INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

The Shape of Learning: Topological and Geometric Perspectives on Neural Representations

A THESIS

submitted by

SURYAKA SURESH
(PhD22004)

for the award of the degree

of

DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
IIIT DELHI, INDIA
June, 2026

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

The Shape of Learning: Topological and Geometric Perspectives on Neural Representations

A THESIS

submitted by

SURYAKA SURESH
(PhD22004)

for the award of the degree

of

DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
IIIT DELHI, INDIA
June, 2026

About change - *"Time, particle and dimension elude the constraints of existence and perceive beyond the infinity"*
- Me

Dedicated to my mother and father,
Shobha & Suresh,

and my brother, Sonu,
for their endless support.

Certificate

This is to certify that the thesis titled "The Shape of Learning: Topological and Geometric Perspectives on Neural Representations" being submitted by Suryaka Suresh to Indraprastha Institute of Information Technology, Delhi, for the award of the Doctor of Philosophy, is an original research work carried out under my supervision. In my opinion, the thesis has reached the standards, fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

Dr Vinayak Abrol
Department of Computer science and Engineering
Indraprastha Institute of Information Technology Delhi

Declaration

I Suryaka Suresh, hereby declare that the entire work embodied in this thesis is the result of investigations carried out by me in the Department of Computer Science and Engineering, Indraprastha Institute of Information Technology Delhi, with the supervision of Dr. Vinayak Abrol, and that it has not been submitted elsewhere for any degree or diploma. In keeping with the general practice, due acknowledgements have been made wherever the work described is based on findings of other investigators.

Suryaka Suresh
Roll Number : PhD22004
Place : New Delhi
Date: June 29, 2026

Declaration Regarding Use of AI in Thesis

I acknowledge that I am fully responsible for the entire content of my thesis, including any sections assisted by online tools, including Artificial Intelligence-based tools. I accept full accountability for any violations of ethical standards in publications arising from the use of such tools.

Suryaka Suresh
Roll Number : PhD22004
Place : New Delhi
Date: June 29, 2026

Acknowledgement

This thesis marks the end of a journey made possible by the incredible support of many individuals, to whom I extend my heartfelt thanks. Life at IIITD was made truly special by its vibrant 24 hour campus and canteen, which offered a wonderfully serene, green, and pristine atmosphere to work in. I hold many fond memories of hot summer ice creams and snowy winter nights spent here. The vibrant campus life also gave me the memorable experience of attending a live concert for the first time during the college festival.

I wish to express my deepest appreciation to my supervisor, Dr Vinayak Abrol, for his invaluable guidance and mentorship throughout my PhD journey. I am incredibly grateful for his suggestion to work at the intersection of computational topology, archetypes, and deep neural networks, which gave me the opportunity to dive into the complex world of machine learning.

I would also like to thank Dr Kaushik Kalyanaraman for his continued mentorship regarding mathematical proofs, definitions, and theorems. Though not my official supervisor, he graciously allowed me to consult him regularly for clarifications on mathematical language. I know I still have a long way to go to match the immense skills and abilities of my supervisor and professors.

My thanks go to Dr Sumantra Dutta Roy and his students for their continuous support throughout this thesis. I am also grateful to my internal examiners, Dr Supratim Shit and Dr Ashish Kumar Pandey, who generously gave their time and insights during my PhD journey, helping to bring the necessary rigour and complexity to this work. Furthermore, I extend my gratitude to my external examiners, Prof. Bastian Grossenbacher Rieck (University of Fribourg, Switzerland), Prof. Manas Kamal Bhuyan (IIT Guwahati, India), and Prof. Gautam Dasarathy (ASU, USA), who dedicated their valuable time to evaluating and refining this thesis.

I would like to express my gratitude to all the professors who taught me during my coursework. In particular, I extend my thanks to Dr Jainendra Shukla and Dr Saket Anand, under whom I had the privilege of working during my teaching assistantships at IIITD. Furthermore, I thank everyone for their cooperation and patience during my time as Head TA.

My gratitude goes to my peers for setting the bar high and driving my professional growth. I also thank the IIITD staff for their around the clock support and for keeping the campus clean and safe. Finally, I owe everything to my family and friends, without whom I wouldn't be who I am. Thank you for your immense support and sacrifices throughout my PhD, I could not have reached this milestone without you.

Suryaka Suresh
Roll Number : PhD22004

About complexity- “We only know a tiny proportion about the complexity of the natural world. Wherever you look, there are still things we don’t know about and don’t understand. [...] There are always new things to find out if you go looking for them.” — David Attenborough”

Abstract

This thesis presents a unified framework for analysing the "expressivity" of Deep Neural Networks by combining factor analysis with topological data analysis (TDA). The central idea is to study the learning in Neural Networks as a geometric and topological object that encodes the representational capacity of the model. Towards the same, the geometric structure of Archetypal Analysis is exploited to study the structural manifold underlying the neural embeddings. The archetypal subspace provides a scalable foundation for applying tools from computational topology, enabling the characterisation of topological transitions that reflect changes in representational structure and network complexity. Building upon this methodological foundation, the framework is applied to the model selection problem in the context of transfer learning, where the objective is to identify an optimal model from a collection of pre-trained architectures. By quantifying the topological and geometric signatures of the embedding space, the approach provides an interpretable measure of expressivity that correlates with the generalisation behaviour of the Neural Network. Furthermore, the study extends the analysis of expressivity to generative learning, through the topological inference of their generative manifolds forging a relationship between manifold geometry, data diversity, and generative performance. Overall, this research contributes a topology-guided and geometrically interpretable framework for understanding, comparing, and selecting neural architectures. It advances the theoretical and empirical interpretations of deep representation learning and establishes a scalable approach to bridging topological structure, expressivity, and generalisation in neural architecture.

Publications

The overall theme of the research pertains to a unified framework for analysing the "expressivity" of Deep Neural Networks by combining factor analysis with topological data analysis (TDA). The central idea is to study the learning in Neural Networks as a geometric and topological object that encodes the representational capacity of the model. Experimental code is available at the following GitHub Repo.

Journal:

- S. Suresh, V. Abrol, K. Kalyanaraman, and S. Dutta Roy, "Topological Persistence of the Neural Embedding of the Archetypal Subspace," in IEEE Journal of Selected Topics in Signal Processing, vol. 19, no. 7, pp. 1514-1524, 2025.
- S. Suresh, B. Das, V. Abrol, and S. Dutta Roy, "On characterizing the evolution of embedding space of neural networks using algebraic topology," in Elsevier Pattern Recognition Letters, vol. 179, pp. 165–171, 2024.

Conference/Workshop:

- S. Suresh, and V. Abrol, "Evaluating Generative Models via Cubical Homology based Persistent Entropy," in ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), V.2, pp. 2768–2777, 2025, Vancouver, Canada.
- S. Suresh, V. Abrol, and A. Thakur, "Pitfalls in measuring neural transferability," in NeurIPS 2023 Workshop on Symmetry and Geometry in Neural Representations, New Orleans, USA, 2023. (Proceedings Track)

GitHub Repo: Experimental code is available at the following **Repository** .

Contents

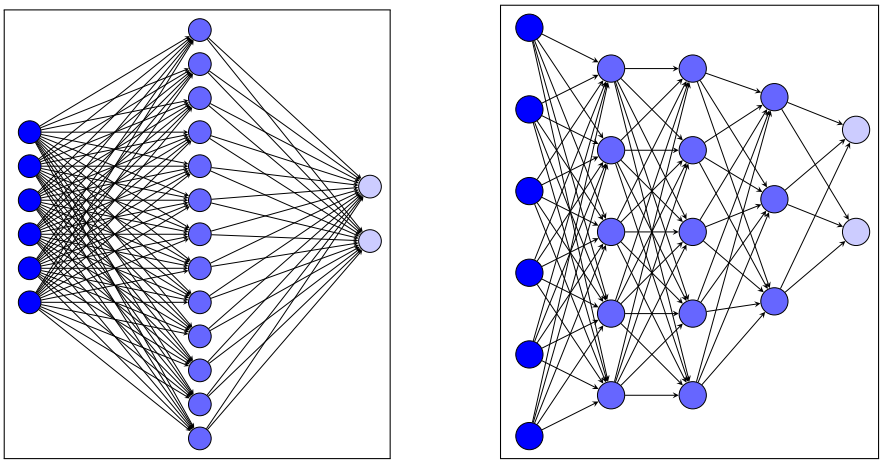
“ The important thing is not to stop questioning. Curiosity has its own reason for existing”-Albert Einstein

Symbols

$\mathbb{R}$	Real number
$\mathbb{N}$	Natural Numbers
$\mathcal{D}$	Dataset
D	point cloud/ feature space
$\|D\|$	Number of instances /cardinality of a set
$\mathbb{D}$	Collection of cubical set
D_i	Instance/ Images/cubical set
C	class index
$\mathcal{X}$	topological space
$\cong$	isomorphism
$\beta(\mathcal{X})$	Betti Number of topological space $\mathcal{X}$
$\omega(\mathcal{X})$	Homological/Topological complexity of $\mathcal{X}$
η	threshold in Persistent homology
Γ_η	subcomplex at η in PH
dgm	persistent diagram
$\widehat{H_i}$	Homology in Persistent diagram
$\widehat{\beta_i}$	i-th Betti number in Persistent homology
ω_η	homological complexity at Γ_η
λ	average life span
PL	Persistent Landscape
E	Persistence Entropy
$\text{conv}(D)$	Convex hull of D
$\mathring{A}_D$	Archetypes of D
$\mathring{A}_c$	Archetypes of class $c \in C$
$\mathring{A}_C = \bigcup_c \mathring{A}_c$	Union of class archetypes
$\mathring{A}_{np(c)}$	nearest points to the archetypes of class
$\mathring{A}_{np(C)} = \bigcup_c \mathring{A}_{np(c)}$	Union of nearest archetypes
ρ	metric/ distance measure
$\rho_{\mathbf{e}}$	Hausdroff distance
f	Neural Network/classifier
$H_s f$	Support homology of f
$\mathcal{F}$	collection of functions
Φ_f	Homological capacity or $MaxTCE - NC$
TL	transferability score
μ	Mean
$corr$	Correlation

Chapter 1

Introduction



(a) Shallow Wide (b) Deep

Figure 1.1: **Width vs Depth**: The number of hidden neurons in both the networks (a) and (b) is the same. Which architecture is better in terms of its "expressivity"?

Deep Neural Networks have helped shape the new era of "Artificial Intelligence" by enabling learning from data. However, due to the "black box" nature of the network, there is a lack of interpretability in the "function" (pattern) it learns. The consequences of this are often reflected when models are deployed in the real world, such as bias towards a particular class, hallucinations in LLMs, etc. In terms of training, currently, complex real-world data requires large Neural Architectures with millions of neurons. However, this, in turn, consumes a large amount of natural resources due to its high computational complexity. Thus, among the model zoo [1], an optimised model in terms of parameters is preferred, so as to minimise the resources consumed. Other methods used to tackle the issue include representation learning techniques, such as transfer learning, which avoids training from scratch, as well as rigorous hyper-parameter tuning to optimise space and time complexity. For example, in the paper [2], the author illustrates neural networks' ability across various factors, including the influence of the initial weight space on learning ease. Alternatively, in this work, the embedding space of the Neural Network is explored using topological tools to understand and quantify the complexity of the function it approximates.

The Neural Network zoo [1] illustrates that there exists a vast landscape of potential architectures, making the selection of architecture through training infeasible. A possible approach is to use Occam's razor, or the principle of parsimony, which suggests that, among models of equivalent explanatory power (or expressivity), the simplest configuration is preferred. A practical application of this principle is identifying the lower bound on the depth required for a Neural Network to approximate a target function. This is because expressivity in Neural Networks is studied in terms of depth and width, with deeper

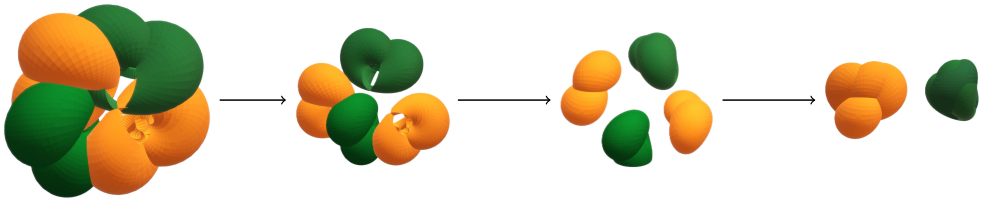


Figure 1.2: **Topological Transitions: Non-homeomorphism vs Homeomorphism**: Data complexity is quantified via the homological complexity [3] of the underlying manifold, where the nature of the topological transition is task-dependent. For instance, an ideal classifier performs a non-homeomorphic mapping to collapse the homology of a class to that of a contractible space (a single point), whereas generative tasks seek to preserve the manifold structure. The green and orange regions denote distinct class manifolds. The visualisation illustrates the evolution of these manifolds within a representation learning model as it transforms the feature space to achieve class separability and establish a decision boundary.(See Books [4, 5] and Chapter 3 for definitions)

networks found to have better explanatory power than their shallow counterparts under the same resources. As depicted in Figure 1.1, the deeper network has the same number of parameters as its shallow counterpart but is suggested to have the capability to address more complex functions. Thus, in this work, we attempt to study expressivity in terms of the "Topological shape" of data.

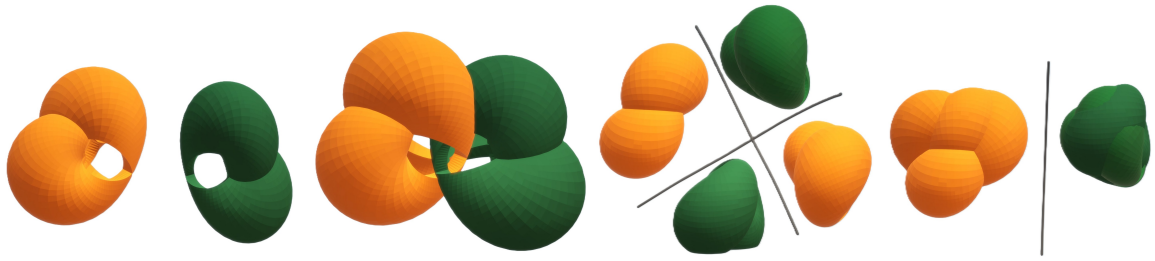
1.1 Topological "Shape" of Data

In topological terms, data is often considered sampled from an underlying manifold. These manifolds are further characterised by topological invariants such as Betti number (β). In the paper [3], the authors propose homological complexity (the sum of Betti numbers) as a measure of the ability of a Neural Network to approximate a function. The paper further illustrates that the propositions align with the principle that Deeper Neural Networks outperform Shallow Networks under similar resource constraints. The superiority of depth over width in terms of network expressivity is discussed across various frameworks, ranging from Random matrix theory [2] to Information theory [6]. Existing work in topology often studies the evolving weight space during training [7] or compares the extracted shape using topological tools such as Mapper [8]. In contrast, this work explores the following:

Topological Transitions : How do the Topological transitions of the embedding space (layer wise outputs) of the Neural Network affect its functional capacity? Computational topology provides a robust framework for interpreting high-dimensional, complex data manifolds. We empirically demonstrate that the non-homeomorphic mappings typically associated with the weight space can be effectively analysed through the persistent homology of the embedding space of the deep neural classifier.

Dimensionality : While computational topology enables the inference of latent structures from discrete data, these algorithms are often constrained by the curse of dimensionality. To address this, we investigate scalable methods for extracting topological invariants from a representative subsample. Specifically, we exploit the geometric nature of data through "Archetypes" of data. We study the topological space of "Archetypes" of data and track the layer-by-layer topological transition of the Embedding space of the Archetypal Subspace of the data for a Deep Neural Classifier.

Generative Framework : Can topological measures be used to study different deep network systems? By studying the "topological diversity" of images within the Output space of generative models, we use the topology of the feature space as a measure of quality and diversity of generated images. Further, we examine the distribution of "topological diversity" in generated data, measured via persistent homology and relate the stability of persistence diagrams to learning in Neural Architecture.



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

Figure 1.3: **Topology vs Decision Boundary**: Figures (a) and (b) illustrate manifolds where classes are well separated and intertwined, respectively. We argue that a decision boundary can be established in (a) independently of the underlying class homology, whereas the intertwined structure in (b) necessitates a topological transition for separation. In Figures (c) and (d), we observe different decision boundaries for the same manifold in Figure 1.2. These cases illustrate that while topological structures fundamentally constrain the decision boundary, the impact of a given structure is further mediated by its specific influence on class separability.

In summary, this work investigates the topological requirements of neural networks across distinct learning paradigms. By shifting the analytical focus from the high dimensional weight space to the intrinsic topology of the embedding space and the generated manifold, we establish a task driven framework for evaluating model expressivity. The following chapters detail the theoretical foundations of homological capacity and present empirical evidence of its utility for systematic model selection diagnostic tool.

1.2 Framework and Objectives

The empirical evaluations in this work are primarily conducted using computer vision architectures and image based datasets. Although we seek to investigate the "expressivity" of the Neural Network in a model and data agnostic setting, existing results are restricted by empirical evidence, the curse of dimensionality of data, and theoretical gaps [3]. We explore the embedding space of the layers of Convolutional Neural Network [9] across various paradigms, quantifying the extracted topological structures through persistence vectorisations.

Furthermore, this work examines the theoretical relationships between data homology and Deep Neural Networks. The main results in this work are based on the computational tool "persistent homology". While this method is valued for its stability under perturbations and its interpretability, it is characterised by high computational complexity in both time and space. We address these computational bottlenecks through several strategies, including subsampling via archetypal analysis and selecting specific filtrations for persistent homology. A formal introduction to the fundamentals of Neural Networks (Chapter 2) and Topology (Chapter 3 and Reference Books [4, 5]) is provided in the preliminary chapters to facilitate a comprehensive understanding of the subsequent contents.

Topological transition of the Embedding space of Neural Network: Since classifiers have a finite, discrete output space with data instances reduced to an associated label, the underlying manifold of the embedding space of the layers of the Neural Network collapses the homologies. With respect to the same, we ask a similar, but computationally feasible, question: how do the homological structures of the features extracted from an instance (image) change layer by layer in Deep Network classifiers? Towards the same end, we study the topological transition of the embedding space of individual image instances, enabling us to exploit computational tools in lower dimensions and address the curse of dimensionality. In the Chapter 4, we empirically study the embedding space via cubical homology, verify the non-homeomorphic mapping of the layers of the Deep Neural Classifier, and attempt to relate this to the approximation ability of different model architectures. For real-world applications, we measure the

topological transition to evaluate model performance in a transfer learning paradigm.

Topology of classifiers via Geometrical Exploitation of Archetypes: While computational topology enables the inference of topological transition of the embedding space (as seen in Figure 1.2), these algorithms are often constrained by high computational complexity. To mitigate the curse of dimensionality, we investigate scalable methods for extracting topological features from a representative subsample. Specifically, we exploit the extremal geometric properties of archetypes (see Chapters 5 and 6) as a proxy for the homological expressivity required for deep neural architectures to achieve class separability in binary classification tasks. Towards the same, we also explore the topological "shape" of data as captured by the subsample of Archetypes in the Chapter 5 and its variant, the kernel Archetype in the Chapter 7.

Generated Manifold of Generated models: In this Chapter 8, we show the importance of the topological paradigm beyond the classification framework with the example of a generative framework. Here, the topological measure known as persistence entropy is used to examine how the distribution of topological diversity within generated images transitions during training and how it relates to mode collapse.

Visualising Training via Archetypes: In this Chapter 7, topological transition is analysed during the phase of training using point clouds with classes intertwined as in the Figure 1.3(b). For the study, various multi-layered networks at different depths with different activations were used.

1.2.1 Prerequisite Mathematical Foundations

The theoretical framework of this work relies on established principles of real analysis and calculus (limits, differentiation, and interpolation) [10, 11], vector space theory and convex analysis [12, 13, 11, 14], and statistical inference (Probability, Expectation, Correlation, the Central Limit Theorem, p-values, Risk functions and Hypothesis testing) [15, 16, 17]. The topological foundations, ranging from point set topology to the group theoretic structures of homology used to prove the main theorems in Chapter 6, are adapted from Bredon [4] and Dey and Wang [5]. Comprehensive definitions for critical measures such as cardinality, metric spaces, and the Hausdorff distance have been integrated into the Topological Preliminaries (Chapter 3) to maintain a self-contained narrative while adhering to the formal rigor of the cited literature.

1.2.2 Topological Constructions

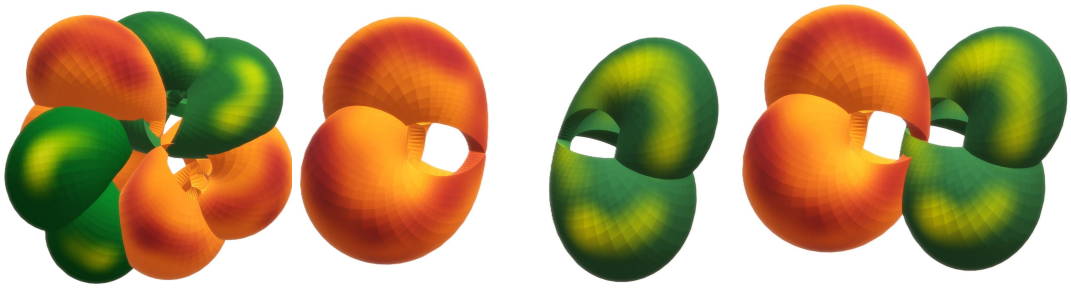
This work employs two distinct frameworks within computational topology to study the Deep Neural Network (in particular, the Convolutional Neural Networks):

Cubical complex : Leveraging the grid like structure of CNN feature maps, cubical complexes provide a significant computational advantage over simplicial constructions. This enables a direct layer wise interpretation of the homology of embedding space. (see chapter 4)

Simplicial complex : When using a simplicial complex, we have a well-grounded theory that helps understand the changing manifold of data. To mitigate the instance based dimensional complexity, we adopt archetypal subsampling. This method is selected for its geometric interpretability and the specific susceptibility of class archetypes to boundary-region misclassification (see Chapter 5).

1.3 Linking Geometry and Topology

In this work, Archetypal Analysis ($\mathring{A}$) (Chapter 5) is utilised to investigate the interplay between the decision boundary and topological transitions within a neural network. As archetypes represent the extremal points of a dataset lying on the principal convex hull, in a classification context, these archetypes define the extremal boundary of a class, allowing for a structured analysis of the embedding space



(a) (b) (c)

Figure 1.4: **Homology of Dataset and Archetypes $\mathring{A}$**: Here, various topological manifolds with different complexity, which can be studied via concepts of Algebra and Topology, are given. While Figure (a) has a different topology, Figures (b) and (c) are based on the concept of "Links" with the same topology. The red and yellow shading on the manifolds is used to provide conceptual clarity about the relative proximity and influence of class archetypes on the decision boundary. These visualisations serve as qualitative illustrations; for a rigorous mathematical treatment, see Chapter 5.

transitions (see Figure (see Figure 1.4). Additionally, Archetypes $\mathring{A}$ are disjoint approximations from the data that could span the data. In this work, we explore two directions :

Class Archetypes: This direction exploits the extremal representation of archetypes alongside topological inference theorems to derive a formal relationship between the homology of class archetypes and the support homology of the classifier (see Chapter 6)

Archetypal Homology and Visualisation: This study examines the properties of persistence vectorisations to establish a rigorous relationship between the homology of the full dataset and its archetypal proxy. Furthermore, the archetypal subspace is utilised to visualise the evolution of the decision boundary throughout the network layers.(see Chapter 5)

1.4 Brief Overview Of The Thesis

While Figure 1.3 provides an insight into the homology and decision boundary, Figure 1.4 provides an intuition into the possible influence of Archetypes $\mathring{A}$ on the homology of the dataset and decision boundary. Here, both Archetypal Analysis and Archetypes ($\mathring{A}$) are used interchangeably. The two major contributions of this work are as follows :

Computational topological tools in Neural Network : This research investigates the application of computational topology to quantify the expressivity of Neural Networks. The contribution includes a formal framework for measuring topological transitions and a statistical analysis of these metrics. While the derived approximations demonstrate a moderate correlation $(0.6 - 0.7)$ with classification accuracy in transfer learning paradigms, the framework provides a theoretically grounded methodology for interpreting architectural capacity beyond empirical performance.

Geometric Integration in computational topology : This study investigates how to utilise the extremal geometric properties to characterise the homology of the dataset. Furthermore, it presents a novel application of the Homological Inference Theorem to relate the geometry of the class boundary to the required homological expressiveness of the Neural Network.

The frameworks in the work investigate the notions on a relatively light level. More detail is required to bring in the rigour needed to establish the properties of Neural Networks, such as their "Expressivity" and

"Generalisability", which require further mathematical introspection into the theory of Neural Networks. Here, we compile many existing works on "Expressivity" and ask questions to spur further research in the direction of Algebra and Topology to possibly categorise Neural Networks based on a measure of complexity.

1.5 Conclusion

The principal results of this work investigate the application of computational topology to the study of Neural Networks, addressing the challenges of scalability and interpretability within real world constraints. Furthermore, we explore the theoretical and empirical extension of topological analysis to diverse representation learning paradigms such as classification and generative models. Our findings validate the premise that characterising the intrinsic topology of the data is a fundamental requirement for effective model learning.

Notably, this research leverages the geometric extremal properties of archetypes to approximate the class feature space, enabling a structured analysis of the embedding space of the neural network. While this work does not posit that algebraic topology is the singular optimal method for neural network analysis, it demonstrates that algebraic structures provide a mathematically rigorous and theoretically grounded framework for the scrutiny of deep learning architectures. The following chapter presents a formalisation of existing work on "expressivity" in Neural Networks.

Chapter 2

Background 1: Expressivity of Neural Network

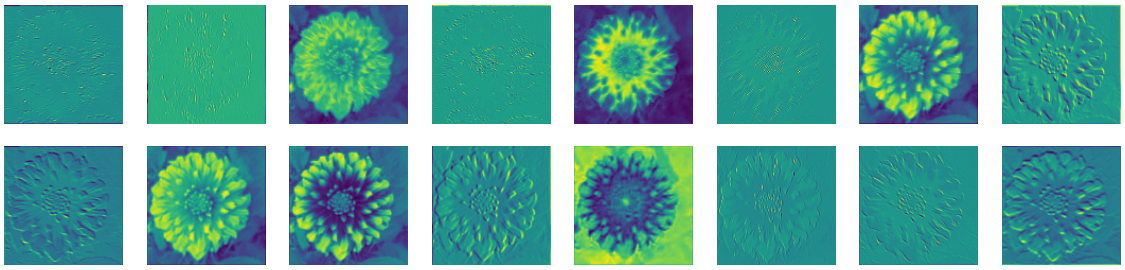


Figure 2.1: **Convolution operator :** In a convolutional Neural Network, the weights are kernels that are known for filtering out and preserving frequency components of the data. How does "depth" affect the filtered frequency components?

"Deep learning" (DL) has played a pivotal role in the much-needed onset of artificial intelligence (AI). The DL framework transforms "Data" into a "refined" form that helps solve an underlying task. But there is much less understood about this "refined" form, creating a lack of interpretability resulting in unforeseen consequences. Thus, a deep learning model is often considered a "Black box". The practical implications of this include ethical concerns that arise in the real-world deployment of the models, in addition to the computational cost in training and efficient updating of the models. In particular, deep learning that uses Neural Networks is a data-driven model and thus depends largely on the "data complexity" (volume, dimension, nonlinearity, quality ...). In this, we define the "data complexity" in terms of algebraic topology (see chapter 4). In this chapter, we let $D = \{(z, y)\}$ represent the dataset with $|D| = N \in \mathbb{N}$ (finite) and leave the more general definition to chapter 3. In this chapter, we give an overview of Neural Networks, problem statement and current research.

2.1 What is a "Deep" Neural Network?

Neural Networks are artificial neurons stacked together to learn features from data that can then be used to derive decisions for a particular task, such as classification. A neuron is the fundamental unit of Artificial Neural Networks and is a weighted sum of the input with nonlinearity incorporated using an activation function σ. The layers of the network contribute to the "depth" and the number of neurons in a layer contributes to the "width". An infinite "depth" or "width is often studied as a part of understanding the approximation capability of an idealised Neural Network. But for the practical real-world computers, "infinity" is unphysical, and therefore it is necessary to introduce a possible association between the data

(a) Dahlia pinnata (b) Dahlia M Pinnata (c) Sunflower

Figure 2.2: **What is a classifier ?:** Given are different flowers with Random Image Augmentations. From the human perspective, the data augmentations do not change the "Identity" or "class" or "category" of the flower.

complexity and parameter complexity of Neural Network. For a network of L layers, each layer ℓ as n_ℓ neurons.

$$\widehat{W}_\ell = W_\ell^0 + (W_\ell)^T \sigma(\widehat{W}_{\ell-1})$$
$$\text{where } \widehat{W}_1 = W_1^0 + (W_1)^T z$$

The linear Neural Network are differentiable with back propagation (the fundamental method of training neural networks) built on the idea. On the other hand, the differentiability of a nonlinear Neural Network is subject to the type of activation functions σ. In the equation above, W represent the weight matrix, with σ an identity map or a linear map producing a linear function. It is important to remember that linear models are the simplest and most interpretable.

2.1.1 Classifier

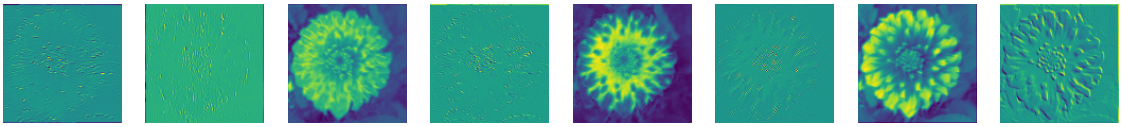
Humans are capable of distinguishing and categorising objects in the real world based on their patterns. For example, consider Figure 2.2, all the images in (a) and (c) represent a Dahlia pinnata and Sunflower, respectively, which is visually apparent for humans, but it is hard to describe such functions f in terms of elementary mathematical operations such as addition. The paradigm of machine learning [18] extends to approximating such functions ($\hat{f}$) that can mimic this capability of humans. In this case, the input to the function is a digital image (z vector), and the output y is 0 if the image is Dahlia pinnata and 1 if it is a Sunflower. In machine learning, these problems where the outputs or "responses" belong to a finite set ($y \in \{0, 1..., n\}, n \in \mathbb{N}$) are called classification tasks. These tasks are mostly solved by discriminative or probabilistic models. In this work, we explore transfer learning (see chapter 5) on the classification task [18, 11]. There are many different machine learning models Figure 2.4, with classical models having both the theoretical structural strength and Neural Networks, which are capable of multi-tasking and solving complex real-world problems such as automation.

2.1.2 Convolutional Neural Networks

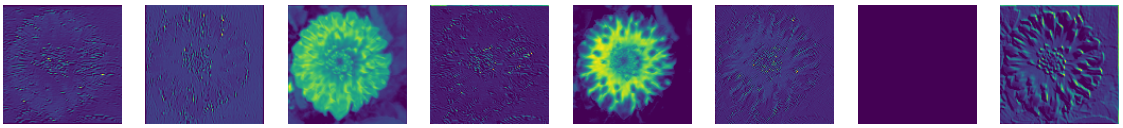
Here, for the real-world model, we work mostly with Convolutional Neural Networks (CNN). Convolutional Neural Networks [9, 19] are a popular class of Neural Network architecture developed by Yann LeCun. CNNs are Neural Networks specialised in data of grid-like topology, such as digital images. The mathematical formulae for the Convolution operator [20] for a discrete 2-dimensional grid I and kernel/filter W as described by Goodfellow in the deep learning book [18, 11], are as follows :

$$\hat{I}(i,j) = I * W(i,j) = \sum_m \sum_n I(m,n) W(i-m, j-n)$$

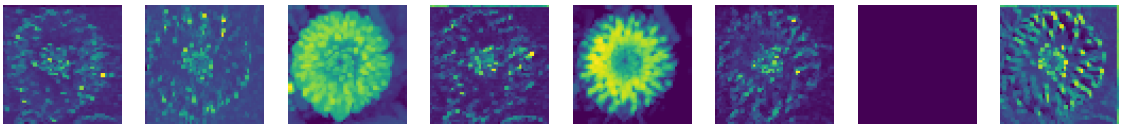
Thus, outputs of convolutional operators have a grid topology themselves (see Figure 2.1 & 2.3). Convolution is equivalent to applying cross-correlation after flipping the filter horizontally and vertically. This induces communicative property in the convolution operator, which does not exist in the cross-correlation



(a) Batch norm



(b) ReLU



(c) Max pooling

Figure 2.3: **Auxiliary component :** Neural Networks further have additional features such as Batch Normalisation and Max pooling, which add complexity to their representation, making it difficult to interpret the patterns extracted from the hidden layers.

operation. Nevertheless, for computational efficiency and the ability to learn without significant performance loss, correlation is preferred over convolution for practical implementation. In the experiments that follow, the PyTorch library [21] is used to implement the deep learning models, which employ cross-correlation. These outputs in sequence represent the compositional effect of the layers of the ResNet. "Composition" effect of the Neural Network induces the trade-off between the complexity and interpretability in Neural Networks. This composition effect, if looked at from the perspective of kernels under the convolution operator, one can think of the communicative property of the convolution operator. Although the multistage convolution in a CNN differs from a simple sequential convolution, here the focus is on the size of the kernel. Let f_ℓ denote the ℓ-th kernel where $\ell = 1, ..., L$. Then, on a 2D matrix input D, the multistage convolution ($\circledast$) can be rewritten as the convolution using a single kernel f :

$$f_L \circledast (f_{L-1} \circledast (....f_2(f_1(D)))) = (f_L \circledast f_{L-1} \circledast ....f_1)(D)$$

$$f_L \circledast (f_{L-1} \circledast (....f_2(f_1(D)))) = f(D)$$

Now consider no padding on D and a stride of 1; if D was filtered 10 times sequentially using a 2×2 kernel, then the output size is the same as filtering with a 11×11 kernel. Now the 10, 2×2 kernels have a total parameter space of 40, while the single kernel would require 121. Thus, the deeper models provide a large receptive field with fewer learnable parameters, reducing the risk of overfitting and computational cost. Receptive field (see figure 2.4) in the context of deep learning is the size of the input that a neuron in a layer is influenced by to produce the output.

2.1.3 Different Architectures

In this work, we explore the different architectures used in this work (see chapter 4 for more details, such as the depth and implementations):

VGG : Very deep convolutional Network [22] is one of the empirical Neural Networks that demonstrate the benefits of "depth". It is known for the simplicity of its architecture as well as its performance and generalisation ability, but at a high computational cost. These models have a large parameter space, with the main contribution being the fully connected layers, which receive a large number of outputs

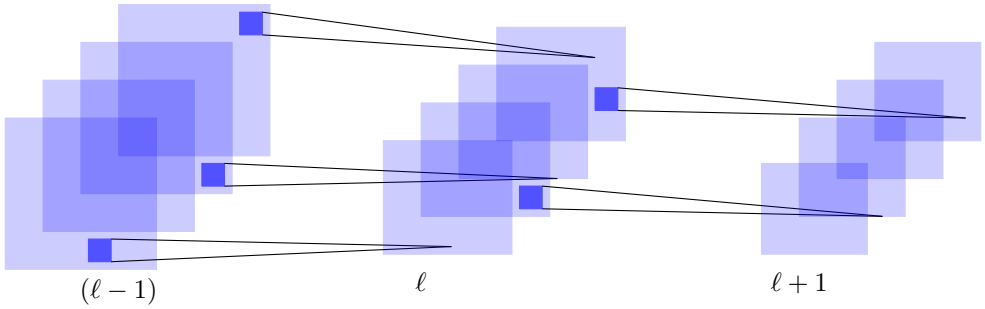


Figure 2.4: **Embedding Space:** A CNN implements a complex multistage convolution which cannot be represented by a single kernel. The receptive field of the input image increases with each convolution layer.

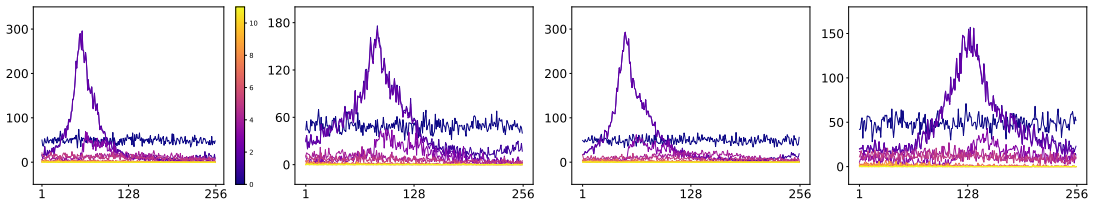
from the preceding convolutional layer. This also makes fine-tuning of these models (even just the FC layers) highly computationally inefficient.

ResNet : The learning at "depth" is often affected by the degradation in the information. Residual block [23] is one of the prominent methods that address this problem. A residual connection simply adds an identity mapping of a representation, bringing the information into the deeper layers of the network, and thus is said to have a recursive structure. The intricate characteristics of ResNet, attributed to the residual connections, often lead to its expressivity being examined independently from other neural networks.

DenseNet : DenseNet [24] is a subsequent advancement built upon ResNet by allowing the concatenation of features of all preceding layers, allowing for more efficient feature extractions and reduced parameter redundancy. Note that in ResNets the residual connection combines features through the summation, but in a DenseNet, the input to the $\ell-$th layer consists of outputs from all preceding convolutional blocks. The feature reuse increases the variation in the input in the subsequent layer, leading to implicit deep supervision, and the final classifier makes a decision based on all the convolutional output spaces in the Neural Network.

MobileNet : With a preference for faster and computationally limited real-world tasks, the efficient network architecture, such as MobileNet [25], provides compact models. The MobileNet architecture, in particular, involves depthwise separable convolution, which allows computations to be broken down into smaller convolution operators. Further, MobileNetV2 [25] with an additional inverted residual structure and MobileNetV3 with a squeeze and excitation model added to MobileNetV2 are extensions of the MobileNet series. The MobileNetV2 has inverted residual blocks with a linear bottleneck that prevents the over-erasing of information due to nonlinearity. The name inverted residual blocks is due to its sequence of operation, compression then expansion, which is the converse of the residual blocks. The inverse residual connection allows for reduced parameter space and computation, but the deep convolutional separability often limits their capacity to learn complex features. Thus, using them as a pre-trained model is the same as starting with a less powerful set of foundational features, and their fine-tuning accuracy is at the bottom.

MnasNet : The Neural Network is built using the automated Mobile Neural Architecture Search (MNAS) [26] approach, which uses a latency-aware multi-objective reward and factorised hierarchical search space. While the latency-aware multi-objective allows for better accuracy and inference latency, the factorised hierarchical search space allows the incorporation of the diversity of layers and flexibility. Unfortunately, similar to MobileNets, these models have low fine-tune accuracy when only the fully connected layers are trained. This shows that the capacity of the model is much less compared to other models. There do exist other Neural Networks with better performance belonging to the same category



(a) Kernel 0 (b) Kernel 20 (c) Kernel 40 (d) Kernel 60

Figure 2.5: **Histogram of the Embedding space:** The given figure represent the histograms of the n^{th} kernel in the embedding space of a Convolutional Neural Network. With deeper layers, the values are flattening, representing a uniform distribution.

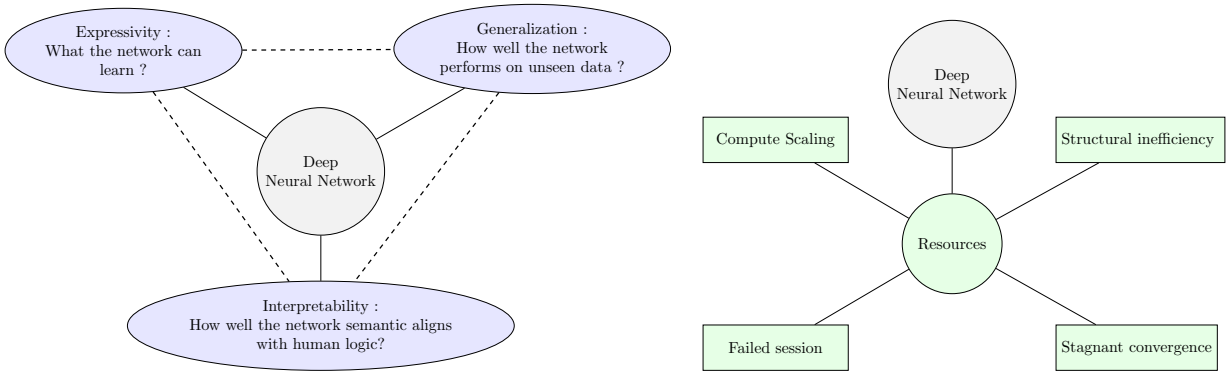
as MobileNet and MnasNet, but here only these two models are considered. Due to fewer parameters and operations (1.8× faster than MobileNet), they are faster than other Neural Networks.

The neural network architecture is further characterised by additional features such as Batch Normalisation that Normalise the input space, the nature of the activation function, such as ReLU that constrains the input space to the non-negative space and Max pooling that reduces the feature space to retain relevant information (see Figures 2.3 a,b & c respectively). We observe that outputs from Figure 2.1 is similar to Figures 2.3(a,b & c) with Figure 2.3(a,b & c) inducing a darker contrast but . Since the embedding space in Figures 2.3 is based on the image, we observe how the histogram of the images changes with each layer. We observe that the histograms are getting equalised, and the higher contrast in the image represents the information erased as a result of nonlinearity induced by ReLU. The Figure 2.5 is the histogram of the embedding space of the layers of the ResNet50 Convolutional Neural Network for a single image. The observations from Figure 2.5 also show that the histogram flattens with deeper layers. However, the output from different kernels has different peaks in the initial layers, possibly due to the feature preference of the kernel. Even if the analyses of the histogram would provide further understanding into the Deep Neural Network, it is not feasible to do the same for individual images in a large dataset with thousands of images. Here, computational topological tool persistent homology is used to study this change in chapter 4 and extend towards real-world applications, taking into consideration the computational feasibility.

2.1.4 Computational Cost Of Neural Networks

Computing an iteration of a Deep Neural Network will require more time than its shallower counterpart [27]. However, a Deeper Network with the same resources is known for its ability to approximate complex functions. This trade-off in the relation falls in polynomial bounds for the Deep network and exponential bounds for the wider network [3], making it more optimal to use the Deeper Network. On the other hand, Deeper Networks struggle with training issues such as vanishing gradients, leading to poor learning, which is addressed by adding features, such as residual connections. These additional features add to the further complexity in the embedding space, making modern real-world neural architectures even less interpretable. Large Language models require a large number of parameters, leading to high computational cost, and intricate methods of quantisation of the models end up creating black boxes that are susceptible to multiple factors, such as the bias in the data they are trained on. At the end of the day, due to their large impact in the real world, they end up as headlines for newspapers for all the wrong reasons.

Their impact on the environment (see Figure 2.6(b)) is the same reason we attempt to limit the resources required to compute the complexity in this work.



(a) "Trinity" of Machine Learning (b) Carbon cost

Figure 2.6: **Learning:** The figure (a) illustrates the three fundamental pillars affecting the complexity of learning in Neural Networks. (b) The limited understanding of the learning trinity governing neural network complexity forces unoptimised, massive resource allocation to combat compute scaling limits, convergence stagnation, session failures, and structural inefficiencies. Thus, consequently accelerates ecological and carbon expenditures.

2.2 Foundational learning

In this section, we briefly review the concepts of generalisation, decision boundaries, and model capacity in the context of VC dimension and Rademacher complexity. Given that the foundational depth required to explain these concepts is extensive, we adopt notations consistent with the cited references, and these notations are specific to this section and do not necessarily apply to other parts of this thesis. (See Books [18, 15, 16, 17] for various mathematical definitions)

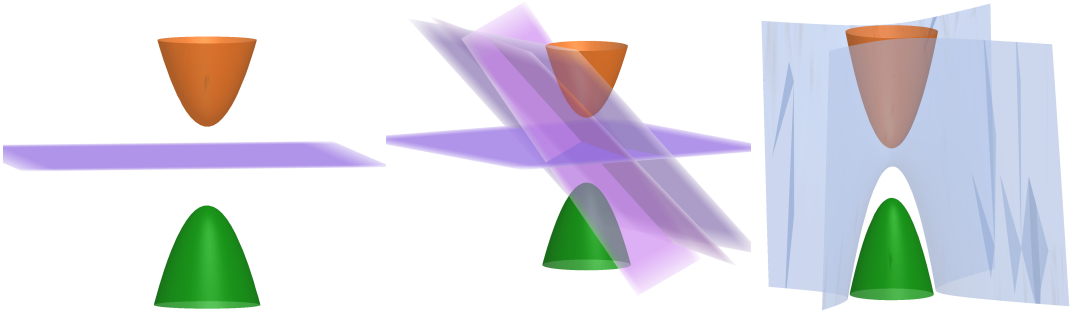
2.2.1 Generalisation

As per Mitchell et al. [28], the formal definition of a computer program learning is given by: *"A computer program is said to learn from experience E with respect to some class tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience"*. Thus, a well-defined learning problem must have a task, a performance measure and training experience. Informally, learning is the method of attaining the capability to perform a task. The performance measure is specific to the task; in the context of classification, we typically employ the **accuracy or error rate** [11]. In this work, accuracy is used to validate the proposed topological measures. According to Goodfellow et al. [18], *a learning algorithm can be understood as being allowed to experience an entire dataset.* Furthermore, Mitchell et al. [28] state that the design of a learning system requires a target function and a corresponding representation. This is referred to as point estimation [18] or *the estimation of the relationship between input and target variables.* In particular, the function estimation $\tilde{f}$ of a function f is defined as a point estimator in function space. Statistically, it is the same as estimating a parameter space [15, 16, 17].

Let the error rate be denoted by R. A dataset $\mathcal{D}$ can be partitioned into two "training set" and "test set". While the training set is used for learning, the performance of the algorithm on the unseen test set is called generalisation [18]. The generalisation error or test error is the Error rate over the test set (R_{test}) or the expected error over the new instance. In terms of mean squared error (R_{test}) over a test set $\mathcal{D}_{test}$ of N values:

$$R_{test} = \frac{1}{N} \sum_{z \in \mathcal{D}_{test}} (f(z) - \tilde{f}(z))^2$$

The model capacity [15, 16, 17], on the other hand, is the ability of the model to approximate the function. A model with high capacity can approximate functions of higher complexity. The capacity



(a) Linear plane (b) Different solutions (c) Complex

Figure 2.7: **Complexity of Decision boundary:** For well separable classes in this example, it can be observed that there are infinitely many solutions (b) for the decision boundary, from the simple linear (a) to the complex nonlinear in (c). One can now ask the question, which Neural Network architecture will be capable of approximating the different hyperplanes in this figure with the fewest parameters?. The orange and green manifolds represent different classes. It is essential to recognise that the infinitely many decision boundaries exhibit both linear and nonlinear characteristics, whereas the data itself is inherently nonlinear. Then, what structure of data affects the decision boundary? (see Chapter 7)

determined through varying the parameter space is called the representational capacity. On the other hand, the learning limitation, such as imperfection in the optimisation algorithm, results in a capacity that is less than the representational capacity called effective capacity [18]. In this work, we utilise the homological complexity necessary [29] for classification tasks to study real-world architectures.

2.2.2 Decision Boundary

In a classification model, a decision boundary (Figure 2.7) is a conceptual hypersurface used to separate different classes or categories in a feature space. It represents the surface where the prediction transitions from one class to another, and their shape and complexity are learned from the training data and influenced by the model framework and complexity. When it comes to Neural Networks, the decision boundary could be a complex nonlinear hypersurface or a simple linear hyperplane, as demonstrated in Figure 2.7. A single linear neuron is capable of a linear hyperplane; these are simple and fast as compared to Figure 2.7(c), which would require nonlinearity, implying more parameters and depth. But then what about the generalisation ability of the decision boundaries in representation learning, such as transfer learning, where we want to classify a task on target data using a Network trained on source data. Although one can argue here that the hypersurface in Figure 2.7(c) is not the best, having lesser generalisation capability than Figure 2.7(a) (since one tends to believe that a simpler function is better), a reasonable conclusion would be that there exists a trade-off between generalisation capability and the complexity in such a scenario. Here in particular, the expressivity of the Neural Network or the approximation capability of the Neural Network, which studies the neural architectures that are capable of approximating hypersurfaces, such as in Figure 2.7(c), is combined with generalisation through the study of transfer learning and model selection diagnostic tool (see chapter 4). This book also explores the reason behind using embedding space and representation learning, such as transfer learning, to study and empirically verify claims of generalisability and expressivity.

2.2.3 Classical Models vs Neural Networks

When it comes to interpretability, classical models often provide a reliable theoretical background for understanding the complexity as well as the structure of the decision boundary, such as in the case of linear models, such as support vector machines [30] or logistic regression [11]. On the other hand,

the single-layered perceptron on which the Neural Networks are based is itself a linear model, but the hidden layers equipped with additional optimisation hyperparameters and attributes make them into black boxes. Often, classical models such as support vector machines [30] are used to study the Decision boundary of a Neural Network. They are also integrated into the Network architecture, such as loss functions, to improve the performance of the models. Further Model capacity in classical systems can be quantified through various statistical methods, such as the Vapnik-Chervonenkis dimension or VC dimension [31, 32].

Definition. *The Vapnik-Chervonenkis dimension [31, 32] of C, denoted as $VCdim(C)$, is the cardinality of the largest set S shattered by C. If arbitrarily large finite sets can be shattered by C, then $VCdim(C) = \infty$.*

The capacity of a learning algorithm often fails to capture its generalisation due to the idea that for a given training set, models with greater capacity will exhibit a larger gap between the R_{train} and R_{test}. While the VC dimension is standard for binary classical machine learning, it is difficult to extend to multilayered neural networks, primarily due to their non-convex landscapes and multiple local minima. Rademacher complexity [33] provides a more robust measure :

$$R_N := \mathbb{E}_D[\hat{R}_N] \quad \text{where} \quad R_N(\mathcal{F}) := \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \left(\frac{1}{N} \sum_{i=1}^{N} \sigma_i f(z_i) \right) \right]$$

where $\mathcal{F}$ is the set of real valued functions $f : Z \to \mathbb{R}$ and D is the distribution over Z, σ_i is the Rademacher vector. This complexity is used to provide tighter bounds on generalisation error.

