THE UNIVERSITY *of* EDINBURGH

Metric magnitude and topological methods for machine learning and biomedical data analysis

Rayna Andreeva

Doctor of Philosophy

Institute for Adaptive and Neural Computation
School of Informatics
University of Edinburgh

2025

Abstract

We live in a world which generates vast amounts of data with highly complex structure. Methods based on geometry and topology are suited to analyse the shape of high-dimensional data and thus can provide unique insights. While geometry is concerned with studying distances, topology focuses on connectivity relations. The main advantage of these methods is that they can generate compact summaries of the data to highlight and unravel distinct patterns and relationships. Magnitude is a recently introduced geometric invariant, capable of capturing important properties of the intrinsic geometry of a space. It has potential for applications in machine learning as it can measure a number of geometric quantities such as curvature, volume and diameter. In this thesis, we provide the first applications of magnitude to theoretical deep learning, representation learning and biomedical data analysis. In addition, we compare the geometric insights from magnitude with the topological insights from persistent homology. This thesis contains three parts, the first addresses one of the main difficulties in the application of magnitude, which is the computational cost. To compute magnitude, one needs to invert a matrix, which is an expensive procedure, particularly for large datasets. We provide new faster algorithms for speeding up this computation and approximate magnitude well. These new algorithms enable the applicability of magnitude to data analysis, providing a solid foundation for its wider adoption. The second part examines the intrinsic geometric aspect of machine learning. Here we show the unique uses of magnitude to generalization and the space of latent representations. In the third part, we demonstrate novel biomedical applications of magnitude to the surface of the human tongue and brain artery trees.

Lay Summary

In today's world, we generate large amounts of complex data. To understand this data, we can use methods from geometry (which focuses on distances) and topology (which focuses on connectivity). These methods help us summarize and uncover important patterns in the data. A new geometric concept, called "magnitude," can measure properties like curvature and volume, making it potentially useful for machine learning. This thesis presents the first applications of magnitude to areas such as deep learning and biomedical data analysis. It also compares magnitude's geometric insights with those from a topological method called persistent homology. The thesis is divided into three parts: first, it addresses the high computational cost of calculating magnitude by introducing faster algorithms. These improvements make magnitude more practical for data analysis. The second part explores how magnitude can be used to understand key geometric aspects in machine learning, like how models generalize and the structure of their internal representations. The third part demonstrates how magnitude can be applied in biomedical contexts, such as analysing the surface of the human tongue and brain artery trees.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. Rik Sarkar for his invaluable guidance and support throughout this journey. His insightful feedback helped me refine and sharpen my ideas, pushing me to think critically and approach problems from new perspectives. During moments of doubt and fatigue, his unwavering enthusiasm and encouragement provided the motivation I needed to keep going. His ability to identify weaknesses in my arguments and provide constructive suggestions for improvement was instrumental in shaping the quality of my work. I am deeply appreciative of his dedication, patience, and belief in my potential, all of which have been crucial to the completion of this thesis.

During my PhD, I was fortunate to be involved in several collaborations that provided valuable experiences and deepened my understanding of the academic world. I would like to thanks our collaborators Prof. Jie Gao, Dr. Primož Škraba and James Ward for the paper which Chapter 3 is based on. The experiments in Section 3.5.2.2 and 3.5.2.3 were carried out by James, as well as the proofs of Theorem 3.3.1 and 3.3.2.

I would like to thank our collaborators Benjamin Dupuis, Prof. Tolga Birdal and Prof. Umut Şimşekli for the collaboration which Chapter 5 is based on. The generalization bound proofs in 5.3.2 and 5.3.3 were carried out by Ben.

I would like to thank Prof. Bastian Rieck and Katharina Limbeck for our collaboration, presented in Chapter 6. Experiments in Section 6.4.2, 6.4.3 and 6.4.4 were carried out by Katharina.

On a more personal level, Ben, thank you for showing me the beauty of writing out definitions and attempting proofs on the white board, and for exchanging cool mathematical ideas. I am glad that you picked up my magnitude enthusiasm and carried on with it. Katharina, thank you for fuelling my enthusiasm and believing in my ideas when I didn't. And for never giving up even when it was hard. We made it to NeurIPS! James, thank you for your hard work, and for choosing to work with magnitude in the first place.

I would like to thank all members of the ANGLE research group for all our group meetings and stimulating intellectual discussion. In particular, to Dr. Lauren Watson for our many PhD, code and research discussion which have helped me a lot during Covid and after. I sincerely appreciate all members of

the CDT in Biomedical AI, and Domas, Matúš, Michael and Nikitas from my cohort in particular – for the board games nights and keeping me sane during Covid. Further, I would like to thank the numerious people who I met at a number of conferences (Oxford, Bedlewo, Rome, Kyoto), for keeping my enthusiasm for doing research and TDA. Additionally, I would like to thank Nina Otter, our collaboration has been one of the best things happening to me, thank you for choosing me to work on your WinCompTop3 project! Your advice and wisdom has helped me understand better the academic system, and your personal stories have been of great motivation. Thank you for showing me the beauty of doing mathematics with pen and paper.

I would like to thank my mum who has always supported me. She is my biggest critique and always reads my drafts first, even though her specialisation is far from what I do; for listening to my endless complaining when things were not going so well, and even when things were well; for pushing me to complete this work; and for always being there for me, and only ever wanting me to be happy and not too ambitious.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Rayna Andreeva*)

Contents

Chapter 1

Introduction

Modern Artificial Intelligence (AI) relies heavily on machine learning (ML) models. These models are often intricate and large, having millions or possibly billions of parameters. They are capable of producing sophisticated, even human-like, responses [Ma et al., 2023]. In natural language processing (NLP), advanced models like GPT-3 can generate coherent text and engage in conversation [Ray, 2023]. The input data that modern AI models operate on is usually high-dimensional and complex, for example biomedical data. In order to ensure that AI systems are reliable and efficient it is necessary to better understand the models, data and the interactions between them. One approach suggested in order to further such understanding is through the geometry and topology of relevant point clouds [Snášel et al., 2017]. The ability to view these models through the lens of these two fields of mathematics provides a unique analytical perspective by focusing on shape, structure, and relationships between data points.

There has been growing interest in the role that geometry and topology play in machine learning and data science [Carlsson, 2009, Adams and Moy, 2021]. This has culminated in the development of Topological Machine Learning (TML), combining ideas from Topological Data Analysis (TDA) and ML, with Persistent Homology (PH) emerging as one of the primary tools of the field. There are two particular cases where one would want to combine topological methods with traditional ML methods: (1) when there is interest in a quantitative compact summary of the global features or local geometry in a data, or (2) to explore if local geometry or global topology may be discriminatory for the ML task at hand [Adams and Moy, 2021].

Magnitude is a relatively new shape descriptor which has emerged as a prom-

ising tool with broad applications to ML and data science. Magnitude has several desirable characteristics: (i) it provides a compact summary of a space, (ii) it is easier to understand conceptually than PH as its definition for finite spaces is straightforward, (iii) it has an intuitive interpretation as the effective number of points in a space, and (iv) can measure curvature and fractal dimension, similar to PH.

The central theme of this thesis is to gain insight into the underlying geometry of traditionally important ML spaces via magnitude. We achieve this by developing novel applications of magnitude in ML and biomedical data analysis. First, we describe a number of computational challenges which have prevented the wider adoption of magnitude for ML, and then we propose solutions. As a result we enable the use of magnitude for data science. The work in this thesis falls under 3 categories:

1. Dealing with the computational challenges presented by magnitude.
2. Using magnitude to analyse ML algorithms and latent space representations.
3. Computing features based on magnitude and PH, and demonstrating their utilities in biomedical applications.

The theory of PH is well developed, it has found numerous applications in data science more broadly [Giunti et al., 2022]. There has been significant theoretical research on magnitude and its relationship with persistent homology [Otter, 2021]. However, several unanswered questions persist, particularly regarding its practical implications in machine learning and biomedical data analysis. Recent studies [Bunch et al., 2021, Adamer et al., 2021] have started to establish links between quantities derived from magnitude, called magnitude vectors, and ML applications, but this area of research is still in its very early stages. The strengths and weaknesses which magnitude might possess over persistent homology are not clear *a priori* for practical purposes and there have been no comparative studies to date. Therefore, apart from paving the way for magnitude applications, we also provide comparisons where appropriate.

In terms of interpretation, PH and magnitude are very different: the former is a topological invariant that counts the number of connected components and holes, while the latter is an isometric invariant that captures the effective number of points in a space. One similarity is that both of them are capable of measuring

curvature. Another similarity is their ability to capture intrinsic dimensionality: under some mild conditions, it can be proven that both PH dimension and magnitude dimension measure the Minkowski dimension of a space.

This thesis is split into three parts. Here we briefly outline the problems which each chapter is concerned with. In Section 1.4, we describe how to address these problems.

1.1 Computational challenges of magnitude

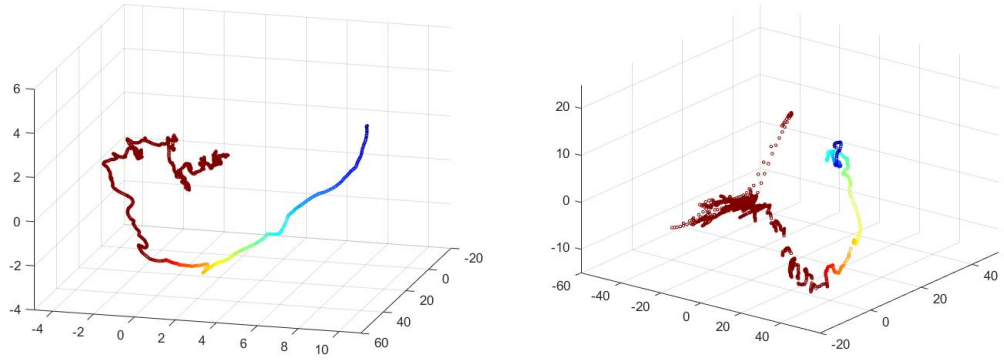
Wider application of magnitude is limited by the computation cost. For a set of n points, the standard method of computing magnitude requires inverting an $n \times n$ matrix. The best known lower bound for matrix multiplication and inversion is $\Omega(n^2 \log n)$ [Raz, 2002]; the commonly used Strassen's algorithm [Strassen, 1969] has complexity $O(n^{2.81})$. By definition the magnitude computation requires consideration of all pairs of input points. This is costly for large datasets.

Developing methods which compute magnitude faster at the cost of accuracy are important for further usability in data analysis. This has significant ramifications in clustering, neural network regularization and generalization.

In a set of n points, there are some which are more important in the computation of magnitude than others. In the terminology of magnitude, every point is assigned a weight and the sum of the weights gives the total magnitude. A point with higher weight can be considered to be more important than a point with a lower weight for approximating the value of magnitude. Hence, developing methods for selecting the points which contribute the most would result in a sensible approximation. Creating a systematic approach towards choosing such subsets of data points leads to another way to avoid the computational bottleneck of magnitude.

1.2 Two aspects of analysing ML models

A deeper understanding of ML models can be achieved in a number of ways. It can be done during the multiple stages of learning that occur: by considering the training stage (where one can look at the model space) or the internal representations (latent space). Both the training stage and the internal representations can be conceptualised as (usually high-dimensional) metric spaces. In this view, each



(a) (b)

Figure 1.1: **Training trajectories patterns after applying Multidimensional scaling (MDS).** The trajectory in plot (a) corresponds to a neural network with lower generalization gap, while the training trajectory in plot (b) depicts a neural network with larger generalization gap. There is a clear difference between these patterns, which can be quantified using fractal dimensions (Chapter 3) and (positive) magnitude and α-weighted lifetime sums, which is a concept from PH (Chapter 4).

data instance or internal representation vector is a 'point', and the finite collection of these points, which we work with in practice, constitutes a point cloud. We use 'point cloud' here to specifically denote this concrete, finite sample of a larger, potentially continuous, metric space. These two spaces are equipped with rich intrinsic geometric and topological structure. Providing methods for the analysis of such point clouds is essential for deriving deep insights about these models and the various spaces they inhabit such as generalizability and understanding the outputs of generative models. Failure to generate such analysis can lead to failure to understand why a neural network does not perform well on unseen data, or why a generative model does not produce desirable outputs.

In general, model spaces and latent spaces both involve high-dimensional mathematical representations that enable a model to learn from and generalize across data. While model spaces refer to the parameters that define the function or architecture of the model itself, latent spaces represent hidden or transformed data representations that capture essential features or structures of the input. Both types of spaces play central roles in the learning process, from optimizing

for performance to improving inference and generalization.

Model space. The model space is where the optimisation algorithm takes consecutive steps in order to reach its goal and minimise the loss function. It contains important information about the performance of the model. The recorded steps in the model space are called optimization trajectories, and they leave traces which contain particular patterns. More formally, in the case of neural networks, the hypothesis class is denoted by $\mathcal{W} \subset \mathbb{R}^d$, and each $w \in \mathcal{W}$ is a parameter vector. We call the set of all hypothesis classes returned from the optimisation procedure of a given training algorithm $\mathcal{A}$ for a given training data S the *optimisation trajectories*, and denote them by $\mathcal{W}$. To access the iterates at every step of the process, we denote by w_i an element of $\mathcal{W}$ at iteration i. More concretely, $\mathcal{W} := \{w \in \mathbb{R}^d : \exists i \in [0, I], w = [\mathcal{A}(S)_i]\}$, where I is the number of training iterations. In other words, when we fix i, we are interested in the weights at iteration i, returned by the optimisation algorithm $\mathcal{A}$.

In Figure 1.1 we can see the optimization training trajectories of a model with a small generalisation gap on the left and a model with a larger generalisation gap on the right. It is clear that the point patterns are different. Quantifying their geometry and topology has an important role for generalization (Chapter 3, 4). Taking into account the above, two key questions arise:

1. It has been previously shown that these trajectories possess fractal structure [Simsekli et al., 2020], and hence that the local geometry matters. Can we quantify their shape using magnitude and magnitude dimension? Would that lead to additional advantages? (Chapter 3)

2. The fractal dimension quantities proposed in the literature work in the infinite regime. What about developing more practical complexities, which bring closer the theory and practice? Would they scale to practically relevant architectures like Vision Transformers? What about other domains like Graph Neural Networks (GNNs)? (Chapter 4)

Latent space. Measuring the intrinsic diversity of latent representations is of utmost importance for representation learning. Determining the strength to which the outputs generated by a model resemble the properties of the input distribution is crucial for preventing common problems like mode collapse and mode dropping. The existing diversity metrics in the literature suffer from a number of limitations

both in the reference-free case and where a reference distribution is available. Chapter 5 then answers the following questions: (i) Can we define a theoretically justified diversity measure based on magnitude that works across a number of different modalities such as text, image and graph data? (ii) Is this measure stable? (iii) Can we use magnitude to measure curvature and validate the theoretical motivation that magnitude encodes geometry?

1.3 Magnitude and TDA for biomedical data

After we have fully explored applications of magnitude to ML, it is natural to look at biomedical data, where topological and geometrical considerations are important due to inherent geometric and topological structure. In a biomedical context, determining the locations of certain structures and quantifying the patterns which they form might be associated with certain information about the person, such as age. Applying magnitude and TDA to the surface of the human tongue and brain artery trees is a non-trivial task. It is not known if tongue prints are unique and if they can be used to identify a person's age, gender or identity. Further, a principled way to quantify the quantitative information extracted from the shape of papillae, which are tiny projections covering the surface of the tongue has not been established, and both geometry and topology are suitable for this analysis. More broadly, we are interested in developing geometric and topological features based on the shapes of papillae, and using them as predictors of age, gender and individual (Chapter 7, 1st part). The distribution of points on the brain artery trees might be linked with age (Chapter 7, 2nd part). In the next sections, we provide more information about each biomedical problem.

Magnitude and TDA for the surface of the human tongue. The tongue is a highly sophisticated anatomical structure and its operation is fundamental to speech, friction regulation and oral processing of food. The papillae on the surface of the tongue enable perception of taste, texture and oral mechanics. Of these numerous anatomical structures, *fungiform papillae* are linked to taste perception as they house the taste buds [Miller Jr and Reedy Jr, 1990], whereas *filiform papillae* that are devoid of taste buds are thought to be crucial for textural perception.

The intricate geometry of the tongue at a microscopic scale can be observed in 3D scans. Although there has been significant research on the importance of

papillae density, our understanding of the papillae shapes and surface properties of the tongue suffers from the difficulty of extracting and analysing geometry of papillae at microscopic scales. ML has recently emerged as a powerful technique for diagnosis where large volumes of medical data or images are available [Cai et al., 2020]. These approaches have largely focused on computing global functions such as a medical diagnosis from an image. However, to date there is no ML model that has classified microscopic tongue papillae based on 3D tongue scans.

Magnitude and TDA for brain artery trees. Changes in the network of blood vessels, also known as vasculature, are often the first signs of development of diseases like Alzheimer's or stroke. If we are able to develop methods aimed at identifying these alterations, we will be better equipped to treat these conditions early and to develop preventative therapies. With ageing, brain vasculature changes and it is important to be able to recognise and quantify such changes. Previous studies have demonstrated the usefulness of topology for the age detection problem [Bendich et al., 2016]. Brain vasculature has been found to be correlated with age from two different methods of analysis — statistical and TDA. Methods from the former have found age to be correlated with total artery length [Gutierrez et al., 2016]. On the other hand, TDA has been useful in identifying correlation between age and the positions of arteries in space in a way that statistical analysis is not capable of discovering [Bendich et al., 2016]. Magnitude has not been applied to this problem, therefore it is of interest to establish how useful it might be in the task of predicting age from brain artery trees. In addition, previous approaches have not utilised ML.

1.4 Contributions

We now summarise the key contributions of this thesis to the topics of ML and biomedical data analysis.

Chapter 3, Approximating Metric Magnitude of Point Sets. We study the magnitude computation problem, and show efficient ways of approximating it. We show that it can be cast as a convex optimization problem, but not as a submodular optimization problem. The chapter describes two new algorithms – an iterative approximation algorithm that converges fast and is accurate, and a subset selection method that makes the computation even faster. It has been previously proposed that magnitude of model sequences generated during stochastic gradient

descent is correlated to the generalization gap. Extension of this result using our more scalable algorithms shows that longer sequences in fact bear higher correlations. We also describe new applications of magnitude in ML – as an effective regularizer for neural network training, and as a novel clustering criterion. This chapter is based on the preprint

- *Approximating Metric Magnitude of Point Sets* [Andreeva et al., 2025], which has been published at **AAAI 2025**.

Chapter 4, Metric Space Magnitude and Generalisation in Neural Networks. We study the problem of generalization in neural networks and propose quantifying the learning process of deep neural networks through the lens of magnitude. Moreover, we theoretically connect magnitude dimension and the generalisation error, and demonstrate experimentally that the proposed framework can be a good indicator of the latter. This chapter is based on the preprint

- *Metric Space Magnitude and Generalisation in Neural Networks* [Andreeva et al., 2023a], published in the Proceedings of Machine Learning Research as part of 2nd TAG (Topology, Algebra, Geometry) **ICML workshop 2023**.

Chapter 5, Topological Generalization Bounds for Discrete-Time Stochastic Optimization Algorithms. Here we provide the first theoretically justified link between the generalization error and a quantity derived from magnitude, called positive magnitude. We further prove that the α-weighted lifetime sum, which is a topological quantity known as total persistence for $\alpha = 1$, is also associated with generalization. We call these newly established generalization measures topological complexities, and demonstrate that they are computationally friendly and flexible. Our experimental results demonstrate that our new complexity measures correlate highly with generalization error in industry-standards architectures such as transformers and deep graph networks. Our approach consistently outperforms existing topological bounds across a wide range of datasets, models, and optimizers, highlighting the practical relevance and effectiveness of our complexity measures. This chapter is based on the paper

- *Topological Generalization Bounds for Discrete-Time Stochastic Optimization Algorithms* [Andreeva et al., 2024], which has been published at **NeurIPS 2024**.

Chapter 6, Metric Space Magnitude for Evaluating the Diversity of Latent Representations. We develop a family of magnitude-based measures of the intrinsic diversity of latent representations, developing a novel notion of dissimilarity between magnitude functions of finite metric spaces. Our measures are provably stable under perturbations of the data, can be efficiently calculated, and enable a rigorous multi-scale characterisation and comparison of latent representations. We show their utility and superior performance across different domains and tasks, including (i) the automated estimation of diversity, (ii) the detection of mode collapse, and (iii) the evaluation of generative models for text, image, and graph data. This chapter is based on the paper

- *Metric Space Magnitude for Evaluating the Diversity of Latent Representations* [Limbeck et al., 2024], which has been published at **NeurIPS 2024**.

Chapter 7, Magnitude and TDA for biomedical data: the tongue and the brain. We first present the first ML framework on 3D microscopic scans of human papillae (n = 2092), uncovering the uniqueness of geometric and topological features of papillae. The finer differences in shapes of papillae are investigated computationally based on a number of features derived from discrete differential geometry, magnitude and computational topology. Interpretable ML techniques show that persistent homology features of the papillae shape are the most effective in predicting the biological variables. Models trained on these features with small volumes of data samples predict the type of papillae with an accuracy of 85%. The papillae type classification models can map the spatial arrangement of filiform and fungiform papillae on a surface. Remarkably, the papillae are found to be distinctive across individuals and an individual can be identified with an accuracy of 48% among the 15 participants from a single papillae, increasing to 50% when we add magnitude-based features, indicating that the magnitude of these structures varies between people. Collectively, this is the first evidence demonstrating that tongue papillae can serve as a unique identifier inspiring new research direction for food preferences and oral diagnostics. We further demonstrate the utility of magnitude and TDA for analysing brain artery trees, and demonstrate the topological and geometric features are capable of detecting age. This chapter is partially based on the paper

- *Machine learning and Topological data analysis identify unique features*

of human papillae in 3D scans [Andreeva et al., 2023b], which has been published in **Nature Scientific Reports**.

We conclude by proposing future directions of research inspired by the results of this thesis.

Chapter 2

Background

2.1 Metric Magnitude

While the magnitude of metric spaces is a general concept [Leinster, 2013], we restrict our focus to subsets of $\mathbb{R}^n$, where magnitude is known to exist [Meckes, 2013].

For a finite metric space (X, d) with distance function d, we define the similarity matrix $\zeta_{ij} = e^{-d_{ij}}$ for $i, j \in X$. The metric magnitude $\mathrm{Mag}(X, d)$ is defined [Leinster, 2013] in terms of an inverse or a *weighting*. In this section we present both definitions as they are important in the different subsequent chapters.

2.1.1 Metric Magnitude as an inverse

Definition 2.1.1. Let X be a metric space with similarity matrix ζ_{ij}. If ζ_{ij} is invertible, magnitude is defined as

$$\mathrm{Mag}(X) = \sum_{ij} (\zeta^{-1})_{ij}. \tag{i}$$

When X is a finite subset of $\mathbb{R}^n$, then ζ_{ij} is a symmetric positive definite matrix as proven in Leinster [2013] (Theorem 2.5.3). Then, $(\zeta^{-1})_{ij}$ exists, and hence magnitude exists as well.

Magnitude is best illustrated when considering a few sample spaces with a small number of points.

Example 2.1.2. Let X denote the metric space with a single point a. Then, ζ_X is a 1×1 matrix with $\zeta_X^{-1} = 1$ and using the formula for magnitude, we get $\mathrm{Mag}_X = 1$.

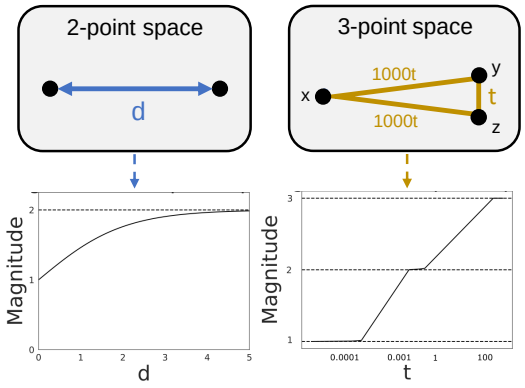


Figure 2.1: **Magnitude of the 2- and 3-point space.** On the left, we see the 2-point space, where the distance between the points is d. On the right we see the 3-point space for an isosceles triangle with distance t between y and z, and $1000t$ between x and y and x and z. Below each space, we see the respective magnitude function.

Example 2.1.3. A more illustrative example is given by the space of two points. Let $X = \{a, b\}$ be a finite metric space where $d_X(a, b) = d$. Then

$$\zeta_X = \begin{bmatrix} 1 & e^{-d} \\ e^{-d} & 1 \end{bmatrix}, \tag{ii}$$

so that

$$\zeta_X^{-1} = \frac{1}{1 - e^{-2d}} \begin{bmatrix} 1 & -e^{-d} \\ -e^{-d} & 1 \end{bmatrix}, \tag{iii}$$

and therefore

$$\mathrm{Mag}(X) = \frac{2 - 2e^{-d}}{1 - e^{-2d}} = \frac{2}{1 + e^{-d}}. \tag{iv}$$

This example is also illustrated in Figure 2.1. Using Eq. (iv), if d is very small, i.e. $d \to 0$, $\mathrm{Mag}(X) \to 1$. Similarly, when $d \to \infty$, $\mathrm{Mag}(X) \to 2$. In practice, as it can be seen from Figure 2.1, $\mathrm{Mag}(X) = 2$ for a value of d as small as 5.

2.1.2 Metric Magnitude as a weighting

For a finite metric space (X, d) with distance function d, we define the similarity matrix $\zeta_{ij} = e^{-d_{ij}}$ for $i, j \in X$. The metric magnitude $\mathrm{Mag}(X, d)$ is defined [Leinster, 2013] in terms of a *weighting* as follows.

Definition 2.1.4 (Weighting w)**.** A weighting of (X, d) is a function $w : X \to \mathbb{R}$ such that $\forall i \in X, \sum_{j \in X} \zeta_{ij} w(j) = 1$.

We refer to the $w(i)$ as the magnitude weight of i, and interchangeably write it as w_i.

Definition 2.1.5 (Metric Magnitude $\mathrm{Mag}(X, d)$)**.** The magnitude of (X, d) is defined as $\mathrm{Mag}(X, d) = \sum_{i \in X} w(i)$, where w is a weighting of (X, d).

2.1.3 Scaling and the Magnitude Function

More information about a metric space can be obtained by looking at its rescaled counterparts. The resulting representation is richer, and is called the magnitude function, which we describe next.

For each value of a parameter $t \in \mathbb{R}^+$, we consider the space where the distances between points are scaled by t, often written as tX.

Definition 2.1.6 (Scaling and tX)**.** Let (X, d) be a finite metric space. We define (tX, d_t) to be the metric space with the same points as X and the metric $d_t(x, y) = td(x, y)$.

Definition 2.1.7 (Magnitude function)**.** The magnitude function of a finite metric space (X, d) is the function $t \mapsto \mathrm{Mag}(tX)$, which is defined for all $t \in (0, \infty)$.

Example 2.1.8. Consider the magnitude function of the 3-point space in Figure 2.1. In this example, the points x, y and z form an isosceles triangle. When $t = 0.0001$, all the three points are very close to each other and almost indistinguishable. Hence, we say that the space has 1 effective point. In contrast, when $t = 0.01$, the distance between the two points on the right y and z is very small, and x is quite far. Therefore, we say that the space looks like two points. Finally, when t is large, all the three points are far away from each other, and the value of magnitude is 3, which is also the cardinality of the space.

The Magnitude Function is important in computing magnitude dimension, which is a determined by growth rate of $\mathrm{Mag}(tX)$ with respect to t. It is a quantity similar to fractal dimension and useful in predicting generalization of models computed via gradient descent, as we will demonstrate in Chapter 4.

2.1.4 Magnitude Dimension

There are various notions that can be used to measure the intrinsic dimension of a space. One of them is the magnitude dimension.

Definition 2.1.9 (Magnitude dimension)**.** When

$$\dim_{\mathrm{Mag}} X = \lim_{t \to \infty} \frac{\log(\mathrm{Mag}(tX))}{\log t} \tag{v}$$

exists, we define this to be the magnitude dimension of X [Meckes, 2015].

2.2 Evaluation criteria for measuring generalization

In this section we explain the evaluation measured used to quantify generalization. Pearson's r correlation coefficient is highly sensitive to outliers and assumes that the data are approximately normally distributed. Extreme values can disproportionately influence the coefficient. As a non-parametric, rank-based statistic, Kendall's τ is much more robust to outliers and doesn't assume a specific distribution. It's less influenced by extreme values because it only considers their rank, not their magnitude.

Real-world data is often noisy, contains outliers, and rarely follows perfect normal distributions. A model that truly generalizes should be robust to these real-world imperfections. Using Kendall's τ helps assess this robustness in the face of such data characteristics.

While Pearson's correlation is valuable for detecting linear relationships and is a foundational statistical tool, the average granulated Kendall coefficients offer a more nuanced and often more appropriate measure of generalization in machine learning contexts because they focus on monotonic/ordinal relationships, which are often more relevant to how complex models generalize than strict linearity; are robust to outliers and non-normal data, reflecting the realities of real-world datasets; and are highly sensitive to the stability of relative orderings under small perturbations or subsampling, which is a key indicator of a model's ability to learn robust, fundamental patterns rather than memorizing noise or brittle correlations.

In further chapters, we assess the correlation between our complexities and the generalization error by using the granulated Kendall's coefficients (GKC) [Jiang et al., 2019]. While the classical Kendall's coefficients (KC) [Kendall, 1938a] (denoted τ) measures the correlation between two quantities, it may fail to capture their causal relationship. Instead, one "granulated" coefficient is defined in [Jiang et al., 2019] for each hyperparameter (*i.e.*, $\boldsymbol{\psi}_{\mathrm{LR}}$ for η and $\boldsymbol{\psi}_{\mathrm{BS}}$ for b); it measures

the correlation when only this hyperparameter is varying. We also compute the averaged GKC, $\boldsymbol{\Psi} := (\boldsymbol{\psi}_{\mathrm{LR}} + \boldsymbol{\psi}_{\mathrm{BS}})/2$. Here we provide exact definitions of the correlation coefficients and further reasons why they might be more appropriate. For more details, please consult [Jiang et al., 2019].

2.2.1 Kendall's Rank-Correlation Coefficient

To assess a complexity measure's μ quality, one can use ranking. Given a set of models trained with hyperparameters in the set Θ, their associated generalization gap $\{g(\theta) \mid \theta \in \Theta\}$, and their respective values of the measure $\{\mu(\theta) \mid \theta \in \Theta\}$, our aim is to see how consistent the measure (e.g., L_2 norm of network weights) is with the empirically observed generalization. To this end, we construct a set T, where each element of the set is associated with one of the trained models. Each element has the form of a pair: complexity measure μ versus generalization gap g.

$$T = \bigcup_{\theta \in \Theta} \{\mu(\theta), g(\theta)\}. \quad (1)$$

With an ideal complexity measure, the following must hold true for any pair of trained models: if $\mu(\theta_1) > \mu(\theta_2)$, then $g(\theta_1) > g(\theta_2)$ also holds. We use Kendall's rank coefficient τ [Kendall, 1938b] to quantify the degree of consistency amongst the elements of T.

$$\tau(T) = \frac{1}{|T|(|T|-1)} \sum_{(\mu_1, g_1) \in T} \sum_{(\mu_2, g_2) \in T \setminus \{(\mu_1, g_1)\}} \mathrm{sign}(\mu_1 - \mu_2)\mathrm{sign}(g_1 - g_2). \quad (2)$$

τ ranges from 1 to -1. A value of 1 signifies perfect agreement between the two rankings, while -1 indicates perfect disagreement (one ranking is the exact reverse of the other). If complexity and generalization are independent, τ will be zero.

2.2.2 Granulated Kendall's Coefficient

While Kendall's correlation coefficient is an effective tool widely used to capture the relationship between two rankings of a set of objects, certain measures can achieve high τ values in a trivial manner—i.e., the measure may strongly correlate with the generalization performance without necessarily capturing the cause of generalization [Jiang et al., 2019]. To mitigate the effect of spurious correlations, a new quantity for reflecting the correlation between measures and generalization based on a more controlled setting has been introduced [Jiang et al., 2019].

None of the existing complexity measures are perfect. However, they might have different sensitivity and accuracy with respect to different hyperparameters. For example, sharpness may do better than other measures when only a certain hyperparameter (say batch size) changes. To understand such details, in addition to τ (T), we compute τ for consistency within each hyperparameter axis Θ_i, and then average the coefficient across the remaining hyperparameter space. Formally, we define:

$$m_i = |\Theta_1 \times \cdots \times \Theta_{i-1} \times \Theta_{i+1} \times \cdots \times \Theta_n| \tag{vi}$$

$$\psi_i = \frac{1}{m_i} \sum_{\theta_1 \in \Theta_1} \cdots \sum_{\theta_{i-1} \in \Theta_{i-1}} \sum_{\theta_{i+1} \in \Theta_{i+1}} \cdots \sum_{\theta_n \in \Theta_n} \tau \left(\bigcup_{\theta_i \in \Theta_i} \{\mu(\theta), g(\theta)\} \right) \tag{vii}$$

The inner τ reflects the ranking correlation between the generalization and the complexity measure for a small group of models where the only difference among them is the variation along a single hyperparameter θ_i. We then average this value across all combinations of the other hyperparameter axes. Intuitively, if a measure is good at predicting the effect of hyperparameter θ_i over the model distribution, then its corresponding ψ_i should be high. Finally, we compute the average ψ_i across all hyperparameter axes, and name it $\mathbf{\Psi}$:

$$\mathbf{\Psi} = \frac{1}{n} \sum_{i=1}^{n} \psi_i \tag{viii}$$

If a measure achieves a high $\mathbf{\Psi}$ on a given hyperparameter distribution Θ, then it should also achieve high individual ψ_i values across all hyperparameters. A complexity measure that excels at predicting changes for a single hyperparameter (high ψ_i) but fails for others (low ψ_j for $j \neq i$) will not yield a high $\mathbf{\Psi}$. Conversely, a high $\mathbf{\Psi}$ indicates that the measure can reliably rank the generalization for changes across each hyperparameter.

To illustrate why $\mathbf{\Psi}$ better captures the causal nature of generalization than Kendall's τ, consider a thought experiment: Suppose a measure perfectly captures the network's depth but produces random predictions when two networks share the same depth. Such a measure would perform reasonably well in terms of τ, but significantly worse in terms of $\mathbf{\Psi}$. In experiments conducted in [Jiang et al., 2019], the authors found that such a measure would yield an overall $\tau = 0.362$ but a $\mathbf{\Psi} = 0.11$.

This measure represents only a small step towards the challenging problem of empirically capturing the causal relationship between complexity measures and generalization. In [Jiang et al., 2019], the authors further propose Conditional Independence Tests, which we have not computed in this thesis, but it will be important future work.

Chapter 3

Approximating Metric Magnitude of Point Sets

Metric magnitude of a point cloud is a measure of its "size." It has been adapted to various mathematical contexts and recent work suggests that it can enhance machine learning and optimization algorithms. But its usability is limited due to the computational cost when the dataset is large or when the computation must be carried out repeatedly (e.g. in model training). In this paper, we study the magnitude computation problem, and show efficient ways of approximating it. We show that it can be cast as a convex optimization problem, but not as a submodular optimization. The paper describes two new algorithms – an iterative approximation algorithm that converges fast and is accurate in practice, and a subset selection method that makes the computation even faster. It has previously been proposed that the magnitude of model sequences generated during stochastic gradient descent is correlated to the generalization gap. Extension of this result using our more scalable algorithms shows that longer sequences bear higher correlations. We also describe new applications of magnitude in machine learning – as an effective regularizer for neural network training, and as a novel clustering criterion.

3.1 Introduction

Magnitude is a relatively new isometric invariant of metric spaces. It was introduced to characterize ecology and biodiversity data, and was initially defined as an Euler Characteristic of certain finite categories [Leinster, 2008]. Similar to

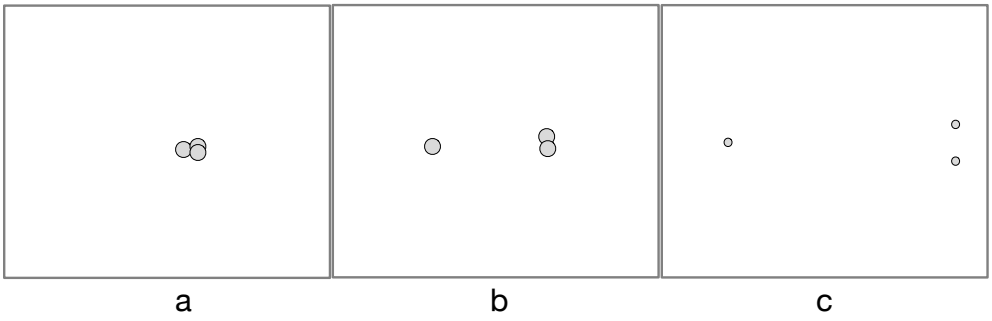


Figure 3.1: Consider the magnitude function of a 3-point space, visualized above at different scales. (a) For a small value of the scale parameter (e.g. $t = 0.0001$), all the three points are very close to each other and appear as a single unit. This space has magnitude close to 1. (b) At $t = 0.01$ the distance between the two points on the right is still small and they are clustered together, and the third point is farther away. This space has Magnitude close to 2 (c) When t is large, all the three points are distinct and far apart, and Magnitude is 3.

quantities such as the cardinality of a point set, the dimension of vector spaces and Euler characteristic of topological spaces, Magnitude can be seen as measuring the "effective size" of mathematical objects. See Figure 3.1 for an intuition of Magnitude of Euclidean points. It has been defined, adapted and studied in many different contexts such as topology, finite metric spaces, compact metric spaces, graphs, and machine learning [Leinster, 2013, Leinster and Willerton, 2013, Barceló and Carbery, 2018, Leinster, 2019, Leinster and Shulman, 2021, Kaneta and Yoshinaga, 2021, Giusti and Menara, 2024].

In machine learning and data sciences, the magnitude of a point cloud can provide useful information about the structure of data. It has recently been applied to study the boundary of a metric space [Bunch et al., 2021], edge detection for images [Adamer et al., 2024], diversity (Chapter 6) and dimension (Chapter 4) of sets of points in Euclidean space, with applications in data analysis as well as generalization of models (Chapter 5). Wider applications of magnitude are limited by the computation cost. For a set of n points, the standard method of computing Magnitude requires inverting an $n \times n$ matrix. The best known lower bound for matrix multiplication and inversion is $\Omega(n^2 \log n)$ [Raz, 2002]; the commonly used Strassen's algorithm [Strassen, 1969] has complexity $O(n^{2.81})$[1].

[1]Faster algorithms for matrix inversion exist, such as the Coppersmith-Winograd algorithm [Coppersmith and Winograd, 1990] with running time $O(n^{2.376})$ and optimized CW-like algorithms with the best running time $O(n^{2.371552})$ [Williams et al., 2024].

By definition, Magnitude computation requires consideration of all pairs of input points, making it expensive for large datasets.

Our contributions. In this chapter, we take the approach that for many scenarios in data sciences, an approximate yet fast estimate of magnitude is useful, particularly in real-world modern applications where datasets and models are large and noisy and often require repeated computation.

Given a point set $X \subset \mathbb{R}^D$, we first show (Section 3.3.1) that computing the magnitude $\mathrm{Mag}(X)$ can be formulated as finding the minimum of a convex function, and so can be approximated using suitable gradient descent methods. We then define a new algorithm that iteratively updates a set of weights called the Magnitude weighting to converge to the true answer (Section 3.3.2). This method converges quickly and is faster than matrix inversion or gradient descent.

While avoiding inversion, both these methods need $n \times n$ matrices to store and use all pairs of similarities between points. To improve upon this setup, we take an approach of selecting a smaller subset $S \subset X$ of representative points so that $\mathrm{Mag}(S)$ approximates $\mathrm{Mag}(X)$. We first prove that Magnitude is not a submodular function, that is, if we successively add points to S, $\mathrm{Mag}(S)$ does not satisfy the relevant diminishing returns property. In fact, for arbitrarily high dimension D, the increase in $\mathrm{Mag}(S)$ can be arbitrarily large with the addition of a single point. Though in the special case of $D = 1$, $\mathrm{Mag}(S)$ is in fact submodular, and the standard greedy algorithm [Nemhauser et al., 1978] for submodular maximization can guarantee an approximation of $(1 - 1/e)$ (Section 3.3.3). In practice, the greedy algorithm is found to produce accurate approximations on all empirical datasets – both real-world ones and synthetic ones. This algorithm adds points to S one by one; in each step it iterates over all remaining points to find the one that maximizes $\mathrm{Mag}(S)$. These magnitude computations are faster due to the smaller size of S, but the costs add up as they are repeated over X.

Section 3.3.4 describes an approach to speed up the approximations further. It uses properties of Magnitude such as monotonicity and growth with scale, to develop a selection method – called Discrete centers – that does not require repeated computation of Magnitude. It is particularly useful for computing the *Magnitude Function* – which is magnitude as a function of scale, and useful in dimension computation (Chapter 4). This method can also easily adapt to dynamic datasets where points are added or removed. Faster estimates of magnitude allows new applications of Magnitude in machine learning. Section 3.4 describes the

use of Magnitude as a regularizer for neural network, and a clustering algorithm similar to density based clustering methods, using Magnitude as a clustering criterion.

Experiments in Section 3.5 show that the approximation methods are fast and accurate. Iterative Normalization outperforms inversion for larger dataset sizes and converges fast; for the subset selection algorithms, Discrete centers approximates the Greedy Maximization approach empirically at a fraction of the computational cost. The more scalable computation allows us to produce new results in the topic of generalization, where we extend prior work on computing topological complexities based on magnitude (Chapter 5) to a larger number of training trajectories, extending from 5×10^3 due to computational limitations to 10^4, and observe that the correlation coefficients average Granulated Kendall ($\mathbf{\Psi}$) and Kendall tau (τ) improve significantly with the increased number of trajectories. The new regularization and clustering methods based on Magnitude are also shown to be effective in practice.

Related work and discussion can be found in Sec. 3.6.

3.2 Technical Background

This section introduces the definitions needed for the rest of the chapter.

3.2.1 Submodular Functions and Maximization Algorithm

The notion of submodularity is inspired by diminishing returns observed in many real world problems.

Definition 3.2.1 (Submodular Function)**.** Given a set V, a function $f : 2^V \to \mathbb{R}$ is a submodular set function if:

$$\forall S, T \subseteq V, f(S) + f(T) \geq f(S \cup T) + f(S \cap T).$$

The definition implies that marginal utility of items or subsets are smaller when they are added to larger sets. An example is with sensor or security camera coverage, where the marginal utility of a new camera is smaller then its own coverage area as its coverage overlaps with existing cameras.

The submodular maximization problem consists of finding a subset $S \subset V$ of a fixed size k that maximizes the function f. It shows up in various areas of machine

learning, such as active learning, sensing, summarization, feature selection and many others. See [Krause and Golovin, 2014] for a survey. The maximization problem is NP-hard, but is often approximated to within a factor of $1 - 1/e$ using a greedy algorithm [Nemhauser et al., 1978].

3.3 Approximation Algorithms

We first examine algorithms that start with an arbitrary vector of weights for all points, and then iteratively adjusts them to approximate a Magnitude weighting. Then we describe methods that increase efficiency by selecting a small subset of points that have magnitude close to that of X.

3.3.1 Convex optimization formulation and gradient descent

The problem of finding weights w can be formulated as a convex optimization using the squared loss:

$$\min_w \sum_i \left(\sum_j \zeta_{ij} w_j - 1 \right)^2 \tag{i}$$

This loss function is based on weighting (Definition 2.1.4), and reflects the error with respect to an ideal weighting where for each i, $\sum_j \zeta_{ij} w_j$ will add up to 1.

This is a strongly convex optimization problem and can be addressed using methods suitable for such optimization, including gradient descent or stochastic gradient descent.

3.3.2 Iterative Normalization Algorithm

In this section we present a different algorithm that we call the Iterative Normalization Algorithm. It starts with a weight vector of all ones. Then for every point i, it computes the sum $G(i) = \sum_j \zeta_{ij} w_j$. For a proper magnitude weighting every $G(i)$ should be 1, thus the algorithm simulates dividing by $G(i)$ to normalize the row to 1, and saves $w_i \leftarrow w_i / G(i)$. It does this in parallel for all rows (points).

Observe that compared to matrix inversion, which has a complexity of $O(n^{2.371552})$, the iterative normalization uses $O(n^2)$ per iteration, and achieves useable accuracy in relatively few iterations. In this problem, unlike usual optimization problems,

Algorithm 1 Iterative normalization algorithm for the approximation of magnitude

Input: The set of points X

Initialise $w_i = 1$ for all $i \in X$

while not converged **do**

Compute $G(i) = \sum_j \zeta_{ij} w_j$ for all $i \in X$

Update $w_i = w_i / G(i)$ for all $i \in X$

end while

we in fact know the optimum value of the loss for each point, and as a result we can use this approach of pushing the parameters toward this minimum value.

A caveat is that this algorithm produces a weighting that consists of positive weights. While individual magnitude weights can in principle be negative, Magnitude of a point cloud is always positive and in our experiments, the algorithm always finds a weighting whose sum converges toward the true magnitude. In this context, note that positive weights have been found to be relevant in predicting generalization of neural networks. See Chapter 5.

It appears that Algorithm 1 bears close resemblance to Gauss-Seidel iteration. Since the similarity matrix is positive definite, the algorithm is guaranteed to converge.

3.3.3 Approximation via greedy subset selection

To approximate more efficiently, we can attempt to identify a subset S of points that approximate the magnitude of X. Magnitude increases monotonically with addition of points to S [Leinster, 2013], which suggests approximation via algorithms that greedily add points to X similar to Nemhauser's submodular maximization [Nemhauser et al., 1978]. However, magnitude of a point set is not quite submodular, and thus the approximation guarantees do not carry over.

The non-submodularity can be seen in the following counterexample. Let $e_1, ..., e_D$ be the standard basis vectors of $\mathbb{R}^D$, so $e_1 = (1, 0, ..., 0)$ etc. Let $t > 0$ be a real number and $te_1, ..., te_D$ be the scaled basis vectors of $\mathbb{R}^D$, so $te_1 = (t, 0, 0..., 0)$ etc. Consider $X = \{te_1, -te_1, ..., te_D, -te_D\}$, with the usual metric. Thus X consists of the points on the axes of $\mathbb{R}^D$ that are a distance of t away from the origin. For a numerical example: when $t = 5$ and $D = 500$, we get $\mathrm{Mag}(X \cup \{0\}) - \mathrm{Mag}(X) \approx 7.18$. Thus, while the magnitude of a single point

Algorithm 2 Greedy algorithm for the computation of original magnitude

Input: The set of points S

Parameter: Tolerance k, k is a positive number between 0 and 1

Output: The approximated total magnitude and the maximising set S'

Initialise S' to be the empty set
Add a random element s_1 from S to S'.
while $Mag(S' \setminus s_i) < (1-k) * Mag(S')$ **do** (The previous computation of magnitude is within the tolerance parameter)
  Find the element s_i in $S \setminus S'$, maximising $Mag(S' \cup s_i)$
  Add s_i to S'
end while
return $S', Mag(S')$

(origin) is 1 by itself, adding it to X produces an increase far greater than 1.

This construction can be generalised to higher dimensions D, and behaves as follows in the limit:

Theorem 3.3.1. *Let $X = \{te_1, -te_1, ..., te_D, -te_D\}$ be a set of points in $\mathbb{R}^D$ as described above. Then in the limit:*

$$\lim_{D\to\infty} (\mathrm{Mag}(X \cup \{0\}) - \mathrm{Mag}(X)) = \frac{(e^t - e^{t\sqrt{2}})^2}{e^{2t} - e^{t\sqrt{2}}}.$$

3.3.3.1 Greedy set selection algorithm

While the theorem above implies that submodularity does not hold in general for magnitude, our experiments suggest that in practice, Nemhauser's algorithm [Nemhauser et al., 1978] adapted to Magnitude achieves approximation rapidly. A version of this idea can be seen in Algorithm 2.

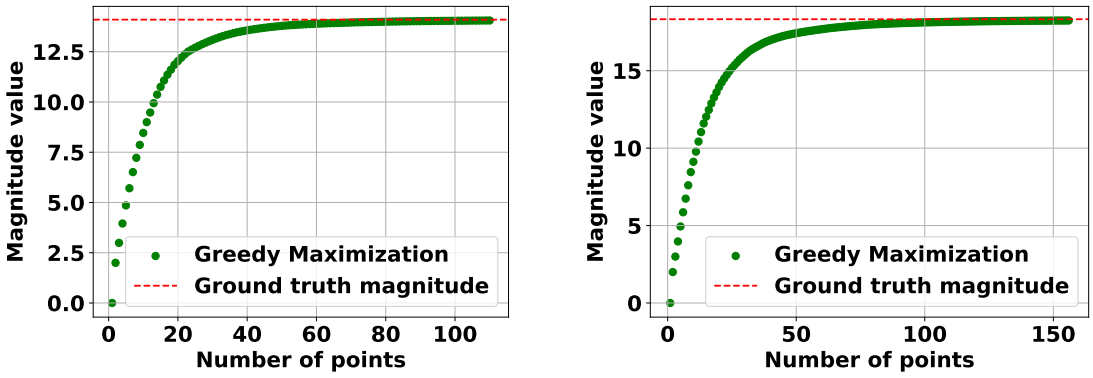
In certrain restricted cases, submodularity can be shown:

Theorem 3.3.2. $\mathrm{Mag}(X)$ *is submodular when $X \subset \mathbb{R}$.*

Thus, when $X \subset \mathbb{R}$, the greedy approximation of $(1 - 1/e)$ holds.

3.3.4 Discrete Center Hierarchy Algorithm

The computational cost of the greedy algorithm arises from the need to repeatedly compute magnitude at each greedy step to examine $\Omega(n)$ points and compute



(a) (b)

Figure 3.2: **Greedy algorithm approximates magnitude with small number of points.** Plot (a) shows magnitude approximation of a Gaussian blobs, 3 centers, with 500 points. Plot (b) shows Gaussian blobs with 3 clusters and 10^4 points.

magnitude each time to find the next point to add. To avoid this cost, we propose a faster approximation method.

In addition to being monotonically increasing with addition of points in X, the $\mathrm{Mag}(tX)$ also grows with t, and at the limit $\lim_{t\to\infty} \mathrm{Mag}(tX) = \#X$, where $\#X$ is the number of points in X [Leinster, 2013]. Therefore, point sets with larger distances between the points will have larger magnitude. Thus an iterative subset selection algorithm that prefers well-separated points is likely to increase the estimate faster toward the true magnitude.

This effect is achieved using Algorithm 3, which creates a hierarchy of discrete centers and uses them to successively approximate magnitude. The hierarchy is constructed as a sequence $S_0, S_1, S_2, \ldots$ of independent covering sets. Given a set S, a subset s is a minimal independent covering set of radius r, if it satisfies the following properties: (1): for every $x \in S$, there exists $y \in s$ such that $d(x, y) \leq r$ (2): $\forall x, y \in s, d(x, y) > r$ and (3) s is minimal with respect to these properties, that is, removing any point from s will violate the first property. With this in mind, we can construct the hierarchy as follows:

The hierarchy will have a height of at most $h = \log_2(\max_{x,y\in X} d(x, y))$, that is, log of the diameter of X. This hierarchy is used to successively approximate magnitude by traversing it from top to bottom. That is, starting from $s = \emptyset$, we first add points in S_h to s, followed by those in S_{h-1}, S_{h-2} etc, with $\mathrm{Mag}(s)$ increasing towards $\mathrm{Mag}(X)$.

Algorithm 3 Discrete Center Hierarchy construction

Input: (X, d).

$S_0 = X$

$S_i \leftarrow \emptyset$ for $i = 1, 2, ...$

for $i = 1, 2, ...$ **do**

Select $S_i \subseteq S_{i-1}$ where S_i is a minimal independent covering set of S_{i-1} of radius 2^{i-1}

end for

Observe that when computing Magnitude function (Definition 2.1.7) which requires computation for several values of t, this same sequence can be used for approximations at all the scales. Experiments described later show that a small number of points in this sequence (from the top few levels) suffice to get a good approximation of magnitude.

Incremental updates to the hierarchy. This hierarchy can be efficiently updated to be consistent with addition or removal of points. When a new point q is added, we traverse top down in the hierarchy searching for the center in S_i within distance 2^{i-1} to q. When such a center does not exist, we insert q to be a center in S_j for all $j \leq i$. With a data structure that keeps all centers of S_i within distance $c \cdot 2^{i-1}$ for some constant c, we could implement efficient 'point location' such that insertion takes time proportional to the number of levels in the hierarchy. If a point q is deleted from the hierarchy, we need to delete q from bottom up. At each level i if there are centers of S_{i-1} within distance 2^{i-1} from q, some of them will be selectively 'promoted' to S_i to restore the property. For more detailed description of a similar geometric hierarchy, see [Gao et al., 2006].

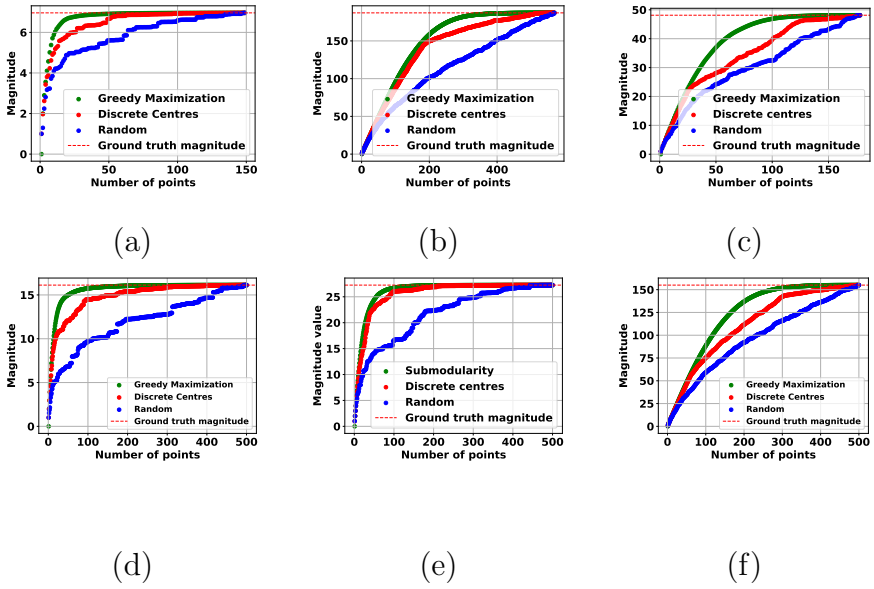


Figure 3.3: **Discrete centers are close to Greedy Maximization at a fraction of the computational cost and better than random.** In plot (a) we have the Iris dataset, in plot (b) the Breast cancer dataset, in plot (c) the Wine dataset. In the remaining plots, we see subsamples of size 500 for popular image datasets: (d) MNIST, (e) CIFAR10 and (f) CIFAR100.

3.4 Applications in Machine Learning

Here we describe the use of magnitude in two novel applications: as a regularization strategy for neural networks and for clustering.

3.4.1 Neural Network Regularization

High-varying neural network weights can be an indicator of overfitting to noise in the training data. Methods like weight decay add a term to the model's loss function to penalise large weights. We use Magnitude of the weights as a regulariser term. If the weight parameters are given by a vector p, where each $p_i \in \mathbb{R}$, then the magnitude of this metric space $(p, \mathbb{R})$ with the ambient metric of $\mathbb{R}$ is submodular (Theorem 3.3.2) with guaranteed approximation of $1 - 1/e$.

Specifically, we use the following algorithm to estimate magnitude. First select 1000 randomly chosen weights of the network, and then add the network weights with the smallest and largest values. As the set of the smallest and largest weights is the set of two weights with the largest possible magnitude, these points will be returned by the initial execution of the Greedy Maximization algorithm

Algorithm 4 Magnitude Clusterer

Let X be a set of points (scaled so the average pairwise distance is 1) and $t \geq 0$ be some threshold.
Initialise $R = X \setminus \{a\}$ and $C = \{\{a\}\}$ for some random point a.
while $R \neq \varnothing$ **do**
  Initialise **best increase** $= \infty$ and **best point** $= \varnothing$, **best cluster** $= \varnothing$.
  for $b \in R$, $c \in C$ **do**
    Set **increase** $= \mathrm{Mag}(c \cup \{b\}) - \mathrm{Mag}(c)$
    if increase $<$ **best increase then**
      best increase $=$ increase
      best point $=$ b
      best cluster $=$ c
    end if
  end for
  if increase $<$ t **then**
    Replace $c \in C$ with $c \cup \{$ **best point** $\}$.
  else
    Add $\{$ **best point** $\}$ to C.
  end if
  Remove **best point** from R.
end while
return C

which, as magnitude is submodular on the real line, has a theoretical guarantee of performance. Then select a random subset of the remaining weights.

3.4.2 Clustering

Inspired by the greedy approximation algorithm for submodular set functions, we propose a novel magnitude-based clustering algorithm. The key idea behind this algorithm is that, given a pre-defined set of clusters, if a new point belongs to one of those clusters then its inclusion in the cluster should not cause the magnitude of the cluster to increase significantly. Thus the algorithm works as follows: In every round, the algorithm tries to find a point b that is coherent with an existing cluster c, where coherence is measured as the change in magnitude of c being below some threshold t when adding b to c. If no such point-cluster pair can be found, then the algorithm initializes b as a new cluster. The details are in Algorithm 4.

Good thresholds can be found by carrying out magnitude clustering over a range of threshold values and monitoring the number of clusters. The cluster counts that persist over a range of threshold values are likely to be represent natural clusterings of the data. Selecting the most persistent count is natural way to determine clustering without any other parameter.

This clustering algorithm bears resemblance to ToMATo. Investigating further similarities and comparing the running times will be the subject of future work.

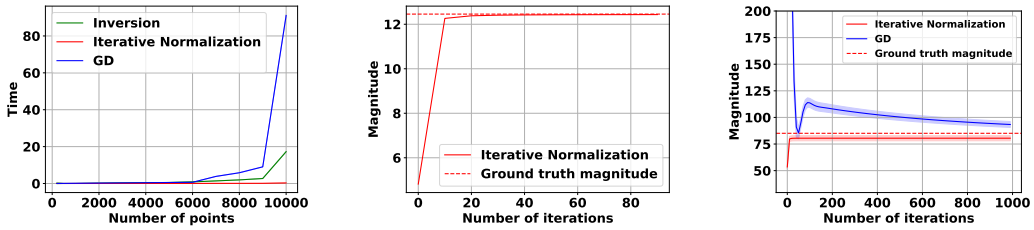
3.5 Experiments

Experiments ran on a NVIDIA 2080Ti GPU with 11GB RAM and Intel Xeon Silver 4114 CPU. We use `PyTorch`'s GPU implementation for matrix inverison. The SGD experiments used a learning rate of 0.01 and momentum of 0.9.

3.5.1 Accuracy and computation cost comparison

3.5.1.1 Iterative algorithms

In Figure 3.4 we see a comparison of the iterative algorithms on points sampled from $\mathcal{N}(0, 1)$ in $\mathbb{R}^2$ with 10^4 points. We observe that the Iterative Normalization algorithm is faster than Inversion and Gradient descent in plot (a) and it only



(a) (b) (c)

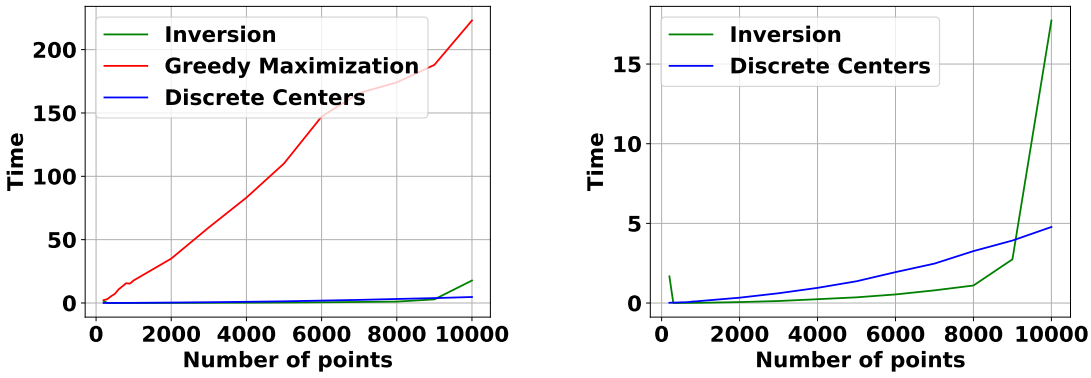
Figure 3.4: **Iterative algorithms comparison** Comparison of Inversion, Iterative Normalization and GD (a) Mean and standard deviation over 10 different runs, with 50 iterations of both iterative algorithms. (b) Number of iterations for convergence of Iterative Normalization for a randomly generated sample of 10000 points. (c) Iterative Normalization vs GD. Iterative Normalization converges fast, GD takes a longer number of iterations. 100 runs. Comparison on larger point sets in supplementary materials.

needs a few iterations (less than 20) to converge as seen in plot (b), while Gradient descent takes a longer number of iterations. In plot (c) we see the convergence performance over 100 different runs, and again we note that Iterative Normalization converges fast, while GD requires a larger number of iterations.

3.5.1.2 Subset selection algorithms

Figure 3.5, shows a comparison of the subset selection algorithms on a randomly generated dataset with 10^4 points sampled from $\mathcal{N}(0, 1)$ in $\mathbb{R}^2$.

Figure 3.3 shows the performance of the subset selection algorithms for a number of `scikit-learn` datasets (Iris, Breast Cancer, Wine) and for subsamples of MNIST, CIFAR10 [Krizhevsky et al., 2014] and CIFAR100 [Krizhevsky, 2009]. We note that the Greedy Maximization performs the best, but Discrete Centers produces a very similar hierarchy of points. Selecting points at Random does not lead to an improvement in a sense that you need to approach the cardinality of the set to get a good enough approximation of magnitude.



(a) (b)

Figure 3.5: **Subset selection algorithms comparison** (a) Time taken for Inversion, Greedy Maximization and Discrete Centers to execute. (b) zoom on the performance of Inversion and Discrete Centers, and note that Discrete Centers performs better as the number of points increases. Comparison on larger datasets in supplementary materials.

3.5.2 Applications in ML

3.5.2.1 Training trajectories and generalization

It has been shown that Magnitude and a quantity derived from Magnitude called Positive magnitude (PMag), consisting of positive weights are important in bounds of worst case generalization error. The method relies on computing a trajectory by taking n steps of mini-batch gradient descent after convergence, and computing the Magnitude of corresponding point set on the loss landscape. See Chapter 5 for details.

The experiments up to now have been limited to training trajectories of at most size 5×10^3 due to computational limitations. Our faster approximation methods can allow us to verify the results on larger trajectories.

We denote by $\mathrm{Mag_n}$ and $\mathrm{PMag_n}$ the relevant quantities trajectories of length n. Sizes up to 5000 have been considered in the original paper. We extend to sizes of 7000 and 10000. We use ViT [Touvron et al., 2021a] on CIFAR10, ADAM optimizer [Kingma and Ba, 2017], and perform the experiment over a grid of 6 different learning rates and 6 batch sizes, where the learning rate is in the range $[10^{-5}, 10^{-3}]$, and the batch size is between $[8, 256]$ resulting in 36 different experimental settings.

