KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Computer Science and Engineering

Second cycle, 30 credits

Topological regularization and relative latent representations

ALEJANDRO GARCÍA CASTELLANOS

Stockholm, Sweden, 2023

Topological regularization and relative latent representations

ALEJANDRO GARCÍA CASTELLANOS

Master's Programme, Machine Learning, 120 credits
Date: November 13, 2023

Supervisors: Martina Scolamiero, Giovanni Luca Marchetti
Examiner: Florian T. Pokorny
School of Electrical Engineering and Computer Science
Swedish title: Topologisk regularisering och relativa latenta representationer

Abstract

This Master's Thesis delves into the application of topological regularization techniques and relative latent representations within the realm of zero-shot model stitching. Building upon the prior work of Moschella et al. (2022) that introduces relative latent representations to enhance the similarities between latent spaces of different models, we incorporate the approach of Hofer et al. (2021), which combines Topological Data Analysis (TDA) and Machine Learning techniques for topological densification of class distributions in the latent space.

The main research objective is to investigate the impact of topological regularization on zero-shot stitching performance when employing relative latent representations. Theoretical foundations for the relative transformation are established based on the intertwiner groups of activation functions. Empirical analyses are conducted to validate the assumptions underlying the construction of the relative transformation in the latent space. Moreover, experiments are performed on a Large Language Model trained on multilingual Amazon Reviews datasets to evaluate the effectiveness of zero-shot stitching while using the topological densification technique and the relative transformation.

The findings indicate that the proposed methodologies can enhance the performance of multilingual model stitching. Specifically, enforcing the relative transformation to preserve the H_0 homology death times distributions proves beneficial. Additionally, the presence of similar topological features plays a crucial role in achieving higher model compatibility. However, a more in-depth exploration of the geometric properties of the post-relative transformation latent space is necessary to further improve the topological densification technique.

Overall, this work contributes to the emerging field of Topological Machine Learning and provides valuable insights for researchers in transfer learning and representation learning domains.

Keywords

Algebraic Topology, Large Language Models, Relative Representation, Representation Learning, Model Stitching, Topological Data Analysis, Zero-shot

Sammanfattning

Denna masteruppsats undersöker tillämpningen av topologiska regleringstekniker och relativa latenta representationer inom området för zero-shot model stitching. Genom att bygga vidare på tidigare arbete av Moschella et al. (2022), som introducerade relativa latenta representationer för att förbättra likheterna mellan latenta rummet hos olika modeller, inkorporerar vi tillvägagångssättet av Hofer et al. (2021), som kombinerar topologisk dataanalys (TDA) och maskininlärningstekniker för topologisk "förtätning" av klassfördelningar i det latenta utrymmet.

Den huvudsakliga forskningsuppgiften är att undersöka effekten av topologisk reglering på zero-shot model stitching-prestanda när man använder relativa latenta representationer. Teoretiska grunder för den relativa transformationen etableras baserat på intertwinergrupperna för aktiveringsfunktioner. Empiriska analyser genomförs för att validera antagandena som ligger till grund för konstruktionen av den relativa transformationen i det latenta rummen. Dessutom utförs experiment på en stor språkmodell tränad på multilinguella Amazon Reviews-dataset för att utvärdera effektiviteten hos zero-shot model stitching med Hofer's topologiska reglering och relativa transformation.

Resultaten visar att de föreslagna metoderna kan förbättra prestationen hos zero-shot model stitching för flerspråkiga modeller. Specifikt är det fördelaktigt att tvinga den relativa transformationen att bevara H_0 homologins dödstidsfördelningar. Dessutom spelar närvaron av liknande topologiska egenskaper en avgörande roll för att uppnå högre modellkompatibilitet. Dock krävs en mer ingående utforskning av de geometriska egenskaperna hos det latenta utrymmet efter den relativa transformationen för att ytterligare förbättra Hofer's topologiska reglering.

Sammanfattningsvis bidrar detta arbete till det framväxande området Topologisk Maskininlärning och ger värdefulla insikter för forskare inom "transfer-inlärning" och representationsinlärningsdomäner.

Nyckelord

Algebraisk topologi, Stora språkmodeller, Relativ representation, Representationsinlärning, Modell sömmar, Topologisk dataanalys, Zero-shot

Acknowledgments

I would like to take a moment to express my heartfelt gratitude to all those who have played a pivotal role in shaping this work, representing the culmination of my academic endeavors thus far. Their unwavering support and guidance have been instrumental in this journey, and it is with sincere appreciation that I acknowledge their contributions.

First and foremost, I extend my deepest thanks to my supervisor, Martina Scolamiero, for her consistent support and guidance since the beginning. Martina has always been incredibly supportive, allowing me the freedom to pursue topics I am passionate about. Her valuable directions have shaped this Thesis into what it is today. I am truly grateful for the trust and support she has shown me throughout these months.

I am also incredibly grateful to my co-supervisor, Giovanni Luca Marchetti, whose infectious enthusiasm and invaluable ideas have pushed this project beyond my expectations. Giovanni's constant availability and support during challenging times have been invaluable. I deeply admire his passion, which permeates every aspect of his work. Collaborating with him has been an absolute privilege, and I am determined to approach future research endeavors with the same level of dedication he exemplifies.

Furthermore, I would like to express my sincere gratitude to Luca Moschella, the first author of the relative transformation paper, for generously providing access to the project's materials and clarifying any uncertainties I had about the methodologies involved.

In addition, I wish to extend my heartfelt appreciation to Hector Barge for sparking my fascination with topology. During the most challenging moments of the pandemic lockdown, I found solace in the fascinating world he unveiled in each lecture.

Lastly, I am forever indebted to my family for their unwavering belief in me and their constant encouragement. Their love and support have been the bedrock upon which I have built my academic career. To my dear friends, I am grateful for your enduring presence and patience as you listened to my endless monologues about the usefulness of counting holes in a 768-dimensional space. Your presence in my life has been a constant source of inspiration and

joy.

