2	51.2 ± 0.3
DGK	67.0 ± 0.6	44.6 ± 0.5	73.1 ± 0.3	78.0 ± 0.4	41.3 ± 0.2
PSCN	71.0 ± 2.3	45.2 ± 2.8	72.6 ± 2.2	86.3 ± 1.6	49.1 ± 0.7
RetGK	71.9 ± 1.0	47.7 ± 0.3	81.0 ± 0.3	92.6 ± 0.3	56.1 ± 0.5

Table 5.3: Average and standard deviation (mean±sd) of the classification accuracy (%) for chemoinformatics and bioinformatics datasets MUTAG, BZR, COX2, DHFR, FRANKENSTEIN, PROTEINS, NCI1, and NCI109. Presented is only a comparison of the methods using the connectivity between nodes. In each dataset, the best and the second-best scores are colored in dark pink and light pink, respectively.

Method	MUTAG	BZR	COX2	DHFR	FRANKEN STEIN	PROTEINS	NCI1	NCI109
Scale-variant	88.2±1.0	85.9±0.9	78.4±0.4	78.8±0.7	69.0±0.2	72.6±0.4	71.3±0.4	69.8±0.2
Scale-average	83.0±1.3	78.9±0.4	78.2±0.0	66.9±0.5	61.3±0.2	70.8±0.2	66.5±0.2	65.8±0.2
Scale-norm -average	84.6±0.9	81.7±0.2	78.2±0.0	61.0±0.0	60.2±0.1	71.7±0.4	65.2±0.1	65.7±0.1
Common Measures	84.9±0.3	82.8±0.3	78.2±0.0	71.1±0.6	62.0±0.2	75.3±0.3	67.8±0.3	65.4±0.1
KStepRW	81.8±1.3	86.5±0.5	78.0±0.1	73.3±0.4	65.4±0.2	71.8±0.1	51.7±0.7	50.4±0.0
GeometricRW	82.9±0.5	79.2±0.4	78.2±0.0	71.4±1.9	55.4±0.1	72.2±0.1	62.6±0.0	63.2±0.0
ExponentialRW	83.0±0.5	79.5±0.5	78.2±0.0	74.6±0.3	55.4±0.1	72.2±0.1	62.7±0.1	63.2±0.1
ShortestPath	81.8±0.9	85.6±0.6	78.1±0.1	73.2±0.5	63.8±0.1	72.0±0.3	64.2±0.1	61.1±2.0
Graphlet	83.0±0.3	78.8±0.0	78.2±0.0	61.0±0.0	55.4±0.0	70.6±0.1	62.4±0.2	62.1±0.1
Weisfeiler –Lehman	83.8±0.8	84.0±1.2	78.3±0.2	77.2±0.6	62.3±1.2	71.3±0.5	63.2±0.1	63.6±0.1

second-best average accuracy scores for each social network dataset are colored dark pink and light pink, respectively. For the social network datasets, the scale-variant method either is comparable or outperforms the state-of-the-art classification methods.

For the chemoinformatics network datasets, we predict the function classes of chemical compounds in chemoinformatics. Here, molecules are represented as small networks with nodes as atoms and edges as covalent bonds. For the bioinformatics dataset (PROTEINS), proteins are represented as networks, where the nodes are secondary structure elements and the edges represent the neighborhood within the 3-D structure or along the amino acid chain. We aim to classify the function class membership of the protein sequences into enzymes and non-enzymes. Note that these chemoinformatics and bioinformatics network datasets contain information on the labels and attributes of the nodes, which is leveraged in DGK, PSCN, and RetGK methods. For a fair comparison of characterizing the structure of networks, we present in Table 5.3 the average accuracies and standard deviations of the methods that only use the connectivity between nodes. In the table, the best and the second-best average accuracy scores for each dataset are colored dark pink and light pink, respectively. Here, on average, the scale-variant method outperforms all the other methods, and offers the best results for six of the eight datasets and the second-best result for two more. Further, the classification accuracies of the scale-variant method on MUTAG, FRANKENSTEIN, NICI1, and NCI109 datasets are at least two percentage points higher than those of the best baseline algorithms. These results suggest that the scale-variant method can be considered as an effective approach in classifying real-world network data.

5.5.4 Detection of transition points in time-evolving networks

We apply the scale-variant topological features to detect transition points between the developmental stages of *Drosophila melanogaster* in the time-evolving gene regulatory networks. Particularly, we use a genome-wide microarray profiling, which shows the expression patterns of 4028 genes simultaneously measured during the developmental stages of *Drosophila melanogaster* [134]. Herein, 66 time points are chosen from the full developmental cycle: embryonic stage (1–30), larval stage (31–40), pupal stage (41–58), and adulthood stage (59–66) [135]. We use kernel reweighted logistic regression method [136] to reconstruct the time-evolving networks for 588 genes, which are known to be related to the developmental process based on their gene ontologies. Therefore, the networks are reconstructed via logistic regression using only binary information, i.e., activation or non-activation of gene expression data. For the likelihood being

Figure 5.8: Three-dimensional persistence diagrams of one-dimensional holes for the *Drosophila melanogaster* gene regulatory networks spanning from $t = 1$ to $t = 18$. The birth-scale and the death-scale axes of the diagrams are represented at the logarithmic scale.

Figure 5.9: Three-dimensional persistence diagrams of one-dimensional holes for the *Drosophila melanogaster* gene regulatory networks spanning from $t = 19$ to $t = 36$. The birth-scale and the death-scale axes of the diagrams are represented at the logarithmic scale.

Figure 5.10: Three-dimensional persistence diagrams of one-dimensional holes for the *Drosophila melanogaster* gene regulatory networks spanning from $t = 37$ to $t = 54$. The birth-scale and the death-scale axes of the diagrams are represented at the logarithmic scale.

Figure 5.11: Three-dimensional persistence diagrams of one-dimensional holes for the *Drosophila melanogaster* gene regulatory networks spanning from $t = 55$ to $t = 66$. The birth-scale and the death-scale axes of the diagrams are represented at the logarithmic scale.

Figure 5.12: Kernel Fisher discriminant ratio κ calculated from the three-dimensional persistence diagrams of one-dimensional holes for the time-evolving *Drosophila melanogaster* gene regulatory networks. Transition time points are detected in the sliding windows spanning between two different developmental stages. In each window, the transition time point t_c is the time index of the maximum κ value marked with the orange point of the dashed line. (a) Windows from the embryonic stage to the larval stage with time points $26 \to 36$ ($t_c = 28$), $27 \to 37$ ($t_c = 28$), and $28 \to 38$ ($t_c = 31$). (b) Windows from the larval stage to the pupal stage with time points $36 \to 46$ ($t_c = 41$), $37 \to 47$ ($t_c = 41$) and $38 \to 48$ ($t_c = 41$). (c) Windows from the pupal stage to the adulthood stage with time points $54 \to 64$ ($t_c = 59$), $55 \to 65$ ($t_c = 59$) and $56 \to 66$ ($t_c = 59$).

maximized for network inference, a kernel weight function is employed to obtain the dynamic networks structures with smooth transition at adjacent time points [136].