Metric	$\boldsymbol{\psi}_{\text{lr}}$	$\boldsymbol{\psi}_{\text{bs}}$	$\boldsymbol{\Psi}$	τ
Mag_{5000}	0.68	0.62	0.65	0.64
Mag_{7000}	0.71	0.77	0.74	0.69
Mag_{10000}	**0.75**	**0.82**	**0.79**	**0.74**
PMag_{5000}	0.91	0.67	0.79	0.85
PMag_{7000}	0.93	0.73	0.83	0.88
PMag_{10000}	**0.97**	**0.79**	**0.88**	**0.90**

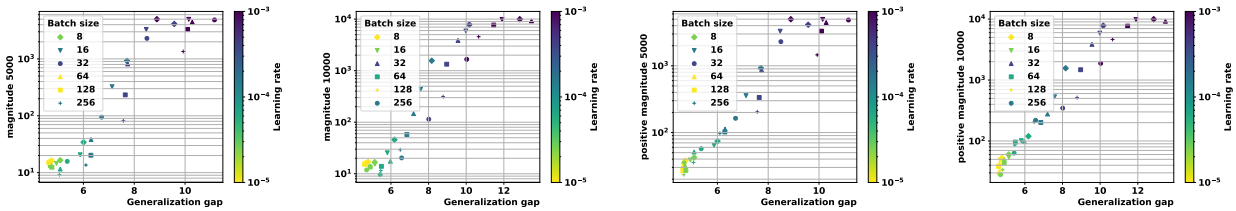
Table 3.1: Generalization gap correlation improvement using an increasing number of points. $\boldsymbol{\psi}_{\text{lr}}$ and $\boldsymbol{\psi}_{\text{bs}}$ are the granulated Kendall coefficient for the learning rate and for batch size respectively, and $\boldsymbol{\Psi}$ is the Average Kendall coefficient, which is the average of $\boldsymbol{\psi}_{\text{lr}}$ and $\boldsymbol{\psi}_{\text{bs}}$[Jiang et al., 2020])
. τ is Kendall tau. For the full definition and discussion on correlation measures, please consult Section 2.2

The results can be found in Table 3.1, showing a number of correlation coefficients relevant for generalization [Jiang et al., 2020] between generalization gap and Mag_{n} and PMag_{n} for $n = \{5000, 7000, 10000\}$. We use the granulated Kendall's coefficients ($\boldsymbol{\psi}_{\text{lr}}$ and $\boldsymbol{\psi}_{\text{bs}}$ are the granulated Kendall coefficient for the learning rate and for batch size respectively, and $\boldsymbol{\Psi}$ is the Average Kendall coefficient, which is the average of $\boldsymbol{\psi}_{\text{lr}}$ and $\boldsymbol{\psi}_{\text{bs}}$ [Jiang et al., 2020]), which are more relevant than the classical Kendall's coefficient for capturing causal relationships.

We observe that all correlation coefficients improve with the increase of trajectory size. In particular, the Kendall tau coefficient and the Average Granulated Kendall coefficient increases by 0.14 for Mag_{10000} compared to Mag_{5000}, and by 0.09 for PMag_{10000}. Similarly, Kendall tau improves by 0.10 for Mag_{10000} and 0.05 for PMag_{10000}. This is an interesting result which needs to be investigated further for more models and datasets. Further visualisation results can be seen in Figure 3.6, where we see how the proposed quantities change with the generalization gap, and when more trajectories are considered.

3.5.2.2 Neural Network Regularization

Utilising the magnitude approximation described in Section 3.4.1, we train five neural networks each with two fully connected hidden layers on the MNIST dataset for 2000 epochs, using cross entropy loss on MNIST. We train the models with



(a) (b) (c) (d)

Figure 3.6: **Extended complexity measures vs. the generalization gap** We compare the original topological complexity measure Mag_{5000} (a) and PMag_{5000} (c) with the extended complexity measures Mag_{10000} (b) and PMag_{10000} (d) for a ViT trained on CIFAR10.

λ	Train. Loss	Test Loss	Gap	Magnitude
0	0.0021	0.0757	0.0736	1.5810
0.1	0.0041	0.0641	0.0600	1.1567
0.2	0.0061	0.0607	0.0546	1.1293
0.5	0.0103	0.0602	0.0499	1.0842
1	0.0167	0.0631	0.0464	1.0668

Table 3.2: Magnitude and performance of Neural Networks after training to minimise Training Loss(weights) + λMag(weights).

a scalar multiple of the magnitude of the weights as a penalty term. One of the networks we train (with a regularization constant of 0) corresponds to an unregularised model. We then evaluate the differences in magnitude as well as train and test loss for each model.

Our results are shown in Table 3.2. We first observe that as expected, adding a magnitude-based penalty term causes the network's magnitude to decrease. More interestingly magnitude regularization causes the neural network to perform better. This increase in performance occurs both in terms of test loss and generalization error, with the unregularised model recording both the largest test loss and generalization gap. It is also interesting to note that the generalization appears to increase consistently with the strength of regularization, whereas test loss appears to have an optimal strength of regularization at $\lambda = 0.5$.

3.5.2.3 Clustering

The results of using the Clustering algorithms described in the previous section are presented in Figure 3.7. We note that our algorithm performs well, providing a better clustering than Agglomerative, k-means and DBSCAN.

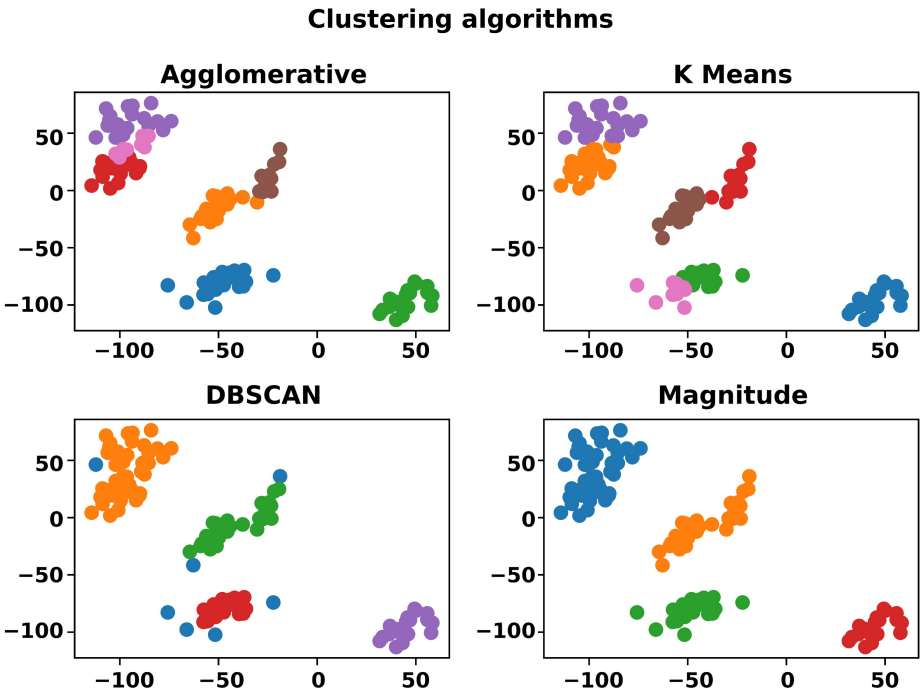


Figure 3.7: Results of applying the Magnitude clustering algorithm to an artificial dataset. We can see that the magnitude-based algorithm manages to find natural clusters and is able to determine a suitable number of clusters, whereas K-means and hierarchical clustering (Ward clustering in `scikit-learn`) need the user to determine this. A main difference between DBSCAN and the magnitude algorithm is in the treatment of outliers.

The magnitude clustering algorithm's true strength emerges when dealing with unknown, randomly generated datasets, unlike toy datasets where manual parameter tuning for other models masks their limitations. We observed that on linearly separable toy datasets, magnitude clustering performs comparably to existing methods. However, like k-means and agglomerative linkage, it struggles with non-linearly separable data, a significant limitation shared by its meta-version, as it relies on the same distance metric. For unknown datasets, magnitude clustering performs very well, often merging small, closely spaced clusters that other algorithms keep separate. It also occasionally differs from DBSCAN, some-

times classifying points as anomalies or belonging to larger clusters, suggesting a tendency to create larger clusters due to its threshold-searching behavior. While magnitude clustering shows promise, its superiority is not universally clear across all datasets. This ambiguity is a common challenge in clustering analysis when ground truth is unavailable. An idea was to evaluate the algorithm with labeled real-world datasets and to see if that would offer quantitative insights. Comparing all the algorithms in more detail was outside the scope of the current chapter, but it will be interesting to do so in future work.

3.6 Related work

The literature relevant to magnitude and machine learning has already been discussed in previous sections. We have studied closely the relation of magnitude to generalization in Chapters 4 and 5, and the diversity of latent representations in Chapter 6. Magnitude based clustering has been suggested in [O'Mally, 2023]. The algorithms proposed make use of a similar quantity called alpha magnitude [O'Malley et al., 2023], but unlike ours, requires multiple parameters as input.

Recent developments such as Magnitude for graphs [Leinster, 2019] and relation between Magnitude and entropy [Chen and Vigneaux, 2023] are also likely to be of interest in machine learning – as is the interpretation of magnitude as a dual of the Reproducing Kernel Hilbert Space [Meckes, 2015].

The computational problem can be seen as solving the linear system $\zeta w = \mathbb{1}$, with $\mathbb{1}$ as a vector of all 1. Notice that our matrix ζ is symmetric positive definite but also a dense matrix. The iterative normalization algorithm we proposed bears resemblance to basis pursuit, Gauss-Seidel and other algorithms in compressive sensing [Foucart and Rauhut, 2013], but the relation is not yet clear. Furthermore, obtaining approximation bounds relates to the 1-norm, making the extensive body of work on 2-norm approximation not applicable to this setting.

3.7 Conclusion

In this chapter, we introduced fast and scalable methods for approximating metric magnitude, which should help greater application and exploration of magnitude in improving machine learning and our understanding of it. The novel applications to deep learning and clustering also can be explored further.

Chapter 4

Metric Space Magnitude and Generalisation in Neural Networks

Deep learning models have seen significant successes in numerous applications, but their inner workings remain elusive. The purpose of this work is to quantify the learning process of deep neural networks through the lens of a novel topological invariant called *magnitude*. Magnitude is an isometry invariant; its properties are an active area of research as it encodes many known invariants of a metric space. We use magnitude to study the internal representations of neural networks and propose a new method for determining their generalisation capabilities. Moreover, we theoretically connect magnitude dimension and the generalisation error, and demonstrate experimentally that the proposed framework can be a good indicator of the latter.

4.1 Introduction

Deep neural networks (DNNs) have become ubiquitous due to their remarkable performance in a range of tasks, including computer vision [Xie et al., 2017], natural language processing [Vaswani et al., 2017], and scientific discovery [Jumper et al., 2021]. However, state-of-the-art DNNs often comprise millions to billions of parameters, making it impractical for human users to gain a precise understanding of the inner workings of these networks. One crucial yet unanswered question is to understand the generalisation of neural networks, i.e. their ability to perform

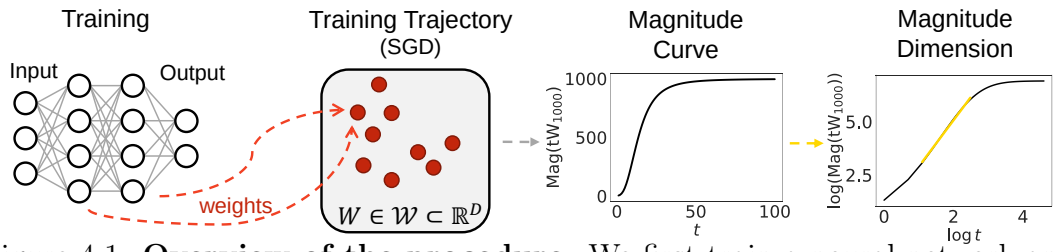


Figure 4.1: **Overview of the procedure.** We first train a neural network and monitor the training trajectory. At each 1000 iterations of the trajectory, we collect the training weights $W \in \mathbb{R}^d$ into a point cloud. Then, we compute the magnitude curve of the point cloud at selected scales t, create the log-log plot and estimate the magnitude dimension based on it.

well on unseen data. In recent literature, various notions of neural networks' intrinsic dimensions have been proposed [Simsekli et al., 2020, Birdal et al., 2021, Dupuis et al., 2023], which demonstrate that the most important ingredient for generalisation is effective capacity rather than the number of parameters.

In this work, we propose a novel measure of the intrinsic dimension of neural networks, based on a novel topological invariant called magnitude. Magnitude's topological roots stem from its original definition as the Euler characteristic of specific finite categories [Leinster, 2008], marking it as a fundamental topological invariant. Unlike previous approaches, magnitude benefits from a simple interpretation, and potentially better computational complexity. This choice is also theoretically justified. Magnitude is an isometry invariant of a metric space and its properties are an active area of mathematical research due to the fact that it encodes many important invariants from geometric measure theory and integral geometry [Leinster, 2013]. Further, magnitude has roots in theoretical ecology and it can capture the effective number of species in an ecosystem. In a broader context, it has been shown that magnitude can thus encode the effective number of distinct points in a space [Leinster, 2013]. We further build on this idea and propose a novel method for evaluating the generalisation error using the concept of an effective number of models, which acts as an effective capacity.

Although there has been significant theoretical research on magnitude, a considerable number of its characteristics are yet to be discovered, and several unanswered questions persist, particularly regarding its practical applications in machine learning. While recent studies [Bunch et al., 2021, Adamer et al., 2021] have started to establish links between magnitude vectors and machine learning

applications, this area of research is still in its very early stages.

Recently, it has been shown that a concept derived from magnitude, known as the magnitude dimension, is the same as the Minkowski dimension [Meckes, 2015] under certain conditions. As a result, we can theoretically show that the generalisation error of the trajectories of a training algorithm is intrinsically linked to the magnitude dimension of the so-defined metric space. This enables us to generate novel insights about the learning process of neural networks. Further, our contribution is the first work exploring the magnitude dimension, which has solid theoretical foundations, as a notion of the intrinsic dimension of neural networks.

Contributions. Our work is the first to introduce the mathematical concept of magnitude into theoretical deep learning and the study of neural network generalisation; we believe that this is an exciting novel contribution to the nascent field of topological machine learning [Hajij et al., 2022, Hensel et al., 2021]. After establishing a theoretical bound, we explore the change in the magnitude function across multiple experimental settings, and compare the value of magnitude at multiple scales to the test accuracy. Further, we demonstrate experimentally that there is a link between magnitude and the test accuracy. By making a novel connection with the persistent homology dimension, we then compare our proposed measure with the intrinsic dimension introduced by Adams et al. [2020] and demonstrate that ours benefits from a better computational complexity and interpretability. In short, our contributions are as follows:

- We propose a novel method for evaluating the generalisation of neural networks based on magnitude and the effective number of models, which allows us to monitor performance without a validation set.
- We prove a new upper bound for the generalisation error, linking the generalisation error to a magnitude-based characteristic of the training trajectories.
- We empirically show that the evolution of these measures throughout the training process correlates with the accuracy of the test set.
- We prove that all notions of previously proposed intrinsic dimensions are the same as the magnitude dimension, and we verify this result empirically.

4.2 Related Work

We briefly review the literature related to generalisation in neural networks, intrinsic dimension, and magnitude.

Generalisation Bounds and Intrinsic Dimension.. Several works explore intrinsic dimension for capturing the generalisation capabilities of neural networks. Simsekli et al. [2020] demonstrated that the fractal dimension of a hypothesis class is associated with the generalisation error, which is further linked to the heavy-tailed behavior of the trajectory of networks [Simsekli et al., 2019, Hodgkinson and Mahoney, 2021, Mahoney and Martin, 2019]. However, many assumptions were required for the bound to be computed in practice. A more recent work relaxed some of the assumptions, and developed the notion of the persistent homology dimension [Adams et al., 2020], $\dim_{\mathrm{PH}}$. The authors in Birdal et al. [2021] were the first to offer a theoretical justification for using topological invariants for the analysis of deep neural networks. Another work [Magai and Ayzenberg, 2022] investigated $\dim_{\mathrm{PH}}$ at different depths and layers of the network and observed its evolution. In Dupuis et al. [2023], the authors developed a data-driven dimension and compared it with the $\dim_{\mathrm{PH}}$ of Birdal et al. [2021]. They have demonstrated stronger correlation with the generalisation error than previously shown and managed to relax some of the restrictive assumptions.

Magnitude and its Applications in Machine Learning.. Magnitude was first proposed in Solow and Polasky [1994] for measuring biodiversity, albeit without any reference to its mathematical properties. It was only approximately twenty years later when Leinster [2013] formalised its mathematical properties using the language of category theory. Further, magnitude has been realised as the Euler characteristic in magnitude homology [Leinster and Shulman, 2021]. While magnitude has theoretical foundations, its applications to machine learning are scarce. Recently, there has been renewed interest in introducing magnitude into the machine learning community. The first work to develop the concept of magnitude in the context of machine learning demonstrated that the individual summands of magnitude, known as magnitude weights, can be used as an efficient boundary detector [Bunch et al., 2021]. Further, it has been used for working in the space of images and it has demonstrated its usefulness as an effective edge detector [Adamer et al., 2021]. However, our contribution constitutes the first direct application of magnitude to deep learning.

Using Topology to Characterise Neural Networks.. Earlier research has established a connection between neural network training and topological invariants [Fernández et al., 2021], using topological complexity as proxy for generalisation performance, for instance [Rieck et al., 2019]. However, these studies focused solely on analyzing the trained network after completing the training process [Fernández et al., 2021], potentially missing critical aspects of the training dynamics [Birdal et al., 2021]. By contrast, we propose the use of another topological invariant—magnitude—which affords more interpretability than previous approaches. Moreover, we compute magnitude on the training trajectories instead of on the trained network, offering crucial topological insights into training dynamics.

4.3 Background

The relevant background on magnitude can be found in Chapter 2. Here we define various notions of intrinsic dimensions.

4.3.1 Intrinsic Dimension

There are various notions that can be used to measure the intrinsic dimension of a space. In this work, we will focus on three such notions: the upper-box dimension (Minkowski), the magnitude dimension and the persistent homology dimension. The box dimension is based on covering numbers and can be linked to generalization via Simsekli et al. [2020], whereas the magnitude dimension is built upon the concepts we defined earlier.

Definition 4.3.1 (Minkowski dimension)**.** For a bounded metric space X, let $N_\delta(X)$ denote the maximal number of disjoint closed δ-balls with centers in X. The upper box/Minkowski dimension is defined as

$$\dim_{\text{Mink}} X = \lim_{\delta \to 0} \sup \frac{\log(N_\delta(X))}{\log(\frac{1}{\delta})}. \tag{i}$$

There is a subtle point to be made here. In general, the Minkowski and Hausdorff dimensions do not coincide and are not equivalent. However, in Simsekli et al. [2020] the authors provide conditions under which the Hausdorff dimension of the space we are interested in coincides with the Minkowski dimension. In fact,

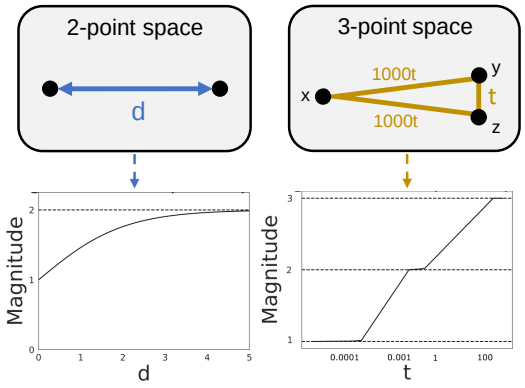


Figure 4.2: **Magnitude of the 2- and 3-point space.** On the left, we see the 2-point space, where the distance between the points is d. On the right we see the 3-point space for an isosceles triangle with distance t between y and z, and $1000t$ between x and y and x and z. Below each space, we see the respective magnitude function.

for many fractal-like sets, these two notions of dimensions are equal to each other; see Mattila [1999a, Chapter 5].

Definition 4.3.2 (Magnitude dimension)**.** When

$$\dim_{\mathrm{Mag}} X = \lim_{t \to \infty} \frac{\log(\mathrm{Mag}(tX))}{\log t} \tag{ii}$$

exists, we define this to be the magnitude dimension of X [Meckes, 2015].

The magnitude dimension can be approximately interpreted as the rate of change of the magnitude function for a suitable interval of values for t. We can introduce another notion of the fractal dimension, known as the persistent homology dimension ($\dim_{\mathrm{PH}}$) [Adams et al., 2020].

Definition 4.3.3. The persistent homology dimension of a bounded metric space (X, d), denoted by $\dim^0_{\mathrm{PH}}$, is defined as

$$\inf\{\alpha > 0, \exists C > 0, \forall W \subset X \text{finite}, E_\alpha < C\}, \tag{iii}$$

where E_α is the α-lifetime sum, defined as

$$E_\alpha(W) \coloneqq \sum_{(b,d) \in PH^0(Rips(W))} (d-b)^\alpha,$$

and b and d are the birth and death values respectively for the persistent homology of degree 0 (PH^0). It measures all connected components in the Vietoris-Rips filtration of W, denoted by $Rips(W)$.

The definition is rather technical and it is not crucial for the work here, for more details please refer to Adams et al. [2020], Mémoli and Singhal [2019], Edelsbrunner and Harer [2022]. We note that it can be defined for persistent homology of any degree, but for the purpose of this chapter, we are only interested in degree 0. Therefore, for ease of notation, we will omit the 0 and denote $\dim_{\mathrm{PH}}^0$ by $\dim_{\mathrm{PH}}$, i.e. $\dim_{\mathrm{PH}} := \dim_{\mathrm{PH}}^0$.

4.4 Theoretical Results

We first elucidate connections between different notions of intrinsic dimension before proving connections to the generalisation error.

4.4.1 Connection between Notions of Intrinsic Dimension

After we have introduced three different notions of intrinsic dimensions, we demonstrate that they are in fact the same under some mild assumptions. Our novel contributions is proving that $\dim_{\mathrm{Mag}} X$ and $\dim_{\mathrm{PH}}^0 X$ are equal. The result is formalised in the following theorem, which assumes that all notions of dimension exist.

Theorem 4.4.1. *Let $X \subset \mathbb{R}^n$ be a compact set and either $\dim_{\mathrm{Mag}} X$ or $\dim_{\mathrm{Mink}} X$ exist. Then*

$$\dim_{\mathrm{Mag}} X = \dim_{\mathrm{PH}}^0 X \tag{iv}$$

Proof. Since X is compact and either $\dim_{\mathrm{Mag}} X$ or $\dim_{\mathrm{Mink}} X$ exist, from Corollary 7.4, [Meckes, 2015] it follows that both $\dim_{\mathrm{Mag}} X$ and $\dim_{\mathrm{Mink}} X$ exist and $\dim_{\mathrm{Mag}} X = \dim_{\mathrm{Mink}} X$. Since X is compact, from the Heine-Borel Theorem, X is both closed and bounded, and therefore from a result in Kozma et al. [2006], Schweinhart [2021], we have that $\dim_{\mathrm{Mink}} X = \dim_{\mathrm{PH}}^0 X$. Hence, $\dim_{\mathrm{Mag}} X = \dim_{\mathrm{Mink}} X = \dim_{\mathrm{PH}}^0 X$, which implies that $\dim_{\mathrm{Mag}} X = \dim_{\mathrm{PH}}^0 X$.

□

4.4.2 Connection to the Generalisation Error

After having established equality between the various notions of dimensions, we proceed to formalise the required language of machine learning theory, culminating

in a novel generalisation result.

For the beginning of this section, we follow the notation from Shalev-Shwartz and Ben-David [2014]. We briefly recall some standard definitions to make our chapter self-contained. In a standard statistical learning setting, $\mathcal{X}$ denotes the set of features and $\mathcal{Y}$ the set of labels. Together, the cross product $\mathcal{X} \times \mathcal{Y}$ represents the space of data $\mathcal{Z}$. The learner has access to a sequence of data of m samples, called the training data, denoted by $S = ((x_1, y_1), ..., (x_m, y_m))$ in $\mathcal{X} \times \mathcal{Y} = \mathcal{Z}$. We assume that the training set S is generated by some unknown probability distribution over $\mathcal{X}$, which we denote by $\mathcal{D}$, and that each of the samples $\{x_i, y_i\}$ are independent and identically distributed (i.i.d) samples from $\mathcal{D}$. We will focus on a restricted search space for finding a set of predictors, which we call a hypothesis class, $\mathcal{H}$. Each element, called a hypothesis, $h \in \mathcal{H}$, is a function from $\mathcal{X}$ to $\mathcal{Y}$. An optimal hypothesis, or parameter vector $h \in \mathcal{H}$ is selected by computing a quantity called a loss function, defined by $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$. The empirical error is then $L_S(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h, z)$ for $z \in \mathcal{Z}$ and the true error or risk is $L_{\mathcal{D}}(h) = \mathbb{E}_z[\ell(h, z)]$ for $z \in \mathcal{Z}$. The generalisation error is defined as the difference between the true error and the empirical risk, or $|L_S(h) - L_{\mathcal{D}}(h)|$.

In the case of neural networks, the hypothesis class is $\mathcal{W} \subset \mathbb{R}^d$, and each $w \in \mathcal{W}$ is a parameter vector. Given a training algorithm $\mathcal{A}$, we want to study the set of all hypothesis classes returned from the optimisation procedure $\mathcal{A}$ for a given training data S. We call this the *optimisation trajectories*, and denote them by $\mathcal{W}$. To access the iterates at every step of the process, we denote by w_i an element of $\mathcal{W}$ at iteration i. More concretely, $\mathcal{W} := \{w \in \mathbb{R}^d : \exists i \in [0, I], w = [\mathcal{A}(S)_i]\}$, where I is the number of training iterations. In other words, when we fix i, we are interested in the weights at iteration i, returned by the optimisation algorithm $\mathcal{A}$.

For the following result, there is a more technical condition required from Birdal et al. [2021, Assumption H1], which can be found in the Appendix. We denote this assumption as H1. We require the existence of the constant $M > 0$, which quantifies how dependent the set $\mathcal{W}_S$ is on the training sample S. In simple terms, Assumption H1 says that the way your loss function behaves is not too tightly "tied" to the precise spatial layout of your training data. There's enough "randomness" or "independence" between these two aspects to allow for statistical analysis and the derivation of meaningful bounds on generalization performance. Smaller M indicates that the dependence of $L_S(w)$ on the training sample S is weaker. It means that how the loss function behaves (given by $L_S(w)$) is not

overly constrained or determined by the specific geometric arrangement of the training data samples S. In other words, knowing how the training samples are spatially arranged doesn't give you too much information about the specific loss values, and vice versa. Also known as the ψ-mixing condition [Bradley, 1983] is a standard tool in probability theory to allow for the decoupling of dependencies between different components of a stochastic process. Here, it helps to separate the randomness associated with the loss values from the randomness associated with the geometry of the training data.

This is a relatively strong theoretical assumption. It's highly unlikely to be literally satisfied with a small M for arbitrary deep learning models and datasets. The dependence between the loss landscape and data geometry is often very strong. Researchers typically make these assumptions for mathematical tractability to derive initial theoretical bounds, which can then serve as a basis for understanding, even if the assumptions themselves might need to be relaxed or empirically validated for specific applications.

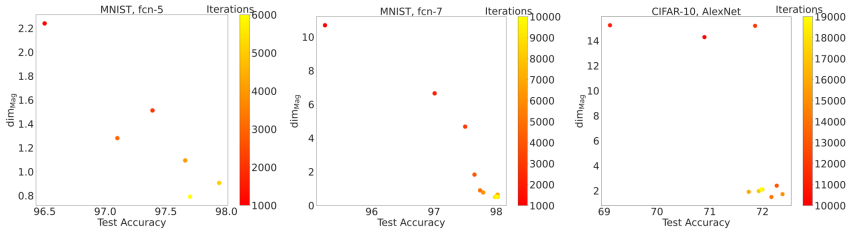
Now that we have all the required definitions, we can proceed with the novel result.

Theorem 4.4.2. *Let $\mathcal{W} \in \mathbb{R}^d$ be a compact set. Under the assumption that $H1$ holds, ℓ is bounded by a constant C and K-Lipschitz continuous in w. For n sufficiently large, we then have the following bound:*

$$\sup_{w \in \mathcal{W}} |L_S(w) - L_{\mathcal{D}}(w)| \leq 2C\sqrt{\frac{[\dim_{\mathrm{Mag}}\mathcal{W} + 1]\log^2(nK^2)}{n} + \frac{\log(7M/\gamma)}{n}} \tag{v}$$

with probability $1 - \gamma$ over $S \sim \mathcal{D}^{\otimes n}$, where M is the constant from Assumption H1.

Proof. Since $\mathcal{W}$ is bounded, we have $\dim^0_{\mathrm{PH}}\mathcal{W} = \dim_{\mathrm{Mag}}\mathcal{W}$. Therefore, the result follows from substituting $\dim^0_{\mathrm{PH}}\mathcal{W}$ with $\dim_{\mathrm{Mag}}\mathcal{W}$ in Proposition 1 [Birdal et al., 2021]. □



(a) (b) (c)

Figure 4.3: **The magnitude dimension correlates with the test accuracy.** In plots (a-c), we see the magnitude dimension plotted against the test accuracy over a varying number of iterations, which are depicted in different colours, from red to yellow. We note that there is correlation between the magnitude dimension and the test accuracy across both MNIST and CIFAR-10, and across different architectures (FCN-5, FCN-7, AlexNet), which is stronger for MNIST than for CIFAR-10.

4.5 Methods

In contrast with previous work on the intrinsic dimension, we propose to estimate the fractal dimension using the magnitude. As we have seen, the concept of magnitude dimension coincides with the Minkowski dimension and the persistent homology dimension. This theoretical connection enables us to confidently explore the concept of magnitude in the context of neural networks. We do this as follows: at selected points on the weight trajectory $\mathcal{W}$, we subsample a number of models W, which can be interpreted as a point cloud, where each point is a model in the model space. This space has a very high dimension equal to the number of parameters in the network. We compute the magnitude and the magnitude dimension of each such point cloud. We then investigate the connection between these quantities and the test accuracy.

In light of our novel result in Theorem 4.4.2, linking the intrinsic dimension and the generalisation bound, it is natural to ask if the magnitude function can also be used to explore the learning process of neural networks. Since the magnitude dimension can be roughly interpreted as the rate of change of the magnitude function, and therefore, if there is a link between the magnitude dimension and the generalisation error, then there should also be a link between the values of the magnitude function at different scales and the generalisation error. What we want to study is the change of the space of model trajectories as the learning process

advances. In other words, does the space look like a bigger number of distinct points or does it resemble fewer points as the training progresses? Translating this idea to the language of magnitude, is the *effective number of models* increasing or decreasing when performing more iterations of the learning algorithm?

Since magnitude is a function, we would like to take a cross-sectional slice of the magnitude curve and examine the magnitude values for a particular choice of the scale parameter t. Therefore, we formulate the definition of the effective number of models for a fixed t.

Definition 4.5.1. We define the **effective number of models** as the value of $\mathrm{Mag}(t_i\mathcal{W})$ at scale t_i.

4.5.1 Analyzing Deep Neural Network Dynamics via the Magnitude Dimension

Estimating the magnitude dimension is not a straightforward task, as the limit from Equation ii needs to be approximated by finding the longest straight part of the magnitude function, which cannot be computed automatically, but needs to be done manually. Here we provide the algorithm for computation of $\dim_{\mathrm{Mag}}\mathcal{W}$ from a finite sample W, approximating an infinite process. We follow the procedure similar to the estimations of the magnitude dimension [Willerton, 2009, O'Malley et al., 2023]. The details are described in Algorithm 5. After choosing a suitable representative interval $[t_i, t_j]$, the log-log plot of magnitude versus t is generated, and the slope m of the line and the intercept b are computed at the selected interval $[t_i, t_j]$. Then, m is taken to be the estimate of $\dim_{\mathrm{Mag}}\mathcal{W}$. This procedure can be essentially seen as computing the limit based on the slope, which is an application of l'Hôpital's rule. The representative interval has been chosen manually by determining the longest straight line segment in the plot. This procedure could introduce errors and an automated way to estimate the dimension could improve the estimate.

Algorithm 5 Estimation of $\dim_{\text{Mag}}\mathcal{W}$

Input: The set of the training trajectories $\mathcal{W} = \{w_i\}^n$ of size n, scale parameter $t : [0, t_k]$, interval $[t_i, t_j], i < j < k$

Output: $\dim_{\text{Mag}}$

for $r = 1$ **to** k **do**

  Compute $\text{Mag}(t_r\mathcal{W})$

end for

$m, b \leftarrow \text{fitline}(log(\text{Mag}(\mathcal{W}_n)[t_i : t_j]), log([t_i : t_j]))$

$\dim_{\text{Mag}}\mathcal{W} \leftarrow m$

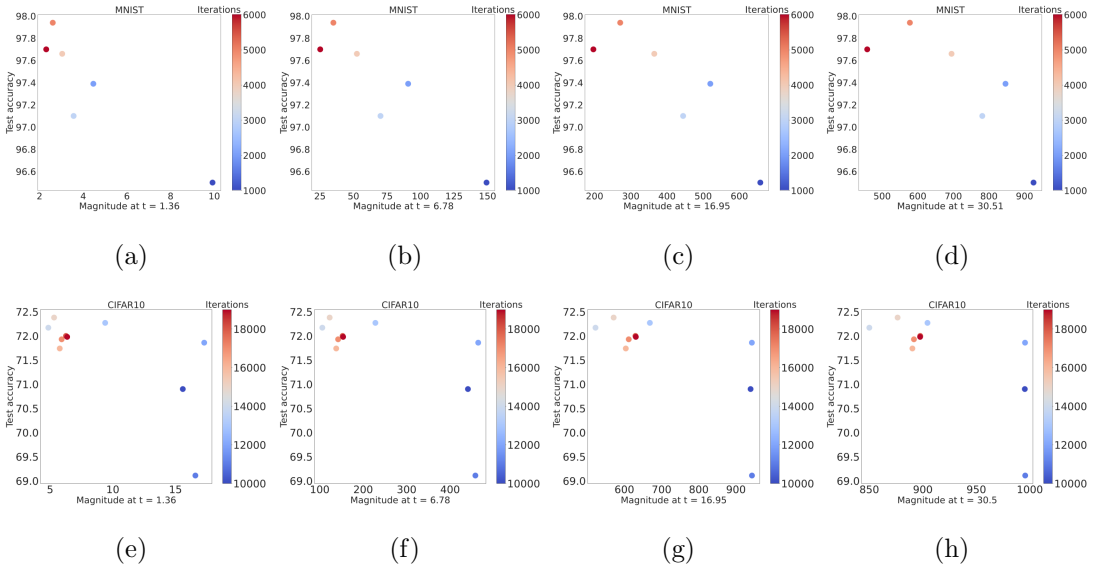


Figure 4.4: **The effective number of models correlates with the test accuracy.** Here we see the cross-sectional evaluation of the magnitude curve at different values of t. Each point represents the model trajectory over 1000 iterations, over which the magnitude is computed, as well as the test accuracy at the last model in the sliding window. The first row (plots (a-d)) shows the magnitude for MNIST. The second row (plots (e-h)) shows the plots for CIFAR10. We note that across all scales, there is similar pattern of correlation between the test accuracy and the magnitude values.

4.6 Experimental Results

The first goal of this section is to verify our main claim, namely that our estimate of the magnitude dimension is capable of measuring generalisation. The second goal is to establish a connection between magnitude itself and the test accuracy.

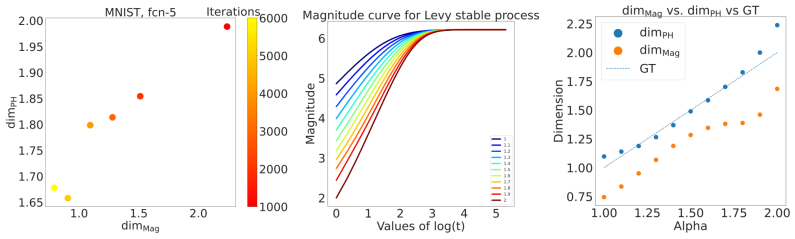
The third goal is to empirically compare the two very close measures of the fractal dimension, namely $\dim_{\mathrm{Mag}}\mathcal{W}$ and $\dim_{\mathrm{PH}}\mathcal{W}$. The fourth goal is to attempt to explain what our results mean for generalisation in neural networks in general, arriving at novel insights. In order to show this, we use our procedure on a number of different neural network architectures, training settings and datasets. In particular, we train a 5-layer (fcn-5) and 7-layer (fcn-7) on MNIST and AlexNet [Krizhevsky et al., 2017] on CIFAR10, over a different range of learning rates and batch size of 100. We consider a sliding window of 1000 training iterations. We then estimate $\dim_{\mathrm{Mag}}\mathcal{W}$ of each window, following the steps in Algorithm 5.

4.6.1 Exploring the learning process

We assess the main claim of the chapter, which links the magnitude dimension with the generalisation error. Figure 4.3 reveals multiple findings. First, we observe that there is a correlation between the magnitude dimension and the test accuracy—the lower the magnitude dimension, the higher the test accuracy. This result agrees with what has previously been observed in [Birdal et al., 2021, Simsekli et al., 2020], albeit from the perspective of *magnitude*, an invariant that is arguably simpler to compute and more interpretable in practice than persistent homology. Second, this holds across both MNIST and CIFAR10, and also across different architectures, which shows that even though the parameters differ considerably, the intrinsic dimension of different datasets can still be similar. To achieve good generalisation performance, it is important to keep the dimension as small as possible, without losing important representational features by collapsing them onto the same dimension.

4.6.2 Analysing and visualising network trajectories

Next, we want to investigate the effect on the magnitude function as the network is trained for more iterations. For this purpose, we select four values of t across the range $[0, 40]$, $t \in \{1.36, 6.78, 16.95, 30.51\}$ and we compute the effective number of models $\mathrm{Mag}(t\mathcal{W})$ to give us a glimpse into the space of training trajectories. We further fix the learning rate and vary the number of iterations. The experiments were performed with a learning rate of 0.1 for illustrative purpose. However, we note that this pattern holds for multiple learning rates. In Figure 4.4 we can see the resulting magnitude value at each of the selected scales of t, and the colour depicts



(a) (b) (c)

Figure 4.5: **The magnitude dimension, persistent homology dimension and ground truth for an α-Levy stable process.** In plot (a) we compare the $\dim_{\text{Mag}}\mathcal{W}$ and $\dim_{\text{PH}}\mathcal{W}$ against the number of iterations. There is strong correlation of 0.96, with statistically significant p-value ($p < 0.05$). In plot (b) we see the magnitude curves for different values of α. In plot (c) we see the magnitude curves of the α-stable Levy process for multiple values of α.

the number of iterations. We note that there is a concentration of red points in the upper left corner, indicating that higher test accuracy corresponds to a lower value of magnitude. Similarly, for the blue points, the higher value of magnitude is linked to worse test set performance. This pattern holds across both MNIST and CIFAR10. Moreover, the lower the test accuracy, the higher the magnitude value, implying that models with lower magnitude values generalise better. This result intuitively makes sense. When the magnitude value is small, the space looks like a smaller number of points. With the increase of t, the magnitude converges to the cardinality of the set. The effective number of models in (a) equals approximately 2 for high test accuracy, which is an interesting phenomenon. This means that out of the 1000 models, for $t = 1.38$, the space looks like 2 models, indicating that there are 2 large clusters formed by the training trajectories. Surprisingly, this is not the case for network trained across 1000 iterations, with the lowest test accuracy. In that case, the space of models looks like 10 points and is more scattered.

Note that the model with high test accuracy has smaller magnitude even at larger scales, suggesting that the models exhibit some sort of clustering behavior. In the magnitude terminology, the effective number of distinct models is smaller when the network generalises better; a "good optimisation trajectory" has a lower magnitude than a "bad optimisation trajectory"; the learning algorithm is implicitly optimising the magnitude, by clustering the models into a small number

of clusters. Throughout our experiments we thus observed strong correlation between magnitude itself and the test accuracy, which persists across the different scales, learning rates and datasets. Therefore, the main insight from the results in Figure 4.4 is that a small effective number of models is good for generalisation and it indicates that there is a clustering pattern.

4.6.3 Similarities between $\dim_{\mathrm{Mag}}\mathcal{W}$ and $\dim_{\mathrm{PH}}\mathcal{W}$

In Figure 4.5(a), we demonstrate that there is a strong correlation between $\dim_{\mathrm{Mag}}\mathcal{W}$ and $\dim_{\mathrm{PH}}\mathcal{W}$ on MNIST. Although there is a good correspondence between the two, $\dim_{\mathrm{PH}}\mathcal{W}$ is more difficult to interpret. In order to give some interpretation, the authors had to take a step back and produce the persistent diagrams which were used to calculate the specified dimension. However, due to the fact that they have to sample the space to estimate the dimension, these calculations involve a big number of different diagrams, hence linking $\dim_{\mathrm{PH}}\mathcal{W}$ with the original trajectory is not so straightforward. On the other hand, there is a more clear connection between magnitude and the magnitude dimension and the appearance of clusters of weight parameters.

Ablation study. In Figure 4.5 we see the results from an ablation study. We simulate data from a process similar to the weight trajectories with known ground truth, using a d-dimensional α-stable Levy process [Seshadri and West, 1982], where we take $d = 10$, and compare the values of the magnitude dimension to the fractal dimension. Further, we compare it with the persistent homology dimension, which as we have proven in Theorem 4.4.1, is the same as the magnitude dimension. The purpose of this study is to empirically evaluate this theoretical result. As seen in Figure 4.5(b), our dimension highly correlates with the ground truth. It seems that it is consistently lower than the true dimension by approximately 0.3. Our previous theoretical result implies equality between the two quantities in a compact space, but our empirical results show some discrepancy. This could be due to the sampling of points used for the computation of $\dim_{\mathrm{Mag}}\mathcal{W}$, as it is sensitive to the sampling. Increasing the sample size and using approximation methods as those outlined in Chapter 2 of this work could improve the estimate. In future work this difference can be investigated further.

Nonetheless, the high statistically significant correlation indicates that the magnitude dimension is indeed very close to the fractal dimension for a process

which resembles the iterations of SGD. This gives us high confidence that the magnitude dimension measures exactly what it is supposed to measure.

4.7 Conclusion

Our work provides the first connection between magnitude and the generalisation error. We proved a theoretical result linking the magnitude dimension of the optimisation trajectories and the test accuracy. We verified our results experimentally by exploring the evolution of magnitude across multiple experimental settings and datasets. We have further expanded our understanding about the clustering property of the weight trajectory: through both theoretical and empirical results, we demonstrated that models with better generalisation error tend to cluster more, compared to models with worse test performance, which are of higher magnitude and more spread out. This phenomenon has been previously described [Brüel-Gabrielsson et al., 2019, Birdal et al., 2021], and in this work our observations supplement it. In future work, we will improve on the estimation of the magnitude dimension and investigate the use of magnitude as a regulariser. Moreover, by using magnitude itself, we demonstrated that we can monitor the performance of the neural network without a validation set. In future work we can explore magnitude as an early stopping criteria, similar to Rieck et al. [2019]. Furthermore, from our empirical ablation study, the magnitude dimension seems to be consistently lower than the ground truth, which will be investigated further. In particular, we want to consider to what extent it is possible to improve on the generalisation bound in Theorem 4.4.2.

Chapter 5

Topological Generalization Bounds for Discrete-Time Stochastic Optimization Algorithms

We present a novel set of rigorous and computationally efficient topology-based complexity notions that exhibit a strong correlation with the generalization gap in modern deep neural networks (DNNs). DNNs show remarkable generalization properties, yet the source of these capabilities remains elusive, defying the established statistical learning theory. Recent studies have revealed that properties of training trajectories can be indicative of generalization. Building on this insight, state-of-the-art methods have leveraged the topology of these trajectories, particularly their fractal dimension, to quantify generalization. Most existing works compute this quantity by assuming continuous- or infinite-time training dynamics, complicating the development of practical estimators capable of accurately predicting generalization without access to test data. In this chapter, we respect the discrete-time nature of training trajectories and investigate the underlying topological quantities that can be amenable to topological data analysis tools. This leads to a new family of reliable topological complexity measures that provably bound the generalization error, eliminating the need for restrictive geometric assumptions. These measures are computationally friendly, enabling us to propose simple yet effective algorithms for computing generalization indices. Moreover, our flexible framework can be extended to different domains, tasks,

and architectures. Our experimental results demonstrate that our new complexity measures correlate highly with generalization error in industry-standards architectures such as transformers and deep graph networks. Our approach consistently outperforms existing topological bounds across a wide range of datasets, models, and optimizers, highlighting the practical relevance and effectiveness of our complexity measures.

5.1 Introduction

Generalization, a hallmark of model efficacy, is one of the most fundamental attributes for certifying any machine learning model. Modern deep neural networks (DNN) display remarkable generalization abilities that defy the current wisdom of machine learning (ML) theory [Zhang et al., 2017, 2021]. The notion can be formalized through the *risk* minimization problem, which consists of minimizing the function:

$$\mathcal{R}(w) := \mathbb{E}_{z\sim\mu_z}\left[\ell(w,z)\right], \tag{i}$$

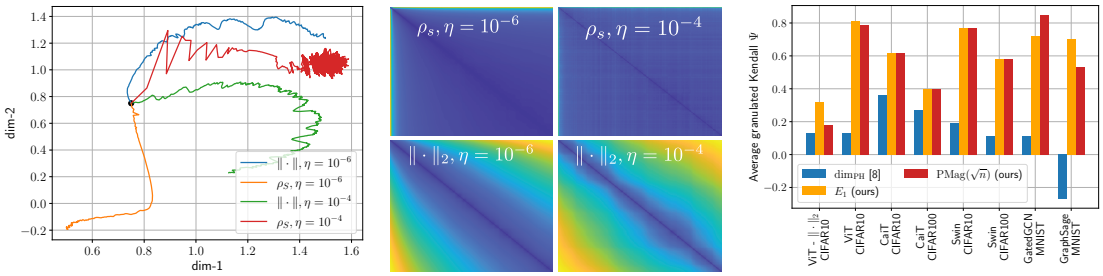
where $z \in \mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ denotes the data, distributed according to a probability distribution μ_z on the data space $\mathcal{Z}$. In practice, as μ_z is unknown, ML algorithms focus on minimizing the empirical risk,

$$\widehat{\mathcal{R}}_S(w) = \frac{1}{n}\sum_{i=1}^{n}\ell(w,z_i), \tag{ii}$$

where $S := (z_1,\ldots,z_n) \sim \mu_z^{\otimes n}$ are independent samples from μ_z. In many applications, the minimization of (ii) is achieved by discrete stochastic optimization algorithms, such as stochastic gradient descent (SGD) or the ADAM [Kingma and Ba, 2017] method. Such algorithms generate a sequence of iterates in $\mathbb{R}^d$, denoted $\mathcal{W}_S := \{w_k\}_{k\geq 0}$, which depends on the data S, the initialization $w_0 \in \mathbb{R}^d$, and some additional randomness U, *e.g.*, the random batch indices in SGD. The *generalization error* characterizing the model's performance is then defined as:

$$G_S(w_k) := \mathcal{R}(w_k) - \widehat{\mathcal{R}}_S(w_k). \tag{iii}$$

The empirical risk (ii) typically has numerous local minima, which raises the question of how to characterize their generalization properties. Recently, training trajectories (*cf.*, Figure 5.1a) have been shown to be paramount to answer this



(**a**) MDS Trajectory Embeddings (**b**) Distance Matrices of Trajectories (**c**) Our Topological Generalization Measures

Figure 5.1: We devise a novel class of complexity measures that capture the topological properties of discrete training trajectories. These generalization bounds correlate highly with the test performance for a variety of deep networks, data domains and datasets. Figure shows different trajectories (**a**) embedded using multi-dimensional scaling based on the distance-matrices (**b**) computed using either the Euclidean distance ($\|\cdot\|_2$) between weights as in [Birdal et al., 2021] or via the loss-induced pseudo-metric (ρ_S) as in [Dupuis et al., 2023]. (**c**) plots the *average granulated Kendall coefficients* for two of our generalization measures ($\boldsymbol{E}_\alpha$ and $\mathbf{PMag}(\sqrt{n})$) in comparison to the state-of-the-art persistent homology dimensions [Birdal et al., 2021, Dupuis et al., 2023] for a range of models, datasets, and domains, revealing significant gains and practical relevance.

question [Xu et al., 2023, Fu et al., 2023]. Indeed, these trajectories can quantify the quality of a local minimum in a compact way, because they depend simultaneously on the algorithm, the hyperparameters, and the data, which is crucial for obtaining satisfactory bounds [Gastpar et al., 2023]. A wide family of trajectory-dependent bounds has been developed [Xu et al., 2023, Fu et al., 2023, Lyu et al., 2023, Arora et al., 2019, Humayun et al., 2023]. For instance, several results on stochastic gradient Langevin dynamics [Mou et al., 2017, Pensia et al., 2018, Luo et al., 2022], continuous Langevin dynamics [Mou et al., 2017] and SGD [Neu et al., 2021] take into account the impact of the whole trajectory on the generalization error.

Parallel to these developments, several studies have brought to light the empirical links between topological properties of DNNs and their generalization performance [Naitzat et al., 2020, Magai and Ayzenberg, 2022, Pérez-Fernández et al., 2021, Rieck et al., 2018, Watanabe and Yamana, 2022], hereby making new connections with topological data analysis (TDA) tools [Adams and Moy, 2021]. These studies focus on the structural changes across the different layers of the network [Magai, 2023] or on the final trained network [Pérez-Fernández et al.,

2021, Rieck et al., 2018, Watanabe and Yamana, 2022], and are almost exclusively empirical. This partially inspired a new class of trajectory-dependent bounds focusing on topological properties of the trajectories. In particular, recent studies [Simsekli et al., 2020, Dupuis et al., 2023, Hodgkinson et al., 2022, Dupuis et al., 2024, Birdal et al., 2021] and Chapter 4 have proposed to relate the generalization error to various kinds of intrinsic *fractal* dimensions [Falconer, 2014, Mattila, 1999b] that characterize the learning trajectory. Informally, these bounds provide the guarantee that with probability at least $1-\zeta$, we have:[1]

$$\sup_{w\in\mathcal{W}_S} G_S(w) \lesssim \sqrt{\frac{\dim(\mathcal{W}_S)+\mathrm{IT}+\log(1/\zeta)}{n}}, \tag{iv}$$

where $\dim(\mathcal{W}_S)$ denotes various equivalent fractal dimensions, in particular the persistent homology dimension (PH-dim) [Birdal et al., 2021, Dupuis et al., 2023] and the magnitude dimension (Chapter 4). The term IT is an information-theoretic quantity that takes different forms among different studies. Despite providing rigorous links between the topology of the trajectory and generalization, these bounds have major drawbacks. First and foremost, as noted in [Sefidgaran et al., 2022, Sefidgaran and Zaidi, 2024, Camuto et al., 2021], fractal-trajectory bounds, such as Equation (iv), do not apply to discrete-time algorithms. This creates a discrepancy between these theoretical results and the TDA-inspired methods to numerically evaluate them on commonly used discrete algorithms [Birdal et al., 2021, Dupuis et al., 2023] and Chapter 4. Additionally, existing bounds rely on very intricate geometric assumptions, such as Ahlfors-regularity [Simsekli et al., 2020, Hodgkinson et al., 2022] or geometric stability [Dupuis et al., 2023], that are not realistic in a practical, discrete setting.

Previous attempts were made to address this discretization issue. Specifically, under the assumption that the training dynamics possess a stationary measure $\mu_{w|S}^{\infty}$ for $T\to\infty$ (T is the number of iterations), it was shown in [Camuto et al., 2021] that with probability $1-\zeta$ over $S\sim\mu_z^{\otimes n}$ and $w\sim\mu_{w|S}^{\infty}$:

$$G_S(w) \lesssim \sqrt{\frac{\dim(\mu_{w|S})+\mathrm{IT}+\log(1/\zeta)}{n}}, \tag{v}$$

where $\dim(\mu_{w|S})$ corresponds to the fractal dimension of the measure μ_w (see [Pesin, 2008] for formal definitions). While this was an important step, this bound

[1] We use $\lesssim$ in informal statements to indicate that absolute constants and/or small terms are missing.

only becomes practically relevant when the number of iterations grows to infinity, which is never attained in real-life experiments. Other attempts make use of so-called finite fractal dimensions [Sachs et al., 2023] or fine properties of the Markov transition kernels associated with the dynamics [Hodgkinson et al., 2022]. However, these studies also rely on impractical assumptions and involve intricate quantities which make them not amenable to numerical evaluation.

Despite the theoretical limitations of existing topology-dependent generalization bounds, TDA-inspired tools have been developed to numerically estimate the proposed intrinsic dimensions in practical settings. Two particular methods have emerged and successfully demonstrate correlation with the generalization error, based on *persistent homology* [Birdal et al., 2021, Dupuis et al., 2023] (PH-dim) and *metric space magnitude* (Chapter 4) (magnitude dimension); these two dimensions are equivalent for compact metric spaces (as shown in Chapter 4). Because of the limitations discussed above, existing theories do not account for these experiments, conducted with finite-time discrete algorithms. Moreover, existing empirical studies [Birdal et al., 2021, Dupuis et al., 2023, Simsekli et al., 2020] and Chapter 4 only consider very simple models and small (image) datasets. Because of their lack of theoretical foundations, it is not clear whether they could be extended to more practical setups.

Contributions In this chapter, we investigate the building blocks of PH and magnitude dimensions, in order to propose new topology-inspired generalization bounds that rigorously apply to widely used discrete-time stochastic optimization algorithms, and experimentally test our new topological complexities[2] on practically relevant DNN architectures. Our detailed contributions are as follows:

- We start by establishing the first theoretical links between generalization and a new kind of computationally thrifty topological complexity measure, the *α-weighted lifetime sums* [Schweinhart, 2020, 2021].
- We propose and elaborate on another novel topological complexity, *positive magnitude* (**PMag**), a slightly modified version of magnitude [Leinster, 2013, Meckes, 2013]. We rigorously link **PMag** with the generalization error, by relying on a new proof technique. Overall, our generalization bounds, rooted in

[2]Our term "topological complexity" should not be confused with the homonym topological invariant.

TDA, admit the following generic form:

$$\sup_{w \in \mathcal{W}_S} G_S(w) \lesssim \sqrt{\frac{(\text{Topological complexity}) + \text{IT} + \log(1/\zeta)}{n}}.$$

- We then provide a flexible computational implementation based upon dissimilarity measures between neural nets (Figure 5.1b), which enables quantifying generalization across different architectures and models, without the need for domain or problem-specific analysis as done in [Kiani et al., 2024, Behboodi et al., 2022].
- Unlike existing trajectory-based studies [Birdal et al., 2021, Dupuis et al., 2023] operating on small models, our experimental evaluation is extensive. We consider several vision transformers [Dosovitskiy et al., 2021] and graph neural networks (GNN) [Gori et al., 2005] trained on multiple datasets spanning regular and irregular data domains (*cf.* Figure 5.1c). Our findings robustly demonstrate that the novel topological generalization measures we introduce exhibit a strong correlation with test performance across diverse architectures, hyperparameters, and data modalities actually used in practice.

5.2 Technical Background

5.2.0.1 Information-theoretic quantities

The following definition is a precise definition of the total mutual information term that appears in our main theoretical results. The reader may consult [van Erven and Harremoës, 2014, Hodgkinson et al., 2022, Dupuis et al., 2024] for further information on this notion.

Definition 5.2.1 (Total mutual information)**.** Let X and Y be two random elements defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ (note that the codomains of X and Y may be distinct). We define the total mutual information between X and Y by the following formula:

$$\mathrm{I}_\infty(X, Y) = \log\left(\sup_A \frac{\mathbb{P}_{X,Y}(A)}{\mathbb{P}_X \otimes \mathbb{P}_Y(A)}\right).$$

Such a term has already been used in the fractal-based generalization literature [Hodgkinson et al., 2022, Dupuis et al., 2024]. Other works used intricate variants of this total mutual information term [Dupuis et al., 2023, Birdal et al., 2021,

Camuto et al., 2021] and Chapter 4. We stress the fact that our proposed bounds are simpler.

5.2.0.2 Topological complexities background

Our generalization indicators will be based upon α-weighted lifetime sums and magnitude, capturing different topological features, as we shortly dicsuss below. Let (X, ρ) be a finite pseudometric space.

α-weighted lifetime sums. Persistent homology (PH) is an important concept in the analysis of geometric complexes [Boissonat et al., 2018]. We focus on the persistent homology of degree 0 (PH^0). Informally, it consists in tracking the "connected components" of a finite set at different scales. We provide in Sections A.3.1.3 and A.3.1.4 an overview of these notions. For simplicity, we present here an equivalent formulation of the α-weighted lifetime sums based on minimum spanning trees (MST) [Kozma et al., 2006, Schweinhart, 2020].

A tree over X is a connected acyclic undirected graph (a set of edges) whose vertices are the points in X. Given an edge e linking the points a and b, we define its *cost* as $|e| := \rho(a, b)$. An MST $\mathcal{T}$ on X is a tree minimizing the *total cost* $\sum_{e\in\mathcal{T}} |e|$. The α-weighted lifetime sums $\boldsymbol{E}^{\rho}_{\alpha}$ are then written as:

$$\forall \alpha \geq 0,\ \boldsymbol{E}^{\rho}_{\alpha}(X) := \sum\nolimits_{e\in\mathcal{T}} |e|^{\alpha}.$$

The celebrated *persistent homology dimension* (PH-dim) [Adams et al., 2020], of a compact pseudometric space (A, ρ) is then defined as

$$\dim^{\rho}_{\mathrm{PH}}(A) = \inf_{\alpha\geq 0} \{\exists C > 0, \forall Y \subset X \text{ finite}, \boldsymbol{E}_{\alpha}(Y) \leq C\}.$$

The PH-dim has been proven to be related to generalization error for different pseudometrics ρ [Birdal et al., 2021, Dupuis et al., 2023].

Magnitude. Magnitude is a recently introduced topological invariant [Leinster, 2013] which encodes many important invariants from geometric measure theory and integral geometry [Leinster, 2013, Meckes, 2013, 2015]. Magnitude can be interpreted as the effective number of distinct points in a space [Leinster, 2013]. For $s > 0$, we define a *weighting* of the modified space $(X, s\rho)$ as a map $\beta : X \to \mathbb{R}$, such that $\forall a \in X,\ \sum_{b\in X} e^{-s\rho(a,b)}\beta(b) = 1$. Given such a weighting β, the magnitude function of $(X, s\rho)$ is defined as

$$\mathrm{Mag}^{\rho}(sX) := \sum\nolimits_{a\in X} \beta(a). \tag{vi}$$

The parameter $s > 0$ should be interpreted as a "scale" through which we look at the set (X, ρ). Note that magnitude is usually defined in metric spaces; we show in Section 5.2.1 that we can seamlessly extend it to the pseudometric setting. Magnitude can be extended to (infinite) compact spaces [Leinster, 2013, Meckes, 2013] and, as for PH, an intrinsic dimension, the *magnitude dimension*, can be defined from magnitude by the formula $\dim^{\rho}_{\mathrm{Mag}}(A) = \lim_{s\to\infty} \frac{\log \mathrm{Mag}(sA)}{\log(s)}$. It is known that $\dim^{\rho}_{\mathrm{PH}}$ and $\dim^{\rho}_{\mathrm{Mag}}$ coincide for compact metric spaces [Meckes, 2015, Schweinhart, 2020] and Chapter 4. As a result, $\dim^{\rho}_{\mathrm{Mag}}$ has also been proposed as a topological generalization indicator (Chapter 4).

5.2.1 Magnitude in pseudometric spaces

In this section, we fix (X, ρ) a finite pseudometric space. We denote by ${}^{X}/_{\sim}$ its metric identification and by $\pi : X \longrightarrow {}^{X}/_{\sim}$ the canonical projection.

We define the notion of *metric identification*, which will be used in several of the following subsections. This is the same setting that was used in [Dupuis et al., 2023] to naturally extend the persistent homology dimension to pseudometric spaces.

Definition 5.2.2 (Metric identification)**.** Let (X, ρ) be a pseudometric space. We can define an equivalence relation on X by $a \sim b \iff \rho(a, b) = 0$. The associated quotient space, which is denoted ${}^{X}/_{\sim}$ is a metric space for the naturally induced metric, which we still denote ρ.[3] We will also use the canonical projection,

$$\pi : X \longrightarrow {}^{X}/_{\sim}.$$

These notations will be used throughout the text.

We directly extend Definition A.3.17 to the pseudometric case. In order for this definition to make sense in our context, we first need to verify that it provides a well-posed definition of magnitude. This follows from the following lemma.

Lemma 5.2.3. *We assume that the finite pseudometric space (X, ρ) has magnitude. Then magnitude is independent of the choice of weighting.*

Proof. The proof is straightforward and identical to the metric case. Let β, β' be two weightings, we have:

$$\sum_{a\in X} \beta(a) = \sum_{a\in X}\sum_{b\in X} e^{-\rho(a,b)}\beta'(b)\beta(a) = \sum_{b\in X} \beta'(b) \sum_{a\in X} e^{-\rho(a,b)}\beta(a) = \sum_{b\in X} \beta'(b).$$

[3] Indeed, if $a \sim b$, then we have $\forall c \in X,\ \rho(a, c) = \rho(b, c)$.

The key ingredient to the proof is the definition of magnitude weightings, which states that a weighting of X is a function $\beta : X \longrightarrow \mathbb{R}$ such that

$$\forall a \in X,\ \sum_{b \in X} e^{-\rho(a,b)} \beta(b) = 1.$$

□

In the following theorem, we show that magnitude is invariant through metric identification.

Theorem 5.2.4 (Invariance of magnitude through metric identification)**.** *X has magnitude if and only if ${}^{X}/_{\sim}$ has magnitude, in which case we have:*

$$\mathrm{Mag}(X) = \mathrm{Mag}\left({}^{X}/_{\sim}\right).$$

Proof. We decompose X into equivalence classes as:

$$X = \coprod_{\bar{a} \in {}^{X}/_{\sim}} \bar{a} =: \coprod_{i \in I} \bar{a}_i,$$

where $\coprod$ denotes disjoint union and the points $(a_i)_{i \in I} \in X^I$ represent each equivalence class. We denote by $\bar{a}$ the equivalence class of $a \in X$.

Let $\beta : X \longrightarrow \mathbb{R}$ be any function. We have:

$$\forall a \in X,\ \sum_{b \in X} e^{-\rho(a,b)} \beta(b) = \sum_{i \in I} e^{-\rho(\bar{a},\bar{a}_i)} \sum_{b \in \bar{a}_i} \beta(b). \tag{vii}$$

$\Longrightarrow$: If X has magnitude, then we take β to be a weighting of X, we define:

$$\forall \bar{a} \in {}^{X}/_{\sim},\ \bar{\beta}(\bar{a}) := \sum_{b \in \bar{a}} \beta(b).$$

By Equation (x), $\bar{\beta}$ is a weighting of ${}^{X}/_{\sim}$.