2.3 Why Expressivity Is Important ?

Expressivity of a Neural Network refers to its ability to approximate a function. The capability of a machine learning algorithm is affected by the representation of the data it is provided. Representation learning methods are learning algorithms that can extract patterns or variations from data, which can be used to explain the relationships between objects in the data. A deep neural network is often referred to as a representation learning method whose layers learn different patterns and compose them to derive a rich representation of the data. "Expressivity" is an exploration of the nature of this representation.

Many methods have explored the expressivity in neural networks, such as the Random matrix methods, which explore the role of the initialisation in the expressivity of the deep neural network, and the Information bottleneck theorem, which explores expressivity in terms of the flow of information through the deep neural networks. In this section, we will explore some of these methods.

2.3.1 Universal Approximation Theory

The theory of how these networks learn is still pieced together by a number of assumptions. Of the simplest, any function that is differentiable can be approximated as a sum over its derivative (some extending to ∞), the Taylor series. A machine learning model is often defined as a function f approximator. The universal approximation theory [34, 35] is such an approach to understand the functions approximated by the Neural Network paradigm. A number of papers have studied the universal approximation properties of Neural Networks, of which the popular works include the papers by Cybenko [34] and Hornik [35]. In the paper "Approximation by superpositions of sigmoidal function", Cybenko [34] showed that a continuous single hidden-layered feed-forward Neural Network with sigmoidal activation function can approximate any arbitrary decision boundary. Based on the same, Stone Weierstrass and the Wiener Tauberian theorem, extended to include a number of activation functions such as sine, cosine and exponential functions. The paper by Hornik [35] showed that Neural Network (activations here are squashing functions such as sigmoid, hyperbolic tangents) can approximate the class of "Borel measurable functions" between finite dimensions to a desired accuracy provided *sufficient hidden units*. The

universal approximation theory in the paper by Cybenko [34] and Hornik [35] focuses on proving the existence of Neural Networks that are capable of approximating a pool of measurable functions. This raises many questions, such as how many parameters are required to fit a decision boundary. Cybenko, in the paper, concluded that an "astronomical" (∞) number of neurons would be required to solve (or achieve a particular desired performance) practical real-world data due to the curse of dimensionality. One way to understand this problem is to realise that the approximation of the function improves with the increase in the summation of terms (similar to the Taylor series) and thus forms an infinite number of parameters. This fundamental problem in Neural Network with respect to the trade-off between the parameters of the model and its predictive accuracy is often studied from the perspective of the "width" and "depth" of Neural Network (see chapters 4& 6, which we study with respect to algebraic topology.

2.3.2 Initialisation Of Weight Space

In this work, we are not interested in the distribution used to initialise the weights of the Neural Network (weight space). Nevertheless, Weight space initialisation [2, 8, 7] is an important part of understanding and tackling issues related to learning in Deep Neural Networks. For example, a Gaussian distribution Wassermann04,NFLT1996,duda2000 as an initial weight space is often exploited in theoretical studies, such as in interpreting the expressivity of Neural Networks as chaotic phase [36]. Weight space in particular influences the learning dynamics, such as the disappearing gradients, exploding gradients or convergence rate in the depth, this is often achieved by *preserving a balanced variance between the layers*. The influence of the weight space is studied with relation to the dynamics of learning [27] of the Deep Neural Network. In the paper [27], the authors explore a number of factors such as depth, convergence, pretraining, etc. In particular, the paper [27] shows that random orthogonal networks have depth-independent learning properties and can ensure learning by preventing gradient saturation in Deep nonlinear networks, provided the networks operate within a regime called the "edge of chaos". Training of the Neural Network is further constrained by the optimisation, including the multiple local minima/-maxima, loss function [10]. On the other hand, the question of the number of parameters reduces to that of depth-width comparison and is based more on the expressivity or capacity of the Neural Network. This is due to the fact that the number of neurons often determines the approximation ability of the Neural network, with "infinity" being the idealistic (but infeasible) existing solution for measurable functions. Towards the same end in this paper, an empirical analysis of two architectures, ResNet and DenseNet, is performed.

2.3.3 Information Theory

Information theory was developed by Claude Shannon (1940 [37]) with the intention to quantify a discrete information source. "It is a measure of the uncertainty involved in the selection of the event." Entropy is defined as the negative expectation of the probability mass/density function as the logarithm of the set of probabilities. Further incorporated is the conditional probability that gives a weighted analysis over two events, giving rise to the concept of mutual information. Mutual Information measures the difference in the entropy of an event and its conditional entropy of another event. Thus, it provides a quantification of the notion of the "amount of information" in terms of probabilities. The popular Kullback-Leibler divergence is also related to the expected relative entropy through the marginal distribution.
Information bottleneck theory [38, 39] in the comprehension of Deep Neural Network analyses the information plane based on the input variable and the desired output, measured via mutual information. Information bottleneck analysis optimises the problem by rephrasing the problem statement to approximate the feature that maximises the information about the desired output. Further, it is shown that optimised layers converge to the information bottleneck bound and that the computational advantage of the hidden layers can be attributed to the information compression from the previous layers. The information bottleneck paradigm also provides a visualisation framework for the information plane of the layers in a Deep Neural Network for studying the internal propagation of mutual information. Although

the theory has a reliable probabilistic theoretical analysis, the computation required to approximate the entropy estimates is not feasible. Mutual information is afflicted by the lack of a general, unique solution in its computation due to the problem of estimation of a good enough statistical presentation of the underlying distribution [38]. The analysis of the compression in the internal representations with respect to the optimisation epochs using Stochastic Gradient Descent in a Deep Neural Network suggests the possible existence of many randomised networks capable of optimising the performance. This gives rise to the need to put all such Deep Neural Networks into a single class, and abstract algebra provides a wide range of mathematical concepts to categorise or group objects. In the field of Topology, homology (see chapter 3) is one of the most popular topological invariants that categorises manifolds, and with the emergence of computational topology, it becomes possible to compute the persistent homology of a point cloud, an approximate counterpart to the underlying homology of the manifold of the point cloud.

2.3.4 Topological Expressiveness

While classical approaches to neural network expressivity often rely on universal approximation theorems, recent advancements have shifted toward quantifying expressivity in terms of topology. This transition is supported by the emergence of computational topology, which provides the rigorous algorithmic framework necessary to extract and analyse topological structures from data:

Betti numbers in Expressivity [3] : The authors show that the topological expressivity of an architecture can be used to study the expressivity of a neural network without specification of the activation function (Pfaffian activation functions). In work [3], the authors exploit topological/homological complexity, the sum of Betti numbers, to derive an upper and lower bound on the Network complexity to compare deep and shallow Networks. These theorems provide a basis for approaching the expressivity of a neural network as a function of its homological complexity.

Empirical Analysis of Topological phase transition [40] : The authors use persistent homology to study the homological complexity of the transformation of embedding space in a classification model. In particular, the empirical results extend topological analysis to non-Panfain functions such as ReLU and leaky ReLU. The study looks at the global manifold as it transforms through the network and studies the impact of the change in the activation function. The study summarises that some topological features would require more layers to simplify and would depend on the complexity of the dataset.

Homological Complexity of Functions [29] : The paper proves the homological generalisation theorem that states that an architecture incapable of producing a certain homological complexity will misclassify for a binary classifier. In our paper, we expand the theorem to the subspace of archetypes, tackling the problem of computational complexity as mentioned in the paper. Further, the paper shows empirical results on toy datasets with different homological complexity, demonstrating that depth is relevant for expressing specific topological structures.

Mapper [8] : In the paper, Mapper is used to interpret the weight space of the CNN networks. In particular, the investigation into the expressivity of the neural network by enforcing an idealised topological structure helped improve the model's generalisability to downstream tasks. The results indicate a relation between the "topological simplicity" and test accuracy.

Topological entropy [41] : A theoretical exploration of the Topology of the chaoticity of Dynamic systems is the " Topological Entropy". Topological entropy is defined with respect to the topological structure and is used to measure the complexity of a dynamic system (often related to the Conley Index [42] and VC dimension [31, 43]), with a higher value demonstrating higher chaoticity. In the paper [41], Topological entropy is used to overcome the issues of global nonlinearity and sensitivities in dynamic systems. Within the realm of Neural Networks, studies such as [44] attempt to extend the ability of topological entropy in dynamic systems [45] to the depth-width trade-off [44] in Deep Neural Networks.

Since real-world neural networks need to evolve over time to accommodate changing real-world scenarios, they can be considered to have properties of dynamic systems [45, 44]. Additionally, Neural networks exhibit chaotic behaviour with depth [36], a property also characteristic of dynamic systems. This work leverages persistent homology [5] from computational topology to relate the homological complexity of the representation space of neural networks to study the expressiveness of different real world models. (see Chapter 4 - Chapter 7).

2.4 Conclusion

In the original paper on Perceptron [46], Rosenblatt introduces perceptrons (or neurons), the fundamental building blocks of Artificial Neural Networks, relating them to the properties of the neurons in the sensory organs. It was introduced as a probabilistic model based on the idea that sensory information creates new connections that can be utilised by new stimuli to inform decision-making. Further, it includes some core assumptions based on the biological nervous system, such as:

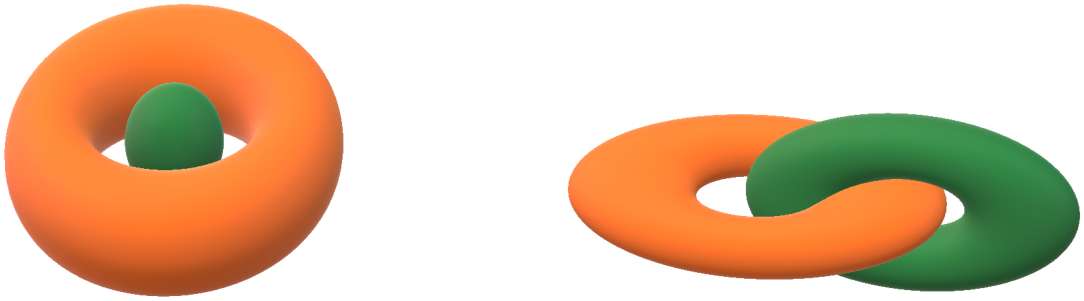
a Learning and Recognition are not identical in different organisms, thus there may exist more than one function that satisfies the relation.

b Plasticity in Neural Networks. This property implies that the probability of neurons affecting each other changes after a period of activity.

The above assumption provides insight into the trainability of a Neural Network as well as its susceptibility. One can argue that, since we have many feasible models, it is not possible to leave it entirely to optimisation. This is, in fact, the current picture of AI, but as AI becomes more integrated into society, the possibility of something going wrong would naturally have a significant impact on the real world, including the potential to endanger life. As it goes in Murphy's law that states "Anything that can go wrong will go wrong". However, from a more scientific perspective, it becomes necessary to understand Deep Networks in Order to sufficiently minimise their potential negative consequences and provide more freedom for applications based on them to safely integrate with society.
On the other hand, the use of mathematical concepts from algebra and topology becomes significant due to their ability to study groups/classes of objects. As in agreement with the assumption (a) above, we can define $\mathcal{F}$ as the set of all functions $f \in \mathcal{F}$ that can satisfy a particular task and dataset. In this research, we provide the pivot into the study of the topology, via persistent homology, of realistic deep Neural Networks through the analysis of the embedding space, input space, and output space. In this research proposal, we conduct a topological analysis of the representations learned by Deep Neural Networks (DNNs). The theoretical study of expressivity provides insight into how to measure changes in the probability of neuron interactions in deep learning models. Furthermore, these techniques help build Deep Neural Networks with improved efficiency and reduced computational cost. In this chapter, we introduce Neural Networks as universal approximators, review previous work on "expressivity," and discuss applications based on this. The proceeding Chapter 3, provides the definitions and notations used for topological invariants and fundamental concepts in computational topology (See Books [4, 5]).

Chapter 3

Background 2: Topological Preliminaries



(a) Solid Torus & Sphere (b) 2 Solid Torus

Figure 3.1: **Betti number** : How do we measure and compare the complexity of two manifolds? Betti numbers ($\beta = (\beta_0, \beta_1, \dots)$) are an example of invariants that can be used to form an inference regarding the homological structure of the space. Example β(Solid Torus) $= (1, 1, 0, 0 \dots)$ and solid sphere as β(Solid Sphere) $= (1, 0, 0, \dots)$. Considering the whole manifold in the figures, then $\beta(a) = (2, 1, 0, 0 \dots)$ and $\beta(b) = (2, 2, 0, 0 \dots)$. Note that Figure (b) above also demonstrates an example for "Links". In particular, here the green and orange manifolds are the Hopf link, the simplest non-trivial link.

3.1 Introduction

This chapter establishes the foundational framework of computational topology and its associated metrics to investigate the topological transition within the embedding space of Neural Network architecture. We introduce the fundamental constructions of persistent homology and address the practical challenges of computational complexity. Given the primary focus on image dataset analysis in this work, we emphasise how datasets with varying structural and dimensional properties are interpreted through different topological complexes. Furthermore, here we explore various subsampling methods that are used to mitigate the curse of dimensionality inherent in topological tools. To provide a basis for these methods, we first examine fundamental topological manifold concepts and understand how machine learning problems are interpreted through a topological lens. Formal definitions for topology, open sets, metric space, continuity, homeomorphism and the construction of homology groups are as in [4]. Unless otherwise specified, we operate within the Euclidean metric space.

3.2 Topology and Homeomorphisms

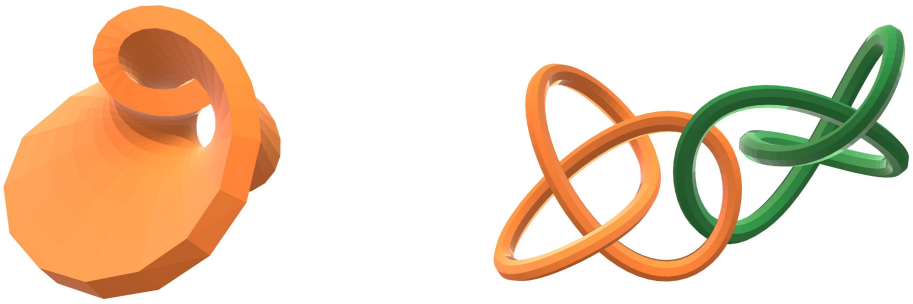
Before diving into the intricate concepts that shape Computational Topology [5], one should have a brief understanding of the mathematical language of Topology. Topology is based on the connectivity of the neighbourhoods, which are preserved under continuous deformations. With respect to a point set, the topological space is one that is equipped with a topology, a collection of subsets ("open sets") based on the point set such that the arbitrary union and finite intersection of the open sets themselves are open sets. (see References from the Books [4, 5] for definition)

Topological concepts play a central role in machine learning, such as the manifold hypothesis assumption [47], where data is assumed to lie in a lower-dimensional manifold. While this hypothesis lacks a universal formal proof, it serves as a critical justification for the use of dimensionality reduction techniques to mitigate the curse of dimensionality.

Manifolds [4, 5] are characterised by intrinsic properties such as dimension, orientation, and homotopy type. For instance, a surface or the generalisation of a "plane" in the $\mathbb{R}^k$ and the famous Möbius band 3.2 ($\beta = (1, 1, 0....)$) are both categorised as a $2-$ manifold, where the index denotes their intrinsic dimension, not the ambient Euclidean dimension. For example, the 2-sphere is intrinsically a 2-manifold whether it is embedded in any higher-dimensional space $\mathbb{R}^k$, $k \leq 3$. Topological invariants, such as homology, allow for the classification of these manifolds into distinct equivalence classes. Such manifolds preserve the homological structure through homeomorphisms [4, 5].

The layers of a neural network are characterised by either homeomorphic or non-homeomorphic properties, depending on the topological transformations required to resolve the objective task. For instance, segmentation [48] typically necessitates homology preserving mappings to maintain the structural integrity of the input manifold. In contrast, classification [3] fundamentally requires non-homeomorphic layers to perform a topological collapse, reducing each class to a contractible space (represented as a single point in the label space).

Topology incorporates notions beyond homology, such as knots and links 3.2, through specialised sub-

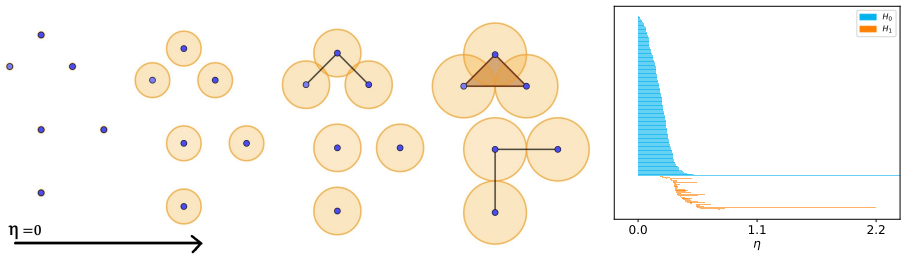


(a) Möbius (b) Knots

Figure 3.2: **Homology vs Knots vs Link**: Topological structures influence the nature of the Neural Network [44, 48, 3]. In a classification model, among the topological structures (knot, link, homology), which do you think will have the most impact on the decision boundary? (see Chapter 7.) The Figure (a) shows a single Möbius strip with Betti number $\beta = (1, 1, 0, \dots)$, while such a manifold is unknotted, one can introduce multiple helical twists to form knots, an example would be the Trefoil knot in the Figure (b). In the Figure (b), the orange and green trefoils represents two different classes and they have a "link" between them, the Hopf link as in the Figure 3.1. While the trefoil is a single connected knot, the link involves two or more knots that do not intersect.

fields, including knot theory. While homology is a strong tool in algebraic topology for measuring "holes" in a space, knot theory focuses on the embedding of circles in three-dimensional space. In the datasets, for the classification problem, class separability is related to the "degree and type" of overlap between classes, which in turn affects model complexity. The topological structure provides a possible method

to describe and visualise this "degree and type" of overlap. Further Topological categorisation of the manifolds is a far better visualisation of the higher-dimensional manifold than visualisation through dimensionality reduction. Since dimensionality reduction methods cannot faithfully represent all the relationships and intricacies of the original high-dimensional structure, there is a loss of information and distortions. On the other hand, homological equivalence helps both categorise and theoretically describe the underlying manifold of the data distribution.



(a) Filtration (b) Barcode of a $\mathbb{C}^1$

Figure 3.3: Persistent homology on a point cloud based on simplicial complexes. As η increases, the sub-complexes are formed by connecting points, and when all the points are connected ($\eta = \infty$), there is only one connected component.

Definition 3.1 (Cardinality)**.** *Let S be a set, then the* ***cardinality*** *or the number of elements in a set S is denoted $|S|$.*

The cardinality gives the number of elements in a set S. For a metric space [4] with metric ρ, one can define the diameter of the subset as follows :

Definition 3.2 (diameter)**.** *if $U \subset S$ with metric ρ, the diameter of A is given by $diam(S) : \sup\{\rho(u, v) : u, v \in A\}$.*

Another popular measure of distance between subsets is the Hausdorff distance, given by :

Definition 3.3 (Hausdorff distance)**.** *Consider a metric space (equipped with the metric ρ then for nonempty subsets U, V of the metric space. We define $\rho(v, U) = \inf_{u \in U} \rho(v, u)$ and $\mathbf{e}(V, U) = \sup_{v \in V} \rho(v, U)$. Then $\mathbf{e}(V, U)$ is called the excess of V over U. The* ***Hausdorff distance*** *between V and U is defined as $\rho_{\mathbf{e}}(V, U) = \max\{\mathbf{e}(V, U), \mathbf{e}(U, V)\}$.*

In this book, unless specified, we use the Euclidean metric space [4] (S, ρ) where ρ is the Euclidean metric [4].

Definition 3.4 (Dataset)**.** ***Dataset*** *($\mathbb{D} = \{(D_i, y_i) : 1 \leq i \leq N \in \mathbb{N}\}$) in machine learning typically includes a feature space D (here subspace of a vector space, for real number space $D_i \in \mathbb{R}^n$) and response/task space $Y/\mathbb{D}_Y$. In a classification problem $\forall i, y_i \in Y, y_i = c$ where $c \in C$ is a finite set with $1 < |C| \in \mathbb{N}$ (natural numbers). C is the index set of all classes. That is, in a classification problem, there exists an onto map from $C \to Y$ while in regression $C = \mathbb{R}$ and $Y \subseteq R$. Let for some $c \in C, \mathbb{D}_c := \{(D_i, y_i) \in \mathbb{D} : y_i = c\}$, then D_c denotes a class feature space in D and Y_c is the response space for the class.*

Consider the STL10 [49] dataset, its $\mathbb{D}$ consists of digital *RGB* images, and C is a finite set with 10 different categories, each describing the "object" in the image. Further in the digitalisation of images, each image is represented as a matrix. For example, for a greyscale image, the spatial coordinates are 2-tuples of the row index and column index, and the value at each spatial coordinate is the intensity (or pixel) value of the image. $\mathbb{D}$ in the real world is always finite, and its underlying probability distribution is given by $\varphi(D)$. In our mathematical proofs, the point cloud D is embedded in some appropriate Euclidean space.

Remark 3.1. *Throughout this thesis, we say a classifier is non-homeomorphic with respect to the class manifold. This is because, in a classification task, each point associated with a class is expected to collapse [40] to a single point or a discrete label. Since this "collapsing" mapping is not bijective, it cannot be a homeomorphism. While the inclusion functions between stages of the filtration induce homomorphisms between homology groups, these maps may be trivial if the underlying topological features are contracted during the classification process.*

The non-homeomorphic nature of the maps is quantified through changes in the topological invariant. The following definitions are adopted from the Books [4, 5]

Definition (Topological invariants)**.** ***Topological invariants*** *are mathematical quantities associated with a* ***topological space*** $(\mathcal{X}, X)$*, where* X *is the topology associated with the set* $\mathcal{X}$*, which do not change under continuous deformations [5, 50] (homeomorphism) .*

In the context of this work, these invariants allow us to classify the structural stability of data representations as they pass through different layers of a neural network.

Definition 3.5 (simplicial complex)**.** *A* ***simplicial complex*** Γ *is a set whose elements are simplices. A* 0*-simplex is a point, a* 1*-simplex is a line segment, a* 2*-simplex is a filled triangle, and a* k*-simplex is the* k*-dimensional counterpart. The characterizing properties of* Γ *are that (a) for a simplex* $\gamma \in \Gamma$*, all its faces are also in* Γ*, and (b) for simplices* $\gamma, \hat{\gamma} \in \Gamma$*, either* $\gamma \cap \hat{\gamma}$ *or it is a shared face of* γ *and* $\hat{\gamma}$*. Formal sum in mathematics refers to the sum of elements from a basis set.*

This combinatorial structure serves as the discrete approximation of the underlying manifold of the high-dimensional image dataset.

Definition 3.6 (Chain)**.** *A* **k-*chain*** *is a formal linear combination of* k*-simplices of* Γ *with real-valued coefficients. The* k*-chains of* Γ *form a vector space denoted* $\mathcal{C}_k(\Gamma; \mathbb{R})$*, which we shall also simply refer to as* $\mathcal{C}_k$*. The* ***boundary map*** $\partial_k : \mathcal{C}_k \to \mathcal{C}_{k-1}$ *maps a* k*-chain to its* $(k-1)$*-dimensional boundary through a formal linear combination of the boundaries of each of its constituent* k*-simplices. The chains along with boundary maps* $(\mathcal{C}_k, \partial_k)$ *form an algebraic* ***exact sequence*** *with* $\partial_{k-1} \circ \partial_k = 0$*:*

$$\{0\} \hookrightarrow \mathcal{C}_n \xrightarrow{\partial_n} \mathcal{C}_{n-1} \xrightarrow{\partial_{n-1}} \cdots \xrightarrow{\partial_2} \mathcal{C}_1 \xrightarrow{\partial_1} \mathcal{C}_0.$$

Definition 3.7 (simplicial map)**.** *Given simplicial complexes* Γ, Γ'*, a* ***simplicial map*** $g : \Gamma \to \Gamma'$ *linearly maps simplices of* Γ *to simplices of* Γ'*.*

We use simplicial maps to model how topological features are transformed or collapsed as data is mapped from the input space to the embedding space.

Definition 3.8 (Homology)**.** ***Homology*** *of a topological space* $\mathcal{X}$ *is defined to be* $H(\mathcal{X}) \coloneqq \{H_k(\mathcal{X})\}_{k=0}^{\infty}$ *where* H_k *is the* k^{th} *homology. The homology* H_0 *represent the connected components,* H_1 *holes,* H_2 *voids, and* H_k *their* k *dimensional counterparts.*

A formal definition of homology theory is as in the Book [4, 5].

Definition 3.9 (Betti number)**.** *For a topological space* $\mathcal{X}$*, the* k^{th} ***Betti number*** (β_k) *is the dimension of* k^{th} *homology or* $\dim H_k(\mathcal{X})$*.*

Betti numbers serve as fundamental topological invariants that characterise the homological structure of a manifold. These invariants are utilised to investigate the representational capacity and topological transitions across diverse neural network architectures. We represent the Betti numbers of a topological space $\mathcal{X}$ as follows $\beta(\mathcal{X}) = (\beta_0, \beta_1, ..., \beta_k)$, but to account for data realised in higher dimensional topological space, we use the generalised Betti number notation $\beta(\mathcal{X}) = (\beta_0, \beta_1, \dots)$.

Definition 3.10 (Euler characteristic)**.** *The Euler characteristic of* $\mathcal{X}$ *can also be obtained as* $\chi(\mathcal{X}) = \sum_{k=0}^{\infty}(-1)^k \beta_k$*.*

The Euler characteristic offers a singular global value that captures the integrated complexity of the space. The homological complexity is similar to the idea of the Euler characteristic.

Definition 3.11 (Filtration)**.** *In this work, we denote the* ***data*** *as $D \subset \mathbb{R}^p$ and its associated topological space by $\mathcal{X}_D$.* ***Filtration*** *F_g is the $\{\mathcal{X}_\eta\}_\eta$ nested sequence of subspaces of the Topological space $\mathcal{X}$ connected by inclusion determined through a real-valued function $g : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X}_\eta = g^{-1}(-\infty, \eta]$ and $\eta \in \mathbb{R}$.*

In our experiments, the filtration parameter typically corresponds to the proximity scale at which individual data points are considered connected.

Definition 3.12 (Persistent homology)**.** ***Persistent homology (PH)*** *is a topological tool that uses filtration to construct subcomplexes Γ_η to extract the homology of a point cloud D (see Figure 3.3(a)). Here, η is the* ***threshold*** *associated with the filtration function.*

Note that not all features extracted through persistent homology are a homology of the topological space $\mathcal{X}_D$. We will consider that the persistent homology was computed on the point cloud until only a single connected component remains. The features extracted from persistent homology $\widehat{H}$ can be represented in the form of barcodes or persistence diagrams (dgm).

Definition 3.13 (Persistence diagram)**.** ***Persistence diagram(****dgm****)*** *for a point cloud D is a series of points in the extended plane $\mathbb{R}^2$ that represents the birth and death information of the homology that appears in the subcomplex during filtration. Further, the η at which i^{th} homology ($\widehat{H^i}$) appears in the dgm is the* ***birth****, denoted as b_i or $b_i(\eta)$ and the η at which the homology disappear is the* ***death*** *denoted as d_i or $d_i(\eta)$. We denote all the homology that arises in the $dgm(D)$ as $\widehat{H}(D)$.*

The significant persisting homology is the possible homology of the underlying topological space $\mathcal{X}_D$ of the point cloud D. The dotted line in the dgm (see Table 3.1) represents ∞, and the points at this line represent homology that never dies.

Definition 3.14 (Barcode)**.** *A* ***barcode*** *of the persistent homology is a collection of bars that represents the persistent homology classes (see Figure 3.3 (b)).*

The barcode is unsuitable for representing persistent homology for high-dimensional data, such as images.

Definition 3.15 (Betti-Curve)**.** *The k^{th}* ***Betti-curve*** *$\widehat{\beta}_k$ (Table 3.1) maps dgm to an integer-valued curve for different values of η. In the Betti-curve diagram, the x-axis represents η, and the y-axis represents the number of k^{th}-homologies for the subcomplex Γ_η that are still persisting at a particular η. The k^{th}* ***Betti number*** *at Γ_η is the value of k^{th} Betti-curve at η denoted by $\widehat{\beta}_k(\eta)$.*

Betti curves enable us to treat topological signatures as functional data, which can then be used as input for further statistical or machine learning models. For a N-point cloud, we have $\widehat{\beta}_0(0) = N$ and $\widehat{\beta}_0(\eta) \to 1$ as $\eta \to \infty$. Similarly, in a N-point cloud for some η', $\widehat{\beta}_k(\eta') \geq 0$ and $\widehat{\beta}_k(\eta) \to 0$ as $\eta \to \infty$ (see Table. 3.1). Ideally, the smaller the max η for which β_0 dies, the closer the points are in the point cloud.

Definition 3.16 (Homological complexity)**.** *We define the* ***homological/topological*** *complexity of a topological space $\mathcal{X}$ as the sum of Betti numbers, that is, $\omega(\mathcal{X}) \coloneqq \sum_k \beta_k(\mathcal{X})$. We denote the homological complexity at Γ_η of the dgm of D as $\omega_\eta(D) = \sum_k \widehat{\beta}_k^D(\eta)$.*

This metric serves as a topological proxy for the model capacity required to learn or represent a given dataset.

Definition 3.17 (Lifespan)**.** *The* ***lifespan*** *of a homology $\widehat{H^i}(dgm)$ that determine its persistence is given by $|b_i - d_i|$.*

Longer lifespans generally indicate more robust geometric features that characterise the global structure of the point cloud. However, the debate on how to justify the "length" as significant is still ongoing.

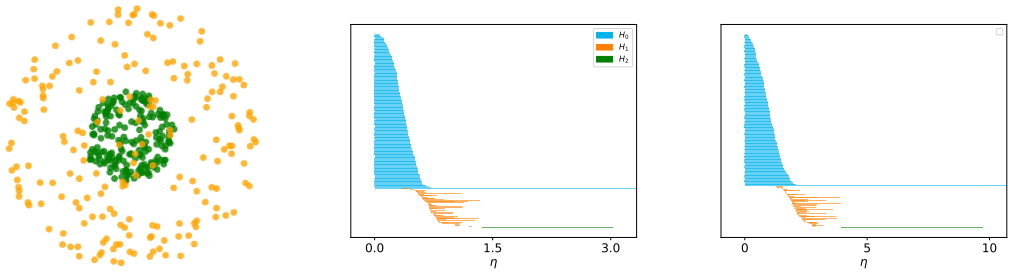
Definition 3.18 (Average life)**.** *The* ***topological average life*** *of the dgm of D is given by $\lambda_D = \frac{\sum_i |d_i - b_i|}{|\widehat{H}(D)|}$ for i^{th} homology that appear in the dgm with finite lifespan. For points that never die, their lifespan is ∞, and we have excluded them from our definition. Further, note that both lifespan and λ are functions of η.*

We utilise average life to summarise the overall persistence of topological noise versus signal within the network embedding space.

Definition 3.19 (Big O notation)**.** *For computational complexity, the Big O notation (O) or Bachmann–Landau notation will be used. It properties include $g_1 = O(h_1), g_2 = O(h_2)$ and c is a constant then $kg_1g_2 = O(g_1g_2)$ and $kg_1 + g_2 = O(\max(|g_1|, |g2|))$.*

3.3 Foundations of Persistent Homology

This section discusses the importance and limitations of the stability theorem for persistent diagrams, the computational complexity and data agnosticity of persistent homology [5] associated with different complexes, and finally, the existing approaches used to tackle them. In this section, we denote a (n-1) sphere as $\mathbb{C}^{n-1}$.



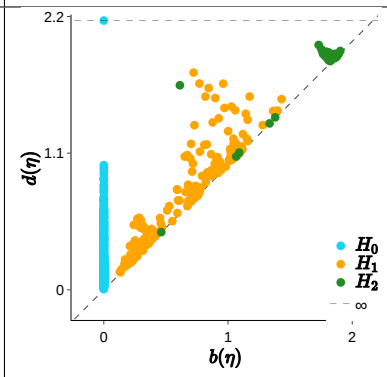
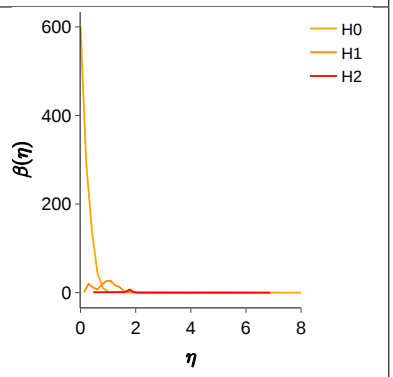
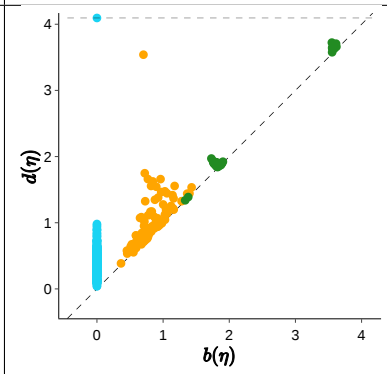
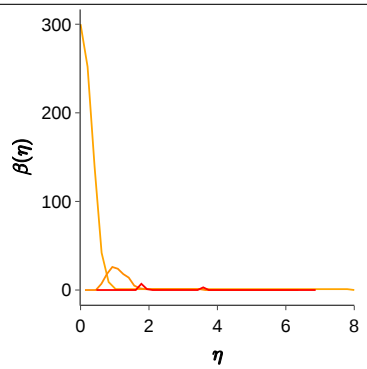
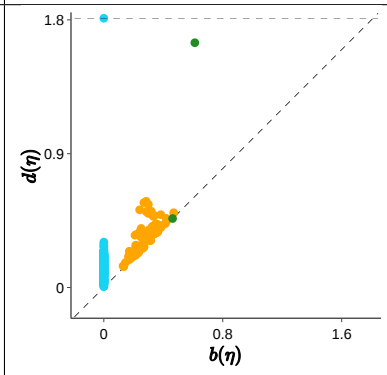
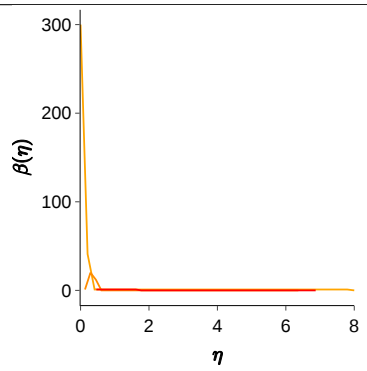
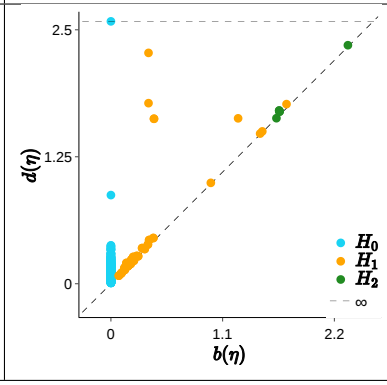
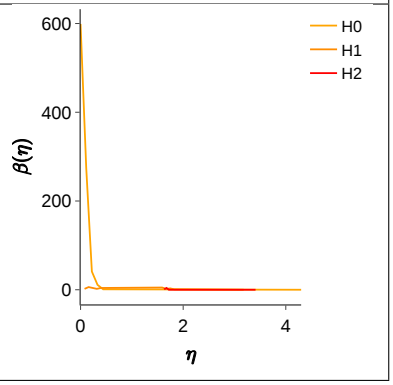
(a) Sphere (b) $\mathbb{C}^2(2)$ (c) $\mathbb{C}^1(6)$

Figure 3.4: The Figure (b)&(c) are the barcodes of the orange and green spheres in the Figure (a), respectively. The persistence "bars" in both the Figures (b)&(c) are the same as the homology of a sphere $\beta = (1, 0, 1)$, but the bottleneck and Wasserstein distances between the *dgm* are not zero. Thus, these measures on the *dgm* are not distance agnostic. For simplicity, we denote the spheres as follows $\mathbb{C}^{n-1}$.

3.3.1 Stability of Persistent Diagrams

The "persistence" of homological classes observed through persistent homology is considered as a measure of its significance. The longer persisting "bars" or lifetime in the barcode (see Figure 3.3 (b)) have a higher chance of representing the actual homology of the underlying topological space as compared to the other. Nevertheless, it is not always possible to determine this significance through visualisation or by choice of η (possible only when there is a clear gap between the longer and shorter lifetimes of the homology). Due to the same reason, here $\widehat{\beta}$ is used for the number of homology observed through persistent homology. Further statistical measures based on the observed homology in the diagram are also susceptible to persistence. Thus, it becomes essential to understand the concept of comparing homology extracted using persistent homology.

Table 3.1: Persistence Diagrams and Betti curves extracted through persistent homology. The datasets represent: (1) a disjoint union of a torus and a sphere, (2) a torus, (3) a sphere, and (4) two disjoint figure-eight symbols in $\mathbb{R}^3$.

Space $\mathcal{X}$ $(\beta_0, \beta_1, \beta_2)$	Persistence Diagram	Betti Curve $\widehat{\beta}(\eta)$
Torus ⊔ Sphere $(2, 2, 2)$		
Torus $(1, 2, 1)$		
Sphere $(1, 0, 1)$		
Two Figure-Eights $(2, 4, 0)$		

"Stability theorem for persistent diagrams" [51, 5] study the notion of change in diagram under the function g that induces $\mathscr{F}_g$. Let g_1 and g_2 be two simplex-wise monotone function on the simplicial complex K, with persistence diagrams $dgm(\mathscr{F}_{g_1})$ and $dgm(\mathscr{F}_{g_2})$ (for simplicity denoted by dgm_{g_i}) respectively. Let $G = \{\hat{g} : dgm_{g_1} \to dgm_{g_2}\}$ be the set of bijection function between the diagrams. For $g_1, g_2 : K \to \mathbb{R}$, the infinity norm is defined by $||g_1 - g2||_\infty$. The Bottleneck distance (ρ_b) 3.5 between the two diagrams and the stability theorem for all k is given below by :

$$\rho_b(dgm_{g_1}, dgm_{g_2}) = \inf_{\hat{g}} \sup_{p \in dgm(g_1)} ||p - \hat{g}||$$
$$\rho_b(dgm_{g_1}, dgm_{g_2})_k \leq ||g_1 - g_2||_\infty$$

The theorem can be extended to distance functions such as Wasserstein distance [5] for Lipschitz functions (g_1 and g_2) and for sliced Wasserstein distance on the $p-$ norm over g_1 and g_2. Sliced Wasserstein is slightly more computationally feasible for real-world applications (See Figure 3.5). The p-Wasserstein is given as follows.

Definition. *The (q, p)-th Wasserstein distance between two persistence diagram $dgm(A)$ and $dgm(B)$ is given by :*

$$\rho^p_{w,p}(A, B) := \inf_{\Pi : dgm(A) \to dgm(B)} \left[\sum_{x \in dgm(A)} ||x - \Pi(x)||^p_q \right]^{\frac{1}{p}}$$

Note that the p-Wasserstein distance is $\rho^\infty_{w,p}$ for simplicity, we denote the same as $\rho_{(w,p)}$.

The stability theorem is what makes it possible for persistent homology to be practical and summarises that under "small" perturbations, the *dgm* remains consistent. It is important to bring to the attention of the readers that the distance measures are susceptible to data and often perform better under normalisation of the data between 0 and 1. For example, see the bottleneck and Wasserstein distance for the point cloud of the same size (200 points) sampled from $\mathbb{C}^2(3)$ and $\mathbb{C}^2(6)$ Figures 3.5. Although the homology of the *dgm* is the same as observed from the Figure 3.4, ($\sum_k$ the sum over all dimensions k), the bottleneck distance is greater than 5, and the Wasserstein distance is greater than 50. This can be somewhat overcome by normalising the point cloud to the range 0 to 1.

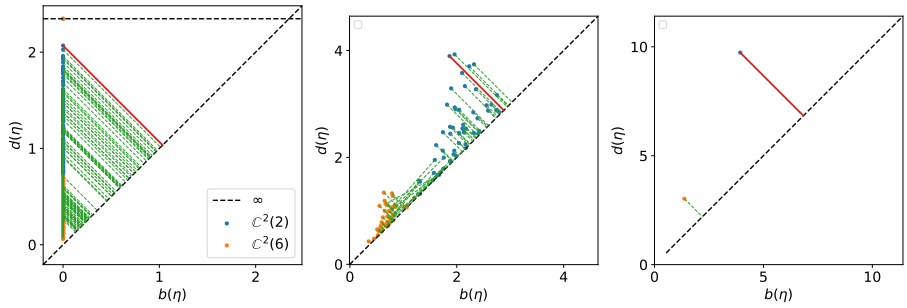
3.3.2 Persistence Filtrations

Simplicial complexes are constructed on the point cloud based on the function g, and their connectivity affects the homological information extracted. Thus, we have different complexes, such as Čech ($\mathbb{CH}$) and Vietoris–Rips ($\mathbb{VR}$) complexes built from simplices. Restricting to the Euclidean space and dataset as a union of open balls centred at the data points (D_i). For a finite set of points D $\in \mathbb{R}^n$, the "interleaving" relationship between the two complexes is intuitively given by :

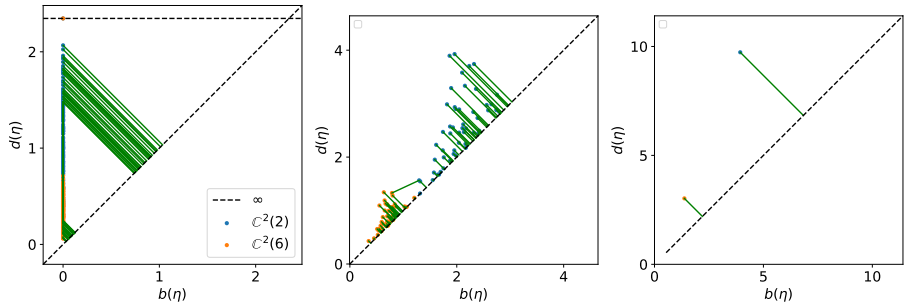
$$\mathbb{CH}^r \subset \mathbb{VR}^r \subset \mathbb{CH}^{r'} \text{ where } r' = \sqrt{\frac{2n}{n+1}} r$$

Čech [50] : This complex requires the computation of all possible intersections of the balls **B**, which is infeasible for tracking embedding space when training a Neural Network, which requires large data for learning. The computational complexity is exponential as given in the Table 3.2, where $\bar{n}_K$ is the average number of neighbours in each simplex and N is the number of simplices. Note that when computing the persistence diagram, the dimension of simplices is proportional to the dimension of the data.

Vietoris–Rips [5] : This complex is widely used as an approximation of the $\mathbb{CH}$ due to their "interleaving" relationship based on the radius of the open ball and the reduction of the computation to the pairwise distances of the points. But $\mathbb{VR}$ in turn contains more higher-order simplices than $\mathbb{CH}$. At the algorithmic level, most existing work makes use of $\mathbb{VR}$ and approximates the $\mathbb{VR}$ through methods such



(a) $\widehat{H_0}$ (b) $\widehat{H_1}$ (c) $\widehat{H_2}$



(d) $\widehat{H_0}$ (e) $\widehat{H_1}$ (f) $\widehat{H_2}$

Figure 3.5: The Figure (a,b)&(c) and Figure (d,e)&(f) represent Bottleneck and Wasserstein distances for the persistence diagrams of $\mathbb{C}^2(3)$ and $\mathbb{C}^2(6)$ in Figure 3.4 respectively.

as hybrid algorithms, dividing the data into sub-regions, etc. The computational complexity is based on the number of simplexes, where n_K is the total number of simplexes ($K = \cup_{k=0}^{\infty} K_k$ where K_k is the collection of $k-$simplexes and the construction of $k-$simplices requires $O(N^{k+1})$ for every N points. In the worst case, the computational complexity of the traditional algorithm is $O(2^{3|D|})$, making $\mathbb{VR}$ more suitable for high dimensional point clouds with a smaller number of points.

Delaunay [50] : This complex subdivides the space to form a collection of covers for the point space called the Voronoi decomposition. These in general lack a geometric realisation with worst case time complexity for $D_i \in \mathbb{R}^n$, given by $O(|D|^{n/2})$.

Cubical [52] : Used for grid data structures, the cubical complex is particularly suitable for extracting the homology from a greyscale image, and in this work, it is put to use to study the embedding space of an individual image to understand the homological transitions in the layers of the Neural Network. The advantage of a cubical complex over a simplicial complex is given in the size of the complex. For example, the ratio of co-faces of the vertex of simplicial to cubical is 6:4 and 26:8 for 2 and 3-dimensional complexes, respectively. The computational complex for $k-$cube and N the input size

Flag [50] : The construction complexity of Flag complex depend on the k the number of vertices and $|D|$ the number of input points. A flag complex is a simplicial complex where each pair of vertices belongs to some simplex. The flag complex is useful for graphical data structures because of its natural relationship with clique graphs.

Complex name	Used for	Time	Storage
Čech complex	Points in metric space	$O(N^2 + N2^{\bar{n}_K})$	$O(2^K)$
Vietoris–Rips	Points in metric space	$O(n_K^{2.4}$	$O(n_K^2)$
Cubical	grid structure	$O(k3^k n_c + k^2 2^k N)$	$O(k2^k N)$
Witness	Point in metric space	$O(K + W)k^2 \log(L)\|$	$O(2^L)$
Delaunay	Point	$O(\|D\|^{d/2})$	$O(\|D\|^{d/2})$
Flag	Directed networks in graphs	$O(\|D\|^k k^2)$	$O(n_K^2)$

Table 3.2: Different complexes influence the computational complexity in persistent homology [53]. Each method attempts to capture the underlying topology, with the Čech complex being the most conventional and computationally complex. Triangulation exponentially increases the number of building blocks compared to hypercubes, making cubical complexes more computationally efficient.

The complex used to construct the filtration in persistent homology plays several important roles, including the homology extracted and the computational complexity (see Table 3.2). An example would be the greyscale digital image dataset, while the grid topology makes it suitable for cubical homology when computing "homology" for individual images.

Remark 3.2. *In this thesis, various types of complexes are employed for experimentation, with the choice of structure determined by the nature of the input data. While the specific data structures are detailed in the experimental section, the core definitions are established using simplicial complexes and extended to other frameworks as needed. Furthermore, concepts central to persistent homology, such as Betti numbers, are applied directly across these different complex types. The specific complexes utilised in this work include Vietoris-Rips complexes and Cubical complexes.*

As the real-world datasets in this thesis consist of images, investigating the underlying manifold transitions within a neural network involves treating each image as a discrete point in a high-dimensional point cloud. This is achieved by flattening pixel intensities into vectors within a Euclidean space. In this framework, a simplicial complex is typically constructed to recover the homological features of the point cloud. However, since the construction of such complexes, particularly the Vietoris-Rips using traditional persistent homology, is computationally expensive as the cardinality of the dataset increases, the use of subsampling or data reduction techniques becomes essential. In the next section, we look at some of the existing subsampling methods.

3.3.3 Subsampling methods

As observed, persistent homology [53] has high computational complexity due to the same various methods that have been introduced to solve the curse of dimensionality.

Dimensionality Reduction : Uniform Manifold approximation and Projection or UMAP [54] approximates a low-dimensional dataset that preserves the topology through minimisation of the cross-entropy between the manifolds. The method requires the assumption that the manifold is uniformly distributed with respect to the metric of the underlying space. Stereographic projection is a homeomorphism, thus it induces a homological equivalence between the spaces. These hyperspheres visualise the "shape" of high-dimensional data and capture higher-dimensional topological information in lower dimensions. The well-known stereographic projection is the projection of a sphere to a disk/plane.

Neural Network : Self-organising map [55] utilises artificial neural network architectures to produce a dimensionally reduced space that preserves the topology through unsupervised learning. Topological

Autoencoder [56] is another method of dimensionality reduction that uses the architecture of an Autoencoder to produce a low-dimensional representation that preserves the topology of the data through a persistent homology-driven topological loss function.

Complex : Witness complex [57] is a combination of subsampling to construct a complex that extracts the homology from the original data. Towards the same end, the data is split into landmark L and witness W points. The method constructs a simplicial complex (K) using the landmark point with the witness points serving as witnesses to the construction of edges or simplices, and acts as an approximation to the restricted Delaunay triangulation. Alpha complex [58] is a subcomplex of the Delaunay triangulations that preserves the topology of the Delaunay complex and is used to reduce the computational complexity with far fewer simplices. While Alpha and witness complex both approximate the homology of the Delaunay complex, the alpha complex is known for its better topological representation. On the other hand, a witness complex is prone to the choice of landmark points and the density of the dataset.

Persistence Vectorisations : Average persistence Landscape [59] is based on the persistence vectorisation known as persistence landscape. The persistence landscape is also stable under the persistence diagram and can be used to measure the shift in persistence diagrams. Further, the method is more computationally efficient as compared to the Wasserstein or bottleneck distance. The Average persistence landscape is an approximation of the persistence landscape of the data when there is a large number of data points. The average persistence landscape is computed through averaging the persistence landscapes extracted from constructing complexes on a random sampling of the data points. In this work, we extend the statistical analysis on the persistence vectorisation to the entropy-based quantity "persistence entropy".

The computational complexity of Persistent homology is affected by both the dimension and the number of samples in the data. In this work, subsampling is preferred because the idea is to study the topological transition of the embedding space using Vietoris-Rips (which is least affected by the dimension of the data) based persistent homology. It is also interesting that on the theoretical side, the duality theorems for the Betti numbers provide the possibility that the higher-dimensional Betti numbers can be determined from the lower-dimensional Betti numbers. While such duality theorems are well established in classical topology, their practical application within the framework of persistent homology remains a subject of ongoing theoretical study. While these methods allow for the calculation of topological invariants such as Betti numbers, a more global, structural view of the data connectivity is often required. To address this, other topological tools such as the Mapper algorithm play a vital role by providing a multi-scale, graph-based visualisation.