We use the scale-variant topological features to detect the transition points with respect to the topological structure of the constructed time-evolving networks. Consequently, the detected transition points agree with the transition points in the dynamics of *Drosophila melanogaster*. Here, the change points between the developmental stages of *Drosophila melanogaster* chosen in the experiment are referred to as the transition points in the dynamics: $t_1 = 31$ between the embryonic stage and the larval stage, $t_2 = 41$ between the larval stage and the pupal stage, and $t_3 = 59$ between the pupal stage and the adulthood stage. For the network of each time point, three-dimensional persistence diagrams are computed for one-dimensional holes with $\tau_1 = 1, \ldots, \tau_{100} = 100$. Figures 5.8–5.11 show1 the examples of the diagrams for networks spanning from $t = 1$ to $t = 18$ (Fig. 5.8), $t = 19$ to $t = 36$ (Fig. 5.9), and $t = 37$ to $t = 54$ (Fig. 5.10) and $t = 55$ to $t = 66$ (Fig. 5.11).

Here, note that the patterns in scale-variant topological features are transformed along with time points. Such patterns transformation corresponds with the transformation in the topological structure of time-evolving reconstructed networks. At the beginning of the developmental process, genes have a more fixed and localized function, therefore they mainly interact with other genes with similar functions to make more dominant cycles in the mesoscale at the high value of τ. However, after mid-embryonic stage, genes become more versatile and involved in more diverse roles to serve the need of rapid development. This leads to the formation of mesoscale at the low values of τ. As the organism turns into an adult, its growth slows down and each gene may be restored to its more specialized role.

Moreover, we quantify the transition points with respect to the topological structure by observing the sliding windows spanning two different developmental stages of *Drosophila melanogaster*. In each sliding window, we compute the kernel Fisher discriminant ratio [57] for each time point from the three-dimensional persistence diagrams of the networks. The time point of the maximum ratio can be identified as the transition time point t_c in each window (Fig. 5.12). From the embryonic stage to the larval stage, we obtain the transition time points in the topological structure as $t_c = 28$ and $t_c = 31$, relatively close to the experimentally known transition time point $t_1 = 31$. Furthermore, from the larval stage to the pupal stage, and from the pupal stage to the adulthood stage, we obtain the transition time points in the topological structure as $t_c = 41$ and $t_c = 59$, respectively. These points agree with the experimentally known transition time points $t_2 = 41$ and $t_3 = 59$.

5.5.5 Considerations on the maximum value of the diffusion timescale

We investigate the maximum timescale $\tau_{\max}$ to examine the length of the diffusion process that must be explored. Note that the timescale functions as a resolution parameter to unravel the multi-scale and hierarchical structure of a network. A small timescale restricts random walkers in local interactions, which produces many communities in the network. In contrast, a large timescale leads to a substantial contribution of long walks and therefore yields a small number of communities because random walkers tend to remain in these communities for a long time. This resolution problem has been addressed in Refs. [137, 138], in which the relevance of partitions as community structures is characterized over timescales. Instead of characterizing the network structure at a fixed resolution, the scale-variant topological features obtained with a sufficiently large $\tau_{\max}$ can contain information about the network at multiple resolutions.

Here, we study the method for determining $\tau_{\max}$ through the spectral decomposition of the normalized Laplacian of the network. Denoting the eigenvalues of $\boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}}$ by λ_i in increasing order $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$, the spectral decomposition of $\boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}}$ is expressed as follows:

$$\boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}} = \sum_{i=1}^{n} \lambda_i \boldsymbol{\theta}_i^\top \boldsymbol{\theta}_i, \tag{5.27}$$

where $\boldsymbol{\theta}_i$ is the eigenvector associated with λ_i. Therefore, the solution for Eq. (5.1) can be written as follows:

$$\boldsymbol{p}_{\mathcal{G}}(\tau|i) = \sum_{i=1}^{n} \exp\left(-\lambda_i \tau\right) \boldsymbol{\mu}_i \boldsymbol{\theta}_i^\top \boldsymbol{\theta}_i. \tag{5.28}$$

From Eq. (5.28), each eigenvalue λ_i of $\boldsymbol{L}_{\mathcal{G}}^{\mathrm{rw}}$ is associated with a decaying mode in the diffusion process with the characteristic timescale $\tau = 1/\lambda_i$. Therefore, if there are large gaps between eigenvalues, for example if the k_0 smallest eigenvalues $\{\lambda_1 = 0, \ldots, \lambda_{k_0}\}$ are greatly separated from the remaining eigenvalues ($\lambda_{k_0} \ll \lambda_{k_0+1} = \lambda_{\mathrm{sep}}$), we can ignore the terms associated with $\{\lambda_{k_0+1}, \ldots, \lambda_n\}$ in Eq. (5.28) at τ satisfying $\tau\lambda_{\mathrm{sep}} \gg 1$. Thus, there is no significant change in the formation of clusters or loops in the mapped point cloud $P_{\mathcal{G}}(\tau)$, nor in the formation of communities in the network at the τ-scale. Therefore, as a heuristic method, if we consider $\tau_{\max}$ such that $\tau_{\max}\lambda_{\mathrm{sep}} \gg 1$, the structure of the network is well characterized via the scale-variant topological features.

Here, we verify the above consideration by distinguishing the network of the Barabási–Albert (BA) growth model [139] with its configuration network. The network is initialized with m_0 nodes and no edges. At each time step, each new node is

Figure 5.13: Eigenvalues λ_k such that $0 \leq \lambda_k \leq 0.5$ for (a) Barabási–Albert (BA) networks and (b) scale-free configuration networks. The colors of the points correspond to values of $m_0 = 1, 2, 3, 4, 5, 6$ (number of initial nodes in the BA growth model). The value of λ_{sep} to effectively separate eigenvalues is identified for each network. The smallest λ_{sep} values for the BA networks and configuration networks are $\lambda_{\text{sep}}^{\text{BA}} \approx 0.18$ and $\lambda_{\text{sep}}^{\text{conf}} \approx 0.16$, respectively, as marked.

added with no more than m_0 links to the existing nodes in the network. The probability that a new node is connected to an existing node is proportional to the degree of the existing node. Note that both the BA networks and their configuration networks have a scale-free property with degree exponent 3. We set the number of nodes to 128, vary the number of initial nodes $m_0 = 1, 2, \ldots, 50$, and generate 10 networks via this process for each value of m_0. The 10 networks generated at each m_0 are split into two parts, with five networks for training and the remaining five for testing. Figure 5.13 depicts eigenvalues λ_k such that $0.0 \leq \lambda_k \leq 0.5$ for (a) BA networks and (b) scale-free configuration networks. The colors of the points correspond to values of $m_0 = 1, 2, 3, 4, 5, 6$. From Fig. 5.13, we can identify the value of λ_{sep} to separate the eigenvalues for each network. The smallest λ_{sep} values for BA networks and configuration networks are $\lambda_{\text{sep}}^{\text{BA}} \approx 0.18$ and $\lambda_{\text{sep}}^{\text{conf}} \approx 0.16$, respectively. Therefore, $\tau_{\max}$ should be set to $\tau_{\max}\lambda_{\text{sep}}^{\text{BA}} \gg 1$ and $\tau_{\max}\lambda_{\text{sep}}^{\text{conf}} \gg 1$, for instance, $\tau_{\max}\lambda_{\text{sep}}^{\text{BA}} > 10$ and $\tau_{\max}\lambda_{\text{sep}}^{\text{conf}} > 10$, or $\tau_{\max} \geq 65$.