In conclusion, I would like to emphasize that the successful completion of this work would not have been possible without the unwavering support, guidance, and contributions of these exceptional individuals. Their impact on my academic journey is indelible, and for that, I offer my most profound appreciation.

Stockholm, November 2023
Alejandro García Castellanos

Contents

List of Figures

List of Tables

Listings

List of acronyms and abbreviations

Acc	Accuracy
DL	Deep Learning
MAE	Mean Absolute Error
ML	Machine Learning
NN	Neural Network
TDA	Topological Data Analysis
TopoML	Topological Machine Learning

Chapter 1

Introduction

1.1 Background

Multiple Machine Learning (ML) techniques rely on the manifold hypothesis [1], which claims that high-dimensional data are sampled from a lower-dimensional manifold. One approach to leveraging this hypothesis is through the lens of computational topology, specifically Topological Data Analysis (TDA). TDA enables us to explore qualitative geometric properties of our datasets, such as identifying topological features (e.g., connected components, holes, and cavities) of the underlying manifold. Additionally, TDA can help us assess the relevance of observed features when additional complexity is introduced due to measurement or discretization issues.

Furthermore, as shown in [2], there is a growing interest in integrating TDA techniques into Deep Learning (DL) approaches, forming a new field referred to as Topological Machine Learning (TopoML). The survey showcases three distinct ways TDA tools are utilized in DL: incorporating topological feature extraction as a layer in a Neural Network (NN) (either as an input or a hidden layer), imposing specific properties based on obtained topological information, and analyzing the topological properties of data and model architecture to assess specific characteristics of the trained model.

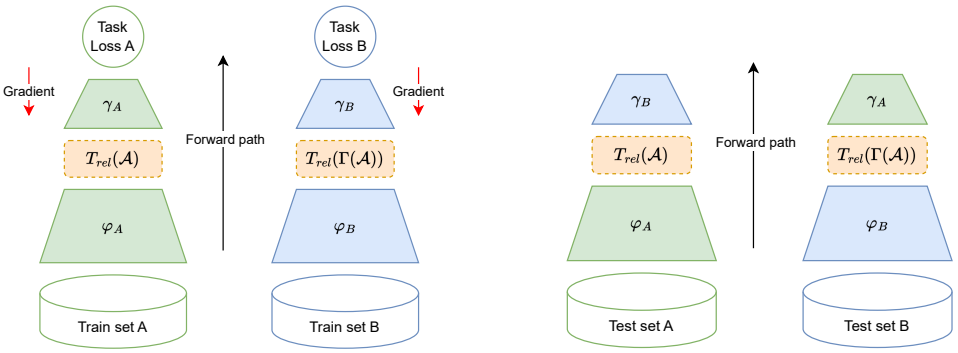
In our case, we will primarily focus on the second option — imposing certain properties through what we term *"topological regularization"*. Specifically, we will expand upon the work proposed by Hofer *et al.,* [3], where they demonstrate that imposing *"topological densification"* of each class distribution can enhance generalization capabilities. The main concept

is to condense the density function of each class in the latent space, improving the likelihood of having each class distribution contained within its decision region. The authors of the paper observed that this property could be enforced by using an algebraic topology construction called Persistent Homology.

One of the main fields that have studied the properties of latent spaces of ML models is Representation Learning. Furthermore, one of the main topics of interest in this field is to analyze the similarities of latent representation between networks. It has been shown in [4] that by using *"relative latent representations,"* we can increase the similarities between latent spaces. The relative latent representation is obtained as follows: let $S \subset \mathcal{X}$ a dataset, $\varphi : \mathcal{X} \to \mathcal{Z}$ the feature extractor component of your network, and $\mathcal{A} = \{a_1, ..., a_k\} \subset \mathcal{X}$ a set of points called *anchors*. Then for any similarity function sim we define the relative representation of a point $x \in S$ w.r.t. $\mathcal{A}$ as

$$(sim(\varphi(x), \varphi(a_1)), ..., sim(\varphi(x), \varphi(a_k)) \in \mathbb{R}^k.$$

Furthermore, the authors demonstrated that this technique enables zero-shot stitching, meaning that models with different architectures, datasets, or seeds can be combined without additional fine-tuning, as illustrated in Figure 1.1.



(a) Train w/ relative transformations. (b) Model stitching

Figure 1.1: Cross-domain model stitching training and testing

For further mathematical background and detailed explanations of the described methods, please refer to Chapter 2.

1.2 Problem

The main objective of this project is to try to expand the work proposed on [3] into the setup proposed on [4], so we can have a better understanding of how these intrinsic representation regularization techniques interact with novel latent space representations.

1.2.1 Research question

Therefore our main research question is: *To what extent does the application of the topological densification technique [3] impact the classification performance of zero-shot stitching when utilizing relative latent representations [4]?*

1.3 Purpose

One of the purposes of this project is to showcase a brief introduction to TDA so that anyone with a ML background can comprehend the main philosophy and methodology proposed in this novel field of Topological Machine Learning. Additionally, this work will be valuable for researchers interested in transfer learning and representation learning as it explores the significance of latent space topology in the context of zero-shot stitching.

1.4 Goals

Considering that the chosen topological regularization technique [3] requires a supervised classification setup, our focus will be on studying the zero-shot stitching of encoders and decoders that are both fine-tuned with a relative representation. This differs from the approach presented in [4], where only the decoder is fine-tuned using the relative representation while the encoder remains frozen. Consequently, our main goals are as follows:

- Analyze the performance of zero-shot stitching after fine-tuning both the encoder and decoder with relative representation.
- Evaluate the performance when training with topological regularization and the relative representation. This entails the following tasks:
  - Assess the accuracy of zero-shot stitching in this setup.

- Assess whether we obtain better accuracy of the zero-shot stitching if we apply the topological regularization before or after the relative representation transformation.
- Assess the relevance of the used distance metric for the topological regularization.