3.4 Other Tools of Computational Topology

The K-mapper or Mapper algorithm [60, 8] is one of the most popular tools for high-dimensional data visualisation. The method combines simplex triangulation, graphs, clustering and topological nerve covering to extract intricate data patterns by partitioning the space into its connected components. Mapper introduce the graph-based topological data analysis and is used in a large number of fields, including interpretability in Neural Network [8]. For example, in the paper [8], the mapper algorithm was used as an approximation of the topological information of the weight space of CNN with different configurations during training. The empirical study suggests a possible relation between the test accuracy and the most persistent homological features, supporting the idea of topological simplicity being a property of generalisation in neural networks. Nevertheless, it is yet to be theoretically established.

Although Mapper algorithms are extensively used for the representation of high-dimensional data, they can fail to capture specific topological features. As shown in Figure 3.6, the algorithm fails to capture the interlocking structure in the data. This limitation often stems from the choice of the filter function, the discretisation of the cover, or the clustering parameters, which may inadvertently collapse essential topological information or introduce additional structures.

(a) Torus & Sphere (b) 2-figure-eight

Figure 3.6: **Mapper** : Shows the images generated by the Mapper algorithm of the (a) a sphere within a torus and (b)linked 2-figure-eight dataset. In the Figure, note that in (a) the torus is not a connected component and (b) is missing the link between the figure-eight realised in R^3.

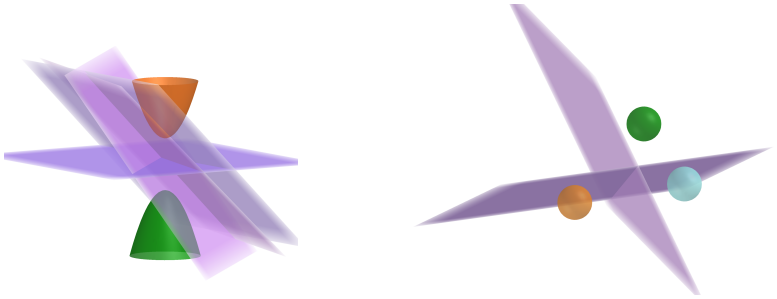
3.5 Conclusion

This chapter establishes the formal mathematical framework required in the following chapters. While topology studies the theoretical concepts of connectedness and manifolds, computational topology contains tools used to extract topological structure from real world data. The algebraic theoretical foundations of Homology and Betti numbers provide a rigorous method for quantifying "holes" and connectivity, which can be utilised to understand the "topological complexity" or the shape of the data. Using algebraic topology and the discrete construction of complexes, computational topology developed the tools necessary to approximate possible topological invariants from different data structures.

The core computational topological tool used in this work is Persistent Homology, which allows us to distinguish between stable topological signals and localised noise through persistence diagrams or persistence barcodes. Measures such as the homological complexity curve and topological average life are used to quantify the homological information to analyse the topological evolution in neural networks. Thus, offering a computable topological counterpart to traditional complexity measures like the VC dimension [31]. These metrics serve as the primary descriptors for the "topological work" performed by a network. It also explores the "role" of topological structure both theoretically and empirically through computational topology. Exisiting packages used for persistent homology include the Ripser [61], Persim, Giotto TDA [62], Gudhi [63], and cubical software [64]. This work transitions from theory to application, which enables us to investigate how the representation learned by the neural network influences its capacity in a pretrained setting to resolve the homological complexity of real world image datasets.

Chapter 4

Comprehending Embedding Transitions in Transfer Learning



(a) 2 class (b) 3 classes

Figure 4.1: **Choosing "f":** The complexity of the decision boundary in classification is influenced by the shape of the classes in the dataset. When considering choosing a neural network architecture, how can knowledge from an existing model be exploited to solve for a new dataset (or downstream or target dataset)?

4.1 Introduction

The interpretation of Neural Network from individual neurons in the Network is "practically meaningless" [65]. Thus, one needs to study the Network as a whole or its layers. Existing work, including in topology, studies network weights [39, 7, 8] directly, while others, such as [36], study them through layer-by-layer transformations of the Neural Network. This work is based on the latter method, studying the topological shape of the transformation. This work presents a framework for investigating the depth and breadth of neural networks in relation to topological transformations. In this work, the outputs of the layers of the Neural Network are called the Embedding space. This chapter in particular explores the relevance of studying the embedding space of the Neural Network in understanding the expressivity and utilising it for the problem of model selection in transfer learning representation methods, which attempt to exploit knowledge from the "source space" to solve for a new space, the "target space".

The chapter begins with a discussion of transfer learning, addressing the critical model selection problem and the existing methodologies employed to identify optimal architectures. This is followed by an exploration of the importance of the embedding space in model selection. Central to this work is the development of a formal theoretical framework to quantify the "topological work" performed by a neural network. We propose Theorem 4.1, which establishes the existence of a critical rate of topological change and defines the homological capacity Φ_f as a metric for architectural sufficiency relative to depth

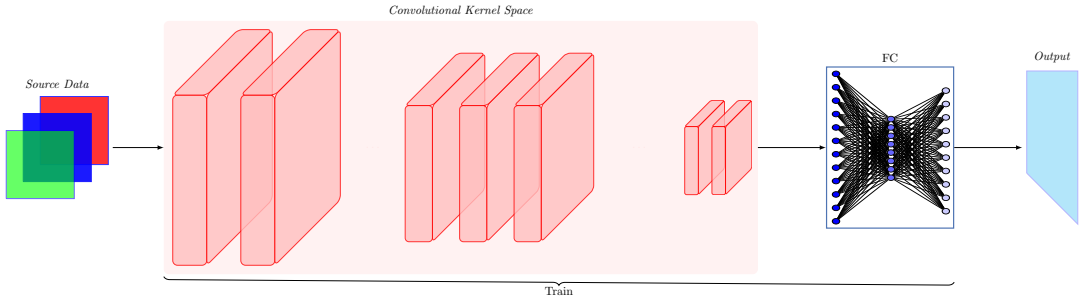


Figure 4.2: **Traditional Training :** In a machine learning task, a model is traditionally trained on a dataset by optimising the weight space (initialised randomly or according to a particular distribution) to minimise the loss function for the task. The task could be classification, segmentation, regression, or another type. The following image represents a simple convolutional architectures with fully connected layers.

L. Building on this, Theorem 4.3 demonstrates that for a fixed dataset complexity, shallower optimal networks pose minimal homological capacity Φ_f, providing a formal criterion for selecting the most compact architecture.

The experimental section subsequently evaluates these claims using cubical homology within an image transfer learning context for classification tasks. This is achieved by deriving practical proxies for the homological capacity Φ_f, which are denoted as θ and $\widetilde{\Theta}$ and approximated using polynomial interpolation. The results and experiments are influenced by the theoretical claims [3] on the topological expressiveness between shallow and deep neural networks. In this the topological score can be considered as a tool of pre-screening, a model selection diagnostic tool that complements on the expressiveness of the model for the domain data space.

4.2 What is transfer learning ?

Neural Networks are known for their ability to extract information from data that can help solve a desired task. This property, which essentially performs the role of dimensionality manipulation, gives the reputation of representing learners. Unfortunately, this requires a large parameter space, which increases the complexity of training and storing the model. One such method is Transfer learning. Transfer learning is a method of representation learning that belongs to a different category from the traditional method (See Figure 4.2), "borrowing" knowledge from a pre-existing model. There are various methods of transfer learning, such as Transductive, Inductive, Unsupervised, Homogeneous, and others. In terms of a Source space (Source dataset $\mathbb{S}$) and a Downstream/Target space (Downstream dataset $\mathbb{DS}$), transfer learning is the method of improving the learning of $\mathbb{DS}$ using $\mathbb{S}$. In transfer learning, one keeps either the learning task or the dataset similar for the Source and Downstream. When both are the same, then the problem reduces to traditional machine learning. Note that even initialising with the weights of a trained model [2] helps improve training.
Based on the source and downstream space, transfer learning can be categorised into transductive and inductive transfer learning. In an inductive learning setting, the assumption is that the source and downstream spaces have different but related tasks. For example, consider the tasks of classification and object detection on the same dataset. On the other hand, transductive learning assumes a different but related feature space for a similar task. For example, a sentiment analysis model applied to English movie reviews and then extended to French movie reviews. In this work, the inductive method of learning is explored. The model is trained on a source task that involves classifying images from the 1000-class ImageNet dataset [66] using a Convolutional Neural Network (CNN). The acquired knowledge is then transferred to a related downstream task: classifying image datasets with varying numbers of categories. The model trained on the source dataset is defined as the pre-trained model. An example of inductive

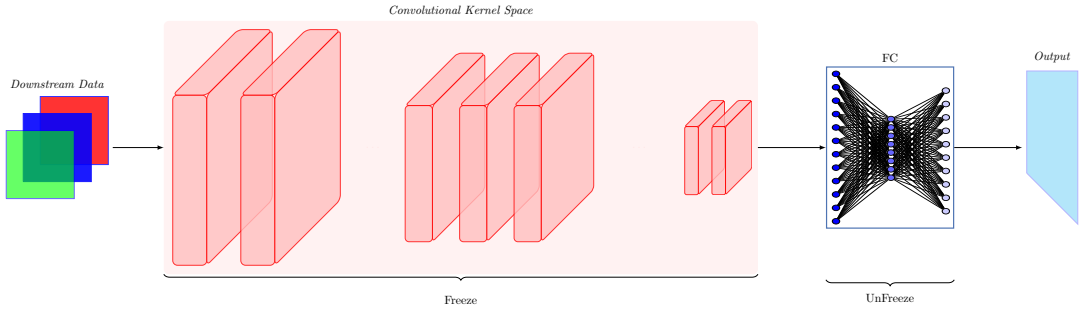


Figure 4.3: **Fine tuning :** Various methods can be employed to enhance the performance of the model on downstream data. Fine-tuning is one such application of transfer learning that utilises the "partial training" of the Neural Network (often the fully connected layers). In this study, we aim to derive a metric from the embedding space that correlates with fine tuning accuracy, providing a measure of the adaptive capacity of the model.

learning is fine-tuning, in which a part of a pre-trained model is trained on a downstream task to improve the model's performance (see Figure 4.3). The parts of the model that are not trained are said to be kept "frozen" during training, for example, in the Figure 4.3, the terms "freeze" and "unfreeze" are used to refer to parts of the model that are not trained and trained, respectively, during fine-tuning. Here, the assumption is that they have the same data probability distribution, such as the mean and variance. Note that this would require information about the source dataset for training the model.

4.3 Model selection

Consider the dataset $\mathcal{D}$ and let $\mathcal{M}$ be a collection of models. Then the model selection problem is an attempt to find the best model from $\mathcal{M}$ such that the performance as well as the resources are best optimised, that is, a model with the best performance and least resource consumption. This problem has real-world consequences due to the large number of models and the high dimensionality of the available datasets. Here, the models are Neural Networks with different architectures and parameter settings. During neural network model selection, their inherent 'black box' nature renders the acquired patterns incomprehensible. Consequently, identifying an optimal configuration requires an iterative process of training multiple candidates from scratch. However, Neural networks have ample parameter space and training data, making traditional training methods impractical for every dataset. Now, if one were to consider fine-tuning a pre-trained model, one could ask: which parts of the model should be unfrozen for training? In this work, only the fully connected (see FC in the Figure 4.3) part of the Neural Network is unfrozen for training. Note that there are better fine-tuning [67] methods that involve techniques such as regularisation, intermediate task training, and others.

4.3.1 Transfer learning Assessment Score

A primary challenge in transfer learning [68] lies in the identification of an optimal pre-trained model. Model selection is a thankless and burdensome task, given the large number of models, the possibility that the best model is significantly smaller when trained from scratch, and the fact that the selected model may be prone to over-fitting. Neural network model selection is inherently challenging due to the expansive hyperparameter space, which includes attributes such as architectural depth and the choice of optimiser.

Definition 4.1 (Transfer Learning)**.** *On the basis of multiple Source* $\mathbb{S}_j$ *and downstream dataset* $\mathbb{DS}_i$*, the premises of the model selection with inductive transfer learning are as follows assuming finite collection of Neural Network* $\mathcal{F} = \{f_i\}, i \in \{1, 2, \ldots N_{\mathcal{F}}\} \subset \mathbb{N}$*, source datasets* $\mathcal{S} = \{\mathbb{S}_i\}_i, i \in \{1, 2, \ldots N_{\mathcal{S}}\} \subset \mathbb{N}$

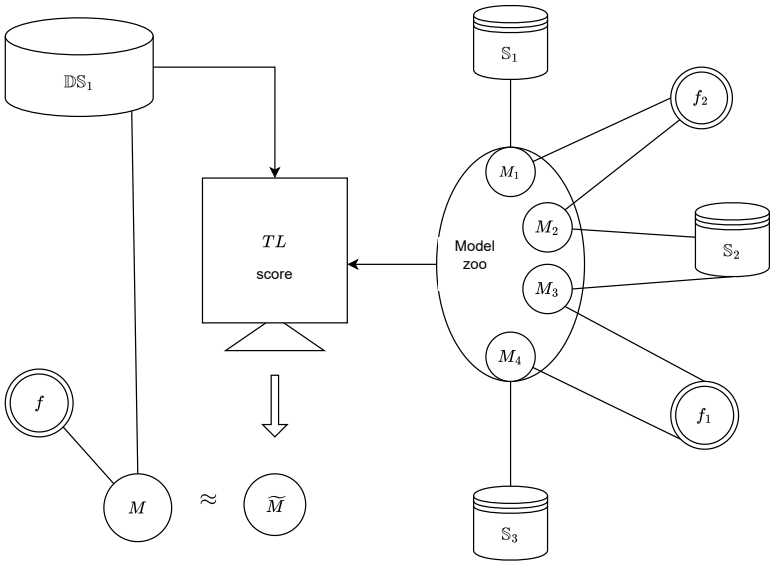


Figure 4.4: **Transfer learning assessment (TL score):** In this work various TL scores are used to assess the best model from $\mathcal{M}$ that approximate the function f for downstream dataset $\mathbb{DS}$. In the Figure each model M_m is the resulting model when the function f_i is trained on the Source data $\mathbb{S}_j$.

and downstream datasets $\mathcal{DS} = \{\mathbb{DS}_i\}_i, i \in \{1, 2, \ldots, N_{\mathcal{DS}}\} \subset \mathbb{N}$. Let us denote the pre-trained models as $\mathcal{M} = \{M_i\}_i, i \in \{1, 2, \ldots N_{\mathcal{M}}\} \subset\in \mathbb{N}$ where $M_i = (f_j, \mathbb{S}_{\widehat{j}})$ is the Neural Network f_j trained on the source dataset $\mathbb{S}_{\widehat{j}}$. Then the transfer learning score ($TL(\mathbb{DS}_j, M_i)$) is used to find the best model for the downstream task $\mathbb{DS}$, denoted as $TL(\mathbb{DS}_j)$:

$$TL(\mathbb{DS}_j) := best_i\{TL(\mathbb{DS}_j, M_i)\}$$

Given above is the definition for Transfer learning Assessment Score (TL_j) for downstream dataset $\mathbb{DS}_j$ (see Figure 4.4). The term "best" is used in the equation as TL can be any ordering (Ranking) score. For example, if TL takes values in the range 0 to 1 and if 0 implies the best model, then "best" can be replaced by min.

$$TL(\mathbb{DS}_j) := \min_i\{TL(\mathbb{DS}_j, M_i)\}$$

On the other hand, if 1 implies the best model, then 'best" can be replaced by max. To establish a robust model ranking (see [Figure 4.5]), the TL must align with objective evaluation criteria, such as test accuracy. The validity of this relationship is assessed by quantifying the correlation between the TL score and the observed model performance. Correlation [15] ($Corr$) is the statistical measure that determines the strength of the relation between two random variables or bivariate data. In this chapter, Pearson correlation [15] is used to validate the relation. To further validate the statistical significance of the Pearson correlation, the p-value hypothesis test is used. The following are examples of some TL scores, based on statistical information from the task and feature space.

NCE : Negative Conditional Entropy (NCE [70]) is based on the assumption of optimal loss on the source and downstream tasks with a cross-entropy loss function. The theoretical proof of the relation between transferability and conditional entropy is based on measuring the information required to estimate the response $\mathbb{DS}_Y$ given $\mathbb{S}_Y$, assuming that $\mathbb{DS}_D = \mathbb{S}_D$. A "hardness" measure is defined to respect the capacity of the neural network and is approximated by the conditional entropy of $\mathbb{DS}_Y$ given a trivial task (or constant sequence). It measures how hard it is to transfer from a trivial task to the downstream task. Under the following setting, when there is a one-to-one correspondence between $\mathbb{DS}_Y$ and $\mathbb{S}_Y$, the

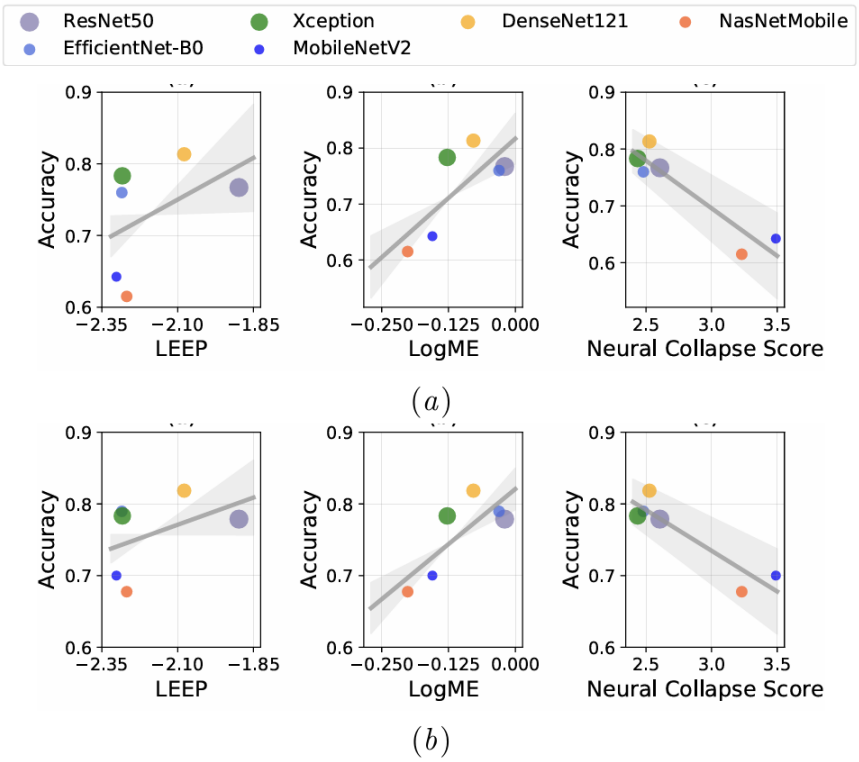


Figure 4.5: **Neural collapse score:** The figures show transferability scores vs. model performance on CIFAR-10. Pearson coefficients between accuracy and LEEP [69], LogME [67], or neural collapse [68] scores during training: (a) only FC; absolute values are 0.44, 0.76, and 0.93, respectively. (b) whole pre-trained model 0.47, 0.79, and 0.91, respectively.

conditional entropy is approximately zero. These assumptions lead to several constraints, such as when the feature spaces of the source and downstream spaces are not necessarily the same.

LEEP : Log Expected Empirical Prediction (LEEP [69]) requires the computation of the "empirical conditional distribution of $\mathbb{DS}_Y$ given $\mathbb{S}_Y$" and is defined as the average log likelihood of the "expected empirical predictor" given $\mathbb{DS}_Y$. This method requires computing the predicted distribution of $\mathbb{DS}_Y$ using the pre-trained model (M) on $\mathbb{S}$. Note that the same forward pass can be used to extract the embedding space of the layers.

LogME : Logarithm of Maximum Evidence (LogME [67]) is based on the principle of maximising the marginalised likelihood or the model evidence given the extracted feature. Similar to the previous method, here also, conditional probability is exploited to measure the TL score, but it is extendable to regression problems and has better generalisation property as compared to NCE. Here, the proposed method is based on the maximum marginalised evidence of the task given the embedding space of the model. However, the computational time complexity of the naive LogME implementation is comparable to that of fine-tuning a pre-trained model.

NCS : Neural Collapse score (NCS [68]) is based on the neural collapse [71] during the terminal stage of supervised training with cross entropy loss. This score measures the discriminative nature of the class embedding space and is far more computationally efficient than methods that require computing conditional probabilities or entropy. For the downstream $\mathcal{DS} = \{(D_i, y_i)\} = (D, Y)$ where for each $i, y_i = c \in C$ and pre-trained model M_j and $M_j^{\ell}(\mathbb{ED}_i)$ is the embedding obtained after the penultimate layer the Neural Collapse score or $TL(\mathbb{DS}, M_i)$ score is given as follows:

$$TL(\mathbb{DS}, M_i) = \frac{\sum_{c \in C}(\frac{1}{|D_c|}\sum_{i=1}^{|D_c|}||M_j^{\ell}(\mathbb{ED}_i) - \mu_c||_2}{\sum_{\forall(c,c')}||\mu_c - \mu_{c'}||_2}$$

The experimental section explores the Transfer Learning Assessment Score (TL) from a topological perspective. It is important to note that while these algebraic topology based scores do not necessarily outperform existing metrics across all settings, they offer distinct advantages: they are defined independently of specific constraints (such as cross-entropy loss [11]), are computationally efficient, and are applicable in both model and data agnostic scenarios. There do exist topological theoretical (TL) scores [72, 29], which are studied and extended in Chapter 6. Further, the methods described in this thesis are new and computationally feasible.

4.3.2 Embedding space

In this thesis, one of the focus points is to understand how the layers of the neural network process the feature space. Towards the same end, the primary focus of the experimental section is to analyse the output space from each layer. As previously mentioned, the output from the $i-$th layer (ℓ_i) of the neural networks f in this thesis will be called the embedding space of the ℓ_i layer and denoted by f^{ℓ_i}. The embedding space represents the features extracted by the neural network to solve an underlying task; this characteristic is precisely why neural networks are categorised as representation learners. Unlike traditional machine learning models that address the dimensionality curse through methods such as principal component extraction or discriminant analysis, neural networks train their layers to extract representations that best solve the underlying task. The width and depth of the layers act like sponges, determining which information to extract and retain.

Nevertheless, without understanding the embedding space, it becomes challenging to exploit neural networks to their full extent due to uncertainty when deployed in the real world and the inability to make consistent changes to the models over time. As observed in previous literature on expressivity [36] and transferability scores [67, 69, 68], utilising the embedding space improves the performance of the model. The paper [73] studies the embedding space by utilising the Riemann geometry and chaotic [36] to define trajectory length that grows with depth to measure the complexity of the function. Here, the work explores how the trajectory length changes with configurations such as weight space or batch normalisation. An important conclusion is that the initial layers matter. Neural networks are more sensitive in the initial layers as compared to the lower layers. This observation is important here, as the results extracted from the cubical complex become less interpretable in deeper layers.

4.4 Experimental Section

The pre-trained models are all trained on the ImageNet dataset [66], while the downstream datasets are different image datasets as follows :

CIFAR10 [74] : The CIFAR10 dataset is a classification problem with 10 different mutually exclusive classes. Each image is 32×32 colour image, with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is balanced.

STL10 [49] : The images are 92×92 pixels and RBG. In addition to the 5000 training images and 8000 test images, there are also 100000 unlabeled images for unsupervised learning. The training and test sets have 10 balanced classes. The classes include images of animals and vehicles acquired from the ImageNet dataset.

AIRCRAFT [75] : The Fine-Grained Visual Classification of Aircraft (AIRCRAFT) dataset with 102 different aircraft model variants. There are 10,200 images in total, with 100 per class. The images have variable resolution; therefore, data augmentation and centre crop are used during training or fine-tuning.

In the preceding experiments, the size of each image is resized or cropped to 224×224 and further normalised using the mean and variance of ImageNet. Pytorch [21] is used to process the dataset and extract the embedding space of the neural network. Building upon the formal definitions of neural network theory provided in Chapter 2, this section details the specific architectural configurations utilised in our empirical analysis. There are a total of 10 different neural networks used in the experiment, which are as follows.

VGG [22] : This very deep Convolutional neural network has 13 convolutional layers and 3 fully connected layers (FC). This model is designed based on the concept of better capacity for deeper models. Thus, the network has a uniform architecture with small 3×3 filters, with the number of kernels doubling after each max pooling layer.

ResNet [23] : There are 4 pre-trained ResNets with different length from $18, 50, 101$ and 152. ResNet18 implies the model has 17 convolutional layers and one fully connected (FC) layer. The PyTorch implementations of the ResNet models of different depths have a varying number of blocks, each designed around a residual connection that allows the input of the block to be directly added to the output of the convolutional layers within that block. Consequently, deeper networks possess a higher density of residual connections; notably, among these variants, ResNet152 achieves the highest accuracy on the ImageNet dataset.

DenseNet [24] : There are 3 pre-trained DenseNet with different depth $121, 169$ and 201. The value 121 in the DenseNet121 implies there are a total of 121 layers with 120 convolutional and 1 FC layer. The layers are distributed between "dense blocks". It can be observed that for the PyTorch implementation used in this chapter, all the models have the same number of dense blocks. This consistency arises because the implementation keeps the block count constant while increasing the number of layers within each block to achieve higher density.

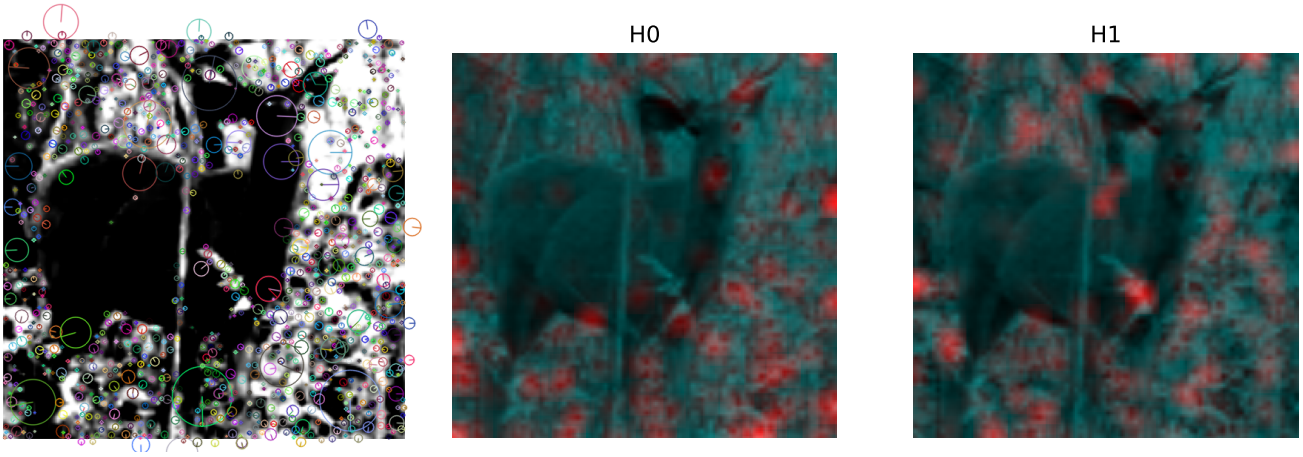
MobileNet [25] : These belong to the class of "mobile models" that are suitable for resource-constrained environments. In this work, the MobileNetV2 implementation in PyTorch is used. The PyTorch modules called "Sequence" are used to define blocks. These blocks consist of one or more inverted residual blocks, which form the main body of the model.

MnasNet [26] : Belonging to the category of mobile models, these models have an automated search of specific configurations that can be optimised for performance and efficiency during training. For the experiments, the Mnasnet1 model in PyTorch is used, where 1 denotes the depth multiplier, a scaling factor that controls the depth of the embedding space at each layer. The scaling factor of 0.5 reports an accuracy of $\approx 67\%$ on ImageNet, while a scaling factor of 1.3 reports 76.506%, which is approximately the reported accuracy for DenseNet201.

In Convolutional Neural Networks, blocks are sequences of layers that often form repeating substructures, which describe the neural network. A block may include one or more convolutional layers followed by additional components such as pooling, batch normalisation, or dropouts. In the preceding sections and chapters, it can be observed that blocks can be used more effectively to approximate the visualization of layer-by-layer embedding transitions. Thus, it also reduces the computational overhead needed. For extracting the homology, the implementation of cubical homology from the paper [64] was used.

4.5 Topology transition of the embedding space

We assume the provided accuracy represents the inherent capacity of the model architecture, rather than a localized state within the weight space. This assumption reflects the expressivity of the model space, facilitated by the flexibility of the topological framework. The scrutiny and the importance of this assumption will be left to the Chapter 9. Further, the "hardness" measure defined in the paper [70] indicates that if the task is "hard", then a more complex model will naturally be required to solve



(a) SIFT (b) Cubical Homology

Figure 4.6: In this work, we use cubical homology on the individual images. The Figures from left to right include (a) the image highlighted with SIFT [76] features, (b) the highlighted cubical homology as extracted from the image. The scale-invariant feature transform (SIFT) is a computer vision algorithm for detecting, describing, and matching local image features. Observe that the local features captured through SIFT are similar to the homological features in Figure (b).

it, regardless of the weight space or loss. Note that the universal approximation theorem only states about the existence of a neural network and not the parameter space, and "more parameters" naturally accommodate more complexity. While neural networks are inherently susceptible to local optima, causing observed accuracy to vary significantly across the weight space, we assume that the provided accuracy represents the inherent capacity of the model architecture itself, rather than a localized parameter state. This assumption allows the performance of the model to be treated as a global property of its architecture. However, this is an inclusive assumption since the acquired test score may not reflect the model capacity; consequently, a critical examination of the implications of this premise is provided in Chapter 9.
In this thesis, we introduce a "hardness" measure based on Betti numbers. The theoretical work states that the neural network collapses the Betti number of the feature space for a classification problem (this is a direct consequence of the classification task) and that Betti numbers expressible for a neural network can act as a measure of its capacity (proved for Pfaffian activation function [3]), the metric specifically leverages the topological dynamics of the classification task. Our study focuses primarily on this theoretical foundation, first elucidating how homological features, extracted via computational topology, evolve as they propagate through the neural network. We further examine the relationship between these features and the Betti numbers of the feature space, ultimately evaluating their utility as a metric for model transferability.
In this chapter, homological complexity based on cubical homology is used to study the embedding space. Homology of embedding space associated with each image is extracted via cubical homology, and thus, unlike in the case of a point cloud with underlying manifold X. See that cubical homology of images is essentially the SIFT features (see the Figure 4.6).

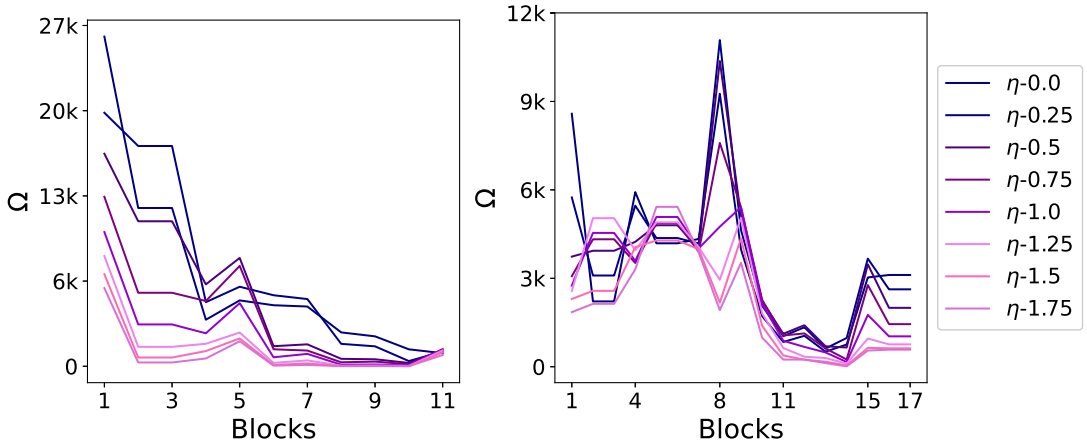
Definition 4.2 (Cubical set)**.** *In this thesis, we adapt the setting for cubical homology from [77]. The maximum number of homologies that can be formed in the dgm is finite and upper bounded by* 2^N*. Let* κ^m *be the set of all* ***elementary cube*** Q*, which is a finite product of the closed interval in* $\mathbb{R}$*. Then, a* ***cubical set*** *is defined as the union of elementary cubes. Let* $\mathbb{A} = \{A \subset \mathbb{R}^m : A = \bigcup_{\in \kappa^m} Q;\ |A| = n\}$ *be a finite collection of cubical set where* Q_i *is a finite elementary cube. For a image dataset* $\mathcal{D} := \{(D_i, y_i)\}_i, n = |D_i|, N = |\mathbb{D}|$*, let us denote the collection of feature space as* $\mathbb{D}$*, then here* $\mathbb{D} = \mathbb{A}$*.*

Notation. *Based on the homological complexity for each* D_i *we have* $\omega_\eta(D_i) := \sum_k \widehat{\beta}_k^{D_i}(\eta)$ *and let*

$\Omega_\eta(\mathbb{D}) := \sum_{i=1}^{N} \omega_\eta(D_i)$. *For simplicity we shall denote* $\widehat{\beta}_k^{D_i}$ *as* $\widehat{\beta}_k^i$, $\omega_\eta(D_i)$ *as* ω_η^i.

In the following section, ImageNet is the only source data ($\mathbb{S}$), and the transferability problem is rephrased as follows.

Notation. *Let* $\mathcal{M} := \{M_i = (f_i, \mathbb{S}) : 1 \leq i \leq N_\mathcal{M}\}$ *denote the finite collection of models, where* $f_i \in \mathcal{F}$ *is the neural network, then* $N_\mathcal{M} = N_\mathcal{F}$. *Moreover, since* M_i *can be characterised by just* f_i *(see Figure 4.4), for simplicity, from here let* f_i *denote* M_i. *There are a total of* 10 *models* $N_\mathcal{M} = N_\mathcal{F} = 10$.



(a) ResNet18 (b) MnasNet

Figure 4.7: **Different threshold** η: It can be observed that for most values of η, a vertical shift is observed for Ω_f thus allowing for a fixed η to compute the Betti curve $\widehat{\beta}(\eta)$.

In this work, we analyse the layer-wise embedding space of the neural network f. So we adopt the following notations when referring to the output of each layer or block:

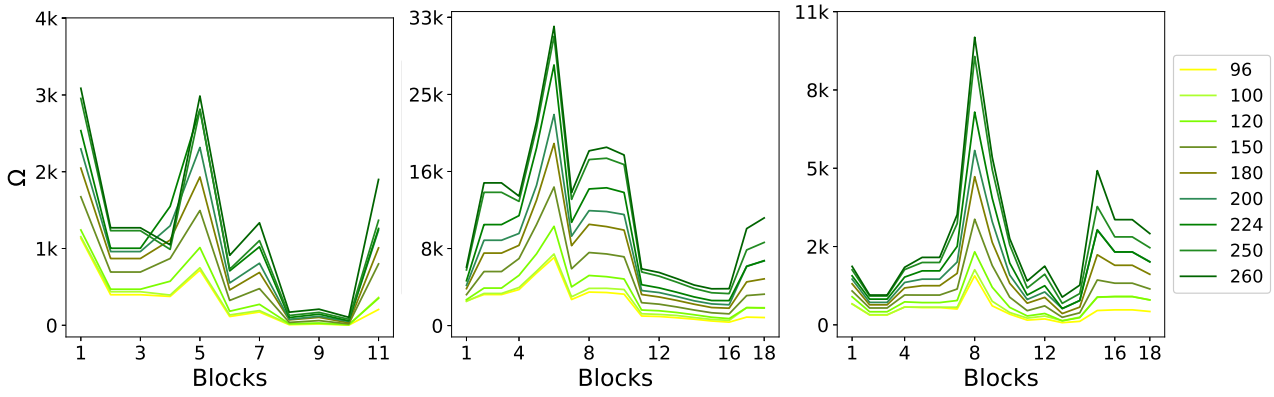
Notation. *For the layer* $\ell-th$ *of* f *operating on the cubical set* D_i, *we denote the output as* $f^\ell(D_i) = D_i^\ell$ *(for simplicity). Furthermore, for each layer* ℓ, *we define the homological complexity at each* $\eta, \Omega_\eta(f^\ell(\mathbb{D}))$. *For the entire network* f, *we denote the resulting sequence of complexity measures across* ℓ *as* $\Omega_\eta(f)$. *In the figures, these are abbreviated as* Ω.

4.5.1 Cubical homology over embedding space

In this section, we empirically demonstrate that the metric based on Ω can be used to characterise the classifier f. Observations indicate that Ω exhibits a vertical translation (y-axis translation) across various parameter regimes. Since vertical shifts do not alter the gradient (slope) [10] of a function, Ω remains a viable tool for neural network characterisation. Specifically, we analyse the following:

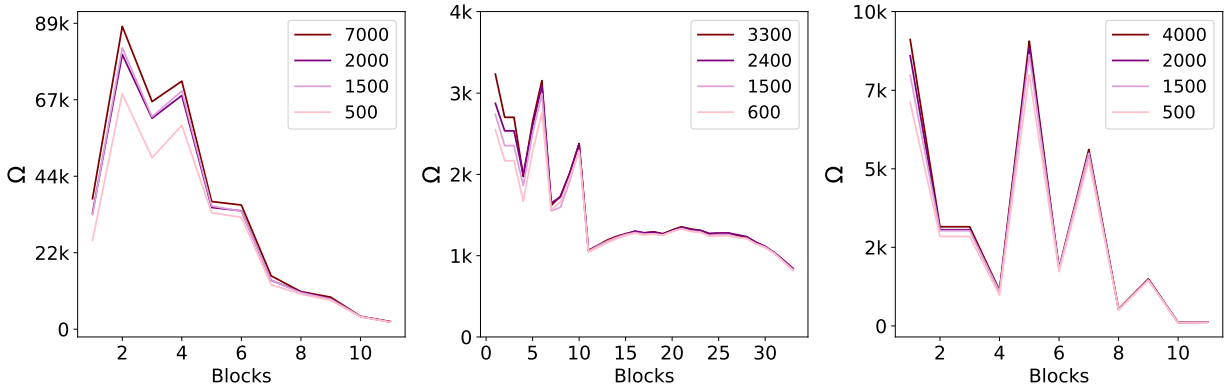
Threshold η : The empirical results across ten models (see Figure 4.7) demonstrate that, for a fixed classifier f, the $\Omega_\eta(f)$ undergoes a vertical translation as η varies. This behaviour suggests that the compositional nature of neural networks is effectively captured by the layer-by-layer transition of the extracted homological features. For clarity, when η is fixed, we denote Ω_η and $\widehat{\beta}(\eta)$ as Ω and $\widehat{\beta}$ respectively. In the subsequent section, a single η is chosen such that $\widehat{\beta}_k \neq 0 \forall k$. While we acknowledge this is not necessarily the optimal selection method, it remains sufficient for the analysis that follows.

Image Resize : As illustrated in Figure 4.8, a similar vertical translation property is observed across different image resizing. This consistency is significant as it allows for a reduction in computational complexity; one can choose a smaller input resolution while still maintaining a reliable characterisation



(a) ResNet18 (b) ResNet50 (c) MnasNet

Figure 4.8: **Different Resize**: It can be observed that for different resize value Ω_f has a vertical shift property. Note that as the Resize value decreases, the Ω curve shifts down. The homological complexity Ω_f depends on the initial homology complexity of the image.



(a) VGG-16 (b) ResNet50 (c) MnasNet

Figure 4.9: **Different sampling complexity** $N \leq |D|$: It can be observed that for different N Ω_f demonstrate a vertical shift thus allowing for a choice of $\tilde{\mathbb{D}} \subseteq \mathbb{D}$. Each image was generated using different datasets: (a) STL10, (b) AIRCRAFT, and (c) CIFAR10.

of the model. Furthermore, these results indicate that the initial value $\Omega(\mathbb{D})$ over the cubical sets directly influences Ω_f, suggesting that Ω_f is not only a property of the classifier but also a reflection of the homological complexity of the instances.

Sample Complexity : For a fixed η, we examine the impact of sampling complexity defined by the number of images, $D_i \in D$. By considering a subset $\tilde{\mathbb{D}} \subseteq \mathbb{D}$, we empirically observe that $\Omega(\mathbb{D}) \approx \Omega(\tilde{\mathbb{D}}) + \rtimes$, where $\mathbf{o} \in \mathbb{R}^+$ denotes a constant offset (see Figure 4.9). This relationship implies that $\Omega(\mathbb{D})$ is a vertical translation of $\Omega(\tilde{\mathbb{D}})$, allowing for a significant reduction in computational overhead by utilising a subset $\tilde{\mathbb{D}}$. For the subsequent empirical study, we assume that the divergence between the class-wise means of $\omega(D_i)$ is sufficiently large such that taking the balanced subsample of the classes ensures that $\mathbf{o}$ remains negligibly small.

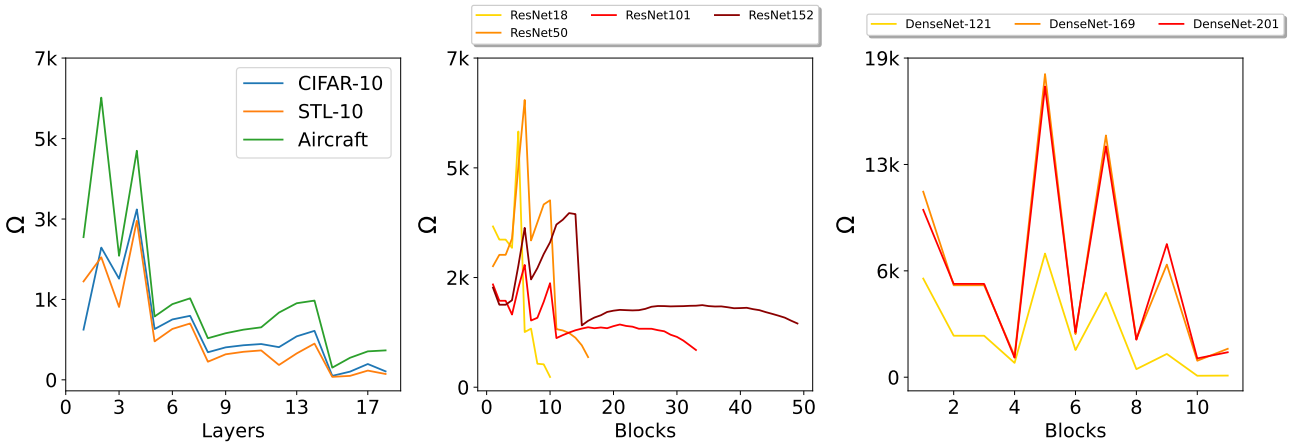
4.5.2 Cubical homology for different models

Building upon the observations in the preceding section, we fix η and evaluate Ω_f across the models. The resulting empirical findings are as follows:

Diverse Datasets : Across different datasets, Ω_f exhibits a consistent vertical translation, indicating that it serves as an intrinsic characteristic of the neural network (See Figure 4.10). Furthermore, the magnitude of Ω_f reflects the underlying data complexity; for instance, the AIRCRAFT dataset yields the highest values, echoing its greater homological complexity and the corresponding difficulty in classification.

Varying Network Depths : When evaluating different depths (see Figure 4.10(b,c)), we observe distinct structural dynamics between architectures. Within ResNets, shallower models appear to be compressed representations of their deeper counterparts, whereas DenseNets exhibit a consistent vertical translation as depth increases. These divergent observations likely reflect the fundamental differences between additive residual connections and feature concatenation, the respective architectural motifs of ResNet and DenseNet, and how they differentially propagate topological information throughout the network.

The decreasing trajectories of Ω_f reflect the non-homeomorphic nature of the classifier layers. These values do not directly equate to the intrinsic homology of the dataset. Within the context of the classification task, the homology of the class-wise feature space is progressively transformed by the classifier toward that of a sphere ($\beta = (1, 0, 0, \dots)$). The observed reduction in Ω_f is likely a direct consequence of this topological convergence. In the subsequent sections, we shall leverage these assumptions to derive a formal metric for model capacity.



(a) MnasNet (b) ResNet (c) DenseNet

Figure 4.10: Ω **of different** f: In the figure(a) we observe that the Ω_f for different dataset and figure(b,c) is the Ω_f for CIFAR10 dataset.

4.5.3 Analysis of the slope of homological complexity

The neural collapse [71] in the classification setting and the Betti number collapse [40] of the class manifold in a classification setting have been extensively studied. In this section we define the framework for homological expressivity required in terms of class manifold.

Notation. *Let us denote the underlying topological space associated with a class $c \in C$ of D (the feature space in the dataset $\mathcal{D}$) as $\mathcal{X}_{D_c} = \mathcal{X}_c$. Then the homological complexity or the sum of Betti numbers is given by $\omega(\mathcal{X}_c)$. Further for each layer $\ell-$ of the neural network f, the homological complexity of the embedding space is denoted as $\omega(f^\ell(\mathcal{X}_{D_c})) = \omega(\mathcal{X}_c^\ell)$ where the initial state at $\ell = 0$ corresponds to the homological complexity of the class, such that $\omega(f^0(\mathcal{X}_{D_c})) = \omega(\mathcal{X}_c)$.*

By the definition of a classification task, the classifier will always attempt to collapse the feature space of the class D_c to a single point (See Figure 1.2 in Chapter 1). Thus for an optimal classifier f the Betti number of the $\mathcal{X}_{D_c}$ will collapse to one with Betti signature $\beta = (1, 0, 0, ..)$ and thus $\omega(\mathcal{X}_{f(D_c)}) = 1$. Here we denote the same in terms of the depth of the neural network $\omega(\mathcal{X}_c^L) = 1$. By the manifold hypothesis and since β_0 is the connected component, one can claim that the feature space of the class D_c has at least the $\beta = (1, 0, 0, 0...)$ and thus $\omega(\mathcal{X}_c) \geq 1$.

Definition 4.3 (Average slope)**.** *The average slope of a function h in the interval $[0, L]$ is given by $avg_s(h) := \frac{h(L)-h(0)}{L}$.*

The instantaneous slope $h'(\hat{\mathbf{o}}) = avg_s(h), \hat{\mathbf{o}} \in [0, L]$ exist by the Mean Value Theorem [10] and $h'(\hat{\mathbf{o}})$ is negative has $h(L) \leq h(0)$. Now for an optimal classifier at L depth, since $h(L) = 1$, then $avg_s(h) = \frac{1-h(0)}{L} \leq 0$. Also note that for classifier $L \to \infty$ then $avg_s(h) \to 0$ and $f'(\hat{\mathbf{o}}) = 0$, a saddle or critical point exists where the slope is at an angle $90°$. Note that for a single-layered neural network (∞ width), an optimal classifier would also have an instantaneous slope at $1 - h(0)$. When $h(L) = h(0)$, the layers are homologically equivalent functions, Betti numbers of $\mathcal{X}_c$ do not change at all, the classifier will learn if and only if the feature space $\beta(\mathcal{X}_c) = (1, 0, 0, \dots) \forall c \in C$. When the slope at all points in $[0, L]$ is positive under this definition, such a classifier is not possible. The above analysis can be put together as the following theorem:

Theorem 4.1 (Approximating capacity)**.** *Consider a neural network classifier f with L layers on a dataset $\mathcal{D}, c \in C$, the index set of classes and $\mathcal{X}_c$ the topological space associated with the underlying manifold of the class c. Let $h : [0, L] \to \mathbb{R}$ be a continuous function differentiable on $[0, L]$ interpolating the homological complexity $h(\ell) = \omega(\mathcal{X}_c^\ell)$ at discrete values $\ell \in \{0, 1, 2..., L\}$ for the class c. Then:*

1. *Existence : $\exists \hat{\mathbf{o}} \in (0, L)$ such that $h'(\hat{\mathbf{o}}) = \frac{\omega(\mathcal{X}_c^L)-\omega(\mathcal{X}_c)}{L}$.*
2. *Asymptotic Decay: For each class c where $h(L) \leq h(0) < \mathbf{r}$ for some finite constant $\mathbf{r} \in \mathbb{R}$, as $L \to \infty$ then the instantaneous slope at $\hat{\mathbf{o}}$ converges to 0.*
3. *Optimal Classifier: For each class c where $h(L) \leq h(0)$, finite L the steepest instantaneous negative slope at $\hat{\mathbf{o}}$ (approaching the slope of y-axis) is attained by the optimal classifier that collapses the $\beta(\mathcal{X}_c^L) = 1 \forall c \in C$.*

Proof. 1. Since h is a function that is continuous on $[0, L]$ and differentiable on $(0, L)$, by the mean value theorem there exist at least one $\hat{\mathbf{o}} \in (0, L)$ such that

$$h'(\hat{\mathbf{o}}) = \frac{h(L) - h(0)}{L} = \frac{\omega(\mathcal{X}_c^L) - \omega(\mathcal{X}_c)}{L}$$

2. The instantaneous slope at $\hat{\mathbf{o}}$ is given by $h'(\hat{\mathbf{o}}) = \frac{h(L)-h(0)}{L}$. By definition $h(\ell) = \omega(\mathcal{X}_c^\ell) \geq 1, \ell \in \{0, 1, 2..., L\}$ as a non-empty manifold must have at least one connected component ($\beta_0 \geq 1$). Given the assumption $h(L) \leq h(0) < \mathbf{r}$ for some finite constant $\mathbf{r}$ then $h(L) - \mathbf{r} < h(L) - h(0) \leq 1 - h(L)$. Now $0 < h(L) \implies -\mathbf{r} < h(L) - \mathbf{r}$ and $1 - h(L) \leq 0$. Thus

$$-\mathbf{r} \leq h(L) - h(0) \leq 0 \implies \lim_{L\to\infty} \frac{-\mathbf{r}}{L} \leq \lim_{L\to\infty} \frac{h(L) - h(0)}{L} \leq \lim_{L\to\infty} 0$$

Applying the Squeeze Theorem for Limits we have $\lim_{L\to\infty} h'(\hat{\mathbf{o}}) = 0$.