For each $\tau_{\max}$ in $\{5, 10, 15, \ldots, 95, 100\}$, we compute three-dimensional persistence diagrams of one-dimensional holes with $\tau_1 = 1, \tau_2 = 2, \tau_3 = 3, \ldots, \tau_K = \tau_{\max}$. The line in Fig. 5.14 depicts the average test accuracy over 100 random train-test splits at each value of $\tau_{\max}$. The shaded area indicates the confidence intervals of one standard deviation calculated using the ensemble of runs. In general, increasing $\tau_{\max}$ serves to increase classification accuracy because the diffusion process gathers more information about the network structure. There is a transition in classification accuracy with large

Figure 5.14: Classification of networks generated from the Barabási–Albert model with their configuration networks using scale-variant topological features. The line depicts the average test accuracy over 100 random train-test splits at each value of $\tau_{\max}$. The shaded area indicates the confidence intervals of one standard deviation calculated using the same ensemble of runs. The inset highlights the transition in accuracy over $\tau_{\max} = 20$–50.

deviations when $\tau_{\max}$ increases from 30 to 40. To demonstrate this transition in more detail, we plot $\tau_{\max} = 20, 21, \ldots, 49, 50$ in the inset of Fig. 5.14. For $\tau_{\max} < 30$, only microscale structures are considered in the features, which has a small effect on the differences between networks. The transition occurs when the mesoscale structures are considered. For $\tau_{\max} > 40$, the deviation is reduced as the mesoscale structures are revealed. For sufficiently large $\tau_{\max}$ ($\tau_{\max} \geq 65$), the method achieves high accuracy ranging from 94% to 94.5%. This observation agrees with the above-mentioned heuristic for determining $\tau_{\max}$.

We have demonstrated above that with appropriate selection of $\tau_{\max}$, our method can distinguish BA networks from their corresponding configuration networks. Additionally, our method can also distinguish BA networks from the networks generated using the scale-free configuration model of degree exponent 3. Here, we set the number of nodes to 200 and generate 100 networks using the scale-free configuration model of degree exponent 3. To compare with a network density of similar magnitude, we set $m_0 = 1, 2, 3, 4, 5$, and generate 20 networks for each value of m_0 in the BA model. For each $\tau_{\max}$ in $\{5, 10, 15, \ldots, 95, 100\}$, we compute three-dimensional persistence diagrams of one-dimensional holes with $\tau_1 = 1, \tau_2 = 2, \tau_3 = 3, \ldots, \tau_K = \tau_{\max}$. In each model, the 100 networks are split into two parts, with 50 networks for training and the remaining 50 for testing. The line in Fig. 5.15 depicts the average test accuracy over 100 random train-test splits at each value of $\tau_{\max}$. The shaded area indicates the confidence intervals of one standard deviation calculated using the ensemble of runs.

Figure 5.15: Classification of networks generated from the Barabási–Albert model with networks generated from the scale-free configuration model of degree exponent 3. The line depicts the average test accuracy over 100 random train-test splits at each value of $\tau_{\max}$. The shaded area indicates the confidence intervals of one standard deviation calculated using the same ensemble of runs.

For sufficiently large $\tau_{\max}$ ($\tau_{\max} \geq 20$), the method achieves high accuracy ranging from 96% to 98%. Note that $\tau_{\max}$ can be determined using the same aforementioned heuristic process.

5.6 Concluding remarks and discussion

Our study mainly aimed to represent the variation of topological scales, capture the nondyadic interactions, and provide robustness against noise in characterizing the structure of complex networks. Here, we proposed a general framework for constructing the scale-variant topological features from the diffusion process exhibited in networks. The scale-variant topological features do not directly correspond to the common statistical measures that are constructed from the dyadic interactions between nodes at a single fixed topological scale. Rather, our features encode the information of both dyadic and nondyadic interactions in networks at variant topological scales.

In the networks classification, our features acted as strong factors to identify the networks. Theoretically, we derived a strong mathematical guarantee for the robustness of these features with respect to the perturbations applied to the networks. Through several experiments, we provided an empirical evidence for the effectiveness of these features in applications, such as classification of real-world networks and detection of transition points, with respect to the topological structure in time-evolving networks. The results suggested that the observation of the topological features induced from the network dynamics over variant scales can characterize the structure and provide important insights for understanding the functionality of networks.

In our experiments, the scale-variant topological features were constructed from zero-dimensional and one-dimensional holes. In principle, we can compute the features from higher-dimensional holes that represent nondyadic interactions, involving a larger number of nodes in each interaction. However, to investigate the features from high-dimensional holes, the Vietoris–Rips filtration used in our study can consist of a large number of simplices. More precisely, to consider l-dimensional holes, the Vietoris–Rips filtration has size $O(N^{l+2})$ of the number of simplices. Herein, N is the number of nodes in the network. This observation shows the difficulty of using the features from l-dimensional holes ($l \geq 2$) for networks with a large number of nodes. On the contrary and as demonstrated in this paper, we found it sufficient to use l-dimensional holes with $l = 0, 1$ in practical applications. Furthermore, one can replace the Vietoris–Rips filtration with the Witness filtration [140] or the approximation of the Vietoris–Rips filtration [141] for more efficient computations. We employed recent algorithmic improvements to efficiently compute the persistence diagrams with the core implementation referenced from Ripser libary [142].

Another point to discuss is the selection of the maximum diffusion timescale. In our experiments, our method was only tested for small and medium-sized networks with less than 5,000 nodes per network. For larger networks, a longer diffusion timescale must be explored. This consideration will increase the computational cost of computing persistence diagrams and the kernel. However, this limitation can be mitigated by

increasing the sampling interval to take discrete values of the timescale while keeping the maximum timescale sufficiently large. It is also possible to study the process of taking values of timescales based on the spectral decomposition of the normalized Laplacian of the network.

Our study is motivated by Ref. [49], in which the diffusion geometry from the diffusion process is used to reveal functional clusters in a network. Based on random walk dynamics, the diffusion distance between a pair of nodes in a network is defined and averaged in a range of timescales to model the underlying geometry of the network. However, the variation in network structure over the diffusion timescale is not discussed. In Ref. [138], a random walk process corresponding to the natural dynamics of a system focuses on recovering dynamically meaningful communities in the network. Our approach involves a diffusion process similar to that mentioned in Refs. [49, 138]; however, it mainly focuses on the systematic representation of networks via topological data analysis by tracking the variation of the topological structures along timescales of the diffusion process.

In general, our study presented a unified analysis of complex networks. This study paves several opportunities for designing effective algorithms in network science, such as an investigation of more complicated network structures. For instance, we can employ our framework to study different aspects of multiplex networks, or to study the structural reducibility of a multilayer network while preserving its dynamics and function.

Chapter 6

Conclusion

This chapter briefly summarizes the contributions of this thesis, discusses their implications and gives an outlook to possible future work.