Lastly, the construction of the relative representation is primarily based on empirical evidence suggesting that, in certain cases, we can achieve latent representations that are nearly isometric [4]. Therefore, an additional objective of this project is to provide a theoretical explanation for this claim and obtain new empirical results to support it. By accomplishing this, we can establish the theoretical validity of the relative transformation and gain valuable insights into the latent space of our network.

1.5 Research Methodology

We will now provide a concise overview of the methodology employed to achieve the objectives outlined in this project. A more in-depth explanation of the methods can be found in Chapter 3.

Firstly, we will enhance the theoretical foundation of the relative representation by leveraging specific characteristics of activation functions. Additionally, we will conduct new empirical analyses to examine the similarities within the latent space across different initializations. This will involve both numerical and visual approaches. For numerical analysis, we will utilize standard metrics such as CKA while also introducing a new metric based on the Frobenius norm of the distance matrix. For visual analysis, Procrustes analysis will be employed to visually illustrate the optimal alignment of two latent spaces. These experiments will be conducted on an autoencoder and a CNN trained on the CIFAR10 dataset.

Upon validating the construction of the relative representation through these initial experiments, we will proceed to replicate selected experiments from [4], adapting them to our new setups. Specifically, we will replicate the Amazon review classification experiment using cross-lingual stitching. Throughout these experiments, we will compare different techniques for combining the relative transformation with the topological densification. Furthermore, we will explore the potential advantages of using alternative metrics for constructing the Vietoris-Rips filtrations.

1.6 Delimitations

Our primary objective is to evaluate the impact of combining the previously defined methods through ablation studies rather than analyzing their performance under ideal conditions. Considering this objective and taking into account our time and computational limitations, we will not extensively tune hyperparameters as long as we can draw meaningful conclusions.

Furthermore, due to our computational constraints, we acknowledge that studying zero-shot stitching across different architectures and seeds is beyond the scope of this project. Additionally, to accommodate our computational limitations and maximize the number of tasks we can complete, we will utilize a 1% sub-sample of the original training dataset instead of the 25% sub-sampling approach employed in the original paper [4].

1.7 Structure of the thesis

The structure of this Thesis is organized as follows:

- Chapter 2 provides the essential background information, covering mathematical concepts such as simplicial complexes, homology, and persistence. It also explores related work in the field of Topological Machine Learning, representation similarity, and model stitching.
- Chapter 3 presents the research methodology, starting with a clear problem formulation. It then describes the conducted experiments and provides methodological justifications for their design.
- Chapter 4 presents the results obtained from the experiments and provides a thorough analysis of these findings.
- Chapter 5 engages in further analysis, offering additional insights, opinions, and hypotheses concerning the methodology and the results obtained in this project.
- Chapter 6 concludes the thesis by summarizing the key conclusions derived from the research. It acknowledges the limitations of the study and proposes future avenues for further investigation. This chapter also includes reflections regarding the environmental and socioeconomic implications of this work.

Chapter 2

Background

2.1 Mathematical background

This section provides the necessary mathematical background to comprehend the methods proposed in [3, 4] and the ones described in Chapter 3. The exposition is mainly based on [5].

2.1.1 Simplicial Complexes

Geometric simplicial complexes

As discussed in the introduction, our objective is to analyze the characteristics of the underlying manifold from which our dataset has been sampled. To achieve this, we aim to introduce structure to our dataset that allows us to assess the desired topological properties.

One approach to represent topological spaces is through decomposition into simpler pieces. A decomposition is referred to as a complex if its pieces are topologically simpler and their intersections are of the same type, but lower dimensional [5]. There exists a wide variety of complexes with different levels of abstraction. However, our focus will be on simplicial complexes, which can represent most spaces encountered in data science and are particularly advantageous from a computational standpoint.

Simplicial complexes can be studied from both geometric and combinatorial perspectives. In this subsection, we will examine their definition and main properties from a geometric viewpoint. To this end, it will be helpful to review the following concepts from affine geometry.

Definition 2.1.1. A subset of points $\{u_0, u_1, ..., u_k\} \subseteq \mathbb{R}^d$ is *affinely independent* if the vectors $\{\overrightarrow{u_0u_1}, ..., \overrightarrow{u_0u_k}\}$ are linearly independent.

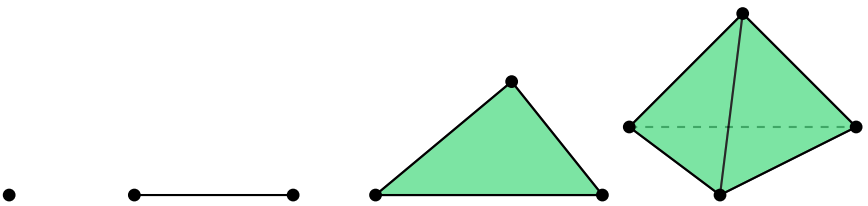
Definition 2.1.2. Given $\{u_0, u_1, ..., u_k\} \subseteq \mathbb{R}^d$, we say that $x \in \mathbb{R}^d$ is a *convex combination* of these points if $x = \sum_{i=0}^{k} \lambda_i u_i$, where $\lambda_i \geq 0$ for all $i \in 0, ..., k$ and $\sum_{i=0}^{k} \lambda_i = 1$.

Definition 2.1.3. The *convex hull* of $u_0, u_1, ..., u_k$, denoted by $\text{conv}\{u_0, u_1, ..., u_k\}$, is the set of all convex combinations of the given points.

Using these concepts, we can define the building blocks of our decomposition as follows:

Definition 2.1.4. A *k-simplex* σ in $\mathbb{R}^d$ with $d \geq k$ is the convex hull of $k+1$ affinely independent points $u_0, u_1, ..., u_k \in \mathbb{R}^d$, i.e., $\sigma \equiv \text{conv}\{u_0, u_1, ..., u_k\}$.

We say that the k-simplex σ has dimension k, and the points $u_0, u_1, ..., u_k$ are referred to as the *vertices of* σ.