$\Longleftarrow$: if $\bar{\beta}$ is a weighting of ${}^{X}/_{\sim}$, then we define:

$$\forall a \in X,\ \beta(a) := \frac{1}{|\bar{a}|} \bar{\beta}(\bar{a}),$$

where $|\bar{a}|$ denotes the cardinality of $\bar{a}$. By Equation (x), β is a weighting of X. □

Total mutual information. Prior intrinsic dimension-based studies relied on "mixing" assumptions ([Simsekli et al., 2020, Assumption H5], [Birdal et al., 2021, Assumption H1], [Sefidgaran and Zaidi, 2024, Camuto et al., 2021]) or various mutual information terms [Hodgkinson et al., 2022, Dupuis et al., 2023]

to take into account the statistical dependence between the data and the training trajectory. Recently, a new framework was proposed in [Dupuis et al., 2024] to unify these approaches by proving data-dependent uniform generalization bounds using simpler and smaller information-theoretic (IT) terms. By leveraging these methods, we derive new generalization bounds involving the same IT terms for all our introduced topological complexities. More precisely, they take the form of a *total mutual information* between the data S and the training trajectory $\mathcal{W}_S$. This term is denoted $\mathrm{I}_\infty(S, \mathcal{W}_S)$ and measures the dependence between S and $\mathcal{W}$. We refer to Appendix A.3.1.1 and [Hodgkinson et al., 2022, van Erven and Harremoës, 2014] for exact definitions.

5.3 Main Theoretical Results

We now introduce our learning-theoretic setup (Section 5.3.1) before delving into our main theoretical results in Sections 5.3.2 and 5.3.3.

5.3.1 Mathematical setup

Random trajectories. The primary goal of our theory is to prove uniform generalization bounds over the training trajectory $\{w_k,\ k \geq 0\}$. We are mostly interested in the behavior near local minima of $\widehat{\mathcal{R}}_S$. To this end, we observe the trajectory between iterations τ and T, where $\tau \in \mathbb{N}$ is the number of iterations before reaching (near) a local minimum and $T \geq \tau$ is the total number of iterations. Therefore, we consider the set $\mathcal{W}_{\tau \to T} := \{w_i,\ \tau \leq i \leq T\}$, which we call the *random trajectory*. Note that $\mathcal{W}_{\tau \to T}$ is a *set*, *i.e.*, it does not contain any information about the time-dependence. Moreover, our setup allows the random times τ and T to depend on the data S through the choice of a stopping criterion as opposed to being fixed predetermined times.

General Lipschitz conditions. The topological quantities described in Section 5.2, as well as the intrinsic dimensions introduced in prior works [Simsekli et al., 2020, Birdal et al., 2021, **?**, Dupuis et al., 2023, 2024], require a notion of distance between parameters (in $\mathbb{R}^d$) to be computed. In the case of fractal-based generalization bounds, two cases have already been considered: the Euclidean distance [Simsekli et al., 2020] and the data-dependent pseudometric defined in [Dupuis et al., 2023]. In our work, we emphasize that both examples are particular cases of a more general family of pseudometrics on the parameter space

$\mathbb{R}^d$. In order to fully characterize this family of pseudometrics, we define the data-dependent map $\boldsymbol{L}_S : \mathbb{R}^d \longrightarrow \mathbb{R}^n$ by $\boldsymbol{L}_S(w) = (\ell(w, z_1), \ldots, \ell(w, z_n))$. To fit into our framework, a pseudometric must satisfy the following general Lipschitz condition.

Definition 5.3.1 ((q, L, ρ)-Lipschitz continuity)**.** For any pseudo-metric ρ on $\mathbb{R}^d$ and $q \geq 1$, we will say that ℓ is (q, L, ρ)-Lipschitz in w when $\forall w, w' \in \mathbb{R}^d$, $\|\boldsymbol{L}_S(w) - \boldsymbol{L}_S(w')\|_q \leq L n^{1/q} \rho(w, w')$.

A wide variety of distances have been proposed to compare the weights of two DNNs [Donnat and Holmes, 2018]. The above condition restricts our analysis to a family of pseudometrics containing the following examples.

Example 5.3.2 (Data-dependent pseudometrics). For any $p \geq 1$, we define the pseudometrics $\rho_S^{(p)}(w, w') := n^{-1/p} \|\boldsymbol{L}_S(w) - \boldsymbol{L}_S(w')\|_p$. The case $\rho_S^{(1)}$ corresponds to the "data-dependent pseudometric" used in [Dupuis et al., 2023]; we will denote it $\rho_S := \rho_S^{(1)}$.

Example 5.3.3 (Euclidean distance). If $\ell(w, z)$ is L-Lipschitz continuous in w, *i.e.*, $|\ell(w, z) - \ell(w', z)| \leq L\|w - w'\|$ for all z, then ℓ is $(p, L, \|\cdot\|_2)$-Lipschitz continuous for every $p \geq 1$.

Assumptions. Given an (q, L, ρ)-Lipschitz continuous (pseudo-)metric, our approach relies only on a single assumption of a bounded loss function. For the case of the pseudometric ρ_S (Example 5.3.2), this assumption is already made in [Dupuis et al., 2023, 2024].

Assumption 5.3.4. We assume that the loss ℓ is bounded in $[0, B]$, with $B > 0$ a constant.

The boundedness of ℓ is classically assumed in the fractal / TDA literature [Dupuis et al., 2023, Hodgkinson et al., 2022, Dupuis et al., 2024]. In [Dupuis et al., 2023], it is shown that the proposed theory seems to be experimentally valid even for unbounded losses. Our experimental findings suggest that this observation also applies to our work.

5.3.2 Persistent homology related generalization bounds

In contrast to all existing fractal dimension-based bounds [Simsekli et al., 2020, Birdal et al., 2021, Camuto et al., 2021, Dupuis et al., 2023], we propose new

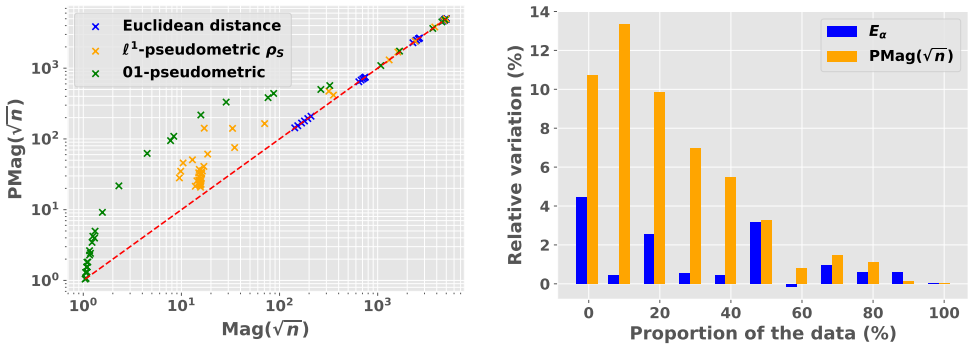
generalization bounds that apply to practical discrete stochastic optimizers with a finite number of iterations. To this end, our key idea involves replacing the intrinsic dimension with intermediary quantities that are used to compute them numerically. Following [Birdal et al., 2021] and Chapter 4, this points us towards the two quantities, $\boldsymbol{E}_\alpha$ and Mag, defined in Section 5.2. We are now ready to state the first generalization bound in terms of the α-weighted lifetime sums, where we denote $\boldsymbol{E}_\alpha^\rho$ for $\boldsymbol{E}_\alpha^\rho(\mathcal{W}_{\tau\to T})$.

Theorem 5.3.5. *Let ρ be a pseudometric on $\mathbb{R}^d$. Supposes that Assumption A.3.31 holds and that ℓ is (q, L, ρ)-Lipschitz, for $q \geq 1$. Then, for all $\alpha \in [0, 1]$, with probability at least $1 - \zeta$, we have:*

$$\sup_{\tau\leq i\leq T} G_S(w_i) \leq 2B\sqrt{\frac{2\log \boldsymbol{E}_\alpha^\rho + \alpha \log\left(\frac{8L\sqrt{n}}{B}\right)}{n}} + \frac{2B}{\sqrt{n}} + 3B\sqrt{\frac{\mathrm{I}_\infty(S, \mathcal{W}_{\tau\to T}) + \log(1/\zeta)}{2n}}.$$

The term $\mathrm{I}_\infty(S, \mathcal{W}_{\tau\to T})$ is the total mutual information (MI) term that is defined in Sections 5.2 and A.3.1.1. It measures the statistical dependence between the random set $\mathcal{W}_{\tau\to T}$ and the data $S \sim \mu_z^{\otimes n}$. Such MI terms appear in previous works related to fractal-based generalization bounds [Simsekli et al., 2020, Camuto et al., 2021, Dupuis et al., 2023, Hodgkinson et al., 2022]. Our proof technique, presented in A.3.2.5, makes use of a recently introduced PAC-Bayesian framework for random sets [Dupuis et al., 2024] to introduce this MI term. It is also shown in [Dupuis et al., 2024] that the MI term $\mathrm{I}_\infty(S, \mathcal{W}_{\tau\to T})$ is tighter than those appearing in the aforementioned works.

We highlight the fact that Theorem 5.3.5 is fundamentally different from the persistent homology dimension (PH-dim) based bounds studied in [Birdal et al., 2021, Dupuis et al., 2023]. Indeed, while the growth of $\boldsymbol{E}_\alpha$ for increasing finite subsets of the trajectory are used in [Birdal et al., 2021] to estimate the PH-dim, it does not provide any formal link between the generalization error and the value of $\boldsymbol{E}_\alpha$. Therefore, the above theorem could not be cast as a corollary of these previous studies. Another important characteristic of the above theorem (as well as the results of Section 5.3.3) is to be non-asymptotic, *i.e.*, it is true for every $n \in \mathbb{N}^*$. This is an improvement over the fractal dimensions-based bounds presented in [Simsekli et al., 2020, Birdal et al., 2021, Dupuis et al., 2023, 2024].



(a) (b)

Figure 5.2: *Left:* Comparison of Mag and **PMag** (for $s = \sqrt{n}$), for different (pseudo)metrics (ViT on CIFAR10). *Right:* relative variation of the quantities $\boldsymbol{E}_\alpha(\mathcal{W}_{\tau\to T})$ and $\mathrm{Mag}(\sqrt{n}\mathcal{W}_{\tau\to T})$, with respect to the proportion of the data used to estimated $\rho_S^{(1)}$ (ViT on CIFAR10).

5.3.3 Positive magnitude (PMag) and related generalization bounds

Recent preliminary experimental results displayed a correlation between the generalization error of DNNs and magnitude (Chapter 4). To provide a theoretical justification for this behavior, it would be tempting to mimic the proof of Theorem 5.3.5 and build on existing covering arguments. However, while lower bounds of magnitude in terms of covering numbers have been derived in [Meckes, 2015], they appear to be impractical in our case. Another possibility would be to use the magnitude dimension bounds of Chapter 4. Yet, this could not apply to our finite and discrete setting where the dimension is 0. Hence, we identify a new quantity, closely related to magnitude, while being more relevant to learning theory. With the notations of Section 5.2, we fix a finite metric space (X, ρ) and a weighting $\beta_s : X \longrightarrow \mathbb{R}$ of $(X, s\rho)$, where $s > 0$ is a "scale" parameter. We define the positive magnitude as

$$\forall s > 0,\ \mathbf{PMag}^\rho(sX) := \sum\nolimits_{a\in X} \beta_s(a)_+, \tag{viii}$$

where $x_+ := \max(x, 0)$ denotes the positive part of x.

After introducing positive magnitude, one might ask how similar are magnitude and positive magnitude, and how can positive magnitude be interpreted. **PMag** is a purely technical quantity introduced so that the generalisation bound can be proved. We have also demonstrated that for a large value of the scale

parameter, both magnitude and positive magnitude coincide, meaning that their interpretations are the same for large enough scale parameter. For small values of the scale parameter, the theoretical and interpretative properties are to be studied in future work.

5.3.4 Definition of positive magnitude in the pseudometric case

Let us extend our new notion of *positive magnitude* in finite pseudometric spaces. This is a rather complicated task. Indeed we need to ensure that the positive magnitude is independent of the choice of weighting, which is not true in general. For this reason, we restrict our definition to pseudometric spaces whose metric identification is positive definite and we choose one particular weighting.

Definition 5.3.6 (Positive magnitude in finite pseudometric spaces)**.** Let (X, ρ) be a finite pseudometric space whose metric identification ${}^{X}/_{\sim}$ is positive definite. Let $\bar{\beta} : {}^{X}/_{\sim} \longrightarrow \mathbb{R}$ be a weighting of ${}^{X}/_{\sim}$, then we define the positive magnitude of X, denoted **PMag**, by:

$$\mathbf{PMag}(X) = \sum_{\bar{x} \in {}^{X}/_{\sim}} \bar{\beta}(\bar{x})_+,$$

where $x_+ := \max(x, 0)$ denotes the positive part of x. We will say that X admits a positive magnitude if its metric identification ${}^{X}/_{\sim}$ is positive definite.

Note that ${}^{X}/_{\sim}$ admits a unique weighting because it is positive definite. However, X still admits several weightings in general. The above definition ensures that the definition of positive magnitude is independent of any choice of weighting. For the need of our proofs, we will need to introduce weightings in pseudometric spaces, whose sums of positive parts yield the positive magnitude. This is possible by using the following definition, which corresponds to a "good" choice of weighting in finite pseudometric spaces.

Definition 5.3.7 (Canonical weighting)**.** Let (X, ρ) be a finite pseudometric space whose metric identification ${}^{X}/_{\sim}$ is positive definite. Let $\bar{\beta} : {}^{X}/_{\sim} \longrightarrow \mathbb{R}$ be a weighting of ${}^{X}/_{\sim}$, we define the *canonical weighting* $\beta^0 : X \longrightarrow \mathbb{R}$ on X by:

$$\forall a \in X,\ \beta^0(a) := \frac{1}{|\pi(a)|}\bar{\beta}(\pi(a)),$$

where $\pi : X \longrightarrow {}^{X}/_{\sim}$ is the canonical surjection.

The following lemma is then obvious but crucial to some of our theoretical results.

Lemma 5.3.8. *With the notation of the previous definition, we have:*

$$\mathbf{PMag}(X) = \sum_{x \in X} \beta^0(x)_+.$$

The next proposition is a consequence of Theorem A.3.19, it shows that the pseudometrics considered in practice in our work (and in our experiments) admit a positive magnitude.

Proposition 5.3.9. *Let $p \in [1, 2]$ and $S \in \mathcal{Z}^n$, then every finite subset of $(\mathbb{R}^d, \rho_S^{(p)})$ admits a positive magnitude, and therefore it also has a canonical weighting.*

Proof. Let $\mathcal{W} := \{w_1, \ldots, w_N\}$ be a finite set in $\mathbb{R}^d$. We have

$$\|\boldsymbol{L}_S(w) - \boldsymbol{L}_S(w')\|_p = n^{1/p} \rho_S^{(p)}(w, w').$$

Therefore, if we denote by $\bar{w}$ the equivalence class of w in the metric identification, it is clear that $\bar{w} = \bar{w'} \iff \boldsymbol{L}_S(w) = \boldsymbol{L}_S(w')$. Hence, the map $\varphi_S := n^{-1/p} \boldsymbol{L}_S$ naturally extends to an isometry between metric spaces:

$$\mathcal{W}/_{\sim} \xrightarrow{\sim} \varphi_S(\mathcal{W}) \underset{\text{finite}}{\subset} \mathbb{R}^n.$$

By Theorem A.3.19, the finite set $\varphi_S(\mathcal{W})$ is positive definite, hence it is also the case of $\mathcal{W}/_{\sim}$. Therefore $\mathcal{W}$ admits a positive magnitude by definition. □

Based on a new theoretical approach, we prove that the positive magnitude can be used to upper bound the generalization error (see the proof in A.3.2.7). This leads to the following theorem:

Theorem 5.3.10. *Let ρ be a pseudometric such that $(\mathcal{W}, \lambda\rho)$ admits a positive magnitude (according to Definition A.3.27) for every $\lambda > 0$. We assume that ℓ is (q, L, ρ)-Lipschitz continuous with $q \geq 1$. Then, for any $s > 0$, we have with probability at least $1 - \zeta$ that*

$$\sup_{\tau \leq i \leq T} G_S(w_i) \leq \frac{2}{s} \log \mathbf{PMag}^{\rho}\left(Ls\mathcal{W}_{\tau \to T}\right) + s\frac{B^2}{n} + 3B\sqrt{\frac{\mathrm{I}_\infty(S, \mathcal{W}_{\tau \to T}) + \log(1/\zeta)}{2n}}.$$

The IT term (I_∞) in the above result is the same as in Theorem 5.3.5. Given a fixed (finite) set $\mathcal{W}$ and a big enough s, we establish $\mathrm{Mag}(s\mathcal{W}) = \mathbf{PMag}(s\mathcal{W})$.

Moreover, we present in Figure 5.2(a) an empirical comparison of Mag and **PMag**, showing a small and almost monotonic relation between both quantities. Therefore, Theorem 5.3.10 may be seen as the first theoretical justification of the empirical relationship between magnitude and the generalization error observed in Chapter 4.

A natural choice for the scale s would be $s \approx \sqrt{n}$, ensuring a convergence rate in $n^{-1/2}$. However, our empirical evaluations (see Section 5.5, in particular, Table 5.1) revealed that small values of s (we typically use $s = 10^{-2}$) can also provide good correlation with the generalization error. This could be explained by the fact that $\mathbf{PMag}(s\mathcal{W}) \to 1$ as $s \to 0$, *i.e.*, the bound may not diverge when $s \to 0$. For our topological complexities to be computationally efficient, we focus our experiments on fixed values of s (in $\{\sqrt{n}, 10^{-2}\}$). We will omit the trajectory and denote $\mathrm{Mag}(s)$ and $\mathbf{PMag}(s)$.

5.4 Computational Considerations

We now detail the numerical estimation of the topological complexities mentioned above.

Computation of $\boldsymbol{E}_\alpha$. We compute $\boldsymbol{E}_\alpha$ by using the `giotto-ph` library introduced in [Pérez et al., 2021, Bauer, 2021a]. This setup is inspired by PH frameworks used in [Birdal et al., 2021, Dupuis et al., 2023]. This technique uses the equivalent formulation of $\boldsymbol{E}_\alpha$ in terms of PH (see A.3.1.3 for details). Theorem 5.3.5, and its proof (presented in A.3.2.6) suggest that the relevant value of α is 1; similar to [Birdal et al., 2021], this is what we used in our experiments.

Computation of Mag **and PMag**. Different methods exist to evaluate magnitude, as demonstrated in Chapter 6. We use the Krylov approximation method [Salim, 2021], which is based on pre-conditioned conjugate gradient iteration, implemented in the Python library `krypy.linsys.Cg` to solve for the magnitude weights. We then sum over the weights to compute Mag, and sum over the positive weights to obtain **PMag**.

Distance matrix estimation.. Given a finite set (*i.e.*, a trajectory) $\mathcal{W} \subset \mathbb{R}^d$, the calculation of our topological complexities requires computing the *distance matrix* $D_\rho := (\rho(w, w'))_{w,w' \in \mathcal{W}}$. For large DNNs, this may become challenging. Depending on ρ, we propose the following solutions.

- Case 1: If ρ is the Euclidean distance on $\mathbb{R}^d$, for large DNNs (in our case for the

Model-dataset	ViT-CIFAR10				Swin-CIFAR100				GraphSage-MNIST				GatedGCN-MNIST			
Compl.-Metric	ψ_{LR}	ψ_{BS}	$\mathbf{\Psi}$	τ	ψ_{LR}	ψ_{BS}	$\mathbf{\Psi}$	τ	ψ_{LR}	ψ_{BS}	$\mathbf{\Psi}$	τ	ψ_{LR}	ψ_{BS}	$\mathbf{\Psi}$	τ
$\dim_{\text{PH}}$ - ρ_S [Dupuis et al., 2023]	0.93	-0.67	0.13	0.61	0.69	-0.47	0.11	0.50	-0.28	-0.26	-0.27	-0.35	0.15	0.07	0.11	-0.06
$\text{Mag}(\sqrt{n})$ - ρ_S	0.68	0.62	0.65	0.64	0.56	0.47	0.51	0.53	0.69	0.71	**0.70**	**0.79**	0.85	0.97	**0.91**	**0.88**
Mag(0.01) - ρ_S	0.41	0.58	0.50	0.47	0.31	0.47	0.39	0.33	0.24	0.10	0.17	0.36	0.35	0.35	0.35	0.49
PMag$(\sqrt{n})$ - ρ_S	0.91	0.67	0.79	0.85	0.69	0.47	0.58	0.62	0.59	0.46	0.53	0.59	0.73	0.97	0.85	0.84
PMag(0.01) - ρ_S	0.86	0.40	0.50	0.80	0.71	0.58	**0.64**	**0.68**	0.24	0.10	0.17	0.36	0.35	0.35	0.35	0.49
$\boldsymbol{E}_\alpha$ - ρ_S	0.95	0.67	**0.81**	**0.86**	0.69	0.47	0.58	0.62	0.67	0.74	**0.70**	0.77	0.48	0.97	0.72	0.74
$\dim_{\text{PH}}$ - $\lVert\cdot\rVert_2$ [Birdal et al., 2021]	0.93	-0.67	0.13	0.61	0.69	-0.47	0.34	0.51	0.32	0.81	0.56	0.51	-0.12	0.70	0.29	0.33
$\text{Mag}(\sqrt{n})$ - $\lVert\cdot\rVert_2$ (Chapter 4)	0.95	-0.59	0.13	0.73	0.71	-0.57	0.07	0.53	0.75	0.77	**0.76**	**0.61**	0.77	0.76	0.77	0.52
Mag(0.01) - $\lVert\cdot\rVert_2$ (Chapter 4)	0.95	-0.60	0.17	0.72	0.69	-0.44	0.12	0.53	0.75	0.74	0.74	0.60	0.77	0.42	0.60	0.47
PMag$(\sqrt{n})$ - $\lVert\cdot\rVert_2$	0.95	-0.59	0.18	0.73	0.71	-0.57	0.07	0.53	0.75	0.74	0.74	0.60	0.77	0.93	**0.85**	**0.54**
PMag(0.01) - $\lVert\cdot\rVert_2$	0.55	0.71	**0.63**	0.58	0.64	0.51	0.58	0.46	0.75	-0.05	0.35	0.51	0.60	-0.47	0.06	0.26
$\boldsymbol{E}_\alpha$ - $\lVert\cdot\rVert_2$	0.95	-0.31	0.32	**0.76**	0.63	0.75	**0.74**	**0.74**	0.75	0.74	0.74	0.60	0.77	0.93	0.84	**0.54**
$\dim_{\text{PH}}$ - 01 [Dupuis et al., 2023]	0.95	-0.20	0.37	0.72	0.64	0.04	0.34	0.51	0.0	-0.13	-0.07	0.0	0.14	0.00	0.07	0.00
$\text{Mag}(\sqrt{n})$ - 01	0.95	0.67	**0.81**	0.88	0.69	0.47	**0.58**	**0.62**	0.64	0.68	**0.66**	**0.75**	0.78	0.85	**0.82**	**0.82**
Mag(0.01) - 01	0.84	0.33	0.59	0.75	0.61	0.27	0.44	0.50	0.13	0.11	0.12	0.26	0.10	0.10	0.10	0.25
PMag$(\sqrt{n})$ - 01	0.95	0.64	0.80	**0.89**	0.69	0.47	**0.58**	**0.62**	0.63	0.65	0.64	0.74	0.76	0.83	0.79	0.80
PMag(0.01) - 01	0.84	0.36	0.60	0.76	0.65	0.49	0.57	0.54	0.13	0.11	0.12	0.26	0.10	0.10	0.10	0.25
$\boldsymbol{E}_\alpha$ - 01	0.95	0.67	**0.81**	0.87	0.69	0.47	**0.58**	0.61	0.63	0.68	**0.66**	0.74	0.78	0.85	**0.82**	**0.82**

Table 5.1: Correlation coefficients associated with the different topological complexities.

transformer experiments) storing the whole trajectory is challenging. In that case, we use sparse random projections inspired by the Johnson-Lindenstrauss lemma [Vershynin, 2020] to project the trajectories onto a lower-dimensional subspace. We use the implementation in `scikit-learn` [Pedregosa et al., 2011a] so that, with high probability, the relative variation of the distance matrices is at most 5%, see A.3.1.7 for details.

- Case 2: If ρ is of the form $\rho_S^{(q)}$ as in Example 5.3.2, then the computation of D_ρ requires the evaluation of the model on the entire dataset at each iteration, which becomes intractable for large DNNs. In [Dupuis et al., 2023, Figure 3], the authors show that the PH-dim based on the pseudometric $\rho_S = \rho_S^{(1)}$ is very robust to a random subsampling of a training dataset, *i.e.* when ρ_S is replaced by ρ_B with $B \subseteq S$ and $|B|/|S| \ll 1$. Figure 5.2(b) shows that $\boldsymbol{E}_\alpha$ and positive magnitude are also robust to this subsampling. We mainly used $|B|/|S| = 10\%$. We refer the reader to A.3.3.2 for details.

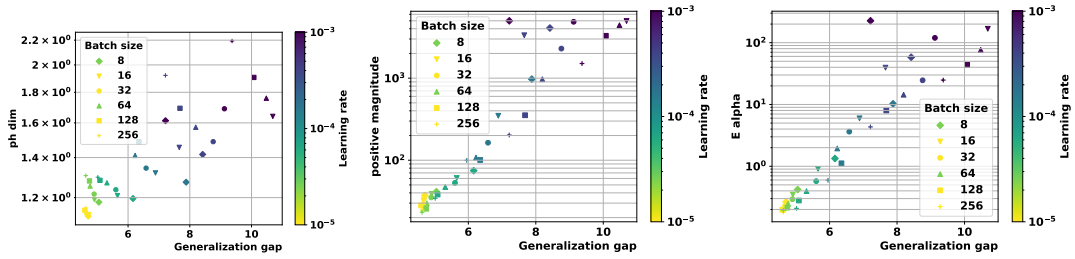
Generalization error. Our theory, like many trajectory-based studies [Simsekli et al., 2020, Birdal et al., 2021, Dupuis et al., 2023] and Chapter 4, predicts upper bounds on the worst-case generalization error over the trajectory $\mathcal{W}_{\tau\to T}$. Yet, experiments in previous works mainly reported the error at the last iteration. To

estimate the worst-case error in a computationally feasible way, we periodically evaluated the test risk between times τ and T (with a period of 100 iterations) and reported (`worst test risk - final train risk`) as the error in our experiments. This is consistent as we start the trajectory $\mathcal{W}_{\tau\to T}$ from a weight w_τ already in a local minimum of the empirical risk. Our main conclusions are still valid if the final generalization gap is used. This observation, which is to the best of our knowledge new, is briefly discussed in A.3.4.1.

5.5 Empirical Analysis

Setup. Given a DNN and a dataset, we start from a pre-trained weight vector w_τ, yielding high training accuracy on classification tasks. By varying the learning rate (η) and the batch size (b), we define a grid of 6×6 hyperparameters. For each pair (η, b), we compute the training trajectory $\mathcal{W}_{\tau\to T}$ for 5×10^3 iterations. Unless specified, we use the ADAM optimizer [Kingma and Ba, 2017]. Based on the set $\mathcal{W}_{\tau\to T}$, we estimate distance matrices as described in Section 5.4. For the sake of clarity, we focus on 3 relevant pseudometrics: (i) the Euclidean distance $\|\cdot\|_2$ as in [Birdal et al., 2021], (ii) the data-dependent pseudometric ρ_S, used in [Dupuis et al., 2023], and (iii) the 01-loss distance. For (ii), ρ_S is computed based on the *surrogate* loss used in training (*e.g.*, the cross-entropy loss), while the reported generalization error is always based on *accuracy gap* (01-loss), which is of interest in most applications (see Section 5.4). For the last one (iii) ρ is defined as in Example 5.3.2, but with ℓ being the 01-loss; we call it 01-pseudometric and denote it by 01 in the tables. This last setup matches exactly our theoretical requirements.

In terms of DNN architectures, we focus on practically relevant models, while previous studies mainly considered small networks [Birdal et al., 2021, Hodgkinson et al., 2022, Dupuis et al., 2023, Sefidgaran and Zaidi, 2024]. We examine two different families of architectures. The first family consists of vision transformers (ViT [Touvron et al., 2021a], CaiT [Touvron et al., 2021b], Swin [Liu et al., 2021], see Table A.1), each evaluated on both the CIFAR10 [Krizhevsky et al., 2014] and CIFAR100 [Krizhevsky, 2009] datasets. Moreover, we also tested our theory on graph neural networks (GNN) architectures, namely GatedGCN [Bresson and Laurent, 2017] and GraphSage [Hamilton et al., 2017] trained on the Super-pixel MNIST dataset [Dwivedi et al., 2023]. To the best of our knowledge, this is the first time these kinds of topological complexities have been evaluated on



(a) (b) (c)

Figure 5.3: ρ_S-based complexity measures vs. generalization gap for a ViT trained on CIFAR10: $\dim_{\mathrm{PH}}$ (*left*), $\mathbf{PMag}(\sqrt{n})$ (*middle*), and $\boldsymbol{E}_1$ (*right*).

transformers and GNNs. We ran the experiments on 18 NVIDIA 2080Ti (11 GB) GPUs.

Granulated Kendall's coefficients.. We assess the correlation between our complexities and the generalization error by using the granulated Kendall's coefficients (GKC) [Jiang et al., 2019]. While the classical Kendall's coefficients (KC) [Kendall, 1938a] (denoted τ) measures the correlation between two quantities, it may fail to capture their causal relationship. Instead, one "granulated" coefficient is defined in [Jiang et al., 2019] for each hyperparameter (*i.e.*, $\boldsymbol{\psi}_{\mathrm{LR}}$ for η and $\boldsymbol{\psi}_{\mathrm{BS}}$ for b); it measures the correlation when only this hyperparameter is varying. In Table 5.1, we report τ, $\boldsymbol{\psi}_{\mathrm{LR}}$ and $\boldsymbol{\psi}_{\mathrm{BS}}$, and the averaged GKC, $\boldsymbol{\Psi} := (\boldsymbol{\psi}_{\mathrm{LR}} + \boldsymbol{\psi}_{\mathrm{BS}})/2$, for several models, datasets and topological complexities. In Figures 5.4(a) and 5.4(b), we represent our topological complexities in the plane $(\boldsymbol{\psi}_{\mathrm{BS}}, \boldsymbol{\psi}_{\mathrm{LR}})$; the red square indicates the region of best correlation (the coefficients are in $[-1, 1]$, their sign is the sign of the correlation).

5.5.1 Analysis

As explained above, we focus our main experiments on the quantities $\boldsymbol{E}_1$, $\mathrm{Mag}(\sqrt{n})$, $\mathbf{PMag}(\sqrt{n})$, $\mathrm{Mag}(10^{-2})$ and $\mathbf{PMag}(10^{-2})$, each computed for the 3 pseudometrics discussed above ($\|\cdot\|_2$, ρ_S, 01). In the interest of comparison, we also compute the PH-dim (proposed in [Birdal et al., 2021] for the $\|\cdot\|_2$ and in [Dupuis et al., 2023] for ρ_S), which is thus tested for the first time on transformers and GNNs.

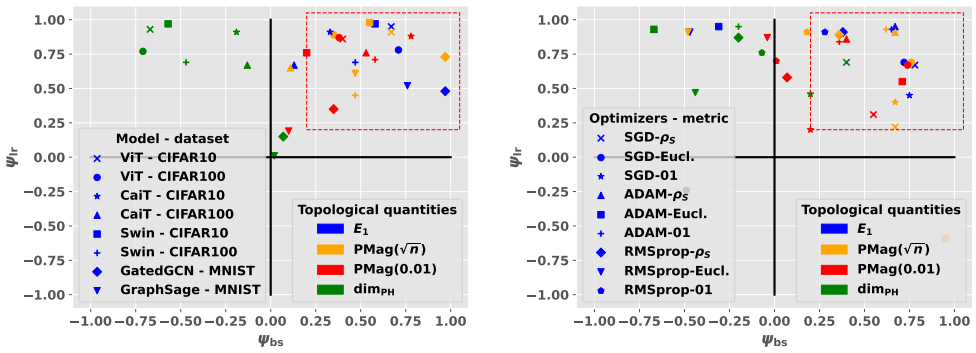
Performance on vision transformers. We see in Table 5.1 and Figure A.10 that our proposed topological complexities consistently outperform the PH dimensions across several vision transformer models and datasets. This suggests that PH-dim, previously tested only on small architectures, is less scalable to industry-standards models with more parameters. Figure 5.4(a), including all (`model`, `dataset`)

pairs for the pseudometric ρ_S, reveals important observations. First, we notice that the GKC of our topological complexities are both positive and close to 1, indicating that they are indeed good measures of generalization. We note that for most models and datasets, $\dim_{\mathrm{PH}}$ has a small or negative $\boldsymbol{\psi}_{\mathrm{BS}}$, indicating that it has less ability to explain generalization for varying batch-sizes. As it was observed in [Dupuis et al., 2023] for PH-dim, our complexities computed from the pseudometric ρ_S correlate very well with the generalization gap while this gap is based on the 01 loss.

Performance on GNNs. An important aspect of our framework is the ability to seamlessly encapsulate different data domains. In particular, the possibility of using different pseudometrics can help define topological complexities that naturally take into account the internal symmetries of GNNs, without any model-specific analysis [Kiani et al., 2024, Behboodi et al., 2022]. The results of Table 5.1 and Figure 5.4(a) confirm that our proposed topological complexities outperform PH-dim and correlate strongly with the generalization error for GNNs. Additionally, it may be observed that $\mathrm{Mag}(\sqrt{n})$ performs significantly well for GNNs, and in particular better than $\mathbf{PMag}(\sqrt{n})$. This points us towards the idea that further theory would be desirable to formally relate magnitude to the generalization error in that case[4].

Comparison of the topological complexities. In Table 5.1 and Figures A.10 and 5.4(a), it can be seen that $\boldsymbol{E}_1$ and $\mathbf{PMag}(\sqrt{n})$ perform equally well for the image and graph experiments across multiple datasets, models, and data domains. We see in Table 5.1 that most topological complexities perform better with data-dependent metrics (*i.e.*, ρ_S and 01) than with the Euclidean distance, for transformer-based experiments. This extends results obtained for PH-dim in [Dupuis et al., 2023], for smaller architectures. However, the poor performance of Euclidean-based complexities may also be partially caused by the projections applied to the Euclidean distance matrices to make them memory-wise computable (see Section 5.4). This is a remaining limitation of our algorithms. On the other hand, the 01 and ρ_S data-dependent pseudometrics seem to yield similar performance in all experiments.

[4]We shall underline that, while Mag with the Euclidean distance was empirically proposed as a complexity measure in Chapter 4, a theoretical justification for Mag results in Table 5.1 is still missing for moderate values of s.



(a) (b)

Figure 5.4: Granulated Kendall coefficients for several models, datasets and topological quantities. Note that our framework is directly applicable to graph networks.

Ablations. In Figure 5.4(b), we reveal that changing the optimizer has little effect on the observed correlation (for the same model and dataset). Interestingly, we note that the PH-dim, computed with pseudometric ρ_S and obtained from the SGD trajectories, exhibits high GKCs. This observation agrees with the results in [Dupuis et al., 2023]. Figure A.10 further displays the typical behavior of several topological complexities for ViT and CIFAR10. In addition to the correlation of our proposed complexities being stronger than for the PH-dim, we observe that $\boldsymbol{E}_\alpha$ and $\mathbf{PMag}(\sqrt{n})$ seem to better correlate with the generalization gap for small learning rates. Finally, it is consistently observed in Table 5.1 and Figures 5.4(a) and 5.4(b) that using a relatively high value of the (positive) magnitude scale ($s = \sqrt{n}$) yields better correlations than small values ($s = 10^{-2}$). However, both cases still provide satisfying correlation, comforting the robustness of magnitude as a generalization indicator.

Due to limited space, we present all the correlation coefficient of one transformer model ViT for CIFAR10 and Swin for CIFAR100 in Table 5.1 as illustrative examples for each dataset. The remaining results appear in the Appendix, Tables A.3, A.5, A.2 and A.4, and they all follow a similar trend. Further empirical results and illustrations of this behavior are provided in Appendix A.3.4.

5.6 Conclusion

In this chapter, we proved novel generalization bounds based on several topological complexities coming from TDA, namely α-weighted lifetime sums and a new variant of metric space magnitude, which we called positive magnitude. Compared to

previous studies, we require fewer assumptions and operate in a discrete setting in which our proposed quantities are fully computable. Our algorithms are flexible enough to be seamlessly integrated with diverse data domains and tasks. These advantages of our framework allowed us to create a computationally cheap experimental setup, as close as possible to the theoretical setup. We thus provided a comprehensive suite of experiments with several industry-relevant architectures across vision transformers and graph neural networks, which have not been explored yet in this literature. We show that our proposed topological complexities correlate well with the generalization error, outperforming the previously studied intrinsic dimensions.

Limitations & future work. The main limitation of our theory is the lack of understanding of the IT terms, while they are still smaller than most prior works. Moreover, a better understanding of the behavior of positive magnitude for small values of the scale factor s would be a necessary improvement. Regarding our experiments, a refinement of the estimation techniques of the topological complexities would be beneficial. Despite experimenting with practically relevant architectures, our future works also include scaling up our empirical analysis to include larger models and datasets, in particular large language models, which are still beyond the scope of this study.

Chapter 6

Metric Space Magnitude for Evaluating the Diversity of Latent Representations

The *magnitude* of a metric space is a novel invariant that provides a measure of the 'effective size' of a space across multiple scales, while also capturing numerous geometrical properties, such as curvature, density, or entropy. We develop a family of magnitude-based measures of the intrinsic diversity of latent representations, formalising a novel notion of dissimilarity between magnitude functions of finite metric spaces. Our measures are provably stable under perturbations of the data, can be efficiently calculated, and enable a rigorous multi-scale characterisation and comparison of latent representations. We show their utility and superior performance across different domains and tasks, including (i) the automated estimation of diversity, (ii) the detection of mode collapse, and (iii) the evaluation of generative models for text, image, and graph data.

6.1 Introduction

Diversity is a key concept in representation learning, referring to the relative abundance and distinctiveness of model outputs. Given the inherent complexity of deep learning models, the evaluation of diversity is thus crucial, enabling (i) the assessment of the *intrinsic richness* of latent representations, and (ii) the evaluation of the extent to which models are capable of *preserving* the properties of an input distribution. While the quantitative evaluation of generative models in

particular relies on assessing trade-offs between fidelity and diversity with regards to a known reference distribution, reference-free diversity measures are becoming increasingly relevant when a ground-truth distribution is unknown or intractable. However, reference-based diversity metrics such as *recall* are notoriously fallible, sensitive to parameter choices and therefore prone to incorrectly approximate the true data manifold, whereas reference-free diversity measures often rely on simple mean summaries that fail to pass basic consistency checks [Friedman and Dieng, 2023]. Thus, existing methods lack expressivity to fully capture what it means for a space to be diverse, resulting in a critical need for novel measures that are (i) theoretically motivated, (ii) robust to noise, and (iii) capable of encoding the intrinsic diversity of data across varying levels of similarity rather than at a single fixed threshold.

Our contributions. Addressing this need, we propose a novel family of diversity measures based on *metric space magnitude*, a mathematical invariant that captures numerous important multi-scale geometric characteristics of metric spaces, including curvature, density, and entropy of an input space. Metric space magnitude merely requires a notion of dissimilarity between data points, permitting it to operate on both *local* and *global* scales. Hence, magnitude is poised to compare latent spaces, yielding a compact holistic summary of diversity that satisfies relevant theoretical requirements. Our work is the first to (i) introduce magnitude as a general tool for evaluating the diversity of latent representations, and (ii) formalise a notion of difference between the magnitude of two spaces across multiple scales of similarity. We demonstrate that magnitude is stable and can detect curvature, highlighting its use as a multi-scale summary of the local and global geometry of data. Moreover, we empirically showcase the utility of our magnitude-based diversity measure across different modalities, namely text, image, and graph embeddings, for which we observe that our measure outperforms alternative embedding-based measures of intrinsic diversity. Finally, when a reference distribution is known, our magnitude-based notion of difference reliably detects *mode collapse* and *mode dropping*, thus assisting practitioners in model evaluation and selection. We further improve the efficiency of magnitude computations using a Cholesky decomposition method for computation.

6.2 Related Work

Latent representations and embeddings have become indispensable tools for analysing data types such as images, text, and graphs. As evidenced by LLMs, understanding semantic relationships in data requires meaningful embeddings. Our work focuses on improving representation-based diversity evaluation and we thus consider the role diversity plays in this context.

Diversity measures. Assessing generative model diversity remains a challenge irrespective of the domain [Theis et al., 2016], as ground truth reference distributions or labelled data are often unavailable, and human evaluation remains costly. Thus, there exists a need for interpretable, automated and unsupervised measures of intrinsic diversity. *Reference-free evaluation* is of particular importance for assessing generated text given the black-box-nature of LLMs [Celikyilmaz et al., 2020], but also applicable across modalities. Motivated by this, a varied collection of diversity measures has been proposed, many of which are task-, domain- or model-specific [Friedman and Dieng, 2023]; only a fraction of them are applicable to analysing latent representations specifically. The most flexible methods summarise intrinsic diversity using average pairwise dissimilarities like L^p distances or BERT-scores [Tevet and Berant, 2021]. More recently, Friedman and Dieng [2023] proposed the Vendi Score, inspired by principles from theoretical ecology. Other diversity measures are computed directly on embedding spaces, using e.g. the geometric mean of the standard deviation across each embedding dimension [Lai et al., 2020] or cluster-based measures [Du and Black, 2019]. However, as we explore in Section 6.3.1, none of these measures satisfy all theoretical guarantees required by an axiomatic approach to diversity, and they are limited in expressivity, providing only snapshots of diversity at a single fixed resolution. *Reference-based metrics* define diversity as the extent to which generated samples cover the full variability of the real data [Naeem et al., 2020]. Examples include the Fréchet Inception Distance (FID) or the Inception score (IS). However, they do not exclusively measure diversity but are also concerned with evaluating fidelity, i.e. the assessment of similarity between generated data and real data, making it unclear how single-number summaries such as FID and IS account for each aspect in the trade-off between diversity and quality. Thus, *precision* and *recall* have been suggested as more informative summary metrics [Sajjadi et al., 2018] and seen various improvements [Kynkäänniemi et al., 2019, Simon et al., 2019, Naeem et al.,

2020]. Unfortunately, as Naeem et al. [2020] show, even the improved versions of precision and recall fail to satisfy the useful conditions for strong evaluation metrics, such as (i) detecting identical reference and generated distributions, (ii) capturing mode dropping, and (iii) simplicity in selecting hyperparameters. To address these concerns, *density* and *coverage* have been proposed [Naeem et al., 2020]. Nevertheless, these metrics still rely on fixed-scale manifold approximations to assess diversity making them sensitive to parameter choices. By contrast, our magnitude-based measures have less stringent assumptions and can be defined in a parameter-free fashion.

Magnitude in machine learning. Since its introduction to measure biological diversity [Solow and Polasky, 1994], magnitude was formalised by Leinster [2013]. Nevertheless, despite strong geometric properties [Leinster, 2021], magnitude has only rarely been applied in a machine learning context. Recent publications started to bridge this gap, linking magnitude to boundary detection [Bunch et al., 2021], edge detection in images [Adamer et al., 2021], and the generalisation error of neural networks (Chapter 4), as well as demonstrating its utility for multi-objective optimisation [Huntsman, 2023]. However, the full potential of magnitude for measuring diversity remains largely unexplored since existing works ignore the nature of magnitude as an intrinsic multi-scale summary, which captures both local and global geometry and diversity of the data manifold. Our work is thus the first to leverage magnitude as a flexible, multi-scale measure of diversity in latent representations.

6.3 Methods

We first discuss the theoretical properties a suitable diversity measure should satisfy and then introduce metric space magnitude. Based on this, we outline our proposed method using magnitude for measuring the diversity of latent representation and its practical implementation.

6.3.1 Desiderata for Diversity Measures

Given the difficulty in defining diversity, diversity metrics never measure diversity itself, but rather quantify related ideas. Entropy-based approaches, including magnitude, in particular share close links to diversity, often favoured in ecology

for their computational benefits and agreement with fundamental axioms of diversity [Daly et al., 2018]. Following this axiomatic approach, we highlight the following key requirements [Leinster, 2021]:

- *Monotonicity in observations*: Including a new observation does not decrease diversity.
- *Twin property*: Including a duplicate observation already in the set does not change diversity.
- *Absence invariant*: Diversity only depends on the samples and features present in the dataset.
- *Multi-scale*: Diversity encodes both local and global trends in the data manifold.

This list is not conclusive. We observe that many diversity measures for evaluating representations in ML do not satisfy these requirements.

For example, average similarity (AvgSim), the most frequently-used diversity measure in ML, cannot capture nuances in diversity and fails even in simple toy scenarios [Friedman and Dieng, 2023]. Likewise, the geometric mean of the standard deviations across each embedding dimension [Lai et al., 2020, GMStds] is *not* absence invariant as it equals zero whenever an embedding feature is constant. Even the Vendi Score (VS) [Friedman and Dieng, 2023], a more purpose-built diversity measure, calculated as the exponential of the Shannon entropy of the eigenvalues of a normalised similarity matrix, shows undesirable behaviour under the inclusion of observations. Moreover, neither one of the aforementioned diversity measures fulfills the twin property nor monotonicity in observations [Leinster, 2021], leading to counter-intuitive behaviour when capturing changes in diversity. For example, an exact repetition of the reference data could be wrongly judged to be more diverse than a model that generates more samples with small but relevant deviations from the reference. This discussion thus points out a glaring need for more principled diversity measures. Further, we argue that diversity is a multi-scale trend that should describe the data manifold across multiple levels of similarity rather than rely on fixed-scale snapshots. Addressing this, *magnitude functions* are particularly promising candidates for improved diversity measures that inherently satisfy all desiderata listed above.

6.3.2 Magnitude for Evaluating Diversity

We note a few particularities about magnitude and magnitude functions before discussing diversity. First, for negative definite metrics d like the L^1 and L^2 distance, ζ_X is invertible [Feragen et al., 2015]. Subsequently, we assume that (X, d) permits the calculation of magnitude; in particular X must *not* have any duplicate points. Further, for $t \in (0, \infty)$, the magnitude function is defined for all but finitely many values of t [Leinster, 2013]. The magnitude function is also *continuous* [Meckes, 2015, Corollary 5.5][1] for negative definite metrics. For finite metric spaces, we have $\lim_{t\to\infty} \mathrm{Mag}(tX) = |X| = n$, i.e. the *cardinality* of X [Leinster, 2013, Proposition 2.2.6]. This limit behaviour exemplifies to what extent the magnitude function describes the diversity of a space as 'the effective number of points at scale t.'

In this Chapter, we extend magnitude functions to the domain $[0, \infty)$ by defining $\mathrm{Mag}_X(0) := 1$.[2] Intuitively, this extension means that any metric space, when viewed from far away, looks like a single point. Notice that neither Definition 2.1.1 nor Definition 2.1.7 explicitly require specific properties of a metric (like the triangle inequality) and we find magnitude computable for generalised distance functions, including cosine distances, provided the similarity matrix ζ_X is invertible. Figure 6.1 illustrates how magnitude functions measure the effective number of distinct points for toy data, thus describing their diversity. Moreover, it provides an overview of our diversity evaluation framework, which we will now introduce.

As a multi-scale geometric invariant, magnitude can be extended to evaluate the diversity of latent representations. Here, we are studying a set of latent representations $\mathcal{X} = \{X_1, X_2, \dots\}$, where each $X_i \in \mathcal{X}$ is a finite subset of some latent space sharing the same notion of distance, e.g. $X_i \subseteq \mathbb{R}^D$. Given a latent representation $X \in \mathcal{X}$, e.g. a text, image, or graph embedding, we can use the L^1 or L^2 distance as a metric or semi-metrics like the cosine distance. Based on the choice of metric, we can interpret $\mathrm{Mag}_X(t)$ as the effective number of points at scale t. In practice, this summarises how diverse points in the space are when observed at said scale factor. This multi-scale behaviour motivates us to propose a simple but expressive summary of a representation's magnitude function.

Definition 6.3.1 (Area under the magnitude function, MagArea)**.** Let X be a

[1] Mag_X is continuous for $t > t_{\mathrm{crit}}$, where t_{crit} is the supremum of its finitely many singularities.

[2] This assumes the so-called *one-point property*, i.e. $\lim_{t\to\infty} \mathrm{Mag}_X(0) = 1$, which was shown to hold generically for almost all finite metric spaces [Roff and Yoshinaga, 2023].

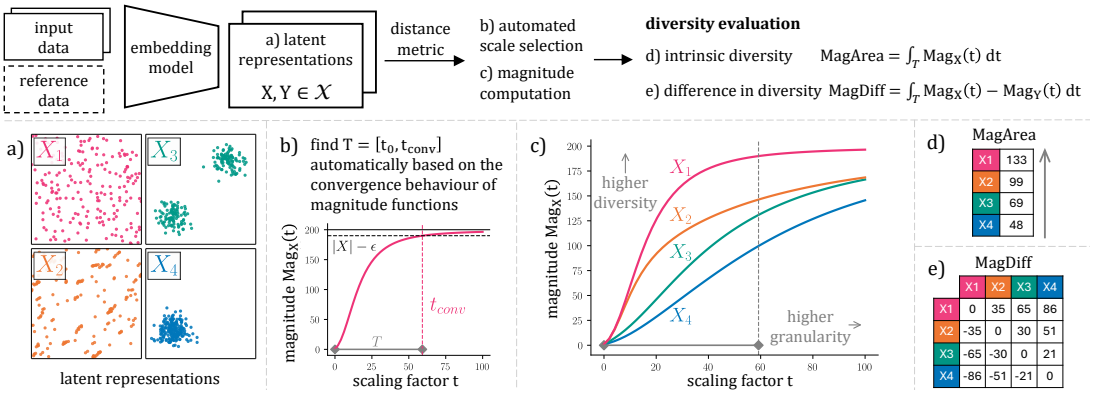


Figure 6.1: **Overview of our diversity evaluation pipeline.** (a) We start with an example of four latent spaces with 200 points, varying in diversity. Points in X_1 are from Poisson Process (uniform pattern), and the most diverse; points in X_2 are from a more clustered process, namely Hawkes Point process, and hence less diverse; point in X_3 are two Gaussians, and hence less diverse than X_2, and points in X_4 are from one Gaussian, and hence the least diverse out of all four. (b) The magnitude function measures the effective number of points at t, a scale of distance between observations. When the scale factor t almost equals zero, magnitude is close to 1, and a space effectively looks like one point. For large t, the number of effective points is noticeably higher and magnitude converges towards the cardinality. We find the approximate convergence scale, t_{conv}, at which magnitude almost equals the cardinality, and use it to define the evaluation interval T across which diversity changes most notably. (c) The more diverse the space, the higher the value of its magnitude function. By construction, X_1 is more diverse than X_2, X_3, and X_4, respectively. We leverage this behaviour to define novel multi-scale indicators of diversity. (d) Our proposed measure of intrinsic diversity, MagArea, summarises the area under each magnitude function for *reference-free* diversity evaluation. (e) In a *reference-based* setting, we assess the difference in diversity using MagDiff, the area between two magnitude functions.

metric space whose magnitude function $\mathrm{Mag}_X(t)$ has been evaluated across the interval $T = [t_0, t_{\mathrm{cut}}]$. We define the area under the magnitude function to be $\textsc{MagArea} := \int_{t_0}^{t_{\mathrm{cut}}} \mathrm{Mag}_X(t)dt$.

Moreover, we extend this proposed summary to measure the difference in diversity between two representations generated by the *same* (embedding) model. Notice that distances in these spaces are directly comparable and the respective magnitude functions can be compared across the same domain.

Definition 6.3.2 (Magnitude function difference, MagDiff)**.** Let X and Y be two metric spaces that share the same notion of distance. Assume the associated magnitude functions $\mathrm{Mag}_X(t)$ and $\mathrm{Mag}_Y(t)$ have been evaluated across the same interval $T = [t_0, t_{\mathrm{cut}}]$. We define the magnitude function difference to be $\textsc{MagDiff} := \int_{t_0}^{t_{\mathrm{cut}}} \left(\mathrm{Mag}_X(t) - \mathrm{Mag}_Y(t)\right) dt$.

We note that in Definition 6.3.1, we could obtain a difference of 0 even when the two functions are not exactly the same: in one region the difference might be positive, and in another negative, and vice-versa, and both can cancel out. In order to avoid this, we can improve the definition to be: $\textsc{MagDiff} := \int_{t_0}^{t_{\mathrm{cut}}} \left(|\,\mathrm{Mag}_X(t) - \mathrm{Mag}_Y(t)|\right) dt$. In practice, we control for this behaviour to avoid this scenario. We further note that one can propose alternative definitions capturing the difference between two magnitude functions using p-norms. More formally, the following quantity can be defined as

$$\|\,\mathrm{Mag}_X(t) - \mathrm{Mag}_Y(t)\|_p = \left(\int_{t_0}^{t_{\mathrm{cut}}} |\,\mathrm{Mag}_X(t) - \mathrm{Mag}_Y(t)|^p\, dt\right)^{1/p}.$$

The properties of this quantity are outisde the scope of this work, but it will be interesting in future studies to determine how and where they could be useful.

Definition 6.3.1 and Definition 6.3.2 constitute novel multi-scale approaches for summarising and comparing magnitude functions, leading to theoretically well-founded diversity measures. MagArea measures the cumulative value of magnitude summarising a space's intrinsic diversity while MagDiff measures the accumulated difference in diversity between two spaces. As we will later demonstrate in our experiments, integrating the changes in magnitude across a *range* of scale factors retains the desirable properties of single-scale magnitude, but yields more robust multi-scale summaries of diversity (see Appendix A.4.1 for an investigation of stability to perturbations). Furthermore, this comparison in terms of the effective number of points across scales remains directly interpretable.

6.3.3 Practical Usage

In order to use our magnitude metric for reference-free and reference-based diversity evaluation, we obviate the choice of evaluation interval using knowledge about the convergence behaviour of magnitude functions. As a consequence, our magnitude-based diversity measures do not require manual parameter selection. First, we define a magnitude function's convergence scale.

Definition 6.3.3 (Convergence scale, t_{conv})**.** Given a magnitude function $\text{Mag}_X(t)$, we define its approximate convergence scale as $t_{\text{conv}} \in \mathbb{R}$, with $\text{Mag}_X(t_{\text{conv}}) = |X| - \epsilon$ for some small $\epsilon > 0$. We set $\epsilon \leq 0.05|X|$ in this work.

This convergence scale thus indicates the resolution at which at least 95% of observations are recognised by magnitude as being distinct. After reaching this convergence scale, we know that magnitude functions and hence diversity can increase by at most ϵ based on the convergence of magnitude towards the cardinality as illustrated in Figure 6.1. In practice, however, we find that *all* relevant changes in diversity happen at smaller scales of distance when individual points are not yet clearly separated. We thus choose the convergence scale defined in Definition 6.3.3 to be the upper bound of the evaluation interval T to determine the most informative range of scales. More formally, we define $T = [t_{\text{min}}, t_{\text{max}}]$, and by convergence scale we refer to the upper bound of$T$, which is t_{max}.

We then find the convergence scale using numeric root-finding procedures.

When comparing the intrinsic diversity of multiple embeddings *without* a reference dataset, we compute MagArea across $T = [0, t_{\text{cut}}]$ and choose t_{cut} to equal the median of the convergence scales of the embeddings. Taking the median here provides a stable compromise between the convergence behaviour of all functions. For *reference-based comparisons*, we simply calculate MagDiff, the difference between the magnitude functions, across $T = [0, t_{\text{ref}}]$ where t_{ref} is the convergence scale of the reference embedding. In practice, we *approximate* the integrals in Definition 6.3.1 and Definition 6.3.2 via numerical integration across evenly-spaced scales sampled from the evaluation interval T. Choosing the number of scales is a trade-off between *accuracy* and *computational performance* as computational costs increase linearly with the number of times magnitude is evaluated. In terms of implementations, we also improve the efficiency of magnitude computations using a Cholesky decomposition (see Appendix A.4.4 for more details). Together with our automated scale-selection procedure, we thus

overcome the main algorithmic hurdles that hitherto prevented the wider use of magnitude functions.

6.3.4 Limitations

MagDiff is a reference-free measure of intrinsic diversity, but does not measure *fidelity*. Being referece-free is a crucial characteristic. It means that to calculate MagDiff, you do not need to compare your data to a "ground truth" or "reference" dataset. You can assess the diversity of a single dataset (e.g., generated images, text, or latent representations) without needing a corresponding "real" dataset to compare against. This is highly valuable in situations where a true reference distribution is unknown, intractable, or difficult to obtain. It should therefore not be interpreted in isolation, but jointly with coverage-based metrics, for instance. In the context of evaluating generative models (like those that create images, text, or other data), coverage-based metrics refer to measures that assess how well the generated data covers or spans the entire range of diversity present in the real-world data distribution. Coverage-based metrics tell you about the completeness of what the model should generate. If coverage is high, it means the generated samples adequately represent the full spectrum of the real data. In order to see why MagDiff should be combined with coverage-based metrics, one of the following could happen: if MagDiff is high, but the coverage is low, you can get a model which generates very diverse, but irrelevant or off-distribution samples (e.g., diverse noise). On the contrary, if you get low MagDiff and high Coverage, the model could generate samples that cover the real distribution well, but they are all very similar (e.g., generates all animal types, but each animal type only has one very specific example, lacking intra-class diversity). Both of these scenarios are undesirable and ideally you will have both high MagDiff and coverage, in which case the model will generate a wide variety of samples, and that variety accurately reflects the diversity of the real world.

Moreover, while we improve the efficiency of magnitude computations (see Appendix A.4.4) compared to previous implementations [Bunch et al., 2021], thus making magnitude calculations feasible for practical analyses, novel approximation methods would be required to enable scaling to hundreds of thousands of observations. Finally, we focus on evaluating representation-based diversity and show that, given a latent representation, magnitude yields a better notion of diversity than cur-

rent embedding-based methods. We do not investigate whether embedding-based similarities are outperformed by alternative task- or domain-specific similarities. Instead, our evaluation relies on the utility of embedding models and assumes that latent spaces encode useful/realistic relationships between samples.

6.4 Experiments

Our experiments demonstrate how magnitude leads to a better understanding of representational diversity. We show the following results: (i) Magnitude functions capture the curvature of a space. (ii) Magnitude functions are interpretable measures of the intrinsic diversity of embeddings, yielding superior results than other diversity measures when predicting the diversity of sentence embeddings across different text-generation tasks. (iii) Magnitude functions characterise and distinguish latent representations of large language models. (iv) Magnitude functions successfully detect mode dropping in distributions of image, and graph embeddings, while also reliably detecting mode collapse in graph embeddings. We subsequently use MagArea in reference-free settings to characterise intrinsic diversity (i, ii), while using MagDiff for reference-based comparisons (iii, iv).

6.4.1 Magnitude Functions Summarise Geometry

Magnitude functions encode the 'shape,' i.e. the geometry that is characteristic of the intrinsic data manifold, by capturing curvature and diversity. Scalar curvature estimation is an important task in numerous domains like computer vision, computational geometry, and computer-aided design. Previous works have shown that alternative multi-scale methods, such as *persistent homology*, are able to detect curvature [Turkes et al., 2022, Bubenik et al., 2020]. Here, we demonstrate that the magnitude function is capable of achieving comparable performance, using simpler methods and only a single feature.

The curvature experiments builds on the approach by Turkes et al. [2022]. We generate a unit disks D_κ of surfaces of constant curvature κ, with 3 cases: the first one is when $\kappa = 0$ (we then have the Euclidean plane), $\kappa < 0$ (we have a space of negative curvature, the Poincare disk model of the hyperbolic plane), $\kappa > 0$ (sphere with radius $1/\sqrt{\kappa}$). We vary the curvature κ to be in the interval $[-2, 2]$. For each value of κ, we construct point clouds by sampling 500 points from D_κ. We

Table 6.1: **Magnitude estimates curvature.** MagArea outperforms more complex methods [Turkes et al., 2022] using a *single feature*.

Method	MSE (↓)
SVR (selected PH features)	0.27 ± 0.07
SVR (PH vectorisation)	0.17 ± 0.05
SVR (all PH features)	0.16 ± 0.03
SVR (distance matrices)	0.24 ± 0.04
MLP (shallow)	1.15 ± 0.52
MLP (deep)	1.56 ± 0.68
MagArea (quantile)	0.10 ± 0.05
MagArea (piecewise linear)	0.05 ± 0.03

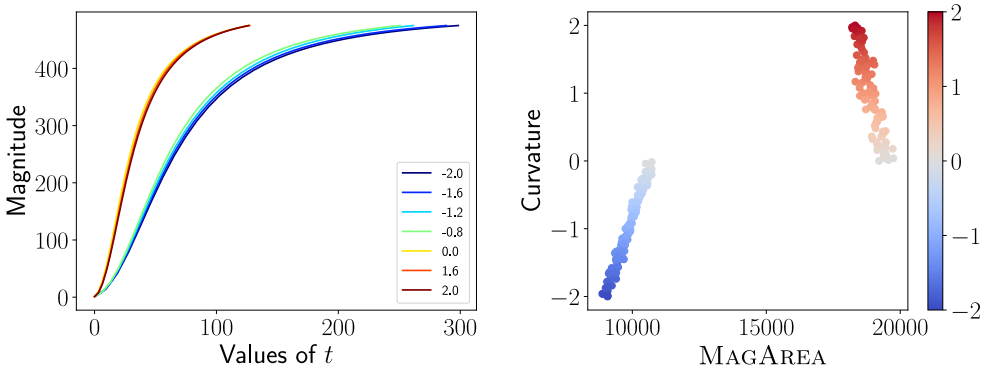


Figure 6.2: **Magnitude detects curvature.** Left: Magnitude functions for unit disks with varying curvature between $[-2, 2]$. Right: Curvature is positively correlated with MagArea, indicating that it serves as a expressive predictor.

generate 201 surfaces with equally spaced curvature in the interval $[-2, 2]$. Then, we compute magnitude for each space using Euclidean distance and 30 evenly spaced intervals until the scale $t_{\text{cut}} = 73$. For the results reported in Table 6.1 we further apply 5-fold cross-validation. We first train a quantile regression model on the MagArea after applying polynomial feature transformation of degree 2 to the training data suspecting a quadratic-looking relationship between MagArea and curvature after exploratory analysis. Further, we compare this to piecwise linear regression with two breakpoints under the assumption that the relationship between MagArea and curvature as plotted in Figure A.36 rather depicts a piecewise linear relationship clearly separating spaces of positive and negative

curvature. We further report six alternative models from Turkes et al. [2022], which are using features from persistent homology (PH) summarising persistence diagrams (PDs). See Bubenik et al. [2020] for a more detailed explanation on PH and its relationship to curvature. Specifically, in Table 6.1 we reproduce the following models from Figure 4. and Table 3. of Turkes et al. [2022]:

- SVR (all PH features) referred to as 0-dim PH simple by Turkes et al. [2022], which uses the lifespans of the persistence diagram computed on the samples;
- SVR (selected PH features) denoted 0-dim PH simple 10 by Turkes et al. [2022], which uses the 10 longest lifespans; and
- SVR (PH vectorisation) corresponding to 0-dim PH by Turkes et al. [2022], which selects the best PD vectorisation amongst a number of options, namely persistence images (PI) or persistence landscapes (PL).