In this thesis, we proposed for the first time the so-called framework *variant topological data analysis*, which extends the topological data analysis framework, and specifically, persistent homology to analyze various types of complex data. We provided the mathematical definition and the systematic method to obtain variant topological features of data from the framework, then constructed the kernel method for these features to be used in statistical-learning tasks (Chapter 3). We also demonstrated that these features could be used effectively in a robust way in time-series data analysis (Chapter 4), and complex networks data analysis (Chapter 5).

More generally, the next long-term step for this work is to fit current trends, both in Topological Data Analysis and in Machine Learning techniques in physical science. Although the revolutionized machine learning approaches open new avenues in many areas of physics [143], these approaches struggle with physical interpretations and intuitive explanations. We need to address the issue of physical interpretations by relying on the nature properties that are relevant to the robust topological features of the physical states. The basic idea of topological data analysis is that topology can indicate the topological properties of a space that remain invariant under stretching and shrinking, such as the number of holes and the number of connected clusters. These concepts motivate the application of topological data analysis in studying many problems in physics, such as the understanding of the topological phase of matter or detecting topological phase transitions.

This thesis was motivated by a large number of complex data sets in many application domains, which require a useful and universal method to analyze the variant-scale properties in the dynamics of the data. The main goal was to demonstrate that the exploration of algebraic topological concepts in data via variant scales can be used as both qualitative and quantitative features to understand the complex dynamics of data. As demonstrated in the thesis through various applications, it became clear that variant topological data analysis is capable of augmenting, supporting, and even

surpassing existing approaches for complex data analysis. Although the applicability to data such as time series and complex networks were extensively studied, this thesis only paves a stepping stone for the research in this field. It opens new promising possibilities such as the application of variant topological data analysis to understand the quantum dynamics of data and to strengthen the robustness of machine learning methods in physical science.

Appendix A

Appendix A

A.1 Common measures for a network

For each network, we calculate the following 18 common measures: the density (the ratio of the existing to the possible edges), the transitivity [94] (the proportion of triangles), the diameter (the maximum eccentricity), the radius (the minimum eccentricity), the degree assortativity coefficient [114], the global efficiency [113], the number of connected parts, the average clustering coefficient, the average number of triangles that include a node as a vertex, the average local efficiency [113], the average edge betweenness centrality [115], the average node betweenness centrality [112], the average node closeness centrality [112], the average eccentricity, the average shortest paths, the average degree centrality [112], the maximum modularity which is obtained by Louvain heuristic [104, 110], and the average of global mean first-passage times of random walks on the network [144]. We normalize the measures in the range of $[0, 1]$ using the min-max normalization (i.e. $f_* = (f_* - f_{\min})/(f_{\max} - f_{\min})$, where $f_{\min}, f_{\max}$ are the minimum and the maximum values of a measure in the data).

A.2 Graph kernel methods

We describe the graph kernel methods used in the main text. The implementations of these graph kernels can be found in Ref. [116].

A.2.1 Random walk kernels

The random walk graph kernels measure the similarity between a pair of graphs based on the number of equal-length walks in two graphs. Given two unlabeled graphs $\mathcal{G}$ and $\mathcal{G}'$ with their vertex and edge sets as $(\mathcal{V}, \mathcal{E})$ and $(\mathcal{V}', \mathcal{E}')$, respectively, the direct product graph $\mathcal{G}_\times = (\mathcal{V}_\times, \mathcal{E}_\times)$ of $\mathcal{G}$ and $\mathcal{G}'$ is a graph with the node set $\mathcal{V}_\times = \{(v, v') \mid v \in \mathcal{V}, v' \in \mathcal{V}'\}$ and the edge set $\mathcal{E}_\times = \{((v_a, v'_a), (v_b, v'_b)) \mid (v_a, v_b) \in \mathcal{E}, (v'_a, v'_b) \in \mathcal{E}'\}$.

The KStepRW kernel is the k-step random walk kernel $\mathcal{K}_{\times}^{k}$ defined as

$$\mathcal{K}_{\times}^{k}(\mathcal{G},\mathcal{G}') = \sum_{i,j=1}^{|\mathcal{V}_{\times}|} \sum_{m=0}^{k} [\lambda_m \boldsymbol{W}_{\times}^{m}]_{ij}\,, \tag{A.1}$$

where $\boldsymbol{W}_{\times}$ is a weight matrix of $\mathcal{G}_{\times}$ and $\lambda_0,\ldots,\lambda_k$ is a sequence of positive, real-valued weights. In our experiments, we set $k = 2$ and $\lambda_0 = \lambda_1 = \lambda_2 = 1.0$.

The GeometricRW kernel is a specific case of the k-step random walk kernel, when k goes to infinity and the weights are the geometric series, i.e., $\lambda_m = \lambda^m$ ($\lambda = 0.05$ in our experiments). The GeometricRW kernel is defined as

$$\mathcal{K}_{\mathrm{GR}}(\mathcal{G},\mathcal{G}') = \sum_{i,j=1}^{|\mathcal{V}_{\times}|} \sum_{m=0}^{\infty} [\lambda^m \boldsymbol{W}_{\times}^{m}]_{ij} = \sum_{i,j=1}^{|V_{\times}|} \left[(\boldsymbol{I} - \lambda \boldsymbol{W}_{\times})^{-1}\right], \tag{A.2}$$

where $\boldsymbol{I}$ is an identity matrix of size $|\mathcal{V}_{\times}| \times |\mathcal{V}_{\times}|$.

The ExponentialRW kernel is a specific case of the k-step random walk kernel, when k goes to infinity and the weights are the exponential series, i.e., $\lambda_m = \frac{\beta^m}{m!}$ ($\beta = 0.1$ in our experiments). The ExponentialRW kernel is defined as

$$\mathcal{K}_{\mathrm{EX}}(\mathcal{G},\mathcal{G}') = \sum_{i,j=1}^{|V_{\times}|} \sum_{m=0}^{\infty} \left[\frac{(\beta \boldsymbol{W}_{\times})^m}{m!}\right]_{ij} = \sum_{i,j=1}^{|V_{\times}|} \left[e^{\beta \boldsymbol{W}_{\times}}\right]_{ij}. \tag{A.3}$$

A.2.2 ShortestPath kernel

The ShortestPath kernel compares all pairs of the shortest path lengths from $\mathcal{G}$ and $\mathcal{G}'$ defined as

$$\mathcal{K}_{\mathrm{SP}}(\mathcal{G},\mathcal{G}') = \sum_{v_i,v_j \in \mathcal{G}} \sum_{v'_k,v'_l \in \mathcal{G}'} \delta(d(v_i,v_j), d(v'_k,v'_l)), \tag{A.4}$$

where $d(v_i,v_j)$ and $d(v'_k,v'_l)$ are the lengths of the shortest path between nodes v_i and v_j in $\mathcal{G}$, and the shortest path between nodes v'_k and v'_l in $\mathcal{G}'$, respectively. Here, $\delta(x,y) = 1$ if $x = y$, and 0 if $x \neq y$.