(a) 0-simplex (b) 1-simplex (c) 2-simplex (d) 3-simplex

Figure 2.1: Representation of 0, 1, 2, and 3-dimensional simplices

We can see that any subset of the vertices of σ will be affinely independent and will therefore define a lower dimensional simplex τ. Hence, we will say that τ *is a face of* σ if it is a convex combination of a non-empty subset of the vertices of σ, and we will denote it by $\tau \leq \sigma$. If the subset is proper, we will say that τ *is the proper face of* σ, and we will denote it by $\tau < \sigma$.

Based on the concept of faces, we can establish the notions of the *boundary and interior* of a simplex σ.

Definition 2.1.5. Let a simplex σ. Then we define

- *boundary of* σ as

$$\text{bd}\,\sigma = \bigcup_{\tau < \sigma} \tau\,.$$

- *interior of* σ as

$$\text{int } \sigma = \sigma - \text{bd } \sigma \, .$$

Now that we have established the components of our decomposition, we need to understand how to combine them and explore the main properties of the resulting complexes.

As previously mentioned, for a decomposition to be considered a complex, its components must be topologically simple, and the intersections between them should yield lower-dimensional components of the same type. The most natural way to achieve this is by gluing simplices together using their faces.

Definition 2.1.6. A *simplicial complex* is a finite collection of simplices K that satisfy the following properties:

1. If $\sigma \in K$ and $\tau \leq \sigma$ then $\tau \in K$.
2. If $\sigma_0, \sigma_1 \in K$ and $(\sigma_0 \cap \sigma_1) \neq \emptyset$ then $\sigma_0 \cap \sigma_1 \leq \sigma_i$ for $i = 0, 1$.

We define the dimension of K as the maximum of its simplices dimensions.

We can see in Figure 2.2 an example of a simplicial complex, while Figure 2.3 shows an example of something that does not verify the abovementioned properties.

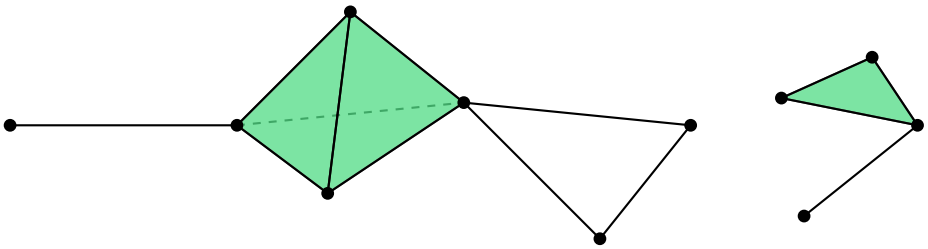


Figure 2.2: Example of a simplicial complex

Definition 2.1.7. The *underlying space* of a simplicial complex K, denoted $|K|$, is the union of the simplices in K with the induced topology of $\mathbb{R}^d$, where the simplex are contained. This underlying space is also called *polyhedron*.

As can be seen, the underlying space of a simplicial complex is compact since it is a finite union of simplices. The following result characterizes the open and closed sets of the underlying space $|K|$ of a simplicial complex K.

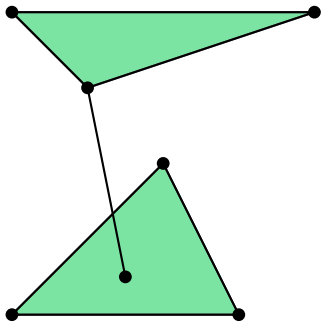


Figure 2.3: Example of simplices that does not verify the simplicial complex conditions

Proposition 2.1.1 ([5, Chapter 3])**.** *Let K be a simplicial complex and $A \subset |K|$ a subset. Then A is an open (closed) set in K if and only if for every $\sigma \in K$, $A \cap |\sigma|$ is an open (closed) set of $|\sigma|$.*

Definition 2.1.8. A *triangulation* of a topological space X is a pair (K, h) where K is a simplicial complex and $h : X \to |K|$ is a homeomorphism (i.e., h continuous, bijective and h^{-1} continuous).

We say that a topological space is *triangulable* if it admits triangulation.

It will also be helpful for us to be able to study the simplicial complexes contained in another simplicial complex.

Definition 2.1.9. A *subcomplex* L of a simplicial complex K is a simplicial complex $L \subseteq K$.

A particular type of subcomplex that is of great interest is the j*-skeletons*, defined as follows:

$$K^{(j)} = \{\sigma \in K \mid dim\, \sigma \ \leq j\}\,.$$

Abstract simplicial complexes

Now that we have established the concept of simplicial complexes from a geometric perspective, we will approach them from a combinatorial standpoint, which will significantly aid in the representation and manipulation of simplicial complexes.

Definition 2.1.10. An *abstract simplicial complex* A is a finite collection of finite sets such that if $\alpha \in A$ and $\beta \subset \alpha$, then $\beta \in A$.

In this way, it is fulfilled that

- Non-empty sets in A are called *abstract simplices*.
- The *dimension* of an abstract simplex $\alpha \in A$ is $\dim \alpha = \text{card}(\alpha) - 1$. And the dimension of the complex is the maximum of the dimensions of its simplices.
- A *face* of $\alpha \in A$ is any non-empty subset of $\beta \subset \alpha$.
- The *set of vertices* of A, denoted by $\text{Vert}\, A$, is the union of all its simplices.
- A *subcomplex* B of an abstract simplicial complex A is an abstract simplicial complex $B \subset A$.

Example 2.1.1. The following set forms an abstract simplicial complex:

$$A = \{\{0\},\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{0,1\},\{1,2\},\{1,3\},\{1,4\},\\ \{2,3\},\{2,4\},\{3,4\},\{4,5\},\{4,6\},\{5,6\},\\ \{1,2,3\},\{1,2,4\},\{1,3,4\},\{2,3,4\},\{1,2,3,4\}\}\,.$$

Where the set of vertices is: $\text{Vert}\, A = \{0,1,2,3,4,5,6\}$.