All the PH-based methods use support vector regression (SVR) with a RBF kernel. Hyperparameter tuning for these models is conducted as reported by Turkes et al. [2022] using grid search with a choice of C parameters in $\{0.001, 1, 100\}$. We further reproduce 1 method based on pairwise distance matrices:

- SVR (distance matrices) denoted as ML by Turkes et al. [2022].

Finally, we restate the performance scores of these two methods directly from Turkes et al. [2022]:

- MLP (shallow) denoted as NN shallow by Turkes et al. [2022]; and
- MLP (deep) denoted as NN deep by Turkes et al. [2022].

We also note that the other models achieve different performance scores on our dataset than reported by Turkes et al. [2022] due to a slight difference in dataset and cross-validation splits. We use a smaller subset of samples than Turkes et al. [2022] each having a unique curvature value as described above, and ensure that all models are evaluated on the same splits of data across 5-fold CV for fair comparison. Finally, we summarise the MSE achieved by each model in Table 6.1. Illustrating this, Figure 6.2 further shows examples of both magnitude functions for negative and positive curvature as well as the clear piecewise-linear trend between MagArea and curvature.

We first assess to what extent the magnitude function can detect whether a unit disk has positive or negative curvature. Our main observation from plotting the functions for both groups (Figure A.36 in the appendix) is that there is a clear separation between spaces of negative and positive curvature. We further test if we can predict curvature as a regression task. To this end, we try both piecewise linear

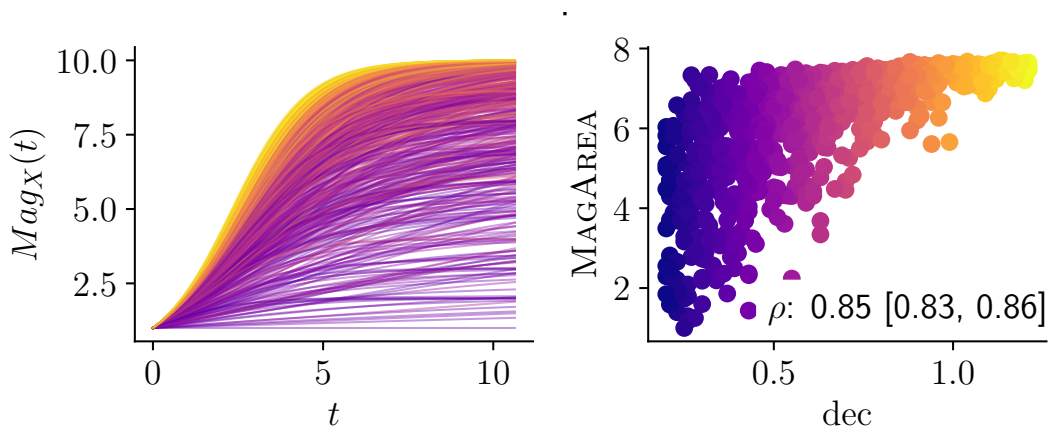


Figure 6.3: **MagArea correlates well with** dec **indicating the true diversity.** Here, we use `mpnet` embeddings for the `resp` dataset. ρ denotes the rank correlation between MagArea and dec (95% bootstrap interval, 1000 resamples).

and quantile regression, using the area under the magnitude curve, MagArea, as a single feature. With 5-fold cross validation, we achieve an MSE of 0.05 ± 0.03 with the piecewise linear model and 0.10 ± 0.05 using quantile regression. Both scores substantially improve on previous methods [Turkes et al., 2022] that made use of highly-sophisticated topology-based features and more heavily-parametrised deep learning models (see Table 6.1). These results underscore the expressivity and power of magnitude-based metrics, which enable us to solve the *same* task with a highly-simplified model. Moreover, this also demonstrates how magnitude describes the data manifold across multiple resolutions, motivating the use of magnitude functions as flexible, geometry-aware descriptors of diversity.

6.4.2 Magnitude Measures the Intrinsic Diversity of Text Embeddings

Next, we demonstrate the utility of using magnitude for intrinsic diversity evaluation and study its correspondence to known ground-truth diversity of text data. We analyse data from Tevet and Berant [2021], consisting of 1K sets of 10 sentences each, generated for unique input prompts for 3 different sentence generation tasks, namely story completion (`story`), dialogue response generation (`resp`), and 3-word prompt completion (`prompt`). Per task, 10 response sets have been generated using the same decoding parameter, the softmax-temperature dec, which controls the diversity and randomness of the

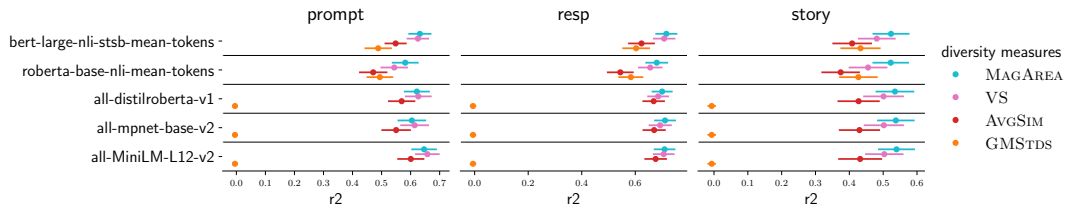


Figure 6.4: **MagArea outperforms alternative diversity measures** at predicting the ground truth-diversity of generated sentences, controlled by the softmax-temperature across 3 tasks and 5 embedding models. Baseline measures, AvgSim and GMStds, perform worse in terms of the R^2 scores. Points show the mean of the R^2 scores, while lines represent the standard deviations across 5-fold cross-validation (repeated 10 times).

generated text. As dec decreases, models are skewed towards avoiding low-probability tokens. This leads to potentially higher quality and fidelity but lower diversity and creativity in generated text. We embed each set of responses using 5 pre-trained sentence transformer models [Reimers and Gurevych, 2019], i.e. (1) `bert-large-nli-stsb-mean-tokens`, (2) `roberta-base-nli-mean-tokens`, (3) `all-mpnet-base-v2`, (4) `all-distilroberta-v1`, and (5) `all-MiniLM-L12-v2`. For each dataset and model, we compute the area under the magnitude function MagArea, evaluated until the median convergence scale across all embeddings as detailed in Section 6.3.3 using cosine distances. We compare this to the Vendi Score (VS), AvgSim, and GMStds, calculated using cosine similarities. Moreover, we analyse the performance of each diversity metric at predicting the ground-truth diversity scores, dec, using 5-fold cross-validation repeated 20 times, trained via isotonic regression models;[3] and report their performance in terms of the coefficient of determination, R^2. Figure 6.3 depicts the positive rank correlation between magnitude and the softmax-temperature for one example setting, while Figure 6.4 shows results concerning the predictive performance of different diversity measures.

We observe that MagArea consistently outperforms alternative diversity measures computed from the same representations. MagArea achieves a median rank of 1 across experiments in terms of R^2 scores, followed by VS, AvgSim and GMStds. Indeed, MagArea is most frequently the best-performing diversity measure for 77% of resamples when predicting decoding parameters, ranking

[3]We use these models to capture the non-linear monotonic relationship between dec and diversity.

second in the remaining cases. Meanwhile, VS most often achieves second place. This demonstrates the strength of MagArea as a theoretically-motivated and entropy-based measure of intrinsic diversity. By contrast, the baseline measure GMStds fails for any embedding that has at least one constant dimension, even reaching negative R^2 values for three of the five embedding models. This is followed by AvgSim, which, while being less fallible than GMStds, simply measures average similarity and even ranks last across 27% of resamples. A further comparison of performance scores shows that MagArea outperforms AvgSim by 0.12 higher mean R^2 scores on `story` and 0.07 on `resp` or `prompt` across embedding models. We find no dataset for which either AvgSim or GMStds can be considered preferable predictors of the ground-truth diversity of text. Our results thus show the benefits of replacing simple summaries as the current standard for automated diversity evaluation with more sophisticated diversity measures like MagArea.

6.4.3 Magnitude Distinguishes and Characterises Embedding Models

Motivated by the capability of magnitude functions to encode representations, we now check whether the embedding spaces of different large language models can be distinguished via their intrinsic structure. To this end, we analyse 16384 documents of four different HuggingFace datasets, as embedded by Wayland et al. [2024] using six different models

We then either use PCA and normalisation to reduce each embedding space to 384 dimensions (to obtain a comparable dimensionality) or use the original embeddings without preprocessing. Further we subsample 300 documents at random from each space, repeating this procedure 200 times. Finally we use a 5-NN classifier to predict the embedding model based on the values of each diversity measure. Table 6.2 reports the results of 5-fold cross-validation with 20 repetitions for both prepossessing choices. We either use Euclidean distances between single number summaries or, in the case of magnitude, use MagDiff directly as the input distances for k-NN classification. We first observe that MagDiff best predicts the embedding model (with accuracies typically above 90%). Surprisingly, the results remain consistent for both pre-processing choices. This indicates that there are inherent differences in the structure and diversity of

Table 6.2: **Magnitude characterises text embedding models.** We show the accuracy (↑) of different diversity scores for distinguishing between six embedding models, using a 5-NN classifier.

	No pre-processing				PCA pre-processing			
DatasetMethod	MagDiff	AvgSim	VS	GMStds	MagDiff	AvgSim	VS	GMStds
cnn	$\mathbf{0.94 \pm 0.02}$	0.87 ± 0.01	0.63 ± 0.01	0.66 ± 0.02	$\mathbf{0.90 \pm 0.02}$	0.88 ± 0.02	0.67 ± 0.03	0.66 ± 0.03
patents	$\mathbf{0.99 \pm 0.01}$	0.92 ± 0.01	0.63 ± 0.02	0.66 ± 0.02	$\mathbf{0.96 \pm 0.01}$	0.91 ± 0.02	0.64 ± 0.03	0.66 ± 0.03
arXiv	$\mathbf{0.99 \pm 0.01}$	0.89 ± 0.01	0.78 ± 0.01	0.66 ± 0.02	$\mathbf{0.99 \pm 0.01}$	0.88 ± 0.02	0.78 ± 0.02	0.66 ± 0.03
bbc	$\mathbf{0.98 \pm 0.01}$	0.74 ± 0.01	0.84 ± 0.02	0.66 ± 0.02	$\mathbf{0.95 \pm 0.01}$	0.73 ± 0.03	0.84 ± 0.02	0.66 ± 0.03

embedding spaces, which are preserved throughout dimensionality reduction and captured by magnitude. By using the difference between magnitude functions as a holistic summary, we once again surpass other summary statistics (which we observe to fail in distinguishing the smaller embedding models). Our results thus demonstrate that using MagDiff for comparing latent spaces across multiple scales is considerably more expressive than using single-number summaries of diversity.

6.4.4 Magnitude Evaluates Image Embeddings

Mode dropping is a common issue in generative modelling, referring to the inability of a model to capture all parts of an input distribution (for instance, a model trained to generate images of animals suffers from mode dropping if it can only generate images of dogs). To simulate this, we randomly sample 100 images from each of the 10 classes in CIFAR10 and embed them using a pre-trained Inception V3 model [Szegedy et al., 2016]. Subsequently, we re-sample increasingly more observations from *one* preferred image class. We either drop modes sequentially, or we move the same number of observations simultaneously from all other classes. Thus, diversity decreases gradually with the same 'speed' across both procedures, but fidelity should not change. We treat each class as the preferred image class twice, leading to 20 re-samples per mode dropping scenario [Naeem et al., 2020]. Our analysis compares the changes in recall and coverage, setting the number of nearest neighbours to $k = 10$. Further, we calculate the relative change in $\mathrm{Mag}(0.5t_{\mathrm{ref}})$, i.e. magnitude computed at half the convergence scale of the reference using Euclidean distances. Similarly, MagDiff is the difference between the magnitude functions relative to the area under the reference magnitude function.

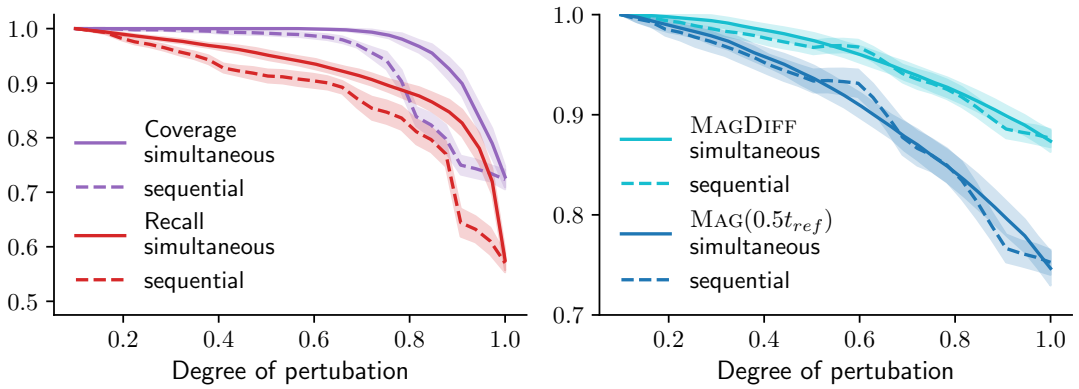


Figure 6.5: **Magnitude correctly detects that diversity decreases in the same manner across simultaneous and sequential mode dropping** outperforming recall and coverage. Lines show the mean values of each metric across 20 resamples, shaded areas the standard deviations.

Figure 6.5 shows the changes in diversity as modes are being dropped. Ideally, every diversity measure should show the *same* decrease in diversity, irrespective of resampling strategy. However, we observe that both recall and coverage wrongly assess that diversity decreases faster during sequential resampling. Even worse, coverage only detects simultaneous mode dropping after around 70% of all points have shifted to one mode. This undesirable behaviour of both metrics is caused by their reliance on a fixed neighbourhood size for approximating the underlying manifold, thus overestimating the extent to which the perturbed samples reflect the diversity of the reference distribution. In comparison, MagDiff as well as magnitude evaluated at a single scale both successfully measure the gradual decrease in diversity across both mode dropping scenarios.

6.4.5 Magnitude Evaluates Graph Generative Models

Diversity evaluation in graph learning is fraught with difficulties, in particular when aiming to detect common problems like *mode collapse* or *mode dropping* [Thompson et al., 2022, O'Bray et al., 2022]. In the following, we will study graph generative models (GGMs), which take a set of input graphs and generate new samples that should follow the *same* distribution. The question that we aim to answer here is whether our proposed magnitude-based metric is more expressive in capturing the diversity of the generated graphs than classical metrics like *maximum mean discrepancy* (MMD) and measures inspired from evaluating image gener-

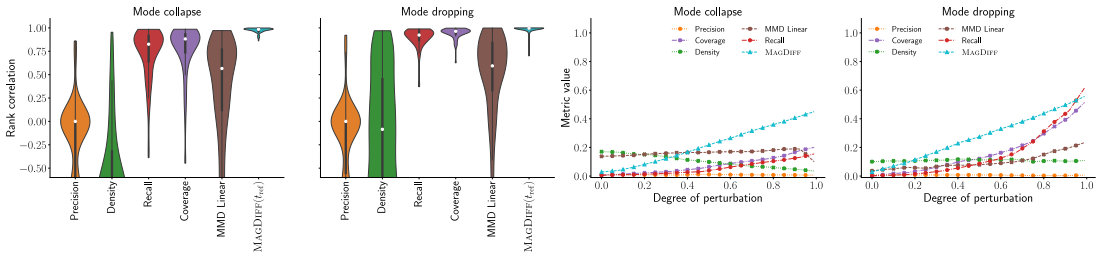


Figure 6.6: **MagDiff outperforms existing graph diversity metrics at detecting mode collapse and mode dropping.** We report the Spearman correlation between each metric and the degree of perturbation p for the Lobster dataset (the same pattern holds for Proteins, Community, Ego, Grid, see A.4.5.2). Violin and box plots show the distributions across different hyperparameter choices. Measures that capture the decrease in diversity accurately should increase as a function of p. Rank correlation of 1 corresponds to an ideal metric. Our metric best captures the changes in diversity for both mode dropping and collapse.

ative models (precision, recall, coverage, density). To this end, we analyse 3 synthetic (Lobster, Grid, and Community) and 2 real-world (Proteins and Ego) graph datasets, and compute commonly-used evaluation metrics [Thompson et al., 2022, O'Bray et al., 2022] as detailed in A.4.5.2. To test the diversity of generated samples, we replicate the experimental setup of Thompson et al. [2022] and add our own measure, MagDiff computed using L^2 distances from Graph Isomorphism Network [Xu et al., 2019a, GIN] embeddings with varying hyperparameters. For the *mode collapse* experiments, we substitute each embedded graph with its cluster centre. Thus, the degree of perturbation p equals the proportion of clusters collapsed in this manner. The larger the value of p, the more clusters have been perturbed decreasing the diversity. For the *mode dropping* experiments, we remove clusters, and keep the size of the generated dataset the same as the reference by randomly resampling from the remaining classes.

Figure 6.6 shows the results of both *mode collapse* and *mode dropping* for the Lobster dataset. We observe similar trends across all datasets, but have chosen this dataset as a running example. Ideal measures should exhibit high rank correlation to the degree of perturbation, indicating that they are capable of capturing the decrease in diversity properly, i.e. as a function of p. We note that in contrast to our magnitude-based metric, *recall* and *coverage* exhibit worse results, as evidenced by their lower mean correlation coefficient. Despite being

specifically designed to measure the diversity of a dataset [Thompson et al., 2022], they only catch up to our magnitude metric when the degree of perturbation p is around 0.9 (see Figure 6.6, right-hand plots). Magnitude dominates in the majority of the values of p best showing the steady decrease in diversity, while recall and coverage become more sensitive for exceedingly large values of p, i.e. in unrealistic situations where most of the modes have been dropped. Moreover, their performance is highly contingent on k, the parameter used to construct a k-NN graph. Magnitude functions meanwhile give more holistic summaries of both local and global patterns in diversity. Please refer to Figure A.37 for the aggregated results over all datasets, which exhibit a similar pattern (in that our metric outperforms both *recall* and *coverage*).

6.5 Conclusion

We have proposed novel diversity measures for evaluating latent representations. Our measures are based on *metric space magnitude*, a multi-scale invariant summarising geometrical characteristics of the input data. We have demonstrated axiomatically and empirically that our magnitude-based measures are superior to current baseline measures of intrinsic diversity. In a reference-free scenario, we observe that magnitude outperforms alternative measures when predicting the ground truth diversity for text embeddings. Given a reference dataset, we find that magnitude captures mode collapse and mode dropping better than existing metrics for evaluating generative models for both image and graph modalities. Furthermore, we have shown that magnitude can measure the intrinsic curvature of input data, outperforming previous methods. Magnitude thus gives a provably stable, unsupervised diversity metric that can be computed efficiently and allows users to flexibly choose a notion of dissimilarity. For future work, we believe that magnitude exhibits a strong potential for applications to unaligned spaces with varying notions of distances. Moreover, we believe that integrating magnitude into deep learning models would be beneficial for obtaining novel diversity- and geometry-based regularisation strategies.

Chapter 7

Biomedical applications of magnitude and TDA

In this chapter, we start by presenting two novel applications of magnitude: first, to the surface of the human tongue in 7.1 and second, to the brain artery tree dataset in 7.2.

7.1 Machine learning, topological data analysis and magnitude identify unique features of human papillae in 3D scans

7.1.1 Introduction

The tongue is a highly sophisticated, heterogeneous anatomical structure and its operation is fundamental to speech, friction regulation and oral processing of food. The surface of the tongue is covered with tiny projections known as *papillae* which enable perception of taste, texture and oral mechanics. Of these numerous anatomical projections, *fungiform papillae* are considered as phenotypic markers of chemosensation of taste as they house the taste buds [Miller Jr and Reedy Jr, 1990], whereas *filiform papillae* that are devoid of taste buds are considered to be regulators of mechanoreception [Lauga et al., 2016] for textural perception. Women are believed to have more fungiform papillae and are classed more frequently as supertasters [Bartoshuk et al., 1994]. On the other hand, increased number of papillae have been found to be associated with enhanced fatty perception

[Jilani et al., 2017, Zhou et al., 2021]. In addition to taste perception, papillae on the tongue are responsible for mechano-sensing. Mechano-sensing refers to our ability to sense the texture, friction, lubrication and touch on the tongue surface, and is carried out mainly by numerous filiform papillae that act as fine strain-amplified sensors on the tongue surface. These sensory functions are critical for manipulation and transport of food and liquids in the mouth[Lauga et al., 2016, Sarkar et al., 2019]. Such textural properties also influence our psychological reaction to food. For example, feelings such as satiety and therefore hunger are influenced by perception of friction and lubrication [Stribiţcaia et al., 2020, Krop et al., 2019]. It has recently been shown that our preference for certain food such as chocolates is driven by surface lubrication that can be measured by artificial tongue-like surfaces [Soltanahmadi et al., 2023]. Besides food preferences, there is burgeoning interest in understanding the complex morphology of the tongue due to its involvement in various age-related oral conditions [Tamura et al., 2012, Xu et al., 2019b, Hu et al., 2021], mucosal degeneration and systemic diseases [Murphy et al., 2016, Porter et al., 2017, Huang et al., 2021, Jin, 2020]. Certain medical conditions [Maeda, 2006] and inter-individual differences are known to be associated specifically with the morphology of the papillae and the tongue. Understanding the finer details in morphology, differences in papillae structures can thus lead to fabricating novel bio-inspired artificial surfaces in biomedical engineering, food engineering and therapeutics [Andablo-Reyes et al., 2020, Arzt et al., 2021].

The intricate geometry of the tongue at a microscopic scale can be appreciated in 3D scans (see Figure 7.1). These images are obtained *via* surface reconstruction of 3D optical scans of a silicone-polymer mask of a human tongue. Fungiform papillae (Figure 7.1(b)) are larger, sparsely distributed over the surface, and have a simple hemisphere-like shape. The average diameter of a fungiform papilla is about 878μm [Andablo-Reyes et al., 2020], and they are clearly visible in larger images (Figure 7.1(a)). The filiform papillae show a more intricate crown shape (Figure 7.1(c)). They are smaller (about 355μm in diameter) and substantially more numerous. A square centimeter of human tongue surface is estimated to contain between 100 and 200 filiform papillae [Andablo-Reyes et al., 2020].

Although there has been significant research on the importance of papillae density, our understanding of the papillae shapes and surface properties of the tongue suffers from the difficulty of extracting and analysing geometry of papillae

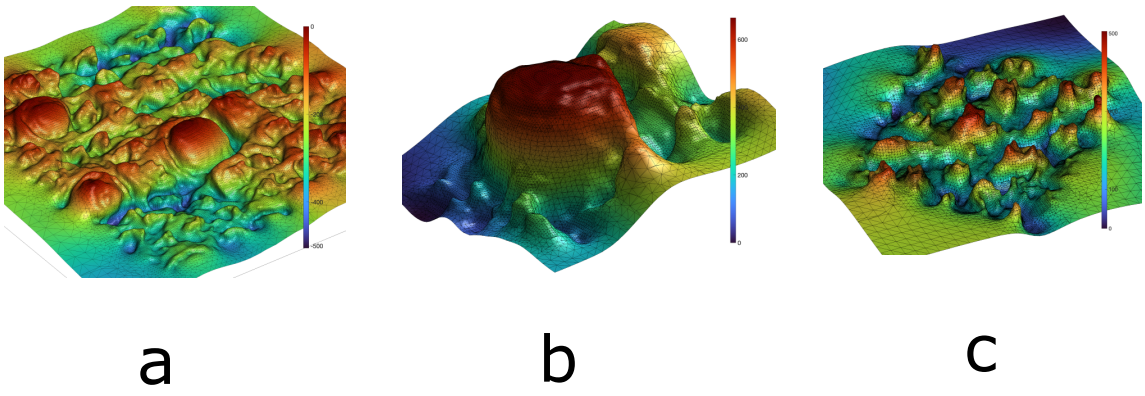


Figure 7.1: **3D representation of a small portion of the dorsal part of the human tongue**. Plot (a) shows the 3D mesh of tongue surface obtained from masks taken on a real human tongue. The color bar shows the z-coordinate of the points on the surface representing the height. In plots (b) and (c) we see regions of the tongue with (b) Single Fungiform papilla and (c) Multiple filiform papillae. We note the distinctive shapes of papillae in plots (b) and (c), i.e, the dome-shaped Fungiform papilla in (b) and the crown-like shaped Filiform papillae in (c). Impressions of human tongue was collected at University of Leeds (Ethics DREC ref: 120318/AS/245, University of Leeds) [Andablo-Reyes et al., 2020] from healthy adults (n = 15 subjects, 9 females, age 18–55 years)

at microscopic scales. Previous studies have thus focused on manually localising papillae from 2D images [Nuessle et al., 2015], primarily focusing on fungiform papillae [Cattaneo et al., 2020]. Other works on biological surface data have used conformal geometry and computational topology at larger scales. Examples of such techniques include shape registration [Hong et al., 2006], segmentation and topological data analysis [Amézquita et al., 2020, Nicolau et al., 2011, Oyama et al., 2019, Krishnapriyan et al., 2021, Saadat-Yazdi et al., 2021, Khalil et al., 2022]. Machine learning has recently emerged as a powerful technique for diagnosis where large volumes of medical data or images are available [Cai et al., 2020]. These approaches have largely focused on computing global functions such as a medical diagnosis from an image. However, to date there is no machine learning model that has classified microscopic tongue papillae based on 3D tongue scans.

Herein, we present the first study of the 3D shapes of filiform and fungiform papillae in humans, with an emphasis on the variations in the microscopic geometry seen in Figure 7.1. We develop a machine learning based framework applied to custom designed topological and geometric properties to understand one fundamental issue: *What separates one type of papillae from another?* We also ask the questions whether papillae are unique across and within individuals based on finer geometric details. Instead of applying machine learning as a black box application, we use statistics and explainable machine learning [Molnar, 2020, Du et al., 2019] to differentiate one type of papillae from another and identify the most distinctive features.

We follow the process of Topological Data analysis, where implicit shapes in data are extracted as topological *features* that form the basis of machine learning models. However, in addition to topological features, we also make use of geometric features computed from magnitude, which captures the effective number of points in a space, and it has been shown to detect curvature, and discrete curvatures to understand the uniqueness of tongue papillae. These features together are seen to have a high accuracy of 85% in correctly identifying the papilla type (filiform or fungiform) in a small segment of a surface. As a result, we can now map the papillae arrangement for the first time – including filiform papillae that are critical for developing biorelevant tribological surface and unravelling mechano-sensing – as seen in Figure 7.5.

Unprecedented analysis from our model reveals differences in papillae shapes across gender, age and individuals. We find that given a papilla, the age group and gender of the participant can be predicted to moderate accuracy, and even the exact individual from among 15 participants can be identified with approximately 48% accuracy, showing the first evidence for papillae to act as a unique identifier. This study demonstrating the uniqueness of papillae geometry at microscopic length scales using discrete differential geometry, magnitude and computational topology stands to benefit future development of 3D tongue models for enabling rational food design diagnosis of oral medical conditions.

7.1.2 Methods

Data collection

Collection of Human Tongue Silicone Impressions.

The data in this study has been obtained from 3D optical scans of masks of real human tongues from 15 healthy participants performed using an Alicona InfiniteFocus (IF), details of the data collection has been described in a previous publication [Andablo-Reyes et al., 2020]. Negative impressions of the upper surface of the tongue were collected from ($n = 15$ subjects, the mean age in years is 29.1, SD=3.7), 6 male and 9 female. More detailed information can be found in Table A.9.

Experimental protocol.

The study adhered to all relevant guidelines and regulations. Signed informed consent was obtained from all participants before undertaking the experimental protocol. The ethics declaration is included at the end of this section.

Dataset generation for papillae

Each participant's point cloud was split into two smaller parts of approximate size 13mm by 9mm in order to reduce the size of the point cloud. On each part, The Screened Poisson surface reconstruction[Kazhdan and Hoppe, 2013] in Meshlab[Cignoni et al., 2008] is applied. Then, a number of circular segments of radius $r+\delta$, where r is set to match $max(r_{fungiform}, r_{filiform})$µm and $\delta = 100$, were extracted according to our algorithm for extracting candidates for papillae locations described below. Based on previous work [Andablo-Reyes et al., 2020] and our experiments in detecting papillae, we find that $r_{fungiform} = 439, r_{filiform} = 177.5$ work well for automated detection. These segments have been manually labelled into one of three classes: fungiform papillae, filiform papillae or None (neither a fungiform nor a filiform). The final dataset consists of 414 fungiform, 1489 filiform and 190 None, resulting in 2092 tongue segments in total. The number of segments per participants can be found in Table A.8.

Finding Candidates for papillae locations

The pipeline for segment extraction works as follows. First, we pick a random point P on the surface. Then, we select a radius $r + \delta$ of points around P, where we set r to match $max(r_{fungiform}, r_{filiform})$µm. Note that the distance from P is computed as the $3D$ Euclidean distance in the ambient space. If the set of points contain disconnected components, then components not containing P are discareded from the computation. We set $\delta = 100$ to fully cover any papilla in the region. After that, we fit a plane based on the RANSAC algorithm [Fischler and Bolles, 1981] and identify the point M furthest away from the plane, which will be a *local maxima*. We identify it to be the centre of the segment. Finally, we cut a region of radius r around m as a candidate segment. This process is applied repeatedly to identify multiple papilla segments. In future iterations, any maximum within a previously processed segment is ignored. Number of iterations and samples in our experiments were limited by the need for manual labelling. In applying our model for mapping papillae (e.g. as in Figure 7.5) the process can be continued until no new papillae is found.

Baseline, geometric and topological features

Three sets of features are extracted from each of the selected segments – baseline, curvature and topological.

Baseline features

We use geometric measurements for baseline feature identification, which comprises of two quantitative shape characteristics of the papillae: height and radius. From the data presented in Table 1 in [Andablo-Reyes et al., 2020], based on Tukey's test for statistical significance of the means and standard deviation, the diameter (and the radius, respectively) and height are different between fungiform and filiform. Therefore, they can serve as features for distinguishing between the three classes.

We note that defining the height and radius automatically is a challenging task due to the irregular nature of these structures and to our knowledge no unambiguous definitions exist in the literature to date to identify these features accurately. Human participants do this manually by observing the continuity of the papillae from the base to the tip.

We compute height and radius as follows. The point m, identified as the local maximum for the segment, as the centre of the structure. Then we define the radius r as the radius value of the sphere, centered at M, which contains 90% of the points in the segment. We compute this iteratively, by first guessing the value of the radius i as a small value (100μm), and count the number of points in the neighbourhood of radius i (we use KDTree with FLANN [Muja and Lowe, 2009] for nearest neighbor search). We then increase i by 10, until the number of points contained in the neighbourhood exceeds 90% of all points. The value of i at the stopping condition is our candidate for radius value, r. The computation of the height, h, is dependent on the value of r. It works as follows: we first cut a region around the centre of radius r, we then fit a plane using the RANSAC algorithm [Fischler and Bolles, 1981] and find the maximum distance from the plane to the local maximum point M. This value is our height value, h. All the computations have been performed using the Python libraries `open3d` and `numpy`. The algorithm is mimicking the manual procedure which a tongue expert would use to compute these values. An illustration of the procedure can be found in Figure A.41.

Geometric features

Curvature features For each $x \in H$, where H is the surface generated by the Poisson surface reconstruction process, we compute the discrete curvature as defined by Meyer et al. [Meyer et al., 2003]. The definition in the discrete case on the triangular mesh is via the vertex's angular deficit $k_H(v_i) = 2\pi - \sum_{j \in N(i)} \theta_{ij}$, where $N(i)$ are the triangles incident on vertex i and θ_{ij} is the angle at vertex i in triangle j. The way that Gaussian and mean curvatures are computed uses averaging Voronoi cells and the mixed Finite-Element/Finite-Volume method [Meyer et al., 2003]. We use the existing implementation from the Python version of Meshlab, called `pymeshlab` [Muntoni and Cignoni, 2021].

We use the maximum and minimum of the Gaussian and mean curvature as features, the ratio of positively curved points to the number of all points in the mesh ($k_{positiveratio}$), and we introduced a new feature called curvature ratio (k_{ratio}). Let x be the number of points of positive curvature, and y be the number of points of negative curvature. Therefore, we define the curvature ratio k_{ratio} to be $k_{ratio} = \frac{y}{x}$ if $y \leq x$ and $k_{ratio} = \frac{x}{y}$ if $x \leq y$. The signs of the mean and

the Gaussian curvature provide plenty of information about the local behavior of the surface [Colombo et al., 2006]. We computed the discrete Gaussian and mean curvature for all meshes and calculated the number of vertices of positive and negative curvature (after the Poisson surface reconstruction filter). The ratio of positively curved points to the number of all points in the mesh is defined as $k_{positiveratio} = \frac{x}{x+y}$. The full list and intuitive interpretations are provided in the Supplementary material, Table A.7.

Magnitude-based feature Let (X, d) be a finite metric space with distance metric d. Then, the similarity matrix of X is defined as $\zeta_{ij} = e^{-d_{ij}}$ for $1 \leq i, j \leq n$, where n denotes the cardinality of X. Magnitude is then defined as the sum of the entries of the inverse of the similarity matrix ζ, $\mathrm{Mag}(X) = \sum_{ij}(\zeta^{-1})_{ij}$. We use magnitude at a single scale, in this case $t = 0.25$, as this appears to be the most distinctive value between the young and the old brain.

Topological features

We subsample the 3D point clouds to 1000 points each and compute the Vietoris-Rips complex, using the Euclidean distance as a filtration. Persistent homology [Edelsbrunner and Harer, 2022] of the 3D point cloud was computed using the `giotto-tda` library [Tauzin et al., 2021] and `ripser` [Bauer, 2021b]. We then generate 12 features which are one number summary of the diagram, providing different topological information. For more details on persistent homology, please refer to the Supplementary material.

Short bars are the number of intervals of length between 0 and 10. We compute them both in homology dimension 0 and 1. This features has been found to capture the local geometry of an object [Bubenik et al., 2020]

Persistent entropy [Chintakunta et al., 2015, Atienza et al., 2020] is the measure of the entropy of the points in a persistent diagram. Concretely, let $D = \{(b_i, d_i)\}_{i \in I}$ be a persistent diagram with non-infinite death times, i.e., $d_i < \infty$. Then, the persistence entropy of D is defined as $P_E(D) = \sum_{i \in I} p_i \log(p_i)$, where $p_i = \frac{(d_i - b_i)}{L_D}$ and $L_D = \sum_{i \in I}(d_i - b_i)$. We compute persistent entropy in dimension 0 and 1, and denote it by Persistent entropy (0) and Persistent entropy (1).

Persistence landscapes: Given a persistent diagram $D = \{(b_i, d_i)\}_{i \in I}$, its persistence landscape is the set $\{\lambda_k\}_{k \in \mathbb{N}}$ of functions $\lambda_k(t) : \mathbb{R} \to [0, \infty]$, where

$\lambda_k(t)$ is the k-th largest value of the set $\{g_{(b_i,d_i)}(x)\}_{i=1}^n$, where $g_{(b,d)} = 0$ if $x \notin (b,d)$; $g_{(b,d)} = x - b$ if $x \in (b, \frac{b+d}{2})$ and $g_{(b,d)} = -x + d$ if $x \in (\frac{b+d}{2}, d)$. The parameter k is called a layer. In this work we consider the case when $k = 1$.

Persistence image: diagrams are converted to sums of Dirac deltas. The convolution with Gaussian kernel is performed, where the computation is done over a grid with rectangular shape. The locations of the points are evenly sampled from the values of the filtration, turning it into a raster image, which is then flattened into a vector.

We note that including both persistence landscape and persistence image might appear redundant in general, as these are two different vectorizations of the same diagrams. We did not find correlation between the features derived from them and this is why we kept them. However, this could be due to the procedure used to adjust the parameters of the persistence image (the bandwidth might not be tuned well), and as a result the corresponding Gaussian functions are too spread or narrow. We note that this question warrants further investigation which is outside the scope of this chapter.

Amplitude can be defined as the distance from the persistent diagram to the empty diagram, which contains only the diagonal points. Here we use 2 kernels (persistence landscapes [Bubenik and Dłotko, 2017] and persistence image [Adams et al., 2017]) and the amplitude of the kernel is computed using the $L2$ norm, and 2 metrics (Wasserstein and Bottleneck). For the computation, we use the default parameters in `giotto-tda`.

We here denote Persistence image amplitude by Amplitude (Image, 0) Amplitude (Image, 1) for the computation of the amplitude with the persistent image kernel (which is the is the $L2$ norm of that vector) in homology dimension 0 and 1, respectively. Similarly, Amplitude (Landscape, 0) and Amplitude (Landscape, 1) is the Persistence Landscape amplitude in homology dimension 0 and 1.

The Wasserstein amplitude of order p is the Lp norm of the vector of point distances to the diagonal, which is $A_w = \frac{\sqrt{2}}{2}(\sum_{i\in I}(d_i - b_i)^p)^{\frac{1}{p}}$. Here we use $p = 2$. Similarly, the *Bottleneck amplitude*, A_B, is defined by letting p to ∞ in the definition of the Wasserstein amplitude. In other words, it is a fraction of the longest bar $A_B = \frac{\sqrt{2}}{2}\sup_{i\in I}(d_i - b_i)$. We denote them by Amplitude (Wasserstein, 0), Amplitude (Wasserstein, 1) and Amplitude (Bottleneck, 0), Amplitude (Bottleneck, 1) respectively, corresponding to the different homology dimensions.

Machine learning and statistics

Classification models. The experiments use classes of simple models – Support vector machines (SVMs) and Logistic regression models. The implementations from `scikit-learn` [Pedregosa et al., 2011b] were used without modification and with the default hyperparameters. The SVMs were used with a radial basis kernel (RBF). We use 20% of the data for testing and the other 80% for training using a random split. The procedure is repeated 50 times.

Performance Metrics for machine learning. Accuracy represents the proportion of correct predictions made by the model out of the total number of predictions. To adjust for the varying number of samples across classes, we compute the balanced accuracy. It calculates the average of the correct classification proportions for both positive and negative observations.

Feature Importance. The plots are based on classification by the best balanced accuracy split of the data, and 30 permutations of the features for that split. The black line represents the standard deviation of the feature importance over the 30 runs.

7.1.3 Results

Our analytic framework processes the data, computes the features, and then applies machine learning driven analysis. We briefly explain the data processing and feature extraction. Then we proceed with a machine learning driven analysis of the feature set, prediction of gender, age and papillae type that reveals insights about papillae.

The data is obtained as 3D digital scans. The process starts with taking masks of the dorsal area of tongue of participants on silicone polymers. These masks are scanned using a 3D scanner, which yields a set of 3D points. These points are then passed through a surface reconstruction algorithm [Kazhdan and Hoppe, 2013] implemented in Meshlab [Cignoni et al., 2008], which yields a mesh and a corresponding surface (see Figure 7.1). This process was developed by Andablo et al [Andablo-Reyes et al., 2020].

From this mesh data, we extract *segments* that are candidates for papillae. The extraction process is as follows. Around a point P on the surface, select the set B of points within a radius $r + \delta$, where $r = \max(r_{\text{fungiform}}, r_{\text{filiform}})$ μm and $\delta = 100\mu$m, which we find to work well in practice. A plane fit to B based on the

RANSAC algorithm [Fischler and Bolles, 1981] represents our best approximation of the plane of the segment base. The *local maximum* m in the segment is defined as the point furthest away from the plane. This point is assumed to be the peak of a papilla, if present. Finally, we cut a region of radius r around m representing a candidate mesh for a papilla. Figure 7.2(a) and Figure 7.2(b) show such extracted segments for a fungiform and filiform papilla, while Figure 7.2(c) shows general surface area without any papilla. These three kinds of elements are the basis of our study.

A total of 2092 segments extracted from scans of 15 participants were labeled manually as Fungiform, Filiform or None. All accuracies reported in this chapter are accuracy on the test set of unseen data. The analysis and machine learning are carried out on a large set of features (Table A.7). In past work[Andablo-Reyes et al., 2020], baseline features height and radii have been found to be distinctive between papillae types. Our more comprehensive segment dataset and computational models improve upon these baseline features to attain high accuracy automated detection of papillae type and other tasks.

Features and feature visualisation

Features can be considered at different scales. At the global scale, a topological invariant of the entire papilla may be a distinctive feature; or an isometry invariant like magnitude can accurately capture the effective number of points in a papilla. At the local scale of the neighborhood of a point on the surface, geometric properties – in particular, curvature of points in the neighborhood – best characterise the local shape of the surface. Local properties can be aggregated over the entire papilla to obtain a global feature. We describe below the significance of topological and geometric quantities in this context.

Topological features. In this work, topological properties are computed via *persistent homology*. In this approach, each vertex (for us, a point on the reconstructed surface) is treated as the center of a growing ball, and the union of these balls is observed for changing topology. One way to interpret computational persistent homology is that it monitors topological features of different dimensions as they are born and die with the growth of the balls. Connected components in 0-dimension, loops in 1-dimension, and higher dimensional spheres in higher dimeensions. For a comprehensive introduction see the text by Edels-

brunner and Harer [Edelsbrunner and Harer, 2022]. Figures 7.2(d – i) show the persistent topological components for the three types of segments, where the scale is measured in μm. Figures 7.2(d – f) show the persistent diagram view, where each component manifests as a point indexed by its birth and death time. The difference in distribution of the points across plots suggests that there are variations in topological features for different segments. Figures 7.2(g – i) show an alternative view of the same data, called the barcode view – where each bar shows the life duration of a topological component. From these sets of bars we can derive statistical features based on the distribution of bar lengths and more sophisticated methods. The feature which we have used in this work are based on persistent entropy, persistent images, persistence landscapes and amplitudes (please refer to the Methods section for detailed definition of each of the features and Table A.7).

The distribution of bars at different lengths for H_0 (connected components) are shown in Figure 7.2(j,k) as the kernel density estimates. Fungiform bar lengths in Plot 7.2(j) have higher density for shorter bars of length between 0 and 10 as compared to Filiform and None (around 0.01), and then again in the mid range between 17 and 25, where all densities achieve their maximum. There are considerably fewer longer bars for Fungiform as compared to Filiform and None, which dominate the longer bar end of the spectrum. In plot (k) with densities H_1, we note that the density of short bars (lengths between 0 and 10) are higher for Fungifrom (0.07), followed by Filiform (0.065) and None (0.06). Thus there seems to be one predominant region of major difference, while H_0 shows greater variation across types.

Geometric features. As geometric features we consider both magnitude and curvature. *Curvature* is locally defined at each point and is a complete descriptor of a surface. Positive curvature occurs where the surface matches a region of a sphere, for example at the top of a fungiform papilla. Sharp peaks are characterised by high positive curvature, while gentle tops, such as at the top of the fungiform papillae, have lower positive curvature. Negative curvatures are observed in saddle shaped neighborhoods, for example, around the base of papillae.

In digital discrete data, where manifolds are piecewise linear (triangulated) meshes, as in our case, curvature is computed at each vertex of the mesh as the angle deficit of the manifold (see Methods section for details). For our analysis, we compute curvatures on a sample of points in the segment. The geometric features

of a segment include quantities such as the maximum and minimum of Gaussian curvatures, percentage of points with positive and negative Gaussian curvature, and other aggregated quantities (See Table A.7).

The distribution of curvatures of the segments in Figure 7.2(a-c) are shown in Figure 7.2(l). For all types of papillae, most points are seen to be concentrated around small values of curvature close to zero. In particular, fungiform papillae have more points of near zero curvature, as can be expected from fungiforms having mostly flat or gently curving surfaces. In contrast, filliform and even generic surface areas are seen to have greater fraction of sharper curvature points.

For *magnitude*, we plot in Figure 7.2(m) the magnitude functions of fungiform, filiform and none papillae. We compute magnitude in the interval $[0, 1.5]$, and notice that the magnitude function of fungiform papilla has a distinctive pattern at the lower values of the scale parameter t.

Feature analysis and feature importance

Various features may have different levels of importance in the distinction between papillae. The importance of a feature is a fundamental question in the field of explainable machine learning, and is usually determined by its contribution to a classification model. It is a somewhat complex measure that is difficult to derive by looking at the feature in isolation. For our purposes, we use the technique called permutation feature importance [Breiman, 2001], and compute the contribution of these features to a class of standard classifiers called Kernel SVMs. The permutation feature importance method evaluates a feature f by nullifying f of the test data and observing the drop classification accuracy of the model. A large drop in accuracy implies f is an important attribute for the classifier model. The effect of nullifying f is achieved by permuting the values of f among the test data points.

Figure 7.3, shows three most important features in determining each of the four labels of interest to us: the papillae type, the gender, the age and the participant id. The main observation here is that certain topological features are seen to be consistently important in these tasks (Figures 7.3(a-d)). Topological features overall are also found to contribute more to prediction accuracy than other features (Figure 7.3(e)).

Type prediction features. The KDE plots of the most important features for

the papillae type classification task are presented in Supplementary Figure A.40, and the box plots and the aggregated distributions are shown in Supplementary Figure A.39. The three distinctive features are seen to have very different distributions for the different types of segments, which explains their effectiveness in classification.

Gender prediction features. We have two topological and one curvature feature at the top three for gender prediction task, whose box plots and aggregated distributions can be found in Figure 7.4. Persistent entropy (0) (Figure 7.4(a)), Maximum Gaussian curvature (Figure 7.4(b)) and Short bars (1) (Figure 7.4(c)) are all important features for determining gender. Figure 7.4(a), shows that the female participants tend to have a higher median value of the max Gaussian curvature (which holds for both Fungiform and Filiform) as compared to male participants, which could be linked to female papillae being 'sharper', or 'pointier'.

Age prediction features. Topological features also dominate the age-prediction task, as seen in Figure 7.3. Persistent entropy (0), Amplitude(Image,0) and Maximum Gaussian are the most important features for the age classification task. The baseline features (Height, Radius) are not amongst the most essential for this task, suggesting that their characteristics do not differ much for the two age groups in this study. An interesting observation is that height is more important than radius. Similar to the gender-prediction task, the Maximum Gaussian curvature feature is one of the most important. The median for the younger age group is 0.269 ($n = 840$) and for the older is 0.166 ($n = 640$), implying some difference between the two groups, with the younger group having 'pointier' papillae. This holds both for Fungiform and Filiform.

Predicting gender, age and participant from papillae structure

Having understood the differences in papillae structure based on gender and age, we ask if one can easily predict gender, age and the participant given a papilla. Specifically, we ask if the papillae and the features identified above contain sufficient information to allow simple statistical methods to carry out accurate prediction.

Gender prediction

In this task we predict the biological gender of the participants. The classification performance is presented in Table 7.1. The models trained on topological features result in accuracy of 65%, outperforming the geometric features without magnitude by 5% and with magnitude by 3%, and baseline features by 14%. Using all the features together marginally improves accuracy to 67%.

Age prediction

The participants are split into two groups depending on their age. The cut-off is 29 to achieve a close to equal split. The classification statistics are shown in Table 7.1. The results follow similar pattern to the gender prediction task. The topological features on their own achieve classification accuracy of 0.73, closely followed by curvature with 0.67. The baseline features are behind by almost 0.10, with a score of 0.58. Combining the features once again improves accuracy to 0.75. Results for Leave One Group Out test, where the age and gender of an unseen participant is predicted based on data from the others is shown in Supplementary Table A.10.

Model	**Balanced accuracy(Age)**	**Balanced accuracy(Gender)**	**Balanced accuracy(Participant)**
Baseline features	0.57 ± 0.02	0.52 ± 0.03	0.18 ± 0.02
Geometric features without Mag	0.66 ± 0.02	0.59 ± 0.03	0.22 ± 0.02
Geometric features with Mag	0.66 ± 0.02	0.62 ± 0.03	0.34 ± 0.03
Topological features	0.72 ± 0.01	0.65 ± 0.02	0.39 ± 0.03
All Combined without Mag	$\mathbf{0.74 \pm 0.02}$	$\mathbf{0.67 \pm 0.02}$	$\mathbf{0.48 \pm 0.02}$
All Combined with Mag	0.73 ± 0.01	$\mathbf{0.68 \pm 0.02}$	$\mathbf{0.51 \pm 0.03}$

Table 7.1: **Balanced accuracies for age, gender and participant prediction tasks** The topological features outperform the curvature and baseline features across all three tasks, and adding all features together does not improve the accuracy significantly for the age and gender tasks (only 0.02 increase). However, this is not the case for the participant prediction task, where the performance improves with 0.09. These results suggest that the topological information is a good indicator of age and gender. We note that adding magnitude to the curvature-based features further improves the accuracy (apart from in the age prediction task, but since the standard deviations are the same and the difference is 0.02, it did not lead to a significant drop in performance
), and adding magnitude to all features leads to 0.03 increase in the Participant prediction task.

Participant identity prediction

In this task we predict the participant from their papillae. The balanced accuracy of the topological features (39%) are almost double that of curvature features (22%). This is illustrated by the most important features as well, as all three of them are topological. Unlike in the previous two tasks for gender and age, here combining all the features brings a significant improvement in the balanced accuracy score to 48%, suggesting that both the local and global information can contribute to predicting the identity of the participant. Further improvement is noted when we add the magnitude feature to the curvature features, where we notice an improvement of 12%, indicating that magnitude complements the curvature features well in capturing global geometry of the space. In addition, the overall performance improves by further 3% when we add magnitude to all features. Note that while accuracies around 40% to 50% as seen here are not good on binary classification tasks, in this case the task is distinction among 15 participants. A baseline rate of random prediction in this case will produce an accuracy of only 6%. The features thus distinguish participants to a high degree of distinctiveness.

Papillae detection and type classification.

The final result is the accuracy of the classification task for 3-class classification (fungiform vs. filiform vs. none) based on the features (Table A.7). The classification statistics are shown in Table 7.2. The accuracy of the topological features is better than the baseline and the curvature features, and combining all features together provides the best accuracy. We achieve balanced accuracy of 0.72 for the topological, 0.67 for the curvature and 0.62 for the baseline. Combining all the features increases the performance to 0.85.

Application of classification model

The machine learning model developed can be used for accurate papillae detection and positioning on segments from a single person's tongue. Figure 7.5 shows the method accurately positions the fungiform form (in blue) and filiform (in yellow) on a tongue segment from one participant. This automated approach can thus efficiently and accurately construct maps or *tongue prints* from given tongue masks.

	Bal. Acc (SVM)	Bal. Acc (LR)	Bal. Acc (SVM-LOGO)	Bal. Acc (LR-LOGO)
Baseline (height, radius)	0.62 ± 0.03	0.57 ± 0.03	0.59 ± 0.14	0.55 ± 0.11
Geometric (our method)	0.67 ± 0.03	0.60 ± 0.03	0.67 ± 0.05	0.65 ± 0.03
Topological (our method)	0.72 ± 0.03	0.67 ± 0.03	0.72 ± 0.08	0.69 ± 0.08
All Combined	**0.85 ± 0.02**	0.80 ± 0.02	**0.83 ± 0.05**	0.80 ± 0.06

Table 7.2: Comparison of classification results for the classification task for 3-class classification (fungiform vs. filiform vs. none) with random split and using Leave-One-Group-Out (LOGO), where the test data are taken from a single participant and training is carried out on samples from all other participants. The models used are Support vector machines (SVM) and Logistic regression (LR). The standard deviation for the baseline and topological features is larger for LOGO, suggesting that there is higher variation between participants for these feature sets. This is not the case for the curvature features, which appear to be more similar and stable across participants. However, when all the features are combined, the balanced accuracy is improved and the standard deviation is relatively low.

7.1.4 Conclusion

We have presented here the first study of the 3D shapes of human papillae based on high resolution scans. Our study is based on a novel framework combining geometry, topology and machine learning. Past research [Sanyal et al., 2016, Valencia et al., 2016] has focused on fungiform papillae in 2D images. In contrast, our microscale 3D reconstruction based approach can detect filliform papillae and non-papillated areas of the tongue, which are hard to distinguish with the naked eye and 2D images. Recent research has shown that the human perception of food is governed not only by the chemical sensation of *taste*, but also heavily by the mechanosensation, i.e. *texture* perceived by filliform papillae, for example, in the perception of soft textured delicacies such as chocolates [Soltanahmadi et al., 2023]. Of more importance, the framework proposed here can be extended beyond the tongue papillae to the general study of shape and arrangement of microscale surface elements such as finger-like projections that are omnipresent in biology.

To capture the intricate biological shape information, we have developed a pool of geometric and topological features. While 3D geometric and topological transformations have previously been used to process biological scan information [Zhao et al., 2006, Sundaram et al., 2008], we employ a unique approach

and treat them as statistical data that are fed to a machine learning system. In addition, magnitude-based features have not been used in machine learning. In this approach, curvature statistics are used for aggregated local information, while persistent homology and magnitude are used for global characteristics. Based on the subject of study, other features may be used. In our analysis topological features turn out to be more informative in prediction. Recent research [Bubenik et al., 2020] has suggested that persistent homology can capture local shape information as well as global properties. Our results on tongue papillae are consistent with this idea.

The analytics are based on machine learning models. The models themselves are built to predict the relevant variables of type, age, gender and participant, but our objective was to gain a better understanding of variations across classes and features. We thus used permutation feature importance to evaluate how each feature contributes to each model. From a pure accuracy point of view, large neural network models [Andreeva et al., 2020] trained on big datasets are considered the most successful current paradigm [Shahid et al., 2019]. However, our objective in this study was to develop an interpretable framework for investigation of biological surface features, operating on relatively few samples from few participants. We have thus used simpler models that can be trained with smaller quantities of data. The accuracy of the results with simple models gives us confidence in our conclusion of feature importances and in the feasibility of highly accurate machine learning models in future research.

The tasks for prediction of age group and gender suffer from the small number of participants. Machine learning models for these tasks achieve balanced accuracies of approximately 74% and 67% respectively. Note that for such binary prediction, a random prediction model achieves 50% accuracy. The results suggest that geometric and topological features do vary to an extent across these variables, but more data will be needed to confirm the result and the nature of variation. The higher max Gaussian curvature appears as an important feature for female participants and the younger age group, suggesting more sharply curved or pointy shapes in these demographics. In past research, women and younger people have been noted to have higher density of fungiform papillae, which has been attributed to variations in taste perception, and women have been observed to be supertasters more frequently [Bartoshuk et al., 1994, Fischer et al., 2013, Zhang et al., 2009]. The curvature variation implies a difference in papillae shapes that could be

contributing to the sensory differences as well. Fungiform papillae density has been noted [Karikkineth et al., 2021] to drop above an age of 65. In our study the participants were within the relatively young range of $22-37$. The shape features show some variations to reach a classification accuracy of 74% between age groups $22-28$ and $29-37$. The Leave One Group Out test on age and gender (Supplementary Table A.10) shows lower accuracy and greater variability. Certain individuals seem harder to model in this task. Further investigation with more participants will be required to gain greater insight into this issue.

The papillae type detection results are more accurate at 85% and based on a large number of papillae, which gives us confidence that the model is truly accurate. To confirm that the models generalise to unseen participants, we carry out the Leave One Group Out test, and find that the accuracy holds up even on samples from a completely unseen participant, which confirms that the models can be used to classify and localise papillae on new tongue impressions. The papillae type model can thus be used to automatically identify filiform and fungiform papillae on scans of new tongue impressions.

The individual participant model shows 48% balanced accuracy and 51% raw accuracy. This score is not impressive in a binary classification task, but our participant prediction task is a multi-class one, with 15 possible classes. A papilla could have belonged to any one of the 15 classes, and a random predictor would have an accuracy of only 6.66%. Considering the sample sizes from different participants, (Table A.8) a predictor that always predicts the largest class can achieve an accuracy of 11%. In comparison, the model achieves between 4 to 8 times the accuracy of these baselines based on the distinctiveness in the data of a single papilla. This distinctiveness may have multiple contributing factors – these can be true inter-individual variations as well as variations in experimental conditions in collecting the masks. The exact cause of this difference will require further study. Note that while the age, gender and participant identification tasks suggest unique individual characteristics, the success of the type identification task suggest a complementary conclusion of significant similarity within types and across individuals. Larger studies can potentially address some of these issues using larger models and more complex features, such as persistent homology of curvature functions.

The framework and discriminative models presented here enable deeper study of the papillae structure and their variations and arrangements. The model

for localising and classifying papillae (as seen in Figure 7.5) enables the study of the overall tongue surface, or *tongue prints*. Such arrangements of papillae are known to influence the surface properties of the tongue and its perception abilities [Andablo-Reyes et al., 2020]. Our data and past research have shown that the distribution of papillae vary across individuals. A detailed study of this variation across various demographic parameters could reveal insights into preferences, cultures and medical conditions. Arrangements identified by our models could be used to build generative models that can fuel such insights and can create more realistic surfaces for use in food engineering and development of oral diagnostics. Ultimately, this study offers a new dimension showing papillae as an unique identifier for the first time in the literature which needs further validation using this developed method for a larger dataset of participants.

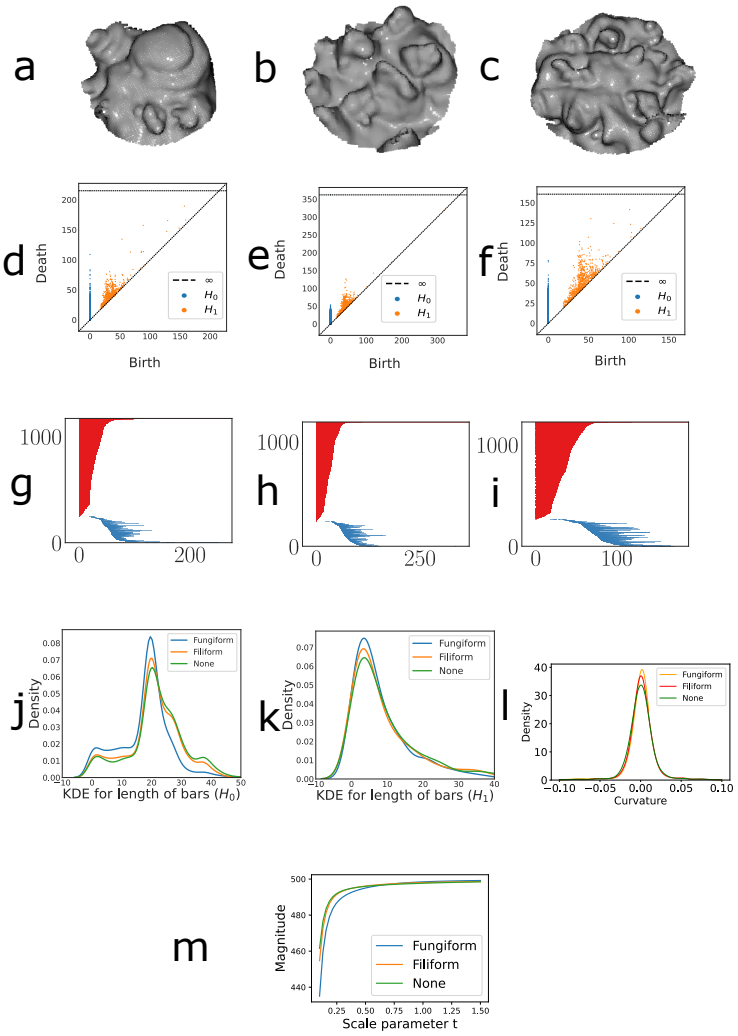


Figure 7.2: **Papillae identification and topological feature characterization** Plots (a-c) show how the candidates for papillae from one Participant (Participant id 3) as meshes, using the library `open3d`. They are representatives from the 3 classes (a) Funigform, (b) Filiform and (c) None – no papilla. Plots (d-f) show their respective topological representations of (a-c) in the form of persistent diagrams measuring two main topological features: H_0 – the connected components and H_1 – the equivalent loops. Plots (g-i) show the equivalent representation of the persistent diagram in the form of a barcode, where the bars in red correspond to the connected components and the bars in blue – to the loops. Each bar represents a persistent generator, which is an interval where its left end point corresponds to the first filtration level where this topological feature appears, and its right end point is the filtration level where it disappears. Plots (j) and (k) show the kernel density estimate (KDE) using Gaussian kernels – plots representing the distribution of the lengths of bars from the barcode (for a,b and c.). Plot (l) reveals the curvature distribution across the different labels, and plot (m) shows that there is a difference between the magnitude functions of fungiform and filiform.

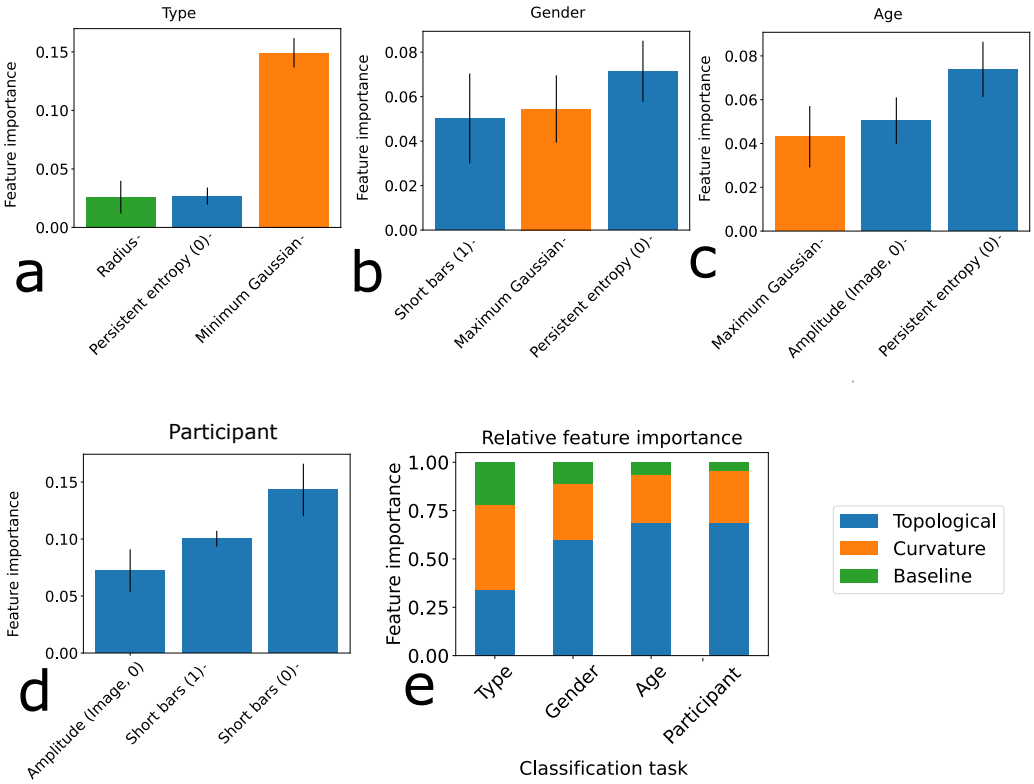


Figure 7.3: **Feature importance across the classification tasks** The plots (a-d) represent the three most important features in the individual classification tasks. In particular, in plot (a) we see the papillae type task feature importance, in plot (b) Gender task features ordered by importance, in plot (c) the Age task features and in plot (d) the participant task features ordered by importance. The x-axis represents the accuracy drop when the feature of interest is permuted, and the black line represents the standard deviation over 30 runs. In plot (d) we see the relative importance of all features from each kind in each task. The curvature followed by topological features are the most important for papillae type classification; the topological are the most important for the Gender classification task; the topological are even more important for the Age classification task. We note the growing relative importance of topological features from 0.34 to 0.69 and the diminishing importance of the baseline features from 0.22 to 0.04, from left to right. The curvature features are the most important for the Type task with 0.44 and maintain consistent medium importance across the Gender, Age and Participant prediction task with 0.28, 0.25 and 0.27, respectively.