A.2.3 Graphlet kernel

A size-k graphlet is an induced and non-isomorphic sub-graph of size k. Let $\mathcal{S}_k = \{G_1,\ldots,G_{N_k}\}$ be a set of size-k graphlets, where N_k denotes the number of unique graphlets of size k. For an unlabeled graph $\mathcal{G}$ (the graph does not contain attributes for nodes), we define a vector $\boldsymbol{f}_{\mathcal{G}}$ of length N_k such that the ith component of $\boldsymbol{f}_{\mathcal{G}}$ denotes the frequency of graphlet G_i appearing as a subgraph of $\mathcal{G}$. Given two unlabeled

graphs $\mathcal{G}$ and $\mathcal{G}'$, the graphlet kernel is defined as

$$\mathcal{K}_{\mathrm{GK}}(\mathcal{G}, \mathcal{G}') = \langle \boldsymbol{f}_{\mathcal{G}}, \boldsymbol{f}_{\mathcal{G}'} \rangle, \tag{A.5}$$

where $\langle \cdot, \cdot \rangle$ represents the Euclidean dot product. We set $k = 4$ for MUTAG, BZR, DHFR and FRANKENSTEIN datasets and $k = 3$ for the other datasets.

A.2.4 Weisfeiler–Lehman kernel

Weisfeiler–Lehman kernel decomposes a graph into its subtree patterns and compares these patterns in two graphs. For an unlabeled graph $\mathcal{G}$, all vertexes v of $\mathcal{G}$ are initialized with label $\varphi(v) = 0$. We iterate over each vertex v and its neighbour to create a multiset label as $\varphi^{(i)}(v)$ such that $\varphi^{(1)}(v) = \varphi(v)$, and $\varphi^{(i)}$ with $i > 1$ is defined as $\varphi^{(i)}(v) = (\varphi^{(i)}(v), \mathrm{Q}_v^{(i-1)})$, where $\mathrm{Q}_v^{(i-1)}$ is the sorted labels of v's neighbours. To measure the similarity between graphs, we count the co-occurrences of the labels in both graphs for h iterations with the kernel defined as

$$\mathcal{K}_{\mathrm{WL}}(\mathcal{G}, \mathcal{G}') = \langle \mathbf{1}_{\mathcal{G}}, \mathbf{1}_{\mathcal{G}'} \rangle. \tag{A.6}$$

Here, $\mathbf{1}_{\mathcal{G}}$ is the vector concatenation of h vertex label histograms $\mathbf{1}_{\mathcal{G}}^{(1)}, \ldots, \mathbf{1}_{\mathcal{G}}^{(h)}$ in h iterations. We set $h = 5$ in our experiments.

Bibliography

[1]G. Carleo, and M. Troyer, "Solving the quantum many-body problem with artificial neural networks", Science **355**, 602–606 (2017).

[2]X. Gao, and L.-M. Duan, "Efficient representation of quantum many-body states with deep neural networks", Na. Commun. **8**, 662 (2017).

[3]J. J. Sørensen, M. Aranburu, T. Heinzel, and J. Sherson, "Approaching the quantum speed limit with global-local optimization", Preprint at arXiv:1802.07521 (2018).

[4]M. Pyzh, S. Krönke, C. Weitenberg, and P. Schmelcher, "Quantum point spread function for imaging trapped few-body systems with a quantum gas microscope", New Journal of Physics **21**, 053013 (2019).

[5]C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks", in Proceedings of the 2nd international conference on learning representations (2014).

[6]G. Carlsson, "Topology and data", Bull. Amer. Math. Soc. **46**, 255–308 (2009).

[7]H. Edelsbrunner, and J. Harer, *Computational topology. an introduction* (American Mathematical Society, Providence, RI, 2010).

[8]A. Hatcher, *Algebraic topology* (Cambridge University Press, Cambridge, 2002).

[9]H. Edelsbrunner, and J. Harer, "Persistent homology—a survey, in "surveys on discrete and computational geometry. twenty years later"", Contemporary Mathematics **453**, 257–282 (2008).

[10]H. Edelsbrunner, D. Letscher, and A. Zomorodian, "Topological persistence and simplification", Discrete Comput. Geom. **28**, 511–533 (2002).

[11]M. Kramar, A. Goullet, L. Kondic, and K. Mischaikow, "Persistence of force networks in compressed granular media", Phys. Rev. E **87**, 042207 (2013).

[12]S. Ardanza-Trevijano, I. Zuriguel, R. Arévalo, and D. Maza, "Topological analysis of tapped granular media using persistent homology", Phys. Rev. E **89**, 052212 (2014).

[13]M. Kramár, A. Goullet, L. Kondic, and K. Mischaikow, "Evolution of force networks in dense particulate media", Phys. Rev. E **90**, 052203 (2014).

[14]T. Nakamura, Y. Hiraoka, A. Hirata, E. G. Escolar, and Y. Nishiura, "Persistent homology and many-body atomic structure for medium-range order in the glass", Nanotechnology **26**, 304001 (2015).

[15]Y. Hiraoka, T. Nakamura, A. Hirata, E. G. Escolar, K. Matsue, and Y. Nishiura, "Hierarchical structures of amorphous solids characterized by persistent homology", Proc. Natl. Acad. Sci. USA **113**, 7035–7040 (2016).

[16]T. Ichinomiya, I. Obayashi, and Y. Hiraoka, "Persistent homology analysis of craze formation", Phys. Rev. E **95**, 012504 (2017).

[17]T. Takahashi, A. H. Clark, T. Majmudar, and L. Kondic, "Granular response to impact: topology of the force networks", Phys. Rev. E **97**, 012906 (2018).

[18]D. Taylor, F. Klimm, H. A. Harrington, M. Kramár, K. Mischaikow, M. A. Porter, and P. J. Mucha, "Topological data analysis of contagion maps for examining spreading processes on networks", Nat. Commun. **6**, 7723 (2015).

[19]S. Maletić, Y. Zhao, and M. Rajković, "Persistent topological features of dynamical systems", Chaos **26**, 053105 (2016).

[20]I. Donato, M. Gori, M. Pettini, G. Petri, S. De Nigris, R. Franzosi, and F. Vaccarino, "Persistent homology analysis of phase transitions", Phys. Rev. E **93**, 052138 (2016).

[21]K. Mittal, and S. Gupta, "Topological characterization and early detection of bifurcations and chaos in complex systems using persistent homology", Chaos **27**, 051102 (2017).

[22]M. Sinhuber, and N. T. Ouellette, "Phase coexistence in insect swarms", Phys. Rev. Lett. **119**, 178003 (2017).

[23]L. Speidel, H. A. Harrington, S. J. Chapman, and M. A. Porter, "Topological data analysis of continuum percolation with disks", Phys. Rev. E **98**, 012318 (2018).

[24]Q. H. Tran, and Y. Hasegawa, "Topological time-series analysis with delay-variant embedding", Phys. Rev. E **99**, 032209 (2019).

[25]A. Myers, E. Munch, and F. A. Khasawneh, "Persistent homology of complex networks for dynamic state detection", Phys. Rev. E **100**, 022314 (2019).

[26]J. M. Chan, G. Carlsson, and R. Rabadan, "Topology of viral evolution", Proc. Natl. Acad. Sci. USA **110**, 18566–18571 (2013).