3. For finite L , let $\hat{h}$ be the differentiable function on $[0, L]$ interpolating the homological complexity for the optimal classifier. Given for the optimal classifier $\forall c \in C, \beta(\mathcal{X}_c^L) = 1$. Therefore $\omega(\mathcal{X}_c^L) = 1 \implies \hat{h}(L) = 1$. Now for any other neural network classifier with L layers given $h(L) \leq h(0)$ and

$1 \leq h(L)$ from definition, we have $1 - h(0) \leq h(L) - h(0) \leq 0$. Now for each class $c \in C, \hat{h}(0) = h(0) = \omega(\mathcal{X}_c)$, thus

$$1 - \omega(\mathcal{X}_c) \leq h(L) - \omega(\mathcal{X}_c) \leq 0 \implies \frac{1 - \hat{h}(0)}{L} \leq \frac{h(L) - h(0)}{L} \leq 0$$

Thus the instantaneous negative slope at $\hat{\mathbf{o}}$ for functions $\hat{h}$ and h satisfy $\hat{h}'(\hat{\mathbf{o}}) \leq h'(\hat{\mathbf{o}}) \leq 0$.

□

The proposed theorem provides a formal framework for understanding the topological dynamics of manifold collapse through the lens of functional interpolation. The first part of the theorem establishes the existence of a critical point $\hat{\mathbf{o}}$ via the Mean Value Theorem so that extrapolation over the interval $[0, L]$ (in particular $h(0)$ and $h(L)$) can be used to approximate the average slope, the measure of "topological work" required by a neural network to classify $\omega(\mathcal{X}_c)$. Such extrapolation is fundamental to this study, as the direct measurement of $\omega(\mathcal{X}_c)$ (or $h(0)$) is often not possible for real-world complex data. Further in this work, we want to explain the potential existence of a correlation of TL score based on homological complexity to the accuracy after fine-tuning ($h(L)$). Building on this, the second part provides the asymptotic limit, that for infinitely deep networks, the "topological work" is distributed so broadly across the network that the topological simplification per layer vanishes. Finally, the third part demonstrates that for a fixed depth L the optimal classifier is the one that maximises the magnitude of the instantaneous negative slope (see Figure 4.11). Because the optimal classifier achieves the theoretical minimum homological complexity at the final layer ($h(L) = 1$), it maximises the numerator of the derivative, thereby attaining a steeper instantaneous slope than any non-optimal counterpart. Consequently, the training process can be viewed as a pursuit of an optimal slope value for a given $\omega(\mathcal{X}_c)$.

Definition 4.4 (Maximal Homological Complexity)**.** *Let $\mathcal{D}$ be a dataset with class index set C. Let $\mathcal{X}_c$ denote the underlying topological space for each class $c \in C$. We define the* ***maximal homological complexity*** *of the dataset, denoted by $\phi := \max_{c \in C} \omega(\mathcal{X}_c)$. The index of the class exhibiting this maximal complexity is given by $\hat{c} = \arg\max_{c \in C} \omega(\mathcal{X}_c)$.*

The maximal homological complexity is defined for any dataset and is just the homological complexity of the class with the "maximum" homology. It is possible that there are multiple classes with the maximal homological complexity, and certain homologies influence the decision boundary more than others. However, in this thesis, we focus on the homological satisfaction necessary for classifiers to achieve perfect classification (see Chapter 6).

Corollary 4.1. *Consider an optimal neural network classifier f with L layers. For each $c \in C$ let the continuous function $h_c : [0, L] \to \mathbb{R}$ be differentiable on $[0, L]$ such that $h_c(\ell) = \omega(\mathcal{X}_c^\ell)$, for $\ell \in \{0, 1, 2..., L\}$, then there exist $\hat{\mathbf{o}}_{\hat{c}} \in (0, L)$ such that for the class $\hat{c} \in C$ possessing the maximal homological complexity $\omega(\mathcal{X}_{\hat{c}})$ exhibits an instantaneous slope $h'_{\hat{c}}(\hat{\mathbf{o}}_{\hat{c}})$ whose inclination angle is closest to $90°$.*

Proof. Given an optimal classifier implies $h_c(L) = 1 \forall c \in C$. Let $\hat{c} \in C$ be the class with maximal homological complexity, that is $\omega(\mathcal{X}_{\hat{c}}) \geq \omega(\mathcal{X}_c) \forall c \neq \hat{c}, c \in C$. Then $\omega(\mathcal{X}_{\hat{c}}) \geq \omega(\mathcal{X}_c) > 0 \implies 1 - \omega(\mathcal{X}_{\hat{c}}) \leq 1 - \omega(\mathcal{X}_c) < 1 \forall c \in C, c \neq \hat{c} \in C$. By the Mean Value Theorem, $\exists \hat{\mathbf{o}}_c \in (0, L)$ such that $h'_c(\hat{\mathbf{o}}_c) = \frac{h_c(L) - h_c(0)}{L} = \frac{1 - \omega(\mathcal{X}_c)}{L}$. Substituting the inequality above we obtain $h'_{\hat{c}}(\hat{\mathbf{o}}_{\hat{c}}) \leq h'_c(\hat{\mathbf{o}}_c)$. To evaluate the geometric orientation, consider $\tan^{-1}(h'_{\hat{c}}(\hat{\mathbf{o}}_{\hat{c}}))$ and since $\tan^{-1}$ is a monotonically increasing function within the range $[-90°, 90°]$(or $[0°, 180°]$)

$$-90° \leq \tan^{-1}(h'_{\hat{c}}(\hat{\mathbf{o}}_{\hat{c}})) \leq \tan^{-1}(h'_c(\hat{\mathbf{o}}_c)) \leq 0 \text{ or } 90° \leq \tan^{-1}(h'_{\hat{c}}(\hat{\mathbf{o}}_{\hat{c}})) \leq \tan^{-1}(h'_c(\hat{\mathbf{o}}_c)) \leq 180°$$

Thus, the class $\hat{c}$ with the maximal homological complexity attains an instantaneous slope whose orientation is closest to the vertical $y-$axis. □

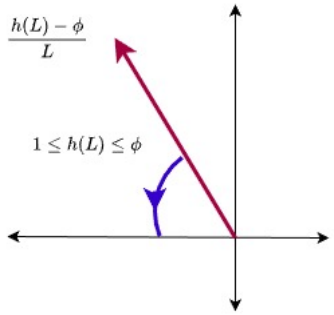


Figure 4.11: **Negative slope**: Homological capacity Φ_f is evaluated in the negative range (assuming $h(L) \leq \phi$). Thus the optimal/ideal classifier f, has homological capacity $\Phi_f = \frac{1-\phi}{L}$, the largest negative value.

Having observed above that the results in Theorem 4.1(a) establishes a framework for approximating the homological transition between the homological complexity of the class feature space $\omega(\mathcal{X}_c^0)$ and its representation at the terminal $L-$th layers of classifier f, $\omega(f(\mathcal{X}_c))$ (or $\omega(\mathcal{X}_c^L)$), by leveraging the slope of the differentiable mapping h provided it exist in the interval $[0, L]$. Furthermore, the results in Theorem 4.1(b,c) and Corollary 4.1, demonstrates that the instantaneous slope $h'_{\hat{c}}(\hat{\mathbf{o}}_{\hat{c}})$ provides a rigorous quantification of the "topological work" across L-the transformational burden imposed on the classifier f in terms of the homological complexity of $\mathcal{X}_c$. To formalise this relationship, we define a measure of "hardness" for model capacity called Homological capacity Φ_f. This approach mirrors the "hardness" analysis in the paper [70].

Definition 4.5 (Homological capacity)**.** *Let the homological capacity of the neural network classifier f with depth L on the dataset $\mathcal{D}$ with class index set C be denoted by Φ_f and defined as*

$$\Phi_f := \frac{\omega(\mathcal{X}_{\hat{c}}^L) - \phi}{L}$$

where ϕ represents the maximal homological complexity, $\hat{c}$ is the corresponding index of the class with maximal homological complexity and $\omega(\mathcal{X}_{\hat{c}}^L)$ is its homological complexity at $L-$th layer, or more clearly the response space.

Note that $\omega(\mathcal{X}) \geq 1$ for any non empty topological space.

Remark 4.2. *Further, for the aforementioned function h in Theorem 4.1 to satisfy the theorem, it must be continuous on the closed interval $[0, L]$ and differentiable on the open interval $(0, 1)$. While the interpolation $h_c(\ell) = \omega(\mathcal{X}_c^\ell)$, for $\ell \in \{1, 2..., L-1\}$ is not a strict requirement for the existence of $\hat{o}$, it serves as a functional basis for the extrapolation of $h(0)$ and $h(L)$ in practical applications where direct manifold measurement is restricted.*

The above Theorem 4.1 establishes that the homological capacity Φ_f can be effectively approximated through such a function h, while its corollary 4.1 demonstrates that the optimal slope is steepest for the class possessing the maximal initial homological complexity $(\omega(\mathcal{X}_{\hat{c}}^L))$. Whereas the work by Bianchini and Scarselli [3] on the deep and shallow network, homological complexity was used to study the homological expressivity of deep versus shallow networks, the current framework utilises these topological properties to define a homological capacity (Φ) specifically grounded in the homological complexity of the underlying dataset $\mathcal{D}$.

Theorem 4.3 (Architectural Efficiency)**.** *Let $\phi := \max_{c \in C} \omega(\mathcal{X}_c)$ be the maximal initial homological complexity of a dataset $\mathcal{D}$ with class index C. Consider a finite collection of neural network classifiers $\mathcal{F} = \{f_i : i \in \{1, 2, ..., N_{\mathcal{F}}\} \subset \mathbb{N}\}$ such that each achieves optimal class manifold collapse at depth L_i that is $\omega(\mathcal{X}_c^L) = 1 \forall c \in C$. If the depths are ordered such that $L_i \leq L_{i+1}$ for all i, then the homological capacity Φ_{f_i} is greatest in magnitude for the shallowest classifier f_1, with arctangent value closest to $90°$.*

Proof. Given $\mathcal{F} = \{f_i : i \in \{1, 2, ..., N_{\mathcal{F}}\} \subset \mathbb{N}\}$ with $\omega(\mathcal{X}_c^{L_i}) = 1 \forall c \in C$ at depth L_i and $L_i \leq L_{i+1} \forall i$ on the dataset $\mathcal{D}$. By definition $\Phi_{f_i} = \frac{\omega(\mathcal{X}_{\hat{c}}^{L_i}) - \phi}{L_i}$ then by substitution $\Phi_{f_i} = \frac{1-\phi}{L_i}$. By definition, $\phi \geq 1$ as a non-empty manifold must have at least one connected component ($\beta_0 \geq 1$) and given $L_i \leq L_{i+1} \forall i$:

$$\frac{1-\phi}{L_i} \leq \frac{1-\phi}{L_{i+1}} \leq 0, \implies \Phi_{f_i} \leq \Phi_{f_{i+1}} \leq 0 \implies |\Phi_{f_i}| \geq |\Phi_{f_{i+1}}|$$

This inequality demonstrates that $|\Phi_{f_i}| \geq |\Phi_{f_{i+1}}|$, indicating that the absolute magnitude of capacity decreases as depth increases for a fixed ϕ. Thus, the classifier f_1 with the minimum depth required for collapse exhibits the maximum absolute homological capacity $|\Phi_{f_1}|$. To evaluate the geometric orientation, we apply the $\tan^{-1}$ function. Within the interval $[90°, 180°]$ (corresponding to negative slopes), the arctangent function is monotonically increasing:

$$90° \leq \tan^{-1}(\Phi_{f_i}) \leq \tan^{-1}(\Phi_{f_{i+1}}) \leq 180° \forall i \in \{1, 2, ..., N_{\mathcal{F}}\}$$

For the shallowest classifier f_1, the value $\tan^{-1}(\Phi_{f_1})$ as inclination angle closest to the vertical 90°. □

Thus, the arctangent of Φ_f provides a robust geometric metric for identifying the most efficient architecture for a given dataset $\mathcal{D}$. Specifically, the maximum magnitude of Φ_f identify the shallowest classifier f_1 capable of satisfying the initial homological complexity ϕ. This framework facilitates the selection of the most compact architecture necessary to achieve total manifold collapse of the class with maximal homological complexity, effectively solving the classification problem with minimal depth.

Remark 4.4. *This leads to a natural question: how does Homological Capacity Φ_f relate to expressivity? We define Φ_f as a metric measured across the depth of a neural network. It quantifies whether a model possesses the homological expressivity required such that each class is collapsed to a single point. Rather than establishing that a deeper model is inherently superior, this score adapts to the data. Consequently, while deeper models offer higher topological expressivity, Φ_f identifies the minimum depth L necessary to resolve the homological complexity of the given dataset*

Furthermore, it can be demonstrated that for $\omega(\mathcal{X}_{\hat{c}}^L) = 1, \hat{c} = \arg\max_{c \in C} \omega(\mathcal{X}_c)$, the homological capacity Φ_f serves as a universal lower bound for the topological transition of $\mathcal{X}_c \forall c \in C$ within the classifier f regardless of the value $\omega(\mathcal{X}_c^L)$. This is proved in the corollary below.

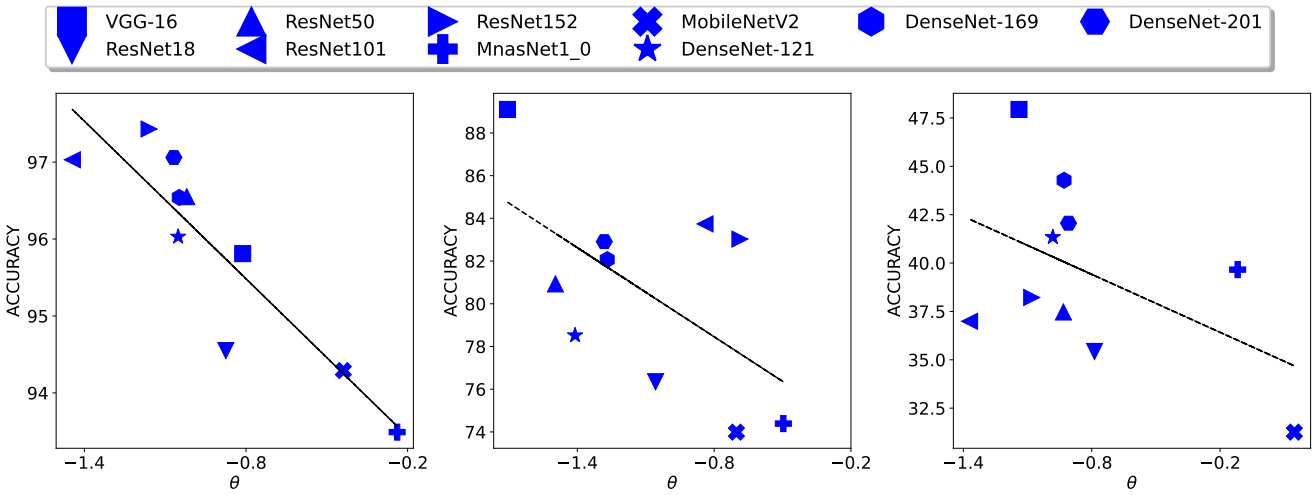
Corollary 4.2. *Consider a neural network classifier f with depth L acting on a dataset $\mathcal{D}$ with class index set C. If $\omega(\mathcal{X}_{\hat{c}}^L) = 1, \hat{c} = \arg\max_{c \in C} \omega(\mathcal{X}_c)$, then the homological capacity Φ_f serves as a universal lower bound for the topological change across all classes:*

$$\forall c \in C, \quad \Phi_f \leq \frac{\omega(\mathcal{X}_c^L) - \omega(\mathcal{X}_c)}{L}$$

Proof. The proof follows from the definition of Φ_f. Given $\omega(\mathcal{X}_{\hat{c}}^L) = 1, \hat{c} = \arg\max_{c \in C} \omega(\mathcal{X}_c)$ and by definition we have $\Phi_f = \frac{\omega(\mathcal{X}_{\hat{c}}^L) - \phi}{L} = \frac{1 - \max_{c \in C} \omega(\mathcal{X}_c)}{L}$. We consider two cases for any class $c \in C$:
Case 1: Suppose $\omega(\mathcal{X}_c^L) \leq \omega(\mathcal{X}_c)$. Since the minimum possible homological complexity (ω) is 1, we have $1 - \omega(\mathcal{X}_c) \leq \omega(\mathcal{X}_c^L) - \omega(\mathcal{X}_c) \leq 0$. Dividing by L:

$$\frac{1 - \max_{c \in C} \omega(\mathcal{X}_c)}{L} \leq \frac{1 - \omega(\mathcal{X}_c)}{L} \leq \frac{\omega(\mathcal{X}_c^L) - \omega(\mathcal{X}_c)}{L} \leq 0 \implies \Phi_f \leq \frac{\omega(\mathcal{X}_c^L) - \omega(\mathcal{X}_c)}{L} \leq 0$$

Case 2: Suppose $\omega(\mathcal{X}_c^L) > \omega(\mathcal{X}_c)$. In this instance, the topological transition of $\mathcal{X}_c$ within the classifier f is positive or $\omega(\mathcal{X}_c^L) - \omega(\mathcal{X}_c) > 0$. Since $\Phi_f = \frac{1-\phi}{L}$ and $\phi \geq 1$, it is clear that $\Phi_f \leq 0$. $\omega(\mathcal{X}_c^L) - \omega(\mathcal{X}_c) > 0 \implies \Phi_f \leq 0 < \frac{\omega(\mathcal{X}_c^L) - \omega(\mathcal{X}_c)}{L}$. Therefore, the inequality $\Phi_f \leq 0 < \frac{\omega(\mathcal{X}_c^L) - \omega(\mathcal{X}_c)}{L}$ holds trivially. □



(a) STL10 (b) CIFAR10 (c) AIRCRAFT

Figure 4.12: Given are the Accuracy vs θ_Ω score scores for different datasets.

Thus, in all cases, the established homological capacity Φ_f provides a universal lower bound for the topological transition of the classifier f. This suggests that if a Network possesses sufficient homological capacity (Φ_f) to collapse the most complex manifold $\mathcal{X}_{\hat{c}}$, it inherently possesses the homological capacity (Φ_f) to resolve all other manifolds within the dataset $\mathcal{D}$. Within this framework, the condition for $\omega(\mathcal{X}_c^L)$ remains mathematically flexible, even allowing for instances where $\omega(\mathcal{X}_c^L) > \omega(\mathcal{X}_c)$. The corollary 4.2 is more of a generalisation of the corollary 4.1 above, where $\omega(\mathcal{X}_c) = 1$ and the homological transition is approximated for each class via the slope of differentiable functions (h_c).

However, from a functional perspective, if a classifier results in a terminal state where the homological complexity of a class increases $(\omega(\mathcal{X}_{f(D_c)}) > \omega(\mathcal{X}_c))$, the model is failing to fulfil the fundamental objective of the classification task. As the standard objective in neural network training is to drive each class manifold toward a singular point $(\beta = (1, 0, 0, \dots))$, any increase in complexity signifies a failure in manifold collapse and, consequently, a misclassification. **Therefore, while Φ_f serves as a mathematical lower bound for all transitions, the realisation of an optimal classifier is contingent upon achieving a negative change across all class manifolds.** Nevertheless, Φ_f defines the minimum homological capacity (see Figure 4.11) required for a classifier to successfully resolve the topological complexity of the dataset.

It must be noted that the current analysis is not yet exhaustive, as it does not explicitly account for the interplay between network width and depth. Our framework operates under a "brute force" assumption, presupposing that a specific depth L is necessitated to resolve a given homological complexity $(\omega(\mathcal{X}_c))$. While the trade-off between layer width and compositional depth requires further rigorous scrutiny, this theoretical foundation nonetheless provides a robust justification for our choice of the TL score (Transfer Learning score). By isolating the depth-wise rate of topological change, we have established a measurable proxy for the "topological work" performed by a pre-trained architecture.

4.5.4 Empirical Approximation of Homological capacity

In the preceding section, by isolating the depth-wise rate of topological change, we have established a measurable proxy for the "topological work" performed by an architecture. Further, Theorem 4.1 demonstrates that the function h can be used to approximate the Φ_f as the average slope. In this section, we describe an empirical methodology to approximate h, based on cubical homology. This choice is motivated by the fact that cubical settings are computationally more efficient and highly feasible for the structured grid data inherent in image analysis.

In an image classification task (where $f(D_i) = y_i$) in the cubical setting, each image is reduced to a single point, and the homology of $f(D_i)$ is isomorphic to the Betti signature $(1, 0, 0, \dots)$. The classifier in the cubical setting has also been empirically observed in the previous sections as reducing the $\widehat{\beta}(\eta)(D_i)$ and Ω_η (the sum over the homological complexity ω across the images, for a single image take $N = 1$). However, this assumption is not guaranteed, as it often reduces to a vector $R^{|C|}$ of probabilistic values used to make a decision. On the other hand, by letting $\widehat{\beta}^{D_i}(\eta) >> (1, 0, 0 \dots)$ for some η, although not strictly monotonic, one can still derive a negative slope and assume that the data complexity is defined by the higher value of Ω_η . However, this assumption is not guaranteed, as it often reduces to a vector $R^{|C|}$ of probabilistic values used to make a decision. One can define the maximal homological complexity $(\phi(\mathbb{D}) = \Omega_\eta)$ over the individual image in $\mathbb{D}$ as follows :

$$\Omega_\eta := (1/|\mathbb{D}|) \sum_{D_i \in \mathbb{D}} \widehat{\omega}_\eta(D_i)$$

Using average values instead of maximum values stabilises metrics across different subsampling. Take $\widehat{\beta}^{(D_i^L)} \approx (1, 0, 0 \dots)$ for all D_i.

Theorem 4.5. *Consider the sequence $\{D_i^\ell\}_{\ell=1}^L$ generated by classifier f on a dataset $\mathcal{D}$ for each cubical set $D_i \in \mathbb{D}$. Let Ω^ℓ denote the average homological complexity across the samples of D_i at layer $\ell \in \{0, 1, 2, \dots L-1, L\}$ over the cubical set D_i. If there exists a filtration threshold η such that it satisfy $\widehat{\beta}_k^{(D_i^0)} \geq \widehat{\beta}_k^{(D_i^L)}$ for all $D_i \in \mathbb{D}$ and the continuous polynomial $p : [0, L] \to \mathbb{R}$ interpolates Ω^ℓ, then there exists a point $\hat{\ell} \in (0, L)$ such that the rate of change is negative:*

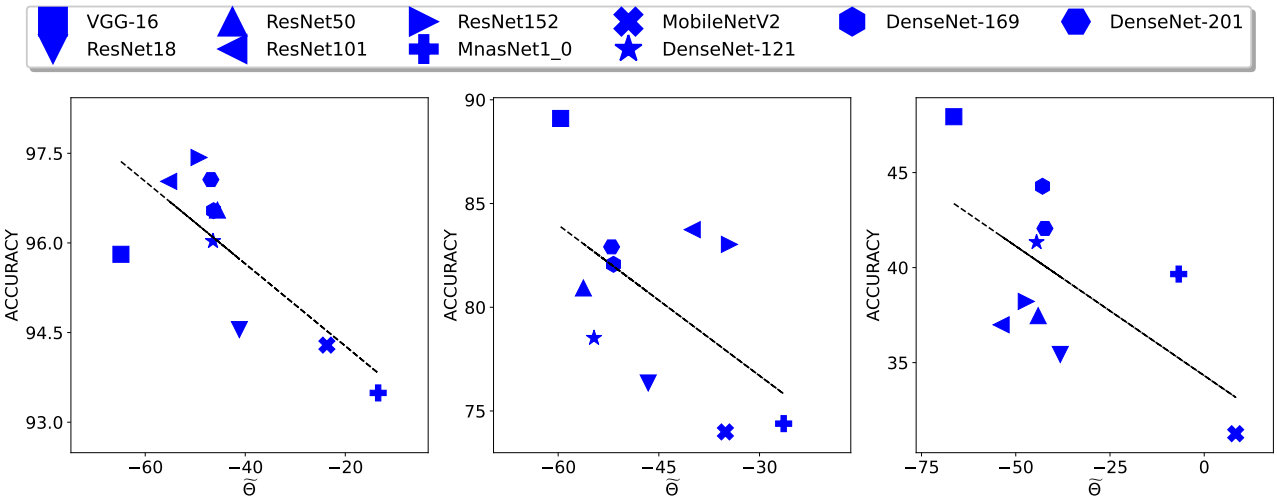
$$p'(\hat{\ell}) = \frac{\Omega^L - \Omega^0}{L} \leq 0$$

Proof. By definition $\Omega^\ell = \frac{1}{|D^\ell|} \sum_{i=1}^{|D^\ell|} \widehat{\omega}(D_i^\ell)$ with $\ell = 0, D_i^0 = D_i$. Since $\widehat{\beta}_k^{(D_i^\ell)} \geq 0$, we have $\widehat{\omega}(D_i^0) \geq 1 \implies \sum_{i=1}^{|D|} \widehat{\omega}(D_i^0) \geq |D| \implies \Omega^0 \geq 1$ and $\Omega^L \geq 1$. Since p is a polynomial, it is differentiable everywhere and then by the Mean Value Theorem, $\exists \hat{\ell} \in (0, L)$ we have $p'(\hat{\ell}) = \frac{\Omega^L - \Omega^0}{L}$. Given that $\exists \eta$ such that $\widehat{\beta}_k^{(D_i^0)} \geq \widehat{\beta}_k^{(D_i^L)}$, we obtain $\Omega^0 \geq \Omega^L \implies 0 \geq \Omega^L - \Omega^0$. Therefore, $p'(\hat{\ell}) \leq 0$. □

Here, the cardinality $|D^\ell|$ represents the number of feature representations at layer ℓ and here $|D^\ell| = |D^0| \forall \ell \in \{0, ...L\}$. The theorem above demonstrates that it is possible to define the homological capacity over the individual images. It is important to note that in the theorem above, it is not the class manifold but the homological complexity over the cubical set (D_i, representing the input images). While the theoretical definition of Φ_f requires the manifold complexities $\omega(\mathcal{X}_c^L)$ and $\omega(\mathcal{X}_c)$ in this Chapter and Chapter 6, these values are not directly computed. Instead, we approximate the rate of change $p'(\hat{\ell})$ from the polynomial p interpolated over the internal layers $\ell \in \{1, ...L-1\}$. Consequently, here the values for Ω^0 and Ω^L are defined as the extrapolations to the endpoints of the interval $[0, L]$, to ensure that the mean rate of change remains consistent with the observed internal transformation of the feature spaces.

Notation. *Let us denoted the extrapolated values of $\omega(\mathcal{X}_c^L) \& \omega(\mathcal{X}_c)$ as $\widetilde{\Omega}(\mathcal{X}_c^L)$ & $\widetilde{\Omega}(\mathcal{X}_c)$, respectively.*

While $\omega(\mathcal{X}_c)$ in Theorem 4.5 is assumed to be extrapolated as average homological complexity across the samples of D_i, the empirical framework goes one step further. We assume that $\omega(\mathcal{X}_c)$ and $\omega(\mathcal{X}_c^L)$ can be extrapolated by the polynomial p that interpolates the observed complexities Ω^ℓ at layers $\ell \in \{1, 2, \dots L-1\}$. Specifically, we assume $p(0) = \widetilde{\Omega}(\mathcal{X}_c)$ and $p(L) = \widetilde{\Omega}(\mathcal{X}_c^L)$ and the instantaneous slope $p'(\hat{\mathbf{o}}) \approx \Phi_f, \hat{\mathbf{o}} \in (0, L)$. In the Chapter 6, the behaviour of this polynomial p is explored in terms of the layer-wise $\omega(\mathcal{X}_c^\ell)$ through the metric based on topological average life. While Chapter 7, provides an analysis of how $\omega(\mathcal{X}_c^\ell)$ evolve during the training process.



(a) STL10 (b) CIFAR10 (c) AIRCRAFT

Figure 4.13: Given are the accuracy vs.$\widetilde{\Theta}_\Omega$ scores for different datasets.

Table 4.1: $corr_r$ is the Pearson correlation for the TL scores θ and $\widetilde{\Theta}$ with their p-values. Due to the high p-values in the Spearman [78] and Kendall [79] correlation tests, we do not use the same here for the correlation test (See Chapter 6 for the same).

Dataset	$corr_r(\theta)$	p-value $corr_r(\theta)$	$corr_r\widetilde{\Theta}$	p-value $corr_r\widetilde{\Theta}$
CIFAR10	-0.590	0.07	-0.569	0.085
STL10	-0.595	0.06	-0.761	0.01
AIRCRAFT	-0.695	0.02	-0.637	0.047

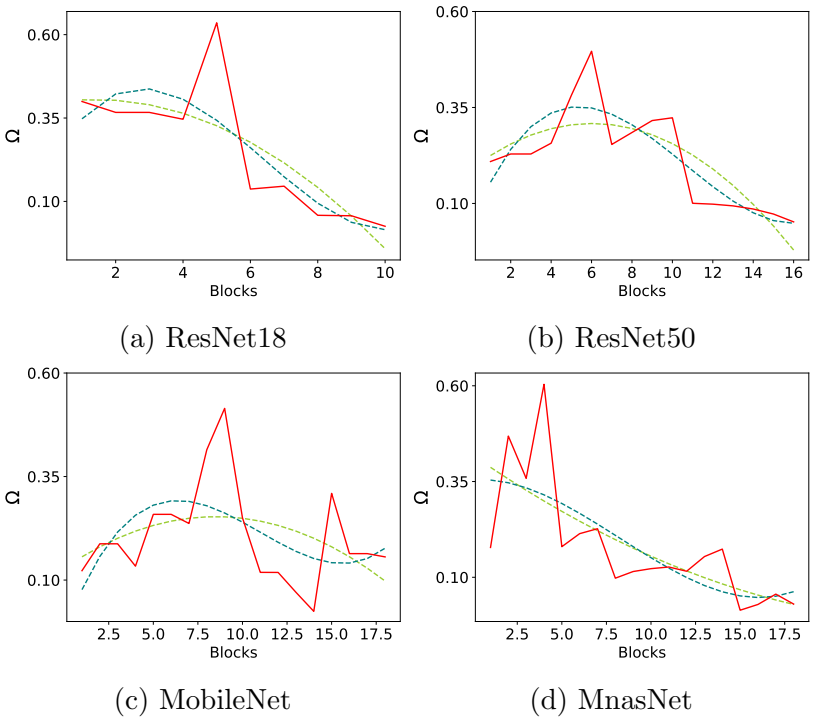
Notation. *Throughout this work, the approximated homological capacity Φ_f is denoted as θ , while its geometric orientation is defined as $\widetilde{\Theta} = \tan^{-1}(\theta)$. To distinguish these from specific variants, we adopt the notation $(\theta_\Omega, \widetilde{\Theta}_\Omega)$ and $(\theta_\Lambda, \widetilde{\Theta}_\Lambda)$, with the latter appearing in Chapter 6.*

In the Figure 4.12 θ is computed using Ω curve across the blocks/layers of the network as an empirical measure of Φ_f. Correspondingly, Figure 4.13 illustrates the values of $\widetilde{\Theta}$. Although the derived TL scores (θ_Ω and $\widetilde{\Theta}_\Omega$) are based on the individual images, they do demonstrate a moderate correlation between the accuracy score and the p-value that suggests a statistically significant relationship (see Table 4.1) supporting the existence of an underlying link between homological capacity and classification performance. Further, it empirically demonstrates that metrics derived from the homological transitions of embedding space can be used to measure the generalisation to a similar downstream task. These results further empirically validate Theorem 4.3.

4.6 Discussion and Scrutiny

In this section, we critically examine the theoretical and empirical underpinnings of our framework:

Existence vs Estimation : While Theorem 4.1 guarantees the existence of a critical point $\hat{\mathbf{o}}$, it does not provide an explicit method for its location. In our experiments, we approximate the differentiable function h using polynomials of degree 1, 2, or 3 (see Figure 4.14). Although the theorem considers a



(a) ResNet18 (b) ResNet50

(c) MobileNet (d) MnasNet

Figure 4.14: Figures show the (green) polynomial approximations of the ω curves across the Embedding space for different models.

fit across $L+1$ points, the primary empirical objective is the accurate extrapolation of $h_{\hat{c}}(0)$ and $h_{\hat{c}}(L)$. Furthermore, while $h_{\hat{c}}(0)$ is theoretically defined by the class manifold, our experimental approach utilises implicit polynomial extrapolation. for which the error is very high.

Capacity vs Transferability : The derived score leans towards a model capacity analysis rather than a transferability score, which requires downstream information, $\phi_{\mathbb{DS}}$. Note that in the case of an RGB image, when we consider the underlying "manifold," it is far more complex, and thus polynomial extrapolation provides a necessary and tractable approximation of the "topological work" required for manifold resolution.

Embedding Space as a Proxy : The embedding space is posited to contain latent information regarding $\omega(\mathcal{X}_c)$ and thus is exploited as a functional alternative to direct manifold measurement $\omega(\mathcal{X}_c)$.

Class Specific vs Data Wide Quantification : Our current metric is restricted to individual class dynamics. To quantify the "hardness" of an entire dataset, we currently rely on the class possessing the maximal homological complexity (ϕ).

Model vs Data complexity : Interpreting the slope as a measure of complexity requires a simultaneous understanding of the numerator (homological complexity) and the denominator (model depth). It is important to note that the current framework focuses primarily on depth-wise topological work; the intricate interplay between network width and depth, and how their combination modulates manifold collapse, remains outside the scope of this study and requires further investigation.

Biasness to Depth : The proposed Φ_f has a bias towards models of lower depth, thus when comparing between Architectures, keeping the L constant would be an ideal scenario. Thus, choosing to compute Φ_f over a block is also preferable here.

Consequently, determining how layer-by-layer topological information can be most effectively integrated into a comprehensive transferability score remains an open research question.

4.7 Conclusion

The overarching purpose of this work is to advance an agnostic framework for the interpretability of neural networks through the lens of computational topology. While the empirical evaluations throughout this work focus on image data, the theoretical models developed herein maintain a data-agnostic orientation. The experimental sections leverage the topological complexes best suited to the specific structure of the embedding space; specifically, the voxel-like grid structure of Convolutional Neural Network feature maps necessitated the use of cubical complexes.

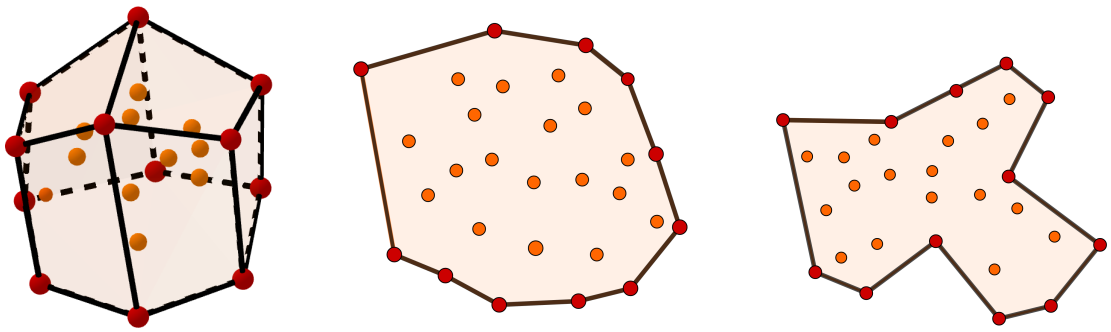
The analysis of our experimental results is fundamentally guided by the theory of topological expressivity [3], wherein the"hardness" of a dataset is approximated by its homological complexity ω. This interpretation is inherently task-driven; for instance, the phenomenon of collapsing Betti numbers in this work is dictated by the objective of the classification task. To further illustrate this task-dependent nature, Chapter 8 explores the evaluation of generative models via persistence vectorisation derived from cubical homology. To achieve a feasible computation of $\omega(\mathcal{X}_c^\ell)$ in deep architectures, Chapter 5 introduces archetypal subspaces as low-dimensional topological proxies. These subspaces allow for the approximation of the homological capacity Φ_f through the Topological Average Life metric, as explored in Chapter 6. Furthermore, Chapter 7 provides an analysis of how various $\omega(\mathcal{X}_c)$ evolve dynamically throughout the training process under different activation functions.

Crucially, an effective TL (Transfer Learning) score must account for both computational cost and model performance. The homological capacity (Φ_f) serves as a measure to approximate a shallower neural network and a robust mechanism for the elimination of unsuitable architectures rather than the definitive identification of an optimal model. Establishing a comprehensive TL score will require deeper scrutiny of the embedding space; while this work prioritises a topological perspective, the necessity of such an analysis is widely supported by existing literature in the field of transferability metrics.

Ultimately, the framework of homological capacity Φ_f provides a formal answer to the challenge of choosing the optimal architecture, a choice which is fundamentally influenced by the shape of the classes in the dataset as illustrated in Figure 4.1. The derived proxy for Homological capacity Φ_f thus serves as a definitive measure for this 'topological shape,' enabling a systematic approach to compact model selection for the downstream dataset.

Chapter 5

Archetypes



(a) 3D Convex hull (b) 2D Convex hull (c) 2D Concave hull

Figure 5.1: **Convexity and Topology:** How does the property of convexity influence machine learning and topology? Archetypes/ Archetypal analysis ($\mathring{A}$) in particular is related to the edge points (red) in the figures, while the hull comprises the interior points (orange) also.

5.1 Introduction

In the previous chapter, we saw that the topological transition of independent images in a CNN is unique to the neural network's architecture. Furthermore, we demonstrated that metrics derived from these transitions, such as θ, can effectively measure the generalisation of a learned representation to a downstream task. While Theorem 4.3 establishes that it is possible to define a homological capacity Φ_f over the depth L, this metric fundamentally requires the computation of the homological complexity $\omega(\mathcal{X}_{\hat{c}})$. In the preceding experiments, this was approximated by polynomially interpolating homological features from individual images.

A more comprehensive approach would be to interpolate directly through the global transitional homological complexities $\omega(\mathcal{X}^{\ell})$ or the class-specific transitional homological complexities $\omega(\mathcal{X}_c^{\ell})$. How do we estimate the topological transition of the embedding space of the entire dataset $\omega(\mathcal{X}^{\ell})$ or class $\omega(\mathcal{X}_c^{\ell})$? This problem is addressed in Chapter 6 utilising Archetypal Subspaces [80]. Specifically, this chapter establishes the foundational theorems required for the extension of the Homological Generalisation Theorem [29] to the subspace of Archetypes, a theoretical development that is subsequently and formally proved in Chapter 6.

To study the topological structures underlying the layer-by-layer topological transition, one needs to overcome the computational complexity of persistent homology. While several dimensionality reduction methods exist, categorised by (a) Feature space, (b) Instances, and (c) complex as seen in Chapter 3, this work approaches the problem through "Archetypal Analysis" [80] of the dataset/ point cloud (D of $\mathcal{D}$). These geometric objects are related to the convexity of the space, and due to their geometric positioning at

the "extreme points", it leaves room for interpretability of the reduced subspace. This chapter introduces Archetypes ($\mathring{A}$) and their counterparts, proceeding to analyse the "Topology of Archetypes" ($\mathcal{X}_{\mathring{A}}$ with respect to the underlying topological space $\mathcal{X}$). The investigation includes the following results:

Subspace Approximation : How to utilise archetypal subspace to approximate the homology of the dataset and its limitations?

Geometric Influence : How the geometric properties of the $\mathring{A}$ influence the lifetimes in the persistence diagram *dgm* of the dataset?

Topological Inference : How can the Persistent homological inference theorem be employed to study the underlying homology of $\mathcal{X}$ and $\mathcal{X} \cup \mathcal{X}_{\mathring{A}}$

By investigating these properties, this chapter provides a rigorous account of the topology of the underlying distribution and its archetypes. The theoretical results established here serve as the necessary framework for the results explored in Chapter 6& 7, where the dynamic relationship between the homology of archetypes and the global dataset is utilised to study the topological transitions of neural networks.

5.1.1 Archetypal Analysis

Archetypal Analysis [80, 81, 82] is uniquely defined as a constrained approximation to the convex hull (conv(D)) [12, 13]. This property ensures that the identified extreme points maintain a high degree of closeness to the data. To understand the geometric properties and influences of archetypes, it is essential to examine the roles of "convexity" and "convex hull" in machine learning. The property of "convexity " is often discussed, particularly in the context of optimisation. For example, in neural networks, a non-convex (concave) loss function is often employed in a nonlinear Network, which leads to countless local minima [83]. Its geometric implications are equally vital for the classification task. For instance, the separating hyperplane theorem [84], based on convex sets, illustrates the existence of a hyperplane separating the sets provided their intersection is empty. In a neural network, the classification problem can be viewed as a layer-by-layer transformation that attempts to construct a representation in which the classes do not intersect, that is, the convex hulls of each class do not overlap. In Chapter 6, it is precisely the archetypes of the class that are utilised to study topological transitions. Their proximity to the convex hull makes archetypes an ideal choice for analysing class separation within the representation.

Definition 5.1 (Archetypal Analysis [85])**.** *Let D be a point cloud, then for $\forall x_j \in D$ and $\hat{x}$, the center of D. Let m be the rank of the vector space of D, $F(\mathring{A}) = [1/|D| \sum_{i=1}^{|D|} \rho^2(x_i, conv(D))]^{1/2}$ and ρ is the metric for the underlying metric space (here Euclidean space). We will denote the archetype of a point cloud D as $\mathring{A}_D$ and the number of archetypes $n = |\mathring{A}|$.*

$$\textit{Objective function: } \min_{\mathring{A}} F(\mathring{A}), \tag{5.1}$$

$$\textit{Geometric property: } \rho(a_i, \hat{x}) \geq \rho(x_j, \hat{x}) \ \forall a_i \in \mathring{A}, \tag{5.2}$$

$$\textit{Basis Property : } \sum_{i=1}^{|\mathring{A}|} \alpha_i a_i = x_j, \ a_i \in \mathring{A} \textit{ and} \sum_i \alpha_i = 1, \tag{5.3}$$

where $\bar{F}(\mathring{A})$ represent the value of the objective function in Eq.(5.1) after optimisation.

5.1.2 Geometric and Topological Alternatives

The efficacy of machine learning is frequently bottlenecked by the relationship between model capacity and data dimensionality. A promising machine learning approach would be optimised to perform well

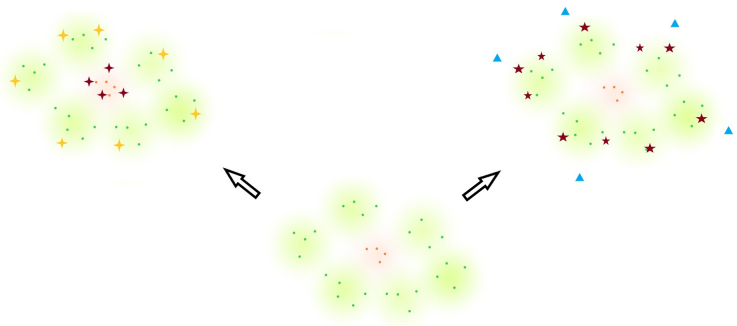


Figure 5.2: **NMF and AA**: The centre figure illustrates a two-class point cloud (green and orange). On the right, blue triangles and red stars denote the NMF and archetypes of the combined cloud, while the left figure highlights the individual class archetypes (Table 5.1 for examples of persistence diagrams and Betti curves of $\mathring{A}$.)

with limited resources. There are many popular methods of dimensionality reduction, such as Principal Component Analysis (PCA) [11], which preserves variance; Kernel Principal Component Analysis, which extends PCA to kernel space; and Singular Value Decomposition, which retains the most significant singular values or information. A similar approach to archetypal analysis is Non-negative Matrix Factorisation (NMF) [86], which extends principal component analysis with non-negativity constraints. Independent Component Analysis [11] is a linear dimensionality reduction method that retains statistically independent components, while Factor Analysis groups correlated features to reduce dimensionality. Since here we are discussing topology, a popular dimensionality reduction based on the preservation of the broad topological structure is UMAP [54] (see Chapter 3).
Here, the problem requiring dimensionality reduction is the dimension of the input processed by persistent homology. The computational complexity of simplex-based persistent homology (the traditional method) is highly influenced by the number of samples in the dataset rather than the dimensionality of the instances/points. This problem is conventionally solved by defining complexes such as the Delaunay complex, the witness complex, and others (see Chapter 3). However, these approaches result in a loss of both information and interpretability. There exist other methods of subsampling, such as the average landscape [59], which approximate the persistence landscape vectorisation by randomly subsampling the dataset; a similar approach is applied in Chapter 8. However, here, the geometric properties of Archetypes are used. One can approach the problem from the perspective of kernel space and extend it to persistent homology. This is discussed further in Chapter 7. Rather, here the inference theorem is exploited to meet the homological requirements, establishing them as a possible subsampling for studying the Embedding space in Chapter 6.

Before discussing $\mathring{A}$, we look at a similar interpretable representation of the dataset called Non-negative Matrix Factorization [86]. In the original article by Cutler and Breiman [80], $\mathring{A}$ is a formulation of Non-negative Matrix Factorisation that is uniquely defined with a useful notion of optimality. Non-negative Matrix Factorisation (NMF) was motivated by principal component analysis. NMF may lie considerably far from the data or the convex hull without learning the "shape" of the data, so their structure is less interpretable as compared to Archetypes. They are only constrained by the non-negative matrix requirement in the optimisation. Archetypes, on the other hand, are geometrically constricted to the convex hull (principal convex hull) as a convex combination of the data, often acting as a basis for the data when every point in the data can be represented as a convex combination of the archetypes [80]. Archetypes are known as "pure" types, representations of the data, and are often not points of the dataset.

Archetypes optimisation [80] over the point cloud D have the following properties (let the number of archetypes be $n = |\mathring{A}|$):

1. For $n = 1$, then $\mathring{A} = \{\sum_{i=1}^{|D|} x_i\}, x_i \in D$.
2. For $1 < n < |D|$, there exist solution to the archetypal analysis optimisation problem and $a_i \in \mathring{A}$ then $a_i \in \text{conv}(D)$.
3. $n \to |D|, \bar{F}(\mathring{A}) \to 0$ and $\mathring{A} = D$

Solving the optimisation problem of archetypal analysis [82, 87] involves an $O(n|D|M)$- iteration, where M is the dimensionality of $x_i \in D$. Apart from the theoretical results, utilising only the archetypes of the class feature space. The experiments also explore how resizing the image and the number of archetype samples affect the persistence vectorisations.

5.2 Experiment

In the preceding section, the datasets are treated as point clouds and persistent homology is constructed using simplicial complexes. We denote D as the set of point clouds. For the image datasets, the images are converted to greyscale, normalised, and then flattened to represent an instance/point of the point cloud. Archetypes are computed over this point cloud, often utilising a dimensionality reduction technique to alleviate computational overhead. Images are resized using downsampling or upsampling with standard interpolation methods.

Notation (Subsample of Dataset)**.** *Let D denote the feature space of dataset $\mathcal{D}$. The term R_D is used to represent random sampling from D, while $\mathring{\text{A}}_D$ represents Archetypes from dataset D.*

The sampling R_D serves as a baseline for the global distribution, whereas $\mathring{A}_D$ captures the extreme points defining the boundary of the dataset.

Notation (Subsample of class)**.** *Let R_C represent the feature space of random balanced sampling from each class, where C is the set of class indices. $\mathring{\text{A}}_C = \bigcup_{c \in C} A_c$ is used to represent archetypes sampled from each class $c \in C$. In the experimental setup, the number of archetypes $|\mathring{\text{A}}_c|$ for each class is the same, thereby keeping the classes balanced.*

These class based partitions $\mathring{A}_c$ significantly reduce the computational overhead for the archetype. Further, the extension of the Homological Generalisation Theorem to archetypal subspaces, developed in Chapter 6, is based on these class archetypes $\mathring{A}_c$.

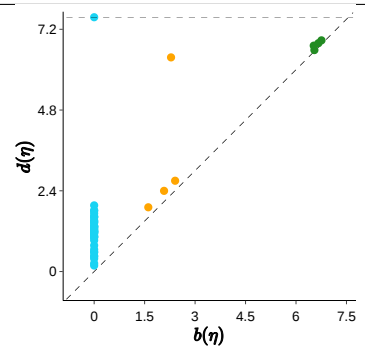
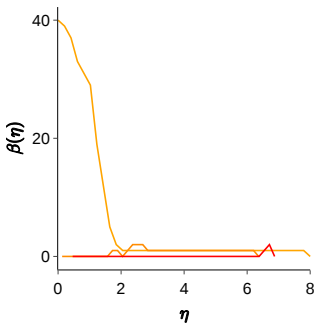
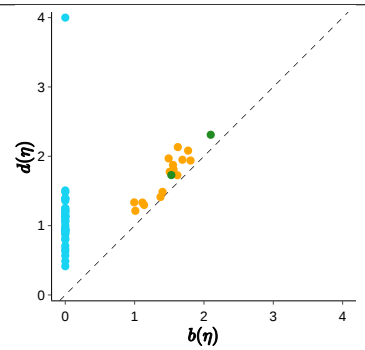
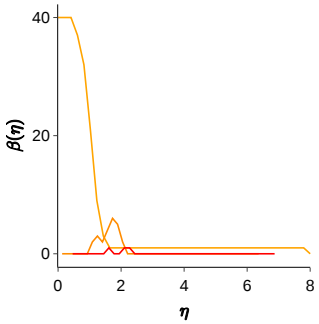
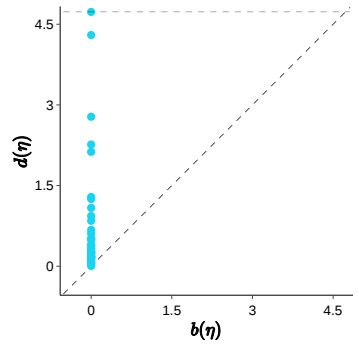
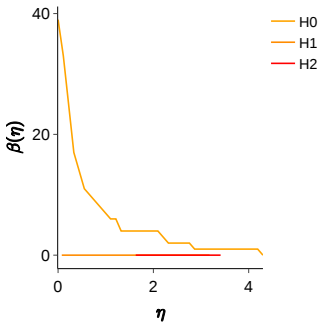
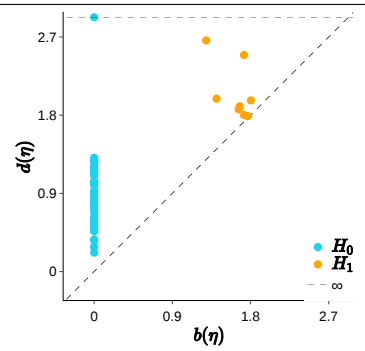
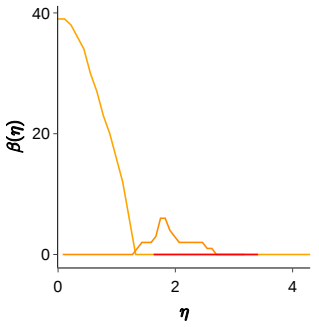
Notation (Nearest point)**.** *Let $\mathring{\text{A}}_{np(D)} = \{x_{\widehat{j}} \in D : \widehat{j} = \arg\min_j \rho(x_j, a_i) \forall a_i \in A_D\}$, denote the set of points in D proximal to the archetypal set. To maintain the condition $|A_D| = |\mathring{\text{A}}_{np(D)}|$, the points are computed via selection without replacement. As previously noted, archetypes are not points in the dataset; therefore, the distance measure ρ (here, Euclidean distance [4]) is used to approximate the data points in D that are closest to the points in $\mathring{\text{A}}$.*

Since archetypes are frequently synthetic, this approximation identifies the observed data points most representative of the archetypal extremes for empirical validation.