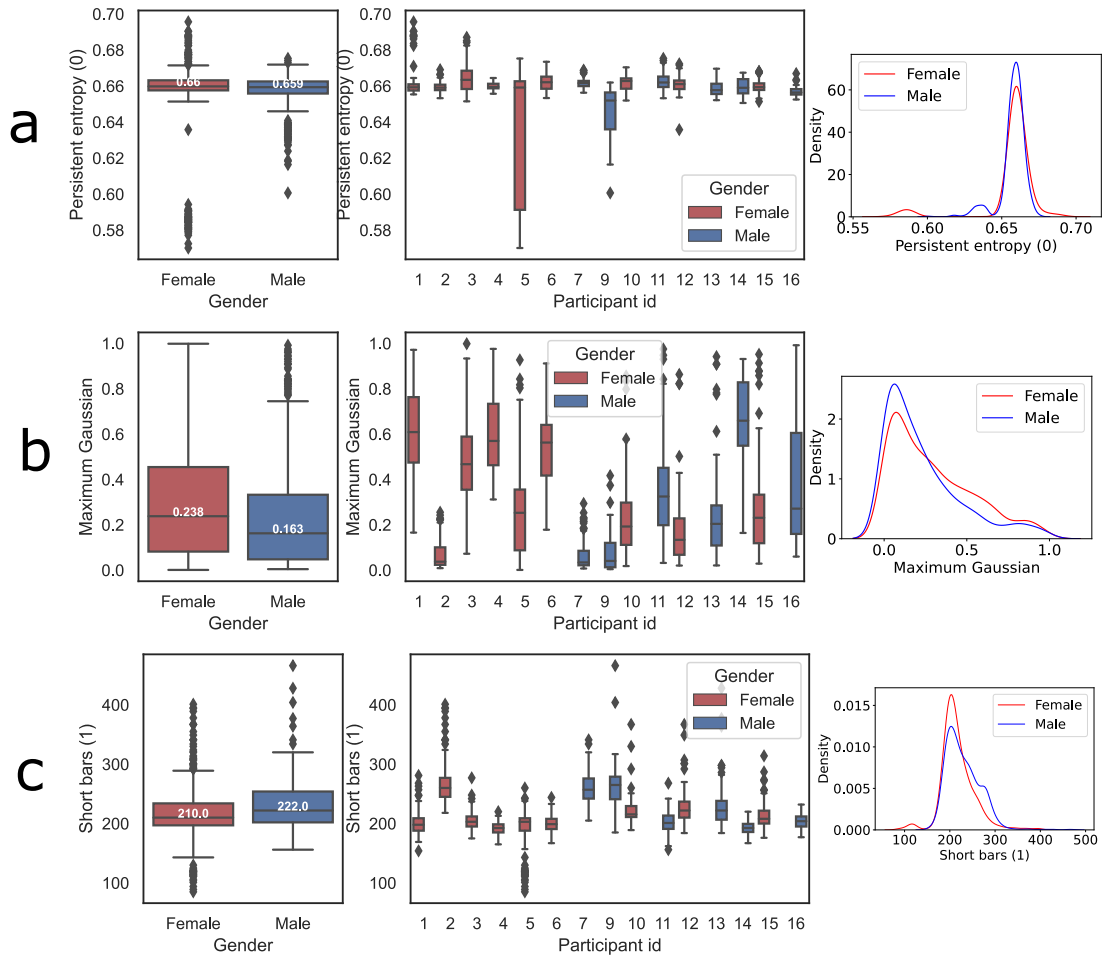


Figure 7.4: **Important features for gender classification** The most important features for gender classification and its aggregate distribution. Both the aggregate and individual distributions show that the females have lower number of short bars than males.

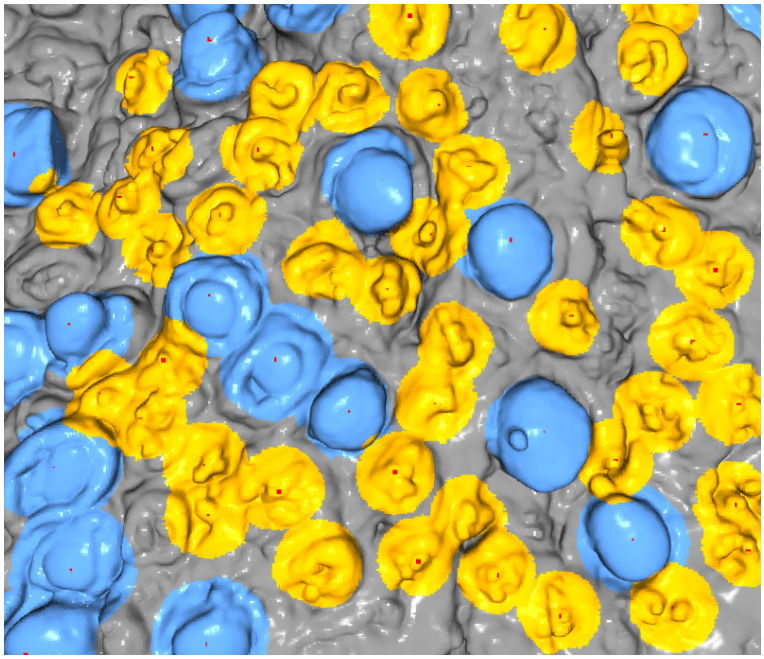


Figure 7.5: **Automatic identification of tongue papillae** Illustration of the result of our tool for positioning papillae on the surface of the human tongue. Here our tool has detected the positions of fungiform (in blue) and filiform (in yellow) on the tongue surface. It has found 14 fungiform and 40 filiform papillae. As a red dot we see the centre of the papillae, which is determined as the local maxima for the structure with the highest distance from a fitted plane, using the RANSAC algorithm.

7.2 Detecting age from brain artery trees

Changes in the network of blood vessels (also known as vasculature) are often the first signs of development of diseases like Alzheimer's or stroke. If we are able to develop methods aimed at identifying these alterations, we will be better equipped to treat early these conditions and to develop preventative therapies. With ageing, brain vasculature changes and it is important to be able to recognise and quantify such changes. Previous studies have demonstrated the usefulness of topology for the age detection problem [Bendich et al., 2016]. Brain vasculature has been found to be correlated with age from 2 different analysis methods — statistical and TDA. Methods from the former have found age to be correlated with total artery length [Gutierrez et al., 2016], which forms the basis of our benchmark classification model, together with gender and handedness of the subjects. On the other hand, TDA has been useful in identifying correlation between age and the positions of arteries in space in a way that statistical analysis is not capable of discovering [Bendich et al., 2016].

In this part, we have focused on deriving a number of features based on Persistent homology (PH) and magnitude, and building a machine learning model capable of distinguishing between brain artery trees coming from 2 different age groups with 72.4% accuracy, compared to 71.3% for the benchmark of pre-computed biomarkers as part of the dataset. Using only magnitude features, we achieve 69.4%, but the best result is achieved when the PH-based features are combined with magnitude: then we achieve 73.5% accuracy. An example of a reconstructed image of an artery tree is shown below (image taken from [Bendich et al., 2016]).

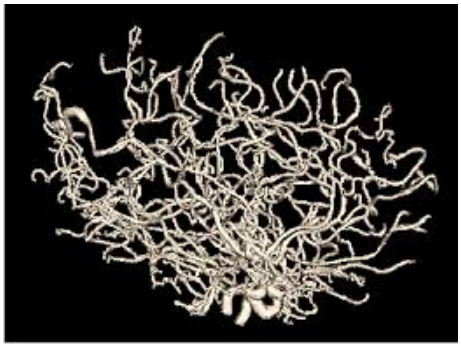


Figure 7.6: Brain artery trees. Image taken from [Bendich et al., 2016].

7.2.1 Topology and Geometry are important

The main approach adopted in this part is on taking into account the geometry and topology of the brain artery trees. To achieve this goal, we use topological data analysis (TDA) and magnitude. Both TDA and magnitude extract numerical features that quantify the shape of point clouds. A point cloud may have different connected components, loops, voids, all these peculiar structures can be extracted via TDA techniques; and for magnitude, summarising the brain artery tree by the effective number of points at different scales seems like a reasonable choice given the geometric nature of our task at hand. For this purpose, we would like to first study how brain vasculature changes with age for people without accompanying brain conditions.

7.2.2 Dataset

In this study we are comparing the arteries between non-pathological cases of 98 brains belonging to individuals of age between 18 and 79. We will explore the point clouds of brain artery trees, extract topological features which would allow us to quantify looping and branching at multiple levels and train a classifier to distinguish between 2 age groups: group 0 with people younger than 45, and group 1 with people older than 45. It has been suggested in [Bendich et al., 2016] that topological features can help in identifying the age of a person's brain, with younger people having significantly longer total artery length and more loops in it.

The dataset consists of 98 artery trees which are the result of applying a tube-tracking algorithm to the 3-dimensional Magnetic Resonance Angiography (MRA) images of the brain. Each tree consists of approx. 120000 vertices, branches and edges. For the purpose of this analysis, we have only used a point cloud of all the vertices for initial visualisation and we have randomly downsampled to 500 vertices per tree in order to reduce the computational time. The labels in the dataset depend on the age of the subject: label 0 means that the subject is younger than 45, while label 1 means that the subject is older than 45.

In Figure 7.7, we see the label distribution of the dataset. We have a well-balanced dataset with almost 50-50 split between the 2 age groups.

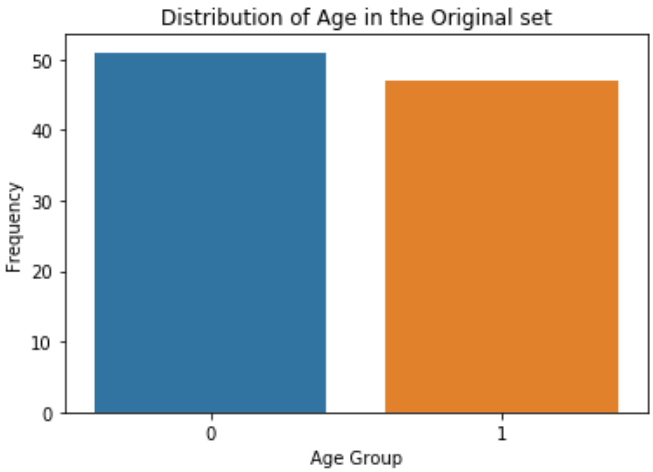


Figure 7.7: Age distribution of the dataset with label 0 means that the subject is younger than 45 while label 1 means that the subject is older than 45.

7.2.3 Methods

The core of topological data analysis are persistence diagrams. These complicated two dimensional diagrams describe the topological properties of a point cloud in a scale-invariant way. We proceed by calculating the persistence diagrams for homology dimension 0 (which accounts for the connected components) and 1 (which is linked to 1-dimensional loops) only, as higher homology groups require a lot of time for computation.

In Figure 7.8(c-d) we plot the persistence diagrams of the two brain artery trees from Figure 7.8(a-b). For homology in dimension 1, we can see that there are more points further away from the diagonal in the case of the younger brain. This signifies that there are more loops and the structure of the arteries is loopier than that of the older brain. In 7.8(e), we compare the magnitude function of the same initial artery trees, and we note that at the lower values of the scale parameter t, the magnitude functions can be distinguished from each other.

From PD to topological features for machine learning. The computed PDs are a multiset and a metric space with the p-Wasserstein distance, but this space is neither a Euclidean space nor a Hilbert space, which is needed for machine learning methods such as SVM or PCA (where the input data is supposed to be in the Euclidean space or Hilbert space). Hence, we cannot use them directly as inputs for a variety of machine learning methods. Therefore, we convert each diagram to a 2-dimensional vector with persistent entropy [Atienza et al., 2021].

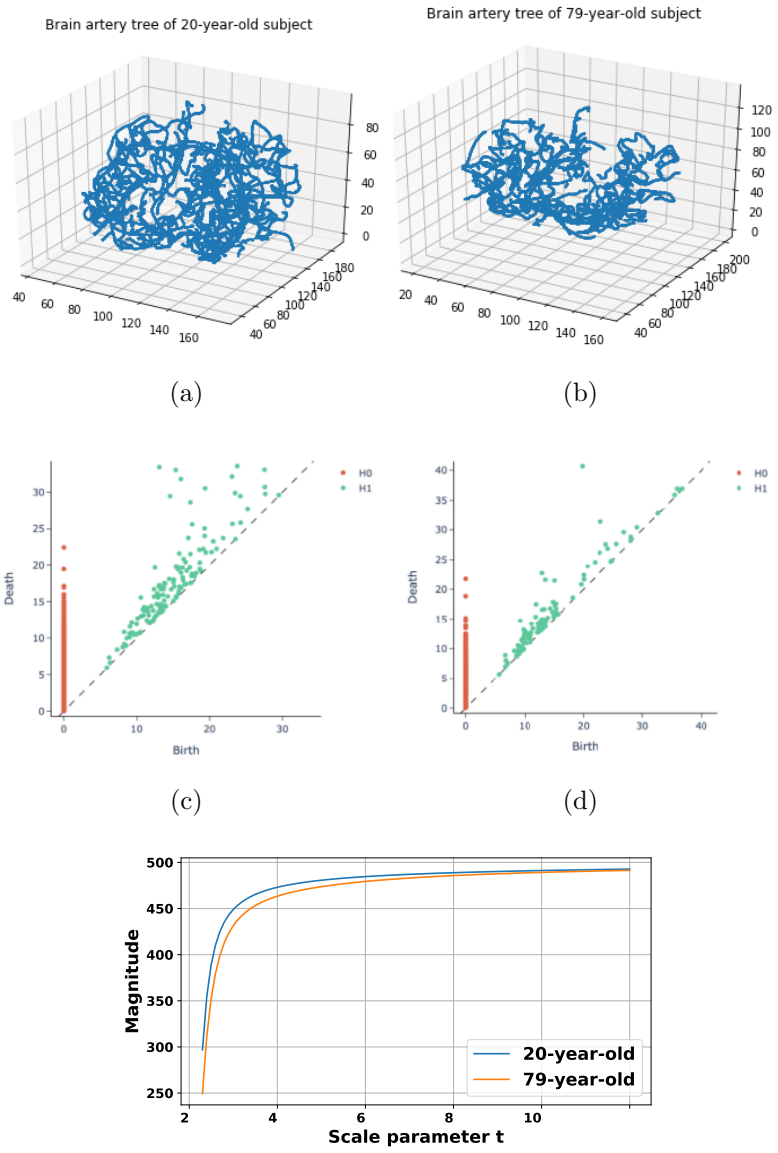


Figure 7.8: **Magnitude and TDA distinguish differences between the brain artery trees of 20- and 79-year old subjects.** In this figure two artery trees are displayed: the artery tree of a 20 years old subject (a), and in (b) we see the brain artery trees of a much elderly subject. In (c) we see the PD of the artery tree of a 20 years old subject from (a). In (d) we see the PD of the brain artery trees of a much elderly subject from (b). In (e) we see the magnitude function plot, allowing us to see difference between the patterns of the younger and older brain in terms of effective number of points.

This generates 2 topological features per persistence diagram: one persistent entropy per homology dimension.

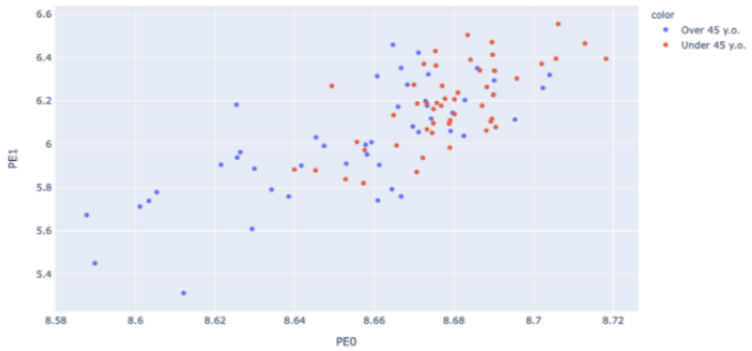
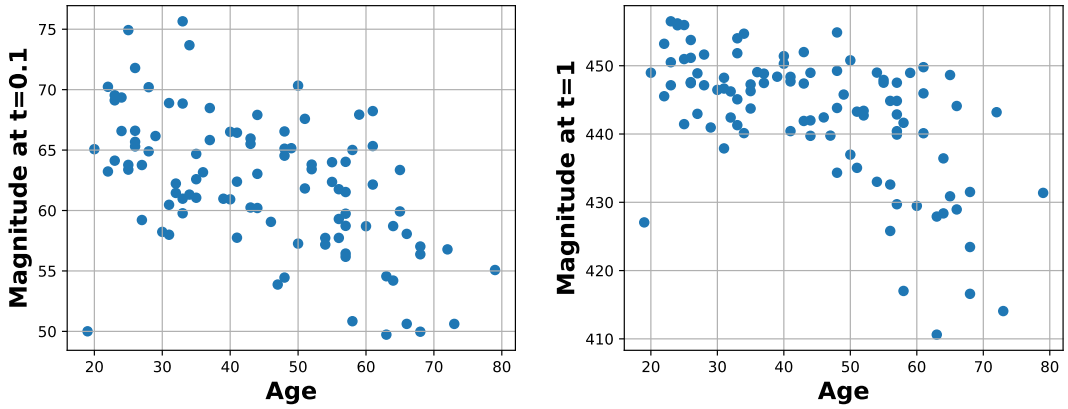


Figure 7.9: Scatter plot of the persistent entropies of the young and old patients.

We are thus able to generate simple numerical feature out of the persistence diagram of a brain. However, when we plot the features in Figure 7.9, there are not any distinct clusters: as a result we cannot expect to have great classification performance based on only these two topological features. What we could try is to add more topological features in order to improve the performances.

In order to improve the separation of the classes, we decided to add four different topological features: they are called amplitudes. The amplitude of a persistence diagram is the result of computing the distance — in a given metric — of such diagram from the empty diagram. Depending on which metric is used in the diagram space, such amplitude value may differ. We consider four different metrics to compute a vector of amplitudes for each persistence diagram: 'bottleneck', 'wasserstein', 'landscape', 'persistence image'. In order to complete the feature engineering step, we also add the number of off-diagonal points per homology dimension. Our intuition relies on the geometric nature of the problem: we believe that to improve the results of a classification tasks, we can add topological features to the dataset. We achieve classification accuracy of 0.735.

From magnitude function to magnitude-based features for machine learning. Here we describe how to use the magnitude function for machine learning. We compute magnitude for 100 intervals evenly spaced between $[0.3, 10]$



(a) (b)

Figure 7.10: **The magnitude of brain artery trees.** In plot (a), we see the scatter plot of Age vs. magnitude at scale value of $t = 0.1$. We notice that there is moderate correlation. In plot (b), we plot Age vs. magnitude at a larger scale value, $t = 1$, and notice weaker correlation with age.

7.2.4 Results

We compare the classifier with topological features with a simple baseline based on a geometric measure — the total artery length. We add two more features, which are gender and handedness (which is tendency to use either the right or the left hand more naturally than the other) which we have available for the dataset. The accuracy is 0.7 for the baseline. Adding both magnitude features at scale $t = 0.1$ and $t = 1$ has the same performance as the total artery length. The topological features outperform the magnitude and baseline features, but when we combine them with magnitude at scale $t = 1$, we achieve the best accuracy of 0.74, indicating that both topological and geometric information leads to the best overall performance. We can see the results in Table 7.3.

We note a 2% increase in performance in the topological features compared to the baseline, and 4% increase when we combine the topological features with magnitude. However, it should be noted that the original dataset contained more than 120K points. The choice to downsample was for computational reasons so that the code can run in less than 15 mins. Attempts with 3000, 2000, 1000 points were made, but the time it took was substantially longer. The results are expected to be better when we have more data points.

7.2.4.1 Magnitude negatively correlates with age

In Figure 7.10, we plot the values of magnitude at two different scales, $t = 0.1$ and $t = 1$, against the age of the participants. We note that magnitude correlates negatively with age, with Pearson correlation coefficient of -0.52 with significant p-value ($(p << 0.05)$); and -0.58 with again significant p-value ($p << 0.05$), which is moderate to strong correlation.

	Accuracy
Total artery length	0.70
Magnitude features	0.70
Topological features	0.72
Topological and Magnitude features	0.74

Table 7.3: Classification results for the benchmark (total artery length) and topological features based on a random subsample of the vertices.

7.2.5 Conclusion

We have obtained interesting results regarding the relationship between brain artery trees and age, and developed a useful classification model which distinguishes between younger and older brains with 73.4% accuracy. Adding magnitude at a set scale to the topological features slightly outperform the geometric measurement of total brain artery length in the age classification task. However, we are only using a small subsample of 500 of the available vertices (out of 120k), which suggests that there might be more accurate classification when one includes more points. It would be interesting to see if adding more points to the point clouds would change the accuracy and if so, by how much. The caveat will be a longer time to produce the full set of persistence diagrams as well as magnitude, with approximately 1 minute per diagram for a point cloud of 3000 points, which would result in 90 mins for the entire dataset. Further methods for speeding up this process could be investigated, together with a more accurate sampling procedure, as we used random subsampling. We could use the approximation methods for magnitude developed in Chapter 3 of this thesis. In addition, investigating the most relevant scales for magnitude by using feature importance analysis, or using MagArea, as developed in Chapter 6, which is a full-scale summary of magnitude, might result

in more accurate classification results. Moreover, in further analysis we should also calculate the homology of dimension 2, as that could provide additional insight. We have considered the ageing effect on healthy individuals' brain arteries. In future work it would be of clinical interest to develop topological techniques for detecting brain abnormalities linked to brain conditions such as stroke and Alzheimer's.

Chapter 8

Conclusion

The widespread adoption of ML in industry and scientific research shows that it is not just a passing trend but a fundamental part of modern technology. In this thesis we have sought to gain deeper insights into ML models via study of their associated ML spaces. This was done through the use of magnitude as a new and powerful tool to examine the underlying geometry of the relevant ML spaces. However, magnitude is not limited in its applications to ML spaces, it can be applied to any metric space. This is a very general condition, which many important data spaces satisfy, in particular biomedical data can often be seen as a metric space. For these reasons we have striven throughout this thesis to emphasise the utility and versatility of magnitude as a data analysis tool.

The first challenge encountered when using magnitude is the high computational cost. Inverting high-dimensional matrices is a computationally intensive problem. The first section of this thesis was dedicated to providing fast algorithms to compute the magnitude of arbitrary metric spaces whilst minimising the loss of accuracy as possible. These faster algorithms allow for the application of magnitude to many problems that previously it was unsuitable for. A selection of areas that we demonstrated the utility of these fast algorithms on included regularisation on deep neural networks, clustering and larger training trajectories.

With the practical challenges of magnitude resolved, we then considered another metric space of interest to researchers; the model space. The goal was to show a direct link between magnitude and the generalisation error. This would then imply that magnitude would allow us another tool with which to understand the reliability of AI. Here, we proved a result that there is a direct link between the magnitude dimension and the generalization error. This has ramifications for the

wider field of AI. More precisely, the result states that for accurate and robust AI predictions it is essential that the training trajectories have as small a magnitude dimension as possible. While progress was made into understanding the link between magnitude and the generalisation error, further clarity was required.

Through experimentation, it became evident that there was a strong connection between generalisation error and magnitude. The difficulty lay in making this connection precise. The solution was to introduce a refined version of magnitude which we call *positive magnitude*. One of the main results of Chapter 4 was a formula providing an upper bound for the generalisation error. From this expression it is clear to see that the upper bound is directly correlated with the positive magnitude. Once again, this reaffirms the idea that to have an accurate AI, we require a small positive magnitude. The impact of this is significant. As AI models continue to grow in importance, it is essential that we use every means available to understand them, and we have shown one such way to gain insight into this.

Continuing with the theme of applying magnitude to important metric spaces for ML, we considered its application to latent spaces. For this purpose we developed a family of new diversity measures derived from magnitude. They serve as excellent metrics for measuring the diversity of latent representations and outperform existing measures both when a reference dataset is available and in its absence. In the first case, we find that magnitude captures mode collapse and mode dropping better than existing metrics for evaluating generative models for both image and graph modalities. In addition, we demonstrate that magnitude can measure curvature better than PH. This is a particularly exciting result and warrants further investigation.

In the broader area of AI, capturing mode collapse is important. When an AI model experiences mode collapse, its outputs become predictable and limited, reducing the system's reliability. In showing that magnitude captures mode collapse, we have paved the way towards a procedure where one can embed our proposed metrics during the optimisation procedure of the model. This will ensure that the impact of mode collapse is limited and will result in more reliable AI models.

Finally, the applications of magnitude to biomedical data was considered. In particular, data sets of 3D scans of tongue surfaces and brain artery trees were examined. An interpretable framework was constructed for investigation

of biological surface features, operating on relatively few tongue samples from a small number of participants. We used a number of features based on PH and magnitude to capture the topology and geometry of the underlying shapes. We used the features to predict the age, gender and identity of the participants in the dataset. The results were impressive, predicting the age and gender with balanced accuracy of approximately 74% and 67% respectively. In the participant prediction task, we achieved 48%, and noticed an improvement of 12% when adding magnitude to the curvature features. This result indicates that magnitude complements the curvature features in capturing global geometry of the space.

By applying almost the same methods and techniques on the brain artery trees data, we achieved similar results in terms of predicting age. We developed a useful classification model which distinguishes between younger and older brains with 73.4% accuracy. Adding magnitude at a set scale to the topological features outperforms the geometric measurement of total brain artery length in the age classification task.

The work contained inside this thesis opens many exciting avenues for exploration. In this work, we drew parallels and comparisons to PH in the cases wherever appropriate. In particular, we find that using magnitude might be more beneficial than PH, such as in computing curvature. The comparison between PH and magnitude has been a consistent theme within our investigation, but many open questions on comparing the two remain. For example, we have compared magnitude dimension and PH dimension for measuring the known fractal dimension of a Levy-stable process. However, a larger scale and more thorough investigation on the benefits of using magnitude-based, PH-based dimensions or a combination of both [Govc and Hepworth, 2021, O'Malley et al., 2023] is needed. Furthermore, they should be compared to other fractal dimensions such as box-counting and correlation dimensions, similar to the work of Jaquette and Schweinhart [2020].

There is still further investigation into the usage of magnitude that could lead to more advanced progress in related areas. For example, is it best to use magnitude at a carefully selected scale, or to use the full-scale magnitude summary? In addition, we have proposed positive magnitude as a generalization measure, however we have not studied its theoretical properties in more detail. Moreover, integrating magnitude-based diversity metric into ML models might be beneficial for controlling the output of generative models.

This dissertation is an invitation to the wider adoption of magnitude. We

have only began to scratch the surface of its potential applications. There are still numerous unexplored avenues in ML, where this versatile and powerful geometric concept can truly excel and generate unique insight, and be all you need.

Appendix A

Appendix

A.1 Appendix for Chapter 3

A.1.1 Proofs of Theorems

Proof of Theorem 3.3.1

Proof. We note that X is a homogeneous metric space, thus using Proposition 2.1.5 from [Leinster, 2013] we can calculate

$$Mag(X) = \frac{2D}{1 + e^{-2t} + 2(D-1)e^{-t\sqrt{2}}}$$

Next, we consider the similarity matrix of $X \cup \{0\}$ to calculate its magnitude.

$$\zeta_{X\cup\{0\}} = \begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix}$$

where A_1 is the similarity matrix for X,

$$A_2 = \begin{pmatrix} e^{-t} \\ e^{-t} \\ \vdots \\ e^{-t} \end{pmatrix} \quad A_3 = A_2^T \quad A_4 = \begin{pmatrix} 1 \end{pmatrix}.$$

We will use the inverse formula:

$$\begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix}^{-1}$$

$$= \begin{pmatrix} A_1^{-1} + A_1^{-1}A_2B^{-1}A_3A_1^{-1} & -A_1^{-1}A_2B^{-1} \\ -B^{-1}A_3A_1^{-1} & B^{-1} \end{pmatrix}$$

where $B = A_4 - A_3A_1^{-1}A_2$

As X is homogeneous, we observe that the sum of rows of A_1^{-1} give $\frac{\mathrm{Mag}(X)}{D}$, so each entry of $A_1^{-1}B$ is $e^{-1}\frac{\mathrm{Mag}(X)}{D}$, and by symmetry the same is true for CA^{-1}. Thus $B = \frac{1}{1-e^2\mathrm{Mag}(X)}$. It then follows that the magnitude of $|X \cup \{0\}|$ is

$$\frac{(1-2e^{-t})\mathrm{Mag}(X)+1}{1-e^{-2t}\mathrm{Mag}(X)}.$$

When this expression is expanded, L'Hopital's rule then gives that

$$\lim_{D\to\infty} \mathrm{Mag}(X\cup\{0\}) - \mathrm{Mag}(X) = \frac{8(e^{2t}-e^{t(1+\sqrt{2})})^2}{8(e^{4t}-e^{t(2+\sqrt{2})})} = \frac{(e^t-e^{t\sqrt{2}})^2}{e^{2t}-e^{t\sqrt{2}}}.$$

□

Definition A.1.1. Let $x, y \in \mathbb{R}^n$ be such that $x_1 \geq x_2 \geq ... \geq x_n$, $y_1 \geq y_2 \geq ... \geq y_n$ and $\sum_{i=1}^n x_i = \sum_{i=1}^n y_i$. Then we say that x majorises y if for all $k = 1, ..., n$

$$\sum_{i=1}^{k} x_i \geq \sum_{i=1}^{k} y_i.$$

Theorem A.1.2 (Karamata's inequality)**.** *Let I be an interval on $\mathbb{R}$ and let $f : I \to \mathbb{R}$ be concave. If $x_1, ..., x_n$ and $y_1, ..., y_n$ are numbers in I such that $(x_1, ..., x_n)$ majorises $y_1, ..., y_n$ then*

$$f(x_1) + ... + f(x_n) \leq f(y_1) + ... + f(y_n).$$

Proof of Theorem 3.3.2

Proof. Given a set X, we can write $B = \{b_1 < ... < b_n\} = X \cup \{-\infty, \infty\}$. The formula provided in corollary 2.3.4 from [Leinster, 2013] gives that

$$\mathrm{Mag}(X) = \sum_{i=1}^{n} \tanh\left(\frac{b_{i+1}-b_i}{2}\right) - 1.$$

Note that here we define $\tanh\infty = 1$.

Then given the points $x_1 < x_2$ such that $x_1 \in (b_j, b_{j+1})$ and $x_2 \in (b_k, b_{k+1})$, we calculate that

$$\begin{aligned}&\mathrm{Mag}(X \cup \{x_1\})\\ &= \sum_{i=1}^{n} \tanh\left(\frac{b_{i+1} - b_i}{2}\right) - 1 - \tanh\left(\frac{b_{j+1} - b_j}{2}\right)\\ &\quad + \tanh\left(\frac{b_{j+1} - x_1}{2}\right) + \tanh\left(\frac{x_1 - b_j}{2}\right).\end{aligned}$$

$$\begin{aligned}&\mathrm{Mag}(X \cup \{x_2\})\\ &= \sum_{i=1}^{n} \tanh\left(\frac{b_{i+1} - b_i}{2}\right) - 1 - \tanh\left(\frac{b_{k+1} - b_k}{2}\right)\\ &\quad + \tanh\left(\frac{b_{k+1} - x_2}{2}\right) + \tanh\left(\frac{x_2 - b_k}{2}\right).\end{aligned}$$

If $j \neq k$ then

$$\begin{aligned}\mathrm{Mag}(X \cup \{x_1, x_2\}) &= \sum_{i=1}^{n} \tanh\left(\frac{b_{i+1} - b_i}{2}\right) - 1\\ - \tanh\left(\frac{b_{j+1} - b_j}{2}\right) &+ \tanh\left(\frac{b_{j+1} - x_1}{2}\right) + \tanh\left(\frac{x_1 - b_j}{2}\right)\\ - \tanh\left(\frac{b_{k+1} - b_k}{2}\right) &+ \tanh\left(\frac{b_{k+1} - x_2}{2}\right) + \tanh\left(\frac{x_2 - b_k}{2}\right).\end{aligned}$$

It follows that

$$\begin{aligned}&\mathrm{Mag}(X \cup \{x_1\}) + \mathrm{Mag}(X \cup \{x_2\})\\ &= \mathrm{Mag}(X \cup \{x_1, x_2\}) + \mathrm{Mag}(X).\end{aligned}$$

If $j = k$, then

$$\begin{aligned}|X \cup \{x_1, x_2\}| &= \sum_{i=1}^{n} \tanh\left(\frac{b_{i+1} - b_i}{2}\right) - 1\\ &- \tanh\left(\frac{b_{j+1} - b_j}{2}\right) + \tanh\left(\frac{b_{j+1} - x_2}{2}\right)\\ &+ \tanh\left(\frac{x_2 - x_1}{2}\right) + \tanh\left(\frac{x_1 - b_j}{2}\right).\end{aligned}$$

So

$$\mathrm{Mag}(X \cup \{x_1\}) + \mathrm{Mag}(X \cup \{x_2\}) - \mathrm{Mag}(X \cup \{x_1, x_2\}) - \mathrm{Mag}(X)$$

$$= \tanh\left(\frac{b_{j+1} - x_1}{2}\right) + \tanh\left(\frac{x_2 - b_j}{2}\right) -$$

$$\tanh\left(\frac{b_{j+1} - b_j}{2}\right) - \tanh\left(\frac{x_2 - x_1}{2}\right).$$

We observe that $\tanh(x)$ is concave on $x \geq 0$ and that since $b_{j+1} - b_j \geq b_{j+1} - x_1$ and $b_{j+1} - b_j \geq x_2 - b_j$ and $b_{j+1} - b_j + x_2 - x_1 = b_{j+1} - x_1 + x_2 - b_j$, $(x_2 - x_1, b_{j+1} - b_j)$ majorises $(b_{j+1} - x_1, x_2 - b_j)$. Thus by Karamata's inequality,

$$\tanh\left(\frac{b_{j+1} - x_1}{2}\right) + \tanh\left(\frac{x_2 - b_j}{2}\right) -$$

$$\tanh\left(\frac{b_{j+1} - b_j}{2}\right) - \tanh\left(\frac{x_2 - x_1}{2}\right) \geq 0.$$

Thus it follows that

$$\mathrm{Mag}(X \cup \{x_1\}) + \mathrm{Mag}(X \cup \{x_2\}) \geq \mathrm{Mag}(X \cup \{x_1, x_2\}) + \mathrm{Mag}(X).$$

Hence magnitude on X is submodular. □

A.1.2 More Experimental Results and Details

Larger datasets In Figure A.1, we see a comparison between matrix inversion and Iterative Normalizaion for 2×10^4 points sampled from $\mathcal{N}(0, 1)$ in $\mathbb{R}^2$ over 5 runs. Iterative Normalization is run for 10 iteration, as we observe fast convergence towards the true magnitude value.

The experiment was ran on a NVIDIA 2080Ti GPU with 11GB RAM.

A.1.2.1 Further investigation of SGD algorithm

We experimented with multiple different batch sizes, but full size of the dataset, or Gradient Descent (GD) appears to achieve fastest convergence. For completeness, here we report the convergence results when we vary the batch size. Note that while this method appears to be slower than Iterative Normalization, it can be used when the size of the dataset cannot fit into memory.

Batch sizes are $= \{8, 16, 32, 64, 128, 256\}$ and show in how many iterations the algorithm converges, and learning rate is fixed at 0.01. Figure A.2 shows mean and standard deviation, repeated over 10 runs, for 50 iterations, for a dataset with 2000 points sampled from $\mathcal{N}(0, 1)$ in $\mathbb{R}^2$.

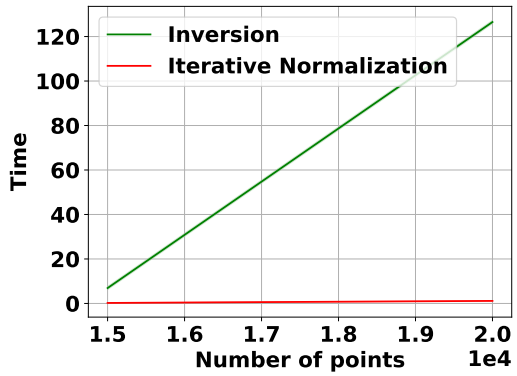


Figure A.1: Time is measured in seconds. It takes 1.12 seconds for Iterative Normalization to execute for 2×10^4 points, while Inversion requires 126.9 seconds.

It appears that all batch sizes tend to converge towards the true magnitude value after iteration 20.

Experiments ran on a NVIDIA 2080Ti GPU with 11GB RAM.

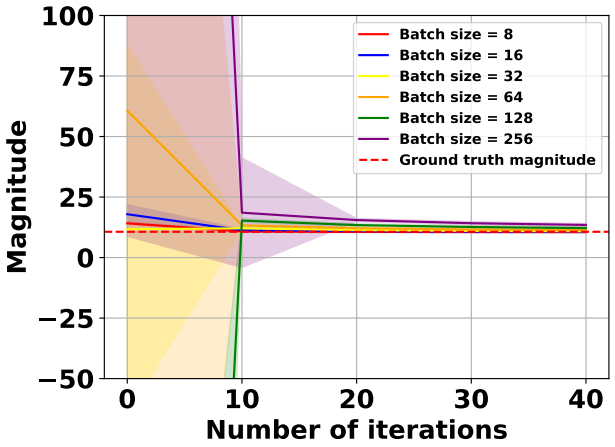


Figure A.2: Varying the batch sizes and its effect on convergence towards the true magnitude value.

A.1.2.2 Experimental details

Algorithm comparison We use `PyTorch`'s GPU implementation for matrix inverison. The SGD experiments used a learning rate of 0.01 and momentum of 0.9. We set the size of the batch to equal the size of the dataset for the GD experiments in the main chapter.

Training trajectories and generalization We use a modified version of the

ViT for small datasets as per [Raghu et al., 2021]. The implementation is based on the [Gani et al., 2022], which is based on the `timm` library with the architecture parameters as follows: depth of 9, patch size of 4, token dimension of 192, 12 heads, MLP-Ratio of 2, resulting in 2697610 parameters in total, as described in more detail in Chapter 5.

We start from a pre-trained weight vector, which achieves high training accuracy on the classification task. By varying the learning rate in the range $[10^{-5}, 10^{-3}]$ and the batch size between $[8, 256]$, we define a grid of 6×6 hyperparameters. For each pair of batch size and learning rate, we compute the training trajectory for 10^4 iterations. We use the Adam optimizer [Kingma and Ba, 2017]. We compute the data-dependent pseudometric, first defined in [Dupuis et al., 2023] by $\rho_S^{(1)}(w, w') = r^{-1}||\boldsymbol{L}_S(w) - \boldsymbol{L}_S(w')||_1$, to obtain a distance matrix. Then we proceed to compute the quantities of interest $\mathrm{Mag_n}$ and $\mathrm{PMag_n}$ for $n = \{5000, 7000, 10000\}$, using the distance matrix as derived from the pseudometric $\rho_S^{(1)}$. We set the magnitude scale $t = \sqrt{r}$, where r is the size of the training set ($r = 50000$ for CIFAR10). This value is motivated by the theory in Chapter 5, and for a fair comparison with their methods. We then compute the granulated Kendall's coefficients ($\boldsymbol{\psi}_{\mathrm{lr}}$ and $\boldsymbol{\psi}_{\mathrm{bs}}$ for the learning rate and for batch size respectively, $\boldsymbol{\Psi}$, which is the Average Kendall coefficient (the average of $\boldsymbol{\psi}_{\mathrm{lr}}$ and $\boldsymbol{\psi}_{\mathrm{bs}}$) [Gastpar et al., 2023]), which are more relevant than the classical Kendall's coefficient for capturing causal relationships; and Kendall tau (τ).

Experiments ran on a NVIDIA 2080Ti GPU with 11GB RAM.

Regularization We train five neural networks each with two fully connected hidden layers on the MNIST dataset for 2000 epochs, using cross entropy loss on MNIST and a learning rate of 0.001.

Experiments ran on NVIDIA GeForce GTX 1060 6GB GPU.

Clustering For DBSCAN, we used $\epsilon = 10$ and minimum number of clusters = 2. For k-means and Agglomerative clustering, the minimum number of clusters was set to the number of cluster centers used to generate the dataset.

Experiments ran on Intel Xeon CPU E5-2603 v4 CPU with 3GB memory.

A.1.2.3 More experimental Results - Clustering

In Figure A.4, we show more results of the magnitude clustering algorithm using a number of different random seeds for dataset generation.

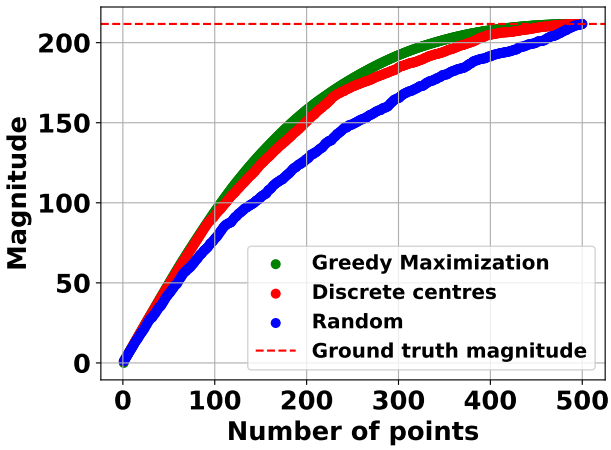


Figure A.3: Comparison of subset selection on the Swiss roll with subsample of 500 points.

A.1.2.4 Subset selection

Figure A.3 shows an example of a synthetic dataset called the Swiss roll. The Discrete centers algorithm produces a hierarchy which provides almost the same approximation as the Greedy Maximization algorithm for a fraction of the computational cost. In Figure A.5 we see a comparison between Greedy Maximization, Discrete Centers and selecting points at random (Random) for the 3 standard `scikit-learn` datasets Iris, Breast cancer and Wine dataset. We have used the entire dataset for generating the plot. In Figure A.6 we see the same comparison, but with a random subset of 500 points from MNIST, CIFAR10 and CIFAR100. We have reduced the dimensions of each dataset using PCA to 100.

A.1.2.5 Magnitude of a compact space

For completeness, we provide the formal definition of magnitude for compact sets, and a few important results.

Definition A.1.3. A metric space is positive definite if every finite subspace is positive definite. The magnitude of a compact positive definite space A is

$$\mathrm{Mag}(A) = \sup\{\mathrm{Mag}(B) : B \text{ is a finite subspace of } A\} \in [0, \infty]. \tag{i}$$

Definition A.1.4. Let a weight measure for a compact space A be a signed measure $\mu \in M(A)$ such that, for all $a \in A$,

$$\int e^{-d(a,b)} d\mu(b) = 1. \tag{ii}$$

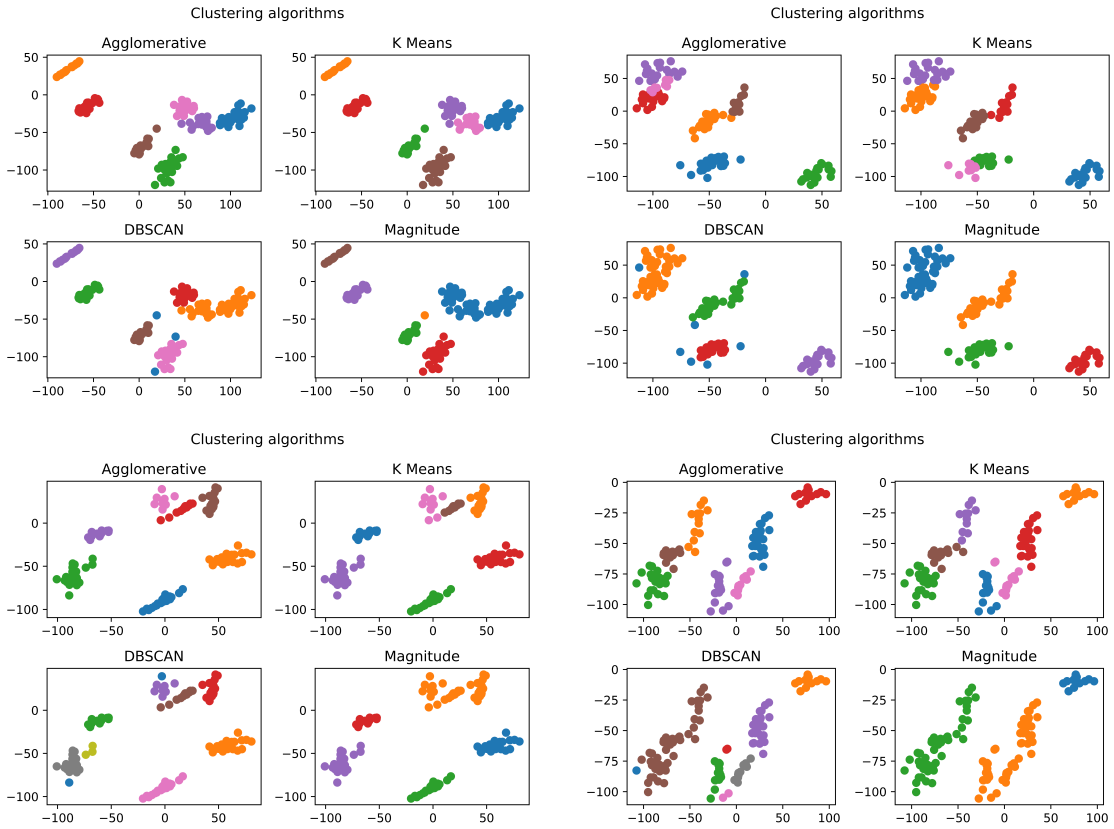
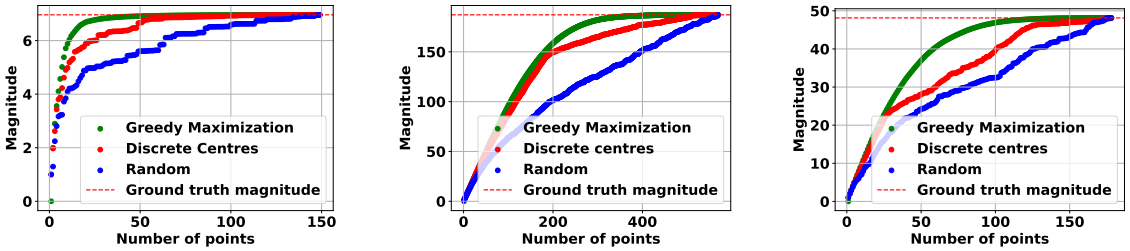


Figure A.4: Magnitude clustering algorithm versus conventional algorithms on randomly generated datasets. The difference between the plots comes from using a different random seed to generate the datasets. We note that the magnitude algorithm consistently identifies a reasonable number of clusters and provides a sensible cluster assignment to each point.



(a) (b) (c)

Figure A.5: **Discrete centers are close to Greedy Maximization at a fraction of the computational cost and better than random.** In plot (a) we have the Iris dataset, in plot (b) the Breast cancer dataset, in plot (c) the Wine dataset.

Then $\mathrm{Mag}(X) = \mu(A)$ whenever μ is a weight measure for A.

Theorem A.1.5 (Theorem 5.4 in [Meckes, 2015])**.** *Let $A \subset \mathbb{R}^n$ be compact and $t \geq 1$. Then*

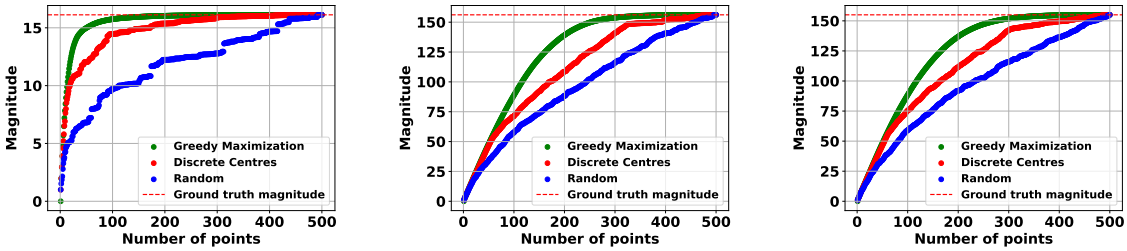
$$\frac{\mathrm{Mag}(A)}{t} \leq \mathrm{Mag}(tA) \leq t^n \mathrm{Mag}(A) \tag{iii}$$

Theorem A.1.6 (Theorem 1 in [Barceló and Carbery, 2018])**.** *Let X be a nonempty compact set in $\mathbb{R}^n$. Then*

$$\mathrm{Mag}(tX) \to 1 \text{ as } t \to 0 \tag{iv}$$

and

$$t^{-n}\mathrm{Mag}(tX) \to \frac{Vol(X)}{n!\omega_n} \text{ as } t \to \infty. \tag{v}$$



(a) (b) (c)

Figure A.6: **Discrete centers are close to Greedy Maximization at a fraction of the computational cost and better than random.** In plot (a) we a subsample from MNIST dataset, in plot (b) from CIFAR10, and in plot (c) we see a subsample of CIFAR100.

A.2 Appendix for Chapter 4

A.2.1 Intrinsic Dimensions and Equalities

The following result has been proven by Meckes [2015], which related the magnitude dimension with the Minkowski dimension of compact subsets of Euclidean space.

Theorem A.2.1 (Corollary 7.4, [Meckes, 2015])**.** *If $X \subset \mathbb{R}^n$ is compact and if* $\dim_{\text{Mag}} X$ *or* $\dim_{\text{Mink}} X$ *exist, then both exist and* $\dim_{\text{Mag}} X = \dim_{\text{Mink}} X$*.*

Using the theorem above, one can use the magnitude dimension to approximate the Minkowski dimension.

Theorem A.2.2 (Heine-Borel)**.** *Let $A \subset \mathbb{R}^n$. Then A is compact if and only if A is both closed and bounded.*

Theorem A.2.3. *[Schweinhart, 2021] Let $X \subset \mathbb{R}^n$ be a bounded set. Then* $\dim^0_{\text{PH}} X = \dim_{\text{Mink}} X$*.*

A.2.2 Assumption H1

The following technical assumption has been introduced in previous work [Birdal et al., 2021, Simsekli et al., 2020] and it is necessary for the statement of the novel bound in Theorem 4.4.2 For any $\delta > 0$, define the fixed grid on $\mathbb{R}^d$ to be

$$G = \left\{ \left(\frac{(2j_i + 1)\delta}{2\sqrt{\delta}}, \ldots, \frac{(2j_d + 1)\delta}{2\sqrt{\delta}} \right) : j_i \in \mathbb{Z}, i = 1, \ldots, d \right\}. \tag{vi}$$

The collection of centres of each ball is denoted by the set N_δ. Now, we define $N_\delta(S) = \{x \in N_\delta : B_d(x, \delta) \cap \mathcal{W}_\mathcal{S} \neq \emptyset\}$, where S is the training set, $B_d(x, \delta) \subset \mathbb{R}^d$ denotes the closed ball centered around $x \in \mathbb{R}^d$ of radius δ. Then $N_\delta(S)$, as defined, is the collection of centres of each ball that intersects $\mathcal{W}$.

H1: Let $\mathcal{Z}^\infty \coloneqq (\mathcal{Z} \times \mathcal{Z} \times \mathcal{Z} \times \ldots)$ denote the countable product endowed with the product topology and let $\mathfrak{B}$ be the Borel σ-algebra generated by $\mathcal{Z}^\infty$. Let $\mathfrak{F}, \mathfrak{G}$ be the sub-σ-algebras of $\mathfrak{B}$, generated by the collections of random variables given by $\{L_S(w) : w \in \mathcal{W}, n \geq 1\}$ and $\left\{\mathbb{1}\{w \in N_\delta : \delta \in \mathbb{Q}_{>0}, w \in G, n \geq 1\}\right\}$ respectively. There exists a constant $M \geq 1$, such that for any $A \in \mathfrak{F}$, $B \in \mathfrak{G}$, we have $\mathbb{P}[A \cap B] \leq M\mathbb{P}[A]\mathbb{P}[B]$.

This is known as the ψ-mixing condition [Bradley, 1983] and it is common in statistics for proving limit theorems. Smaller M indicates that the dependence of $L_S(w)$ on the training sample S is weaker.

A.3 Appendix for Chapter 5

We now provide additional technical details and proofs that are omitted from the chapter, followed by experimental evidence in addition to the experiments in the chapter. We organize the appendix as follows:

- Appendix A.3.1 presents additional technical background related to information theory, Rademacher complexity, and the various topological quantities that appear in our work.
- In Appendix A.3.2, we present the omitted proofs of all our theoretical results, as well as a few additional theoretical contributions.
- In Appendix A.3.3, we show the experimental details needed to reproduce our experiments.
- Finally, Appendix A.3.4 is dedicated to additional empirical results.

A.3.1 Additional Technical Background

A.3.1.1 Information-theoretic quantities

The following definition is a precise definition of the total mutual information term that appears in our main theoretical results. The reader may consult [van Erven and Harremoës, 2014, Hodgkinson et al., 2022, Dupuis et al., 2024] for further information on this notion.

Definition A.3.1 (Total mutual information)**.** Let X and Y be two random elements defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ (note that the codomains of X and Y may be distinct). We define the total mutual information between X and Y by the following formula:

$$\mathrm{I}_\infty(X, Y) = \log\left(\sup_A \frac{\mathbb{P}_{X,Y}(A)}{\mathbb{P}_X \otimes \mathbb{P}_Y(A)}\right).$$

Such a term has already been used in the fractal-based generalization literature [Hodgkinson et al., 2022, Dupuis et al., 2024]. Other works used intricate variants of this total mutual information term [Dupuis et al., 2023, Birdal et al., 2021, Camuto et al., 2021]. We stress the fact that our proposed bounds are simpler.

A.3.1.2 Rademacher complexity

Rademacher complexity [Bartlett and Mendelson, 2002, Shalev-Schwartz and Ben-David, 2014] is a central tool in learning theory. As part of our theory uses

this notion, we now provide its definition and introduce some notation.

Definition A.3.2 (Rademacher complexity on a hypothesis set)**.** Let us fix a dataset $S \in \mathcal{Z}^n$, a set $\mathcal{W} \subset \mathbb{R}^d$ and $\epsilon = (\epsilon_1, \dots, \epsilon_n)$ some iid Rademacher random variable.[1] Whenever it is defined, we will call Rademacher complexity of ℓ over $\mathcal{W}$ the following quantity:

$$\mathrm{Rad}(\ell, \mathcal{W}, S) := \frac{1}{n}\mathbb{E}_\epsilon \left[\sup_{w \in \mathcal{W}} \sum_{i=1}^{n} \epsilon_i \ell(w, z_i) \right].$$

Rademacher complexity has already been used in [Dupuis et al., 2023, Theorem 3.4] to relate the generalization error to the so-called data-dependent fractal dimension. Part of our theory is based on a recent extension of such arguments in the data-dependent setting [Dupuis et al., 2024].

A.3.1.3 Persistent homology

The goal of this short subsection is to present a few notions of persistent homology, which is necessary for a better understanding of our contributions.

Persistent homology [Edelsbrunner and Harer, 2010, Carlsson, 2014, Boissonat et al., 2018] is an important subfield of TDA, capable of providing myriad of new insights for analysing data by extracting meaningful topological features. It has demonstrated its usefulness in a very diverse set of applications from biology [Nicolau et al., 2011, Emmett et al., 2016], to materials science [Hiraoka et al., 2016], finance [Leibon et al., 2008], robotics [Bhattacharya et al., 2015], sensor networks [De Silva and Ghrist, 2007] and a lot more [Otter et al., 2017]. The types of datasets which are amenable to this kind of analysis are finite metric spaces (known as point-cloud datasets), images, networks and also level-sets of functions. More recently, several studies have brought to light empirical links between persistent homology and DNNs [Rieck et al., 2018, Corneanu et al., 2019, Pérez-Fernández et al., 2021]. In particular, recent studies have related the worst-case generalization error to several concept of intrinsic dimensions defined through persistent homology [Birdal et al., 2021, Dupuis et al., 2023]. As mentioned in the introduction, our goal is to extend these last studies to more practical settings.

In general, persistent homology is defined for any degree $k \in \mathbb{N}$ (denoted PH^k). Intuitively, PH^k keeps track of the number of "holes of dimension k" in a set when looked at different scales. However, in our work and as in [Birdal et al.,

[1] A Rademacher random variable is defined by $\mathbb{P}(\epsilon_i = 1) = \mathbb{P}(\epsilon_i = -1) = 1/2$.

2021, Dupuis et al., 2023], we only use PH^0, whose presentation is simpler. In this section, to avoid harming the readability of the chapter, we only present a high-level introduction to PH^0 that is sufficient to understand our work. The interested reader may consult [Boissonat et al., 2018, Chazal and Michel, 2021, Zomorodian, 2012] for a more in-depth introduction to persistent homology.

We first start by introducing briefly homology, which is a classical concept in algebraic topology. We only introduce the most essential concepts for understanding persistent homology. For a more detailed introduction, please consult [Hatcher, 2002].

Definition A.3.3. A simplicial complex is a set K of finite sets closed under the subset relation: if $\sigma \in K$ and $\tau \subset \sigma$, then $\tau \in K$.

In the above definition, σ is a simplex (plural simplices) and τ is a face of σ, its coface.

Definition A.3.4. An abstract simplicial complex $\mathcal{K}$ is a finite collection of simplices where a face of any simplex $\sigma \in \mathcal{K}$ is also a simplex in $\mathcal{K}$.

Definition A.3.5. A simplicial k-chain is the formal sum of k-simplices,

$$\sum_{i=1}^{N} = r_i \sigma_i, \tag{vii}$$

where each $r_i \in R$, where R is a fixed commutative ring with additive identity 0 and multiplicative identity 1, and $\sigma_i \in \mathcal{K}$.

$\mathcal{K}_k$ is the set of simplicial k-chains with addition over R, which is an R-module. Then, the set of all k-simplices of the complex $\mathcal{K}$ is a set of generators for $\mathcal{K}_k$. For each generator σ, the boundary of σ is the sum of all $(k-1)$-faces of σ.

Definition A.3.6. The *boundary* of a k-simplex $\sigma = (x_0, \ldots, x_k)$ is the $(k-1)$-chain

$$\partial_k(\sigma) = \sum_{i=0}^{k} (-1)^i (x_0, \ldots, \hat{x_i}, \ldots, x_k), \tag{viii}$$

where $(x_0, \ldots, \hat{x_i}, \ldots, x_k)$ is the $(k-1)$-simplex spanned by all vertices without x_i.

It is common that the coefficients for homology are considered to be restricted to $\mathbb{Z}_2$, which is the field with 2 elements, 0 and 1, where $1 + 1 = 0$. However, the theory extends to homoogy with coefficeints in any field (and since every field is a ring, the definitions in terms of rings are more general).

Definition A.3.7. A chain complex is a sequence of abelian groups A_k with homomorphisms (called boundary maps) $\partial_k : A_k \to A_{k-1}$, such that $\partial_{k-1} \circ \partial_k = 0$ for all k.

We should note that when considering coefficients in $\mathbb{Z}_2$, a k-chain can be seen as a finite collection of k-simplices.

Introduce topological invariants: simplicial homology groups and Betti numbers.

Definition A.3.8 (Simplicial Homology group)**.** The n-th (simplicial) homology group of a finite simplicial complex $\mathcal{K}$ is

$$H_n = \ker \partial_n / \mathrm{im} \partial_{n+1}, \tag{ix}$$

where ker and im are the kernel and image respectively of the boundary operator.

In order to define the simplicial complexes of use in TDA, we need to first understand what a nerve is.

Definition A.3.9 (Nerve)**.** A simplicial complex associated to a collection of sets is called a nerve. The sets are the vertices of the complex, and a simplex belongs to a complex iff its vertices have a non-empty intersection, $\mathrm{Nrv} = \{\alpha \subseteq S \mid \cap_{A \in \alpha} A \neq \emptyset\}$.

Definition A.3.10 (Čech complex)**.** The Čech complex of X for radius r is $\check{\mathrm{C}}\mathrm{ech}_r(X) = \mathrm{Nrv}\{B(x, r) \mid x \in X\}$, where $B(x, r)$ is the closed ball of radius $r \geq 0$, centered at x.

In other words, the Čech complex is the nerve of the ball neighbourhoods of a set of points $X \subseteq \mathbb{R}^n$. The Čech complex faithfully captures the topology of the space, but it is not computed in practice due to its high computational cost. Instead, a different complex called *Vietoris-Rips* (VR) is used due to ease of construction for higher dimensions. It can be shown that the VR complex is not always homotopy equivalent to the Čech complex, and therefore it can be seen as an approximation.

We first need to introduce the notion of a clique complex to explain what the VR is.

Definition A.3.11 (Clique complex)**.** The *clique complex* for a graph $G = (V, E)$ consists of all cliques of G, which are all simplices $\alpha \subseteq V$ for which E contains all edges of α.

Now we have explicitly states all the necessary components in order to define the main complex used in TDA, the *Vietoris-Rips complex*.

Definition A.3.12 (Vietoris-Rips complex)**.** The *Vietoris-Rips complex* of X for radius r is the clique complex of the 1-skeleton of the Čech complex of X and r, $\text{Rips}_r(X) = \{\alpha \in X \mid ||u - v|| \leq 2r\}$ for all $u, v \in \alpha$.

Now that we have defined the most important complex in TDA, we proceed to explain how we can derive important topological information at multiple scales by introducing the concept of a filtration.

Definition A.3.13. Given a simplicial complex $\mathcal{K}$, a filtration is a totally ordered set of subcomplexes $\mathcal{K}^i$ of $\mathcal{K}$, indexed by nonnegative integers, such that for $i \leq j$, $\mathcal{K}^i \subseteq \mathcal{K}^j$.

Definition A.3.14 (Filtered simplicial complex)**.** A simplicial complex, $\mathcal{K}$, together with a filtration (function $f : \mathcal{K} \to \mathbb{R}$ such that $f(\sigma) \leq f(\tau)$ whenever σ is a face of τ). The sublevel set at a value $r \in \mathbb{R}$ is $f^{-1}(-\infty, r]$, which is a subxomplex of $\mathcal{K}$. Let $r_0 < r_1 < \cdots < r_m$ be the values of the simplices, and $\mathcal{K}_i = f^{-1}(-\infty, r_i]$, then we call $\mathcal{K}_0 \subseteq \mathcal{K}_1 \subseteq \cdots \subseteq \mathcal{K}_m$ the *sublevel set filtration* of f.