[27]P. G. Camara, D. I. Rosenbloom, K. J. Emmett, A. J. Levine, and R. Rabadan, "Topological data analysis generates high-resolution, genome-wide maps of human recombination", Cell Systems **3**, 83 –94 (2016).

[28]M. Nicolau, A. J. Levine, and G. Carlsson, "Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival", Proc. Natl. Acad. Sci. USA **108**, 7265–7270 (2011).

[29]J. Arsuaga, T. Borrman, R. Cavalcante, G. Gonzalez, and C. Park, "Identification of copy number aberrations in breast cancer subtypes using persistence topology", Microarrays **4**, 339–369 (2015).

[30]L. Li, W.-Y. Cheng, B. S. Glicksberg, O. Gottesman, R. Tamler, R. Chen, E. P. Bottinger, and J. T. Dudley, "Identification of type 2 diabetes subgroups through topological analysis of patient similarity", Sci. Transl. Med. **7**, 311ra174–311ra174 (2015).

[31]T. S. Hinks, T. Brown, L. C. Lau, H. Rupani, C. Barber, S. Elliott, J. A. Ward, J. Ono, S. Ohta, K. Izuhara, et al., "Multidimensional endotyping in patients with severe asthma reveals inflammatory heterogeneity in matrix metalloproteinases and chitinase 3–like protein 1", J. Allergy. Clin. Immunol. **138**, 61–75 (2016).

[32]J. D. Cahoy, B. Emery, A. Kaushal, L. C. Foo, J. L. Zamanian, K. S. Christopherson, Y. Xing, J. L. Lubischer, P. A. Krieg, S. A. Krupenko, et al., "A transcriptome database for astrocytes, neurons, and oligodendrocytes: a new resource for understanding brain development and function", J. Neurosci. **28**, 264–278 (2008).

[33]J. A. Perea, A. Deckard, S. B. Haase, and J. Harer, "Sw1pers: sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data", BMC Bioinforma. **16**, 257 (2015).

[34]Q. H. Tran, V. T. Vo, and Y. Hasegawa, "Scale-variant topological information for characterizing the structure of complex networks", Phys. Rev. E **100**, 032308 (2019).

[35]G. Petri, P. Expert, F. Turkheimer, R. Carhart-Harris, D. Nutt, P. J. Hellyer, and F. Vaccarino, "Homological scaffolds of brain functional networks", J. Royal Soc. Interface **11**, 20140873 (2014).

[36]F. A. N. Santos, E. P. Raposo, M. D. Coutinho-Filho, M. Copelli, C. J. Stam, and L. Douw, "Topological phase transitions in functional brain networks", Phys. Rev. E **100**, 032414 (2019).

[37]G. Carlsson, *Why topological data analysis works?*, `https://www.ayasdi.com/blog/bigdata/why-topological-data-analysis-works/`, 2015 (accessed November 3, 2019).

[38]G. Singh, F. Mémoli, and G. E. Carlsson, "Topological methods for the analysis of high dimensional data sets and 3d object recognition", in Symposium on point based graphics (2007), pp. 91–100.

[39]T. Kaczynski, K. Mischaikow, and M. Mrozek, *Computational homology*, Vol. 157 (Springer, 2006).

[40]S. A. Marvel, J. Kleinberg, R. D. Kleinberg, and S. H. Strogatz, "Continuous-time model of structural balance", Proc. Natl Acad. Sci. USA **108**, 1771–1776 (2011).

[41]A. van der Schaft, S. Rao, and B. Jayawardhana, "On the mathematical structure of balanced chemical reaction networks governed by mass action kinetics", SIAM J. Appl. Math. **73**, 953–973 (2013).

[42]D. Horak, S. Maletić, and M. Rajković, "Persistent homology of complex networks", J. Stat. Mech. Theory Exp. **2009**, P03034 (2009).

[43]G. Petri, M. Scolamiero, I. Donato, and F. Vaccarino, "Topological strata of weighted complex networks", PLOS ONE **8**, e66506 (2013).

[44]M. W. Reimann, M. Nolte, M. Scolamiero, K. Turner, R. Perin, G. Chindemi, P. Dłotko, R. Levi, K. Hess, and H. Markram, "Cliques of neurons bound into cavities provide a missing link between structure and function", Front. Comput. Neurosci. **11**, 48 (2017).

[45]A. Sizemore, C. Giusti, and D. S. Bassett, "Classification of weighted networks through mesoscale homological features", J. Complex Netw. **5**, 245–273 (2017).

[46]I. Iacopini, G. Petri, A. Barrat, and V. Latora, "Simplicial models of social contagion", Nat. Commun. **10**, 2485 (2019).

[47]A. Sizemore, J. Phillips-Cremins, R. Ghrist, and D. Bassett, "The importance of the whole: topological data analysis for the network neuroscientist", Network Neuroscience **3**, 1–18 (2018).

[48]F. Takens, "Detecting strange attractors in turbulence", in *Dynamical systems and turbulence, warwick 1980*, Vol. 898, edited by D. A. Rand, and L.-S Young, Lecture Notes in Mathematics (Springer-Verlag, Berlin, Heidelberg, 1981), pp. 366–381.

[49]M. De Domenico, "Diffusion geometry unravels the emergence of functional clusters in collective phenomena", Phys. Rev. Lett. **118**, 168301 (2017).

[50]Y. Wang, *Lecture notes in (cse 5559) computational topology: theory, algorithms, and applications to data analysis*, `http://web.cse.ohio-state.edu/~wang.1016/courses/5559/`, 2018 (accessed July 3, 2019).

[51]N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington, "A roadmap for the computation of persistent homology", EPJ Data Sci. **6**, 17 (2017).

[52]S. Y. Oudot, *Persistence Theory: From Quiver Representations to Data Analysis*, Mathematical Surveys and Monographs 209 (Am. Math. Soc., 2015), p. 218.

[53]J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt, "A stable multi-scale kernel for topological machine learning", in Proceedings of the 28th ieee conference on computer vision and pattern recognition (2015), pp. 4741–4748.

[54]T. Le, and M. Yamada, "Persistence fisher kernel: a riemannian manifold kernel for persistence diagrams", in *Advances in neural information processing systems 31*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, (Curran Associates, Inc., 2018), pp. 10007–10018.

[55]N. Aronszajn, "Theory of reproducing kernels", Trans. Am. Math. Soc. **68**, 337–404 (1950).

[56]A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. J. Smola, "A kernel statistical test of independence", in *Advances in neural information processing systems 20*, edited by J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, (Curran Associates, Inc., 2008), pp. 585–592.

[57]Z. Harchaoui, E. Moulines, and F. R. Bach, "Kernel change-point analysis", in Advances in neural information processing systems 21, edited by D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, (2009), pp. 609–616.

[58]G. Kusano, K. Fukumizu, and Y. Hiraoka, "Persistence weighted gaussian kernel for topological data analysis", in Proceedings of the 33th international conference on machine learning, Vol. 48 (2016), pp. 2004–2013.

[59]M. Casdagli, S. Eubank, J. D. Farmer, and J. Gibson, "State space reconstruction in the presence of noise", Physica D **51**, 52–98 (1991).

[60]A. M. Fraser, and H. L. Swinney, "Independent coordinates for strange attractors from mutual information", Phys. Rev. A **33**, 1134–1140 (1986).