Definition 2.1.11. Let A and B be two abstract simplicial complexes. We say that A and B are *isomorphic* if there exists a bijection $b : \text{Vert}\, A \to \text{Vert}\, B$ such that $\alpha \in A$ if and only if $b(\alpha) \in B$.

Each geometric complex naturally induces an abstract complex as follows:

Definition 2.1.12. Let K be a geometric simplicial complex, and let V be the set of vertices of K. Then, we will call *vertex scheme* the abstract simplicial complex A formed by all those subsets of V that generate simplices in K.

Under certain circumstances, it is possible to construct a geometric simplicial complex from an abstract complex:

Definition 2.1.13. Let A be an abstract simplicial complex and K a simplicial complex. We will say that K is a *geometric realization* of A if A is isomorphic to the vertex scheme of K.

Theorem 2.1.1 ([5, Chapter 3])**.** *Every abstract simplicial complex of dimension d admits a geometric realization in $\mathbb{R}^{2d+1}$.*

Thus, abstract simplicial complexes provide a faithful representation of geometric simplicial complexes.

2.1.2 Simplicial complexes built from point clouds

From the computational point of view, we find ourselves with the problem that we have a representation of a topological space through a finite discretization, and our objective is to be able to recover properties of the original topological space from this cloud of points. Hence, to give some structure to our distance space (X, d), where X is the dataset and $d : X \to \overline{\mathbb{R}}_+$ is a distance function, we will create a simplicial complex with the data points as vertices, and encoding some of the relevant information of d.

Čech complex

The Čech complex is defined from the intersection of a collection of disks (closed balls). The idea underlying this construction is that of the nerve of a collection, which is introduced below.

Definition 2.1.14. Let F be a finite collection of sets. The *nerve* of F is defined as the abstract simplicial complex

$$\text{Nrv } F = \left\{ X \subseteq F \mid \bigcap_{x \in X} x \neq \emptyset \right\} .$$

One of the reasons we are interested in simplicial complexes constructed through the nerve of a collection is based on the implications of the following theorem.

Theorem 2.1.2 (Nerve theorem [5, Chapter 3])**.** *If F is a finite collection of closed and convex subsets in a Euclidean space, then the nerve of F has the same type of homotopy as the union of the sets of F.*

Therefore, according to this theorem, Čech complexes exhibit similar topological properties to subsets of $\mathbb{R}^d$ (as they are homotopy equivalent). This property ensures that our analysis of topological properties obtained through homology groups of the simplicial complex will be equivalent to studying them on neighborhoods of our dataset $X \subset \mathbb{R}^d$ [6].

Considering the case where the sets in the collection are disks (closed balls), denoted as $D_r(x) \equiv \overline{B_r}(x) = \{y \in \mathbb{R}^d \mid d(x, y) \leq r\}$ in $\mathbb{R}^d$, we define the Čech complex of a finite set of points $X \subset \mathbb{R}^d$ as follows:

Definition 2.1.15. Let $X \subset \mathbb{R}^d$ be a finite set of points. We will call *Čech complex* of X of radius r to the abstract simplicial complex

$$\check{\text{C}}\text{ech}(r) = \left\{ \sigma \subset X \mid \bigcap_{u \in \sigma} D_{r/2}(u) \neq \emptyset \right\} .$$

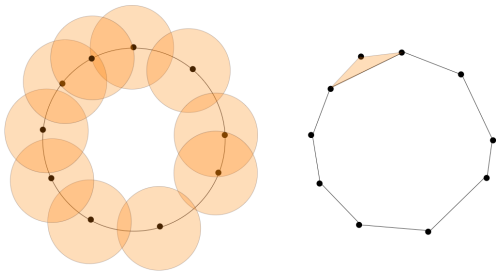


Figure 2.4: Čech complex for a set of nine points and a radius r. CC BY Source: [7]

We can check that for large enough values of r, $\check{\text{C}}\text{ech}(r)$ is a simplex of dimension $\text{card}(X)-1$ [5, Chapter 3], so the Čech complex is computationally inefficient.

Furthermore, in general, the Čech complex of a set of points $X \subset \mathbb{R}^d$ does not possess a geometric realization in $\mathbb{R}^d$. Therefore, we will present a construction that resembles the Čech complex while being more computationally favorable.

Vietoris-Rips complex

Let $\sigma \subset X$, then we recall that the diameter is defined as

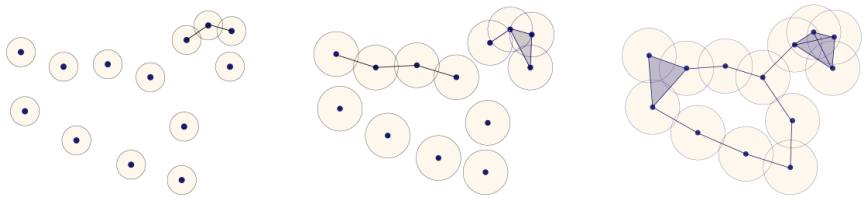
$$\text{diam } \sigma = \max_{u,v \in \sigma} d(u,v) .$$

Definition 2.1.16. Let $X \subset \mathbb{R}^d$ be a finite set of points. We call *Vietoris-Rips* complex of X of radius r to the abstract simplicial complex

$$\begin{aligned} \text{VR}(X,r) =& \{\sigma \subseteq X \mid \text{diam } \sigma \leq r\} \\ =& \{\{x_0, ..., x_n\} \subseteq X \mid d(x_i, x_j) \leq r \; \forall i,j\} . \end{aligned}$$

If the set X is understood from the context, then we can note it as $\text{VR}(r)$.

We can see in Figure 2.5 how the various VR complexes are generated as the radius increases.



(a) Rips complex at some scale (b) ..at a larger scale. (c) ...and at an even larger scale.