5.3 Persistent homology with Archetypes

Comparing the persistence diagrams dgm and Betti curve $\widehat{\beta}(\eta)$ presented in Table 3.1 in Chapter 3 and the Table 5.1, it is evident that homological features in the archetypal representation tend to persist longer than those in the original point cloud. The longer-persisting features $\widehat{H_K}(D)$ in the $dgm(D)$ are associated with the underlying homology $H_k(\mathcal{X}_D)$ of the topological space $\mathcal{X}_D$ associated with the point

Table 5.1: Persistence diagrams and Betti curves for archetypes $\mathring{A}$across different datasets (referencing Table 3.1). For the different datasets, we compare linear archetypes ($\mathring{A}$) and kernel archetypes (KAA) [87].

Space $\mathcal{X}$	Persistence Diagram	Betti Curve $\widehat{\beta}(\eta)$
Torus ⊔ Sphere ($\mathring{A}$)		
Torus ⊔ Sphere (Kernel)		
Two Figure-Eights ($\mathring{A}$)		
Two Figure-Eights (Kernel)		

cloud D. Correspondingly, the Betti number associated with $\widehat{H_k}(D)$ at scale η in the $dgm(D)$ is denoted by $\widehat{\beta_k^D}(\eta)$ while $\beta_k(\mathcal{X}_D)$ denote the $k-$th Betti number associated with the $k-$th homology $(H_k(\mathcal{X}_D))$ of the underlying topological space $\mathcal{X}_D$. Typically, the filtration in persistent homology (such as for Vietoris-Rips), as η increases, points merge. Eventually, at some threshold $\hat{\eta}$, $\Gamma_{\hat{\eta}}$ forms a single, large connected component that is homologically trivial, characterised by the Betti Number $\beta = (1, 0, 0, 0, \dots)$ and denoted as $H(\mathbf{B}_0)$. This final, single connected component "lives forever" and lasts a lifetime, or ∞. In this work, we consider the simplicial complex Γ_η to be homotopy equivalent $\mathbf{B}_0$ $(H(\Gamma_\eta) \cong H(\mathbf{B}_0))$, whenever the filtration parameter η meets or exceeds the maximum pairwise distance between all points in the set. While the relationship between diameter and contractibility can involve deeper topological nuances in general metric spaces, we adopt this threshold condition for the finite point sets analysed in this thesis, leaving more generalised results for future studies.

Let Γ_η be the filtered simplicial complex then :

$$H_k(\Gamma_0) \xrightarrow{\partial_0} H_k(\Gamma_1) \xrightarrow{\partial_1} H_k(\Gamma_2) \xrightarrow{\partial_2} \cdots \xrightarrow{\partial_{\eta-2}} H_k(\Gamma_{\eta-1}) \xrightarrow{\partial_{\eta-1}} H_k(\Gamma_\eta) \cdots \xrightarrow{\partial_{\hat{\eta}}} H_k(\mathbf{B}_0)$$

Notation. *Throughout this thesis, wherever necessary, we assume that the archetypes* $\mathring{\text{A}}$ *satisfy the following property:* $diam(\mathring{\text{A}}_D) \geq diam(conv(D))$*, where* $diam(\cdot)$ *denotes the diameter of the set defined over the Euclidean space [4] with metric* ρ*. We refer to this requirement as the* ***Extremal Preservation Property for Archetypal Analysis*** *(EPA).*

In addition to the aforementioned properties of the $\mathring{A}$, we further impose the constraint, the Extremal Preservation Property on Archetypal Analysis. Although archetypal analysis is often referred to as **Principal Convex Hull Analysis** [81], implying that archetypes serve as the extremal vertices of the $\text{conv}(D)$, optimisation objectives typically prioritise representing the densest regions of the boundary to minimise reconstruction error. Without the EPA, the archetypal set might capture the dense boundary regions but fail to encompass the full spatial extent of the point cloud D.This would undermine the representative nature of $\mathring{A}$ by excluding the true geometric extremes of the underlying manifold. Thus, the EPA property can be used to explain how $\Gamma_\eta(\mathring{A}_D)$ acts as a topological envelope for $\Gamma_\eta(D)$. This relationship is reflected in the computed persistence-based metrics, such as the homological complexity curve ω_η and the topological average life λ, as demonstrated by the empirical results presented in the following sections. The theorems derived here provide the formalisation for these observations. Therefore, the alignment between our theoretical and experimental results indicates that the EPA property is indeed preserved in the generated archetypes $\mathring{A}$.

Theorem 5.1. *Let* $\Gamma_\eta(D)$ *and* $\Gamma_\eta(\mathring{\text{A}}_D)$ *be the simplicial complex at* η *during filtration for* D *and* $\mathring{\text{A}}_D$ *respectively. Assuming EPA holds, for a* η *if there exist a* $k-$*homology in* $\Gamma_\eta(D), k \geq 1$ *then* $H(\Gamma_\eta(\mathring{\text{A}}_D)) \ncong H(\mathbf{B}_0)$*.*

Proof. The proof is by contradiction. Suppose that at the threshold value η, there exist a $k-$homology in $\Gamma_\eta(D), k \geq 1$ and $H(\Gamma_\eta(\mathring{A}_D)) \cong H(\mathbf{B}_0)$. This implies that $\Gamma_\eta(\mathring{A}_D)$ consists of a single connected component that is contractible, which require all the points $\mathring{A}_D$ to be pairwise connected within the distance η:

$$\forall a_p, a_q \in \mathring{A}_D, \rho(a_p, a_q) \leq \eta \implies \text{diam}(\mathring{A}_D) \leq \eta$$

Since the EPA property holds for $\mathring{A}_D$, we have :

$$\text{diam}(D) \leq \text{diam}(\text{conv}(D)) \implies \text{diam}(D) \leq \text{diam}(\mathring{A}_D) \leq \eta$$

Then $\text{diam}(D) \leq \eta \implies \rho(x_i, x_j) \leq \eta \forall x_i, x_j \in D$. That is the pairwise distance in D are all less than η and $\Gamma_\eta(D)$ will be a single connected component that is contractible:

$$H(\Gamma_\eta(D)) \cong H(\mathbf{B}_0)$$

This contradicts the initial assumption that $\exists k \geq 1$, such that $H_k(\Gamma_\eta(D)) \geq 1$. Therefore, $\Gamma_\eta(\mathring{A}_D)$, cannot have collapsed to a single component and $H(\Gamma_\eta(\mathring{A}_D)) \ncong H(\mathbf{B}_0)$. □

The above theorem implies that the filtration converges to the connected component for D faster than $\mathring{A}$ and further note that the above theorem can also be rewritten as the following "For a η if $H(\Gamma_\eta(D)) \ncong H(\mathbf{B}_0)$ then $H(\Gamma_\eta(A)) \ncong H(\mathbf{B}_0)$, provided EPA holds for $\mathring{A}$". This relationship holds because archetypes, by definition and under the EPA, represent the extremal points of the dataset, effectively "bounding" the spatial extent of the point cloud. Based on the same we can derive the following two theorems for homological complexity ω_η and average life $\lambda = \frac{\sum_{i=1}^{|\widehat{H}|}(d_i - b_i)}{|\widehat{H}|}$, where $|\widehat{H}|$ denotes the total number of homology in the *dgm* that have lifetime $d_i - b_i < \infty$. Consequently, while we establish this relationship based on the diameter threshold for the scope of this thesis, the characterisation of more intricate contractibility properties in relation to archetypal structures remains an open topological question which we leave for future investigation.

5.3.1 Homological complexity of Archetypes

We have the following theorem as a direct consequence of the property that archetypes converge to the dataset when $|\mathring{A}| = |D|$.

Notation. *Throughout this section, for the point cloud D with class index set C, $N = |D|$ denotes the total number of instances in the point cloud, $n = |\mathring{\mathrm{A}}|$ denotes the number of archetypes, and N_c denotes the number of points associated with a specific class $c \in C$.*

These variables are utilised to study the properties of the homological complexity of the archetypal subspace. Specifically, the relationship between these quantities enables a formal comparison of the topological features of the original point cloud and those of its archetypal representation.

Lemma 5.2. *Suppose there are N points in $D \subset \mathbb{R}^p$, and we have n archetypes $\mathring{\mathrm{A}}_D$. Then as $n \to N$:*

$$\lim_{n \to N} \left\lceil \frac{N}{n} \right\rceil \omega_\eta(\mathring{\mathrm{A}}_D) = \omega_\eta(D) \quad \forall \eta.$$

Proof. Since $n \in \mathbb{N}$ is strictly sequential, the limit of the sequence at $n = N$ is:

$$\lim_{n \to N} \left\lceil \frac{N}{n} \right\rceil = 1.$$

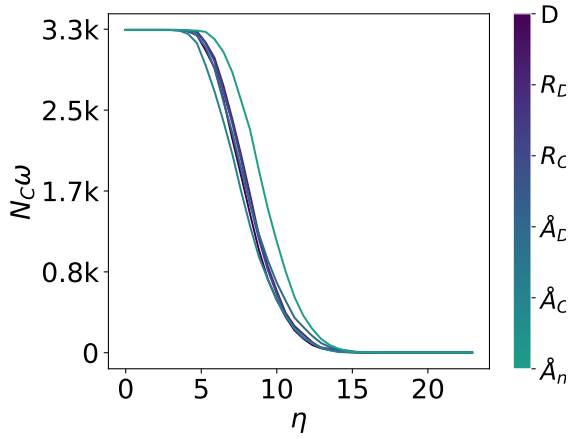
From the convergence property of archetypal analysis, as $n \to N$, $\mathring{A}_D \to D$. Then for all η :

$$\lim_{n \to N} \left\lceil \frac{N}{n} \right\rceil \omega_\eta(\mathring{A}_D) = \lim_{n \to N} \left\lceil \frac{N}{n} \right\rceil \cdot \lim_{n \to N} \omega_\eta(\mathring{A}_D) = 1 \cdot \omega_\eta(D).$$

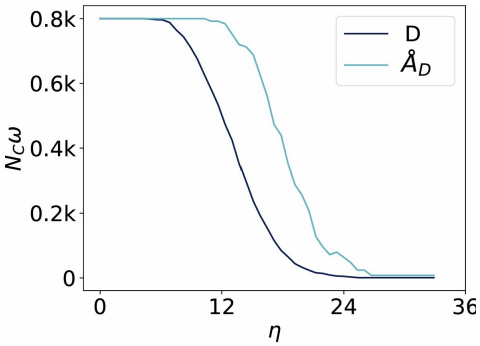
□

This result does not require the EPA property and holds for all η, but within the optimisation property, as $n \geq N$, the archetypes converge to the dataset. Now consider the following Figure 5.3, the lighter blue lines associated with the $\mathring{A}_D$ are upper bounding the $\omega_\eta(D)$ shown in darker blue for all η and $n \leq N$. In Theorem 5.3 and its corollary, we attempt to generalise the empirical results provided there, a surjective simplicial map g between $\Gamma_\eta(D)$ and $\Gamma_\eta(\mathring{A}_D)$. The results are based on the results from [88]. The experimental results presented in this section explore these relationships for homological dimensions $k = 0, 1$ and 2.

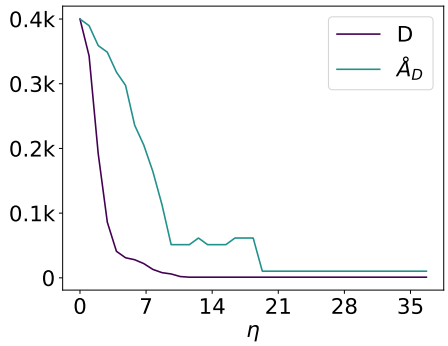
Theorem (1.2, [88])**.** *Let $g : \Gamma \to \hat{\Gamma}$ be a simplicial map of finite simplicial complexes such that $\chi(g^{-1}(\hat{\Gamma})) = \mathbf{r}$ $\forall \hat{\Gamma} \in \hat{\Gamma}$ where $\chi(\Gamma)$ is the Euler characteristic of Γ and g^{-1} denotes the preimage of $\hat{\Gamma} \in \hat{\Gamma}$. Then, $\chi(\Gamma) = \mathbf{r} \cdot \chi(\hat{\Gamma})$.*



(a) AIRCRAFT



(b) Airplane-STL10



(c) Torus

Figure 5.3: **Upper bound for homological complexity :** The Figures (a) AIRCRAFT dataset (b) Airplane class of STL10 dataset, and (c)Torus point cloud, respectively. We observe that $N_C\omega_\eta(\mathring{A}_{np(C)})$ can provide an upper bound for a constant N_C for the homological complexity of the data ω_η. Here, N_C is as in the corollary 5.1. Best viewed in colour.

The result relates the Euler characteristics of simplicial complexes under a specific class of simplicial maps. Now, we provide a similar characterisation for the homological complexity ω of the simplicial complex $\Gamma_\eta(D)$ obtained for a threshold η in a PD for D, and $\Gamma_\eta(\mathring{A}_D)$ is the simplicial complex for the same η in a PD for the archetypes $\mathring{A}_D$.

Theorem 5.3. *Suppose there are N points in $D \subset \mathbb{R}^p$, and we have n archetypes $\mathring{\text{A}}_D$. Provided $n \geq m+1$ where m is the minimum number of $\mathring{\text{A}}_D$ needed to span D, for those thresholds η in the PD such that there exists a surjective map between $\Gamma_\eta(D)$ and $\Gamma_\eta(\mathring{\text{A}}_D)$, we have that:*

$$\omega_\eta(D) \leq \left\lceil \frac{N}{n} \right\rceil \omega_\eta(\mathring{\text{A}}_D). \tag{5.4}$$

Proof. Here we demonstrate that $\mathbf{r} = \left\lceil \frac{N}{n} \right\rceil$. This is readily inferred by considering that any simplicial map from $\Gamma_\eta(D)$ to $\Gamma_\eta(\mathring{A}_D)$ needs to map N vertices of D to n vertices of $\mathring{A}_D$. Thus, a preimage of the simplicial map can contain at most $\left\lceil \frac{N}{n} \right\rceil$ vertices. As a result, for the base case of the proof in the work [88], $\mathbf{r} = \left\lceil \frac{N}{n} \right\rceil$, and we have that:

$$\hat{\beta}_0^D(0) = \left\lceil \frac{N}{n} \right\rceil \hat{\beta}_0^{\mathring{A}_D}(0).$$

Next, for any finite η, due to the defining property of distances for archetypes as in Eqn. (5.1), we have that:

$$\hat{\beta}_0^D(\eta) \leq \left\lceil \frac{N}{n} \right\rceil \hat{\beta}_0^{\mathring{A}_D}(\eta).$$

The rest of the proof follows from an induction argument for $\omega_\eta(D)$ and $\omega_\eta(\mathring{A}_D)$ following a similar line of reasoning as in the study [88] for $\chi(\Gamma)$ and $\chi(\hat{\Gamma})$. □

The following corollary extends the homological complexity bound to a partitioned dataset. For this result, we assume the existence of a surjective simplicial map $g_c : \Gamma_\eta(D_c) \to \Gamma\eta(\mathring{A}_c)$ for each class $c \in C$.

Corollary 5.1. *Suppose we have a finite point cloud* $D = \bigcup_{c \in C} D_c$ *where* $D_c \subset \mathbb{R}^p$, $D_{c_i} \cap D_{c_j} = \emptyset$, *and* $\mathring{A}_C = \bigcup_{c \in C} \mathring{A}_c$ *is the union of* n *archetypes from each* c. *Provided* $n \geq m + 1$, *where* $m = \max_{c \in C}\{m_c\}$ *such that* m_c *is the minimum* $\mathring{A}_c$ *needed to span* D_c *and* $N = \sum_c |D_c|$. *Then we have that:*

$$\omega_\eta(D) \leq N_C \; \omega_\eta(\mathring{A}_C) \text{ where } N_C = \left\lceil \frac{N}{(n|C|)} \right\rceil \tag{5.5}$$

Proof. The proof extends Theorem 5.3 to the union of class-specific archetypes $\mathring{A}_C$. Here, the simplicial map $g : \Gamma_\eta(D) \to \Gamma_\eta(\mathring{A}_C)$ that takes every point in the D_c to $\mathring{A}_c$ such that the N vertices of D map to the $n|C|$ vertices of $\mathring{A}_C$. Thus $\mathbf{r} = N_C = \left\lceil \frac{N}{(n|C|)} \right\rceil$. Given the existence of a surjective map g, the bound on homological complexity follows the same inductive logic as Theorem 5.3. □

The result in Theorem 5.3 and Corollary 5.1 helps us to derive a relation between the homological complexity of D and $\mathring{A}_C$ (see Figure 5.3).

Corollary 5.2. *Let* $M = \max_\eta\{\omega_\eta(\mathring{A}_D)\}$ *and the topological space* $\mathcal{X}_D$ *of the point cloud* D *have at most* $\hat{k}$ *non-zero homology. If for each* $k \in \{0, 1, 2, \ldots, \hat{k}\}$, *there exists a threshold* η *such that the simplicial complex* $\Gamma_\eta(D)$ *captures the true Betti number* $\beta_k(\mathcal{X}_D)$, *then we have* $\omega(\mathcal{X}_D) \leq \mathbf{r} M \hat{k}$ *where* $N = |D|, n = |\mathring{A}_D|$ *and* $\mathbf{r} = \lceil \frac{N}{n} \rceil$.

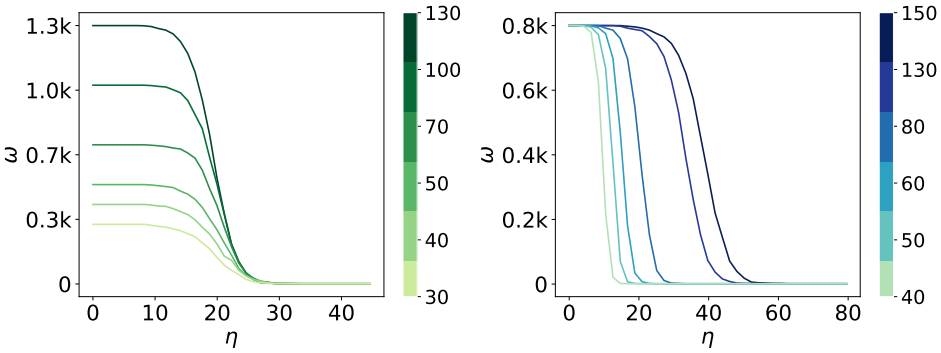
Proof. Given that the point cloud D has $\hat{k}$ non-zero homology. By definition, the total homological complexity of the underlying space is the sum of its Betti Number, $\omega(\mathcal{X}_D) = \sum_{k=0}^{\hat{k}} \beta_k(\mathcal{X}_D)$. We assume that for each dimension k, the empirical Betti number $\widehat{\beta_k^D}(\eta)$ (or $\Gamma_\eta(D)$) captures the true $\beta_k(\mathcal{X}_D)$ at some η. Since $\omega_\eta(D) = \sum_{k=0}^{\hat{k}} \widehat{\beta_k^D}(\eta)$ it follows that $\max_\eta\{\omega_\eta(D)\}$ serves as an upper bound for any individual $\beta_k(\mathcal{X}_D)$. Thus $\omega(\mathcal{X}_D) \leq \sum_{k=0}^{\hat{k}} \max_\eta\{\omega_\eta(D)\}$. Applying the bound from Theorem 5.3 where $\max_\eta\{\omega_\eta(D)\} \leq \mathbf{r} \max_\eta\{\omega_\eta(\mathring{A}_D)\}$:

$$\omega(\mathcal{X}_D) \leq \sum_{k=0}^{\hat{k}} \mathbf{r} \max_\eta\{\omega_\eta(\mathring{A}_D)\} \implies \omega(\mathcal{X}_D) \leq \mathbf{r} \max_\eta\{\omega_\eta(\mathring{A}_D)\}\hat{k}$$

Putting $M = \max_\eta\{\omega_\eta(\mathring{A}_D)\}$ we have $\omega(\mathcal{X}_D) \leq \mathbf{r} M \hat{k}$. □

The proof is a direct consequence of theorem 5.3 and the insight that there is at most $\hat{k}$ non zero Betti numbers $\beta_k(\mathcal{X}_D)$ and each of these are upper bounded at some η by $\mathbf{r}\, \omega_\eta(\mathring{A}_D)$. In this framework, persistent homology is utilised as the primary tool for capturing the intrinsic Betti numbers $\beta_k(\mathcal{X}_D)$, under the expectation that the empirical value $\widehat{\beta_k^D}(\eta)$ successfully recovers these topological features at some point in the filtration. Although this assumption may not hold for all datasets, the total number of

features in the persistence diagram $|\widehat{H}|$, provides a coarse upper bound for the homological complexity of $\mathcal{X}_D$ provided that every underlying homological feature of the point cloud is captured by at least one complex $\Gamma_\eta(D)$ during the filtration. Importantly, the bound proposed in Corollary 5.2 offers a significantly tighter characterisation than $|\widehat{H}|$ by leveraging the geometric scaling $\mathbf{r}$ of the archetypal subspace.



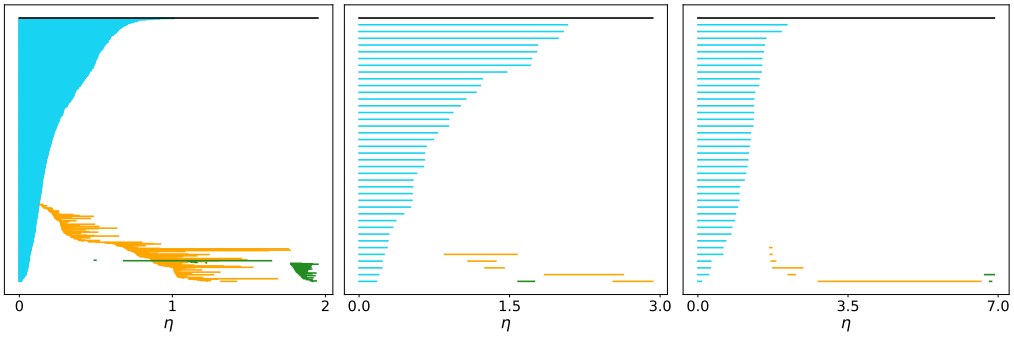
(a) Sampling complexity (b) Resize

Figure 5.4: **Dimensionality :** Sampling complexity denote Number of samples of archetypes $\mathring{A}$ and Resize denote The dimension of $a_i \in \mathring{A}$

Image Resize : The empirical results with respect to different resize (see Figure 5.4(b)) of the images show that ω_η undergoes a horizontal expansion as the dimensionality of the archetypes $a_i \in \mathring{A}$ increases. Let $\mathring{A}_D \subseteq \mathbb{R}^m$ and $\widehat{\mathring{A}_{\hat{D}}} \subseteq \mathbb{R}^{\hat{m}}$ be sets of archetypes derived from dimensionally reduced feature spaces D and $\hat{D}$ of the dataset $\mathcal{D}$ such that $|\mathring{A}_D| = |\widehat{\mathring{A}_{\hat{D}}}|$ and $m < \hat{m}$. Our observations suggest a scaling relationship such that $\omega_\eta(\widehat{\mathring{A}_{\hat{D}}}) \approx \omega_{\eta/\mathbf{r}}(\mathring{A}_D)$ where $\mathbf{r}$ is some constant. This horizontal stretching in the homological complexity curve demonstrates an increase in the persistence of certain homological features, likely those associated with the underlying manifold, across larger filtration scales η. This lack of variation along the vertical $y-$axis implies that the total number of homological features remains invariant. Thus suggesting that increasing the image resolution enhances the robustness of existing features without introducing new homologies.

Sampling : The empirical results demonstrate a vertical translation in the homological complexity ω_η as the sample size increases (see Figure 5.4(a)). Let $\mathring{A}_D$ and $\widehat{\mathring{A}_{\hat{D}}}$ be sets of archetypes derived from the same feature space $D \subseteq R^m$ such that $|\mathring{A}_D| < |\widehat{\mathring{A}_{\hat{D}}}|$. Our observations indicate a translational relationship in the homological complexity between the subsamples of archetypes given by $\omega_\eta(\widehat{\mathring{A}_{\hat{D}}}) = \omega_\eta(\mathring{A}_D) + \mathbf{r}$ where $\mathbf{r}$ is a constant. This is consistent with the nature of persistent homology, where the total number of extracted features $|\widehat{H}|$ is intrinsically linked to the density and cardinality of the initial point cloud. But this does not necessarily implies that all observed homological features are related to the underlying manifold. On the other hand, the larger subsample of archetypes provides a more granular representation of the manifold, thereby increasing the observed homological complexity without necessarily altering the persistence of the features.

Since the archetypes $\mathring{A}_D$ serve as a homological upper bound for the dataset D, it follows that the translational behaviours of the homological complexity ω_η observed during resizing and sampling would be similarly reflected in the point cloud D. These results are primarily restricted to the homological dimensions H_0, H_1 and H_2, for which the empirical computations were performed. While a general theoretical proof for higher dimensions remains beyond the scope of this work, it presents a promising direction for future research.



(a) TS (b) TSR (c) TSAA

Figure 5.5: **Lifetime of $\mathring{A}$:** (a) is the barcode of the Torus-Sphere (TS) dataset, (b) its $\mathring{A}$, and (c) Random sampling. Observe that the η values for archetypes are higher than those of the entire dataset, indicating that the homologies in the case of archetypes persist longer.

5.3.2 Average life of archetypes

The homology of a point cloud represents the connectivity and separability in the data. The more persistent the homology, the larger the gap. One may wonder whether a subset of the data exists that can explain this persistence across the entire dataset. With subsampling, we naturally end up in a space with more persisting homology as a result of sparsity, but this can again be very ambiguous in the sense of 'what is the higher persistence? 'How can we interpret the persistence arising?' (see Figure 5.4(b)). Thus 'average life' λ_D, which is the average behaviour of the persistence of the homologies in the *dgm* is used to interpret the lifetime ($d_i - b_i \geq 0$), where d_i and b_i are the η value of death and birth of the i^{th} homology that appear in the $dgm(D)$ with finite lifespan. Although the archetypes do not capture the homology, it can be observed that in Figure 5.5 each bar of the barcode for $\mathring{A}$ persists longer than that of D. This motivates us to explore archetypes $\mathring{A}$ (a derived subset of D) to approximate the 'persistence' of D.

Definition 5.2 (finite maximum lifetime)**.** *For a persistence diagram $dgm(D)$, we define the* ***finite maximum lifetime*** *as* $\lambda_\infty(D) := \max_i d_i - b_i, \forall (d_i, b_i) \in dgm, d_i - b_i < \infty$.

Recall that the topological average lifetime is defined as

$$\lambda(D) := \frac{\sum_{i=1}^{|\widehat{H}(D)|} d_i - b_i}{|\widehat{H}(D)|}$$

where $(d_i, b_i) \in dgm$ denote the $i-$th homology and $\widehat{H}$ denotes the set of all homological features in *dgm* that have finite lifetime.

Lemma 5.4. *Let $dgm(\mathring{A}_D)$ and $dgm(D)$ be the persistence diagrams for the archetypes and the point cloud, respectively. Under the* ***EPA****, it follows that:* $\lambda_\infty(\mathring{A}_D) \geq \lambda_\infty(D)$

Proof. By the **EPA**, we have $\text{diam}(\mathring{A}_D) \geq \text{diam}(\text{conv}(D))$, which guarantees the existence of a pair of archetypes $a_p, a_q \in \mathring{A}_D$ such that $\rho(a_q, a_p) \geq \rho(x_i, x_j) \forall x_i, x_j \in D$. Furthermore, from the Theorem 5.1, for threshold value η if $H(\Gamma_\eta(D)) \ncong H(\mathbf{B}_0)$ then $H(\Gamma_\eta(\mathring{A}_D)) \ncong H(\mathbf{B}_0)$, provided EPA holds. This implies that the homological features in $\mathring{A}_D$ do not collapse before those in D, as the archetypes represent the principal convex hull of D. Consequently, the threshold $\hat{\eta}$ required for $H(\Gamma_{\hat{\eta}}(\mathring{A}_D)) \cong H(\mathbf{B}_0)$, is at least as large as the threshold η required for $H(\Gamma_\eta(D)) \cong H(\mathbf{B}_0)$, (that is $\hat{\eta} \geq \eta$). Since the maximum finite lifetime of a feature is bounded by these collapse thresholds, we have $\lambda_\infty(\mathring{A}_D) \geq \hat{\eta} \geq \eta \geq \lambda_\infty(D) \implies \lambda_\infty(\mathring{A}_D) \geq \lambda_\infty(D)$. □

The proof is a consequence of Theorem 5.1, which states that a non-trivial homology cannot exist in $dgm(D)$ if the corresponding $H(\Gamma_\eta(\mathring{A}_D))$ has already collapsed to a single connected component $H(\mathbf{B}_0)$.

Theorem 5.5. *Let $dgm(\mathring{\mathrm{A}})$ and $dgm(D)$ be the persistence diagrams for the archetypes $\mathring{\mathrm{A}}_D$ with EPA property and the point cloud D, respectively. Then the average lifetimes satisfy the following relationship: $|\widehat{H}(\mathring{\mathrm{A}}_D)|\lambda(A_D) \geq \lambda(D)$, where $|\widehat{H}(\cdot)|$ denotes the number of finite lifetimes in the corresponding persistence diagram $dgm(\cdot)$.*

Proof. By the definition of average lifetime, the maximum finite lifetime is always greater than or equal to the average lifetime, that is $\lambda_\infty \geq \lambda$. From the previous lemma 5.4, $\lambda_\infty(\mathring{A}_D) \geq \lambda_\infty(D)$, which implies:

$$\lambda_\infty(\mathring{A}_D) \geq \lambda(D)$$

Furthermore, from the definition of average life, we have that the sum of all finite lifespans can be expressed as the product of the number of features and their average lifetime and since $d_i - b_i \geq 0 \; \forall i$:

$$|\widehat{H}(\mathring{A}_D)|\lambda(\mathring{A}_D) = \sum_{i=1}^{|\widehat{H}(\mathring{A}_D)|} (d_i - b_i) \implies |\widehat{H}(\mathring{A}_D)|\lambda(\mathring{A}_D) \geq \lambda_\infty(\mathring{A}_D)$$

Combining the inequalities, we obtain $|\widehat{H}(\mathring{A}_D)|\lambda(\mathring{A}_D) \geq \lambda_\infty(\mathring{A}_D) \geq \lambda(D)$. Thus, by the transitivity property:

$$|\widehat{H}(\mathring{A}_D)|\lambda(\mathring{A}_D) \geq \lambda(D)$$

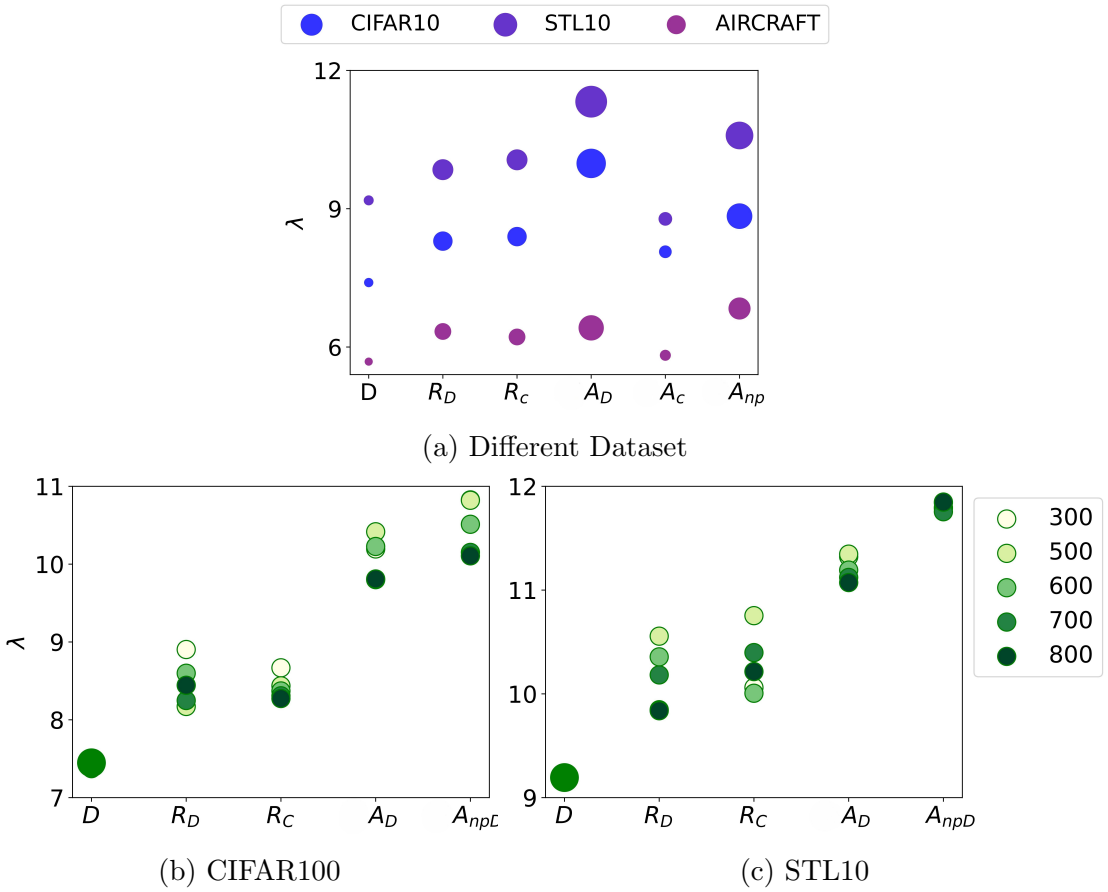
□

The Theorem 5.5 establishes a formal relationship between $\lambda(A_D)$ and $\lambda(D)$, that the average persistence of D can be bounded and approximated directly from $dgm(\mathring{A}_D)$. This result is computationally significant, as it indicates that the average homological characteristics of a large point cloud can be inferred from its archetypal representation, potentially eliminating the requirement to compute the full persistence diagram for the entire dataset D.

Dataset Sensitivity : As shown in Figure 5.6(a), the average lifetime varies across datasets, with the relationship $\lambda(STL10) > \lambda(CIFAR10) > \lambda(AIRCRAFT)$. Notably, the average lifetime of the corresponding archetypal representations follows the same ordering. While the dataset cardinalities follow $|CIFAR10| > |STL10| > |AIRCRAFT|$, the λ values do not correlate with sample size. This suggests that the average lifetime reflects the underlying manifold geometry rather than a simple function of the number of points.

Sampling Invariance : The Figures 5.6(b,c) show that increasing the sample size slightly decreases its average lifetime (λ). Nevertheless, these values remain consistently bounded by $\lambda_{\mathring{A}_D}$. These empirical observation aligns with Theorem 5.5, confirming that the archetypal subspace captures the most persistent homological features of the data. Note that this does not suggest any relation with $\lambda_{\mathring{A}_C}$.

These observations suggest that the average lifetime λ serves as a robust geometric descriptor that is sensitive to the underlying persistence of the topological structure of the data rather than its sampling density. Furthermore, it is observed that the comparative relation between the different datasets, $\lambda(STL10) > \lambda(CIFAR10) > \lambda(AIRCRAFT)$, is consistently preserved within the archetypal subspace $\lambda_{\mathring{A}_C}$. This consistent bounding of $\lambda_{\mathring{A}_C}$ validates the use of archetypes as a computationally efficient proxy for characterising the persistence boundary of high-dimensional point clouds.



(a) Different Dataset

(b) CIFAR100

(c) STL10

Figure 5.6: **Average life:** The Figure (a) shows the average life of different image datasets. Note that the $\lambda_S, S \subseteq D$ as absorbed the relation between the datasets $\lambda(STL10) > \lambda(CIFAR10) > \lambda(AIRCRAFT)$. The Figures (b) and (c) represent the average life computed for different numbers of instances. The results show that $\lambda_{\mathring{A}_D}$ is always greater than λ_D. (see theorem 5.5).

5.4 Homological inference over Archetypes

The **homological feature** of a topological space $\mathcal{X}$ denoted $hfs(\mathcal{X})$ is the smallest positive homological critical value [51] of η. Homological critical values are the values of η that induce a change in the homology. The homology inference theorem [5, 51] states that the quality with which a point cloud approximates the homology of an underlying space depends on the Hausdorff distance $\rho_{\mathbf{e}}$ between them (see Chapter 3). Note that archetypes are not necessarily points in D; rather, they are generally disjoint, that is, $\widehat{D} = D \sqcup \mathring{A}_D$. This proof relies on the assumption that archetypes constitute the extreme points of the data, thereby leaving the interior of the data manifold unaffected.

Theorem 5.6. *Let $\bar{F}(\mathring{A})$ represent the value of the objective function in Eq.(5.1) after optimisation. Let $\hat{\mathcal{X}}$ and $\mathcal{X}$ be the topological spaces associated with the point clouds $D \cup \mathring{A}_D$ and D, respectively. Then, $H(\hat{\mathcal{X}}) \cong H(\mathcal{X})$ (that is, the homology groups of $\hat{\mathcal{X}}$ and $\mathcal{X}$ are isomorphic) given $\bar{F}(\mathring{A})$ is sufficiently small.*

Proof. Let $\widehat{D} = D \cup \mathring{A}_D$ and D be point clouds which approximate the homology groups of their underlying topological spaces $\mathcal{X}$ and $\hat{\mathcal{X}}$ with $\eta \geq \rho_{\mathbf{e}}(\widehat{D}, D)$. Then by the homology inference theorem, we have $\dim \hat{H}_k(\widehat{D}) = \dim \hat{H}_k(D)\ \forall \eta > \rho_{\mathbf{e}}(\widehat{D}, D)$, and thus $\hat{\mathcal{X}}$ and $\mathcal{X}$ have isomorphic homology groups provided $\rho_{\mathbf{e}}(\widehat{D}, D)$ is sufficiently small. We complete the proof by actually showing that $\rho_{\mathbf{e}}(\widehat{D}, D) = \bar{F}(\mathring{A})$. To see this, consider the quantity $e(D, \widehat{D}) = \sup_{x \in D} \rho(x, \widehat{D})$ where $\rho(x, \widehat{D}) = \inf_{x \in \widehat{D}} \rho(x, a) = 0$ which follows since $D \subset \widehat{D}$. This implies that $e(D, \widehat{D}) = 0$. Now consider $e(\widehat{D}, D) = \sup_{a \in \widehat{D}} \rho(a, D)$.

Here $\rho(a, D) = 0\ \forall a \in D$ and $\rho(a, D) = \bar{F}(\mathring{A})\ \forall a \in \mathring{A}$. This implies that $e(\widehat{D}, D) = \bar{F}(\mathring{A})$. Thus $\rho_{\mathbf{e}}(\widehat{D}, D) = \max\{e(D, \widehat{D}), e(\widehat{D}, D)\} = \bar{F}(\mathring{A})$. Consequently, for a sufficiently small $\bar{F}(\mathring{A})$ it follows that $H(\mathcal{X}) \cong H(\hat{\mathcal{X}})$. □

This theorem implies that augmenting a dataset with its archetypal points preserves the underlying homology of the manifold, provided the objective function $F(\mathring{A})$ (for computing the archetypes [85]), is effectively minimised. Consequently, the homological integrity of the class features will also remain invariant under their archetypal augmentation within their approximation errors, the objective function serving as a formal bound on the homological stability of the augmented representation. Note that this Theorem 5.6 is independent of the EPA property and depends only on the value of $\bar{F}(\mathring{A})$.

5.5 Conclusion

This chapter investigated the interplay between the geometric properties of a point cloud and the homology based metrics derived from it. In particular, it is found that geometric properties of the subsample of archetypal analysis can be used to study the homological curve of the entire data. Towards the same, we formalise how the EPA and simplicial mapping affect the bounds of homological complexity and feature persistence. Finally, we established a connection to the homological inference theorem (Theorem 5.6), providing a theoretical basis for augmenting datasets with archetypes without distorting their underlying manifold structure.

While Theorem 5.5 provides a global bound for the topological average lifetime, it does not fully characterise the dataset specific variations observed in Figure 5.6 or the tight upper bound between the average life of the sample and its archetypes. To address this, we propose the following conjecture based on our experimental findings:

Empirical conjecture 5.1. *Given the persistence diagram of the point cloud D and its set of archetypes $\mathring{A}_D$, we have for the average life:*

$$\lambda(S) \geq \lambda(\mathring{A}_D), S \subseteq D \tag{5.6}$$

It is important to note that archetypes do not constitute a universal subsampling method for capturing topology unless the data resides on the convex hull and admits a homeomorphic mapping. We explore the relevance of this property with respect to studying the neural network embedding space in Chapter 7, which further visualises the decision boundary through kernel Archetypes for different activation functions. A primary concern regarding the use of archetypes involves their significant computational complexity. Effectively, we utilise the class archetypes while employing dimensionality reduction through downsampling to manage the instance dimension. But then one can observe in Figure 5.6 that the $\mathring{A}_C$ demonstrate a lower λ value than the R_C. This is due to the lack of preservation of EPA properties caused by an optimisation preference toward high-density regions within the data manifold. However, a comprehensive investigation into the most efficient algorithmic approaches remains beyond the scope of this work.

Nevertheless, the EPA property does not influence Theorem 5.6 and therefore does not affect our foundational theorems in the subsequent chapters. As the primary objective is to study the topological transitions of the embedding space of the neural network (discussed in Chapter 6), the problem effectively reduces to analysing the archetype subsample of the dataset. In particular, the average lifetime of the class-archetypal subspace is used to study topological transitions within the embedding space.

Through Corollary 5.1), it is confirmed that computing over class archetypes serves as a viable proxy for the homological complexity of the entire dataset. Thus, we have demonstrated that the geometric properties such as the Extremal Preservation Property (EPA) of the convex hull (see Figure 5.1) can be used to bridge the gap between computational observations and theory. Consequently, convexity through archetypal representation provides a mathematically tractable framework for analysing persistent homology. In summary, this chapter provides a comprehensive review of how certain persistent vectorisations of a dataset relate to its archetypal subspace.

Chapter 6

Interpreting Neural Network via Archetypes

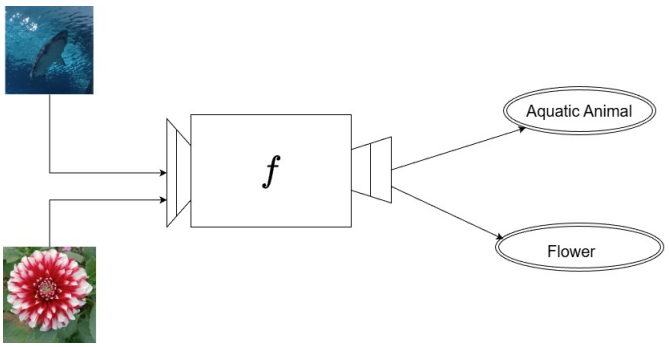


Figure 6.1: **Classification task**: This figure illustrates a classification task. The homological generalisation theorem [29] for the binary class states that for the support homology of the class of the function to be exactly equal to the homology of the class, the classifier f must perfectly classify.

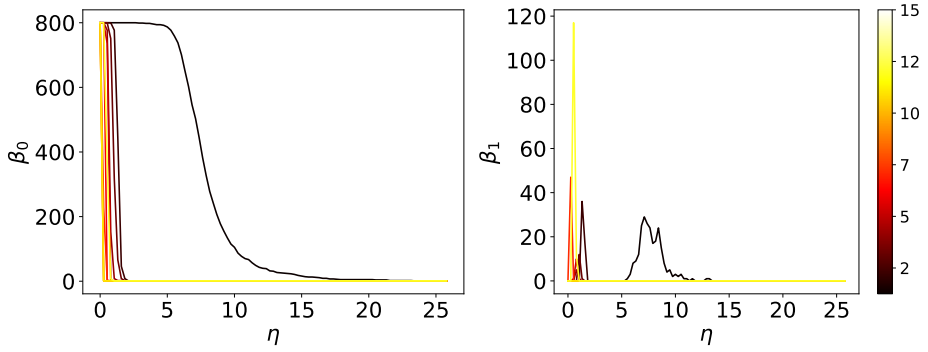
6.1 Introduction

Building on the theoretical framework established in the previous chapter, we now transition from the geometric properties of the data to the dynamic evolution of these features within a deep learning model. Having demonstrated through the Homological Inference Theorem that archetypal augmentation preserves the underlying manifold structure, in this chapter, we utilise the class archetypes ($\mathring{A}_c$) [80, 81, 82] to analyse the layer-by-layer transition of the embedding space through average lifetime λ. Here, the Homological Generalisation Theorem for binary classification is extended to the class archetypal subspace, enabling exploration of the feature space via persistent homology.

Classifier Topology : A study of how the homology of the archetypal subspace (see Figure 6.2) can be used to study the structural properties and decision boundaries of a classifier.

Empirical Analysis: An evaluation of how the archetypal representation $\mathring{A}_C$ of the data traverses the layers of the Neural Network. This includes an empirical exploration of the topological average lifetime λ of $\mathring{A}_C$ as it evolves through the Neural Network architecture.

Transfer Learning Score: The development of a Transfer Learning (TL) score based on the average lifetime λ which is correlated with model accuracy. This analysis relates the observed topological per-



(a) $\widehat{\beta}_0$ (b) $\widehat{\beta}_1$

Figure 6.2: **Betti curve:**Figures (a) and (b) give the Betti curve $\widehat{\beta}_0(\eta)$ and $\widehat{\beta}_1(\eta)$ across the blocks of DenseNet121 [24]. The legends correspond to different blocks of DenseNet121.

sistence to the homological capacity Φ_f discussed in Chapter 4, providing a comparative framework for evaluating the capacity of the network under transfer learning.

There are two reasons for extending homological generalisation to $\mathring{A}_c$, the archetypes of class c: 1) the classification of the $\mathring{A}_c$ is a broader generalisation task than the classification of the class alone, and 2) augmenting a class with its $\mathring{A}_c$ does not introduce any new homology to the class (see Chapter 5). Here, the mathematical definitions are as per the paper [29, 50], and D_c denotes the feature space of the class $c \in C$. Thus, this chapter aims to investigate the topological transition in the neural network by utilising A_c as proxies for D_c (see Figure 6.2).

6.2 Homological Generalisation Theorem for Classifiers

The topological structure of the decision region serves as a fundamental bottleneck for generalisation. While standard statistical learning theory focuses on the capacity of the function class measures like VC dimension [31], the following results demonstrate that the global connectivity and hole structure, captured by support homology, impose a "necessary" constraint on the capacity of a classifier. The following theory is proved only for the binary classification problem, and the theoretical premise for the homology theory based on category group is derived from the Book [4].

Definition 6.1 (Support Homology)**.** *Let $\mathcal{D} = \{(D_i, y_i) : y_i \in \{+, -\}\}$ be a dataset with D drawn from a joint distribution with continuous conditional distribution function on a topological space $\mathcal{X} \times \{0, 1\}$. Let $\mathcal{F} = \{f : \mathcal{X} \to \{0, 1\}\}$ be the collection of binary classifiers on $\mathcal{X}$. The support homology [4, 29] $(H_s f)$ of some function $f : \mathcal{X} \to \{0, 1\}$ is given by $H_s f := H[f^{-1}((0, \infty))]$.*

The following theorem establishes the relationship between the support homology of the function and the target class:

Theorem (3.1 of [29])**.** *If $\mathcal{X} = \mathcal{X}_- \sqcup \mathcal{X}_+$ and $\forall f \in \mathcal{F}$ with $H_s(f) \ncong H(\mathcal{X}_+)$, then $\forall f \in \mathcal{F}$, $\exists V \subset \mathcal{X}_+$ so that f misclassifies every $x \in V$.*

This principle implies that without homological alignment of f, there will always exist points of the data that are misclassified, regardless of how well the classifier trains. In the model selection problem, this condition allows us to eliminate those architectures that are incapable of producing a certain homological complexity (ω). For a binary classifier f, it is a characterisation of homologies in the positive decision boundary; therefore, here the same is denoted as $H_s f(\mathcal{X}_+)$.

$$\begin{array}{ccccc} H_S f(X_+) & \cong & H_S f(X_+ \cup \mathring{A}) & & H_S f(\mathring{A}_+) \\ \wr\| & \Longleftarrow & \wr\| & \Longrightarrow & \wr\| \\ H(X_+) & \cong & H(X_+ \cup \mathring{A}) & & H(\mathring{A}) \end{array}$$

Figure 6.3: **Lemma 6.2**: Given (blue) the isomorphism $H_s f(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+}) \cong H(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+})$. For a perfect classifier f, it implies isomorphism (blue) $H_s f(\mathcal{X}_+) \cong H(\mathcal{X}_+)$, $H_s f(\mathring{A}_{\mathcal{X}_+}) \cong H(\mathring{A}_{\mathcal{X}_+})$ and $H_s f(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+}) \cong H_s f(\mathcal{X}_+)$

6.2.1 Homological Generalisation Theorem Extended to Archetypes

To extend the theorem to include archetypes, let the augmented space be defined as $\mathcal{X} = (\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+}) \cup (\mathcal{X}_- \cup \mathring{A}_{\mathcal{X}_-})$. Let $H_s f(\mathcal{X}_c)$ and $H_s f(\mathring{A}_{\mathcal{X}_c})$ denote support homology for $\mathcal{X}_c$ (original data) and $\mathring{A}_c$ (class archetypal representations) respectively, where $c \in +, -$. We first extend that homological conservative theory to the class archetypal augmentations:

Corollary 6.1. *For a sufficiently small objective value $\bar{F}(A_c)$ sufficiently small. we have $H(\mathcal{X}_c) \cong H(\mathcal{X}_c \cup \mathring{\mathrm{A}}_{\mathcal{X}_c})$.*

Proof. The proof follows from the Theorem 5.6 in Chapter 5 as $\mathcal{X}_c$ and $\mathcal{X}_c \cup \mathring{A}_{\mathcal{X}_c}$ are the topological spaces associated with the point cloud $D = D_c$ and $D' = D_c \sqcup \mathring{A}_{D_c}$. Consequently, for a sufficiently small $\bar{F}(A_c)$, the isomorphism $H(\mathcal{X}_c) \cong H(\mathcal{X}_c \cup \mathring{A}_{\mathcal{X}_c})$ holds as a direct consequence of the Hausdorff distance bound under the Homological Inference Theorem. □

Provided the objective function is minimised to a sufficiently small value during the computation of the class archetypes, the resulting augmentation does not introduce significant extrinsic homological features into the underlying manifold under the Homological Inference Theorem [5]. To incorporate class archetypes into the framework of Theorem 6.2, we study the conditions for homological equivalence when the archetypal subspace is included. Following Theorem 6.2, we focus our analysis on the positive subspace ($\mathcal{X}_+$), an similar argument holds for the negative class ($\mathcal{X}_-$). The proof by the authors of [29] is by contradiction and states that if f maps each open set into its proper partition on X, we have $H(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+}) \cong H_s f(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+})$. We build on this key idea to prove our next result for archetypes.