When you start with a simplicial complex $\mathcal{K}$ and you filter it according to a filtration f, it is clear that the homology of $\mathcal{K}_r$ evolves as the radius r increases. For example, new connected components can be formed, loops can appear or disapper, cavities can form. What persistent homology does, and where the importance of the filtering comes in is that now we have the tools to track the topological changes associated with the different stages of the filtering process, and to associate a lifetime to them (track when a topological feature has first appeared and at which stage of the filtration it will disappear). This essential topological information is recorded in a set of intervals known as barcodes, which can be represented as a multiset of points in $\mathbb{R}^2$, where the coordinates correspond to the birth and death points of each interval.

A.3.1.4 Minimum spanning tree

The persistent homology dimension used in existing generalization bounds [Birdal et al., 2021, Dupuis et al., 2023] is closely related to another notion of intrinsic dimension, called minimum spanning tree (MST) dimension [Kozma et al., 2006], in the sense that the PH and MST dimensions of bounded metric spaces are identical. The link between persistent homology and MST is even deeper than the equality between the induced dimensions, as noted by [Schweinhart, 2020]. In this section, we define quantities related to MSTs which will play an important role in our proofs.

In this section let us fix a finite metric space (X, ρ). Let us first specify our notations for trees. A tree $\mathcal{T}$ on X is a connected undirected graph. We represent $\mathcal{T}$ by its set of edges, which are denoted $a \to b$ (or equivalently $b \to a$ as the graph is undirected). For an edge e of the form $a \to b$, we define its length by $|e| = \rho(a, b)$.

Definition A.3.15 (Minimum spanning tree)**.** Let us define the cost of a tree by the sum of the length of its edges, *i.e.*,

$$\boldsymbol{E}_1^{\mathrm{MST}}(\mathcal{T}) := \sum_{e \in \mathcal{T}} |e|.$$

An MST of X is defined as a tree with minimal cost. A consequence of the greedy algorithm to find such an MST [Cormen et al., 2001] is that an MST $\mathcal{T}$ is also minimal for any of the following costs:

$$\boldsymbol{E}_\alpha^{\mathrm{MST}}(\mathcal{T}) := \sum_{e \in \mathcal{T}} |e|^\alpha,$$

with $\alpha \geq 0$.

Our interest in this notion comes from several results that are summed up in the following theorem. The reader can refer to [Adams et al., 2020, Schweinhart, 2020, Boissonat et al., 2018] for more details.

Theorem A.3.16 (Link between MST and persistent homology)**.** *There is a bijection between the two following multisets:*

- *The multiset of the lifetimes in the persistent homology of degree* 0 *of the Vietoris-Rips complex of* X*.*
- *The multiset of the length of the edges of an MST of* X*.*

Therefore, if we fix some $\alpha \geq 0$, the weighted α-sum associated to the persistent homology of degree 0 of the Vietoris-Rips complex of X is equal to the cost $\boldsymbol{E}_\alpha$ of an MST of X, ie:

$$\boldsymbol{E}_\alpha^{\mathrm{MST}}(\mathcal{T}) = \boldsymbol{E}_\alpha(X).$$

In all the following, we will use the notation $\boldsymbol{E}_\alpha$ to denote both quantities.

A.3.1.5 Magnitude

Let us restate formally a few standard definitions. The reader may refer to [Leinster, 2013, Meckes, 2013, 2015] for more details on the notions of magnitude, weighting, and positive definite metric spaces. In this section, we fix a finite *metric* space (X, ρ). Some of the presented concepts will be later extended to pseudometric spaces in A.3.2.2.

As before, the *similarity matrix* [Leinster, 2013] of X is defined by $M(a, b) = e^{-\rho(a,b)}$, for $a, b \in X$. We now define weightings and magnitude of X, according to [Leinster, 2013, Section 2.1].

Definition A.3.17 (Weighting and magnitude)**.** A weighting of X is a function $\beta : X \longrightarrow \mathbb{R}$ such that

$$\forall a \in X, \ \sum_{b \in X} e^{-\rho(a,b)} \beta(b) = 1.$$

If such a weighting exists, the magnitude of X is defined by:

$$\mathrm{Mag}(X) := \sum_{b \in X} \beta(b).$$

It is easily seen that this definition is independent of the choice of weighting β. When a weighting exists, we say that X "has magnitude".

Based on such a definition, it is natural to inquire, whether such a weighting exists. This question has been studied by several authors [Leinster, 2013, Meckes, 2013, 2015]. This question appears to be related to the notion of positive definite space, which we now define, according to [Leinster, 2013].

Definition A.3.18 (Positive definite space)**.** X is positive definite if the similarity matrix M is positive definite.

It is clear that positive definite spaces have magnitude. More interestingly, we have the following result, which ensures that most metric spaces considered in this study are positive definite.

Theorem A.3.19 ([Leinster, 2013, Meckes, 2013])**.** *Let $p \in [1, 2]$ and $d \geq 1$, every finite subset of $(\mathbb{R}^d, \|\cdot\|_p)$ is positive definite.*

A.3.1.6 Covering and packing numbers

In this section, we fix a compact pseudometric space (X, ρ) and give definitions of covering and packing numbers. These quantities have long been of primary interest in learning theory, in particular through the classical covering arguments for Rademacher complexity [Shalev-Schwartz and Ben-David, 2014, Rebeschini, 2020]. More recently, limits of covering arguments have been leveraged by several authors to derive uniform generalization bounds in terms of fractal dimensions [Simsekli et al., 2020, Hodgkinson et al., 2022, Camuto et al., 2021, Dupuis et al., 2023, 2024], which we aim to improve in this study.

For $x \in X$ and $r > 0$, we denote the closed ball centered at x and or radius r by $\bar{B}_r(x) := \{y \in X \ , \rho(x, y) \leq r\}$. We can now define covering and packing.

Definition A.3.20 (Covering number)**.** Let $\delta > 0$, the covering number $N_\delta^\rho(X)$ is the cardinality of a minimal set of points N such that:

$$X \subseteq \bigcup_{x \in N} \bar{B}_\delta(x).$$

Remark A.3.21. There exist several conventions for the definition of such numbers [Falconer, 2014, Mattila, 1999b, Vershynin, 2020], all of which are equivalent up to absolute constants and in particular induce the same fractal dimensions on X (see [Falconer, 2014]).

Definition A.3.22 (Packing number)**.** Let $\delta > 0$, the covering number $N_\delta^\rho(X)$ is the cardinality of a maximal set of disjoint closed balls with centers in X.

A.3.1.7 About Johnson-Lindenstrauss lemma

In our implementation of Euclidean-based topological quantities, we use sparse random projections to project the weight vectors from $\mathbb{R}^d$ to a lower dimensional subspace. This is necessary because of memory constraints. Indeed, storing the

full trajectory $\mathcal{W}_{\tau \to T} \subset \mathbb{R}^d$ (in our experiments $T - \tau = 5 \times 10^3$) can become intractable for large models.

Given a finite set of points $\mathcal{W} \subset \mathbb{R}^d$ and $\epsilon > 0$. Let $N \geq \mathcal{O}\left(\frac{\log |\mathcal{W}|}{\epsilon^2}\right)$, Johnson-Lindenstrauss lemma [Vershynin, 2020, Fernandez-Granda, 2016] ensures the existence of a linear map $P : \mathbb{R}^d \longrightarrow \mathbb{R}^N$ such that:

$$\forall w, w' \in \mathcal{W},\ (1-\epsilon) \left\|w - w'\right\|^2 \leq \left\|Pw - Pw'\right\|^2 \leq (1+\epsilon) \left\|w - w'\right\|^2 .$$

In practice, the linear maps suggested by this result can be obtained through subgaussian random projections [Vershynin, 2020, Section 9.3].

In our work, as the purpose of Johnson-Lindenstrauss embeddings is mainly memory optimization, we have to rely on sparse random projections. We use the implementation provided in `scikit-learn` [Pedregosa et al., 2011a]. More precisely, we used a relative variation ϵ of 5%.

Finally, it should be noted that these projection techniques were only used for the vision transformer experiments, as the GNNs that we used have a small enough number of parameters to avoid the use of random projections.

A.3.1.8 A note on the connection to Topological Deep Learning

Topological deep learning (TDL) is a rapidly evolving field that uses topological features to understand and design deep learning models [Papamarkou et al., 2024, Hajij et al., 2022]. Our topological complexity measures can be seen as a direction towards addressing the Open Problem 7 mentioned in [Papamarkou et al., 2024] concerning the discovery of topological properties of internal representations that are linked to generalization.

A.3.2 Omitted Proofs of the Theoretical Results

In this section, we present the proofs of our main theoretical contributions. We divide our proofs into two groups of subsections:

- Sections A.3.2.1, A.3.2.2 and A.3.2.3 focus on the extension (in a very natural way) of the quantities appearing in our bounds in pseudometric spaces. The main outcome of this analysis is the definition of positive magnitude in the pseudometric case. Note that A.3.2.1 is not a contribution of this chapter, but we include it in this section to improve the readability of the chapter.

- In sections A.3.2.4, A.3.2.5, A.3.2.6 and A.3.2.7, we present the proof of our main theoretical results.

Before proving our main results, we define the notion of *metric identification*, which will be used in several of the following subsections. This is the same setting that was used in [Dupuis et al., 2023] to naturally extend the persistent homology dimension to pseudometric spaces.

Definition A.3.23 (Metric identification)**.** Let (X, ρ) be a pseudometric space. We can define an equivalence relation on X by $a \sim b \iff \rho(a, b) = 0$. The associated quotient space, which is denoted ${}^{X}/_{\sim}$ is a metric space for the naturally induced metric, which we still denote ρ.[2] We will also use the canonical projection,

$$\pi : X \longrightarrow {}^{X}/_{\sim}.$$

These notations will be used throughout the text.

A.3.2.1 Persistent homology and MST in pseudometric spaces

In this short subsection, we first restate results proven in [Dupuis et al., 2023], regarding persistent homology in pseudometric spaces. The main result is the following proposition, which has been proven inside the proof of [Dupuis et al., 2023, Lemma B.9].

Proposition A.3.24 ([Dupuis et al., 2024])**.** *Let (X, ρ) be a finite pseudometric space and $\alpha \geq 0$, then we have:*

$$\boldsymbol{E}_\alpha(X) = \boldsymbol{E}_\alpha\left({}^{X}/_{\sim}\right)$$

where the pseudometric ρ (and its metric identification) have been omitted from the notation.

Based on Theorem A.3.16, the above result is also true when $\boldsymbol{E}_\alpha$ represents the cost of a MST of X.

A.3.2.2 Magnitude in pseudometric spaces

In this section, we fix (X, ρ) a finite pseudometric space. We denote by ${}^{X}/_{\sim}$ its metric identification and by $\pi : X \longrightarrow {}^{X}/_{\sim}$ the canonical projection.

[2] Indeed, if $a \sim b$, then we have $\forall c \in X,\ \rho(a, c) = \rho(b, c)$.

We directly extend Definition A.3.17 to the pseudometric case. In order for this definition to make sense in our context, we first need to verify that it provides a well-posed definition of magnitude. This follows from the following lemma.

Lemma A.3.25. *We assume that the finite pseudometric space (X, ρ) has magnitude. Then magnitude is independent of the choice of weighting.*

Proof. The proof is straightforward and identical to the metric case. Let β, β' be two weightings, we have:

$$\sum_{a\in X}\beta(a) = \sum_{a\in X}\sum_{b\in X} e^{-\rho(a,b)}\beta'(b)\beta(a) = \sum_{b\in X}\beta'(b)\sum_{a\in X} e^{-\rho(a,b)}\beta(a) = \sum_{b\in X}\beta'(b).$$

□

In the following theorem, we show that magnitude is invariant through metric identification.

Theorem A.3.26 (Invariance of magnitude through metric identification)**.** *X has magnitude if and only if ${}^{X}/_{\sim}$ has magnitude, in which case we have:*

$$\mathrm{Mag}(X) = \mathrm{Mag}\left({}^{X}/_{\sim}\right).$$

Proof. We decompose X into equivalence classes as:

$$X = \coprod_{\bar{a}\in {}^{X}/_{\sim}} \bar{a} =: \coprod_{i\in I} \bar{a}_i,$$

where $\coprod$ denotes disjoint union and the points $(a_i)_{i\in I} \in X^I$ represent each equivalence class. We denote by $\bar{a}$ the equivalence class of $a \in X$.

Let $\beta : X \longrightarrow \mathbb{R}$ be any function. We have:

$$\forall a \in X,\ \sum_{b\in X} e^{-\rho(a,b)}\beta(b) = \sum_{i\in I} e^{-\rho(\bar{a},\bar{a}_i)}\sum_{b\in\bar{a}_i}\beta(b). \tag{x}$$

$\Longrightarrow$: If X has magnitude, then we take β to be a weighting of X, we define:

$$\forall \bar{a} \in {}^{X}/_{\sim},\ \bar{\beta}(\bar{a}) := \sum_{b\in\bar{a}}\beta(b).$$

By Equation (x), $\bar{\beta}$ is a weighting of ${}^{X}/_{\sim}$.

$\Longleftarrow$: if $\bar{\beta}$ is a weighting of ${}^{X}/_{\sim}$, then we define:

$$\forall a \in X,\ \beta(a) := \frac{1}{|\bar{a}|}\bar{\beta}(\bar{a}),$$

where $|\bar{a}|$ denotes the cardinality of $\bar{a}$. By Equation (x), β is a weighting of X. □

A.3.2.3 Definition of positive magnitude in the pseudometric case

Let us extend our new notion of *positive magnitude* in finite pseudometric spaces. This is a rather complicated task. Indeed we need to ensure that the positive magnitude is independent of the choice of weighting, which is not true in general. For this reason, we restrict our definition to pseudometric spaces whose metric identification is positive definite and we choose one particular weighting.

Definition A.3.27 (Positive magnitude in finite pseudometric spaces)**.** Let (X, ρ) be a finite pseudometric space whose metric identification ${}^{X}/_{\sim}$ is positive definite. Let $\bar{\beta} : {}^{X}/_{\sim} \longrightarrow \mathbb{R}$ be a weighting of ${}^{X}/_{\sim}$, then we define the positive magnitude of X, denoted **PMag**, by:

$$\mathbf{PMag}(X) = \sum_{\bar{x} \in {}^{X}/_{\sim}} \bar{\beta}(\bar{x})_+,$$

where $x_+ := \max(x, 0)$ denotes the positive part of x. We will say that X admits a positive magnitude if its metric identification ${}^{X}/_{\sim}$ is positive definite.

Note that ${}^{X}/_{\sim}$ admits a unique weighting because it is positive definite. However, X still admits several weightings in general. The above definition ensures that the definition of positive magnitude is independent of any choice of weighting. For the need of our proofs, we will need to introduce weightings in pseudometric spaces, whose sums of positive parts yield the positive magnitude. This is possible by using the following definition, which corresponds to a "good" choice of weighting in finite pseudometric spaces.

Definition A.3.28 (Canonical weighting)**.** Let (X, ρ) be a finite pseudometric space whose metric identification ${}^{X}/_{\sim}$ is positive definite. Let $\bar{\beta} : {}^{X}/_{\sim} \longrightarrow \mathbb{R}$ be a weighting of ${}^{X}/_{\sim}$, we define the *canonical weighting* $\beta^0 : X \longrightarrow \mathbb{R}$ on X by:

$$\forall a \in X,\ \beta^0(a) := \frac{1}{|\pi(a)|}\bar{\beta}(\pi(a)),$$

where $\pi : X \longrightarrow {}^{X}/_{\sim}$ is the canonical surjection.

The following lemma is then obvious but crucial to some of our theoretical results.

Lemma A.3.29. *With the notation of the previous definition, we have:*

$$\mathbf{PMag}(X) = \sum_{x \in X} \beta^0(x)_+.$$

The next proposition is a consequence of Theorem A.3.19, it shows that the pseudometrics considered in practice in our work (and in our experiments) admit a positive magnitude.

Proposition A.3.30. *Let $p \in [1,2]$ and $S \in \mathcal{Z}^n$, then every finite subset of $(\mathbb{R}^d, \rho_S^{(p)})$ admits a positive magnitude, and therefore it also has a canonical weighting.*

Proof. Let $\mathcal{W} := \{w_1, \dots, w_N\}$ be a finite set in $\mathbb{R}^d$. We have

$$\|\boldsymbol{L}_S(w) - \boldsymbol{L}_S(w')\|_p = n^{1/p} \rho_S^{(p)}(w, w').$$

Therefore, if we denote by $\bar{w}$ the equivalence class of w in the metric identification, it is clear that $\bar{w} = \bar{w'} \iff \boldsymbol{L}_S(w) = \boldsymbol{L}_S(w')$. Hence, the map $\varphi_S := n^{-1/p}\boldsymbol{L}_S$ naturally extends to an isometry between metric spaces:

$$\mathcal{W}/_{\sim} \xrightarrow{\sim} \varphi_S(\mathcal{W}) \underset{\text{finite}}{\subset} \mathbb{R}^n.$$

By Theorem A.3.19, the finite set $\varphi_S(\mathcal{W})$ is positive definite, hence it is also the case of $\mathcal{W}/_{\sim}$. Therefore $\mathcal{W}$ admits a positive magnitude by definition. □

A.3.2.4 Warm-up: covering bounds

Assumptions. Given an (q, L, ρ)-Lipschitz continuous (pseudo-)metric, our approach relies only on a single assumption of a bounded loss function. For the case of the pseudometric ρ_S (Example 5.3.2), this assumption is already made in [Dupuis et al., 2023, 2024].

Assumption A.3.31. We assume that the loss ℓ is bounded in $[0, B]$, with $B > 0$ a constant.

The boundedness of ℓ is classically assumed in the fractal / TDA literature [Dupuis et al., 2023, Hodgkinson et al., 2022, Dupuis et al., 2024]. In [Dupuis et al., 2023], it is shown that the proposed theory seems to be experimentally valid even for unbounded losses. Our experimental findings suggest that this observation also applies to our work.

The following is deduced from the transcription of the results of [Dupuis et al., 2024] to our setting. It is the starting point of our persistent homology-based analysis.

Theorem A.3.32. *Let ρ be a pseudometric on $\mathbb{R}^d$. Suppose that Assumption A.3.31 holds and that ℓ is (q, L, ρ)-Lipschitz, for $q \geq 1$. Then, for all $\delta > 0$, with probability at least $1 - \zeta$ over $\mu_z^{\otimes n} \otimes \mu_u^{\otimes \infty}$,*

$$\sup_{\tau \leq i \leq T} G_S(w_i) \leq 2L\delta + 2B\sqrt{\frac{2\log N_\delta^\rho(\mathcal{W}_{\tau\to T})}{n}} + 3B\sqrt{\frac{\mathrm{I}_\infty(S, \mathcal{W}_{\tau\to T}) + \log(1/\zeta)}{2n}}.$$

The proof of this theorem will be given in the next subsection. Before discussing this proof, a few remarks are in order.

Covering bounds, such as A.3.32 have been used in [Simsekli et al., 2020, Camuto et al., 2021, Birdal et al., 2021, Dupuis et al., 2023] to introduce fractal dimensions (more precisely through the notion of upper box-counting dimension) into the generalization bounds. This is done via the following definition of the aforementioned upper box-counting dimension:

$$\overline{\dim}_{\mathrm{B}}^{\rho}(X) := \limsup_{\delta \to 0} \frac{\log N_\delta^\rho(X)}{\log(1/\delta)}.$$

By using a similar procedure, we see that our framework could be used to introduce intrinsic dimensions associated to a wide range of pseudometrics, as soon as they satisfy a (q, L, ρ)-Lipschitz continuity assumption.

However, arguments based on these intrinsic dimensions only make sense in the limit $T \to \infty$, which makes little sense in practical settings. To address this issue, we take inspiration from two other notions that are equal to the upper box-counting dimension (and therefore lay the ground of the numerical approximation of this dimension), namely the PH-dimension [Kozma et al., 2006, Schweinhart, 2020, Birdal et al., 2021, Dupuis et al., 2023] and the magnitude dimension [Meckes, 2013] and Chapter 4. Our approach is to replace the intrinsic dimensions by the "intermediary quantities" used to define them. This leads to the results presented in the next two subsection.

A.3.2.5 Proof of Theorem A.3.32

Before going to the proof of Theorem A.3.32, we specify our theoretical setup, which is the one introduced in [Dupuis et al., 2024]. In this section, we prove our results in the case $T < +\infty$. However, note that one could consider $T = +\infty$ without much technical difficulties.

The setup is the following: let $(F(\mathbb{R}^d), \mathcal{T})$ denote the set of all finite subsets of $\mathbb{R}^d$, endowed with a σ-algebra $\mathcal{T}$.

We consider the following probability distribution on $F(\mathbb{R}^d)$:

$$\forall A \in \mathcal{T},\ \pi(A) := \int_{\mathcal{Z}^n} p_S(A) d\mu_z^{\otimes n}(S). \tag{xi}$$

Here we let the posterior p_S be the conditional distribution of $\mathcal{W}_S$ given S.

As it is discussed in [Dupuis et al., 2024, Section 5.4], we make the following technical measure-theoretic assumption.

Assumption A.3.33. The probability measure $\mu_z^{\otimes n}$ is a strictly positive Borel measure. Moreover, for every $A \in \mathcal{T}$, the map $S \mapsto p_S(A)$ is continuous.

The following example highlights the fact this is a very mild assumption.

Example A.3.34. If the data space $\mathcal{Z}$ is countable and the data distribution μ_z has no null mass, then the above assumption is automatically satisfied with respect to the discrete topology.

Here we introduce Lemma 16 and Theorem 10 from Dupuis et al. [2024], which will be important for the proof of the next theorem.

Lemma A.3.35 (Lemma 16 in Dupuis et al. [2024])**.** *With the same notations, we have, for $\mu_z^{\otimes n}$-almost all S and p_S-almost all $\mathcal{W}$:*

$$\log \frac{dp_S}{d\pi}(\mathcal{W}) \leq I_\infty(\mathcal{W}_S, S).$$

Theorem A.3.36 (Theorem 10 in Dupuis et al. [2024], Data-dependent Rademacher complexity bound)**.** *Suppose that Assumptions 1 and 2 hold. Then, for any $\lambda > 0$ we have*

$$P_S\left(\mathbb{E}_{\mathcal{W}\sim p_S}\left[G_S(\mathcal{W})\right] \leq \mathbb{E}_{\mathcal{W}\sim p_S}\left[2Rad_S(\mathcal{W})\right] + \frac{KL(p_S\|\pi) + \log(1/\zeta)}{\lambda} + \lambda\frac{9B^2}{8n}\right) \geq 1-\zeta,$$

$$P_{S,\mathcal{W}\sim p_S}\left(G_S(\mathcal{W}) \leq 2Rad_S(\mathcal{W}) + \log\frac{dp_S}{d\pi}(\mathcal{W}) + \frac{\log(1/\zeta)}{\lambda} + \lambda\frac{9B^2}{8n}\right) \geq 1-\zeta.$$

We can now state and prove the following theorem:

Theorem A.3.32. *Let ρ be a pseudometric on $\mathbb{R}^d$. Suppose that Assumption A.3.31 holds and that ℓ is (q, L, ρ)-Lipschitz, for $q \geq 1$. Then, for all $\delta > 0$, with probability at least $1-\zeta$ over $\mu_z^{\otimes n} \otimes \mu_u^{\otimes\infty}$,*

$$\sup_{\tau\leq i\leq T} G_S(w_i) \leq 2L\delta + 2B\sqrt{\frac{2\log N_\delta^\rho(\mathcal{W}_{\tau\to T})}{n}} + 3B\sqrt{\frac{\mathrm{I}_\infty(S, \mathcal{W}_{\tau\to T}) + \log(1/\zeta)}{2n}}.$$

Proof. Let us fix some $\zeta \in (0,1)$. First note that thanks to Assumption A.3.33, we have that p_S is absolutely continuous with respect to π, $\mu_z^{\otimes n}$-almost surely. Therefore, we can introduce its Radon-Nykodym derivative, denoted by $dp_S/d\pi$.

Thanks to the above notation, we can apply the data-dependent Rademacher complexity bound from A.3.36 to obtain that with probability at least $1-\zeta$, we have, for any $\lambda > 0$:

$$\sup_{\tau \leq i \leq T} \left(\mathcal{R}(w_i) - \widehat{\mathcal{R}}_S(w_i) \right) \leq 2\mathrm{Rad}(\ell, \mathcal{W}_{\tau \to T}, S) + \frac{1}{\lambda} \left(\frac{dp_S}{d\pi}(\mathcal{W}_{\tau \to T}) + \log(1/\zeta) \right) + \lambda \frac{9B^2}{8n},$$

with $\mathrm{Rad}(\ell, \mathcal{W}_{\tau \to T}, S)$ a Rademacher complexity term, defined by:

$$\mathrm{Rad}(\ell, \mathcal{W}_{\tau \to T}, S) := \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{w \in \mathcal{W}_{\tau \to T}} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \ell(w, z_i) \right],$$

where $\boldsymbol{\epsilon} := (\epsilon_1, \ldots, \epsilon_n)$ is a vector of independent centered Bernoulli random variables.

By A.3.35, we have almost surely that:

$$\frac{dp_S}{d\pi}(\mathcal{W}_{\tau \to T}) \leq \mathrm{I}_\infty(\mathcal{W}_{\tau \to T}, S).$$

Therefore, by optimizing the choice of the parameter λ in the above equation, we have that:

$$\sup_{\tau \leq i \leq T} \left(\mathcal{R}(w_i) - \widehat{\mathcal{R}}_S(w_i) \right) \leq 2\mathrm{Rad}(\ell, \mathcal{W}_{\tau \to T}, S) + 3B \sqrt{\frac{\mathrm{I}_\infty(S, \mathcal{W}_{\tau \to T}) + \log(1/\zeta)}{2n}}. \tag{xii}$$

We now perform a covering argument very similar to classical covering arguments for Rademacher complexity[Shalev-Schwartz and Ben-David, 2014]. Let us fix some $\delta > 0$ and introduce $(x_1, \ldots, x_{N_\delta^\rho(\mathcal{W}_{\tau \to T})})$ the centers of a minimal δ-covering of $\mathcal{W}_{\tau \to T}$ for pseudometric ρ. For any $w \in \mathcal{W}_{\tau \to T}$, there exists j such that $\rho(w, x_j) \leq \delta$. Therefore we have:

$$\begin{aligned}
\sup_{w \in \mathcal{W}_{\tau \to T}} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \ell(w, z_i) &\leq \sup_{1 \leq j \leq N_\delta^\rho(\mathcal{W}_{\tau \to T})} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \ell(x_j, z_i) + \frac{1}{n} \sum_{i=1}^{n} \epsilon_i (\ell(w, z_i) - \ell(x_j, z_i)) \\
&\leq \sup_{1 \leq j \leq N_\delta^\rho(\mathcal{W}_{\tau \to T})} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \ell(x_j, z_i) + \frac{1}{n} \sum_{i=1}^{n} |\ell(w, z_i) - \ell(x_j, z_i)| \\
&\leq \sup_{1 \leq j \leq N_\delta^\rho(\mathcal{W}_{\tau \to T})} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \ell(x_j, z_i) + n^{-1/q} \left\| \boldsymbol{L}_S(w) - \boldsymbol{L}_S(x_j) \right\|_q,
\end{aligned}$$

where the last line comes from Hölder's inequality.

We can now apply Massart's lemma on the first term and the (q, L, ρ)-Lipschitz continuity of ℓ on the second term, this gives us:

$$\mathrm{Rad}(\ell, \mathcal{W}_{\tau\to T}, S) \leq L\delta + B\sqrt{\frac{2\log N_\delta^\rho(\mathcal{W}_{\tau\to T})}{n}},$$

which concludes the proof.

□

A.3.2.6 Persistent homology bounds

We now present the proofs of our persistent homology-based bounds, ie, the results of 5.3.2.

The following lemma is a pseudometric version of a classical result of fractal geometry [Falconer, 2014].

Lemma A.3.37 (Covering and packing in pseudometric spaces)**.** *Let (X, ρ) be a pseudometric space, $\delta > 0$, and $\left\{x_1, \ldots, x_{P_\delta(X)}\right\}$ a maximal δ-packing of X for pseudometric ρ. Then we have:*

$$N_{2\delta}^\rho(X) \leq P_\delta^\rho(X).$$

Proof. Let us fix $\delta > 0$ and let $(x_1, \ldots, x_{P_\delta^\rho(X)})$ be centers of a maximal packing of X with closed δ-balls. Let us assume that:

$$X \setminus \bigcup_{1\leq i\leq P_\delta^\rho(X)} \bar{B}_{2\delta}(x_i) \neq \emptyset,$$

so that we can take some x_0 belonging to the above non-empty set. Now let us fix $i \in \{1, \ldots, P_\delta^\rho(X)\}$ and $w \in \bar{B}_\delta(x_i)$. By the triangle inequality and the definition of w and x_0, we have:

$$\underbrace{\rho(x_0, x_i)}_{>2\delta} \leq \rho(x_0, w) + \underbrace{\rho(w, x_i)}_{\leq\delta}.$$

Therefore, we have $\rho(x_0, w) > \delta$, and hence $\bar{B}_\delta(x_i) \cap \bar{B}_\delta(x_0)$, so that we construct a bigger δ-packing, by adding x_0 to $(x_1, \ldots, x_{P_\delta^\rho(X)})$, which is absurd.

Therefore, we have $X \setminus \bigcup_{1\leq i\leq P_\delta^\rho(X)} \bar{B}_{2\delta}(x_i) = \emptyset$, hence the result. □

The next lemma asserts that $\boldsymbol{E}_\alpha$ is increasing (with respect to the inclusion of sets), if and only if $\alpha \leq 1$. This is the reason why we require $\alpha \in [0, 1]$ in Theorem 5.3.5.

Lemma A.3.38. *Let (X, ρ) be a finite pseudometric space, $\alpha \in [0,1]$ and $\delta > 0$. Then we have:*

$$\boldsymbol{E}_\alpha^\rho(X) \geq \frac{1}{2} P_\delta^\rho(X) \delta^\alpha$$

Proof. In all the following, we fix $\alpha \in [0,1]$ and $\delta > 0$. We also denote $P := P_\delta^\rho(X)$. Without loss of generality, we can assume $P \geq 2$.

We fix $\mathcal{T}$ an MST of X, represented by a set of edges denoted $x \to y$, with $x, y \in X^2$ (note that we identify $x \to y$ and $y \to x$). It is a classical result that there are $|X| - 1$ edges. For an edge e of the form $a \to b$, we denote its length by $|e| := \rho(a, b)$.

For $a, b \in X$, with $a \neq b$, we denote by $\{a \to b\}$ the shortest path between a and b. More precisely, we represent it as a list of edges, denoted $a = a_0 \to a_1 \cdots \to a_K = b$, for some K. When the context is clear, we identify $\{a \to b\}$ to the set of its edges $a \to b$.

Let us introduce $(x_1, \ldots, x_P)$ a maximal δ-packing of X by closed.

For every $i \in \{1, \ldots, P\}$, as $\mathcal{T}$ is connected, there exists $y_i \in X$ such that $y_i \notin \bar{B}_\delta(x_i)$ and y_i is the only point in the path $\{x_i \to y_i\}$ that does not belong to the ball $\bar{B}_\delta(x_i)$.

For each i, we denote e_i the only edge in $\{x_i \to y_i\}$ to which y_i belongs, *i.e.* e_i is of the form $z_i \to y_i$, with $z_i \in \bar{B}_\delta(x_i)$. By construction, those edges e_i are the only ones that can be shared by several paths $\{x_i \to y_i\}$.

Let us introduce the following set of indices:

$$I := \{i \in \{1, \ldots, P\}\,,\ \forall j \neq i,\ e_i \notin \{x_j \to y_j\}\}\,, \quad K := \{1, \ldots, P\} \setminus I.$$

Let us consider $i \in K$. Let us assume that we have $j, j' \in \{1, \ldots, P\}$ such that $e_i \in \{x_j \to y_j\}$ and $e_i \in \{x_{j'} \to y_{j'}\}$. If we denote e_i as $z_i \to y_i$, we have that $z_i \in \bar{B}_\delta(x_i)$, by definition of y_i. Therefore, by definition of y_j, we have $z_i = y_j$ (because $\bar{B}_\delta(x_i) \cap \bar{B}_\delta(x_j) = \emptyset$). We have similarly $z_i = y_{j'}$ and thus $y_j = y_{j'}$. By definition of y_j and $y_{j'}$ we also have $y_i \in \bar{B}_\delta(x_j) \cap \bar{B}_\delta(x_{j'})$, which is absurd, by definition of packing. We conclude the following:

$$\forall k \in K,\ \exists! j \neq i,\ e_i \in \{x_j \to y_j\}\,.$$

For $k \in K$, we denote the corresponding j by $\varphi(k)$.

By definition of K, it is clear that $\varphi(k) \in K$. Moreover, as $y_{\varphi(i)} = z_i \in \bar{B}_\delta(x_i)$, this implies that $\varphi^2(i) = i$. Therefore, we have constructed an involution,

$$\varphi : K \longrightarrow K,$$

such that $\forall k \in K,\ \varphi(k) \neq k$. This implies that the cardinality of K is even and that we can write $K = K_1 \coprod K_2$, with:

$$|K_1| = |K_2|, \quad \varphi(K_1) = K_2.$$

The outcome of this construction is that we now have disjoint paths given by the $(x_i \to y_i)_{i\in I}$ and the $(x_k \to x_{\varphi(k)})_{k\in K_1}$. Therefore, we get the following lower bound on $\boldsymbol{E}_\alpha(X)$.

$$\boldsymbol{E}_\alpha(X) \geq \sum_{i\in I} \sum_{e\in\{x_i\to y_i\}} |e|^\alpha + \sum_{k\in K_1} \sum_{e\in\left\{x_k\to x_{\varphi(k)}\right\}} |e|^\alpha.$$

As $\alpha \in [0,1]$, we have that:

$$\boldsymbol{E}_\alpha(X) \geq \sum_{i\in I} \left(\sum_{e\in\{x_i\to y_i\}} |e| \right)^\alpha + \sum_{k\in K_1} \left(\sum_{e\in\left\{x_k\to x_{\varphi(k)}\right\}} |e| \right)^\alpha.$$

By the triangle inequality, and by definition of packing, we have:

$$\boldsymbol{E}_\alpha(X) \geq \sum_{i\in I} \delta^\alpha + \sum_{k\in K_1} \delta^\alpha = \delta^\alpha(|I| + |K_1|) \geq \frac{1}{2} P_\delta^\rho(X)\delta^\alpha,$$

which concludes the proof. □

Theorem 5.3.5. *Let ρ be a pseudometric on $\mathbb{R}^d$. Supposes that Assumption A.3.31 holds and that ℓ is (q, L, ρ)-Lipschitz, for $q \geq 1$. Then, for all $\alpha \in [0,1]$, with probability at least $1-\zeta$, we have:*

$$\sup_{\tau\leq i\leq T} G_S(w_i) \leq 2B\sqrt{\frac{2\log \boldsymbol{E}_\alpha^\rho + \alpha \log\left(\frac{8L\sqrt{n}}{B}\right)}{n}} + \frac{2B}{\sqrt{n}} + 3B\sqrt{\frac{\mathrm{I}_\infty(S, \mathcal{W}_{\tau\to T}) + \log(1/\zeta)}{2n}}.$$

Proof. For better clarity, we assume $T < +\infty$. Let us fix some $\zeta \in (0,1)$, $\delta > 0$, and $\alpha \geq 0$. By Theorem A.3.32, we have, with probability at least $1-\zeta$:

$$\sup_{\tau\leq i\leq T} \left(\mathcal{R}(w_i) - \widehat{\mathcal{R}}_S(w_i)\right) \leq 2L\delta + 2B\sqrt{\frac{2\log N_\delta^\rho(\mathcal{W}_{\tau\to T})}{n}} + 3B\sqrt{\frac{\mathrm{I}_\infty(S, \mathcal{W}_{\tau\to T}) + \log(1/\zeta)}{2n}}.$$

We now bound the covering number appearing in the above equation. By Lemma A.3.38, we have:

$$\boldsymbol{E}_\alpha^\rho(\mathcal{W}_{\tau\to T}) \geq 2^{-\alpha-1} P_{\delta/2}^\rho(\mathcal{W}_{\tau\to T})\delta^\alpha.$$

Moreover, by Lemma A.3.37, we have:

$$\boldsymbol{E}^{\rho}_{\alpha}(\mathcal{W}_{\tau\to T}) \geq 2^{-\alpha-1} N^{\rho}_{\delta}(\mathcal{W}_{\tau\to T})\delta^{\alpha}.$$

We now combine this with our generalization bound by choosing the value:

$$\delta := \frac{B}{L\sqrt{n}},$$

and we get that with probability at least $1-\zeta$, we have:

$$\sup_{\tau\leq i\leq T}\left(\mathcal{R}(w_i) - \widehat{\mathcal{R}}_S(w_i)\right) \leq \frac{2B}{\sqrt{n}} + 2B\sqrt{\frac{2\log(2\boldsymbol{E}^{\rho}_{\alpha}(\mathcal{W}_{\tau\to T})) + \alpha\log\left(\frac{2L\sqrt{n}}{B}\right)}{n}} + 3B\sqrt{\frac{\mathrm{I}_{\infty}(S,\mathcal{W}_{\tau\to T}) + \log(1/\zeta)}{2n}},$$

leading to the desired result. □

A.3.2.7 Proof of the magnitude-based generalization bounds

Lemma A.3.39. *Let $\mathcal{W}\subset\mathbb{R}^d$ be a finite set and $\epsilon := (\epsilon_1,\ldots,\epsilon_n)$ and ρ a pseudo-metric such that $(\mathcal{W},\lambda\rho)$ admits a positive magnitude (according to Definition A.3.27) for every $\lambda>0$. We assume that ℓ is (L,q,ρ)-Lipschitz continuous with $q\in[1,2]$. Then, for any $\lambda>0$, we have:*

$$\mathbb{E}_{\epsilon}\left[\exp\left\{\frac{\lambda}{n}\sup_{w\in\mathcal{W}}\sum_{i=1}^{n}\epsilon_i\ell(w,z_i)\right\}\right] \leq e^{\frac{\lambda^2B^2}{2n}}\mathbf{PMag}\left((L\lambda)\mathcal{W}\right).$$

where **PMag** *is the positive magnitude, see A.3.2.3*

Proof. We first remark that, by Hölder's inequality and the (L,q,ρ)-Lipschitz condition, we have:

$$\forall w,w'\in\mathcal{W},\ \rho_S(w,w') \leq n^{-1/q}\left\|\boldsymbol{L}_S(w) - \boldsymbol{L}_S(w')\right\|_q \leq L\rho(w,w').$$

Let us fix some $\lambda>0$. As $(\mathcal{W},\lambda\rho)$ admits a positive magnitude, we can introduce a canonical weighting $\beta:\mathcal{W}\longrightarrow\mathbb{R}$. By definition of a weighting, we have

$$\forall a\in\mathcal{W},\ \sum_{b\in\mathcal{W}} e^{-\lambda\rho(a,b)}\beta(b) = 1.$$

Moreover, for any $\epsilon\in\{-1,1\}^n$, we introduce:

$$a_{\epsilon} := \mathrm{argmax}_{a\in\mathcal{W}}\sum_{i=1}^{n}\epsilon_i\ell(a,z_i).$$

With those notations, we can compute:

$$\begin{aligned}
1 &\leq \sum_{b\in\mathcal{W}} e^{-\lambda\rho(a_\epsilon,b)}\beta_+(b)\\
&\leq \sum_{b\in\mathcal{W}} e^{-\frac{\lambda}{L}\rho_S(a_\epsilon,b)}\beta_+(b)\\
&= \sum_{b\in\mathcal{W}} \exp\left\{-\frac{\lambda}{Ln}\sum_{i=1}^n |\ell(a_\epsilon,z_i)-\ell(b,z_i)|\right\}\beta_+(b)\\
&\leq \sum_{b\in\mathcal{W}} \exp\left\{-\frac{\lambda}{Ln}\sum_{i=1}^n \epsilon_i(\ell(a_\epsilon,z_i)-\ell(b,z_i))\right\}\beta_+(b)\\
&= \exp\left\{-\frac{\lambda}{Ln}\sum_{i=1}^n \epsilon_i\ell(a_\epsilon,z_i)\right\}\sum_{b\in\mathcal{W}} \exp\left\{\frac{\lambda}{Ln}\sum_{i=1}^n \epsilon_i\ell(b,z_i)\right\}\beta_+(b).
\end{aligned}$$

Therefore, by dividing by the first term on the right-hand side and using the independence of the ϵ_i, we deduce that:

$$\begin{aligned}
\mathbb{E}_\epsilon\left[\exp\left\{\frac{\lambda}{Ln}\sup_{w\in\mathcal{W}}\sum_{i=1}^n \epsilon_i\ell(w,z_i)\right\}\right] &\leq \mathbb{E}_\epsilon\left[\sum_{b\in\mathcal{W}}\exp\left\{\frac{\lambda}{Ln}\sum_{i=1}^n \epsilon_i\ell(b,z_i)\right\}\beta_+(b)\right]\\
&= \sum_{b\in\mathcal{W}}\prod_{i=1}^n \mathbb{E}_\epsilon\left[e^{\frac{\lambda}{Ln}\epsilon_i\ell(b,z_i)}\right]\beta_+(b).
\end{aligned}$$

By Hoeffding's lemma, we have:

$$\begin{aligned}
\mathbb{E}_\epsilon\left[\exp\left\{\frac{\lambda}{Ln}\sup_{w\in\mathcal{W}}\sum_{i=1}^n \epsilon_i\ell(w,z_i)\right\}\right] &\leq e^{\frac{\lambda^2B^2}{2nL^2}}\sum_{b\in\mathcal{W}}\beta_+(b)\\
&= e^{\frac{\lambda^2B^2}{2nL^2}}\mathbf{PMag}\left(\lambda\mathcal{W}\right).
\end{aligned}$$

The result follows by the change of variable $\lambda = \Lambda L$. □

Theorem 5.3.10. *Let ρ be a pseudometric such that $(\mathcal{W},\lambda\rho)$ admits a positive magnitude (according to Definition A.3.27) for every $\lambda>0$. We assume that ℓ is (q,L,ρ)-Lipschitz continuous with $q\geq 1$. Then, for any $s>0$, we have with probability at least $1-\zeta$ that*

$$\sup_{\tau\leq i\leq T} G_S(w_i) \leq \frac{2}{s}\log\mathbf{PMag}^\rho\left(Ls\mathcal{W}_{\tau\to T}\right) + s\frac{B^2}{n} + 3B\sqrt{\frac{\mathrm{I}_\infty(S,\mathcal{W}_{\tau\to T})+\log(1/\zeta)}{2n}}.$$

Proof. The beginning of the proof is completely similar to the proof of A.3.32 up to Equation (xii). More precisely, we have that with probability at least $1-\zeta$:

$$\sup_{\tau\leq i\leq T}\left(\mathcal{R}(w_i)-\widehat{\mathcal{R}}_S(w_i)\right) \leq 2\mathrm{Rad}(\ell,\mathcal{W}_{\tau\to T},S) + 3B\sqrt{\frac{\mathrm{I}_\infty(S,\mathcal{W}_{\tau\to T})+\log(1/\zeta)}{2n}}.$$

By Jensen's inequality, we have, for all $\lambda > 0$:

$$\mathrm{Rad}(\ell, \mathcal{W}_{\tau\to T}, S) \leq \frac{1}{\lambda} \log \mathbb{E}_\epsilon \left[\exp \left\{ \frac{\lambda}{n} \sup_{w \in \mathcal{W}_{\tau\to T}} \sum_{i=1}^{n} \epsilon_i \ell(w, z_i) \right\} \right].$$

Therefore, we can apply Lemma A.3.39 to write that, for all $s > 0$:

$$\mathrm{Rad}(\ell, \mathcal{W}_{\tau\to T}, S) \leq s\frac{B^2}{2n} + \frac{1}{s} \log \mathbf{PMag}\left(Ls\mathcal{W}_{\tau\to T}\right).$$

We deduce that for all $s > 0$, we have with probability at least $1 - \zeta$ that:

$$\sup_{\tau \leq i \leq T} \left(\mathcal{R}(w_i) - \widehat{\mathcal{R}}_S(w_i) \right) \leq s\frac{B^2}{n} + \frac{2}{s} \log \mathbf{PMag}\left(Ls\mathcal{W}_{\tau\to T}\right) + \sqrt{\frac{\mathrm{I}_\infty(S, \mathcal{W}_{\tau\to T}) + \log(1/\zeta)}{2n}}.$$

□

Remark A.3.40 (Link between magnitude and positive magnitude). Let $\mathcal{W} \subset \mathbb{R}^M$ be a finite set (for some M), of cardinality N, and ρ a metric on $\mathcal{W}$. If we denote the similarity matrix, for a given value of $s > 0$, by $M_s(a, b) = e^{-\rho(a,b)}$, then it is clear that:

$$M_s \underset{s\to\infty}{\longrightarrow} I_N.$$

Moreover, by continuity of the inverse, this implies that the weighting associated to $s > 0$, *i.e.* $\beta_s : \mathcal{W} \to \mathbb{R}$, satisfy:

$$\forall a \in \mathcal{W},\ \beta_s(a) \underset{s\to\infty}{\longrightarrow} 1.$$

From this, we first deduce that, for $s \to \infty$, we have $\mathrm{Mag}^\rho(s\mathcal{W}) \to N$. Moreover, by continuity of the inverse, this means that, up to a certain s, the weighting $(\beta_s(a))_{a\in\mathcal{W}}$ only has positive elements. Therefore, this implies that, for s big enough, one has $\mathrm{Mag}^\rho(s\mathcal{W}) = \mathbf{PMag}^\rho(s\mathcal{W})$.

Thanks to our definitions for positive magnitude in pseudometric spaces, given in A.3.2.3, this observation extends to the pseudometric case.

Remark A.3.41 (Extension to infinite sets). There exist extensions of the definition of magnitude beyond finite sets [Meckes, 2013, 2015]. More specifically, weightings are then represented by measures on the set. It is clear from the above proofs that we can extend the positive magnitude in this setting and that the proof would follow similar lines. Therefore, our theory provides upper bounds of Rademacher complexity in terms of positive magnitude in more general cases than the one we use in this work.

A.3.3 Additional Experimental Details

In this section, we give additional details regarding the models, datasets, and hyperparameters used in our experiments.

A.3.3.1 Experimental setting

Vision Transformers Architecture and implementation details

Table A.1: Architecture details for the vision transformers (taken from [Gani et al., 2022]). *WS refers to Window Size.*

Model	Dataset	Depth	Patch Size	Token Dim	Heads	MLP-ratio	WS	#Params
ViT [Touvron et al., 2021a]	CIFAR10	9	4	192	12	2	-	2697610
ViT [Touvron et al., 2021a]	CIFAR100	9	4	192	12	2	-	2714980
Swin [Liu et al., 2021]	CIFAR10	[2,4,6]	4	96	[3,6,12]	2	4	7048612
Swin [Liu et al., 2021]	CIFAR100	[2,4,6]	4	96	[3,6,12]	2	4	7083262
CaiT [Touvron et al., 2021b]	CIFAR10	24	4	192	4	2	-	8053450
CaiT [Touvron et al., 2021b]	CIFAR100	24	4	192	4	2	-	8070820

The design of the ViT has been modified to accommodate for the small datasets as per [Raghu et al., 2021]. Our implementation is based on the [Gani et al., 2022], which is based on the `timm` library with the architecture parameters presented in Table A.1. The implementation of Swin is based on the Swin-Transformer libarary and the implementation of CaiT is predominantly based on the `timm` library with some modifications. The full version can be found in the supplementary code.

Instead of training from scratch, which is extremely time-consuming, we used the pre-trained weights available from the GitHub repository of the paper [Gani et al., 2022], we further fintetuned them for 100 epochs on the dataset CIFAR10 or CIFAR100 to achieve the optimum performance reported in the paper [Gani et al., 2022]. Then we verified that the finetuned weights achieved 100% training performance, and then they were the starting point of our computational framework. We ran the transformer experiments on 18 NVIDIA 2080Ti GPUs, and the graph experiments on 18 Intel Xeon Silver 4114 CPUs.

GNN Architecture and implementation details We will briefly talk about the details of GraphSage [Hamilton et al., 2017] and GatedGCN [Bresson and Laurent, 2017], prior works we use in our experiments. GraphSage [Hamilton et al., 2017] is an improvement over the GCN (Graph ConvNets) model [Kipf and

Welling, 2016] and it incorporates each node's own features from the previous layer in an explicit way by the update equation:

$$h_i^{l+1} = \text{ReLU}(U^l \text{Concat}(h_i^l, \text{Mean}_{j \in N_i} h_j^l)),$$

where N_i is the neighbourhood of node i, h_i^l is the feature vector and $U^l \in \mathbb{R}^{d \times 2d}$. We use the graph-pooling version of GraphSage, with the following update equation:

$$h_i^{l+1} = \text{ReLU}(U^l \text{Concat}(h_i^l, \text{Max}_{j \in N_i} \text{ReLU}(V^l h_j^l))),$$

where $V^l \in \mathbb{R}^{d \times d}$. GatedGCN (Gated Graph ConvNet) [Bresson and Laurent, 2017] uses the following update equation:

$$h_i^{l+1} = h_i^l + \text{ReLU}(\text{BN}(U^l h_i^l + \sum_{j \in N_i} e_{ij}^l \odot V^l h_i^l)),$$

where $U^l, V^l \in \mathbb{R}^{d \times d}$, $\odot$ is the Hadamard product, and the edge gates e_{ij}^l have the following definitions:

$$e_{ij}^l = \frac{\sigma(\hat{e}_{ij}^l)}{\sum_{j' \in N_i} \sigma(\hat{e}_{ij}^l) + \epsilon},$$

$$\hat{e}_{ij}^l = \hat{e}_{ij}^{l-1} + \text{ReLU}(\text{BN}(A^l h_i^{l-1} + B^l h_i^{l-1} + C^l \hat{e}_{ij}^{l-1})),$$

where σ is the sigmid funciton, ϵ is a small constant for numerical stability, $A^l, B^l, C^l \in \mathbb{R}^{d \times d}$, and BN stands for Batch Normalization.

We used the code provided by [Dwivedi et al., 2023], which relies on the `dgl` library implementation of GraphSage and GatedGCN. We trained GraphSage and GatedGCN until 100% training accuracy, following the setup in [Dwivedi et al., 2023]. All experiments were ran on 18 Intel Xeon Silver 4114 CPUs. Each experiment (one fixed batch size and learning rate) was run on a single CPU and 18 experiments were run on the server at any given time (on different CPUs).

A.3.3.2 Hyperparameter details

Hyperparameters shared among experiments.. For the Vision Transformers experiments, we varied the learning rate range $[10^{-5}, 10^{-3}]$, and batch size in the

range $[8, 256]$. For the graph experiments, $[10^{-6}, 10^{-4}]$, and batch size in the range $[8, 256]$. For all experiments, we used 0.1 proportion of the training data for the computation of the pseudo matrix, apart from CaiT and Swin on CIFAR100, where we used 0.09 proportion of the training data due to memory constraints. All experiments use a 6×6 grid of hyperparameters which is specified as follows.

ViT on CIFAR10**.** We selected 6 values for the learning rate in the range $[10^{-5}, 10^{-3}]$, and the batch size between $[8, 256]$, and data proportion for the computation of the pseudo-distance (ρ_S) of 10%.

ViT on CIFAR100**.** We selected 6 values for the learning rate in the range $[10^{-5}, 10^{-3}]$, and the batch size between $[8, 256]$, and data proportion for the computation of the pseudo-distance (ρ_S) of 10%.

CaiT on CIFAR10**.** We selected 6 values for the learning rate in the range $[10^{-5}, 10^{-3}]$, batch size between $[8, 256]$, and data proportion for the computation of the pseudo-distance (ρ_S) of 10%.

CaiT on CIFAR100**.** We selected 6 values for the learning rate in the range $[10^{-5}, 10^{-3}]$, batch size between $[8, 256]$, and data proportion for the computation of the pseudo-distance (ρ_S) of 9%.

Swin on CIFAR10**.** We selected 6 values for the learning rate in the range $[10^{-5}, 10^{-3}]$, batch size between $[8, 256]$, and data proportion for the computation of the pseudo-distance (ρ_S) of 10%.

Swin on CIFAR100**.** We selected 6 values for the learning rate in the range $[10^{-5}, 10^{-3}]$, batch size between $[8, 256]$, and data proportion for the computation of the pseudo-distance (ρ_S) of 9%.

GatedGCN. We selected 6 values for the learning rate in the range $[10^{-6}, 10^{-4}]$, the batch size between $[8, 256]$ and data proportion for the computation of the pseudo-distance (ρ_S) of 10% (see Section 5.4). We note that for due to time constraints, the experiments with batch sizes of 8 and 256 for the Euclidean metric were not complete.

GraphSage. We selected 6 values for the learning rate in the range $[10^{-6}, 10^{-4}]$, the batch size between $[8, 256]$, and data proportion for the computation of the pseudo-distance (ρ_S) of 10%.

ViT on CIFAR10 **(Adam).** We selected 6 values for the learning rate in the range $[10^{-5}, 10^{-3}]$, and the batch size between $[8, 256]$, and data proportion for the computation of the pseudo-distance (ρ_S) of 10%.

ViT on CIFAR10 **(SGD).** We selected 6 values for the learning rate in the

range $[5 \times 10^{-3}, 10^{-1}]$, and the batch size between $[8, 256]$, and data proportion for the computation of the pseudo-distance (ρ_S) of 10%.

ViT on CIFAR10 (RMSprop). We selected 6 values for the learning rate in the range $[10^{-6}, 10^{-3}]$, and the batch size between $[8, 512]$, and data proportion for the computation of the pseudo-distance (ρ_S) of 10%.

A.3.4 Additional Experimental results

In this section, we present additional empirical results, in addition to what was already presented in the main part of this document. We divide this section into three parts. First, we quickly explore in A.3.4.1 the consequence of our choice of estimation technique of the worst-case generalization error. In A.3.4.2 we report additional experiments based on vision transformers and in A.3.4.3 we include additional illustration of the GNN experiments.

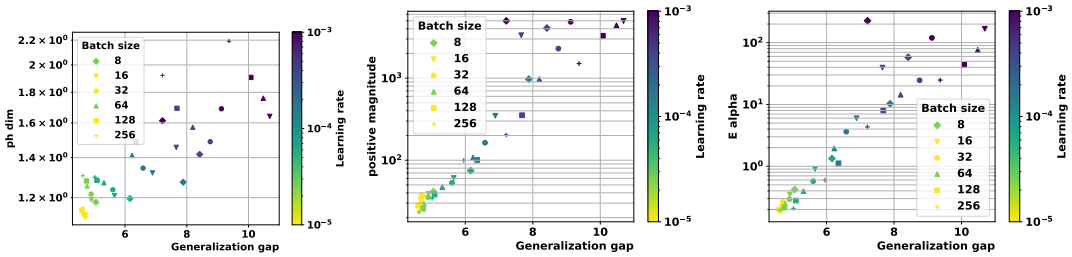
A.3.4.1 About the final accuracy gap and the worst accuracy gap

Our main theoretical results, presented in Section 5.3, apply to the worst-case generalization error over the trajectory, *i.e.* on the quantity $\sup_{\tau \leq k \leq T} \left(\mathcal{R}(w_k) - \widehat{\mathcal{R}}_S(w_k) \right)$. However, computing this quantity over the whole trajectory may be extremely expensive as it requires evaluating the model on the whole dataset at each iteration (this is a similar problem to the one encountered for the computation of the data-dependent distance matrices, discussed in Section 5.4). Previous studies on worst-case TDA-inspired generalization bounds circumvented this issue by reporting the final accuracy gap as the "generalization error" in their experiments (as it is the case in our work, most existing experiments consist of classification tasks).

In our work, we argue that the true worst-case generalization error may however have a different behavior than the final accuracy gap. In order to estimate this quantity in a computationally friendly way, we used the following procedure: we periodically estimated the test accuracy during the training, computed its minimum value $\text{acc}_{\text{test-worst}}$ and substracted it from the final train accuracy ($\text{acc}_{\text{train-final}}$) to obtain the "generalization gap" $\widehat{G}_S$ reported in our main experiments, *i.e.*,

$$\widehat{G}_S := \text{acc}_{\text{train-final}} - \text{acc}_{\text{test-worst}}.$$

Note that in addition to being a good proxy to the true error appearing in our theory, the above quantity could be of independent experimental interest.

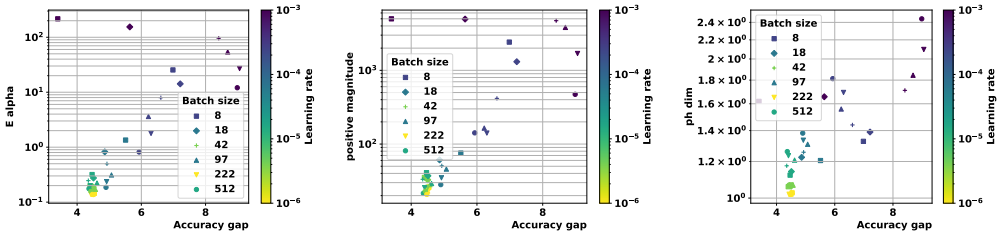


(a) (b) (c)

Figure A.7: ρ_S-based complexity measures vs. generalization gap for a ViT trained on CIFAR10: $\dim_{\mathrm{PH}}$ (*left*), $\mathbf{PMag}(\sqrt{n})$ (*middle*), and $\boldsymbol{E}_1$ (*right*).

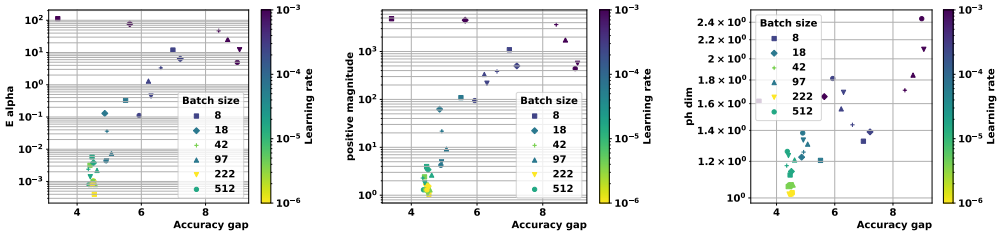
In order to assess that our main conclusions remain valid if the final accuracy gap is used instead of $\widehat{G}_S$, we present here a few additional experiments using the final accuracy gap as a generalization measure (it is denoted `Accuracy gap` in the figures.) In the case of a ViT on CIFAR10, this is shown in Figure A.8 and Figure A.9. We observe that our proposed topological complexities also correlate very well with the final accuracy gap, and outperform the previously proposed PH dimensions [Birdal et al., 2021, Dupuis et al., 2023].

In addition to these findings, we make two additional new observations. First, the Ph dim, while outperformed by our proposed metric, has better granulated Kendall's coefficients when compared to the final accuracy gap than the worst generalization error ($\boldsymbol{\Psi}$ goes from 0.20 to 0.36). This may explain why we observed poor performance of PH-dim in Figure 5.4(a). Second, we observe that the correlation seems to be slightly less good with the final accuracy gap, especially for high learning rates, which seems to be similar behavior to what was reported in [Dupuis et al., 2023].



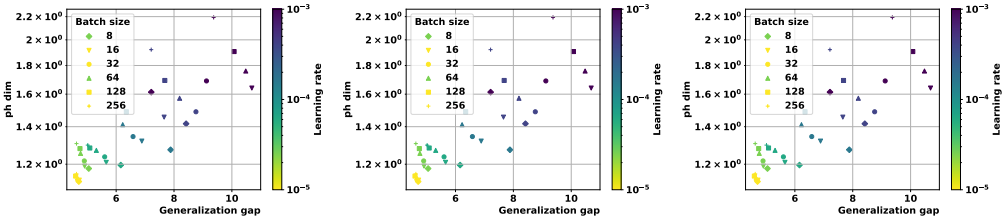
(a) (b) (c)

Figure A.8: ViT on CIFAR10 with ρ_S-pseudometric, using the final accuracy gap as a generalization measure. (a) $\boldsymbol{E}_\alpha$, (b) $\mathbf{PMag}(\sqrt{n})$, (c) $\dim_{\mathrm{PH}}$



(a) (b) (c)

Figure A.9: ViT on CIFAR10 with 01-pseudometric, using the final accuracy gap as a generalization measure. In plot (a), we depict $\boldsymbol{E}_\alpha$, in plot (b), $\mathbf{PMag}(\sqrt{n})$, and in plot (c), $\dim_{\mathrm{PH}}$.



(a) (b) (c)

Figure A.10: ρ_S-based complexity measures vs. generalization gap for a ViT trained on CIFAR10: $\dim_{\mathrm{PH}}$ (*left*), $\mathbf{PMag}(\sqrt{n})$ (*middle*), and $\boldsymbol{E}_1$ (*right*).

A.3.4.2 Vision Transformers - additional experiments

We compare the performance of the different metrics by using the granulated Kendall's coefficients introduced in [Jiang et al., 2019]. The experiments presented here use 3 different Vision Transformers (ViT [Touvron et al., 2021a], CaiT [Touvron et al., 2021b], Swin [Liu et al., 2021]) on CIFAR10 and CIFAR100. As a baseline, we use the $\dim_{\mathrm{PH}}$ introduced in [Birdal et al., 2021] and the data-dependent dimension with the pseudometric $\dim_{\mathrm{PH}}$ from [Dupuis et al., 2023].

Here we present the full results on each dataset and model. They can be found in Table A.3 for CaiT and CIFAR10, A.5 for Swin and CIFAR10, A.2 for ViT and CIFAR100 and A.4 for CaiT and CIFAR100. The plots from each experiment for every computed quantity can be found in (the remaining 3 quantities for ViT and CIFAR10).

A.3.4.3 Graph Neural Networks – additional experiments

In Table 5.1, we already presented the correlation coefficients for all quantities for the GNN models considered in our study (GraphSage, GatedGCN) [Dwivedi

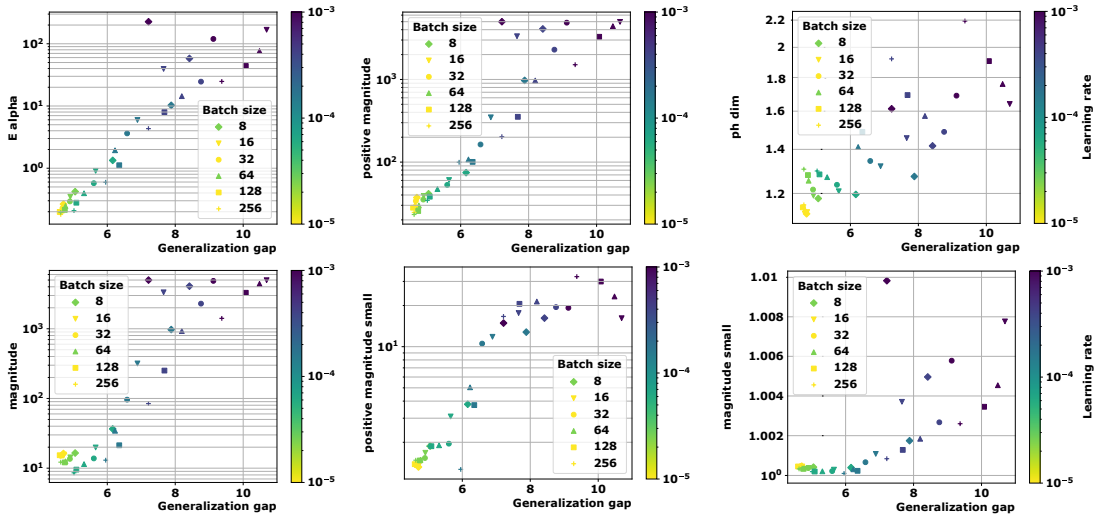


Figure A.11: ViT on CIFAR10 with ρ_S

et al., 2023] (we have selected the models which achieve 100% training accuracy)) and Graph-MNIST. We can observe a nice correlation, outperforing dim-PH in most experiments. As it was observed for the transformer-based experiments, the correlation seems to be better for the data-dependent-metrics. This is an important fact, as no sparse random projection was used to compute the Euclidean distance matrices in the GNN experiments (it was not necessary as these models have less parameters than the tramsformers considered above). This shows that the fact the data-dependent pseudometrics outperform the Euclidean distance also happens in the absence of these projections. It also shows that all quantities seem to yield better correlations in the absence of random projections, at least in the GNN expsriments.