Figure 2.5: Vietoris-Rips complexes for a set of seven points as we increase the radius from left to right. CC BY Source: [8]

Note that $\sigma \in \mathrm{VR}(r)$ if and only if all its edges are in $\mathrm{VR}(r)$. In other words, the Vietoris-Rips complex is entirely determined by its 1-skeleton. This property makes the Vietoris-Rips complex computationally more efficient than the Čech complex. However, similar to the Čech complex, the Vietoris-Rips complex does not admit a geometric realization in $\mathbb{R}^d$.

On the other hand, the Vietoris-Rips complex is not the nerve of any collection of subsets of $\mathbb{R}^d$. Nevertheless, the following result guarantees that the VR complex provides an approximation of the Čech complex:

Lemma 2.1.3 (Vietoris-Rips lemma [5, Chapter 3])**.** *Let $X \subset \mathbb{R}^d$ be a finite set of points and let $r \geq 0$. Then,*

$$\check{\mathrm{C}}\mathrm{ech}(r) \subset \mathrm{VR}(r) \subset \check{\mathrm{C}}\mathrm{ech}(\sqrt{2}r)\,.$$

Another important property regarding VR complexes is their ability to encode ε-connectivity. In Section 2.2.1, we will talk more in detail about ε-connectivity and how it will be a crucial property for constructing our topological regularization.

Definition 2.1.17 ([9])**.** We say that a metric space X is *ε-disconnected* if there exist two subsets U and V with $U \cup V = X$, and

$$d(U, V) \equiv \inf_{x \in U,\, y \in V} d(x, y) > \varepsilon.$$

Hence if the space is not ε-disconnected, we will say that X is ε-connected. Moreover, a subset $A \subset X$ is an *ε-component* if A is ε-connected and $d(A, X \setminus A) > \varepsilon$. Hence, for any given ε, we can decompose our dataset X in disjoint ε-components.

However, to see the connection of VR complex with ε-connectivity, we will use another definition based on *ε-chains*.

Definition 2.1.18 ([9])**.** We call ε*-chain* to a finite sequence of points $x_0, .., x_n$ such as $d(x_i, x_{i+1}) \leq \varepsilon$ for $i = 1, ..., n$.

Therefore, for any set X, if we can form an ε*-chain* between any pair of points, then our set is ε-connected.

Proposition 2.1.2. *For any distance d on X, and $\varepsilon \geq 0$, the partition corresponding to the connected components of* VR(ε)*, denoted as* $\pi_0(\text{VR}(\varepsilon))$*, coincides with the ε-components of X.*

This result follows easily from the fact that two points u, v (vertices) are in the same connected component of a simplicial complex K if there exists a sequence of vertices $u = x_0, ..., x_n = v$ such as $\{x_i, x_{i+1}\} \in K$. So, we can see that in the case that K is a Vietoris-Rips complex of radius ε, then $\{x_i, x_{i+1}\} \in K$ iff $d(x_i, x_{i+1}) \leq \varepsilon$, i.e, there exist an ε-chain between u and v.

Witness complex

A common ML scenario is that we have a big dataset, and we apply our training procedure to small batches of that given set. Hence, ideally, we would want to be able to infer the topology of our whole dataset just from studying our reduced batches. Therefore, this type of ideas motivated Vin de Silva and Gunnar Carlsson when they introduced the Witness Complex in [10]: a subset of the data points, called the landmarks points, is used to construct the simplices "seen" by the witnesses, which are the rest of the points of the data set [11].

However, the construction of the simplicial complex that we will use differs from the ones used in standard TDA libraries, such as GUDHI, when defining Witness complexes [12]. We will focus on a specific type of complex presented in [10], which can be seen as an example of a Dowker complex*.

Definition 2.1.19 ([13])**.** The *Dowker complex* of a relation (R, X, Y) is the simplicial complex $(\text{D}(R), X)$ where

$$\text{D}(R) = \{\sigma \subseteq X \mid \exists y \in Y \text{ with } \sigma \times \{y\} \subseteq R\}.$$

We say that *the simplex $\sigma \in D(R)$ is witnessed by $y \in Y$* if $\sigma \times \{y\} \subseteq R$, i.e., all elements in the simplex σ are related with y.

*Denoted as $W(D; R, 0)$ in the original paper [10], and usually called "lazy witness complexes".

We will call Lazy Witness complex the Dowker complex of (R_ε, L, X) where

- X is the whole dataset,
- $L \subseteq X$ is a batch, and
- $R_\varepsilon = \{(l, x) \in L \times X \mid d(l, x) < \varepsilon\}$.

Definition 2.1.20 (Nested family of witness complexes [10])**.** Let $(\mathcal{X}, d)$ be a metric space, $X \subset \mathcal{X}$ be a dataset, $L = \{l_0, ..., l_n\} \subseteq X$ be a set of landmark points, and $\varepsilon > 0$. Then the k-simplex $\sigma = \{u_1, ..., u_k\}$ with $u_i \in L$ belongs to the *Lazy Witness complex* $\mathrm{W}_\varepsilon(X, L)$ iff all its faces belong to $\mathrm{W}_\varepsilon(X, L)$ and there is a witness $x \in X$, such that:

$$\max\{d(u_i, x) \mid u_i \in \{u_1, ..., u_k\}\} \leq \varepsilon\,.$$

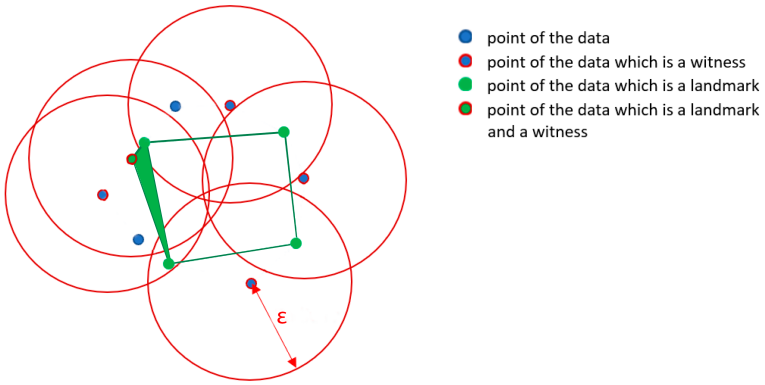


Figure 2.6: Witness complex for a set dataset of 11 points using a subset of 5 landmarks points and radius ε. Adapted from: [11]

A remarkable result about witness complexes is Dowker's Theorem, which states that exchanging the role of the witnesses and landmarks doesn't change the topology of the complex [13]. This result is exploited in [14] to enforce the topological regularization of a Topological Autoencoder (further explained in Section 2.2.1) only using the data points of the batches instead of the whole dataset.