Remark 6.1. *If a classifier f perfectly classifies for every point in the class then $H_s f(\mathcal{X}_+ \cup \mathring{\mathrm{A}}_{\mathcal{X}_+}) \cong H(\mathcal{X}_+ \cup \mathring{\mathrm{A}}_{\mathcal{X}_+})$, also $H_s f(\mathcal{X}_+) \cong H(\mathcal{X}_+)$ and $H_s f(\mathring{\mathrm{A}}_{\mathcal{X}_+}) \cong H(\mathring{\mathrm{A}}_{\mathcal{X}_+})$. Note that $H(\mathcal{X}_+)$ and $H(\mathring{\mathrm{A}}_{\mathcal{X}_+})$ are not required to be isomorphic in the general case.*

We leverage this insight to prove the following lemma, which establishes a fundamental property of a classifier f under the condition of perfect classification:

Lemma 6.2. *Let $f \in \mathcal{F}$ be a binary classifier on the data with associated topological space $\mathcal{X} = (\mathcal{X}_+ \cup \mathring{\mathrm{A}}_{\mathcal{X}_+}) \cup (\mathcal{X}_- \cup \mathring{\mathrm{A}}_{\mathcal{X}_-})$, that perfectly classifies then $H_s f(\mathcal{X}_+ \cup \mathring{\mathrm{A}}_{\mathcal{X}_+}) \cong H_s f(\mathcal{X}_+)$.*

Proof. Since f perfectly classifies, for $\mathcal{X}$, the support homologies for the positive class and its archetypal augmentation satisfy, $H_s f(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+}) \cong H(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+})$ and $H_s f(\mathcal{X}_+) \cong H(\mathcal{X}_+)$. From Corollary 6.1 we know that $H(\mathcal{X}_+) \cong H(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+})$. Thus, we have the following isomorphisms :

$$H_s f(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+}) \cong H(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+}) \cong H(\mathcal{X}_+) \cong H_s f(\mathcal{X}_+)$$

By transitivity, we have $H_s f(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+}) \cong H_s f(\mathcal{X}_+)$. □

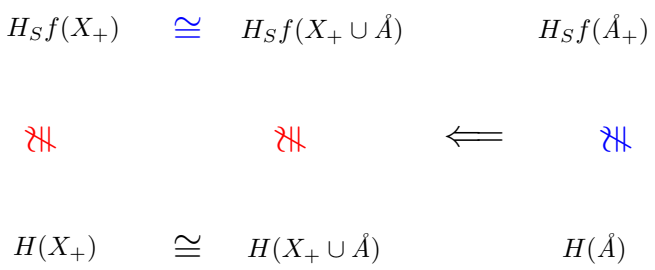


Figure 6.4: **Theorem 6.3:** Given (blue) $H_s f(\mathring{A}_+) \cong H(\mathring{A}_{\mathcal{X}_+})$ and $H_s f(\mathcal{X}_+) \cong H_s f(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+})$ then $H_s f(\mathcal{X}_+) \ncong H(\mathcal{X}_+)$

The above lemma is a consequence of Corollary 6.1 (see Figure 6.3). We now consider the converse: whenever the relation $H_s f(\mathcal{X}_+) \cong H_s f(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+})$ holds, we can extend the homological generalisation theorem to $\mathring{A}_{\mathcal{X}_+}$. We have the homological generalisation theorem extended to the $\mathring{A}_{\mathcal{X}_+}$. This extension suggests that the archetypal subspace serves as a homological proxy for the entire class. If a classifier fails to capture the homology of $\mathring{A}_{\mathcal{X}_+}$, it fundamentally lacks the capacity to generalise to the class manifold.

Theorem 6.3 (Homological generalisation extended to $\mathring{A}_{\mathcal{X}_+}$)**.** *Let X be a topological space, $\mathcal{X}_+$ be some open subspace, and $\mathring{\mathrm{A}}_{\mathcal{X}_+}$ its archetypes. Let $\mathcal{F} \subset 2^X$ such that $\forall f \in \mathcal{F}$ with $H_s f(\mathring{\mathrm{A}}_{\mathcal{X}_+}) \ncong H(\mathring{\mathrm{A}}_{\mathcal{X}_+})$ and $H_s f(\mathcal{X}_+) \cong H_s f(\mathcal{X}_+ \cup \mathring{\mathrm{A}}_{\mathcal{X}_+})$, then $\forall f \in \mathcal{F}$ will misclassify for some points in $\mathcal{X}_+$.*

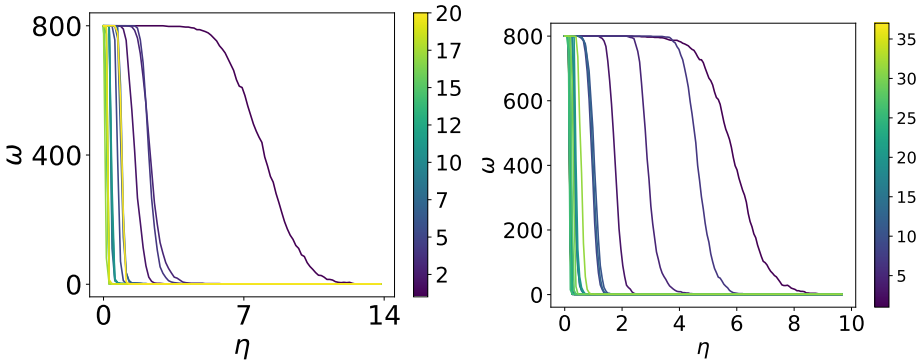
Proof. Since $H_s f(\mathring{A}_{\mathcal{X}_+}) \ncong H(\mathring{A}_{\mathcal{X}_+})$, $\exists V \subset \mathring{A}_{\mathcal{X}_+}$ such that f misclassifies every $v \in V$. Since, $\mathring{A}_{\mathcal{X}_+} \subset \mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+} \therefore V \subset \mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+}$. Now by the Theorem 3.1 of [29], we have $H_s f(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+}) \ncong H(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+})$. But by the claim 6.1, we have $H(\mathcal{X}_+) \cong H(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+})$ and $H_s f(\mathcal{X}_+) \ncong H(\mathcal{X}_+)$. Hence, a set $V' \subset \mathcal{X}_+$ exists such that f misclassifies every $v \in V'$. □

Thus, $H_s f(\mathring{A}_{\mathcal{X}_+}) \cong H(\mathring{A}_{\mathcal{X}_+})$is a necessary homological condition for an architecture to possess the required homological expressivity for the class. While the isomorphism $H_s f(\mathcal{X}_+) \cong H_s f(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+})$ is not guaranteed for an arbitrary classifier f, Lemma 6.2 establishes that this equivalence is a necessary property of any perfect classifier on the augmented space $\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+}$. Furthermore, since $\mathring{A}_{\mathcal{X}_+}$ consists of extrinsic points belonging to the class manifold $\mathcal{X}_+$, a classifier that correctly identifies the archetypal points demonstrates a broader generalisation capability. Because archetypes of the class represent the extreme points of the class, they are inherently more vulnerable to misclassification. The archetypes and data are typically disjoint sets ($D_+ \sqcup \mathring{A}_{D_+}$), rendering them sensitive representatives that act as a proxy for the homology of the data. Finally, a random sampling of the data will generally fail to satisfy this criterion

6.3 Experimental setup

The datasets utilised in this study comprise those from the Chapter 4(CIFAR10 [74], STL10 [49] and AIRCRAFT [75]) in addition to the CIFAR100 dataset. These datasets are similarly structured, consisting of balanced, well-defined, and open-source classes. The CIFAR100 [74] dataset consists of 100 balanced classes, with 500 training images and 100 test images per class. Each image is associated with both a sub-class and a super-class label, spanning 20 super-classes in total. The theoretical foundations for the neural network models utilised in this analysis are established in Chapter 2, while the specific structural designs and architectural configurations of the individual networks are detailed in Chapter 4.

Further in this chapter, we shift our focus from individual images to topological transitions within the point cloud. To analyse these transitions, each image and its corresponding cubical representations from each convolutional layer are mapped to a single instance in the point cloud. Specifically, for the $\ell-$th layer, the convolutional output of an image is converted to a single point in the point cloud by flattening the mean embedding vectors calculated across the spatial dimensions. The point cloud for the



(a) ResNet50 (b) ResNet101

Figure 6.5: **Homological curve:** (a) and (b) show that ω_η across the blocks of ResNets [23] of length 50 and 101, respectively. It can be observed that the lines (homological complexity) demonstrate a distributed monotonic horizontal compression with depth in the ResNet. The legends correspond to different blocks.

layer is then formed by the concatenation of these vectors across all images, which serves as the input for a Vietoris-Rips complex [5]. By generating layer-specific persistence diagrams, we can capture how the entire dataset evolves topologically as it passes through the network. For the computation of persistent homology, we utilise the Python-based libraries Ripser [89] and Persim [1]. These packages enable efficient computation of persistent homology for dimensions $k \leq 2$.

To evaluate the strength and direction of the associations between variables, we employ three statistical measures: Pearson correlation ($corr_r$), Spearman ($corr_s$) and Kendall ($corr_k$) (References [15, 78, 79]). Here, statistical significance is assessed again using the p-value under the null hypothesis that there is no relationship between the variables. The acceptance of the Null hypothesis (p-value > 0.05) implies that the results are more likely an instance of random occurrence. All experiments in this section are based on the subsample of the archetype $\mathring{A}_{np(C)}$ in the Chapter 5.

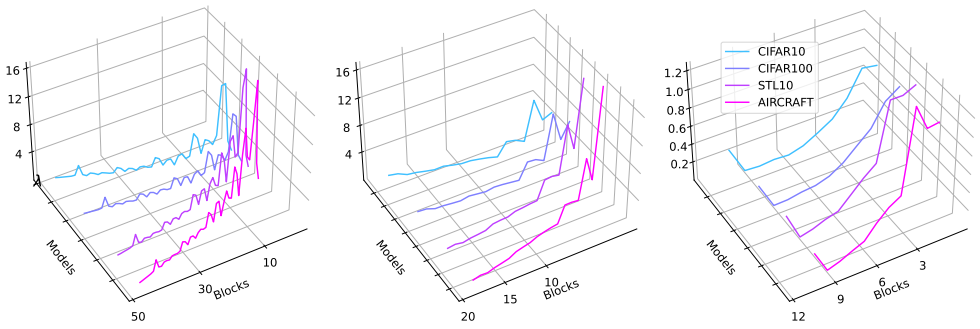
6.4 Average life for homological quantification

Theorem 6.3 justifies the utilisation of class archetypes to analyse the homological expressivity of a dataset. To observe topological transitions within a neural network via persistent homology, one may quantify and visualise the information from a persistence diagram through persistent vectorisation. Consistent with the methodology in Chapter 4, we employ the Betti curve and homological complexity across the filtration parameter η (see Chapter 3) to study the transformation of $X_{\mathring{A}_{np(C)}}$.

For example, the empirical observations of the Betti curves (see Figure 6.2) across threshold η of DenseNet121 exhibit a horizontal compression along the η-axis as the depth increases. This phenomenon indicates a topological transition within the embedding space, specifically a collapsing homology characterised by the diminishing persistence of features extracted from the persistence diagram at deeper layers. A similar observation can be made from the homological curves (ω_η) for ResNet50 and ResNet101 (see Figure 6.5).

The horizontal compression of the homological curves ω_η across successive layers is characterised by a gradual, monotonic decay in ResNet [23] architectures. Whereas DenseNet [24] models exhibit an abrupt initial shift followed by homological curves that remain nearly invariant in deeper layers. In ResNet, the additive residual connections facilitate a more distributed homological contraction of the manifold structure throughout the network depth. In the DenseNet architecture, feature concatenation increases the ambient dimensionality of the embedding space at each layer. Thus, the spatial global average pooling

[1] https://github.com/scikit-tda/persim



(a) VGG16 (b) MobileNetV2 (c) DenseNet121

Figure 6.6: Radial 3D plots of Λ for (a) VGG, (b) MobileNet, and (c) DenseNet121 across all blocks for different datasets. Best viewed in colour.

operation preserves the newly generated features alongside the accumulated features from all preceding layers. This additive aggregation suggests a stabilisation of the spatial mean within the embedding space, thereby resulting in the observed topological invariance and consistent homological curves.

To further investigate the topological transitions within the embedding space. We demonstrate that the topological average lifetime λ across the layers of a deep neural network effectively quantifies the evolving homology of the underlying topological space $\mathcal{X}_{D_c}$ associated with the point cloud for each class $c \in C$. By employing the average lifetime λ, we quantify the information contained in a persistence diagram to a singular measure of persistence for each layer, providing a topological summary that captures the global persistence. This quantification facilitates a rigorous, layer-wise analysis of the topological transition of the neural network. Furthermore, this scalarisation enables the application of Theorem 4.1 to estimate the homological capacity Φ_f. Particularily, the characteristic function h is approximated via a polynomial interpolation p of the average lifetime λ^ℓ across the network depth. This approach extends the polynomial approximation methodology utilised for individual images in Theorem 4.5 to the layer-wise evolution of the entire class manifold, which is more consistent with the definition of homological capacity Φ_f. From chapter 3, we have $\widehat{H}(D)$, which denotes the set of all such homological features in $dgm(D)$ that have finite lifetime.

Theorem 6.4. *Consider a deep neural classifier f with L layers and let λ_c^ℓ be the average life of the ℓ^{th} layer for the points of the class $c \in C$. We show that if the sequence $\{\lambda_c^\ell\}_{\ell=1}^L$, satisfies $\lim_{\ell \to L} \lambda_c^\ell = 0$, then the Betti numbers of the underlying topological space $\mathcal{X}_{D_c}$ satisfy:*

$$\lim_{\ell \to L} \beta_k(\mathcal{X}_{D_c}^\ell) = \begin{cases} 1 \text{ if } k = 0 \\ 0 \text{ if } k > 0 \end{cases}$$

Proof. For simplicity, here we take $\lambda_c^\ell = \lambda^\ell$. By definition

$$\lambda^\ell := \frac{\sum_i |d_i^\ell - b_i^\ell|}{n^\ell},$$

where $n^\ell = |\widehat{H}(f^\ell)|$ is the number of homological features that appeared in the dgm of the $\ell-$th embedding space with finite lifetime. For the sequence $\{\lambda^\ell\}_{\ell=1}^L$, as $\ell \to L$, we have $0 \leq \lim_{\ell \to L} \lambda^\ell = 0$. Since n^ℓ is bounded by the number of points in the point cloud, there exists an $M \in \mathbb{R}$ such that $n^\ell \leq M$. By

the squeeze theorem, it follows that

$$\begin{aligned}
0 \leq \lim_{\ell \to L} \frac{\sum_i |d_i^\ell - b_i^\ell|}{M} &\leq \lim_{\ell \to L} \frac{\sum_i |d_i^\ell - b_i^\ell|}{n^\ell} \\
\implies \lim_{\ell \to L} \frac{\sum_i |d_i^\ell - b_i^\ell|}{M} &= 0 \\
\implies \lim_{\ell \to L} \sum_i |d_i^\ell - b_i^\ell| &= 0 \text{ as } M \in \mathbb{R} \\
\implies \lim_{\ell \to L} |d_i^\ell - b_i^\ell| &\to 0 \text{ as } d_i^\ell - b_i^\ell \geq 0 \ \forall i \\
\implies b_i^\ell &\to d_i^\ell \ \forall i \text{ as } l \to L
\end{aligned}$$

In the context of persistent homology, the vanishing lifetime $|d_i^\ell - b_i^\ell| \to 0$ implies that the corresponding homological features fail to persist across the filtration as $\ell \to L$. Consequently, in the final layer L, the manifold is isomorphic to a single connected component. This leads to the convergence of the Betti numbers $\{\beta(\mathcal{X}_c)\} \to (1, 0, 0, \dots)$ as required. □

The preceding theorem demonstrates that a decreasing sequence of average lifetimes over the layers of the Neural Network characterises a collapsing homology. As the point cloud progresses through the layers, the Betti numbers vanish for all dimensions $k > 0$, such that the homology of the final embedding space $H(\mathcal{X}_{D_c}^L) \cong H(\mathbf{B}_0)$, the homology of a single connected component. In a neural classifier, this collapse reflects the transition of complex manifold structures into a contractible representation. If the sequence $\{\lambda^\ell\}_{\ell=1}^L$ is decreasing, it effectively quantifies this topological convergence, indicating that the network is progressively eliminating higher-order homological features to achieve a clustered or linearly separable state.

Definition 6.2. *Let us denote $\Lambda_D(f) \coloneqq \{\lambda_D^\ell\}_{\ell=1}^L$ as the sequence of λ_D^ℓ across the layers ℓ of the Deep Neural Network for the embedding space of the point cloud D.*

In particular here empirical compute the sequence till $L-1$, so $\Lambda_D(f) \coloneqq \{\lambda_D^\ell\}_{\ell=1}^{L-1}$. Before utilising $\Lambda_{\mathring{A}_{np(C)}}(f)$ to analyse the embedding space of the classifier, we establish the following relationship between the average lifetimes of the persistence diagrams for the dataset D and A_C ($dgm(D)$ and $dgm(A_C)$).

Theorem 6.5. *Let $\mathcal{D}$ be the dataset with feature space D and class index C. Let $\mathring{\mathrm{A}}_C^{(n)} = \bigsqcup_{c \in C} \mathring{\mathrm{A}}_c^{(n)}$ denote a sequence of union of class archetypes with increasing cardinality, that is $|\mathring{\mathrm{A}}_c^{(n)}| \geq |\mathring{\mathrm{A}}_c^{(n-1)}| \forall n \in \{1, 2, 3 \dots N\}, N \geq |D|$. Let $\lambda_{\mathring{\mathrm{A}}_C}^{(n)}$ be the average lifetime of the persistence diagram of the archetypes at step n, and λ_D be the average lifetime of the persistence diagram of D. Then, as the cardinality of the archetypes approaches the cardinality of the dataset ($n \to N$), the average lifetime of the archetypes converges to the average lifetime of the data:*

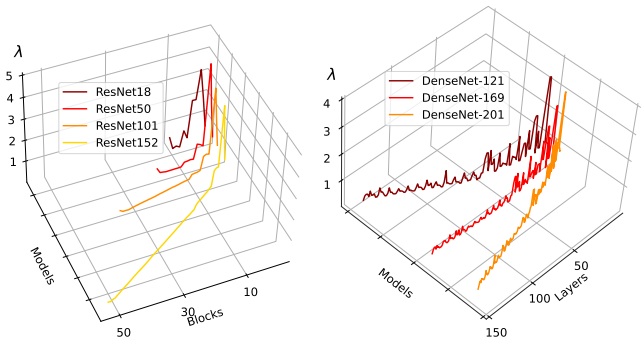
$$\lim_{n \to N} \lambda_{\mathring{\mathrm{A}}_C}^{(n)} = \lambda_D.$$

Proof. By the optimisation property of archetypes, as the cardinality of the archetypes n approaches that of the data, the archetypes converge to the data ($N \geq |D|, n \to N \implies \mathring{A} \to D$). For each class $c \in C$, as $n \to N$, we have $|\mathring{A}_c^{(n)}| \to |D_c| \implies \mathring{A}_c \to D_c$. Since $|\mathring{A}_C^N| \geq |D|$, for each $c \in C$, $|\mathring{A}_c^{(n)}| \geq |D_c|$ as $\sum_{c \in C} |D_c| = |D|$. As $n \to N \implies \mathring{A}_C^{(n)} \to D$, for the sequence, we have $\lim_{n \to N} \lambda_{\mathring{A}_C}^{(n)} = \lambda_D$. □

Note that the cardinality notation $|\cdot|$ refers to the count of unique distinct elements in the point cloud. Furthermore, since the simplicial complex and its associated persistence diagram are determined by the geometric support of the data, duplicate points or replicates in D are treated as a single vertex

during filtration. Thus, once $\mathring{A}$ saturates the unique coordinates of D, their topological summary λ becomes identical. Now we will look at the case when EPA property is preserved by the archetypes. While we have established that $|\widehat{H}(\mathring{A})|\lambda_{\mathring{A}} \geq \lambda_D$ under the EPA property (Theorem 5.5), and the convergence of homological complexity (ω) of the archetype to that of D as $n \to |D|$ (Lemma 5.2), based on the optimisation property of archetypes in the Chapter 4, the above theorem extends to the sequence of average lifetimes. Specifically, it demonstrates that as the cardinality of the union of class archetypes ($\mathring{A}_C$) increases, the average lifetime converges to that of D. However, this convergence of average lifetimes is not, by itself, sufficient to demonstrate the convergence of Betti numbers for the dataset through the archetypal space.

This limitation arises because the feature maps produced by successive layers of the classifier often result in nonlinearity or dimensionality reduction, which does not necessarily preserve the EPA property of the archetypes. While archetypal analysis identifies extremal points in the input space, these points do not necessarily map to the boundary of the convex hull in deeper embedding space. At the same time, such a violation of the EPA would necessarily imply a shift in the homology. Therefore, the preservation of homology under the augmentation of archetypes (Theorem 5.6) implies that it can be used as a measure of evaluating the homological expressiveness of the architecture required to resolve the homology of the underlying manifold. Furthermore, Theorem 6.4&6.3 justify using the average lifetime of the union of class archetypes as a measure of changing Betti numbers in the Neural Network classifiers. Consequently, these results suggest that the outcome is independent of the preservation of the EPA property.



(a) ResNet (b) DenseNet

Figure 6.7: Radial 3D plots of Λ for (a) ResNet family of models across all blocks, and (b) DenseNet family of models across all layers in the initial three blocks on the STL10 dataset. Best viewed in colour.

6.4.1 Empirical Analysis of Average Lifetime

To evaluate the theoretical framework established in the preceding sections, we examine the behaviour of the sequence of average lifetime Λ across the layers/blocks of the Neural Network for various architectures and datasets. For simplicity we denote $\Lambda_D(f)$ as Λ. For the visualisation of Λ, we utilise the 3D radial plots where the x-axis represents the models, the y-axis represents the architectural blocks, and the z-axis represents the corresponding Λ Λ values (see Figures 6.7&6.6).3D plots are used with x-axis the Models, y-axis the Blocks and z-axis the Λ values (see Figures 6.7&6.6). The 3D radial plots provide an aerial view of an intuitive comparison of the topological transitions with depth.

Remark 6.6 (Homological Capacity)**.** *To contextualise the following empirical analysis, we recall the definition of* ***homological capacity*** *(Φ_f) from the Chapter 4. For a neural classifier f of depth L acting on a dataset $\mathcal{D}$, the capacity is defined as $\Phi_f := \dfrac{\omega(\mathcal{X}_{\hat{c}}^L) - \phi}{L}$ where $\hat{c}$ denotes the index of the class exhibiting the maximal homological complexity denoted as $\phi = \omega(\mathcal{X}_{\hat{c}})$ and $\omega(\mathcal{X}_{\hat{c}}^L)$ represents its homological complexity at the $L-$th layer.*

The following findings are derived from the empirical observations of Λ:

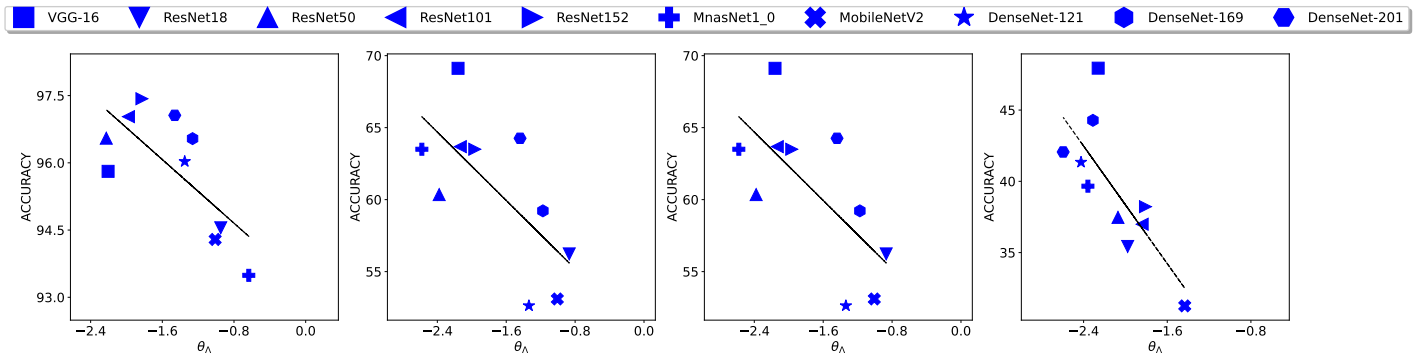
Architectural Depth Transition : For models of varying depths, ResNet architectures exhibit significant topological transitions within the initial layers. In contrast, DenseNet models demonstrate a more gradual, negligible change in Λ across the layers. These observations align with the Betti and homological curves discussed in the previous section. In ResNet, additive residual connections facilitate a distributed homological contraction of the manifold throughout the network depth, resulting in a gradual, monotonic decay. On the other hand, DenseNet exhibits an abrupt initial shift followed by invariance in deeper layers. This is attributed to feature concatenation and global average pooling, which preserve newly generated features alongside accumulated ones. This additive aggregation stabilises the spatial mean within the embedding space, resulting in the observed consistency in deeper layers. These results further strengthen the theoretical framework established in Theorem 6.4.

Dataset Sensitivity : Across different datasets, the Λ values for the initial blocks exhibit a vertical shift that eventually diminishes with depth. These variations in Λ are directly related to the initial homological complexity ($\omega(\mathcal{X})$) of the dataset. Specifically, the initial complexity captures the variance in ω across the different datasets. Consequently, the information from the initial layers allows for the estimation of *phi*.

Model Characteristics : The rate and manner in which Λ evolves differ significantly across architectures, implying that the decay trend of Λ is a distinctive model characteristic. In particular, the empirical results show that $\max \Lambda(DenseNet) \leq \max \Lambda(ResNet) \leq \max \Lambda(VGG) \leq \max \Lambda(MobileNet)$ for all datasets in the observed blocks. This reducing trend reflects the homological transition in the Neural Network as established in Theorem 6.4. Thus, the values at the terminal layers can be used to extrapolate for $\omega(\mathcal{X}_{\hat{c}}^{L})$.

Thus, the scalarisation of persistence diagrams into the average lifetime provides a computable framework, utilising archetypes, for quantifying the homological evolution of the feature space. Building upon these findings, the following section provides a comprehensive discussion of the broader implications. We empirically analyse how these topological transitions relate to model generalisation, the trade-off between architectural complexity and manifold resolution, and the potential for average lifetime to serve as a diagnostic tool for the deep neural classifier.

By approximating the characteristic function h via a polynomial interpolation p of the average lifetime. We assume that the initial and terminal layers reflect the difference in homological complexity ω across the neural network through average life. Thus, intuitively, the slope would reflect information about the homological transition, and we implicitly extrapolate ϕ from the information in the initial layers. These results are consistent with the observation in Chapter 4, thus solidifying the role of homological complexity in the expressiveness of the Neural Network.



(a) STL10 (b) CIFAR10 (c) CIFAR100 (d) AIRCRAFT

Figure 6.8: Accuracy vs θ_Λ score for different datasets.

6.5 Discussion of Transfer Learning score

As established previously in Theorem 4.1, the characteristic function h required to approximate the homological capacity Φ_f must be continuous on the interval $[0, L]$ and differentiable on $(0, L)$. Consistent with the methodology in Theorem 4.5 (Chapter 4), we approximate this function by fitting a polynomial p over the sequence $\Lambda_D(f) \coloneqq \{\lambda_D^\ell\}_{\ell=1}^{L-1}$. We assume that this polynomial extrapolates to the initial homological complexity $\omega(\mathcal{X})$ at $\ell = 0$ and the final complexity $\omega(\mathcal{X}_{\hat{c}}^L)$ at $\ell = L$. Then the instantaneous slope of p at some $\hat{o} \in (0, L)$ then serves as an approximation for Φ_f.

Remark 6.7. *Following the convention from the previous Chapter 4, we denote the approximated homological capacity and its geometric orientation as $\Phi_f \approx \theta$ and $\tan^{-1}(\Phi_f) = \widetilde{\Theta}$, respectively. We utilise the specific subscripts Ω and Λ to distinguish between the variants derived from homological complexity and average lifetime.*

In this analysis, the scores are computed using the average lifetime λ within the embedding space. Thus, we denote them as θ_Λ and $\widetilde{\Theta}_\Lambda$ (see Figure 6.8 and Figure 6.8). A primary merit of θ_Λ over θ_Ω is that θ_Λ is computed over the archetype feature space $\mathring{A}_c$, which serves as a robust proxy for the dataset D_c. Furthermore, because $\Lambda_{\mathring{A}_C}(f)$ exhibits minimal variability across each class $c \in C$ we let $\theta_\Lambda(A_C) \approx \phi$ (see Figure 6.10). The following points detail the empirical correlation of these scores with classification accuracy:

Correlation Analysis : Both θ_Λ and $\widetilde{\Theta}_\Lambda$ demonstrate negative correlation with the accuracy (see Table 6.1). Pearson, Spearman, and Kendall correlations [15, 78, 79] indicate a moderate relationship in most cases. The correlation of θ_Λ and $\widetilde{\Theta}_\Lambda$ are nearly identical; they serve as valid replacements for one another. Notably, $\widetilde{\Theta}_\Lambda$ offers better interpretability due to its angular representation, whereas θ_Λ provides higher numerical precision for comparative analysis.

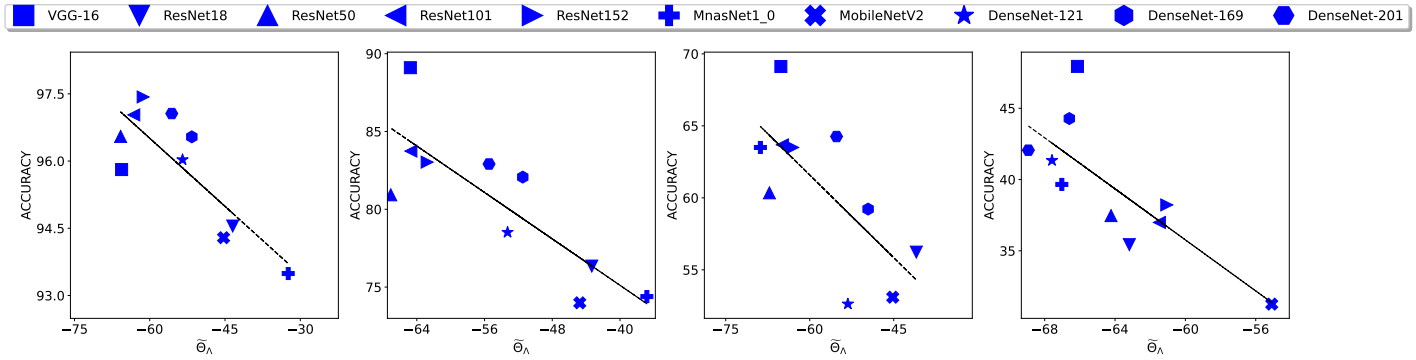
Polynomial Fitting : The θ and Θ cores were computed using polynomials of degree 1 and 2. These polynomials were interpolated and then normalised to the interval $[0, 1]$ relative to the network depth. This normalisation ensures flexibility across different models with varying depths L and facilitates the extrapolation of complexity at the boundary points 0 and L. The resulting score values were evaluated at the midpoint $\hat{o} = 0.5 \in (0, 1)$. It should be noted that increasing the degree of the interpolated polynomial p may lead to higher extrapolation error when deriving the slope. Nevertheless, a degree 1 polynomial interpolation maybe insufficient for all models because the sequence of Λ across the embedding spaces does not always exhibit a strictly monotonic descent.

Theoretical Constraints : Following Theorem 4.1 as established in Chapter 4, the more negative the score θ_Λ higher homological capacity Φ_f. Within this setting, positive values fail to provide a meaningful comparison regarding the reduction of homological complexity. Since $\hat{o}$ is not known, in such instances, a lower degree polynomial or a modified evaluation point $\hat{o}$ should be utilised to compute θ_Λ.

These results are consistent with the observations in Chapter 4, thus solidifying the role of homological complexity in characterising the expressivity of the Neural Network. Both theoretically and empirically, we have demonstrated that the average lifetime λ serves as a robust metric to study the evolving homological complexity within these architectures.

6.6 Conclusion

Topological expressivity is a well-established and widely accepted measure for quantifying the representational power of neural networks. In the paper [3], homological complexity is examined through the lens of network depth-width and activation functions. Theoretically establishing that deeper architectures achieve higher homological expressivity given equivalent resources. Further, the work [29] establishes a relationship between the homology of a dataset and that of its corresponding binary classifier, via the



(a) STL10 (b) CIFAR10 (c) CIFAR100 (d) AIRCRAFT

Figure 6.9: Accuracy vs $\widetilde{\Theta_\Lambda}$ score for different datasets

Table 6.1: Pearson $corr_r$ and Spearman $corr_s$ and Kendall $corr_k$ correlation [15, 78, 79] for the TL scores θ and $\widetilde{\Theta}$ with their p-values.

Dataset	$corr_r(\theta_\Lambda)$	p-value $corr_r(\theta_\Lambda)$	$corr_r\widetilde{\Theta_\Lambda}$	p-value $corr_r\widetilde{\Theta_\Lambda}$	$corr_s(\theta_\Lambda)$	p-value $corr_s(\theta_\Lambda)$	$corr_s\widetilde{\Theta_\Lambda}$	p-value $corr_s\widetilde{\Theta_\Lambda}$	$corr_k(\theta_\Lambda)$	p-value $corr_k(\theta_\Lambda)$	$corr_k\widetilde{\Theta_\Lambda}$	p-value $corr_k\widetilde{\Theta_\Lambda}$
CIFAR10	-0.78	0.007	-0.825	0.003	-0.75	0.01	-0.75	0.01	-0.64	0.009	-0.64	0.009
CIFAR100	-0.683	0.029	-0.712	0.02	-0.613	0.05	-0.613	0.05	-0.40	0.105	-0.40	.105
STL10	-0.73	0.015	-0.828	0.0030	-0.64	0.04	-0.64	0.04	-0.46	0.07	-0.46	0.07
AIRCRAFT	-0.76	0.01	-0.78	0.007	-0.72	0.01	0.72	0.01	-0.5	0.04	-0.5	0.04

support homology and derives a model capacity measure as a fraction of the Betti number of the f and dataset, emphasising the role of Betti numbers in model selection diagnostic scores. In this work, the Homological Generalisation Theorem [29] was extended to the class archetypal space $\mathring{A}_c$, and additionally, the homological complexity was approximated through the computable average lifetime λ. By utilising a polynomial extrapolation method (Chapter 4) based on the average lifetime, we developed a more robust estimator for the model selection diagnostic score Φ_f.

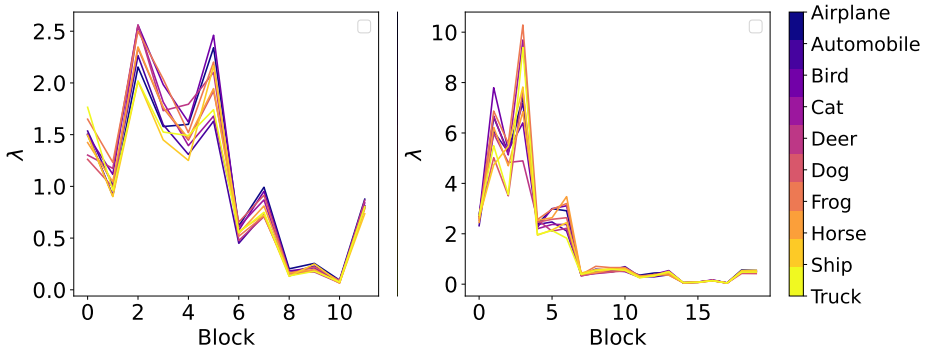
Despite these advancements, several shortcomings remain to be addressed in future research:

Archetypal Generalization : Although Theorem 6.3 justifies using the archetype set $\mathring{A}_C$ and its average lifetime $\Lambda_{\mathring{A}_c}(f)$ to replicate the trend of reducing Betti numbers, the precise functional relationship between $\Lambda_{\mathring{A}_C}(f)$ and $\Lambda_{D_c}(f)$ remains mathematically undefined.

Block-wise Resolution : While computational efficiency is significantly enhanced by evaluating homological features at the block level rather than per layer, this discretisation across the blocks results in a loss of subtle homological information that occurs within individual layers of the architecture.

Correlation : The TL score demonstrates only a moderate association with the fine-tuned classification accuracy. This may stem from several factors, including λ acting as a relatively weak estimator of Φ_f or limitations in the sample size of the point clouds used for persistent homology calculations. Furthermore, since the computed accuracy is a point estimate of performance, it may fail to fully capture the generalised representational power of the architecture that the score aims to quantify. Consequently, future studies establishing a formal relationship between homological capacity Φ_f and classical measures such as VC dimension [31] or Rademacher complexity [33] would provide a more comprehensive understanding of Φ_f and its relation to model performance.

In summary, the average lifetime Λ provides a robust empirical measure for quantifying the topological transition of neural architectures. Building upon these homological insights, the following chapter shifts



(a) ResNet18 (b) MobileNetV2

Figure 6.10: **Different class:** The evolving sequence $\Lambda = \{\lambda\}$ across the architectural blocks for (a) ResNet (b) MobileNetV2, where each line plot corresponds to the class archetypal subspace $\mathring{A}_c$ of the CIFAR10 dataset. In this case, the marginal variation across the classes for the value of λ at each block enables the utilisation of the union of archetypes $\mathring{A}_C$ to approximate Φ_f.

the focus toward assessing the geometric structure of the decision boundary using Archetypal subspace during training.

Chapter 7

Assessing the Topological Transitions in the Training Phase

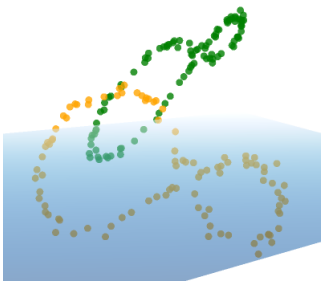


Figure 7.1: **Linear Decision Boundary :** The linear decision boundary is approximated within the archetypal subspace using a single-layer neural network with an identity activation function. Although the model achieves an accuracy exceeding 85%, it fails to fully separate the classes due to the inherent nonlinearity of the true decision boundary. This relatively high performance is attributed to the spatial distribution of the data. The identity activation restricts the model to a linear hyperplane, which is theoretically insufficient for resolving such entangled topological structures (See Figures 3.1(b) in Chapter 3 between the classes.

Building upon the previous chapter, which established the archetypal subspace as a robust representative approximation for binary classification, this chapter analyses the consequences of preserving homological relations through archetypes. Furthermore, we explore the decision boundaries of three-dimensional point clouds using Kernel Archetypal Analysis (KAA) [87] and study the topological evolution of embedding spaces during the training process. This chapter is divided into two primary sections:

Archetypal subspace : This section further explores the characteristics of the archetypal subspace. Theoretically, we demonstrate that maintaining the relationship $H_s f(\mathcal{X}_+) \cong H_s f(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+})$ is significantly more critical than achieving $H(\mathcal{X}_+) \cong H(\mathring{A}_{\mathcal{X}_+})$. We seek to verify the claims established in Chapter 6 by visualising the decision boundary, utilising predictions derived from the archetypal representation to map the structural transitions of the manifold (see Figure 7.1). To achieve this, we employ Kernel Archetypal Analysis (KAA) [87], which is particularly good at approximating nonlinear boundaries in the data. Our empirical results demonstrate that KAA effectively captures the underlying manifold structure of the decision boundary.

Training : This section employs persistent landscapes to analyse the layer-wise topological transitions of the embedding spaces during training. Furthermore, we examine how these transitions vary across different activation functions [40] and architectural depths using a point cloud of known homology and a multilayered Neural Network architecture.

These results demonstrate the possibility that models achieve a topological transition state during the training phase, which is reflected in the similarity of the topological transition across different datasets for the same pre-trained architecture in the preceding chapters. This chapter focuses primarily on empirical findings, specifically observing the evolution of topological structures throughout the training phase, for the point cloud in Figure 7.1 (classes denoted by green and orange respectively) in a binary setting.

Finally, with respect to the homological capacity, we analyse the theoretical upper bound on the depth for Φ_f to avoid biasing towards a smaller depth. This bound serves as a necessary constraint to ensure the defined homological capacity metric does not exhibit an undue bias toward smaller network architectures, providing a more balanced criterion for architectural selection.

7.1 Homological Consistency between Data and Archetypes

In Chapter 6, the Homological Generalisation Theorem was extended to the Archetypal space 6.3. It was demonstrated that the inability of the classifier to satisfy the support homology of the archetypal subspace provides a sufficient condition to reject a network architecture based on insufficient homological expressivity. This theoretical foundation raises two critical questions: (1) what homology does the archetypal subspace possess in relation to the dataset, and (2) how does the isomorphism $H(\mathcal{X}_+) \cong H(\mathring{A}_+)$ influence the generalisation theorem? The following discussion focuses primarily on the latter.

Corollary 7.1 (Homological generalisation extended to isomorphism of Archetypes)**.** *Let $\mathcal{X} = (\mathcal{X}_+ \cup \mathring{\mathrm{A}}_+) \cup (\mathcal{X}_- \cup \mathring{\mathrm{A}}_-)$ be the topological space as defined in Chapter 6 with $H(\mathcal{X}_+) \cong H(\mathring{\mathrm{A}}_{\mathcal{X}_+})$. Let $\mathcal{F} \subset 2^X$ such that $f \in \mathcal{F}$ with $H_s f(\mathring{\mathrm{A}}_{\mathcal{X}_+}) \ncong H(\mathring{\mathrm{A}}_{\mathcal{X}_+})$ and $H_s f(\mathcal{X}_+) \cong H_s f(\mathcal{X}_+ \cup \mathring{\mathrm{A}}_{\mathcal{X}_+})$, then $\forall f \in \mathcal{F}$ will misclassify for some points in $\mathcal{X}_+$.*

Proof. The proof follows the same construction as Theorem 6.3. □

This result demonstrates that the condition, $H_s f(\mathcal{X}_+) \cong H_s f(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+})$, is the essential requirement for the generalisation of the theorem (see Figure 7.2(a)). The isomorphism $H(\mathcal{X}_+) \cong H(\mathring{A}_+)$ does not, in itself, relax or improve the underlying constraints of the Homological Generalisation Theorem.

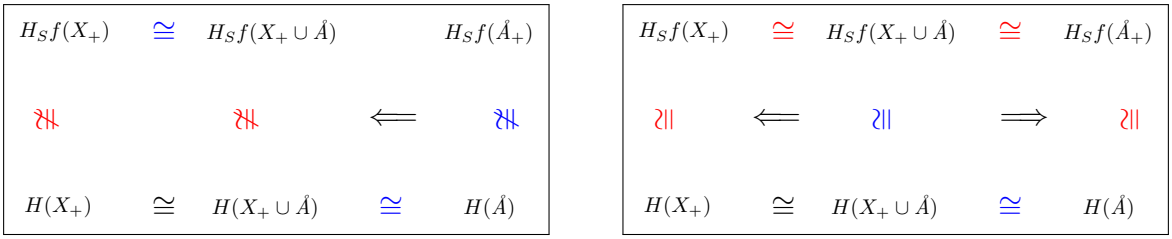
Remark 7.1. *Alternatively, if one implies $H_s f(X+) \cong H_s f(\mathring{\mathrm{A}}_{\mathcal{X}_+})$, and f correctly classifies every point in $\mathring{\mathrm{A}}_+$ and $H(X_+) \cong H(\mathring{\mathrm{A}}_{\mathcal{X}_+})$ then it follows that $H_s f(X_+) \cong H(X_+)$.*

The above remark suggests that f satisfies the required homological equivalence, but this does not strictly guarantee that the model will correctly classify for X_+. Structurally, it is significantly more feasible to assume $H_s f(\mathcal{X}_+) \cong H_s f(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+})$ than to assume the stronger $H_s f(\mathcal{X}_+) \cong H_s f(\mathring{A}_{\mathcal{X}_+})$. However, a compelling relationship emerges when we assume both ground truth isomorphism and perfect classification on the union (see Figure 7.2(b)).

Corollary 7.2. *Let $f \in \mathcal{F}$, if f perfectly classifies every point in the classes and their respective archetypes and $H(\mathcal{X}_+) \cong H(\mathring{\mathrm{A}}_{\mathcal{X}_+})$ then $H_s f(\mathring{\mathrm{A}}_{\mathcal{X}_+}) \cong H_s f(\mathcal{X}_+)$.*

Proof. Since f perfectly classifies the dataset and its archetypes, it holds for the positive class that $H_s f(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+}) \cong H(\mathcal{X}_+ \cup \mathring{A}_{\mathcal{X}_+})$, $H_s f(\mathcal{X}_+) \cong H(\mathcal{X}_+)$ and $H_s f(\mathring{A}_{\mathcal{X}_+}) \cong H(\mathring{A}_{\mathcal{X}_+})$. Given the assumption $H(\mathcal{X}_+) \cong H(\mathring{A}_{\mathcal{X}_+})$ it follows by transitivity that $H_s f(\mathcal{X}_+) \cong H_s f(\mathring{A}_{\mathcal{X}_+})$. □

Corollary 7.2 demonstrates that a homological isomorphism arises between the support homologies when the structural condition $H(\mathcal{X}_+) \cong H(\mathring{A}_{\mathcal{X}_+})$ is satisfied. It should be noted that while the preceding derivations focus on the positive class (+), identical relations can be established for the negative class (−). While further properties regarding the homological relationship between $H(\mathcal{X}_+)$ and $H(\mathring{A}_{\mathcal{X}_+})$ may be derived using the direct sum $\oplus$ property (see Book [4]), such an exploration lies beyond the scope of this work.



(a) corollary 7.1 (b) corollary 7.2

Figure 7.2: **Homology of archetypal subspace**: The figures demonstrate the corollary based on the Homological generalisation extended to $H(\mathcal{X}_+) \cong H(\mathring{A}_{\mathcal{X}_+})$.

Nevertheless, there are instances where a topological isomorphism exists between the archetypes and the data that can be utilised to study the generalisability of the neural network. Towards the same, in the following section, we consider the class point clouds that are solely represented by extreme points of the convex hull. In such a case, archetypes can be used to describe the homology of the data exactly (see Figure 7.3). Thus, the decision boundary of the neural network can also be interpreted in terms of the archetypes of this particular toy point cloud.

7.2 Experimental Setup

This chapter utilises the 3-dimensional point cloud data illustrated in Figure 7.1. Each class (represented in green and orange) is specifically designed to possess a homology characterised by the Betti number $\beta = (1, 2, 0)$, corresponding to a figure-eight realised in $\mathbb{R}^3$. The full point cloud (D) consists of 200 points distributed across two balanced classes (Y). Given this balanced distribution, the baseline expected accuracy is 50%. Model training is conducted using a subset of 80% points, while the archetypes are employed specifically to map and generate the decision boundaries.

We evaluate the performance and topological transitions of the network across the following activation functions (f_a) :

Sigmoid : Defined by the formula $f_a(x) := \frac{1}{1+e^{-x}}$.Its outputs are interpretable as probabilities, and the function is smooth and continuously differentiable across its entire domain.

Tanh : The hyperbolic tangent $f_a(x) := \frac{e^x - e^{-x}}{e^x + e^{-x}}$ squashes input values to the range $[-1, 1]$. In addition to its smoothness and differentiability, the function is zero-centred, which aids in optimising training dynamics.

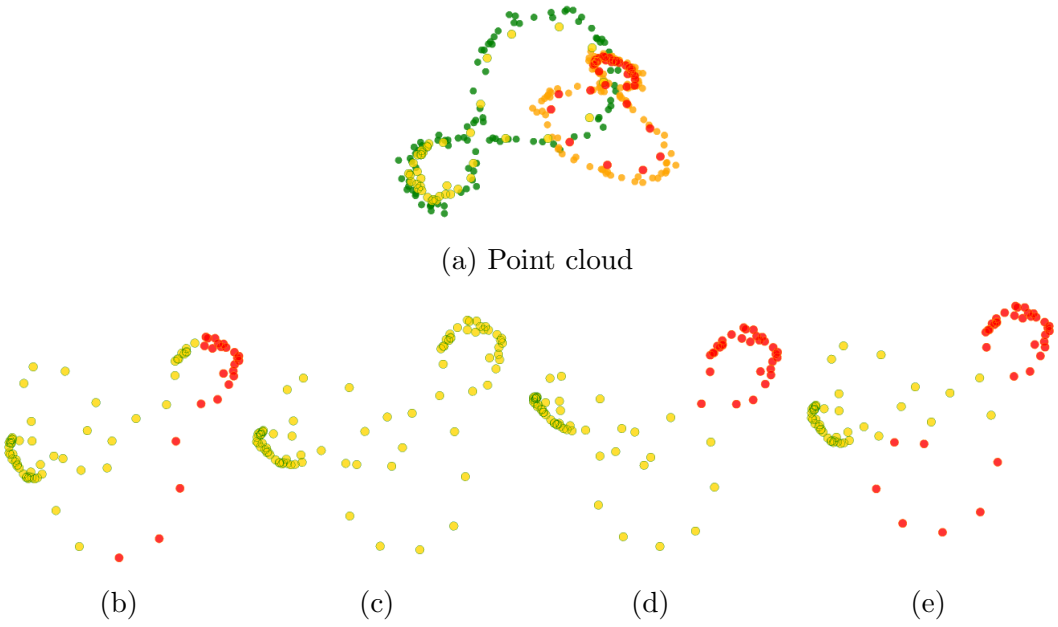
ReLU : The Rectified Linear Unit is defined as $f_a(x) := \max(x, 0)$. While non-differentiable at 0, it effectively mitigates vanishing gradients in the positive domain and promotes network sparsity.