The corresponding plots for GatedGCN can be seen in Figure A.32 with the pseudometric, Figure A.33 for the Euclidean and A.34 for 01. The plots for GraphSage are reported in Figure A.29, Figure A.30 and Figure A.31.

We can observe a strong correlation on these figures, outperforing dim-PH in most cases. As it was observed for the transformer-based experiments, the correlation seems to be better for the data-dependent-metrics. This is an important fact, as no sparse random projection was used to compute the Euclidean distance matrices in the GNN experiments[3]. This shows that data-dependent pseudometrics outperform the Euclidean distance also in the absence of these projections. In

[3]A sparse random projection was not necessary as these models have less parameters than the tramsformers considered above

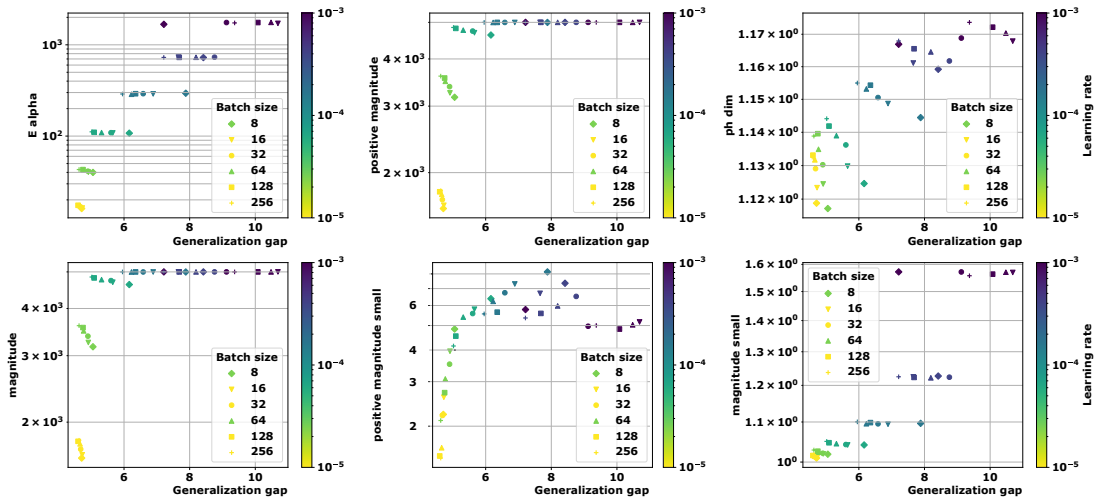


Figure A.12: ViT on CIFAR10 with $\|\cdot\|_2$

addition, all quantities seem to yield better correlations in the absence of random projections, at least in the GNN expsriments.

Interestingly, a few failure cases can be seen on these plots. Indeed, Mag(0.01) and **PMag**(0.01) seem to be almost constant and near 1. This indicates that the scale choice $s = 0.01$ was not suited for these experiments; this behavior was already reflected in Table 5.1 through very low Kendall's coefficients, indicating the absence of meaningful correlation. However, Mag($\sqrt{n}$) and **PMag**($\sqrt{n}$) provide significantly better correlation, which supports our main claims, as $s = \sqrt{n}$ has been argued in Section 5.3.3 to be a particulary relevant choice of scale factor.

Note finally that the PH-dim plots for the 01-pseudometric failed to produce numbers in these graphs experiments (this is why they are either missing or look irrelevant). As before, we gave away this fact in Table 5.1 by imposing our granulated Kendall's coefficients implementation to return zeros in the absence of correlation, hence the small numbers observed in this case. That being said, this behavior should not be seen as an issue. Indeed, PH-dim with 01-pseudometric consists (in theory) in estimating the dimension of a subset of a discrete hypercube, which is always 0. The reason we still reported PH-dim for this pseudometric is for consistence and to test the implementation of [Birdal et al., 2021, Dupuis et al., 2023] in this non-standard setting; it is however not theoretically grounded.

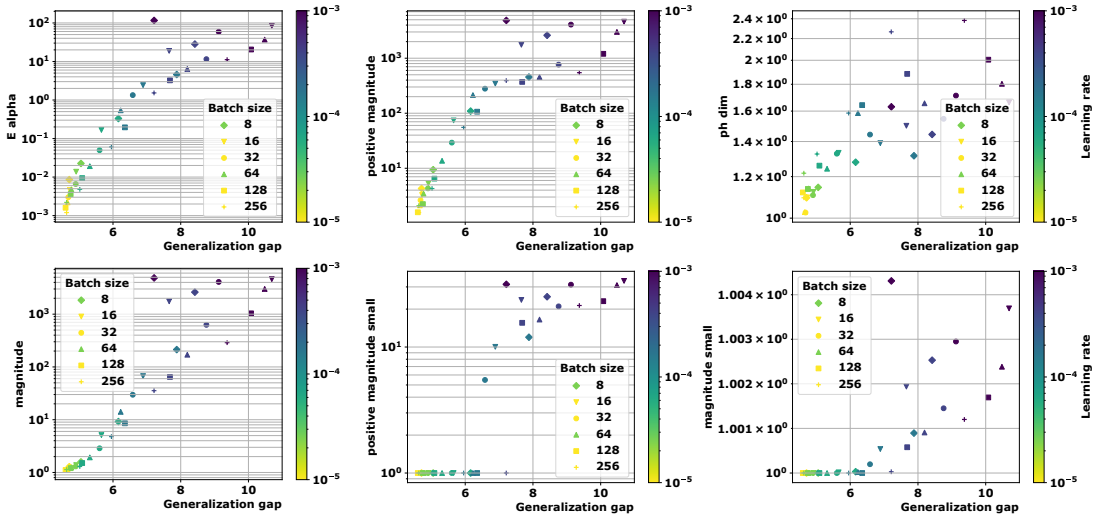


Figure A.13: ViT on CIFAR10 with 01-pseudometric

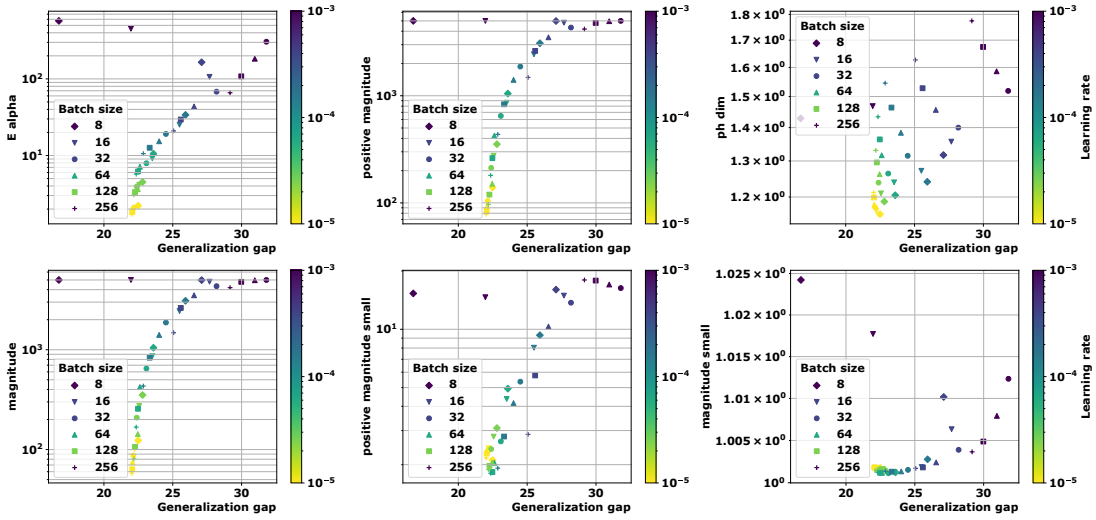


Figure A.14: ViT on CIFAR100 with ρ_S

Table A.2: Correlation coefficients for all quantities for **ViT model** and **CIFAR100 dataset**. The corresponding plots are presented in Figures A.14, Figure A.15 and Figure A.16.

Metric	Complexity	$\boldsymbol{\psi}_{\text{LR}}$	$\boldsymbol{\psi}_{\text{BS}}$	$\boldsymbol{\Psi}$	τ
ρ_S	$\boldsymbol{E}_\alpha$	0.78	0.71	**0.74**	0.70
	$\text{Mag}(\sqrt{n})$	0.78	0.71	**0.74**	**0.72**
	$\text{Mag}(0.01)$	0.15	0.11	0.13	0.17
	$\textbf{PMag}(\sqrt{n})$	0.78	0.71	**0.74**	**0.72**
	$\textbf{PMag}(0.01)$	0.60	0.62	0.61	0.56
	$\dim_{\text{PH}}$ [Dupuis et al., 2023]	0.77	-0.71	0.03	0.36
$\lVert \cdot \rVert_2$	$\boldsymbol{E}_\alpha$	0.77	0.51	0.64	**0.67**
	$\text{Mag}(0.01)$ (Chapter 4)	0.77	-0.69	0.04	0.50
	$\text{Mag}(\sqrt{n})$	0.77	-0.45	0.16	0.54
	$\textbf{PMag}(0.01)$	0.82	0.53	**0.68**	0.66
	$\textbf{PMag}(\sqrt{n})$	0.78	-0.45	0.16	0.54
	$\dim_{\text{PH}}$ [Birdal et al., 2021]	0.77	-0.71	0.03	0.37
01	$\boldsymbol{E}_\alpha$	0.77	0.71	**0.74**	0.70
	$\text{Mag}(\sqrt{n})$	0.77	0.71	**0.74**	**0.71**
	$\text{Mag}(0.01)$	0.68	0.51	0.59	0.59
	$\textbf{PMag}(\sqrt{n})$	0.77	0.71	**0.74**	0.70
	$\textbf{PMag}(0.01)$	0.72	0.71	0.71	0.63
	$\dim_{\text{PH}}$	0.73	0.02	0.37	0.57

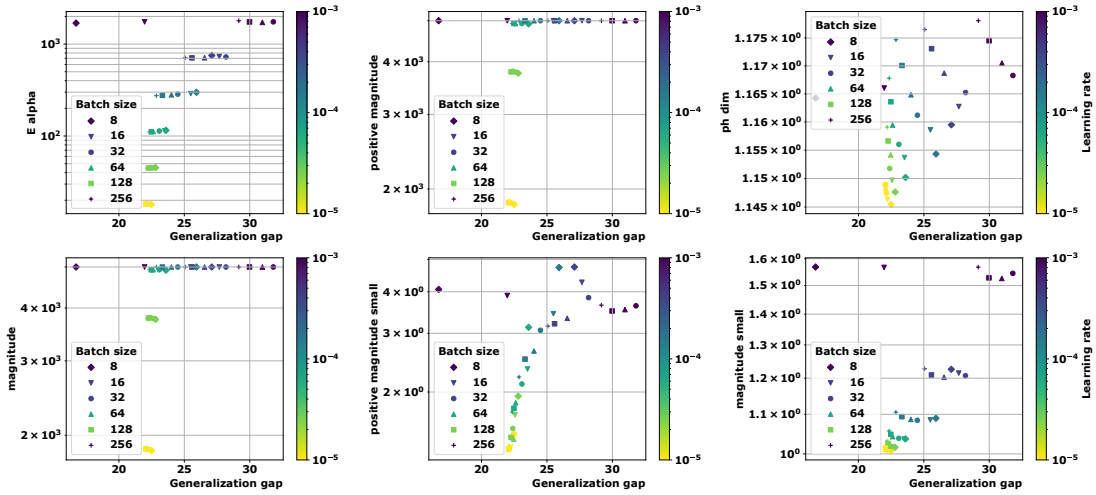


Figure A.15: ViT on CIFAR100 with $\|\cdot\|_2$

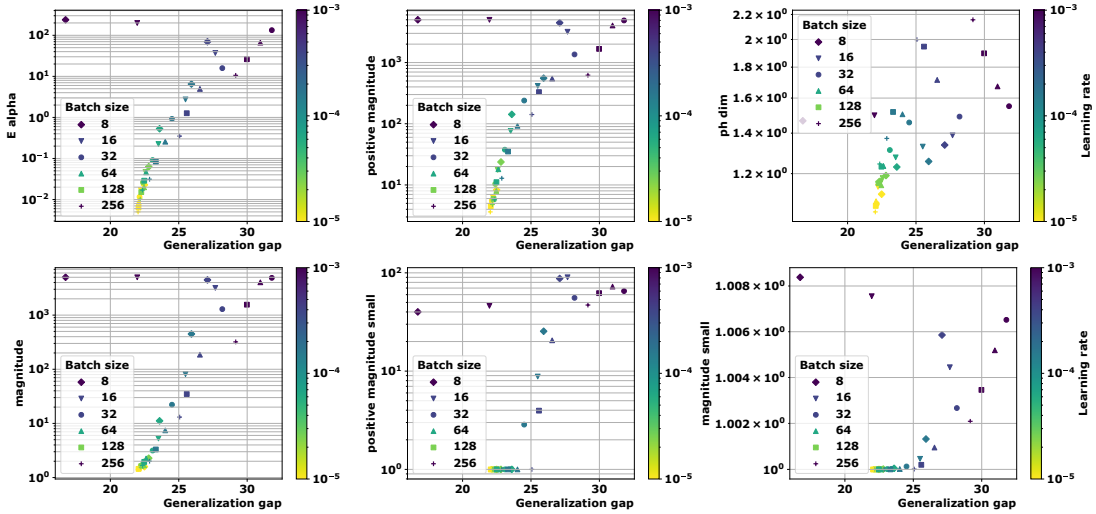


Figure A.16: ViT on CIFAR100 with 01-pseudometric

Table A.3: Correlation coefficients for all quantities for **CaiT model** and **CIFAR10 dataset**. The corresponding plots can be seen in Figures A.17, A.18 and A.19.

Metric	Complexity	$\boldsymbol{\psi}_{\text{LR}}$	$\boldsymbol{\psi}_{\text{BS}}$	$\boldsymbol{\Psi}$	τ
ρ_S	$\boldsymbol{E}_\alpha$	0.91	0.33	**0.62**	**0.78**
	$\text{Mag}(\sqrt{n})$	0.91	0.33	**0.62**	0.75
	$\text{Mag}(0.01)$	0.75	0.29	0.52	0.69
	$\textbf{PMag}(\sqrt{n})$	0.91	0.33	**0.62**	0.75
	$\textbf{PMag}(0.01)$	0.87	0.38	**0.62**	0.75
	$\dim_{\text{PH}}$ [Dupuis et al., 2023]	0.91	-0.19	0.36	0.75
$\|\cdot\|_2$	$\boldsymbol{E}_\alpha$	0.91	0.38	**0.64**	**0.85**
	$\text{Mag}(\sqrt{n})$	0.89	-0.42	0.23	0.73
	$\text{Mag}(0.01)$ (Chapter 4)	0.91	-0.15	0.37	0.77
	$\textbf{PMag}(\sqrt{n})$	0.89	-0.42	0.23	0.73
	$\textbf{PMag}(0.01)$	0.53	0.26	0.4	0.48
	$\dim_{\text{PH}}$ [Birdal et al., 2021]	0.91	-0.31	0.30	0.67
01	$\boldsymbol{E}_\alpha$	0.91	0.33	0.62	**0.84**
	$\text{Mag}(\sqrt{n})$	0.91	0.33	0.62	0.77
	$\text{Mag}(0.01)$	0.86	0.33	0.60	0.76
	$\textbf{PMag}(\sqrt{n})$	0.91	0.33	0.62	0.79
	$\textbf{PMag}(0.01)$	0.88	0.44	**0.66**	0.71
	$\dim_{\text{PH}}$	0.91	-0.13	0.39	0.78

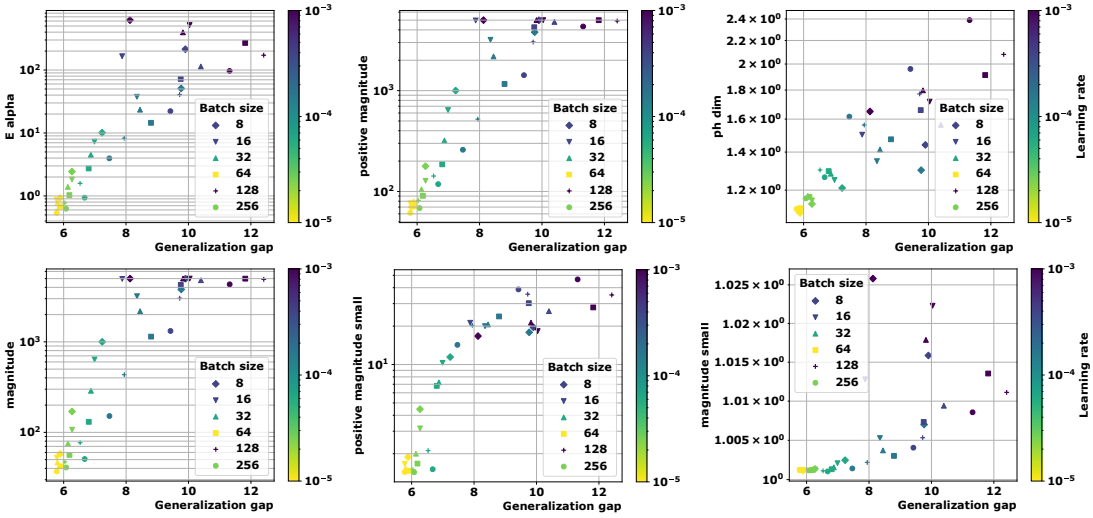


Figure A.17: CaiT on CIFAR10 with ρ_S-pseudometric.

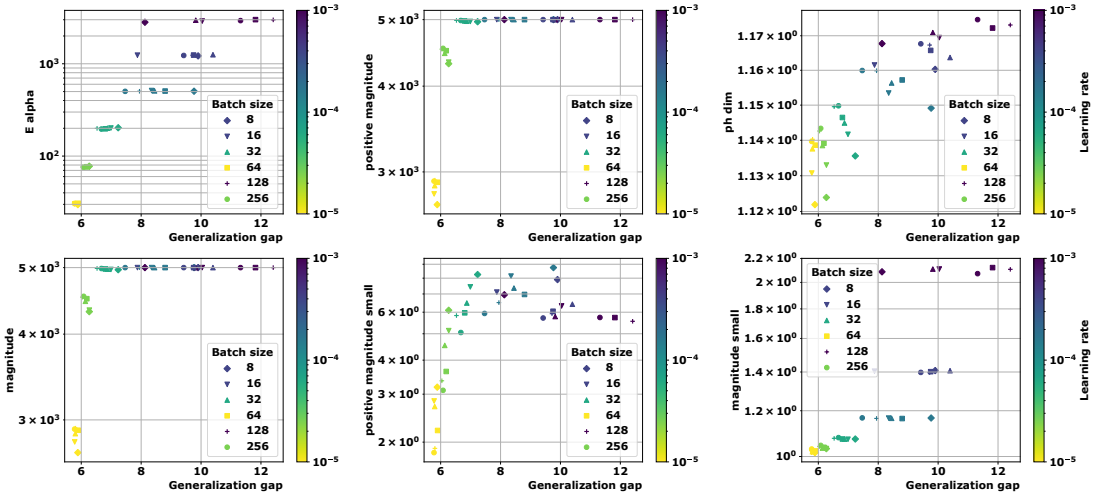


Figure A.18: CaiT on CIFAR10 with $\|\cdot\|_2$ distance.

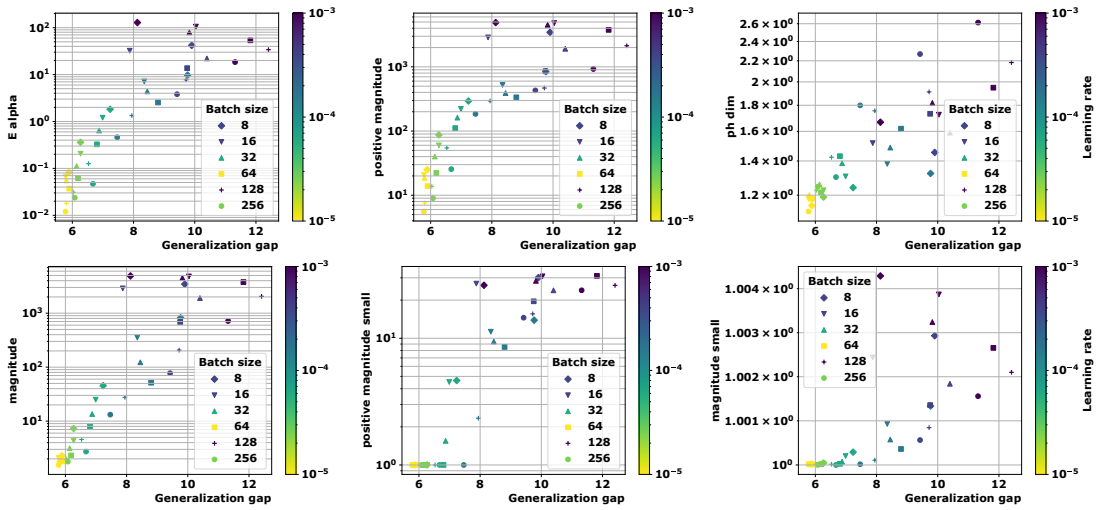


Figure A.19: CaiT on CIFAR10 with 01-pseudometric.

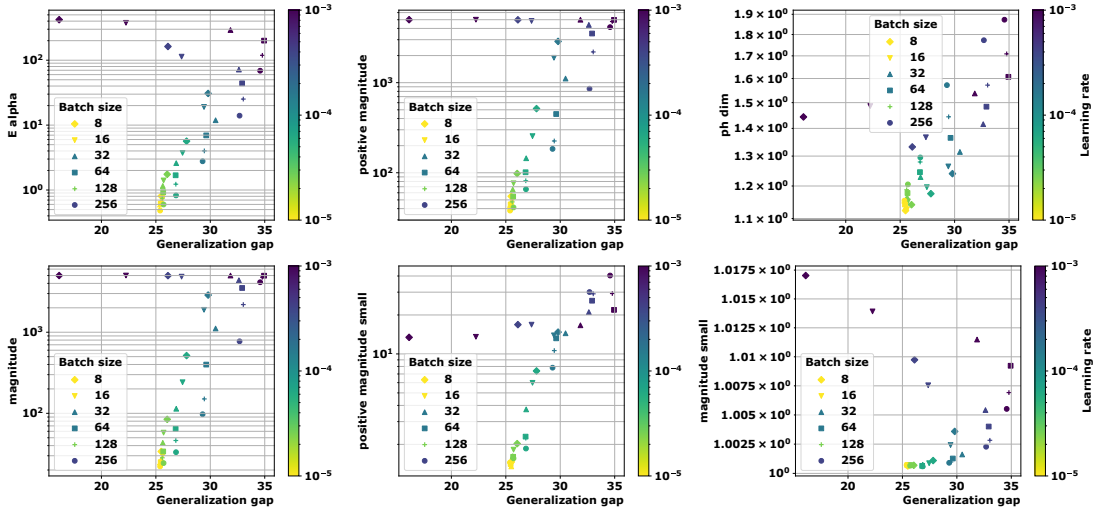


Figure A.20: CaiT on CIFAR100 with ρ_S-pseudometric.

Table A.4: Correlation coefficients for all quantities for **CaiT model** and **CIFAR100 dataset**. The corresponding plots can be seen in A.20, A.21 and A.22

Metric	Complexity	$\boldsymbol{\psi}_{\text{LR}}$	$\boldsymbol{\psi}_{\text{BS}}$	Ψ	τ
ρ_S	E_α	0.67	0.13	0.40	0.54
	$\text{Mag}(\sqrt{n})$	0.67	0.13	0.40	0.52
	$\text{Mag}(0.01)$	0.47	-0.18	0.14	0.36
	$\textbf{PMag}(\sqrt{n})$	0.67	0.13	0.40	0.53
	$\textbf{PMag}(0.01)$	0.76	0.53	**0.64**	**0.71**
	$\dim_{\text{PH}}$ [Dupuis et al., 2023]	0.67	-0.13	0.27	0.56
$\|\cdot\|_2$	E_α	0.67	0.40	**0.53**	0.64
	$\text{Mag}(\sqrt{n})$	0.68	0.33	0.50	**0.65**
	$\text{Mag}(0.01)$ (Chapter 4)	0.66	-0.33	0.17	0.54
	$\textbf{PMag}(\sqrt{n})$	0.68	0.33	0.50	**0.65**
	$\textbf{PMag}(0.01)$	0.62	0.09	0.36	0.43
	$\dim_{\text{PH}}$ [Birdal et al., 2021]	0.64	-0.09	0.28	0.50
01	E_α	0.67	0.13	0.40	0.52
	$\text{Mag}(\sqrt{n})$	0.67	0.13	0.40	**0.57**
	$\text{Mag}(0.01)$	0.61	0.18	0.40	0.43
	$\textbf{PMag}(\sqrt{n})$	0.67	0.11	0.39	0.53
	$\textbf{PMag}(0.01)$	0.65	0.41	**0.53**	0.48
	01 loss	0.58	0.07	0.32	**0.57**

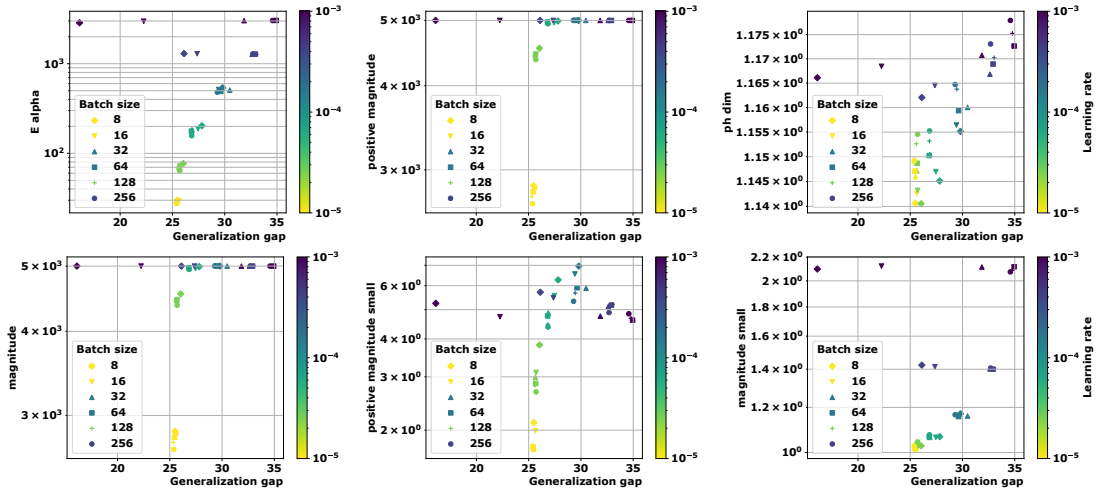


Figure A.21: CaiT on CIFAR100 with $\|\cdot\|_2$.

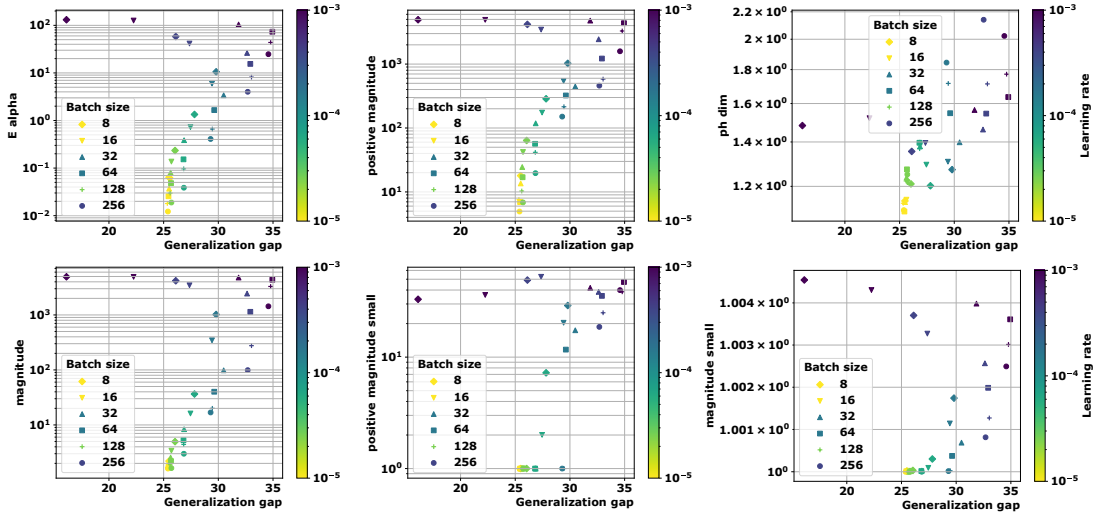


Figure A.22: CaiT on CIFAR100 with 01-pseudometric.

Table A.5: Correlation coefficients for all quantities for **Swin model** and **CIFAR10**. The corresponding plots are in Figure A.23, A.24 and A.25.

Metric	Complexity	ψ_{LR}	ψ_{BS}	Ψ	τ
ρ_S	E_α	0.97	0.58	**0.77**	0.86
	$\text{Mag}(\sqrt{n})$	0.97	0.57	**0.77**	0.84
	$\text{Mag}(0.01)$	0.87	0.58	0.72	0.75
	PMag$(\sqrt{n})$	0.98	0.55	**0.77**	**0.87**
	PMag(0.01)	0.76	0.20	0.48	0.65
	$\dim_{\text{PH}}$ [Dupuis et al., 2023]	0.97	-0.57	0.19	0.67
$\|\cdot\|_2$	E_α	0.97	-0.04	0.46	**0.84**
	$\text{Mag}(\sqrt{n})$	0.97	-0.43	0.27	0.77
	$\text{Mag}(0.01)$ (Chapter 4)	0.98	-0.22	0.38	0.80
	PMag$(\sqrt{n})$	0.98	-0.43	0.27	0.77
	PMag(0.01)	0.51	0.53	**0.52**	0.47
	$\dim_{\text{PH}}$ [Birdal et al., 2021]	0.95	-0.57	0.18	0.69
01	E_α	0.97	0.58	**0.77**	0.84
	$\text{Mag}(\sqrt{n})$	0.97	0.58	0.77	**0.86**
	$\text{Mag}(0.01)$	0.94	0.48	0.71	0.79
	PMag$(\sqrt{n})$	0.98	0.58	0.78	0.87
	PMag(0.01)	0.92	0.42	0.67	0.78
	$\dim_{\text{PH}}$	0.93	-0.28	0.32	0.69

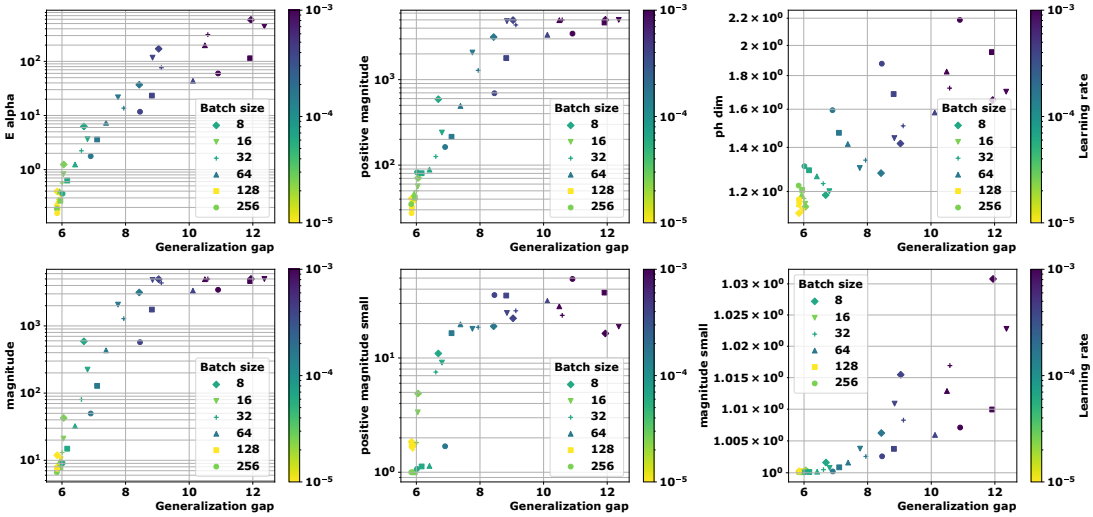


Figure A.23: Swin on CIFAR10 with ρ_S-pseudometric.

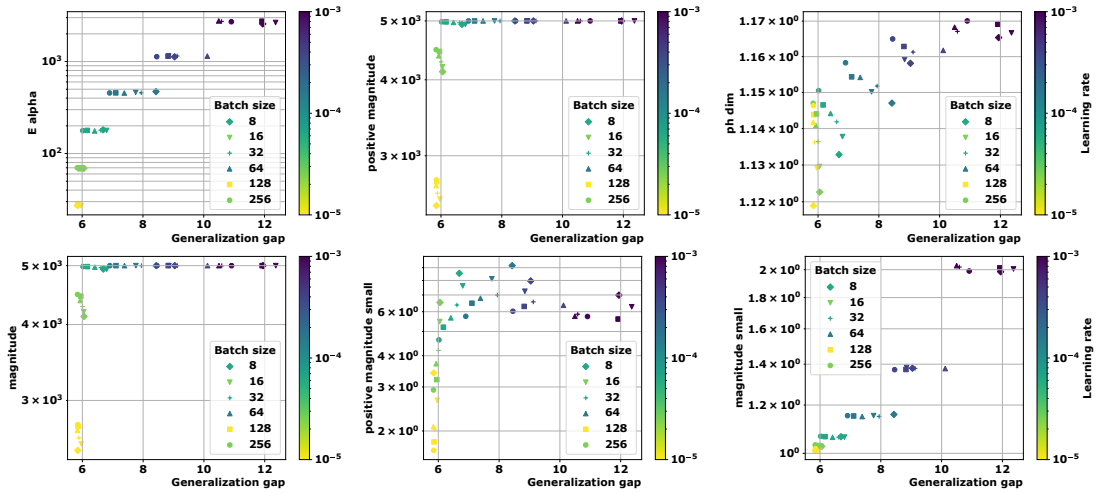


Figure A.24: Swin on CIFAR10 with $\|\cdot\|_2$.

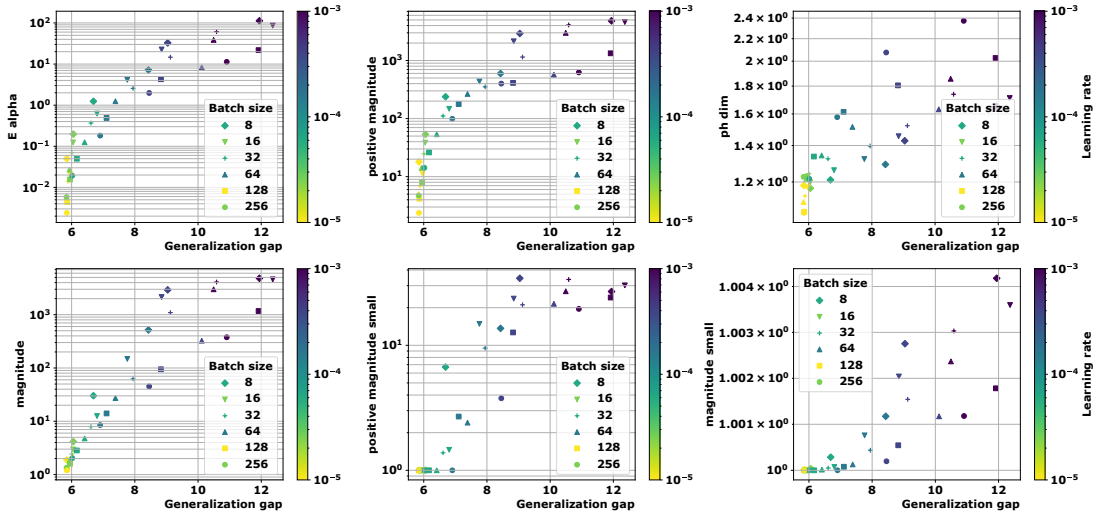


Figure A.25: Swin on CIFAR10 with 01-pseudometric.

Table A.6: Correlation coefficients for all quantities for **Swin model** and **CIFAR100**. See Figures A.26, A.27 and A.28 for the corresponding plots.

Metric	Complexity	ψ_{LR}	ψ_{BS}	Ψ	τ
ρ_S	E_α	0.69	0.47	0.58	0.62
	$\text{Mag}(\sqrt{n})$	0.56	0.47	0.51	0.51
	$\text{Mag}(0.01)$	0.31	0.47	0.39	0.33
	PMag$(\sqrt{n})$	0.69	0.47	0.58	0.63
	PMag(0.01)	0.71	0.58	**0.64**	**0.68**
	$\dim_{\text{PH}}$ [Dupuis et al., 2023]	0.69	-0.47	0.11	0.50
$\|\cdot\|_2$	E_α	0.69	0.22	0.46	0.63
	$\text{Mag}(\sqrt{n})$	0.71	-0.57	0.07	0.53
	$\text{Mag}(0.01)$ (Chapter 4)	0.69	-0.44	0.12	0.53
	PMag$(\sqrt{n})$	0.71	-0.57	0.07	0.53
	PMag(0.01)	0.64	0.51	**0.58**	0.46
	$\dim_{\text{PH}}$ [Birdal et al., 2021]	0.69	-0.47	0.11	0.45
01	E_α	0.69	0.47	**0.58**	0.61
	$\text{Mag}(\sqrt{n})$	0.69	0.47	**0.58**	**0.62**
	$\text{Mag}(0.01)$	0.61	0.27	0.44	0.50
	PMag$(\sqrt{n})$	0.69	0.47	**0.58**	**0.62**
	PMag(0.01)	0.65	0.49	0.57	0.54
	$\dim_{\text{PH}}$	0.64	0.04	0.34	0.51

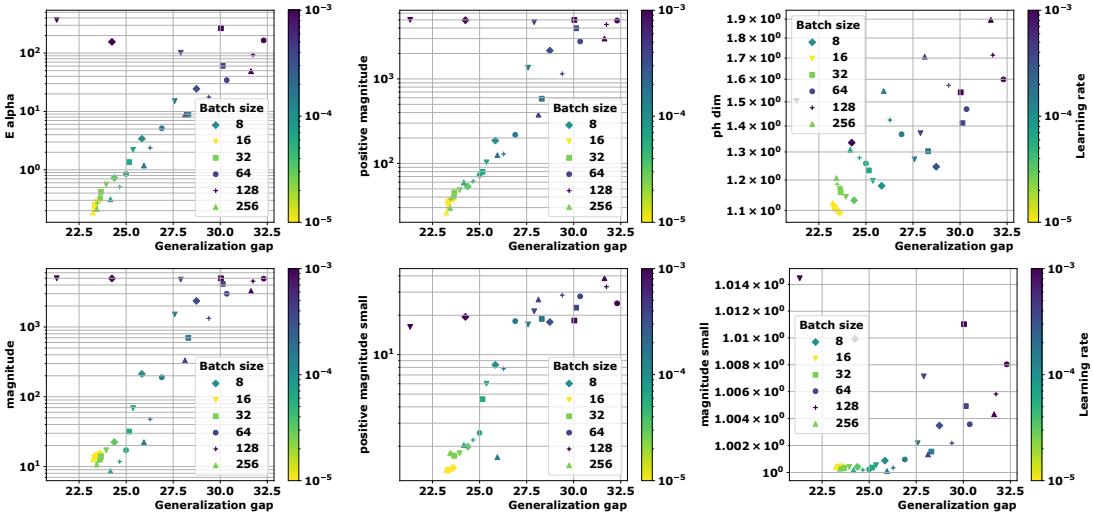


Figure A.26: Swin on CIFAR100 with ρ_S-pseudometric.

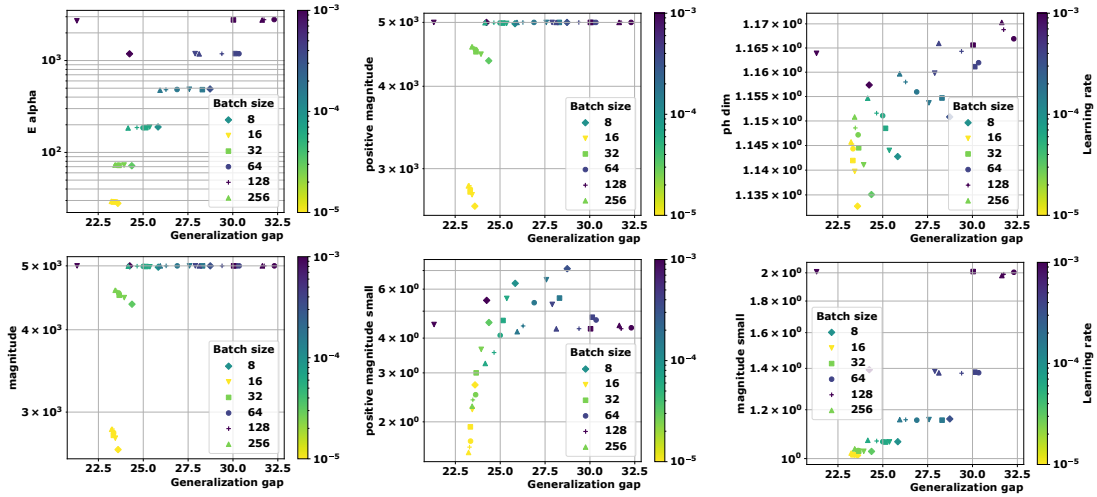


Figure A.27: Swin on CIFAR100 with $\|\cdot\|_2$.

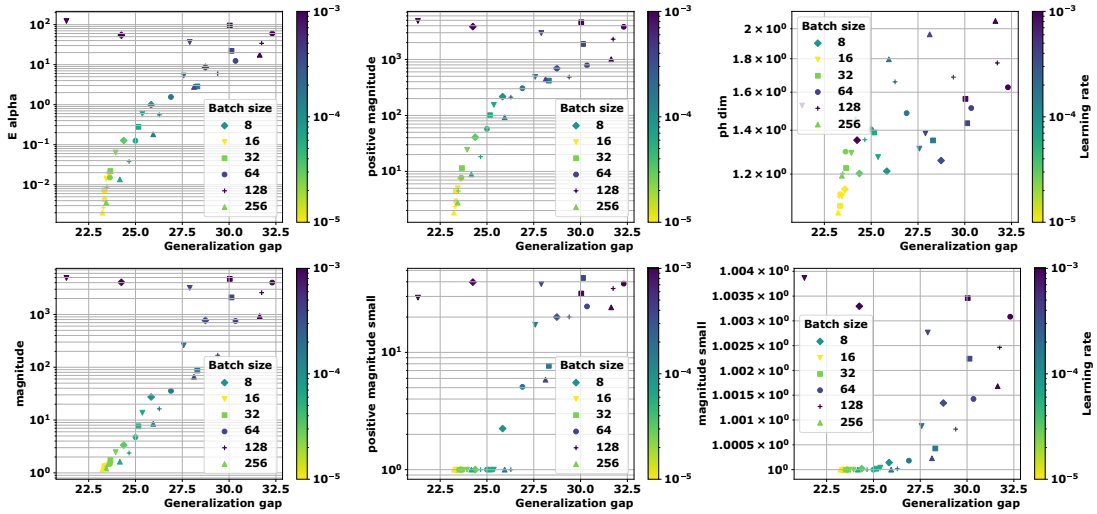


Figure A.28: Swin on CIFAR100 with 01.

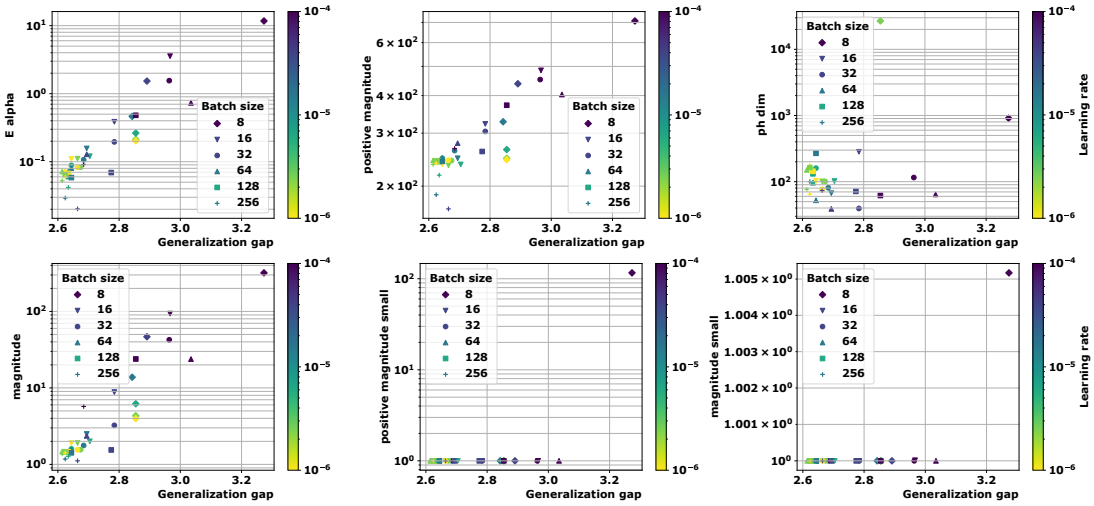


Figure A.29: GraphSage on MNIST with ρ_S-pseudometric.

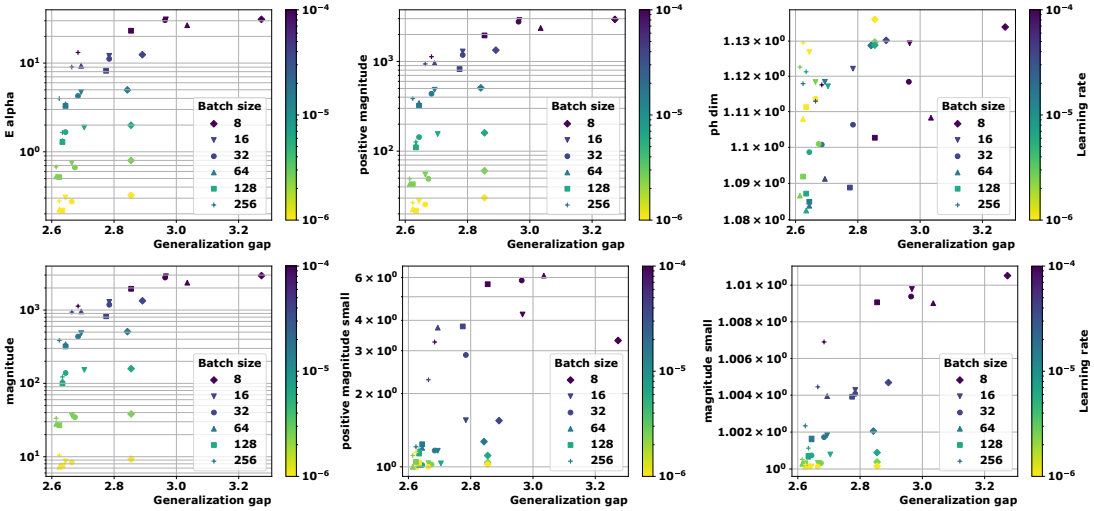


Figure A.30: GraphSage on MNIST with $\|\cdot\|_2$.

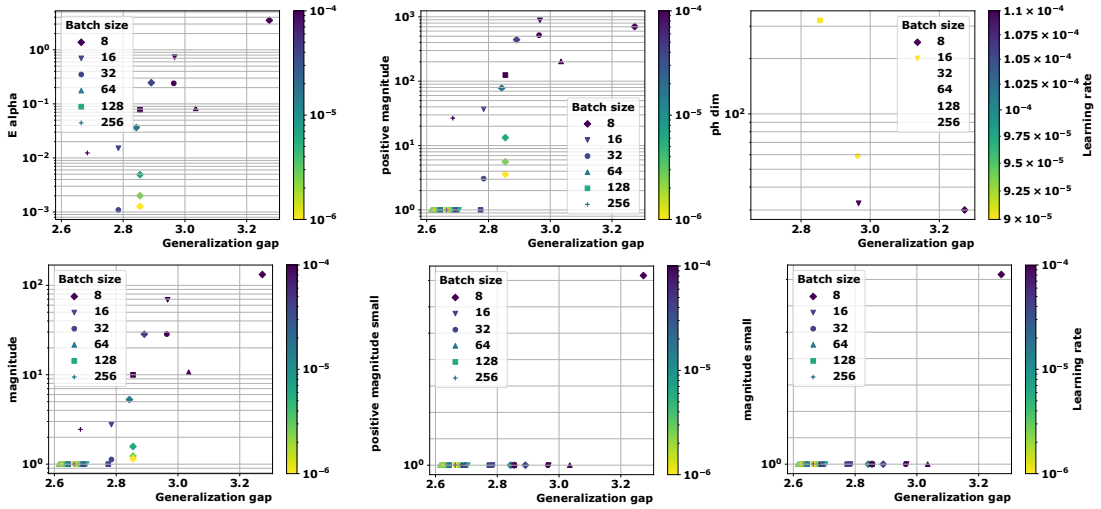


Figure A.31: GraphSage on MNIST with 01.

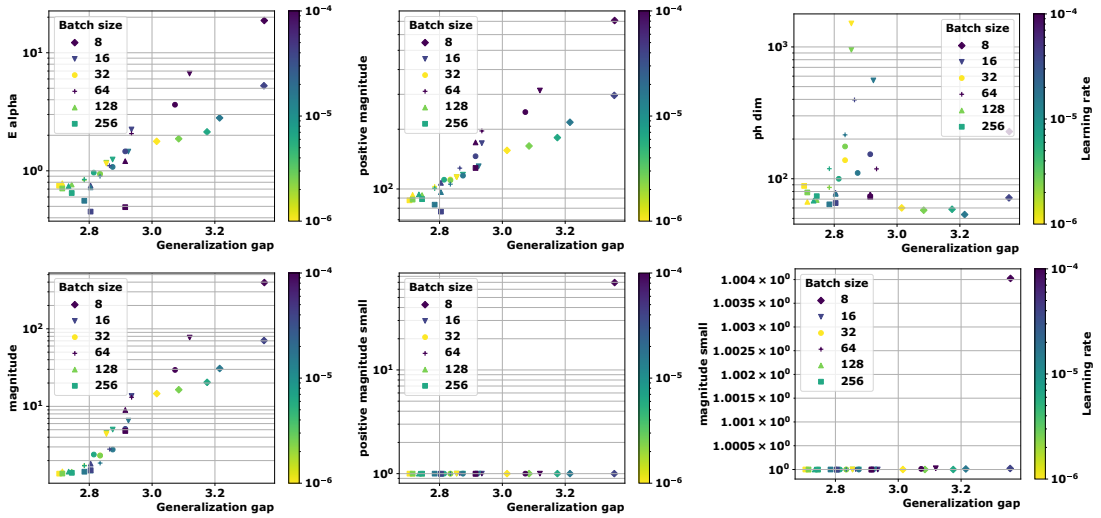


Figure A.32: GatedGCN on MNIST with ρ_S-pseudometric.

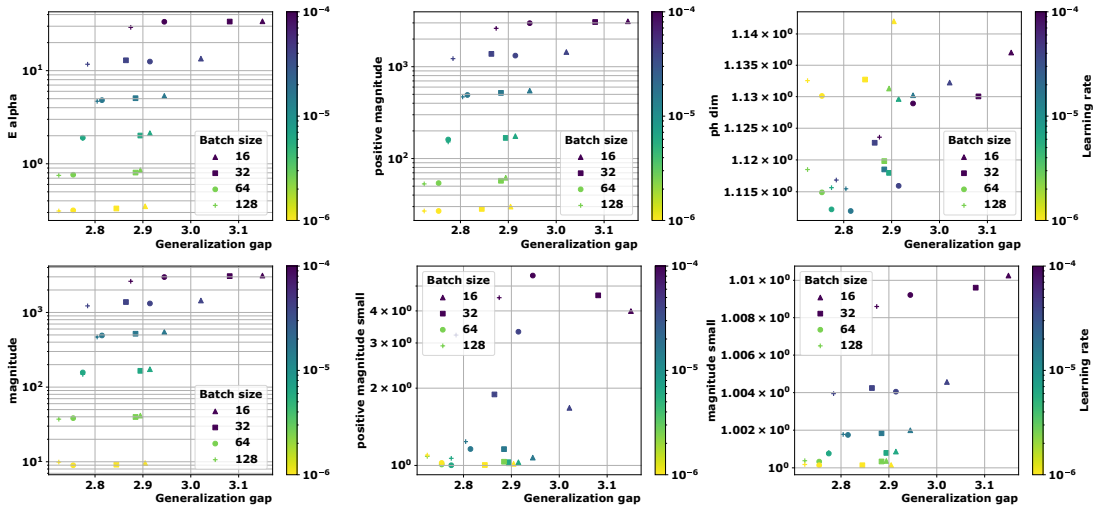


Figure A.33: GatedGCN on MNIST with $\|\cdot\|_2$.

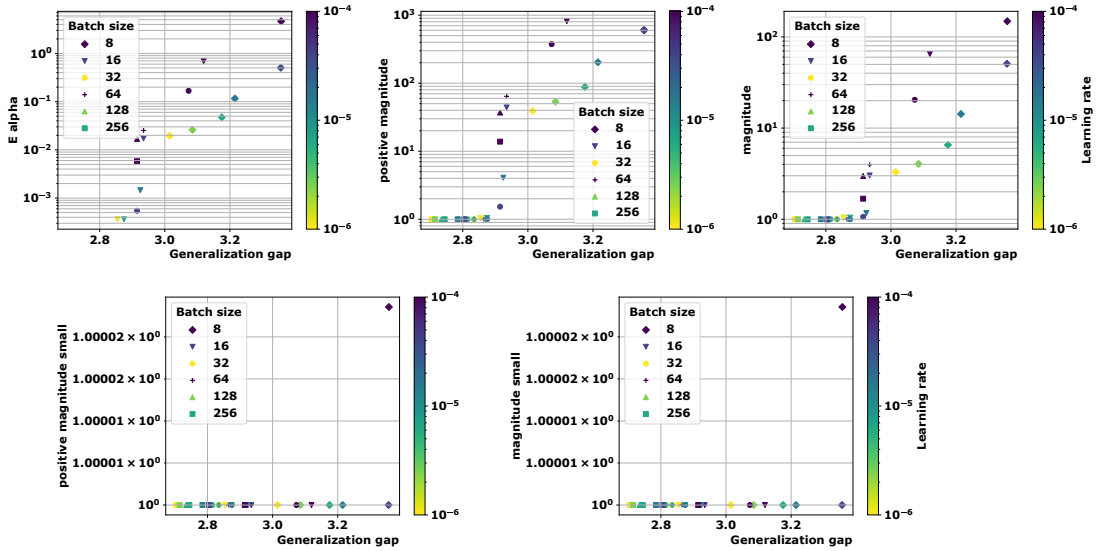


Figure A.34: GatedGCN on MNIST with 01.

A.4 Appendix for Chapter 6

A.4.1 Stability Proof

Next to the theoretical properties linking magnitude to geometrical properties of a space, which we previously outlined, we further prove that magnitude, as a metric space invariant, also satisfies properties that are advantageous in the setting of analysing latent representations. Specifically, we prove that magnitude and thus the proposed magnitude differences satisfy certain *stability properties* in light of perturbations of metric space. By this, we mean that if two metric spaces X, Y are *close*, we want to obtain bounds on the differences between their magnitude values. The canonical choice to measure closeness would be the Gromov–Hausdorff distance, but in the absence of strong results concerning the behaviour of magnitude under this distance [Leinster, 2013], we resort to a more general—but also weaker—notion of similarity in terms of *continuity*. More precisely, we will show that the similarity matrices used in the calculation of magnitude are well-behaved in the sense that closeness of metric spaces (under some matrix norm) translates to a continuous bound on the variation of the similarity matrices. We first prove a general result about matrices and their associated transformations.

Lemma A.4.1. *Let* $\|A\|_2 := \sup\{\|Ax\|_2 : x \in \mathbb{R}^n \text{ with } \|x\|_2 = 1\}$ *refer to the* induced 2-norm for matrices*, and let* A, B *be two* $n \times n$ *matrices with* $\|A - B\|_2 \leq \epsilon$*. Moreover, let* $f(M) := \mathbb{1}^\top M \mathbb{1}$*. Then* $\|f(A) - f(B)\|_2 \leq n\epsilon$*.*

Proof. Because $\|\cdot\|_2$ is a *consistent* norm, we have $\|f(M)\|_2 \leq \|\mathbb{1}^\top\|_2 \|M\|_2 \|\mathbb{1}\|_2 = n\|M\|_2$ for all $n \times n$ matrices M. Without loss of generality, assume that $\|f(A)\|_2 \geq \|f(B)\|_2$ and $\|A\|_2 \geq \|B\|_2$. Thus, $\|f(A)\|_2 - \|f(B)\|_2 \leq d(\|A\|_2 - \|B\|_2) \leq d(\|A - B\|_2) = n\epsilon$. □

Treating A, B as inverse similarity matrices, the preceding statement shows that if the two inverse similarity matrices are close with respect to their spectral radius, the difference between their magnitude can be bounded. The following lemma shows that the similarity matrices satisfy a general continuity condition.[4]

[4]It is clear that the mapping itself is continuous because of the functions involved in its calculation. However, we find it important to remark on the bound obtained with respect to the *spectral norm* of the two similarity matrices.

Lemma A.4.2. *Let (X, d_X) and (Y, d_Y) be two metric spaces with corresponding distance matrices D_X, D_Y and cardinality n. For all $\epsilon > 0$, there exists $\delta > 0$ such that if $|D_X - D_Y| < \delta$ holds elementwise, then $\|\zeta_X - \zeta_Y\|_2 \leq \epsilon$.*

Proof. As a consequence of the continuity of the exponential function, we know that there is δ such that $|\zeta_X - \zeta_Y| < n^{-1}\epsilon$. The row sums of $\zeta_X - \zeta_Y$ are therefore upper-bounded by ϵ. We thus have $\|\zeta_X - \zeta_Y\|_2 \leq \epsilon$ [Minc, 1988, Theorem 1.1, p. 24]. □

As a consequence of Lemma A.4.2, and the continuity of matrix inversion, we know that magnitude is well-behaved under small perturbations of the respective distance matrices. Given a pre-defined threshold ϵ, we can always find perturbations that preserve the magnitude difference accordingly. Notice that this result does not make any assumptions about the Gromov–Hausdorff distance of the metric space and only leverages the distance matrices themselves. Moreover, this result applies in case X, Y are close with respect to the *Hausdorff distance*. If $d_H(X, Y) < \delta$, the elementwise condition $|D_X - D_Y| < \delta$ is satisfied *a fortiori*. This stability of single-scale magnitude then further ensures the stability of the difference between magnitude functions as defined in 6.3.2 in the same sense. Nevertheless, from a theoretical point of view, this result could be made stronger by showing bounds in terms of distances between the metric spaces. We leave such a result for future work, noting in passing that such strong results remain elusive at the moment [Govc and Hepworth, 2021]; it is known, however, that the magnitude function is at least *lower semicontinous* [Meckes, 2013, Theorem 2.6].

A.4.2 Empirical Stability

We further investigate the empirical stability of the magnitude function difference. Given the difficulty in proving strong theoretical stability results, we verify that, in practice, the magnitude function difference remains stable when adding noise to the input space. We thus sample points from a Laplace distribution with mean $\mu = 0$ and variance $2b^2$ with different levels of noise, i.e. $b \in \{0.0001, 0.001, 0.005, 0.01, 0.05\}$. Figure A.35 depicts the errors in magnitude function difference relative the the area under the magnitude function of the unperturbed data across three different datasets (circles, Swiss Roll, Gaussian blobs), using a different number of samples (varying between 100 and 5000 across 50 repetitions). The bound of 5000 points has been chosen given the clear downwards

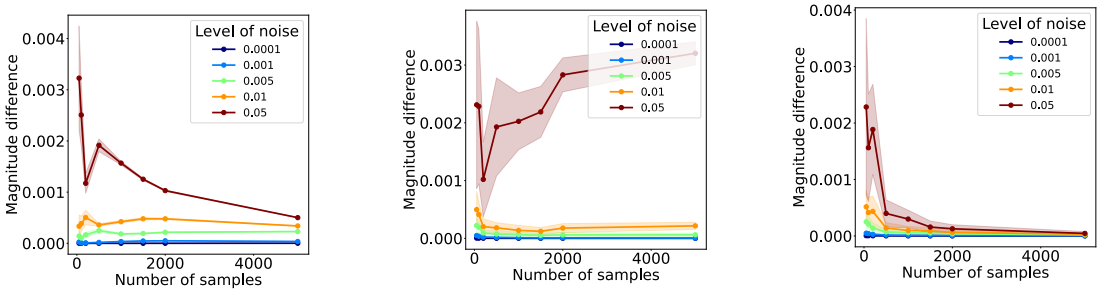


Figure A.35: **Empirical stability of magnitude.** Magnitude difference is stable across different datasets (from left to right: Circles, Swiss Roll, Gaussian blobs) and sample sizes. The lines show the mean magnitude difference relative to the magnitude area of the unperturbed data and the shaded area the standard deviation calculated across 50 repetitions.

trend across multiple noise levels; we expect the same trend to hold for larger sample sizes. We observe that the magnitude function difference does not increase above the value of 1×10^{-3} with increasing sample size. In fact, the difference fluctuates more for smaller number of points, but this is still within a very small range. We therefore conclude that the magnitude function difference between the original space and its noisy version does not change much, which indicates that our measure is reliable and stable across multiple experimental conditions.

A.4.3 Definitions of Intrinsic Diversity Measures

The difficulty in defining diversity in representation learning has led to a few varying proposals for evaluating the intrinsic diversity of latent representations. Amongst these we consider the following three methods as baseline measures:

GMStds:. For a X, a D-dimensional embedding, it is directly computed as

$$\text{GMSTDS} = \sqrt[D]{\prod_{i=j}^{D} \sigma_j} \tag{xiii}$$

where $\sigma_j = \sqrt{\frac{1}{n}(\sum_{i=1,..,n} x_{ij} - \hat{x_j})^2}$ is the standard deviation across the j-th embedding dimension [Lai et al., 2020]. Thus, GMSTDS regards an embedding as a cluster and assessing diversity by quantifying its spread.