Proposition 2.1.3 ([10])**.** *The family of complexes* $\mathrm{W}_\varepsilon(X, L)$ *is closely related to the family* $\mathrm{VR}(L, \varepsilon)$*. Specifically, there are inclusions:*

$$\mathrm{W}_\varepsilon(X, L) \subseteq \mathrm{VR}(L, 2\varepsilon) \subseteq \mathrm{W}_{2\varepsilon}(X, L)\,.$$

2.1.3 Homology

As discussed in [15], homotopy is a valuable algebraic tool for studying properties of topological spaces. However, the computational methods for computing homotopy can often be challenging to manage. To address this limitation, homology is introduced as an algebraic formalism that offers computational advantages, although it may not provide the same level of detailed topological information as other formalisms.

We will start by reviewing some linear algebra concepts that will be useful to understand homology's formal definition fully.

Linear algebra interlude

This subsection is based on Wojciech Chachólski's lecture notes from his course on Topological Data Analysis [16].

In TDA we tend to use vector spaces over F_2, the field with 2 elements $\{0, 1\}$. This field is isomorphic to $\mathbb{Z}_2 \equiv \mathbb{Z}/2\mathbb{Z}$ with operations modulo 2:

$$0 + 0 = 0;\ 1 + 0 = 1;\ 1 + 1 = 0;\ 1 \cdot 0 = 0;\ 1 \cdot 1 = 1\,.$$

We recall that an F_2 vector space is a set V with a zero element $0 \in V$ and an addition operation $V \times V \ni (v, w) \mapsto v + w \in V$ such that, for any u, v, w in V:

- $v + w = w + v$,
- $(v + u) + w = v + (u + w)$,
- $0 + v = v$,
- $v + v = 0$.

Another vector spaces that we will use are *quotient spaces*. Let $W \subset V$ be a vector subspace. We define the following equivalence relation: we say that two elements v and w in V are *equivalent modulo* W if $v - w \in W$. As we can see in Figure 2.7, the equivalence classes are of the form $[v] = v + W \subset V$. Moreover, the set of equivalence classes, denoted by V/W, with the following operations, is a vector space:

- $(v + W) + (w + W) = v + w + W$,
- $0 = 0 + W = W$,

- $a(v + W) = av + W$.

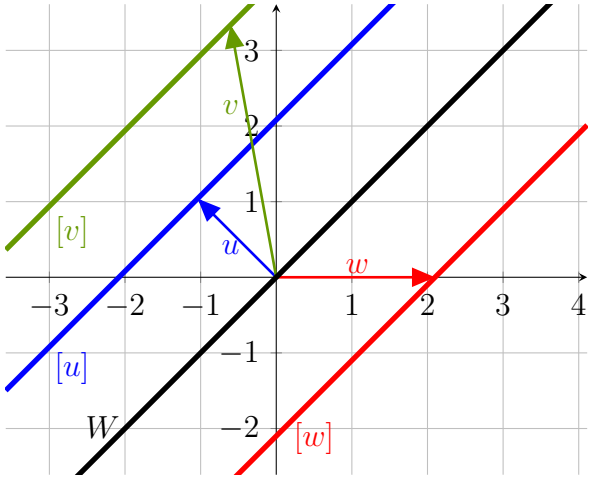


Figure 2.7: Quotient space example: $\mathbb{R}^2/W$

Lastly, we will see that one of our main components for constructing homology is by "transforming" a set to a vector space. Hence, for any finite set X, and field F, we define $FX \equiv \oplus_{x \in X} F$. The vector space FX is isomorphic to $F^{|X|}$, which means that we can represent an element of FX as sequences $(f_x)_{x \in X}$ indexed by elements of X.

Example 2.1.2. Let the set $S = \{a, b, c\}$, and the field $\mathbb{Z}_3$. Then an element of $\mathbb{Z}_3 S$ can be

$$(\underbrace{1}_{\text{a}}, \underbrace{0}_{\text{b}}, \underbrace{2}_{\text{c}}) = 1 \cdot a + 0 \cdot b + 2 \cdot c = a + 2c\,.$$

Therefore, we can see that we can define a basis for this space formed by elements e_x, where

$$(e_x)_y = \begin{cases} 1 & \text{if } y = x \\ 0 & \text{if } y \neq x \end{cases}$$

So the notation that we have seen in the example of representing an element as a (unique) linear combination $\sum_{x \in X} a_x x$ can be formalized if we identify each element x in X with e_x. Furthermore, given two elements $c = \sum_{x \in X} a_x x$ and $c' = \sum_{x \in X} b_x x$, their sum is defined as

$$c + c' = \sum_{x \in X} (a_x + b_x) x\,.$$

Lastly, we will see that we can extend a map of sets $g : X \to Y$ to a linear equation $Fg : FX \to FY$ such as an element $\sum_{x \in X} a_x x$ will be mapped to $\sum_{x \in X} a_x g(x)$.

Chain groups

We will begin by studying the various groups that are involved in the definition of homology.