[61]W. Liebert, and H. Schuster, "Proper choice of the time delay for the analysis of chaotic time series", Phys. Lett. A **142**, 107–111 (1989).

[62]T. Buzug, and G. Pfister, "Optimal delay time and embedding dimension for delay-time coordinates by analysis of the global static and local dynamical behavior of strange attractors", Phys. Rev. A **45**, 7073–7084 (1992).

[63]T. Buzug, and G. Pfister, "Comparison of algorithms calculating optimal embedding parameters for delay time coordinates", Physica D **58**, 127 –137 (1992).

[64]M. T. Rosenstein, J. J. Collins, and C. J. De Luca, "Reconstruction expansion as a geometry-based framework for choosing proper delay times", Physica D **73**, 82–98 (1994).

[65]H. Kantz, and T. Schreiber, *Nonlinear time series analysis*, 2nd ed. (Cambridge University Press, New York, NY, USA, 2003).

[66]E. Bradley, and H. Kantz, "Nonlinear time-series analysis revisited", Chaos **25**, 097610 (2015).

[67]Q. H. Tran, and Y. Hasegawa, *Delay-variant embedding*, `https://github.com/OminiaVincit/delay-variant-embed`, GitHub Repository.

[68]F. Chazal, V. de Silva, and S. Oudot, "Persistence stability for geometric complexes", Geometriae Dedicata **173**, 193–214 (2014).

[69]N. E. Phillips, C. Manning, N. Papalopulu, and M. Rattray, "Identifying stochastic oscillations in single-cell live imaging time series using gaussian processes", PLOS Comp. Biol. **13**, 1–30 (2017).

[70]N. A. Monk, "Oscillatory expression of hes1, p53, and NF-κB driven by transcriptional time delays", Curr. Biol. **13**, 1409 –1413 (2003).

[71]T. Galla, "Intrinsic fluctuations in stochastic delay systems: theoretical description and application to a simple model of gene regulation", Phys. Rev. E **80**, 021909 (2009).

[72]D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions", J. Phys. Chem. **81**, 2340–2361 (1977).

[73]D. F. Anderson, "A modified next reaction method for simulating chemical systems with time dependent propensities and delays", J. Phys. Chem. **127**, 214107 (2007).

[74]T. Brett, and T. Galla, "Stochastic processes with distributed delays: chemical langevin equation and linear-noise approximation", Phys. Rev. Lett. **110**, 250601 (2013).

[75]C. M. Bishop, *Pattern recognition and machine learning* (Springer-Verlag, Berlin, Heidelberg, 2006).

[76]Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, *The UCR Time Series Classification Archive*, `www.cs.ucr.edu/~eamonn/time_series_data/`, 2015.

[77]A. E. X. Brown, E. I. Yemini, L. J. Grundy, T. Jucikas, and W. R. Schafer, "A dictionary of behavioral motifs reveals clusters of genes affecting caenorhabditis elegans locomotion", Proc. Natl. Acad. Sci. U.S.A. **110**, 791–796 (2013).

[78]E. Yemini, T. Jucikas, L. J. Grundy, A. E. Brown, and W. R. Schafer, "A database of caenorhabditis elegans behavioral phenotypes", Nat. Methods **10**, 877 (2013).

[79]J. Lines, and A. Bagnall, "Time series classification with ensembles of elastic distance measures", Data Min. Knowl. Disc. **29**, 565–592 (2014).

[80]J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, "Learning time-series shapelets", in Proceedings of the 20th acm sigkdd international conference on knowledge discovery and data mining (2014), pp. 392–401.

[81]A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances", Data Min. Knowl. Disc. **31**, 606–660 (2016).

[82]A. Bagnall, J. Lines, W. Vickers, and E. Keogh, *The UEA & UCR time series classification repository*, www.timeseriesclassification.com.

[83]J. G. T. Zañudo, G. Yang, and R. Albert, "Structure-based control of complex networks with nonlinear dynamics", Proc. Natl Acad. Sci. USA **114**, 7234–7239 (2017).

[84]M. Santolini, and A.-L. Barabási, "Predicting perturbation patterns from the topology of biological networks", Proc. Natl Acad. Sci. USA **115**, E6375–E6383 (2018).

[85]K. Sun, J. P. Gonçalves, C. Larminie, and N. Pržulj, "Predicting disease associations via biological network analysis", BMC Bioinformatics **15**, 304 (2014).

[86]A. Calderone, M. Formenti, F. Aprea, M. Papa, L. Alberghina, A. M. Colangelo, and P. Bertolazzi, "Comparing alzheimer's and parkinson's diseases networks using graph communities structure", BMC Syst. Biol. **10**, 25 (2016).

[87]T. A. Schieber, L. Carpi, A. Díaz-Guilera, P. M. Pardalos, C. Masoller, and M. G. Ravetti, "Quantification of network structural dissimilarities", Nat. Commun. **8**, 13928 (2017).

[88]L. Carpi, P. Saco, O. Rosso, and M. Ravetti, "Structural evolution of the tropical pacific climate network", Eur. Phys. J. B **85**, 389 (2012).

[89]I. Barnett, and J.-P. Onnela, "Change point detection in correlation networks", Sci. Rep. **6**, 18893 (2016).

[90]W. Bao, and G. Michailidis, "Core community structure recovery and phase transition detection in temporally evolving networks", Sci. Rep. **8**, 12938 (2018).

[91]Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks", Nature **466**, 761 (2010).

[92]R. F. Betzel, and D. S. Bassett, "Multi-scale brain networks", Neuroimage **160**, 73–83 (2017).

[93]R. E. Boulos, N. Tremblay, A. Arneodo, P. Borgnat, and B. Audit, "Multi-scale structural community organisation of the human genome", BMC Bioinformatics **18**, 209 (2017).

[94]L. d. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas, "Characterization of complex networks: a survey of measurements", Adv. Phys. **56**, 167–242 (2007).

[95]M. E. J Newman, *Networks: an introduction* (Oxford Univ. Press, New York, NY, USA, 2010).

[96]A. Arenas, A. Díaz-Guilera, and C. J. Pérez-Vicente, "Synchronization reveals topological scales in complex networks", Phys. Rev. Lett. **96**, 114102 (2006).

[97]M. Sales-Pardo, R. Guimera, A. A. Moreira, and L. A. N. Amaral, "Extracting the hierarchical organization of complex systems", Proc. Natl Acad. Sci. USA **104**, 15224–15229 (2007).

[98]A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks", New J. Phys. **11**, 033015 (2009).

[99]N. Tremblay, and P. Borgnat, "Graph wavelets for multiscale community mining", IEEE Trans. Signal Process. **62**, 5227–5239 (2014).

[100]M. A. de Menezes, and A.-L. Barabási, "Fluctuations in network dynamics", Phys. Rev. Lett. **92**, 028701 (2004).

[101]A.-L. Barabási, et al., *Network science* (Cambridge Univ. Press, Cambridge, 2016).

[102]A. D. Broido, and A. Clauset, "Scale-free networks are rare", Nat. Commun. **10**, 1017 (2019).

[103]G. H. Golub, and C. F. Van Loan, *Matrix computations*, 4th (JHU Press, Baltimore, MD, USA, 2012).