LeakyReLU : A modification of the ReLU function designed to account for the vanishing gradients in the negative region. This makes them more stable and consistent in training. It is piecewise defined as $f_a(x) := x \forall x \geq 0$ and $f_a(x) := ax \forall x < 0$.

SiLU : The Sigmoid Linear Unit $f_a(x) := \frac{x}{1+e^{-x}}$ is a smooth, non-monotonic function that provides implicit regularisation through its global minimum.

The experiments were conducted over multiple trials, and the results presented here are representative of those observations. The models were designed to ensure a nearly fixed parameter count but varying widths and depths, which represents the problem in Figure 1.1 in Chapter 1. The results presented here specifically reflect configurations with approximately 90 parameters. Two distinct architectures were evaluated: a **Wide Network** (comprising an input layer, an activation layer, and an output layer) and a **Deep Network** (comprising an input layer followed by a sequence of alternating hidden layers and

activation functions, terminating in an output layer). All models were trained using the Adam optimiser with a learning rate of 0.1 and binary cross-entropy loss. Weights were initialised according to the Kaiming uniform distribution.



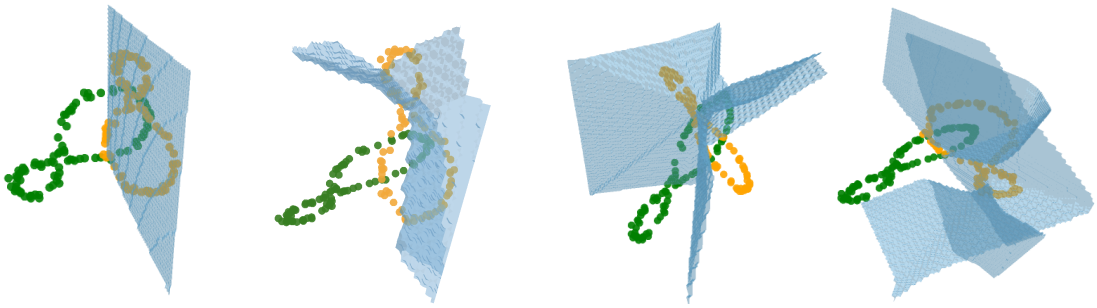
(a) Point cloud

(b) (c) (d) (e)

Figure 7.3: **Linear vs Nonlinear Boundary**: (a) The class archetypes $\mathring{A}_C$ of the 3D point cloud comprising two classes, each sampled from a figure-eight realised in $\mathbb{R}^3$. The homological complexity is defined by $\omega(\mathcal{X}_c) = 3$, corresponding to the Betti numbers $\beta(\mathcal{X}_c) = (1, 2, 0)$. Subfigures (b) through (e) illustrate the predicted class assignments for the archetypal subspace across linear (b) and nonlinear (c, d, e) neural network activation functions.

7.3 Archetypal with Kernel Spaces

In the Figure 7.3, there are two classes: green and orange; The red and yellow represent Archetypes ($\mathring{A}$) identified via Kernel Archetypal Analysis [87]. In traditional Archetypal Analysis [80], the optimisation involves solving nonlinear least-squares problems, which are often prone to local minima; this can result in archetypal representations that underrepresent the underlying data. As noted by Cutler and Breiman [80], these archetype estimates are constrained within the convex hull of the data points, an approach that remains appropriate only under strict separability assumptions. This limitation is addressed by Kernel Archetypal Analysis (KAA) [87], which ensures separability by mapping the data into a Reproducing Kernel Hilbert Space (RKHS) [14, 11]. The necessity of the Reproducibility of Hilbert space is to facilitate the continuous linear functional between the kernel space and the inner product space. In this section, we compute kernel archetypes via the Radial Basis Function (RBF) kernel [11], a standard choice for inducing nonlinear feature mappings. This approach allows the model to capture complex, nonlinear dependencies in the data that are not accessible in the original input space (See Figure 7.3). Empirical findings suggest that these kernel archetypes provide a selective, reduced representation of the data geometry. Since in the previous Chapter 6 we proved that the archetypes can characterise the homological support functions of the data needed to reject architecture, here we visualise the decision boundary using their predictions from various neural network architectures (see Figure 7.3). Our empirical findings confirm that kernel archetypes provide a suitable representation of the data geometry under archetypal isomorphism at the topological level. The following observations are derived from a neural network with a single hidden layer across various activation functions:



(a) Sigmoid (b) ReLU (c) Leaky-ReLu (d) TanH

Figure 7.4: **Different Activation Function**:Comparison of decision boundaries generated using different activation functions. Consistent with its application in binary classification, the Sigmoid function produces a linear boundary (a). Note that the observed nonlinear boundaries are representative examples and are not exclusively unique to each activation function.

Predictions : In the Figure 7.3, the red and yellow points represent archetypes as predictions by the model. None of the models achieved a perfect accuracy, as clearly illustrated in Figure 7.3(b-e). Notably, a nonlinear model need not outperform a linear one, even when the data require a nonlinear decision boundary. This suggests that achieving the appropriate nonlinear complexity for a decision boundary is a delicate balancing act for any classification model. Nevertheless, a linear boundary remains fundamentally incapable of achieving full accuracy for such non-linearly separable data.

Decision boundary : Figure 7.3 illustrates the decision boundaries approximated for different activation functions. While these boundaries are nonlinear, they often fail to capture the manifold structure of the data. This discrepancy is attributed to the linked structure between the classes. Although the classes are strictly separated, they exhibit a topological link (specifically a Hopf link) in addition to each class possessing a homological complexity $\omega = 3$ (See Figure 1.3& 3.1 from the previous chapter).

As previously observed, a high classification accuracy ($> 85\%$) can be misleading. In particular, we observe that the spatial positioning of the classes influenced accuracy rather than the successful resolution of the underlying data structure, particularly when relatively few points define the topological link. This observation gives us further insight into the questions posed in Figure 2.7 in Chapter 2: Which architecture can approximate these hyperplanes with the fewest parameters, and what specific data structures affect the decision boundary?"

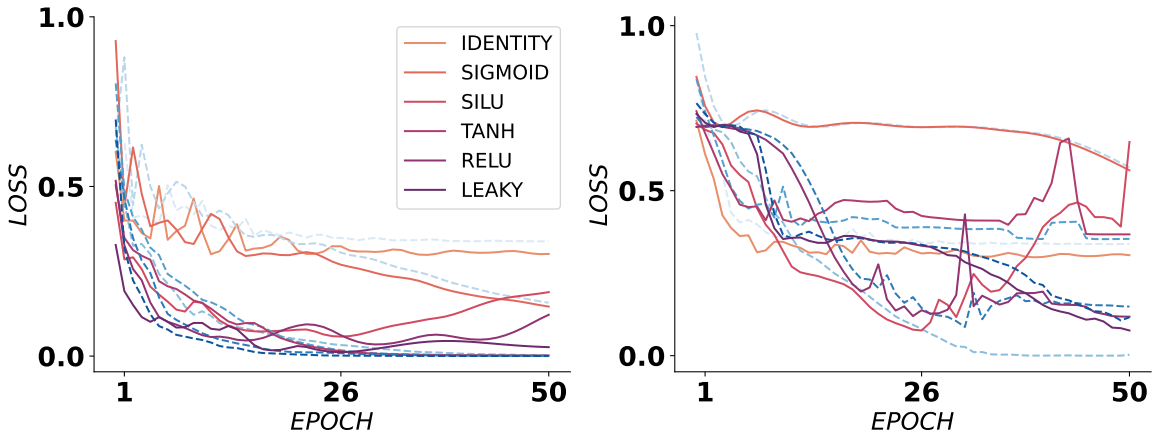
The topological structure inherent in the data plays a foundational role in defining the necessary model capacity. Further, these results reinforce the idea that topological structures, extending beyond simple homology, within the decision boundary, significantly affect the architectural complexity required to solve a classification problem. How these topological links influence the width-depth expressivity of a neural network remains a compelling area for future research. To better quantify the homological transitions across layers, we will use the "persistent landscape" to capture layer-layer changes.

7.4 Topological transition via training

To characterise the structural evolution during the training phase, the Persistence Landscape distance (PL) was computed across the layers of the neural network.

Definition (Persistence landscape [5]). *Given a finite persistence Diagram* $dgm(D) = \{(b_i, d_i)\}$, *the persistence landscape is a function* $\alpha_D : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$ *where*

$$\alpha_D(k,t) := k - th \text{ largest value of the } [\min\{t - b_i, d_i - t\}]_+ \forall i$$



(a) Wide (b) Deep

Figure 7.5: **Loss Functions**: Training and validation loss for the (a) Wide Network and (b) Deep Network, respectively. For the archetypal subspace, consisting of 80 balanced samples, the classification accuracy reaches (a)$\geq$ 85 and (b) $\geq$ 80, respectively. In the plots, red lines indicate the validation loss, while blue lines indicate the training loss.

The p-norm on the persistence landscape is defined as $||\alpha_D||_p^p = \sum_{k=1}^{\infty} ||\alpha_D(k,\cdot)||_p^p$. *Here in particular the p-landscape distance for* $p = 2$ *between two persistence diagram* $dgm(D_1)$ *and* $dgm(D_2)$ *is given by* $PL_2(D_1, D_2) = ||\alpha_{D_1} - \alpha_{D_2}||_2$ *for simplicity denoted as* $PL(\cdot,\cdot)$.

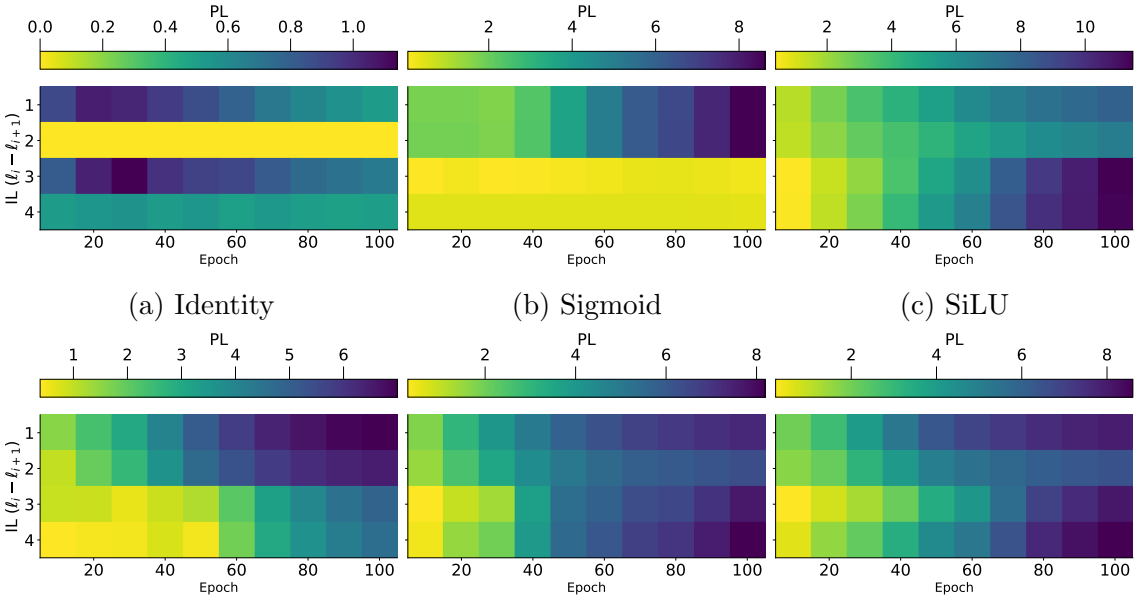
We define $\ell = 0$ as the feature space (D) and $\ell = L$ as the response space (Y). In this analysis, we examine the latent embedding space, with the activation function treated as a distinct layer, to observe its specific impact on the topological flow. For the wide network, $\ell = 1$ denotes the embedding from the input layer, $\ell = 2$ denotes the embedding after the activation function, and $\ell = 3$ represents the final output representation.

Notation. *The topological transition between any two adjacent layers* $IL(\ell_i, \ell_{i+1})$ *is quantified by* ***Inter-Layer Topological Transition*** $PL(\ell_i, \ell_{i+1})$. *This metric represents the distance between the persistence landscapes of the respective layer embeddings. In the corresponding figures (example Figures 7.6), the index* $i + 1$ *is used to denote* $IL(\ell_i, \ell_{i+1})$, *which represents between layers* ℓ_i *and* ℓ_{i+1}.

The Figures 7.6& 7.7 represent the Inter-Layer Topological Transition in terms of persistence landscape distance given by $PL(\ell_i, \ell_{i+1})$ across the different classification architectures. In both figures, $IL(\ell_i, \ell_{i+1})$ indexed as $i + 1$ for $i = \{0 \ldots L - 1\}$, denotes the successive layer of which the persistence landscape distance was computed. For the Neural Network, this encompasses transitions starting from the feature space through to the response space. Specifically, for the wide network, these transitions in terms PL are represented as $PL(\ell_0, \ell_1)$, $PL(\ell_1, \ell_2)$, $PL(\ell_2, \ell_3)$ and $PL(\ell_3, \ell_4)$, where $\ell = 0$ and $\ell = L$ represents the feature space D and the response space Y (one hot encoded) respectively (visualised as heat-maps in Figure 7.6, here IL takes the values $1, 2, 3$ and 4.).

Activation functions : At a fixed depth, transitions are highly dependent on the chosen activation function, suggesting that $PL(\ell_i, \ell_{i+1})$ can serve as a parametric characterisation for network engineering (example residual learning). While a comprehensive interpretation of these functions requires further experimentation across diverse datasets, several preliminary observations emerged. As shown in Figure 7.7(a), the fully linear model exhibits a drop in the PL value at the activation layer. This occurs because the identity function preserves the feature space and its underlying homology.

Layer-by-Layer : The fluctuating PL values between adjacent layers confirm that each layer performs a distinct topological transformation on the data manifold (see Figure 7.7). This transformation is governed by the specific activation function and the homology of the preceding manifold.



(a) Identity (b) Sigmoid (c) SiLU

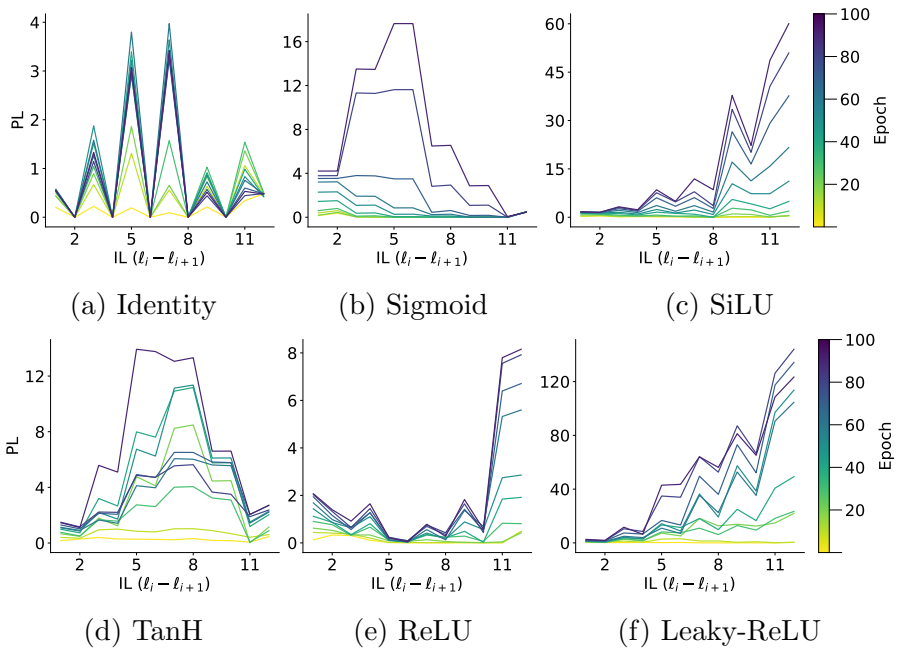
(d) TanH (e) ReLU (f) Leaky-ReLU

Figure 7.6: **Width Comparison:** In the heat-map, the intensity of the gradient represents the magnitude of the persistence landscape distance between adjacent layers. Lighter regions indicate smaller PL values, while darker regions signify higher PL values, representing the magnitude of topological transition. The index $i+1$ on the $y-$ axis is used to denote $IL(\ell_i, \ell_{i+1})$, to denote that the persistence landscape (PL) distance is computed between the i-th and $(i+1)$-th embedding spaces.

Training : Across all architectures, a distinct shift in the topological transition was observed, manifested as a vertical shift during training and a non-monotonic change across the successive layers, IL. Higher peaks on the Figure 7.7 align with the darker regions of the heat map in the Figure 7.6, signifying greater magnitude in topological transition. A vertical translation implies that the relative homological "slope" between layers remains consistent, suggesting that the structural evolution at any given step is heavily constrained by the state of the previous training iteration.

Depth vs Width : The wider networks performed better, probably due to the better representation of the data in the layers and the smaller size of the dataset. In deeper networks, the topological transition is consistently influenced by the preceding layer, thus enabling diverse topological transitions. When coupled with the observed vertical shift during training, this implies a topological consistency in weight optimisation that remains dependent on the initialisation.

Here, we can ask:"What would be the best method to keep the model learning ?" In a vanishing gradient problem, the feature space stops changing, and thus, over the course of an epoch, the topological transition will approach zero. Similarly, in the gradient explosion problem, the extracted embedding space is unstable and undergoes significant changes across the training epochs. The depth provides additional structure to the topological transition across layers, since the non-homeomorphism of the map depends entirely on the homology of the preceding layer. In a classification problem, the Betti numbers of the class need to collapse to a single point. If a network expands the feature space, sparsity will introduce homology that needs to be collapsed in the preceding layers. It is noted that the training was relatively unstable for most of the deeper models, but increasing the parameter space does improve performance. Further, it is observed that for deeper architectures with different activation functions, when the inter-layer topological transition across the layers is "comparatively small", then the persistent landscape distance between the final linear layer and the desired output space is large. This further emphasises the importance of studying topological transitions in deep neural networks.



(a) Identity (b) Sigmoid (c) SiLU

(d) TanH (e) ReLU (f) Leaky-ReLU

Figure 7.7: **Depth :** The line plot illustrates the topological transition across layers for architectures with identical depth but varying activations. Each line represents the transition magnitude at a specific epoch, where the x-axis corresponds to the index $i+1$, which represents that the topological transition is between the adjacent embedding spaces $IL(\ell_i, \ell_{i+1})$. Higher peaks on the plot signify greater topological transitions, while the final data point quantifies the distance between the output space and the desired response space Y. The $y-$ axis represent the value of the persistence landscape distance $PL(\ell_i, \ell_{i+1})$.

In the next section, we analyse how the homological capacity defined in Chapter 4 is influenced by the depth, and the least permissible value for L for which the homological capacity defined is a valid metric. This analysis further permits our choice to substitute blocks in place of layers for the computation of the empirical homological capacity.

7.5 Analysis on Upper Bound on Architectural Depth

Recall the definition (Chapter 4) of $\phi = \max_{c \in C} \omega(\mathcal{X}_c)$, the homological complexity of the space $\mathcal{X}_c$. Given $\beta = (1, 2, 0, \dots)$, for both classes, thus we have $\phi = 3$ and the respective optimal or ideal value for homological capacity $\Phi_f = \frac{1-3}{L} = \frac{-2}{L}$. For a network f and $\phi = 3$ the ideal homological capacity is $\Phi_f = 1 - 3 = -2$, corresponding to a slope angle of approximately $63.4°(\widetilde{\Theta})$ for depth $L = 1$. However, in our empirical evaluations, the models fail to classify either class. Consequently, the observed capacity is calculated as $\Phi_f = \frac{2-\phi}{L} = \frac{-1}{L}$. Specifically, for the wide network f_w where $L = 3, \Phi_{f_w} = -1/3$ and for the deeper network $f_d, L = 11, \Phi_{f_d} = \frac{-1}{11}$. Following Theorem 4.3 in Chapter 4, the defined homological capacity favours the shallower model. Note that for different depths $L = 1, 2, 3 \dots$, the ideal value of Φ_f decreases (e.g., $\Phi_f = -1, -0.6, \dots$), reinforcing the preference for $L = 3$ when both models achieve suboptimal results. This consistency holds across different activation functions $f_{w(a)}$ and $f_{d(a)}$, as their respective Φ_f values remain fixed by their depths.

Since we know that the given f does not classify for any of the classes, we have $\Phi_f = \frac{2-3}{L} = \frac{-1}{L}$. Note that the wider network f_w here has input, activation and output layers, so we let $L = 3$, and $\Phi_{f_w} = \frac{-1}{3}$ and for the deeper network $f_d, L = 11, \Phi_{f_d} = \frac{-1}{11}$. Moreover, as per Theorem 4.3 in Chapter 4, the defined homological capacity would prefer f_w. Note that for different depths $L = 1, 2, 3 \dots$, the ideal value of $\Phi_f = -1, -0.6, \dots$ decreases with depth, thus if a deeper model has an ideal value, then Φ_f

would prefer $L = 3$. Now consider the wide models with different activation functions $f_{w(a)}$ then we have $\Phi_{f_{w(a)}} = \frac{-1}{3}$ for all the models. We have a similar situation for the deep models with different activation functions.

Remark 7.2. *Recall from Chapter 4 that Φ_f identifies the minimum depth L required among the optimal models to resolve the underlying homological complexity. In this study, neither architecture successfully resolves the classification and further highlights the inherent bias of the Φ_f metric towards smaller L.*

To determine the range of L accommodated by Φ_f, we consider the smallest depth L_s alongside the highest non-optimal numerator, $2 - \phi$. Given the condition $\frac{1-\phi}{L} \geq \frac{2-\phi}{L_s}$ we can derive the bound for L. For our parameters ($\phi = 3, L_s = 3$), this implies $-2/L \geq -1/3 \implies 6 \geq L$. Thus, given the complexity ϕ and the minimum depth L_s the permissible depth is bounded by $L_s \frac{(1-\phi)}{(2-\phi)} \geq L$.

Notation (Upper bound on permissible L for Homological Capacity)**.** *Let $\phi = \max_{c \in C} \omega(\mathcal{X}_c)$ be the homological complexity of the data classes. The **Ideal Homological Capacity** is given by:*

$$\Phi_{f(ideal)} = \frac{1 - \phi}{L}$$

*where L denotes the number of layers. In instances where the model fails to resolve the underlying homology, we introduce the **least non-optimal penalty**, $2 - \phi$, into the numerator. For a deeper model L to be considered architecturally permissible over the baseline depth L_s, it must satisfy:*

$$\frac{1 - \phi}{L} \leq \frac{2 - \phi}{L_s} \implies L \leq L_s \frac{1 - \phi}{2 - \phi}$$

It is necessary that the values of depth L with respect to which Φ_f is computed satisfy the least permissible length to qualify as a model selection diagnostic measure. This analysis further explores the limitations of the given definition of the homological capacity.

7.6 Conclusion

The archetypal subspace serves as a critical proxy for validating model performance, particularly when training data is limited. We have demonstrated that while the isomorphism $H(\mathcal{X}_+) \cong H(\mathring{A}_{\mathcal{X}})$ is a useful characterisation, it is not a strictly necessary condition for homological generalisation. The empirical results indicate that the topological transition observed during training can be leveraged for Network Architecture Engineering.

Specifically, architectural depth introduces diverse characteristics to these transitions, while an increase in parameter space enhances the topological transitions in the Neural Network. The observed increase in PL across the epoch likely reflects the refinement of non-homeomorphic mapping[4], which is essential for collapsing the higher-order Betti numbers of the input classes into a single connected component ($\beta = (1, 0, 0 \dots)$). While further experiments are required to verify the broader utility of the homological capacity metric Φ_f, we have derived a theoretical upper bound on the permissible depth L. This bound provides a rigorous criterion for architectural selection based on Φ_f, a measure specifically derived for classification tasks. In the following Chapter 8, we demonstrate how topological transitions are fundamentally task-driven by developing a framework that extends these principles to generative modelling.

Chapter 8

Persistence Entropies In The Generative Space

This chapter examines how persistent homology can be exploited beyond the classification setting by extending its application to Generative Adversarial Networks (GANs) [18]. In this context, we analyse the structural integrity of generated outputs through two primary lenses:

Training : The distribution of Persistence Entropy [90] is employed during the adversarial training process to quantify the topological evolutions of the Generator as it learns the training manifold.

Topological Evaluation : We propose a cubical homology based measure to assess the quality and diversity of generated images, focusing on the preservation of topological structure via persistent homology.

This study demonstrates that a fundamental requirement for generative learning is the successful topological consistency of the training manifold by the generated manifold in a cubical setting. Further, computational topology in the cubical setting can be utilised to derive a framework for evaluating the generated outputs.

8.1 Generative models

Generative learning [18] is the type of learning where the machine learning models are made to learn the underlying distribution of data to generate new, similar outputs. These models have driven the new era of machine learning, with many finding real-world applications, such as large language models in machine translation, summarisation, and creative AI for generating new images, music, or videos. There are many types of generative models, such as the Generative Adversarial Network (GAN), which uses a generator that learns the data distribution and a discriminator that distinguishes between them (see Figure 8.1). Autoencoders are used to learn a lower-dimensional representation. These models can be used to learn representations that preserve topology. Then there are transformer models that process sequential data. These models serve as the foundation for AI models that excel at multitasking.

8.1.1 Assessment of Generated Space

As generative models improve at mimicking real-world data, the need for methods to distinguish between generated and real images has become increasingly evident. The learning in a GAN is based on a game between the Generator and the Discriminator, with a solution known as the Nash equilibrium. A local Nash equilibrium is reached if neither the Generator nor the Discriminator can unilaterally decrease their respective losses. Often resulting in mode collapse when the Generator gets biased towards a few samples.

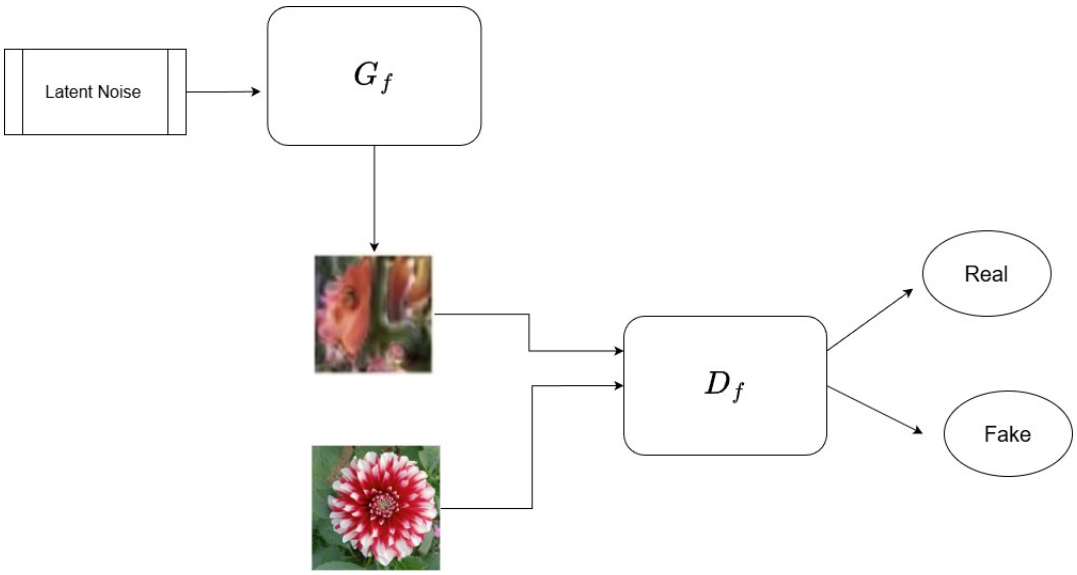


Figure 8.1: **Generative Adversarial Network (GAN) Architecture:** The Figure illustrates an adversarial framework where a **Generator** and a **Discriminator** are trained simultaneously. The Generator learns to map latent noise to the data distribution, while the Discriminator acts as a binary classifier to distinguish between real training samples and synthetic outputs. This competitive training process forces the Generator to produce increasingly high quality samples and thus approximate the topological structure of the training samples.

IS : The Inception Score [91] utilises the conditional class probability based on an image classifier (InceptionV3 Network pre-trained on ImageNet) to measure the quality and diversity of generated images. A high score is expected to indicate images that are clear, diverse, and identifiable. The score is based on the KL-divergence between the conditional and marginal class distribution. Images with meaningful objects are expected to have low label entropy, whereas high image variance results in high entropy. Methods using Inception models are computationally intensive and not interpretable. They are also sensitive to the slightest changes in the Inception model's weights and can yield misleading results on datasets other than the Source dataset (here, the Imagenet dataset).

FID : The Fréchet Inception Distance score [92] for generative models is also based on the Inceptionion Network. The lower FID score indicates that the data is closer to the original (FID $\in [0, \infty)$). The FID score also requires Inception models or other similar pertained models, which makes their results susceptible to the model performance. Given μ_R and μ_G are the means and cov_R and cov_G are the covariances for the real and generated data distribution, respectively, and Tr is the trace of the matrix, then the FID score is given by

$$FID = |\mu_R - \mu_G|^2 + Tr(cov_R + cov_G - 2(cov_R * cov_G)^{\frac{1}{2}})$$

Topological : There exist several topological methods, such as the Betti score [93], which compute a histogram for all real and generated samples and compare them using the chi-squared distance. The maximum mean discrepancy (MMD) [93] between the mean of the persistence diagram based on the persistence vectorisation in the Hilbert space. This method involves computing the Wasserstein distance between the diagrams. However, the Betti score is an unstable persistence measure, and the Wasserstein distance is computationally intensive. The M-top Divergence or Manifold Topology Divergence [94] that computes the topological discrepancy between the manifolds through the construction of simplicial complexes between the real and generated manifolds. This method, although it scales well for high-dimensional data, is not based on the traditional persistent homology. The geometric score [95] is based on the mean Relative Living Times (RLT) of topological features of the witness complex and stochasticity

(a) Image (b) Mean of Image (c) H_0 (d) H_1

Figure 8.2: In this work, we use cubical homology on the individual images. The red-highlighted region is the homology (c) H_0 and (d) H_1 observed in the persistence diagram of (b).

that requires multiple sampling. The R-Top divergence [96] (Representation Topology Divergence) is similar to M-top, a multi-scale topological dissimilarity that involves computing weighted graphs using simplices between two representations of data.

While popular methods like IS and FID are computationally intensive or dataset dependent, topological approaches, as in this work, offer a more structural perspective. Here, we introduce a framework grounded in persistent homology that specifically analyses "topological diversity" via persistence entropy. By doing so, we provide a distinct methodology that quantifies the quality and mode collapse in generated space, through an interpretable, stable, computationally efficient alternative to traditional metrics.

8.2 Persistence Entropy in the Cubical Setting

While GANs are typically evaluated on their ability to match the feature distributions of real data, these features are often high dimensional and opaque. By employing persistence entropy, we can summarise the structural complexity of the generated manifold into a stable, scalar representation. This section formalises the mathematical underpinnings of persistence entropy and establishes the statistical basis for using it as a robust evaluative tool.

Consider a persistence diagram (dmg) where each point (b_i, d_i) corresponds to a feature with lifetime $d_i - b_i \geq 0$. The persistence entropy is computed over the set of finite, normalised lifetimes.

Definition (Persistence Entropy)**.** *Given the dgm corresponding to a filtration F, the normalized lifetime $\mathcal{N}$ is given by*

$$\mathcal{N} := \{l_i = \frac{d_i - b_i}{\bar{l}} | 1 \leq i \leq s\}$$

where $\bar{l} := \sum_{i=1}^{s} d_i - b_i$ such that $d_i - b_i$ is finite, $(b_i, d_i) \in \widehat{H}$ and $s = |\widehat{H}|$. Let $\mathcal{N}_k = \{l_i \in \mathcal{N} : l_i$ is associated with $\widehat{H}_k\}$. Then persistence entropy E_k is defined as [97] :

$$E := \sum_{k=0}^{K} E_k \text{ where } E_k := -\sum_{l_i \in \mathcal{N}_k} l_i \log(l_i)$$

Where $\widehat{H} = \{(b_i, d_i) \in dgm : d_i < \infty\}$ is the collection of homological features in dgm within finite lifetime and $\widehat{H}_k \subseteq \widehat{H}$ denotes the subset of features with dimension k.

Essential infinite lifetimes (l_∞) are excluded from $\mathcal{N}$ in the dgm and do not contribute to E (One can take this as $\{l_\infty = 1\}$, and $\log(l_\infty) = 0$). Here E is the entropy with respect to the whole dgm. We will keep K fixed throughout this work. By definition $E \in [0, \log(s)]$.

8.2.1 Stability of Persistence Entropy

Persistence entropy inherits the stability properties of persistent homology. In this subsection, we revisit the conditions under which persistence entropy remains continuous relative to the Wasserstein distance between persistence barcodes. Further, we built the framework for assessing the persistence entropy using cubical homology. Let $\mathbb{D} = \{D_i\}$ be a finite collection of cubical sets Q for computing the cubical homology (here we do not need the classes, so we ignore them, $D_i \in R^{n_r \times n_c}$ for images [77]).

Notation. *Let $\mathscr{B}_{\mathbb{D}} := \{dgm(D_i) : D_i \in \mathbb{D}\}$ be the collection of finite persistence barcodes corresponding to the dataset $\mathbb{D}$. The set of persistence entropy is given by $\mathscr{E}(\mathbb{D}) = \{E_{D_i} : D_i \in \mathbb{D}\}$., where each E_{D_i} is computed from the normalised $\mathscr{B}_{D_i}$.*

The stability theorem for persistence entropy [97] given over the set of finite persistence barcode $\mathscr{B}$ and the $p-$th Wasserstein distance $\rho_{(w,p)}$ (see Chapter 3) between the barcodes is given as follows:

Theorem (Referring [97])**.** *Let $A, B \in \mathscr{B}$ and let $\rho_{(w,p)}$ be the p-th Wasserstein distance with $1 \leq p \leq \infty$. If we fix a maximum number of intervals and a minimum sum of the lengths of the intervals in a persistence barcode, then the persistent entropy E is continuous on $(\mathscr{B}, \rho_{(w,p)})$:*

$$\forall \epsilon \; \exists \; \delta \text{ such that } \rho_{(w,p)}(A, B) \leq \delta \implies |E_A - E_B| \leq \epsilon$$

While originally formulated in the context of simplicial complexes, the stability properties of persistence entropy depend only on the persistence diagrams; thus, Theorem 8.2.1 applies naturally to the cubical setting used in this work.

8.2.2 Distribution of Persistence Entropy

While individual persistence entropy values provide a structural summary of a single image or sample, evaluating a generative model requires an aggregate understanding of the entire generative manifold. By treating the entropy values of a collection of images as a statistical distribution, we can compare the global characteristics of real and generated manifolds. This subsection defines the mean and variance of these entropy distributions, providing the necessary metrics to quantify both the quality and the diversity of the generative space.

Remark 8.1. *Let $\mathscr{B}$ be the space of finite persistence barcodes for which Theorem 8.2.1 holds. Let $\mathbb{D}$ be a finite dataset such that the resulting barcodes $\mathscr{B}_{\mathbb{D}}$ are a subset of $\mathscr{B}$, where Theorem 8.2.1 is applicable. We denote the collection of associated persistence entropies as $\mathscr{E}_{\mathbb{D}}$.*

Since the values in $\mathscr{E}_{\mathbb{D}}$ are real and positive, we treat them as a distribution and define the following statistical summaries:

Definition 8.1 (Mean and Variance of Persistence Entropy)**.** *Given the collection $\mathscr{E}_{\mathbb{D}}$, the mean persistence entropy $\mu(\mathscr{E}_{\mathbb{D}})$ and the sample variance $Var(\mathscr{E}_{\mathbb{D}})$ over the dataset $\mathbb{D}$ are defined as :*

$$\mu(\mathscr{E}_{\mathbb{D}}) = \frac{1}{|\mathscr{E}_{\mathbb{D}}|} \sum_{E_{D_i} \in \mathscr{E}_{\mathbb{D}}} E_{D_i};$$

$$Var\,(\mathscr{E}_{\mathbb{D}}) = \frac{1}{|\mathscr{E}_{\mathbb{D}}| - 1} \sum_{E_{D_i} \in \mathscr{E}_{\mathbb{D}}} (E_{D_i} - \mu(\mathscr{E}_{\mathbb{D}}))^2$$

where $|\mathscr{E}_{\mathbb{D}}| = |\mathbb{D}|$ represents the finite number of cubical sets in the collection. For brevity, we denote $\mu(\mathscr{E}_{\mathbb{D}}) = \breve{\mu}_{\mathbb{D}}$ and $\mathrm{Var}(\mathscr{E}_{\mathbb{D}}) = \breve{\sigma}^2_{\mathbb{D}}$, $E_{D_i} = E_i$ from here onwards. These measures contribute to the **topological diversity** of the given collection of cubical sets.

Remark 8.2. *Let $\mathbb{A}, \mathbb{B} \subseteq \mathbb{D}$ be two finite subsets of the superset $\mathbb{D}$ of cubical sets. We denote the associated collections of persistence entropies as $\mathscr{E}_{\mathbb{A}}$ and $\mathscr{E}_{\mathbb{B}}$, respectively. By this inclusion, $\mathscr{E}_{\mathbb{A}}$ and $\mathscr{E}_{\mathbb{B}}$ inherit the stability and boundedness properties established for $\mathscr{B}$, ensuring that Theorem 8.2.1.*

We now introduce our proposed measure, based on the difference between the entropies of two cubic sets.

Definition 8.2. *The absolute difference in persistence entropy between individual cubical sets $A_j \in \mathbb{B}$ and $B_j \in \mathbb{B}$ is defined as*

$$|\Delta_{ij}| := |E_i - E_j|$$

The absolute mean difference between the two collections of cubical sets $\mathbb{A}$ and $\mathbb{B}$ is given by :

$$|\Delta_{\mathbb{AB}}| = |\breve{\mu}_{\mathbb{A}} - \breve{\mu}_{\mathbb{B}}|$$

. Where $\breve{\mu}_{\mathbb{A}}$ and $\breve{\mu}_{\mathbb{B}}$ are the mean persistence entropies of the respective subsets.

Let us look at some properties of $|\Delta|$:

Lemma 8.3. *Let $\mathbb{A}, \mathbb{B}$ and $\mathbb{Z}$ be cubical sets as defined above, with their collection of persistence entropies as $\mathscr{E}_{\mathbb{A}}, \mathscr{E}_{\mathbb{B}}$ and $\mathscr{E}_{\mathbb{Z}}$, then:*

1. *$|\Delta_{ij}|$ is symmetric, $E_i \in A$ and $E_j \in B$ any two persistence entropies. $|\Delta_{\mathbb{AB}}|$ is symmetric for the cubical sets $\mathbb{A}$ and $\mathbb{B}$.*
2. *$||\Delta_{\mathbb{AB}}| - |\Delta_{\mathbb{BZ}}|| \leq |\Delta_{\mathbb{AZ}}|$*

Proof. The proof is trivial. Given $\mathbb{A}, \mathbb{B}$ and $\mathbb{Z}$ are cubical sets with their collection of persistence entropies $\mathscr{E}_{\mathbb{A}}, \mathscr{E}_{\mathbb{B}}$ and $\mathscr{E}_{\mathbb{Z}}$ then :

1. We know that $E_i \geq 0, E_j \geq 0$. By definition :

$$|\Delta_{ij}| = |E_i - E_j| = |-(E_j - E_i)| = |\Delta_{ji}|$$

   Similarly, we have : $|\Delta_{\mathbb{AB}}| = |\breve{\mu}_{\mathbb{A}} - \breve{\mu}_{\mathbb{B}}| = |-(\breve{\mu}_{\mathbb{B}} - \breve{\mu}_{\mathbb{A}})| = |\Delta_{\mathbb{BA}}|$

2. The inequality follows directly

$$||\Delta_{\mathbb{AB}}| - |\Delta_{\mathbb{BZ}}|| = ||\breve{\mu}_{\mathbb{A}} - \breve{\mu}_{\mathbb{B}}| - |\breve{\mu}_{\mathbb{B}} - \breve{\mu}_{\mathbb{Z}}|| \leq |\breve{\mu}_{\mathbb{A}} - \breve{\mu}_{\mathbb{Z}}| = |\Delta_{\mathbb{AZ}}|$$

□

The metric properties hold for the absolute difference $|\Delta|$, but the signed difference Δ is anti-symmetric. Based on the definition, we have $\mathscr{B}_A \subseteq \mathscr{B}, \mathscr{B}_B \subseteq \mathscr{B}$. Now we prove the following theorem for the absolute mean persistence entropy.

Theorem 8.4 (Topological Mean stability)**.** *Let $\mathbb{A}$ and $\mathbb{B}$ be two collections of cubical sets of equal cardinality, $|\mathbb{A}| = |\mathbb{B}| = N$ and their persistence entropy $\mathscr{E}_{\mathbb{A}}$ and $\mathscr{E}_{\mathbb{B}}$. Suppose $\forall \epsilon > 0$, $\exists \delta > 0$ such that $\rho_{(w,p)}(A_i, B_i) \leq \delta$, $A_i \in \mathbb{A}, B_i \in \mathbb{B}$, $i \in \{1, \ldots, N\}$ then $|\Delta_{\mathbb{AB}}| \leq \epsilon$.*

Proof. By the stability of persistence entropy (Theorem 8.2.1), the condition $\rho_{(w,p)}(A_i, B_i) \leq \delta$ implies $|\Delta_{ii}| \leq \epsilon$ for all $i \in \{1, \ldots, N\}$. By the definition of the absolute mean difference $|\Delta_{\mathbb{AB}}|$ and applying the triangle inequality:

$$|\Delta_{\mathbb{AB}}| = \frac{1}{N}|\sum_{i=1}^{N} E_{A_i} - \sum_{i=1}^{N} E_{B_i}| \leq \frac{1}{N}\sum_{i=1}^{N} |E_{A_i} - E_{B_i}|$$

Substituting the stability bound $|E_{A_i} - E_{B_i}| \leq \epsilon$ into the summation :

$$|\Delta_{\mathbb{AB}}| \leq N \cdot \frac{\epsilon}{N} = \epsilon$$

Thus $|\Delta_{\mathbb{AB}}| \leq \epsilon$ as required. □

Now consider the contrapositive form of the above $\epsilon - \delta$ theorem. The contrapositive statement implies that if the difference of the mean of the cubical sets is not stable with respect to $\rho_{(w,p)}$, then there exists at least one pair $A_i \in A, B_j \in B$ such that their persistence entropies are far apart. This follows from the fact that the sum over the absolute difference between the pairs from a distribution of equal size is greater than the absolute difference between their means.

Theorem 8.5 (Topological Mean Divergence)**.** *Let $\mathbb{A}$ and $\mathbb{B}$ be two collections of cubical sets of equal cardinality $|\mathbb{A}| = |\mathbb{B}| = N$ and their respective persistence entropy $\mathscr{E}_{\mathbb{A}}$ and $\mathscr{E}_{\mathbb{B}}$. For any $\epsilon > 0$ $\exists$ $\delta > 0$ such that $|\Delta_{\mathbb{AB}}| > \epsilon$ then there exist at least one pair $A_i \in A, B_j \in B$, such that $\rho_{(w,p)}(A_i, B_j) > \delta$ and $|\Delta_{ij}| > \epsilon$.*

Proof. The proof is a direct consequence of the above Theorem 8.4 and Theorem 8.2.1. Consider the contrapositive of the following statement :

$$\forall\epsilon\exists\delta\Big(\forall A_i \in \mathbb{A}, B_j \in \mathbb{B} \text{ such that } \rho_{(w,p)}(A_i, B_j) \leq \delta \implies |\Delta_{\mathbb{AB}}| \leq \epsilon\Big)$$
$$\forall\epsilon\exists\delta\Big(\neg(|\Delta_{\mathbb{AB}}| \leq \epsilon) \implies \neg(\rho_{(w,p)}(A_i, B_j) \leq \delta \forall A_i \in \mathbb{A}, B_j \in \mathbb{B})\Big)$$
$$\forall\epsilon\exists\delta\Big(|\Delta_{\mathbb{AB}}| > \epsilon \implies \exists A_i \in \mathbb{A}, B_j \in \mathbb{B} \text{ such that } \rho_{(w,p)}(A_i, B_j) > \delta\Big)$$
$$\forall\epsilon\exists\delta\Big(|\Delta_{\mathbb{AB}}| > \epsilon \implies \exists A_i \in \mathbb{A}, B_j \in \mathbb{B}, \text{ such that } |\Delta_{ij}| > \epsilon\Big)$$

The proof is complete as the contrapositive statement is logically equivalent to the original conditional statement. □

This theorem states that the absolute mean entropy $|\Delta_{\mathbb{AB}}|$ is stable under the Wasserstein distance.

Remark 8.6. *The proof for Theorem 8.5 follows the same logic as the one-to-one correspondence in Theorem 8.4. Consider $\forall\epsilon > 0$, $\exists\delta > 0$ such that $\rho_{(w,p)}(A_i, B_j) \leq \delta$, for all pairs $i, j \in \{1, 2, \ldots N\}$, $A_i \in \mathbb{A}, B_j \in \mathbb{B}$ then $|\Delta_{\mathbb{AB}}| \leq \epsilon$ necessarily follows. Since the all-to-all condition implies the subset condition $\rho_{(w,p)}(A_i, B_i) \leq \delta$, the proof remains structurally identical to that of Theorem 8.4.*

The stability guarantees established in Theorem 8.5 provide the mathematical justification for using persistence entropy as a reliable comparative metric. The absolute mean entropy difference, $|\Delta_{\mathbb{AB}}|$, is thus employed to investigate the topological evolution of the generative manifold during training. Furthermore, by analysing the distribution $\mathscr{E}$ of persistence entropy, we can quantitatively evaluate and compare model performance across diverse datasets. To validate this framework, we conduct experiments using the datasets and models detailed below.

8.3 Experimental setup

In addition to the **sTL10** and **AIRCRAFT** datasets described in Chapter 4, this study incorporates the **Flowers102** dataset (also known as the Oxford 102 Category Flower Dataset), which contains between 40 and 258 images per class. The following Generative Adversarial Network (GAN) architectures are evaluated:

DCGAN [98] : Deep Convolutional Generative Adversarial Network utilises fractionally strided convolutions for upsampling in the Generator and strided convolutions in the Discriminator instead of max-pooling. For stable learning, it eliminates fully connected layers and excludes batch normalisation from the generator output layer and the discriminator input layer

BEGAN [99] : Boundary Equilibrium Generative Adversarial Networks utilises a loss derived from the Wasserstein distance and incorporates an equilibrium hyperparameter to balance the Generator and Discriminator. This approach matches auto-encoder loss distributions rather than directly matching data distributions.

R1GAN [100] : R1 Regularisation in Generative Adversarial Network, in particular, employs a simplified gradient penalty to ensure Nash equilibrium. By penalising the discriminator gradient on real data, this method stabilises training and addresses the limitations inherent in unregularised GAN optimisation.

BDCGAN : Additionally, for comparison we employ BDCGAN, which utilises an architecture identical to DCGAN but operates with a generative space consisting of poor quality similar images. These images are characterised by high levels of noise and a lack of discernible structure.

The GANs implemented are in the GitHub Repo GAN-Tutorials. The R1GAN performed best with satisfactory FID and Inception scores, as in the Table 8.1. The persistence entropy is computed over the channel mean of the RGB images, thereby speeding up the computation. The following theorems and empirical results explore how training in a generative model can be explored through the lens of the generated space.

8.4 Topological evolution of the generated space

Beyond evaluating the generative model, topological analysis offers a powerful lens for observing the learning dynamics of generative networks such as GANs. For instance, as a GAN shifts from generating random noise to structured images, the underlying manifold of the generated space undergoes a series of topological evolutions. If a model successfully learns the target distribution, its generated manifold should not only approximate the appearance of the training data but also its structural complexity and diversity, here the topological diversity.

In this section, we formalise the topological evolution of the generated space in terms of topological diversity. By treating the training process as a sequence of evolving manifolds, we demonstrate that the convergence of a generative model can be characterised by the difference in the mean and variance of the distribution of persistence entropy. This theoretical framework provides a diagnostic tool to identify successful convergence, detect mode collapse, and monitor the stability of the equilibrium reached between the Generator and the Discriminator.