AvgSim:. Average mean similarity (or variations of it) is the most frequently used diversity measure in ML. It is simply computed as

$$\textsc{AvgSim} = \frac{1}{\binom{n}{2}} \sum_{i,j \leq n, j > i} \zeta(i,j) \tag{xiv}$$

across all distinct pairs of points in X assuming ζ is symmetric [Tevet and Berant, 2021]. This approach simply summarises that in a more diverse space, observations should on average be less similar.

Vendi Score (VS):. We also consider the Vendi Score, which is the only entropy-based diversity measure proposed in related ML literature. Let ζ be a positive semi-definite similarity matrix with $\zeta(i,i) = 1$ for all $i \leq n$. Compute λ_i, the eigenvalues of ζ/n. Then the Vendi Score is defined as

$$\text{VS} = \exp(-\sum_{i=1}^{n} \lambda_i \log(\lambda_i)) \tag{xv}$$

taking $0 \log(0) = 0$ by convention. That is, the Vendi Score is the exponential of the Shannon entropy of the eigenvalues of ζ/n [Friedman and Dieng, 2023]. It can thus be interpreted as summarising the effective number of modes in a space at a specific scale of similarity.

A.4.4 Computing Magnitude

A naïve calculation of magnitude according to 2.1.1 requires inverting the similarity matrix ζ_X, which has a worst-case complexity of $\mathcal{O}(n^3)$ and is numerically unstable. However, inverting ζ_X is not required in practice; instead, it suffices to solve certain *linear equations* as also pointed out by Huntsman [2022]. First, we notice that the calculation of magnitude can be written as $\text{Mag}(X) := \mathbb{1}^\top \zeta_X^{-1} \mathbb{1}$. For finite metric spaces and negative definite metrics, ζ_X is a *symmetric positive definite matrix*, thus affording a *Cholesky decomposition*, which factorises $\zeta_X = LL^\top$, with L being a *lower triangular matrix*. This operation is numerically stable and more efficient than matrix inversion [Higham, 2009]. We thus have $\text{Mag}(X) := \mathbb{1}^\top \zeta_X^{-1} \mathbb{1} = \mathbb{1}^\top (LL^\top)^{-1} \mathbb{1} = (L^{-1}\mathbb{1})^\top (L^{-1}\mathbb{1})$. This is equivalent to calculating $x^\top x$ with $x = L^{-1}\mathbb{1}$, which we can efficiently obtain by solving $Lx = \mathbb{1}$ since L is lower triangular. Likewise, we can reformulate the calculation of the *magnitude weight vector* $w_X = \zeta_X^{-1}\mathbb{1}$ as solving $\zeta_X w_X = \mathbb{1}$, which also benefits from the Cholesky factorisation.

A.4.5 Additional Details for Our Experiments

In the following we give further details and elaborate on the experimental setup and datasets used for our experiments as well as showcase extended results.

A.4.5.1 Curvature Experiments

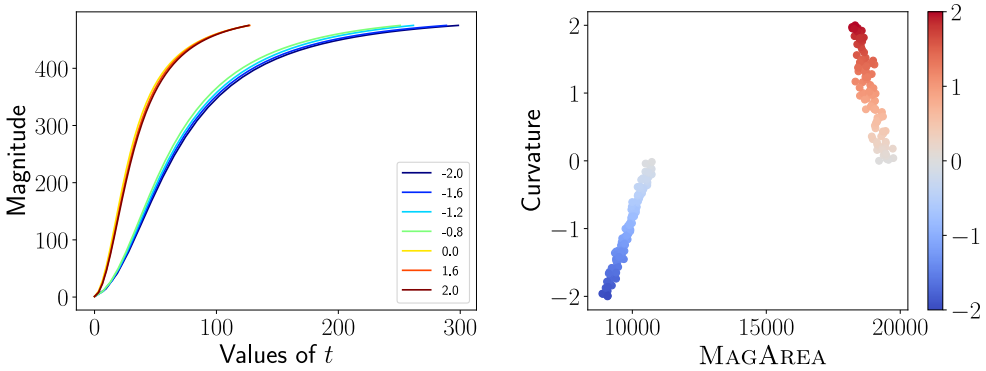


Figure A.36: **Magnitude detects curvature.** Left: Magnitude functions for unit disks with varying curvature between $[-2, 2]$. Right: Curvature is positively correlated with MagArea, indicating that it serves as a expressive predictor.

Here we provide more details about the curvature experiments, which builds on the approach by Turkes et al. [2022][5]. We generate a unit disks D_κ of surfaces of constant curvature κ, with 3 cases: the first one is when $\kappa = 0$ (we then have the Euclidean plane), $\kappa < 0$ (we have a space of negative curvature, the Poincare disk model of the hyperbolic plane), $\kappa > 0$ (sphere with radius $1/\sqrt{\kappa}$). We vary the curvature κ to be in the interval $[-2, 2]$. For each value of κ, we construct point clouds by sampling 500 points from D_κ. We generate 201 surfaces with equally spaced curvature in the interval $[-2, 2]$. Then, we compute magnitude for each space using Euclidean distance and 30 evenly spaced intervals until the scale $t_{\text{cut}} = 73$. For the results reported in Table 6.1 we further apply 5-fold cross-validation. We first train a quantile regression model on the MagArea after applying polynomial feature transformation of degree 2 to the training data suspecting a quadratic-looking relationship between MagArea and curvature after exploratory analysis. Further, we compare this to piecwise linear regression with two breakpoints under the assumption that the relationship between

[5]The code by Turkes et al. [2022] is available at `https://github.com/renata-turkes/turkevs2022on`.

MagArea and curvature as plotted in Figure A.36 rather depicts a piecewise linear relationship clearly separating spaces of positive and negative curvature. We further report six alternative models from Turkes et al. [2022], which are using features from persistent homology (PH) summarising persistence diagrams (PDs). See Bubenik et al. [2020] for a more detailed explanation on PH and its relationship to curvature. Specifically, in Table 6.1 we reproduce the following models from Figure 4. and Table 3. of Turkes et al. [2022]:

- SVR (all PH features) referred to as 0-dim PH simple by Turkes et al. [2022], which uses the lifespans of the persistence diagram computed on the samples;
- SVR (selected PH features) denoted 0-dim PH simple 10 by Turkes et al. [2022], which uses the 10 longest lifespans; and
- SVR (PH vectorisation) corresponding to 0-dim PH by Turkes et al. [2022], which selects the best PD vectorisation amongst a number of options, namely persistence images (PI) or persistence landscapes (PL).

All the PH-based methods use support vector regression (SVR) with a RBF kernel. Hyperparameter tuning for these models is conducted as reported by Turkes et al. [2022] using grid search with a choice of C parameters in $\{0.001, 1, 100\}$. We further reproduce 1 method based on pairwise distance matrices:

- SVR (distance matrices) denoted as ML by Turkes et al. [2022].

Finally, we restate the performance scores of these two methods directly from Turkes et al. [2022]:

- MLP (shallow) denoted as NN shallow by Turkes et al. [2022]; and
- MLP (deep) denoted as NN deep by Turkes et al. [2022].

We also note that the other models achieve different performance scores on our dataset than reported by Turkes et al. [2022] due to a slight difference in dataset and cross-validation splits. We use a smaller subset of samples than Turkes et al. [2022] each having a unique curvature value as described above, and ensure that all models are evaluated on the same splits of data across 5-fold CV for fair comparison. Finally, we summarise the MSE achieved by each model in Table 6.1. Illustrating this, Figure A.36 further shows examples of both magnitude functions for negative and positive curvature as well as the clear piecewise-linear trend between MagArea and curvature.

A.4.5.2 Graph Embedding Experiments

To assess our new diversity measure's utility for graph generative model evaluation we reproduce the benchmark by Thompson et al. [2022][6] and include our proposed reference-based diversity measure, MagDiff, to the diversity evaluation benchmark.

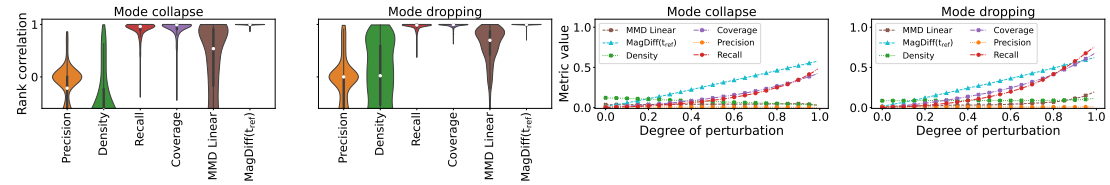
Specifically, we conduct this experiment on five graph datasets:

- **Lobster:** A dataset consisting of 100 stochastic graphs generated so that each node is at most 2 hops removed from a backbone path and the number of vertices varies between 10 and 100.[Dai et al., 2020, Thompson et al., 2022]
- **Grid:** A dataset of 100 two dimensional graphs consisting of 100 to 400 vertices [Dai et al., 2020, Liao et al., 2019, You et al., 2018, Thompson et al., 2022].
- **Proteins:** A dataset of 918 protein networks. Each vertex is an amino acid and edges connect amino acids that are less than 6 Angstroms away from each other [Dobson and Doig, 2003]. Only graphs with between 100 to 500 vertices are selected [Dai et al., 2020, Liao et al., 2019, You et al., 2018, Thompson et al., 2022].
- **Ego:** A dataset of 757 graphs that are 3-hop networks with 50 to 399 vertices [You et al., 2018, Thompson et al., 2022]. These graphs were extracted from the CiteSeer citation network where nodes represent documents [Sen et al., 2008].
- **Community:** A dataset with 500 two-community graphs with between 60 to 160 vertices, where each community has been generated using the Erdős-Rényi model [Erdős et al., 1960] setting n equal to half the number of vertices and $p = 0.3$. Additional edges amounting to 5% of the number of vertices have been added to each graph with uniform probability [You et al., 2018, Thompson et al., 2022].

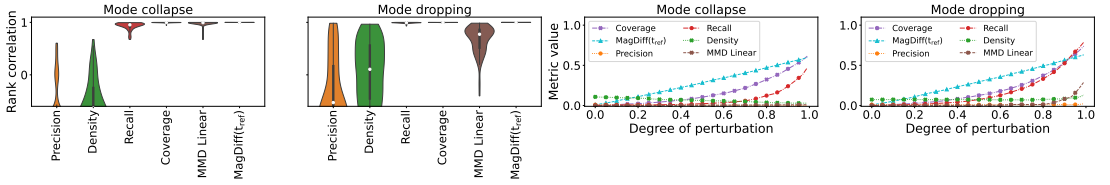
Further, this experiment uses a GIN (Graph Isomorphism Network) [Xu et al., 2019a] architectures as an embedding model and following the procedure by [Thompson et al., 2022] we vary the following hyperparameters for these models: We vary the number of layers between $[2, 3, \ldots, 7]$ and vary the hidden dimensions in the interval $[5, 30]$ with an increment of 5 resulting in a total of 36 architectures.

[6]Code for reproducing this graph evaluation benchmark is available at `https://github.com/uoguelph-mlrg/GGM-metrics` under an MIT licence.

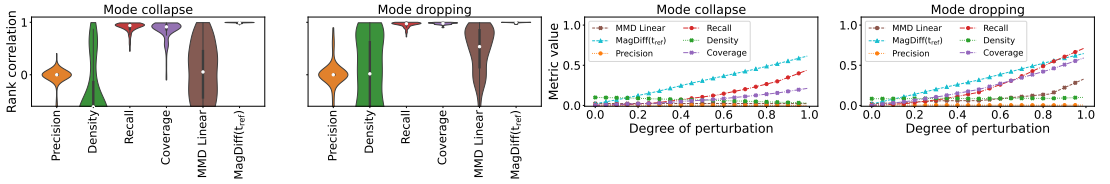
We repeat the experiments for 5 different random seeds. The experimental setup used to evaluate the evaluation metrics for both mode collapse and mode dropping then is as follows: First $P_r \approx P_g$, so that P_r, the real distribution, is identical to P_g, the generated distribution. Then the perturbation parameter $p \in [0, 1]$ is introduced, which transforms the generated graph datasets step-wise and increases the dissimilarity (and hence diversity) between the reference and generated datasets. Therefore, we use it as a proxy to measure the difference in diversity between P_r and P_g. To evaluate this decrease in diversity, we compute magnitude for the corresponding graph embeddings across 40 evenly-spaced scales until the convergence scale of the reference choosing $\epsilon = 0.05|X|$. For precision, recall, density and coverage we take the parameter $k = 5$, as proposed previously by [Naeem et al., 2020], to ensure a fair comparison. We then normalise all metrics such that their value is 0 when $P_r = P_g$ (which is exactly when the degree of permutation is 0). For this, we follow the normalisation strategy by [Thompson et al., 2022] and normalise MagDiff by the cardinality of each embedding. Next, we vary the parameter p and compute each evaluation metric. We report the Spearman correlation coefficient between each metric and the degree of the perturbation p. Hence, the value of a metric which captures the decrease in diversity accurately should increase with the increase of p, and rank correlation of 1 corresponds to an ideal metric. Results for the whole experiment across all datasets are presented in Figure A.37. The violin and boxplots reported in this figure then summarise the distribution of each evaluation measures rank correlation to the degree of perturbation across the 5 random seeds and the aforementioned hyperparameter choices influencing the embedding models. Finally, Figure A.38 investigates the influence the choice of convergence scale has on the results of these experiments and we observe that low values of ϵ lead to better agreement with the true degree of perturbation. Further, the trends in the value of MagDiff are stable across choices of $\epsilon \leq 0.05|X|$ as chosen throughout this study.



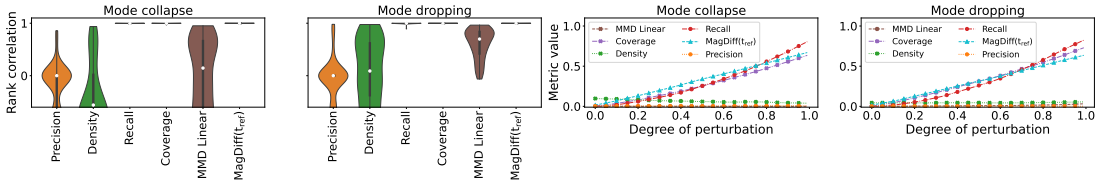
(a)



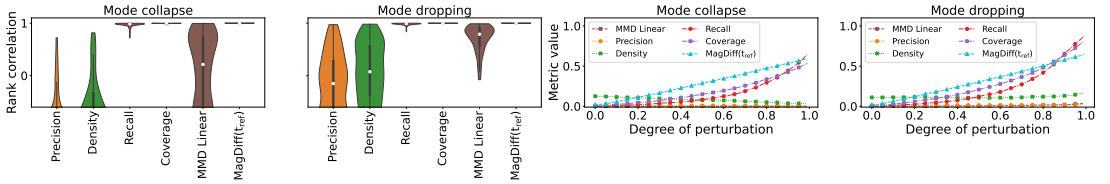
(b)



(c)

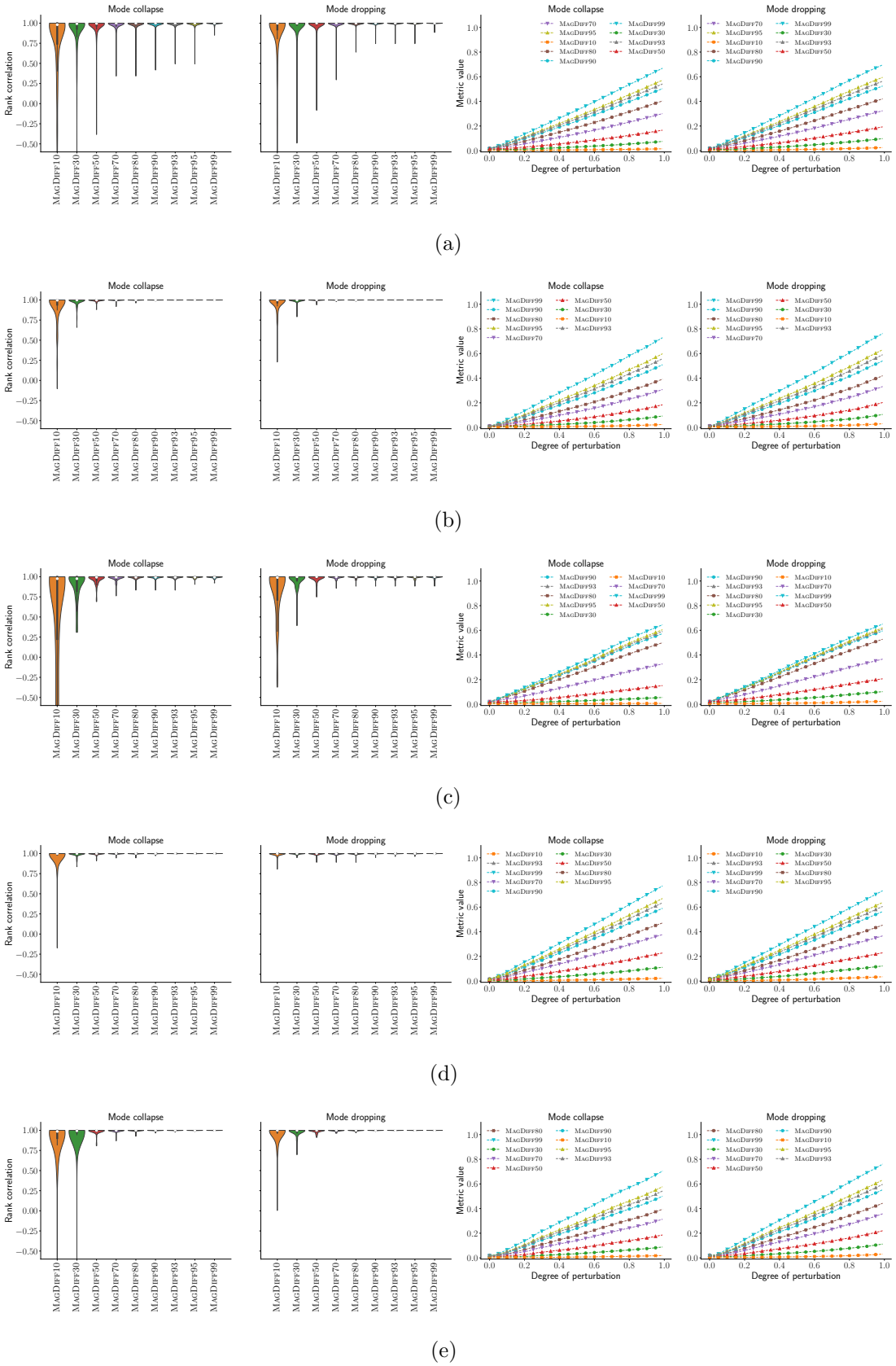


(d)



(e)

Figure A.37: **Results for the mode collapse and mode dropping experiments**. The patterns for each of the datasets is similar to the results on the Lobster graphs, which we show in the paper. (a) Results for all datasets, (b) Protein dataset, (c) Grid dataset, (d) Community dataset, (e) Ego dataset.



(a)

(b)

(c)

(d)

(e)

Figure A.38: **Rank correlation between MagDiff and the degree of perturbation for different choices of convergence scale.** Here we vary the choice of ϵ influencing the reference scale that is chosen to compute MagDiff$(|X|-\epsilon/|X|)$ for the mode collapse and mode dropping experiments. We clearly observe that low values of ϵ as given by MagDiff95 or MagDiff99 lead to higher rank correlation and better agreement with the true decrease in diversity. (a) Results for all datasets, (b) Protein, (c) Grid, (d) Community, (e) Ego.

A.5 Appendix for Chapter 7

Feature type	**Feature name**	**Description**	**Intuitive interpretation**
Baseline features	Radius	The radius of the papillae	The horizontal distance between the projection of the top and the border
	Height	The height of the papillae	The vertical distance between the top of the papillae and the base
Geometric features	Minimum Gaussian	The minimum value of the Gaussian curvature	How pointy downwards an object is
	Maximum Gaussian	The maximum value of the Gaussian curvature	How pointy the object is at its maximum
	Ratio Gaussian	The ratio with (+) $k_{Gaussian}$ over (-) $k_{Gaussian}$, if # of (+) $\leq$ # (-); and the other way round	Approximately measuring the papilla curves towards the normal
	Ratio mean	The ratio with (+) k_{mean} over (-) k_{mean}, if # of (+) $\leq$ # of (-); and the other way round	Approximately measuring how positively curved the papilla is
	Positive Gaussian	The percentage of points with positive Gaussian curvature	A measure of how positively curved the papilla is, what percentage is dome-like
	Positive mean	The percentage of points with positive mean curvature	The percentage of points with positive mean curvature
	Magnitude	The value of magnitude at $t = 0.25$	The number of effective points for a papilla
Topological features	Persistent entropy (0)	The Shannon entropy of the barcode in H_0	Measuring how different the lengths of the bars are in H_0
	Short bars (0) and (1)	The number of short bars in H_0 and H_1, respectively	The number of least persistent connected components in H_0 and loops in H_1 with relatively short life span
	Amplitude (Bottleneck) (0) and (1)	The distance between the persistence diagram and the empty diagram in the Bottleneck metric	Approx. measurement of the length of the longest bar, or the life span of the most persistent feature
	Amplitude (Persistence image)	The distance between the persistence image and the empty diagram	Quantity measuring the topology of the object and how much it differs from a flat surface

Table A.7: Description of non-correlated baseline, geometric and topological features.

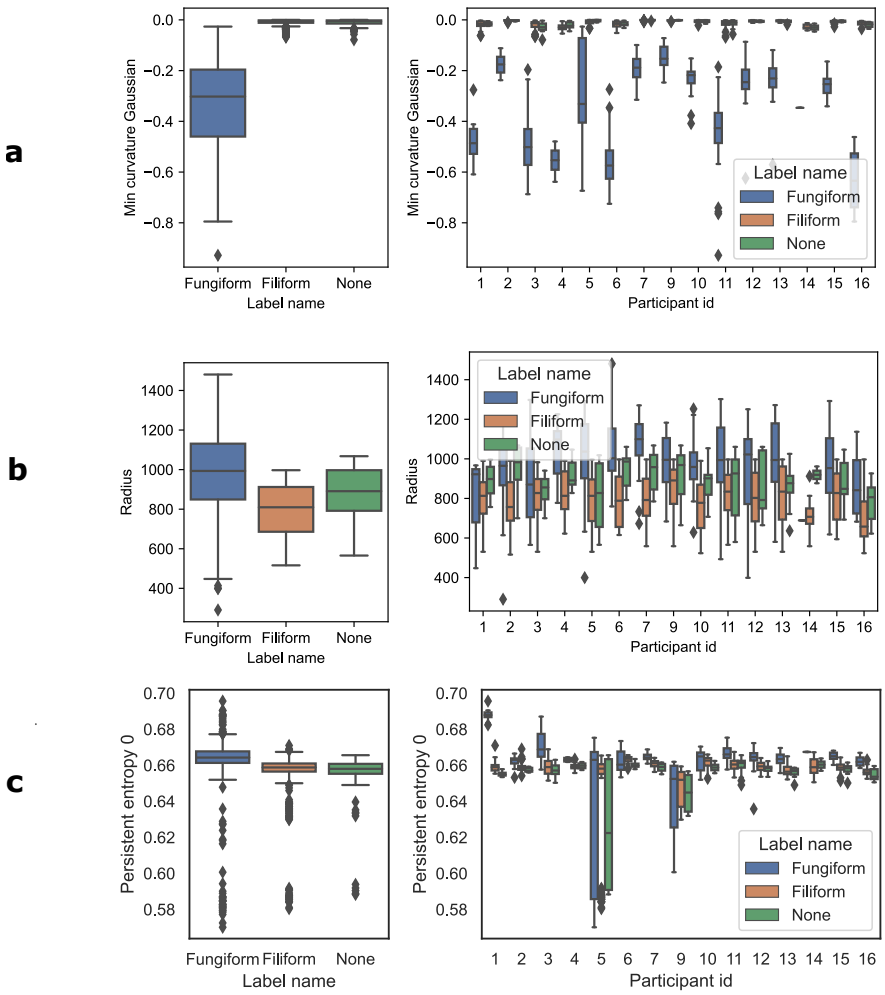


Figure A.39: **The most important features for papillae type classification** (a-c) are the features with the highest importance for the papillae type classification task. (a) and (c) are topological, while (b) is curvature.

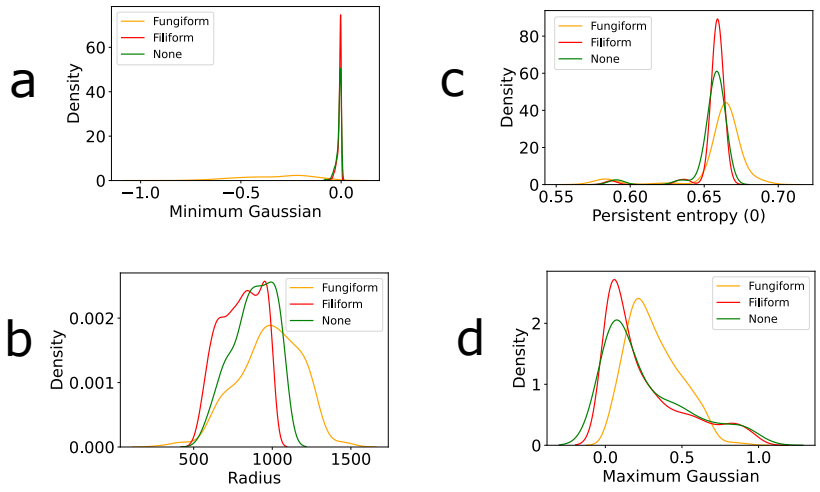


Figure A.40: **KDE plots of most important features for type classification task**. In plot (a), the distribution of the Minimum Gaussian for Fungiform follows a very different pattern from Filiform and None – it is mostly flat and evenly distributed on the interval $(-1, 0)$, while Filiform and None are densely concentrated around 0. In plot (b), the distribution of Filiform and None are very similar, with Fungiform having higher value of Radius, which is as expected from previous work [Andablo-Reyes et al., 2020]. In plot(c), the papillae Types Filiform and None follow a similar pattern to (a). Fungiform has higher value of Persistent entropy (0). In plot (d), even though Maximum Gaussian is not amongst the top three most important features, the value for Maximum Gaussian is higher for Filiform and None compared to Fungiform. This is as expected due to the sharper shape of filiform with more pronounced drop and steeper sides.

Participant id	Segments
1	173
2	163
3	168
4	149
5	199
6	138
7	139
9	115
10	77
11	240
12	132
13	106
14	51
15	121
16	121

Table A.8: Number of segments per participant.

Gender	Age (mean;SD)
Female	29.5 (4.5)
Male	28.3 (3.9)

Table A.9: Demographics of the participants.

Model	**Balanced acc(Gender, all)**	**Balanced acc(Gender, lim)**	**Balanced acc(Age)**	**Balanced acc(Age,lim)**
Baseline features	0.45 ± 0.12	0.52 ± 0.06	0.52 ± 0.11	0.52 ± 0.11
Curvature features	0.44 ± 0.25	0.67 ± 0.15	0.57 ± 0.30	0.59 ± 0.28
Topological features	0.45 ± 0.29	$\mathbf{0.67 \pm 0.11}$	0.57 ± 0.30	$\mathbf{0.61 \pm 0.27}$
All Combined	$\mathbf{0.42 \pm 0.28}$	0.65 ± 0.14	0.50 ± 0.24	0.53 ± 0.21

Table A.10: **Balanced accuracies for age and gender tasks.** The performance when we use Leave-one-group-out approach. The results are worse than before due to a small number of participants having too low accuracies. It will be a question for future work to investigate this. The results on the right are when these small number of participants have been removed. It can be that they are outliers and their topological features do. However, it is difficult to make conclusions given the small sample size. It could be that some people are topological outliers, and their features are not similar at all to the other people in the same age category

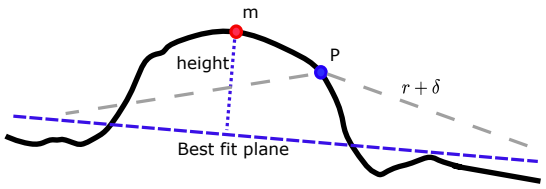


Figure A.41: **Schematic figure: Profile view of processing a segment with a fungiform papilla.** From an arbitrary point P, all mesh vertices within a radius $r + \delta$ are taken. Then a Best fit Plane is found using the RANSAC algorithm. The candidate point for the peak of a papilla (if present) is found as m – the point furthest from the plane. This distance is taken to be the height, and m is assumed to be the centre of the papilla.

Bibliography

Yongqiang Ma, Jiawei Liu, Fan Yi, Qikai Cheng, Yong Huang, Wei Lu, and Xiaozhong Liu. Ai vs. human–differentiation analysis of scientific content generation. *arXiv preprint arXiv:2301.10416*, 2023.

Partha Pratim Ray. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, 3:121–154, 2023.

Václav Snášel, Jana Nowaková, Fatos Xhafa, and Leonard Barolli. Geometrical and topological approaches to big data. *Future Generation Computer Systems*, 67:286–296, 2017.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

Henry Adams and Michael Moy. Topology applied to machine learning: From global to local. *Frontiers in Artificial Intelligence*, 4:54, 2021.

Barbara Giunti, Jānis Lazovskis, and Bastian Rieck. DONUT: Database of Original & Non-Theoretical Uses of Topology, 2022. `https://donut.topology.rocks`.

Nina Otter. Magnitude meets persistence. homology theories for filtered simplicial sets. *Homology, Homotopy and Applications*, 2021.

Eric Bunch, Jeffery Kline, Daniel Dickinson, Suhaas Bhat, and Glenn Fung. Weighting vectors for machine learning: numerical harmonic analysis applied to boundary detection. *arXiv preprint arXiv:2106.00827*, 2021.

Michael F Adamer, Edward De Brouwer, Leslie O'Bray, and Bastian Rieck. The magnitude vector of images. *arXiv preprint arXiv:2110.15188*, 2021.

Ran Raz. On the complexity of matrix product. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 144–151, 2002.

Volker Strassen. Gaussian elimination is not optimal. *Numerische mathematik*, 13 (4):354–356, 1969.

Umut Simsekli, Ozan Sener, George Deligiannidis, and Murat A Erdogdu. Hausdorff dimension, heavy tails, and generalization in neural networks. *Advances in Neural Information Processing Systems*, 33:5138–5151, 2020.

Inglis J Miller Jr and Frank E Reedy Jr. Variations in human taste bud density and taste intensity perception. *Physiology & behavior*, 47(6):1213–1219, 1990.

Lei Cai, Jingyang Gao, and Di Zhao. A review of the application of deep learning in medical image classification and segmentation. *Annals of translational medicine*, 8(11), 2020.

Paul Bendich, James S Marron, Ezra Miller, Alex Pieloch, and Sean Skwerer. Persistent homology analysis of brain artery trees. *The annals of applied statistics*, 10(1):198, 2016.

Jose Gutierrez, Lawrence Honig, Mitchell SV Elkind, Jay P Mohr, James Goldman, Andrew J Dwork, Susan Morgello, and Randolph S Marshall. Brain arterial aging and its relationship to alzheimer dementia. *Neurology*, 86(16):1507–1515, 2016.

Rayna Andreeva, James Ward, Primoz Skraba, Jie Gao, and Rik Sarkar. Approximating metric magnitude of point sets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 15374–15381, 2025.

Rayna Andreeva, Katharina Limbeck, Bastian Rieck, and Rik Sarkar. Metric space magnitude and generalisation in neural networks. In *Proceedings of 2nd Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML)*, volume 221 of *Proceedings of Machine Learning Research*, pages 242–253. PMLR, 2023a.

Rayna Andreeva, Benjamin Dupuis, Rik Sarkar, Tolga Birdal, and Umut Simsekli. Topological generalization bounds for discrete-time stochastic optimization algorithms. *Advances in Neural Information Processing Systems*, 37:4765–4818, 2024.

Katharina Limbeck, Rayna Andreeva, Rik Sarkar, and Bastian Rieck. Metric space magnitude for evaluating the diversity of latent representations. *Advances in Neural Information Processing Systems*, 37:123911–123953, 2024.

Rayna Andreeva, Anwesha Sarkar, and Rik Sarkar. Machine learning and topological data analysis identify unique features of human papillae in 3d scans. *Scientific Reports*, 13(1):21529, 2023b.

Tom Leinster. The magnitude of metric spaces. *Documenta Mathematica*, 18: 857–905, 2013.

Mark W Meckes. Positive definite metric spaces. *Positivity*, 17(3):733–757, 2013.

Mark W Meckes. Magnitude, diversity, capacities, and dimensions of metric spaces. *Potential Analysis*, 42(2):549–572, 2015.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic Generalization Measures and Where to Find Them. *ICLR 2020*, December 2019.

Maurice G. Kendall. A new reasure of rank correlation. *Biometrika*, 1938a.

Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2): 81–93, 1938b.

Tom Leinster. The euler characteristic of a category. *Documenta Mathematica*, 13:21–49, 2008.

Tom Leinster and Simon Willerton. On the asymptotic magnitude of subsets of euclidean space. *Geometriae Dedicata*, 164:287–310, 2013.

Juan Antonio Barceló and Anthony Carbery. On the magnitudes of compact sets in euclidean spaces. *American Journal of Mathematics*, 140(2):449–494, 2018.

Tom Leinster. The magnitude of a graph. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 166, pages 247–264. Cambridge University Press, 2019.

Tom Leinster and Michael Shulman. Magnitude homology of enriched categories and metric spaces. *Algebraic & Geometric Topology*, 21(5):2175–2221, 2021.

Ryuki Kaneta and Masahiko Yoshinaga. Magnitude homology of metric spaces and order complexes. *Bulletin of the London Mathematical Society*, 53(3):893–905, 2021.

Chad Giusti and Giuliamaria Menara. Eulerian magnitude homology: subgraph structure and random graphs. *arXiv preprint arXiv:2403.09248*, 2024.

Michael F Adamer, Edward De Brouwer, Leslie O'Bray, and Bastian Rieck. The magnitude vector of images. *Journal of Applied and Computational Topology*, pages 1–27, 2024.

Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251–280, 1990. ISSN 0747-7171. doi: https://doi.org/10.1016/S0747-7171(08)80013-2. URL `https://www.sciencedirect.com/science/article/pii/S0747717108800132`. Computational algebraic complexity editorial.

Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3792–3835, 2024.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14:265–294, 1978.

Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability*, 3(71-104):3, 2014.

Jie Gao, Leonidas J Guibas, and An Nguyen. Deformable spanners and applications. *Comput. Geom.*, 35(1-2):2–19, August 2006.

Alex Krizhevsky, Vinod Nair, and Geoffrey E. Hinton. The cifar-10 dataset, 2014.

Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009. URL `https://api.semanticscholar.org/CorpusID:18268744`.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021a.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020.

Miguel O'Mally. *Magnitude, Alpha Magnitude, and Applications*. PhD thesis, Wesleyan University, 2023.

Miguel O'Malley, Sara Kalisnik, and Nina Otter. Alpha magnitude. *Journal of Pure and Applied Algebra*, page 107396, 2023.

Stephanie Chen and Juan Pablo Vigneaux. Categorical magnitude and entropy. In *International Conference on Geometric Science of Information*, pages 278–287. Springer, 2023.

Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Birkhäuser Basel, 2013. ISBN 0817649476.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Tolga Birdal, Aaron Lou, Leonidas J Guibas, and Umut Simsekli. Intrinsic dimension, persistent homology and generalization in neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

Benjamin Dupuis, George Deligiannidis, and Umut Şimşekli. Generalization bounds with data-dependent fractal dimensions. *arXiv preprint arXiv:2302.02766*, 2023.

Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, Nina Miolane, Aldo Guzmán-Sáenz, Karthikeyan Natesan Ramamurthy, Tolga Birdal, Tamal K. Dey, Soham Mukherjee, Shreyas N. Samaga, Neal Livesay, Robin Walters, Paul Rosen, and Michael T. Schaub. Topological deep learning: Going beyond graph data. *arXiv e-prints*, art. arXiv:2206.00606, 2022. doi: 10.48550/arXiv.2206.00606.

Felix Hensel, Michael Moor, and Bastian Rieck. A survey of topological machine learning methods. *Frontiers in Artificial Intelligence*, 4:681108, 2021.

Henry Adams, Manuchehr Aminian, Elin Farnell, Michael Kirby, Joshua Mirth, Rachel Neville, Chris Peterson, and Clayton Shonkwiler. A fractal dimension for measures via persistent homology. In *Topological Data Analysis: The Abel Symposium 2018*, pages 1–31. Springer, 2020.

Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In *International Conference on Machine Learning*, pages 5827–5837. PMLR, 2019.

Liam Hodgkinson and Michael Mahoney. Multiplicative noise and heavy tails in stochastic optimization. In *International Conference on Machine Learning*, pages 4262–4274. PMLR, 2021.

Michael Mahoney and Charles Martin. Traditional and heavy tailed self regularization in neural network models. In *International Conference on Machine Learning*, pages 4284–4293. PMLR, 2019.

German Magai and Anton Ayzenberg. Topology and geometry of data manifold in deep learning. *arXiv preprint arXiv:2204.08624*, 2022.

Andrew R Solow and Stephen Polasky. Measuring biological diversity. *Environmental and Ecological Statistics*, 1:95–103, 1994.

David Pérez Fernández, Asier Gutiérrez-Fandiño, Jordi Armengol-Estapé, and Marta Villegas. Determining structural properties of artificial neural networks using algebraic topology. *arXiv preprint arXiv:2101.07752*, 2021.

Bastian Rieck, Matteo Togninalli, Christian Bock, Michael Moor, Max Horn, Thomas Gumbsch, and Karsten Borgwardt. Neural persistence: A complexity measure for deep neural networks using algebraic topology. In *In-*

ternational Conference on Learning Representations (ICLR), 2019. URL https://openreview.net/forum?id=ByxkijC5FQ.

Pertti Mattila. *Geometry of sets and measures in Euclidean spaces: fractals and rectifiability*. Number 44. Cambridge university press, 1999a.

Facundo Mémoli and Kritika Singhal. A primer on persistent homology of finite metric spaces. *Bulletin of mathematical biology*, 81:2074–2116, 2019.

Herbert Edelsbrunner and John L Harer. *Computational topology: an introduction*. American Mathematical Society, 2022.

Gady Kozma, Zvi Lotker, and Gideon Stupp. The minimal spanning tree and the upper box dimension. *Proceedings of the American Mathematical Society*, 134 (4):1183–1187, 2006.

Benjamin Schweinhart. Persistent homology and the upper box dimension. *Discrete & Computational Geometry*, 65(2):331–364, 2021.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Richard C Bradley. On the ψ-mixing condition for stationary random sequences. *Transactions of the american mathematical society*, 276(1):55–66, 1983.

Simon Willerton. Heuristic and computer calculations for the magnitude of metric spaces. *arXiv preprint arXiv:0910.5500*, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90, 2017.

V Seshadri and Bruce J West. Fractal dimensionality of lévy processes. *Proceedings of the National Academy of Sciences*, 79(14):4501–4505, 1982.

Rickard Brüel-Gabrielsson, Bradley J Nelson, Anjan Dwaraknath, Primoz Skraba, Leonidas J Guibas, and Gunnar Carlsson. A topology layer for machine learning. *arXiv preprint arXiv:1905.12200*, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *ICLR 2017*, February 2017.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, February 2021. ISSN 0001-0782. doi: 10.1145/3446776.

Jing Xu, Jiaye Teng, Yang Yuan, and Andrew Yao. Towards Data-Algorithm Dependent Generalization: A Case Study on Overparameterized Linear Regression. *Advances in Neural Information Processing Systems*, 36:79698–79733, December 2023.

Jingwen Fu, Zhizheng Zhang, Dacheng Yin, Yan Lu, and Nanning Zheng. Learning Trajectories are Generalization Indicators, October 2023.

Michael Gastpar, Ido Nachum, Jonathan Shafer, and Thomas Weinberger. Fantastic generalization measures are nowhere to be found, 2023.

Kaifeng Lyu, Zhiyuan Li, and Sanjeev Arora. Understanding the Generalization Benefit of Normalization Layers: Sharpness Reduction, January 2023.

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks, May 2019.

Ahmed Imtiaz Humayun, Randall Balestriero, and Richard Baraniuk. Training Dynamics of Deep Network Linear Regions. https://arxiv.org/abs/2310.12977v1, October 2023.

Wenlong Mou, Liwei Wang, Xiyu Zhai, and Kai Zheng. Generalization Bounds of SGLD for Non-convex Learning: Two Theoretical Viewpoints. In *Proceedings of the 31st Conference On Learning Theory*. arXiv, July 2017. doi: 10.48550/arXiv.1707.05947.

Ankit Pensia, Varun Jog, and Po-Ling Loh. Generalization Error Bounds for Noisy, Iterative Algorithms. *2018 IEEE International Symposium on Information Theory (ISIT)*, January 2018.

Xuanyuan Luo, Luo Bei, and Jian Li. Generalization Bounds for Gradient Methods via Discrete and Continuous Prior, October 2022.

Gergely Neu, Gintare Karolina Dziugaite, Mahdi Haghifam, and Daniel M. Roy. Information-Theoretic Generalization Bounds for Stochastic Gradient Descent, August 2021.

Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. Topology of deep neural networks. *Journal of Machine Learning Research*, 21(184):1–40, 2020.

David Pérez-Fernández, Asier Gutiérrez-Fandiño, Jordi Armengol-Estapé, and Marta Villegas. Characterizing and measuring the similarity of neural networks with persistent homology. *arXiv preprint arXiv:2101.07752*, 2021.

Bastian Rieck, Matteo Togninalli, Christian Bock, Michael Moor, Max Horn, Thomas Gumbsch, and Karsten Borgwardt. Neural persistence: A complexity measure for deep neural networks using algebraic topology. *arXiv preprint arXiv:1812.09764*, 2018.

Satoru Watanabe and Hayato Yamana. Topological measurement of deep neural networks using persistent homology. *Annals of Mathematics and Artificial Intelligence*, 90(1):75–92, 2022.

German Magai. Deep neural networks architectures from the perspective of manifold learning. In *2023 IEEE 6th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, pages 1021–1031. IEEE, 2023.

Liam Hodgkinson, Umut Şimşekli, Rajiv Khanna, and Michael W. Mahoney. Generalization Bounds using Lower Tail Exponents in Stochastic Optimizers. *Proceedings of the 39th International Conference on Machine Learning*, July 2022.

Benjamin Dupuis, Paul Viallard, George Deligiannidis, and Umut Simsekli. Uniform generalization bounds on data-dependent hypothesis sets via pac-bayesian theory on random sets, 2024.

Kenneth Falconer. *Fractal Geometry - Mathematical Foundations and Applications - Third Edition*. Wiley, 2014.

Pertti Mattila. *Geometry of Sets and Measures in Euclidean Spaces*. Cambridge University Press, 1999b.

Milad Sefidgaran, Amin Gohari, Gaël Richard, and Umut Şimşekli. Rate-Distortion Theoretic Generalization Bounds for Stochastic Learning Algorithms, June 2022.

Milad Sefidgaran and Abdellatif Zaidi. Data-dependent Generalization Bounds via Variable-Size Compressibility, January 2024.

Alexander Camuto, George Deligiannidis, Murat A Erdogdu, Mert Gurbuzbalaban, Umut Simsekli, and Lingjiong Zhu. Fractal structure and generalization properties of stochastic optimization algorithms. *Advances in neural information processing systems*, 34:18774–18788, 2021.

Yakov B Pesin. *Dimension theory in dynamical systems: contemporary views and applications*. University of Chicago Press, 2008.

Sarah Sachs, Umut Şimşekli, and Tim van Erven. Generalization Guarantees via Algorithm-dependent Rademacher Complexity - preprint. *COLT 2023*, 2023.

Benjamin Schweinhart. Fractal dimension and the persistent homology of random geometric complexes. *Advances in Mathematics*, 372:107291, 2020.

Bobak T Kiani, Thien Le, Hannah Lawrence, Stefanie Jegelka, and Melanie Weber. On the hardness of learning under symmetries. *arXiv preprint arXiv:2401.01869*, 2024.

Arash Behboodi, Gabriele Cesa, and Taco S Cohen. A pac-bayesian generalization bound for equivariant networks. *Advances in Neural Information Processing Systems*, 35:5654–5668, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, June 2021.

M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734 vol. 2, 2005. doi: 10.1109/IJCNN.2005.1555942.

Tim van Erven and Peter Harremoës. Rényi Divergence and Kullback-Leibler Divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, July 2014. ISSN 0018-9448, 1557-9654. doi: 10.1109/TIT.2014.2320500.

Jean-Daniel Boissonat, Frédéric Chazal, and Mariette Yvinec. *Geometrical and Topological Inference*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2018.

Claire Donnat and Susan Holmes. Tracking network dynamics: a survey of distances and similarity metrics, 2018.

Julián Burella Pérez, Sydney Hauke, Umberto Lupo, Matteo Caorsi, and Alberto Dassatti. Giotto-ph: A Python Library for High-Performance Computation of Persistent Homology of Vietoris-Rips Filtrations, August 2021.

Ulrich Bauer. Ripser: Efficient computation of Vietoris-Rips persistence barcodes. *Journal of Applied and Computational Topology*, 5(3):391–423, September 2021a. ISSN 2367-1726, 2367-1734. doi: 10.1007/s41468-021-00071-5.

Shilan Salim. *The q-spread dimension and the maximum diversity of square grid metric spaces*. PhD thesis, University of Sheffield, 2021.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Number 47 in Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2020. ISBN 978-1-108-41519-4.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011a.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 32–42, 2021b.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023.

Dan Friedman and Adji Bousso Dieng. The Vendi Score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*, 2023.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations*, 2016.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*, 2020.

Guy Tevet and Jonathan Berant. Evaluating the evaluation of diversity in natural language generation. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 326–346. Association for Computational Linguistics, 2021.

Yi-An Lai, Xuan Zhu, Yi Zhang, and Mona Diab. Diversity, density, and homogeneity: Quantitative characteristic metrics for text collections. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1739–1746, 2020.

Wenchao Du and Alan W Black. Boosting dialog response generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International*

Conference on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, pages 7176–7185. PMLR, 2020.

Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

Loic Simon, Ryan Webster, and Julien Rabin. Revisiting precision recall definition for generative modeling. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5799–5808. PMLR, 2019.

Tom Leinster. *Entropy and Diversity: The Axiomatic Approach*. Cambridge University Press, 2021.

Steve Huntsman. Diversity enhancement via magnitude. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 377–390. Springer, 2023.

Aisling J Daly, Jan M Baetens, and Bernard De Baets. Ecological diversity: Measuring the unmeasurable. *Mathematics*, 6(7):119, 2018.

Aasa Feragen, Francois Lauze, and Soren Hauberg. Geodesic exponential kernels: When curvature and linearity conflict. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Emily Roff and Masahiko Yoshinaga. The small-scale limit of magnitude and the one-point property. *arXiv preprint arXiv:2312.14497*, 2023.

Renata Turkes, Guido F Montufar, and Nina Otter. On the effectiveness of persistent homology. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 35432–35448. Curran Associates, Inc., 2022.

Peter Bubenik, Michael Hull, Dhruv Patel, and Benjamin Whittle. Persistent homology detects curvature. *Inverse Problems*, 36(2):025008, 2020.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992. Association for Computational Linguistics, 2019.

Jeremy Wayland, Corinna Coupette, and Bastian Rieck. Mapping the multiverse of latent representations. *arXiv preprint arXiv:2402.01514*, 2024.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

Rylee Thompson, Boris Knyazev, Elahe Ghalebi, Jungtaek Kim, and Graham W. Taylor. On evaluation metrics for graph generative models. In *International Conference on Learning Representations*, 2022.

Leslie O'Bray, Max Horn, Bastian Rieck, and Karsten Borgwardt. Evaluation metrics for graph generative models: Problems, pitfalls, and practical solutions. In *International Conference on Learning Representations*, 2022.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019a.

Eric Lauga, Christopher J Pipe, and Benjamin Le Révérend. Sensing in the mouth: a model for filiform papillae as strain amplifiers. *Frontiers in Physics*, 4:35, 2016.

Linda M Bartoshuk, Valerie B Duffy, and Inglis J Miller. Ptc/prop tasting: anatomy, psychophysics, and sex effects. *Physiology & behavior*, 56(6):1165–1171, 1994.

Hannah Jilani, Wolfgang Ahrens, Kirsten Buchecker, Paola Russo, Antje Hebestreit, IDEFICS consortium, et al. Association between the number

of fungiform papillae on the tip of the tongue and sensory taste perception in children. *Food & nutrition research*, 2017.

Xirui Zhou, Martin Yeomans, Anna Thomas, Peter Wilde, Bruce Linter, and Lisa Methven. Individual differences in oral tactile sensitivity and gustatory fatty acid sensitivity and their relationship with fungiform papillae density, mouth behaviour and texture perception of a food model varying in fat. *Food Quality and Preference*, 90:104116, 2021.

Anwesha Sarkar, Efren Andablo-Reyes, Michael Bryant, Duncan Dowson, and Anne Neville. Lubrication of soft oral surfaces. *Current Opinion in Colloid & Interface Science*, 39:61–75, 2019.

Ecaterina Stribiţcaia, Charlotte EL Evans, Catherine Gibbons, John Blundell, and Anwesha Sarkar. Food texture influences on satiety: systematic review and meta-analysis. *Scientific reports*, 10(1):1–18, 2020.

Emma M Krop, Marion M Hetherington, Sophie Miquel, and Anwesha Sarkar. The influence of oral lubrication on food intake: a proof-of-concept study. *Food Quality and Preference*, 74:118–124, 2019.

Siavash Soltanahmadi, Michael Bryant, and Anwesha Sarkar. Insights into the multiscale lubrication mechanism of edible phase change materials. *ACS Applied Materials & Interfaces*, 15(3):3699–3712, 2023.

Fumiyo Tamura, Takeshi Kikutani, Takashi Tohara, Mitsuyoshi Yoshida, and Ken Yaegaki. Tongue thickness relates to nutritional status in the elderly. *Dysphagia*, 27(4):556–561, 2012.

Feng Xu, Laura Laguna, and Anwesha Sarkar. Aging-related changes in quantity and quality of saliva: Where do we stand in our understanding? *Journal of Texture Studies*, 50(1):27–35, 2019b.

Jing Hu, Efren Andablo-Reyes, Alan Mighell, Sue Pavitt, and Anwesha Sarkar. Dry mouth diagnosis and saliva substitutes—a review from a textural perspective. *Journal of Texture Studies*, 52(2):141–156, 2021.

Lisa Murphy, Paul French, Aoife Waters, W Andrew Clement, and Haytham Kubba. Dorsal midline tongue masses in children. *International Journal of Pediatric Otorhinolaryngology Extra*, 13:40–43, 2016.

SR Porter, V Mercadante, and S Fedele. Oral manifestations of systemic disease. *British dental journal*, 223(9):683–691, 2017.

Ni Huang, Paola Pérez, Takafumi Kato, Yu Mikami, Kenichi Okuda, Rodney C Gilmore, Cecilia Domínguez Conde, Billel Gasmi, Sydney Stein, Margaret Beach, et al. Sars-cov-2 infection of the oral cavity and saliva. *Nature medicine*, 27(5): 892–903, 2021.

Jianqiu Jin. Absence of tongue papillae as a sign of disease. *Journal of the American Academy of Dermatology*, 83(6):e425, 2020.

Manabu Maeda. Dermoscopic patterns of the filiform papillae of the tongue in patients with sjögren's syndrome. *The Journal of dermatology*, 33(2):96–102, 2006.

Efren Andablo-Reyes, Michael Bryant, Anne Neville, Paul Hyde, Rik Sarkar, Mathew Francis, and Anwesha Sarkar. 3d biomimetic tongue-emulating surfaces for tribological applications. *ACS applied materials & interfaces*, 12(44):49371–49385, 2020.

Eduard Arzt, Haocheng Quan, Robert M McMeeking, and Rene Hensel. Functional surface microstructures inspired by nature–from adhesion and wetting principles to sustainable new devices. *Progress in Materials Science*, 120:100823, 2021.

Tiffany M Nuessle, Nicole L Garneau, Meghan M Sloan, and Stephanie A Santorico. Denver papillae protocol for objective analysis of fungiform papillae. *Journal of visualized experiments: JoVE*, (100), 2015.

Camilla Cattaneo, Jing Liu, Chenhao Wang, Ella Pagliarini, Jon Sporring, and Wender LP Bredie. Comparison of manual and machine learning image processing approaches to determine fungiform papillae on the tongue. *Scientific Reports*, 10(1):1–15, 2020.

Wei Hong, Xianfeng Gu, Feng Qiu, Miao Jin, and Arie Kaufman. Conformal virtual colon flattening. In *Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pages 85–93, 2006.

Erik J Amézquita, Michelle Y Quigley, Tim Ophelders, Elizabeth Munch, and Daniel H Chitwood. The shape of things to come: Topological data analysis

and biology, from molecules to organisms. *Developmental Dynamics*, 249(7): 816–833, 2020.

Monica Nicolau, Arnold J Levine, and Gunnar Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108 (17):7265–7270, 2011.

Asuka Oyama, Yasuaki Hiraoka, Ippei Obayashi, Yusuke Saikawa, Shigeru Furui, Kenshiro Shiraishi, Shinobu Kumagai, Tatsuya Hayashi, and Jun'ichi Kotoku. Hepatic tumor classification using texture and topology analysis of non-contrast-enhanced three-dimensional t1-weighted mr images with a radiomics approach. *Scientific reports*, 9(1):1–10, 2019.

Aditi S Krishnapriyan, Joseph Montoya, Maciej Haranczyk, Jens Hummelshøj, and Dmitriy Morozov. Machine learning with persistent homology and chemical word embeddings improves prediction accuracy and interpretability in metal-organic frameworks. *Scientific reports*, 11(1):8888, 2021.

Ameer Saadat-Yazdi, Rayna Andreeva, and Rik Sarkar. Topological detection of alzheimer's disease using betti curves. In *Interpretability of Machine Intelligence in Medical Image Computing, and Topological Data Analysis and Its Applications for Medical Data: 4th International Workshop, iMIMIC 2021, and 1st International Workshop, TDA4MedicalData 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, September 27, 2021, Proceedings 4*, pages 119–128. Springer, 2021.

Reem Khalil, Sadok Kallel, Ahmad Farhat, and Pawel Dlotko. Topological sholl descriptors for neuronal clustering and classification. *PLOS Computational Biology*, 18(6):e1010229, 2022.

Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77, 2019.

Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013.

Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. Meshlab: an open-source mesh processing tool. In *Eurographics Italian chapter conference*, volume 2008, pages 129–136. Salerno, Italy, 2008.

Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

Marius Muja and David Lowe. Flann-fast library for approximate nearest neighbors user manual. *Computer Science Department, University of British Columbia, Vancouver, BC, Canada*, 5, 2009.

Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003.

Alessandro Muntoni and Paolo Cignoni. Pymeshlab. *Zenodo, Jan*, 2021.

Alessandro Colombo, Claudio Cusano, and Raimondo Schettini. 3d face detection using curvature analysis. *Pattern recognition*, 39(3):444–455, 2006.

Guillaume Tauzin, Umberto Lupo, Lewis Tunstall, Julian Burella Pérez, Matteo Caorsi, Anibal M Medina-Mardones, Alberto Dassatti, and Kathryn Hess. giotto-tda:: A topological data analysis toolkit for machine learning and data exploration. *J. Mach. Learn. Res.*, 22:39–1, 2021.

Ulrich Bauer. Ripser: efficient computation of vietoris–rips persistence barcodes. *Journal of Applied and Computational Topology*, 5(3):391–423, 2021b.

Harish Chintakunta, Thanos Gentimis, Rocio Gonzalez-Diaz, Maria-Jose Jimenez, and Hamid Krim. An entropy-based persistence barcode. *Pattern Recognition*, 48(2):391–401, 2015.

Nieves Atienza, Rocio Gonzalez-Díaz, and Manuel Soriano-Trigueros. On the stability of persistent entropy and new summary functions for topological data analysis. *Pattern Recognition*, 107:107509, 2020.

Peter Bubenik and Paweł Dłotko. A persistence landscapes toolbox for topological statistics. *Journal of Symbolic Computation*, 78:91–114, 2017.

Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18, 2017.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011b.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Shourjya Sanyal, Shauna M O'Brien, John E Hayes, and Emma L Feeney. Tonguesim: development of an automated method for rapid assessment of fungiform papillae density for taste research. *Chemical senses*, 41(4):357–365, 2016.

Erendira Valencia, Homero V Rios, Inigo Verdalet, Jesus Hernandez, Sergio Juarez, Rosa Herrera, and Erik R Silva. Automatic counting of fungiform papillae by shape using cross-correlation. *Computers in biology and medicine*, 76:168–172, 2016.

Lingxiao Zhao, Charl Botha, Javier Bescos, Roel Truyen, Frans Vos, and Frits Post. Lines of curvature for polyp detection in virtual colonoscopy. *IEEE Transactions on visualization and computer graphics*, 12(5):885–892, 2006.

Padma Sundaram, Afra Zomorodian, C Beaulieu, and Sandy Napel. Colon polyp detection using smoothed shape operators: preliminary results. *Medical Image Analysis*, 12(2):99–119, 2008.

Rayna Andreeva, Alessandro Fontanella, Ylenia Giarratano, and Miguel O Bernabeu. Dr detection using optical coherence tomography angiography (octa): a transfer learning approach with robustness analysis. In *Ophthalmic Medical Image Analysis: 7th International Workshop, OMIA 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 8, 2020, Proceedings 7*, pages 11–20. Springer, 2020.

Nida Shahid, Tim Rappon, and Whitney Berta. Applications of artificial neural networks in health care organizational decision-making: A scoping review. *PloS one*, 14(2):e0212356, 2019.

Mary E Fischer, Karen J Cruickshanks, Carla R Schubert, Alex Pinto, Ronald Klein, Nathan Pankratz, James S Pankow, and Guan-Hua Huang. Factors related to fungiform papillae density: the beaver dam offspring study. *Chemical senses*, 38(8):669–677, 2013.

Gen-Hua Zhang, Hai-Yun Zhang, Xue-Feng Wang, Yue-Hua Zhan, Shao-Ping Deng, and Yu-Mei Qin. The relationship between fungiform papillae density and detection threshold for sucrose in the young males. *Chemical senses*, 34(1): 93–99, 2009.

Ajoy C Karikkineth, Eric Y Tang, Pei-lun Kuo, Luigi Ferrucci, Josephine M Egan, and Chee W Chia. Longitudinal trajectories and determinants of human fungiform papillae density. *Aging (Albany NY)*, 13(23):24989, 2021.

Nieves Atienza, Maria-Jose Jimenez, and Manuel Soriano-Trigueros. Stable topological summaries for analyzing the organization of cells in a packed tissue. *Mathematics*, 9(15):1723, 2021.

Dejan Govc and Richard Hepworth. Persistent magnitude. *Journal of Pure and Applied Algebra*, 225(3):106517, 2021.

Jonathan Jaquette and Benjamin Schweinhart. Fractal dimension estimation with persistent homology: a comparative study. *Communications in Nonlinear Science and Numerical Simulation*, 84:105163, 2020.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in neural information processing systems*, 34:12116–12128, 2021.

Hanan Gani, Muzammal Naseer, and Mohammad Yaqub. How to train vision transformer on small-scale datasets? *arXiv preprint arXiv:2210.07240*, 2022.

Peter Bartlett and Shahar Mendelson. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. *Journal of Machine Learning Research*, 2002.

Shai Shalev-Schwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014.

Herbert Edelsbrunner and John Harer. Computational Topology - an Introduction | Semantic Scholar. *American Mathematical Society*, 2010.

Gunnar Carlsson. Topological pattern recognition for point cloud data*. *Acta Numerica*, 23:289–368, May 2014. ISSN 0962-4929, 1474-0508. doi: 10.1017/S0962492914000051.

Kevin Emmett, Benjamin Schweinhart, and Raul Rabadan. Multiscale topology of chromatin folding. In *9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pages 177–180, 2016.

Yasuaki Hiraoka, Takenobu Nakamura, Akihiko Hirata, Emerson G Escolar, Kaname Matsue, and Yasumasa Nishiura. Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences*, 113(26):7035–7040, 2016.

Gregory Leibon, Scott Pauls, Daniel Rockmore, and Robert Savell. Topological structures in the equities market network. *Proceedings of the National Academy of Sciences*, 105(52):20589–20594, 2008.

Subhrajit Bhattacharya, Robert Ghrist, and Vijay Kumar. Persistent homology for path planning in uncertain environments. *IEEE Transactions on Robotics*, 31(3):578–590, 2015.

Vin De Silva and Robert Ghrist. Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7(1):339–358, 2007.

Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6:1–38, 2017.

Ciprian A Corneanu, Meysam Madadi, Sergio Escalera, and Aleix M Martinez. What does it mean to learn in deep networks? and, how does one detect adversarial attacks? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4757–4766, 2019.

Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in artificial intelligence*, 4:108, 2021.

Afra Zomorodian. Topological data analysis. *Advances in applied and computational topology*, 70:1–39, 2012.

Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001. ISBN 0262032937. URL `http://www.amazon.com/Introduction-Algorithms-Thomas-H-Cormen/dp/0262032937%3FSubscriptionId%3D13CT5CVB80YFWJEPWS02%26tag%3Dws%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0262032937`.

Patrick Rebeschini. Algorithmic fundations of learning, 2020.

Carlos Fernandez-Granda. Lecture 5; random projections, 2016.

Theodore Papamarkou, Tolga Birdal, Michael Bronstein, Gunnar Carlsson, Justin Curry, Yue Gao, Mustafa Hajij, Roland Kwitt, Pietro Liò, Paolo Di Lorenzo, et al. Position paper: Challenges and opportunities in topological deep learning. *arXiv preprint arXiv:2402.08871*, 2024.

Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, Nina Miolane, Aldo Guzmán-Sáenz, Karthikeyan Natesan Ramamurthy, Tolga Birdal, Tamal K Dey, Soham Mukherjee, Shreyas N Samaga, et al. Topological deep learning: Going beyond graph data. *arXiv preprint arXiv:2206.00606*, 2022.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Henryk Minc. *Nonnegative Matrices*. Wiley, New York, NY, USA, 1988.

Steve Huntsman. Parallel black-box optimization of expensive high-dimensional multimodal functions via magnitude. *arXiv preprint arXiv:2201.11677*, 2022.

Nicholas J. Higham. Cholesky factorization. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(2):251–254, 2009. doi: 10.1002/wics.18.

Hanjun Dai, Azade Nazi, Yujia Li, Bo Dai, and Dale Schuurmans. Scalable deep generative modeling for sparse graphs. In *International conference on machine learning*, pages 2302–2312. PMLR, 2020.

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Will Hamilton, David K Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient graph generation with graph recurrent attention networks. *Advances in neural information processing systems*, 32, 2019.

Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International conference on machine learning*, pages 5708–5717. PMLR, 2018.

Paul D Dobson and Andrew J Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4):771–783, 2003.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3): 93–93, 2008.

Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. math. inst. hung. acad. sci*, 5(1):17–60, 1960.