[104]M. E. J. Newman, and M. Girvan, "Finding and evaluating community structure in networks", Phys. Rev. E **69**, 026113 (2004).

[105]A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms", Phys. Rev. E **78**, 046110 (2008).

[106]A. Lancichinetti, and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities", Phys. Rev. E **80**, 016118 (2009).

[107]D. J. Watts, and S. H. Strogatz, "Collective dynamics of 'small-world' networks", Nature **393**, 440–442 (1998).

[108]P. Erdős, and A. Rényi, "On random graphs I", Publ. Math. **6**, 290–297 (1959).

[109]M. E. J Newman, S. H. Strogatz, and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications", Phys. Rev. E **64**, 026118 (2001).

[110]V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks", J. Stat. Mech.: Theory Exp. **2008**, P10008 (2008).

[111]C. M. Bishop, *Pattern recognition and machine learning* (Springer-Verlag, Berlin, Heidelberg, 2006).

[112]L. C. Freeman, "Centrality in social networks conceptual clarification", Soc. Networks **1**, 215–239 (1978).

[113]V. Latora, and M. Marchiori, "Efficient behavior of small-world networks", Phys. Rev. Lett. **87**, 198701 (2001).

[114]M. E. J Newman, "Assortative mixing in networks", Phys. Rev. Lett. **89**, 208701 (2002).

[115]U. Brandes, "On variants of shortest-path betweenness centrality and their generic computation", Soc. Networks **30**, 136–145 (2008).

[116]M. Sugiyama, M. E. Ghisu, F. Llinares-López, and K. Borgwardt, "Graphkernels: r and python packages for graph comparison", Bioinformatics **34**, 530–532 (2017).

[117]H. Kashima, K. Tsuda, and A. Inokuchi, "Marginalized kernels between labeled graphs", in Proceedings of the 20th international conference on machine learning, edited by T. Fawcett, and N. Mishra, (2003), pp. 321–328.

[118]T. Gärtner, P. Flach, and S. Wrobel, "On graph kernels: hardness results and efficient alternatives", in Learning theory and kernel machines, edited by B. Schölkopf, and M. K. Warmuth, (2003), pp. 129–143.

[119]K. M. Borgwardt, and H.-P. Kriegel, "Shortest-path kernels on graphs", in Proceedings of the 5th ieee international conference on data mining (2005), pp. 74–81.

[120]K. M. Borgwardt, T. Petri, S. Vishwanathan, and H.-P. Kriegel, "An efficient sampling scheme for comparison of large graphs", in Proceedings of the 3rd international workshop on mining and learning with graphs, edited by P. Frasconi, K. Kersting, and K. Tsuda, (2007).

[121]N. Shervashidze, P. Schweitzer, E. J. v. Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler–Lehman graph kernels", J. Mach. Learn. Res. **12**, 2539–2561 (2011).

[122]Q. H. Tran, V. T. Vo, and Y. Hasegawa, *Scale-variant topological information*, `https://github.com/OminiaVincit/scale-variant-topo`, GitHub Repository.

[123]A. K. Debnath, R. L. Lopez de Compadre, G. Debnath, A. J. Shusterman, and C. Hansch, "Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity", J. Med. Chem. **34**, 786–797 (1991).

[124]J. J. Sutherland, L. A. O'brien, and D. F. Weaver, "Spline-fitting with a genetic algorithm: a method for developing classification structure-activity relationships", J. Chem. Inf. Comput. Sci. **43**, 1906–1915 (2003).

[125]J. Kazius, R. McGuire, and R. Bursi, "Derivation and validation of toxicophores for mutagenicity prediction", J. Med. Chem. **48**, 312–320 (2005).

[126]K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels", Bioinformatics **21**, i47–i56 (2005).

[127]N. Wale, I. A. Watson, and G. Karypis, "Comparison of descriptor spaces for chemical compound retrieval and classification", Knowl. Inf. Syst. **14**, 347–375 (2008).

[128]F. Orsini, P. Frasconi, and L. De Raedt, "Graph invariant kernels", in Proceedings of the 24th international joint conference on artificial intelligence, edited by Q. Yang, and M. Wooldridge, (2015), pp. 3756–3762.

[129]P. Yanardag, and S. Vishwanathan, "Deep graph kernels", in Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining, edited by L. Cao, C. Zhang, T. Joachims, G. I. Webb, D. D. Margineantu, and G. Williams, (2015), pp. 1365–1374.

[130]K. Kersting, N. M. Kriege, C. Morris, P. Mutzel, and M. Neumann, *Benchmark data sets for graph kernels*, `http://graphkernels.cs.tu-dortmund.de`, 2016 (accessed November 12, 2018).

[131]C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment", J. Mach. Learn. Res. **13**, 795–828 (2012).

[132]M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs", in Proceedings of the 33rd international conference on international conference on machine learning, Vol. 48 (2016), pp. 2014–2023.

[133]Z. Zhang, M. Wang, Y. Xiang, Y. Huang, and A. Nehorai, "Retgk: graph kernels based on return probabilities of random walks", in Advances in neural information processing systems 31, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, (2018), pp. 3964–3974.

[134]M. N. Arbeitman, E. E. Furlong, F. Imam, E. Johnson, B. H. Null, B. S. Baker, M. A. Krasnow, M. P. Scott, R. W. Davis, and K. P. White, "Gene expression during the life cycle of drosophila melanogaster", Science **297**, 2270–2275 (2002).

[135]J. Zhang, and J. Cao, "Finding common modules in a time-varying network with application to the drosophila melanogaster gene regulation network", J. Am. Stat. Assoc. **112**, 994–1008 (2017).

[136]L. Song, M. Kolar, and E. P. Xing, "Keller: estimating time-varying interactions between genes", Bioinformatics **25**, i128–i136 (2009).

[137]J.-C. Delvenne, S. N. Yaliraki, and M. Barahona, "Stability of graph communities across time scales", Proc. Natl. Acad. Sci. USA **107**, 12755–12760 (2010).

[138]R. Lambiotte, J. Delvenne, and M. Barahona, "Random walks, markov processes and the multiscale modular organization of complex networks", IEEE Trans. Netw. Sci. Eng. **1**, 76–90 (2014).

[139]A.-L. Barabási, and R. Albert, "Emergence of scaling in random networks", Science **286**, 509–512 (1999).

[140]V. De Silva, and G. Carlsson, "Topological estimation using witness complexes", in Proceedings of the 1st eurographics conference on point-based graphics (2004), pp. 157–166.

[141]D. R. Sheehy, "Linear-size approximations to the vietoris–rips filtration", in Proceedings of the 28th annual symposium on computational geometry (2012), pp. 239–248.

[142]U. Bauer, *Ripser: a lean C++ code for the computation of Vietoris–Rips persistence barcodes*, `https://github.com/Ripser/ripser`, 2017.

[143]G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, "Machine learning and the physical sciences", Rev. Mod. Phys. **91**, 045002 (2019).

[144]V. Tejedor, O. Bénichou, and R. Voituriez, "Global mean first-passage times of random walks on complex networks", Phys. Rev. E **80**, 065104(R) (2009).