Consider a generative model G trained on a cubical dataset $\mathbb{A}$ for $T \in \mathbb{N}$ epochs. Let $\mathbb{G}^t = \{G_1^t, \dots G_N^t\}$ denote the collections of cubical sets generated by G at epoch t where $|\mathbb{A}| = |\mathbb{G}^t| = N$. As the model training progresses ($t \to T$), the generated images ideally need to converge toward the training data $\mathbb{A}$. Suppose that for each cubical set, we have at least one cubical set that is homologically similar to each other or a pointwise convergence in the Wasserstein metric:

$$\forall i \in \{1, \dots, N\}, \rho_{(w,p)}(G_i^t, A_i) \to 0 \text{ as } t \to T$$

By the stability of persistence entropy (Theorem 8.2.1), this implies the convergence of individual entropies:

$$|\Delta_{ii}| = |E(A_i) - E(G_i^t)| \to 0$$

By applying the Topological Mean stability Theorem (Theorem 8.4), we can conclude that the collective topological diversity of the generated space also converges to that of the target manifold :

$$|\Delta_{\mathbb{A}\mathbb{G}^t}| = |\breve{\mu}_{\mathbb{A}} - \breve{\mu}_{\mathbb{G}^t}| \to 0 \text{ as } t \to T$$

Theorem 8.7 (Variance Convergence)**.** *Consider a sequence of finite collection cubical sets* $\mathbb{G}^t = \{G_1^t, \dots G_N^t\},\ t \in \{1, 2 \dots T\}$ *that point-wise converges to* $\mathbb{A}$ *given by* $\forall i \in \{1, \dots, N\},\ \rho_{(w,p)}(G_i^t, A_i) \to 0$ *as* $t \to T$*. Then* $|\breve{\sigma}^2_{\mathbb{A}} - \breve{\sigma}^2_{\mathbb{G}^t}| \to 0$ *as* $t \to T$*.*

Proof. Since the sequence of finite collections $\mathbb{G}^t$ pointwise converges, it follows from the stability of persistence entropy (Theorem 8.2.1) $|E(G_i^t) - E(A_i)| \to 0 \forall i$. Furthermore, by the Topological Mean

stability Theorem (Theorem 8.4), $|\breve{\mu}_{\mathbb{A}} - \breve{\mu}_{\mathbb{G}^t}| \to 0$ as $t \to T$. From the definition of $\breve{\sigma}^2$ and identity of Variance, we have :

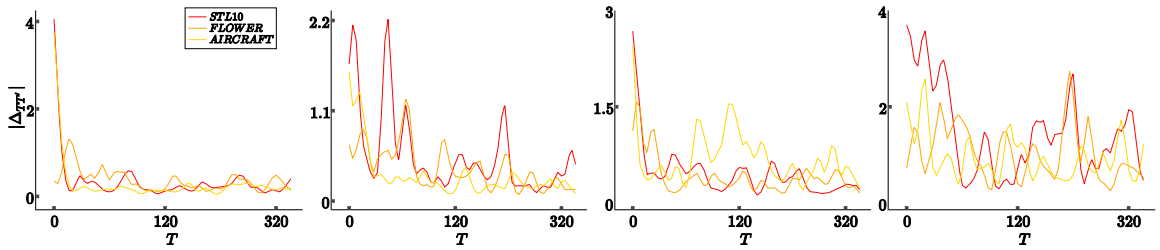
$$|\breve{\sigma}^2_{\mathbb{G}^t} - \breve{\sigma}^2_{\mathbb{A}}| = |\frac{1}{N-1}\left(\sum_{i=1}^{N}\left(E(G_i^t)^2 - E(A_i)^2\right)\right) - \frac{N}{N-1}(\breve{\mu}^2_{\mathbb{G}^t} - \breve{\mu}^2_{\mathbb{A}})|$$

As $t \to T$, $E(G_i^t) \to E(A_i)$ and $\breve{\mu}_{\mathbb{G}^t} \to \breve{\mu}_{\mathbb{A}}$. Since these sequences are non-negative and convergent, by the Algebraic Property of Limit Theorem [10], it follows that $E(G_i^t)^2 \to E(A_i)^2$ and $\breve{\mu}^2_{\mathbb{G}^t} \to \breve{\mu}^2_{\mathbb{A}}$. Substituting these limits into the expression above and since $N \in \mathbb{N}$ finite and fixed, we obtain:

$$\lim_{t \to T} |\breve{\sigma}^2_{\mathbb{G}^t} - \breve{\sigma}^2_{\mathbb{A}}| = 0$$

Thus, the difference in topological variance vanishes as pointwise convergence is achieved. □

The following theorem states that if the generated space converges to the target space, then the relative pointwise topological stability of the generated space results in the convergence of the global topological diversity of the distribution of persistence entropy of generated data toward that of the training manifold. Thus, the variance of the persistence entropy of the generated dataset converges to the variance of the target dataset (see table 8.1). This alignment in variance is a critical analysis that suggests that the model has progressed beyond merely approximating a single 'average' topological structure and has instead begun to learn the full range and distribution of structural complexities inherent in the original data. Thus, a minimal difference in topological diversity serves as an indicator that the generated dataset has achieved a balanced representation of the target topological diversity, effectively mitigating the risk of mode collapse.



(a) R1GAN (b) BEGAN (c) DCGAN (d) BDCGAN

Figure 8.3: **Tracking Topological mean Persistence**: In the following Figure $T' = T + 1$ and $|\Delta_{TT'}| = |\Delta_{\mathbb{G}^T\mathbb{G}^{T'}}|$ is the changing mean persistence entropy between successive layers, across different GAN models during training. In particular, the large change in the mean of the distribution of entropy implies instability, such as observed in the BDCGAN. Further, since this is computed independent of $\breve{\mu}_{\mathbb{A}}$, the change is a characteristic of the learning, such as architecture or optimisation.

Theorem 8.8 (Successive Topological Stability)**.** *Let $\mathbb{G}^t = \{G_1^t \dots G_N^t\}$ be a sequence of finite collection of cubical sets and $\mathbb{A}$ be a collection of cubical sets such that $|\mathbb{A}| = |\mathbb{G}^t| = N$, $\forall t \in \{1, 2 \dots T\}$. Suppose that as at $t \to T$ the persistence entropies converge point-wise $E(G_i^t) \to E(A_i)$, $\forall i \in \{1, \dots N\}$. Then the mean entropy difference between successive sequences $|\Delta_{\mathbb{G}^t\mathbb{G}^{t+1}}|$ vanishes as $t \to T$.*

Proof. By the Triangle Inequality for the absolute mean difference $|\Delta|$, we have :

$$\forall t \ , \ 0 \leq |\Delta_{\mathbb{G}^t\mathbb{G}^{t+1}}| \leq ||\Delta_{\mathbb{A}\mathbb{G}^t}| + |\Delta_{\mathbb{A}\mathbb{G}^{t+1}}||$$

Given the pointwise convergence as $t \to T$, $|\Delta_{\mathbb{A}\mathbb{G}^t}| \to 0$ and $|\Delta_{\mathbb{A}\mathbb{G}^{t+1}}| \to 0$, then by the squeeze theorem, it follows that $|\Delta_{\mathbb{G}^t\mathbb{G}^{t+1}}| \to 0$ as $t \to T$. Thus, the mean entropy difference between successive epochs vanishes when there is a pointwise convergence. □

This theorem demonstrates that as the GAN training converges to a stable state, the incremental topological evolution measured by the mean entropy difference between successive epochs vanishes. Empirically, in the Figure 8.3, we see that the R1GAN with the highest performance demonstrate (see FID and Inception scores Table 8.1) this topological stability during training. The empirical investigation concerning the persistent entropy distribution can be summarised as follows:

ES statistical test : The ES statistical test (Epps-Singleton [101, 102]) is a statistical test used to evaluate the null hypothesis that two samples are drawn from the same underlying distribution. A smaller ES statistic indicates that the generated entropy distribution $\mathscr{E}_{\mathbb{G}}$ is closer to the target distribution $\mathscr{E}_{\mathbb{A}}$. As shown in Table 8.1, these topological measures correlate with traditional performance metrics, specifically the FID and Inception Score.

Training : For architectures with superior convergence properties, such as the R1GAN, the successive epoch difference $|\Delta_{G^tG^{t+1}}|$ vanishes rapidly, signalling a stable approach to a topological equilibrium. Conversely, for the poorly performing DCGAN (referred to here as a BDCGAN or non-convergent baseline),$|\Delta_{G^tG^{t+1}}|$ exhibits persistent oscillations, suggesting a lack of convergence.(see Figure 8.3)

Different Dataset : Across the STL-10, AIRCRAFT, and Flowers102 datasets, the generative models exhibit consistent topological evolution during the learning process [103]. This suggests that the observed topological evolution is a fundamental characteristic of the model optimisation or architecture rather than a dataset-specific phenomenon.

These results demonstrate that persistence entropy is a robust descriptor for the generative space. By providing a stable link between theoretical convergence and empirical performance, this framework offers a rigorous assessment of model convergence. The proposed framework is further validated by applying the Epps-Singleton (ES) statistical test to the persistence entropy distribution. As observed in Table 8.1, lower ES scores, such as for R1GAN and DCGAN architectures, strongly correlate with lower FID and higher IS values. This alignment suggests that models with greater similarity in the topological diversity to the target dataset also exhibit higher generative quality. Conversely, the significantly higher ES scores observed in models such as BEGAN and BDCGAN highlight structural degradation and potential mode collapse that traditional metrics may not fully capture or interpret.

8.5 Conclusion

The application of mean persistence entropy provides a robust framework for measuring topological evolution within generative models. Here, the topology of the generated manifold is expected to converge toward the training manifold during the learning process. On an individual level, the generated cubical sets should ideally recover the structural complexity and resolution of the real images. This topological diversity is also reflected in the ES statistical test in the Table 8.1, where the distributional similarity of persistence entropy between the generated and original data indicates diversity and quality. This is further validated by the correlation between ES test values and both FID and Inception scores.

While persistence entropy may not capture the full topological difference as persistence landscape or Wasserstein distances, we demonstrate that the mean persistence entropy of individual images is sufficient for evaluating generative model performance and interpreting topological evolutions. This framework is significantly more computationally efficient than traditional topological summaries and does not require an external pre-trained model for evaluation. Furthermore, the empirical findings suggest that monitoring the topological evolution of generated outputs, in addition to identifying a Nash equilibrium, provides a deeper understanding of the GAN learning process. The theoretical relation between the same requires further analysis. The proposed method is theoretically data-agnostic and empirically model-agnostic, as it evaluates the intrinsic topology of the output. While the present study centres on generative adversarial architectures, extending this framework to non image data structures and diffusion-based paradigms represents a promising avenue for future research.

Table 8.1: Comparative analysis of various GAN measures/scores.

Data set	Model	FID	IS	*ES*
STL10	R1GAN	20.70	4.70	41.58
STL10	BEGAN	34.79	1.44	1709.47
STL10	DCGAN	22.09	5.8	27.94
STL10	BDCGAN	23.74	1.48	250.217
FLOWERS	R1GAN	12.6	2.65	61.6959
FLOWERS	BEGAN	39.20	2.20	66.937
FLOWERS	DCGAN	24.54	1.85	350.821
FLOWERS	BDCGAN	27.49	1.26	375.07
AIRCRAFT	R1GAN	12.66	2.47	19.429
AIRCRAFT	BEGAN	16.96	1.80	41.6907
AIRCRAFT	DCGAN	11.02	2.09	9.4446
AIRCRAFT	BDCGAN	22.01	1.20	1112.450

In conclusion, this chapter establishes a rigorous mathematical link between the stability of individual persistence diagrams and the collective statistical behaviour of generative datasets. By proving the Mean Stability and Variance Convergence theorems, we have provided a theoretical foundation for using persistence entropy as a reliable "topological sensor." This shift from individual image analysis to a collective distributional perspective enables a more granular evaluation of generative models, moving beyond the limitations of traditional metrics such as the Nash equilibrium or FID scores. The extension to different data structures using different complexes is beyond the scope of this literature and will be left to future studies. Nevertheless any data with grid like topology can be analysed using the given framework. As the field moves toward more complex, high dimensional generative tasks, this model-agnostic topological framework provides a computationally efficient path for validating that artificial intelligence is not merely approximating labels, but is truly reconstructing the underlying manifold of the real world.

Chapter 9

Thesis Summary

Topological assumptions provide a rigorous framework for interpreting the behaviour of Neural Networks. Depending on the specific task, the required topological transition can be characterised as either a non-homeomorphism (which alters the underlying homology and topology) or a homological equivalence. This distinction is evident in established measures of topological expressivity [3, 29] and in applications such as segmentation [48].

Building upon these foundations, this work investigates the interpretation of topological transitions in Neural Networks. In contrast to previous research that primarily approached this problem through the weight space [2, 7], we analyse the classification task through the embedding space and the generative task through the generated manifold. By focusing on the topological transition of these representations rather than specific architectures, the theoretical framework developed in this work remains both model- and data-agnostic.

For the application, this work focuses on the model selection problem (See Figure 9.1). Given a collection of models, the objective is to identify the "best" model. While "best" can be interpreted in various ways, such as generalisation capability, we specifically interpret it in terms of the topological expressiveness [3] of the Neural Network. Theoretical works exist that interpret the dynamic nature of Deep Neural Networks through a topological lens, such as "Topological Entropy" [41, 44] and Euler Characteristics [104]. Another prominent approach is homological complexity (ω) or the sum of Betti numbers [3], which is the focus of this work.

In this work, the primary task is classification, with the Chapter 8 addressing the generative setup. Although image datasets are used, the theory remains task driven. The Chapter 4 formally defines the model selection problem in the Transfer learning setup and derives the "Hardness" measure, called the homological capacity Φ_f, relative to the depth of the Neural Network. Leveraging research on the exponential expressivity of depth [3], we quantify the topological transition over depth required to solve the classification task—specifically, to select the model with the "least depth" that satisfies the topological requirements. This results in a "hard bias" towards lower depth models, which is explored in Chapter 7. To address the immeasurability of the ideal measure Φ_f, we utilise approximations derived from the embedding space, which demonstrated a moderate correlation with the accuracy.

Thus, the transferability score here is determined through the homological capacity of the model. In particular, we compare the score against the fine-tuning accuracies of different model architectures (see Chapters 4 and Chapter 6). Other than the generalisation property inherent from the source to domain transfer learning setup, in this work, there are other instances that can be related to generalisation. An instance is the utilisation of archetypes as subsamples to evaluate the domain transfer score across the whole dataset.

Furthermore, the primary tool for extracting homology in this work is Persistent Homology. To manage the computational complexity of persistent homology [53] over large instance spaces, we utilise Cubical complexes in Chapter 4 and the Archetypal subspace in Chapters 5and Chapter 6. Additionally, the topological transition in a multi-layered neural network was explored using a point cloud of defined topology with different architectural parameters, such as activations and depth, in Chapter 7.

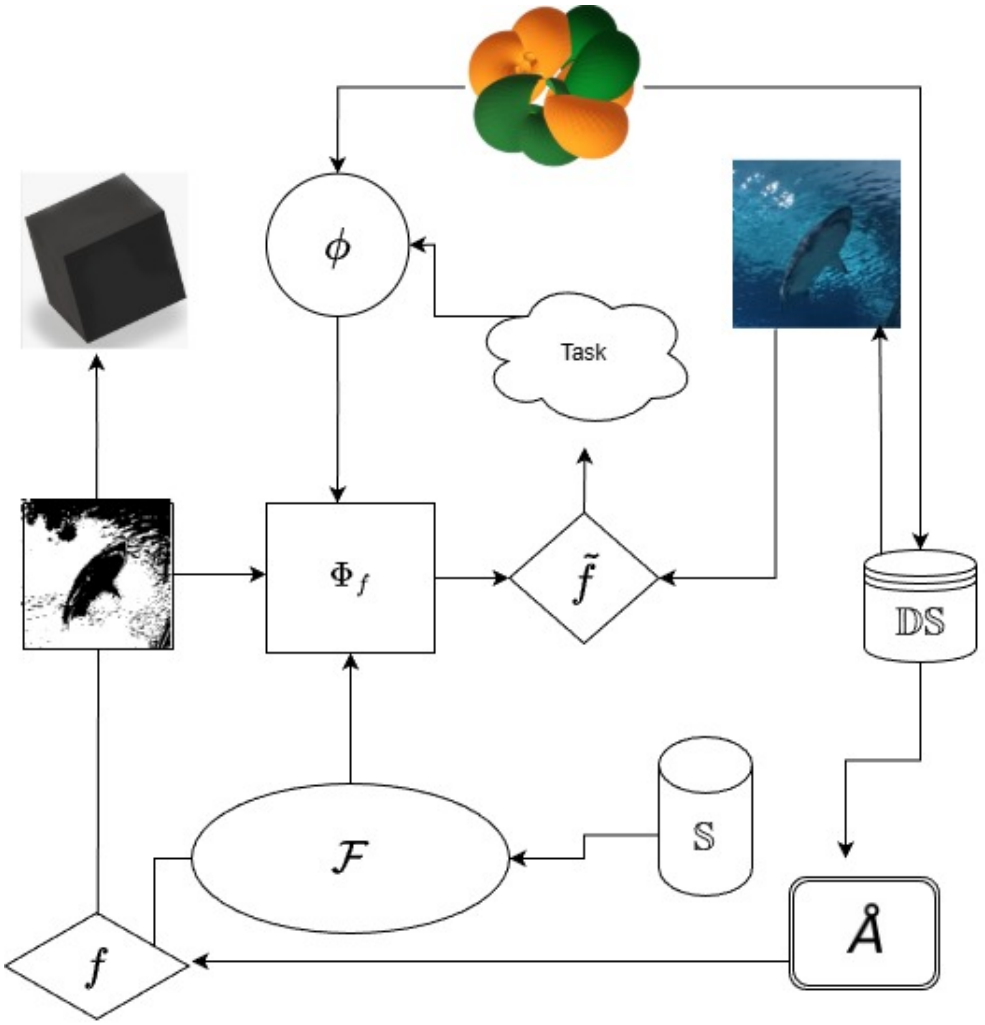


Figure 9.1: **Conceptual workflow for topology based model selection**: Given a source $\mathbb{S}$ trained function space $\mathcal{F}$ and a downstream dataset $\mathbb{DS}$ with maximal class homological complexity ϕ, we derive the homological capacity Φ_f for each model based on the feature space. To manage computational complexity, the archetypal space $\mathring{A}$ of $\mathbb{DS}$ serves as a proxy for extracting the embedding space and approximating Φ_f. The optimal mode $\tilde{f}$ is selected based on its ability to efficiently achieve the minimal negative homological capacity Φ_f given by $1 - \phi$ (For illustration of an optimal classifier see Figure 1.2 in Chapter 1). This framework addresses the model selection problem (See Chapter 4) by identifying which architecture from a given collection [1] best satisfies the homological requirements of the task.

Throughout this work, theoretical derivations are used to explain the empirical observations. This chapter provides a synthesis of the proposed hardness measure, summarises conclusions from the preceding chapters, and outlines their corresponding limitations and directions for future research.

9.1 Analysis of Homological Capacity

In this work, the homological capacity of a neural network classifier is defined relative to its architectural depth. Bianchini and Scarselli [3] established that neural networks express homological complexity exponentially with respect to depth, whereas a single layer (the Pfaffian function) does so only polynomially. Further, the paper by William et al. [29] showed that support homology of the decision boundary can be used to reject architectures incapable of expressing the homological complexity of the class in a binary setting. Specifically, Naitzat et al. [40] demonstrated that classification requires non-homeomorphic mappings to transform input topology into the target output space. Based on the same, the homological capacity of a classifier f with L layers is given by :

$$\Phi_f = \frac{\omega(\mathcal{X}_{\hat{c}}^L) - \omega(\mathcal{X}_{\hat{c}})}{L}$$

where $\hat{c} = \arg\max_{c \in C} \omega(\mathcal{X}_c)$, the class with the maximal homological complexity ($\phi = \omega(\mathcal{X}_{\hat{c}})$). Thus, homological capacity is the topological transition of the class with the maximal homological complexity in a Neural Network with L layers. We shall call this measure "Maximum Topological Class Expressivity for Neural Classifiers" ($MaxTCE - NC$). We have the following property for $MaxTCE - NC$:

Range : An ideal classification task implies a collapse of homological complexity $\omega > 0$, such that $\omega(\mathcal{X}_{\hat{c}}^L) \leq \phi \implies \Phi_f \leq 0$. The range of feasible values for homological capacity is $[-\infty, 0]$, where $\Phi_f = 0$ indicates homological preservation. If $\Phi_f > 0$, the mapping f introduces additional topological structures, suggesting a failure to classify for class $\hat{c}$. Particularly in empirical observations in chapter 7, it was found that the low topological transition in the layer resulted in a high topological difference between the output space and the desired response space.

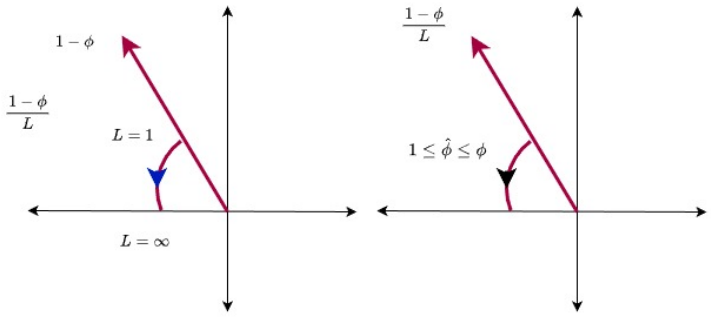
Asymptotic Behaviour : Based on its definition as $L \to \infty$, the homological capacity ($\Phi_f \to 0$) vanishes. The measure is therefore strictly informative for finite $L < \infty$. Similarly, when $\phi \to \infty$, Φ_f remains unbounded, so we assume finite ϕ.

Optimal : The optimal homological capacity is achieved when $\Phi_f = \frac{1-\phi}{L}$ implying $\omega(\mathcal{X}_{\hat{c}}^L) = 1$. (the class is reduced to a single connected component). While the ideal case maps a class to a single point, this may not be the case in real-world problems, such as when the feature space of the class may be a single component but get misclassified to multiple classes ($\omega(\mathcal{X}_{\hat{c}}^L) > 1$), indicating topological fragmentation.

Polynomial Extrapolation : Due to the immeasurability of ω for computing Φ_f. In this work, we employ polynomial extrapolation (h) over the embedding space. Letting the polynomial h to extrapolate $h(0)$ and $h(L)$, the Mean Value Theorem guarantees the existence of a point $\hat{o} \in (0, L)$ such that the instantaneous rate of change $h'(\hat{o}) = \frac{h(L)-h(0)}{L} = \frac{h(L)-\phi}{L}$, the average capacity equals Φ_f .This approach accommodates any monotonic topological behaviour without requiring layer-wise discretisation.

Architectural Selection : Φ_f assume that the classifier if has an optimal slope for ϕ inherently possesses the capacity to express all homologies $\hat{\phi} \leq \phi$. For a fixed ϕ, if two classifiers reach their optimal slopes at $\hat{L}$ and L respectively, where $\hat{L} \leq L$, the architecture with depth $\hat{L}$ is preferred. It achieves the necessary topological transition more efficiently, minimising depth-dependent computational overhead (see Figure 9.2).

Depth Bias : Our framework intentionally prefers lower depth architectures even if the absolute optimal condition $(1 - \phi)$ is not perfectly satisfied. Thus, we establish an upper bound for permissible depth based on the least non-optimal condition $(2 - \phi)$ (see Chapter 7). Given that depth provides exponential



(a) Depth L (b) ϕ

Figure 9.2: **Ideal classifier**: An optimal/ideal classifier f, is expected to have the optimal homological capacity $\Phi_f = \frac{1-\phi}{L}$. The Figure represents the change in Φ_f as the (a) depth L increases and (b) $h(0)$ decreases.

expressivity [3], an architecture meeting this condition is expected to be infinitesimally close to the optimal state. In this work, this bias is practically managed by performing empirical approximations over blocks.

The Chapter 4and Chapter 7 present the aforementioned concepts on homological capacity as theorems. However, several challenges persist regarding the definition of Φ_f, specifically that underlying assumptions do not include multiclass settings and the computational difficulty of determining ϕ. While these issues are partially mitigated through polynomial extrapolation, the nonlinear nature of higher-order polynomials introduces multiple slopes, and the precise identification of the point $\hat{o}$, guaranteed by the Mean Value Theorem, remains non-trivial. Also, while the embedding space contains critical information regarding the homological capacity Φ_f, current polynomial extrapolation methods provide a suboptimal approximation. Nevertheless, our empirical results observe a negative correlation with classification accuracy, which conforms to the definition of homological capacity. Although this correlation is moderate, it confirms the existence of an effective Φ_f that characterises the expressivity of neural networks. This study establishes the theoretical existence of $\hat{o}$ as an indicator of topological transition. However, further analysis is required to define a robust transferability score, and in this work, the same is referenced as a model diagnostic score. Finally, to enhance the interpretability of these transitions, we utilise the angular approximation $\widetilde{\Theta} = \tan^{-1}(\Phi_f)$, as a normalised representation of the capacity slope. The Figure 9.1 gives a glance into the entire framework explored in terms of persistent homology.

9.2 Instance based Complexity

To approximate the homological capacity Φ_f, we extract topological features from the neural network embedding space using Persistent Homology. However, the application of persistent homology is frequently constrained by instance based computational complexity. This work employs two distinct methodologies to mitigate this bottleneck:

Cubical complex : The adoption of cubical complexes is primarily motivated by the grid-like data structure of the input images. This approach enables the extraction of the average homological complexity (Ω) across images in the embedding space (see Chapter 4). A notable limitation of this method is the sensitivity to the filtration threshold η used to compute the homological complexity (ω_η) needed to compute Ω. Furthermore, while Φ_f is defined over the whole feature space of the class, this method primarily captures topological information from individual image instances.

Archetypal Subspace : In this work, archetypes are utilised as a disjoint, representative subsampling of the data to approximate Φ_f using the class feature space. Several theoretical and empirical conclusions

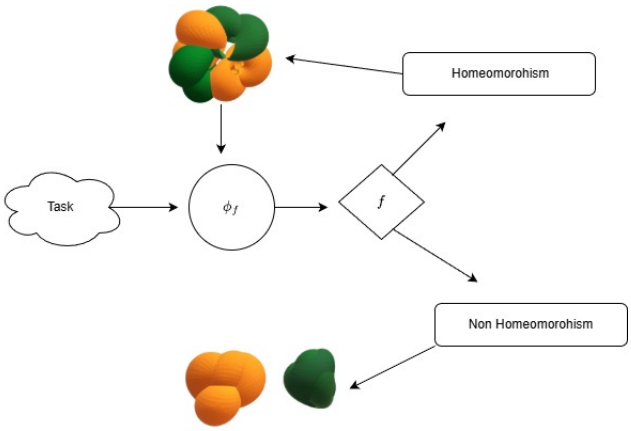


Figure 9.3: **Nature of "f":** The topological properties of f are intrinsically task-dependent and representation space observes these properties of f. As established in Chapter 4, classification tasks necessitate that f is a non-homeomorphism, as the architecture must effectively collapse each class manifold to a single component to achieve class separability. Conversely, the generative learning framework examined in Chapter 8 aims to preserve the underlying topological structure. In this context, f ideally acts as a homeomorphism that maps the latent space to the manifold of the training data. In terms of homology, this would require that the spaces are homologically equivalent.

were established regarding the archetypal subspace (see Chapters 5, 6, and 7). Specifically, archetypes were shown to: (a) introduce no extraneous homological structures to the data, (b) extend the Homological Generalisation Theorem to the archetypal subspace, and (c) empirically represent the underlying manifold structure. The geometric exploitation of the archetypal subspace provides a novel framework for dimensionality reduction in persistent homology, enabling the empirical study of homological expressivity without sacrificing critical topological features.

The suitability of cubical complexes for grid data structures was further adapted in Chapter 8. Specifically, we rephrased mode collapse in generative models in terms of topological diversity of persistence entropy of the generated space. Conversely, for archetypal analysis, the grid-structured output of Convolutional Neural Networks (CNNs) necessitated spatial averaging to approximate the layer-wise homological inference of the global feature space. Thus, both methodologies possess distinct merits: cubical complexes offer natural compatibility with image-based data structures, while archetypal subspaces provide a representative for the analysis of the homological equivalence necessary in a classification setting.

9.3 Empirical Synthesis and Discussion

The empirical results of this work support the task-driven nature of homological expressivity in neural networks. Experiments across Chapters 4 through 8 demonstrate that topological features extracted via computational topology can effectively characterise the Network behaviour. Specifically, across various classification frameworks, the constituent layers function as non-homeomorphic mappings, where the complexity of homological expressivity depends on both the architecture and the data manifold. By building upon established theoretical bounds [3, 40, 29], this study leverages topological expressivity to characterise real-world CNN architectures through their embedding spaces.

While the results in Chapter 7 do not provide a conclusive quantification of the expressive power of depth, they suggest that increased depth significantly alters the topological characteristics of the network, potentially complicating the optimisation landscape. Notably, initial layers appear to amplify topological structures, facilitating learning in subsequent, deeper layers (see the ResNet and DenseNet

analyses in Chapters 4 and 6), providing insight into depth-width dynamics and the resulting topological complexity.

Observations regarding activation functions remain preliminary; however, the Identity activation demonstrated relatively minimal topological transition between layers compared to nonlinear alternatives. Furthermore, the empirical evidence in Chapter 8 concerning the homological preservation of the generative manifold relative to the training data reinforces the task-driven nature of expressivity (see Figure 9.3). These findings underscore the necessity of topological preservation as a fundamental requirement for generative learning.

The results in Chapter 7 are not conclusive of the expressive power of depth as in the paper [3](which we leave for future analysis. However, the findings suggest that depth enhances the topological characteristics of the network, making it difficult to train and optimise. It is known that deeper architectures require special connections, such as Skip connections, to prevent learning degradation. In particular, for deeper CNNs, it is found that the initial layers amplify the topological signal, to prompt learning in the deeper layers (see ResNet and DenseNet in Chapter 4 and Chapter 6). This gives an insight into the depth-width dynamics. Further, this leads to topological complexity within depth, reflected by changes in the preceding layer. Insight into the activation function was limited, and further study is needed for a proper assessment, as non-linearity itself is not enough to demonstrate homeomorphism. On the other hand, Identity activation demonstrates a relatively lower topological transition between the layers. Furthermore, the empirical results in Chapter 8, based on the homological preservation of the generative manifold with respect to the data, support the task-driven nature of homological expressivity in Neural Networks (see Figure 9.3) and the importance of topological preservation as the minimum requirement for learning.

9.4 Scope and Methodological Considerations

The theoretical and empirical frameworks developed in this work provide a task-driven approach to interpreting neural network dynamics. However, the applicability and resolution of these topological measures are subject to specific structural constraints and data dependencies. The following points summarise the scope, advantages, and inherent limitations of the proposed methodology:

Source Agnostic Inference : The methods utilising topological analysis of the embedding space do not require explicit knowledge of the source manifold, making them particularly suitable for scenarios where the source space is latent or hidden. Conversely, the framework is inapplicable if the embedding space representations are inaccessible. Notably, theoretically, the homological capacity Φ_f remains a robust measure that relies solely on the downstream source.

Task Centric Generalisation : The defined homological capacity is based on the Task space and thus does not require assumptions regarding the distributions of the data or the activation functions. (see Figure 9.3) Empirical results in the study are limited to the image space and therefore require further investigation into other data structures. (It is to be noted that there do exist suitable complexes that extract homology for different data structures, therefore, the concepts are transferable to different paradigms.)

Granularity of Analysis : Different learning/tasks require different topological satisfaction, such as here the classification model requires the reduction of Betti numbers. In contrast, the generative models require the preservation of the topological structures of the generated manifold. While the methods in Chapters 4 and 8 offer computational efficiency by bypassing the analysis of the entire embedding space, this reduction entails a loss of global topological information. Although theoretically justified for the specific tasks addressed, this remains a trade-off between computational tractability and topological resolution.

Depth Restriction : As established in Chapter 4, the capacity metric Φ_f identifies the minimal depth L required among candidate models to resolve the underlying homological complexity. In cases where

the architectures do not fully resolve the classification task, the results highlight the inherent bias of Φ_f towards smaller values of L (see Chapter 7). To accommodate such cases, the selection criterion prioritising architectural efficiency is used to derive the permissible depth relative to the least non-optimal solution. This opens the need for a better method than Φ_f for quantifying homological capacity.

Theoretical Synthesis : A significant direction for future research involves bridging the gap between topological and statistical learning theories. Specifically, establishing a formal relationship between homological capacity Φ_f and classical complexity measures, such as VC dimension [31] or Rademacher complexity [33], remains an open challenge. For a classical setting, homological capacity can be defined with $L = 1$. Such an alignment would offer a more rigorous theoretical foundation for using persistent homology as a diagnostic tool for the expressive power in machine learning.

Ultimately, while this work demonstrates that the embedding space contains critical information regarding the homological expressivity of models, the discretisation of layers and the reliance on polynomial approximations highlight a trade-off between computational tractability and topological resolution. The proposed framework establishes a foundational link between architectural depth and topological transitions, providing a systematic, task-driven approach to model selection diagnosis that moves beyond purely empirical performance metrics. This research prompts future investigations that could transcend depth-driven expressivity, shifting toward establishing rigorous bounds on the depth required for a given task while simultaneously minimising the parameter space.

9.5 Theoretical Synthesis and Final Remarks

The geometric exploitation of the archetypal subspace provides a novel approach to the dimensionality-reduction bottleneck in persistent homology, enabling the empirical study of the homological expressivity of the embedding space of neural networks. This work demonstrates that the homological features extracted from the dataset can be employed to study the expressivity of a neural network architecture. Notably, the topological shape of a dataset can be extracted agnostically, independent of its specific statistical state when observed through a normalised manifold. This distinction is critical in contexts such as fine-tuning, where the source and downstream spaces are often assumed to share a probability distribution, yet their underlying topologies may differ significantly.

Empirical results in Chapter 7 utilise the inter-class topological links to demonstrate that the decision boundary is shaped by complex structural properties beyond simple Betti numbers. This suggests that the influence of homological structures on the decision boundary is unique, task-specific, and varies in priority. Thus, we need to define a stronger topological analysis that accounts for the intrinsic dimensionality of the topological structure of the data. Overall, this research confirms that it is possible to empirically study the transition of the shape of the embedding space using methods such as archetypal subsampling through a topological lens.

While this study connects architectural depth and topological transitions at the basic empirical level, future work must extend beyond the image domain to complex data structures, such as text and sequential models, as well as to emerging generative architectures. By formalising the relationship between homological capacity and network depth, this work moves beyond the descriptive analysis of expressivity toward a prescriptive theory for bounding the architectural complexity required to resolve the underlying manifold of a given task. This research establishes a foundational framework that invites a deeper synthesis of geometry, computational topology, and statistical learning theory.

Bibliography

[1] Leijnen, S.; van Veen, F. The Neural Network Zoo. *Proceedings, 47(1):9* **2020**,

[2] Skorski, M.; Temperoni, A.; Theobald, M. Revisiting Weight Initialization of Deep Neural Networks. Proceedings of The 13th Asian Conference on Machine Learning. 2021; pp 1192–1207.

[3] Bianchini, M.; Scarselli, F. On the Complexity of Neural Network Classifiers: A Comparison Between Shallow and Deep Architectures. *IEEE Transactions on Neural Networks and Learning Systems* **2014**, *25*, 1553–1565.

[4] Bredon, G. E. *Topology and Geometry*, softcover repr. of the hardcover 1st ed. 1993 ed.; Graduate Texts in Mathematics; Springer: New York, NY, 2010.

[5] Dey, T. K.; Wang, Y. *Computational Topology for Data Analysis*; Cambridge University Press, 2022.

[6] Shwartz-Ziv, R.; Tishby, N. Opening the Black Box of Deep Neural Networks via Information. *CoRR* **2017**, *abs/1703.00810*.

[7] Rieck, B.; Togninalli, M.; Bock, C.; Moor, M.; Horn, M.; Gumbsch, T.; Borgwardt, K. Neural Persistence: A Complexity Measure for Deep Neural Networks Using Algebraic Topology. International Conference on Learning Representations (ICLR). 2019.

[8] Brüel Gabrielsson, R.; Carlsson, G. Exposition and Interpretation of the Topology of Neural Networks. 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). 2019; pp 1069–1076.

[9] LeCun, Y.; Kavukcuoglu, K.; Farabet, C. Convolutional networks and applications in vision. Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on. 2010; pp 253–256.

[10] Apostol, T. *Calculus: One-variable calculus, with an introduction to linear algebra*; Blaisdell book in pure and applied mathematics; Blaisdell Publishing Company, 1967.

[11] James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning: with Applications in R*; Springer, 2013.

[12] Strang, G. *Linear algebra and its applications*, 4th ed.; Thomson, Brooks/Cole: Belmont, CA, 2007.

[13] *Handbook of discrete and computational geometry*, third edition ed.; Discrete mathematics and its applications; CRC Press: Boca Raton London New York, 2018.

[14] Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*; Springer-Verlag: Berlin, Heidelberg, 2006.

[15] Wasserman, L. *All of Statistics: A Concise Course in Statistical Inference*; Springer Texts in Statistics; Springer: New York, 2004.

[16] Wolpert, D. H. The lack of a priori distinctions between learning algorithms. *Neural Comput.* **1996**, *8*, 1341–1390.

[17] Duda, R. O.; Hart, P. E.; Stork, D. G. *Pattern Classification (2nd Edition)*; Wiley-Interscience: USA, 2000.

[18] Goodfellow, I.; Bengio, Y.; Courville, A. *Deep learning*; Adaptive computation and machine learning; The MIT Press: Cambridge, Massachusetts, 2016.

[19] Kavukcuoglu, K.; Sermanet, P.; Boureau, Y.-L.; Gregor, K.; Mathieu, M.; LeCun, Y. Learning convolutional feature hierarchies for visual recognition. Proceedings of the 24th International Conference on Neural Information Processing Systems - Volume 1. Red Hook, NY, USA, 2010; p 1090–1098.

[20] LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551.

[21] Paszke, A. *et al. Proceedings of the 33rd International Conference on Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2019.

[22] Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* **2014**, *abs/1409.1556*.

[23] He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). 2016; pp 770–778.

[24] Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K. Q. Densely Connected Convolutional Networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017; pp 2261–2269.

[25] Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*; 2017.

[26] Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q. V. MnasNet: Platform-Aware Neural Architecture Search for Mobile. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019; pp 2815–2823.

[27] Saxe, A. M.; McClelland, J. L.; Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings. 2014.

[28] Mitchell, T. M. *Machine learning*, nachdr. ed.; McGraw-Hill series in Computer Science; McGraw-Hill: New York, 2013.

[29] Guss, W. H.; Salakhutdinov, R. On Characterizing the Capacity of Neural Networks using Algebraic Topology. *ArXiv* **2018**, *abs/1802.04443*.

[30] Boser, B. E.; Guyon, I. M.; Vapnik, V. N. A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory. Pittsburgh Pennsylvania USA, 1992; p 144–152.

[31] Vapnik, V. N.; Chervonenkis, A. Y. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. *Theory of Probability and its Applications* **1971**, *16*, 264–280.

[32] Mpoudeu, M.; Clarke, B. Model Selection via the VC Dimension. *Journal of Machine Learning Research* **2019**, *20*, 1–26.

[33] Bartlett, P. L.; Mendelson, S. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. Computational Learning Theory. Berlin, Heidelberg, 2001; pp 224–240.

[34] Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems* **1989**, *2*, 303–314.

[35] Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Networks* **1989**, *2*, 359–366.

[36] Poole, B.; Lahiri, S.; Raghu, M.; Sohl-Dickstein, J.; Ganguli, S. Exponential expressivity in deep neural networks through transient chaos. Advances in Neural Information Processing Systems. 2016.

[37] Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal* **1948**, *27*, 379–423.

[38] Tishby, N.; Pereira, F. C.; Bialek, W. The information bottleneck method. *ArXiv* **2000**, *physics/0004057*.

[39] Butakov, I.; Tolmachev, A.; Malanchuk, S.; Neopryatnaya, A.; Frolov, A.; Andreev, K. Information Bottleneck Analysis of Deep Neural Networks via Lossy Compression. The Twelfth International Conference on Learning Representations. 2024.

[40] Naitzat, G.; Zhitnikov, A.; Lim, L.-H. Topology of Deep Neural Networks. *Journal of Machine Learning Research* **2020**, *21*, 1–40.

[41] Mischaikow, K.; Mrozek, M.; Reiss, J.; Szymczak, A. Construction of Symbolic Dynamics from Experimental Time Series. *Phys. Rev. Lett.* **1999**, *82*, 1144–1147.

[42] Conley, C. C. *Isolated invariant sets and the Morse index*; Regional Conference Series in Mathematics; Published for the Conference Board of the Mathematical Sciences by the American Mathematical Society: Providence, Rhode Island, 1978.

[43] Vapnik, V.; Levin, E.; Cun, Y. L. Measuring the VC-Dimension of a Learning Machine. *Neural Computation* **1994**, *6*, 851–876.

[44] Bu, K.; Zhang, Y.; Luo, Q. Depth-Width Trade-offs for Neural Networks via Topological Entropy. 2020; `https://arxiv.org/abs/2010.07587`.

[45] Chatziafratis, V.; Nagarajan, S. G.; Panageas, I. Better depth-width trade-offs for neural networks through the lens of dynamical systems. Proceedings of the 37th International Conference on Machine Learning. 2020.

[46] Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* **1958**, *65*, 386–408.

[47] Kiani, B. T.; Wang, J.; Weber, M. Hardness of learning neural networks under the manifold hypothesis. Proceedings of the 38th International Conference on Neural Information Processing Systems. Red Hook, NY, USA, 2025.

[48] Hu, X.; Fuxin, L.; Samaras, D.; Chen, C. *Proceedings of the 33rd International Conference on Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2019.

[49] Coates, A.; Ng, A.; Lee, H. An Analysis of Single Layer Networks in Unsupervised Feature Learning. AISTATS. 2011; `https://cs.stanford.edu/~acoates/papers/coatesleeng_aistats_2011.pdf`.

[50] Edelsbrunner, H.; Harer, J. *Computational Topology - an Introduction.*; American Mathematical Society, 2010.

[51] Cohen-Steiner, D.; Edelsbrunner, H.; Harer, J. Stability of persistence diagrams. Annual Symposium on Computational Geometry. New York, NY, USA, 2005; p 263–271.

[52] Wagner, H.; Chen, C.; Vuçini, E. In *Topological Methods in Data Analysis and Visualization II: Theory, Algorithms, and Applications*; Peikert, R., Hauser, H., Carr, H., Fuchs, R., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2012; pp 91–106.

[53] Malott, N. O.; Chen, S.; Wilsey, P. A. A Survey on the High-Performance Computation of Persistent Homology. *IEEE Transactions on Knowledge and Data Engineering* **2023**, *35*, 4466–4484.

[54] McInnes, L.; Healy, J.; Saul, N.; Grossberger, L. UMAP: Uniform Manifold Approximation and Projection. *The Journal of Open Source Software* **2018**, *3*, 861.

[55] Kohonen, T. The self-organizing map. *Proceedings of the IEEE* **1990**, *78*, 1464–1480.

[56] Moor, M.; Horn, M.; Rieck, B.; Borgwardt, K. Topological autoencoders. Proceedings of the 37th International Conference on Machine Learning. 2020.

[57] De Silva, V.; Carlsson, G. Topological estimation using witness complexes. Proceedings of the First Eurographics Conference on Point-Based Graphics. Goslar, DEU, 2004; p 157–166.

[58] Scaramuccia, S.; Iuricich, F.; De Floriani, L.; Landi, C. Computing multiparameter persistent homology through a discrete Morse-based approach. *Computational Geometry* **2020**, *89*, 101623.

[59] Bubenik, P. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.* **2015**, *16*, 77–102.

[60] Singh, G.; Memoli, F.; Carlsson, G. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. Eurographics Symposium on Point-Based Graphics. 2007.

[61] Bauer, U. Ripser: efficient computation of Vietoris-Rips persistence barcodes. *J. Appl. Comput. Topol.* **2021**, *5*, 391–423.

[62] Tauzin, G.; Lupo, U.; Tunstall, L.; Pérez, J. B.; Caorsi, M.; Medina-Mardones, A. M.; Dassatti, A.; Hess, K. giotto-tda: A Topological Data Analysis Toolkit for Machine Learning and Data Exploration. *Journal of Machine Learning Research* **2021**, *22*, 1–6.

[63] The GUDHI Project, *GUDHI User and Reference Manual*; GUDHI Editorial Board, 2015.

[64] Kaji, S.; Sudo, T.; Ahara, K. Cubical Ripser: Software for computing persistent homology of image and volume data. *CoRR* **2020**, *abs/2005.12692*.

[65] Shwartz-Ziv, R.; Tishby, N. Opening the Black Box of Deep Neural Networks via Information. 2017; `https://arxiv.org/abs/1703.00810`.

[66] Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. 2009; pp 248–255.

[67] You, K.; Liu, Y.; Wang, J.; Long, M. LogME: Practical Assessment of Pre-trained Models for Transfer Learning. ICML. 2021.

[68] Suryaka, S.; Vinayak, A.; Anshul, T. Pitfalls in Measuring Neural Transferability. Proceedings of the 2nd NeurIPS Workshop on Symmetry and Geometry in Neural Representations. 2024; pp 279–291.

[69] Nguyen, C. V.; Hassner, T.; Seeger, M.; Archambeau, C. LEEP: a new measure to evaluate transferability of learned representations. Proceedings of the 37th International Conference on Machine Learning. 2020.

[70] Gutmann, M. U.; Hyvärinen, A. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *J. Mach. Learn. Res.* **2012**, *13*, 307–361.

[71] Papyan, V.; Han, X.; Donoho, D. L. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences of the United States of America* **2020**, *117*, 24652 – 24663.

[72] Naitzat, G.; Zhitnikov, A.; Lim, L.-H. Topology of deep neural networks. *J. Mach. Learn. Res.* **2020**, *21*.

[73] Raghu, M.; Poole, B.; Kleinberg, J.; Ganguli, S.; Sohl-Dickstein, J. On the Expressive Power of Deep Neural Networks. Proceedings of the 34th International Conference on Machine Learning. 2017; pp 2847–2854.

[74] Krizhevsky, A. *Learning multiple layers of features from tiny images*; 2009.

[75] Maji, S.; Kannala, J.; Rahtu, E.; Blaschko, M.; Vedaldi, A. *Fine-Grained Visual Classification of Aircraft*; 2013.

[76] Lowe, D. G. Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision* **1999**, *2*, 1150–1157 vol.2.

[77] Hiraoka, Y.; Tsunoda, K. Limit Theorems for Random Cubical Homology. *Discrete & Computational Geometry* **2018**, *60*, 665–687.

[78] Spearman, C. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology* **1904**, *15*, 72.

[79] Kendall, M. G. A New Measure of Rank Correlation. *Biometrika* **1938**, *30*, 81.

[80] Cutler, A.; Breiman, L. Archetypal analysis. *Technometrics* **1994**, *36*, 338–347.

[81] Abrol, V.; Sharma, P. A Geometric Approach to Archetypal Analysis via Sparse Projections. International Conference on Machine Learning (ICML). 2020; pp 42–51.

[82] Chen, Y.; Mairal, J.; Harchaoui, Z. Fast and robust archetypal analysis for representation learning. Proceedings of Computer Vision and Pattern Recognition (CVPR). 2014; pp 1478–1485.

[83] Li, H.; Xu, Z.; Taylor, G.; Studer, C.; Goldstein, T. Visualizing the loss landscape of neural nets. Proceedings of the 32nd International Conference on Neural Information Processing Systems. Red Hook, NY, USA, 2018; p 6391–6401.

[84] Thompson, A. In *Encyclopedia of Physical Science and Technology (Third Edition)*, third edition ed.; Meyers, R. A., Ed.; Academic Press: New York, 2003; pp 717–737.

[85] Osting, B.; Wang, D.; Xu, Y.; Zosso, D. Consistency of Archetypal Analysis. *SIAM Journal on Mathematics of Data Science* **2021**, *3*, 1–30.

[86] Javadi, H. H. S.; Montanari, A. Nonnegative Matrix Factorization Via Archetypal Analysis. *Journal of the American Statistical Association* **2017**, *115*, 896 – 907.

[87] Mørup, M.; Hansen, L. K. Archetypal analysis for machine learning and data mining. *Neurocomputing* **2012**, *80*, 54–63, Special Issue on Machine Learning for Signal Processing.

[88] Johnson, K. Multiplicativity of the Euler Characteristic. PhD thesis, Tulane University, 2015.

[89] Bauer, U. Ripser: efficient computation of Vietoris-Rips persistence barcodes. *Springer Journal of Applied and Computational Topology* **2021**,

[90] Atienza, N.; Gonzalez-Diaz, R.; Rucco, M. Persistent entropy for separating topological features from noise in vietoris-rips complexes. *Journal of Intelligent Information Systems* **2019**, *52*, 637–655.

[91] Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X.; Chen, X. Improved Techniques for Training GANs. Advances in Neural Information Processing Systems. 2016.

[92] Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. Advances in Neural Information Processing Systems. 2017.

[93] Wang, F.; Liu, H.; Samaras, D.; Chen, C. TopoGAN: A Topology-Aware Generative Adversarial Network. European Conference in Computer Vision (ECCV). 2020; pp 118–136.

[94] Barannikov, S.; Trofimov, I.; Sotnikov, G.; Trimbach, E.; Korotin, A.; Filippov, A.; Burnaev, E. Manifold Topology Divergence: a Framework for Comparing Data Manifolds. Advances in Neural Information Processing Systems (NeurIPS). 2021; pp 7294–7305.

[95] Khrulkov, V.; Oseledets, I. Geometry Score: A Method For Comparing Generative Adversarial Networks. Proceedings of the 35th International Conference on Machine Learning. 2018; pp 2621–2629.

[96] Barannikov, S.; Trofimov, I.; Balabin, N.; Burnaev, E. Representation Topology Divergence: A Method for Comparing Neural Network Representations. International Conference on Machine Learning (ICML). 2022; pp 1607–1626.

[97] Atienza, N.; Gonzalez-Díaz, R.; Soriano-Trigueros, M. On the stability of persistent entropy and new summary functions for topological data analysis. *Pattern Recognition* **2020**, *107*, 107509.

[98] Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings. 2016.

[99] Berthelot, D.; Schumm, T.; Metz, L. BEGAN: Boundary Equilibrium Generative Adversarial Networks. *CoRR* **2017**, *abs/1703.10717*.

[100] Mescheder, L.; Geiger, A.; Nowozin, S. Which Training Methods for GANs do actually Converge? 2018; `https://arxiv.org/abs/1801.04406`.

[101] Epps, T. W.; Singleton, K. J. An omnibus test for the two-sample problem using the empirical characteristic function. *Journal of Statistical Computation and Simulation* **1986**, *26*, 177–203.

[102] Goerg, S. J.; Kaiser, J. Nonparametric Testing of Distributions—the Epps–Singleton Two-Sample Test using the Empirical Characteristic Function. *The Stata Journal* **2009**, *9*, 454 – 465.

[103] Suresh, S.; Abrol, V. Evaluating Generative Models via Cubical Homology based Persistent Entropy. Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2. New York, NY, USA, 2025; p 2768–2777.

[104] von Rohrscheidt, J.; Rieck, B. Diss-l-ECT: Dissecting Graph Data with Local Euler Characteristic Transforms. Proceedings of the 42nd International Conference on Machine Learning. 2025; pp 61790–61809.