Let K be a simplicial complex, F a field, p be a non-negative integer, and K_p be the set of p-dimensional simplices of K. Then a *p-chain* in K is an element of FK_p. In other words, c is a p-chain in K if

$$c = \sum a_i \sigma_i$$

with σ_i is a p-simplex for each i and $a_i \in F$ are the *coefficients*. These coefficients can be taken from any commutative ring F; however, we will use coefficients in the field of two elements, i.e., $a_i \in \mathbb{Z}_2$.

Example 2.1.3. We will write the simplices as the list of their vertices, $\sigma = [u_0, u_1, ..., u_p]$.

- In Figure 2.8 the 0-chain $c = [0] + [2] + [6] + [9]$ is shown in red.

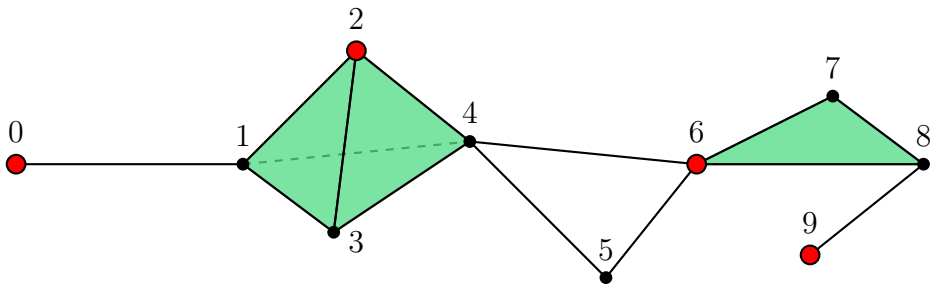


Figure 2.8: Example of 0-chain

- In Figure 2.9 the 1-chain $c = [0, 1] + [1, 2] + [2, 4] + [8, 9]$ is shown in red.
- In Figure 2.10 the 2-chain $c = [1, 2, 3] + [2, 3, 4] + [6, 7, 8]$ is shown in red.

The p-chains with the operation addition $+$ form the *group of p-chains* denoted by $(\mathrm{C}_p, +)$, but since the operation is understood, it is usually represented as $\mathrm{C}_p = \mathrm{C}_p(K)$.

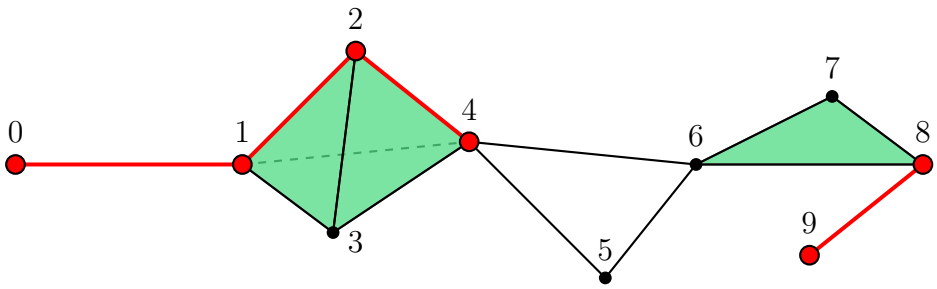


Figure 2.9: Example of 1-chain

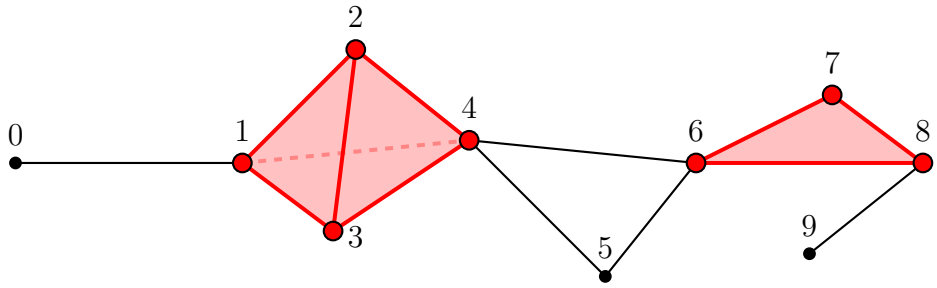


Figure 2.10: Example of 2-chain

This group is an abelian group, and as we have seen above, $\mathrm{C}_p(K)$ is a vector space over $\mathbb{Z}_2$. Hence, fixed $p \in \mathbb{Z}$, a basis of the vector space $\mathrm{C}_p(K)$ is the set $\{\sigma_i^p \mid i = 1, ..., s_p\}$ formed by the simplices of dimension p of K. As a consequence $\mathrm{C}_p(K) = \{0\}$, where $0 = \sum 0 \cdot \sigma_i$, if $p < 0$ or $p > \dim(K)$.

Boundary operator

To be able to relate these groups, we will define the *boundary operator*. Therefore, we will start with the definition of the p-boundary of a simplex.

Definition 2.1.21. Let p be an integer and $\sigma \in K$ be a p-simplex $\sigma = [v_0, v_1, ..., v_p]$ its *p-boundary*, denoted as $\partial_p\sigma$, is defined as the formal sum of its $(p-1)$-dimensional faces, that is,

$$\partial_p\sigma = \sum_{j=0}^{p} [v_0, ..., \hat{v}_j, ..., v_p]$$

where $\hat{v}_j$ denotes that v_j is omitted.

In general, given a p-chain $c = \sum a_i\sigma_i$, its p-boundary is defined by linear extension as $\partial_p c = \sum_{j=0}^{p} a_i\partial_p\sigma_i$. As a consequence, the boundary defines a

linear mapping $\partial_p : \mathrm{C}_p \to \mathrm{C}_{p-1}$ between vector spaces of chains called the *boundary operator*. To simplify the notation, the subscript p of the boundary operator is often omitted since it always matches the dimension of the chain to which it is applied.

Example 2.1.4. Let $c = [0,1] + [4,5]$ a 2-chain, then the boundary of c is:

$$\partial c = \partial[0,1] + \partial[4,5] = [0] + [1] + [4] + [5]\,.$$

Cycles and boundaries

We will distinguish two types of chains, which we will use to define homology groups.

Definition 2.1.22. We say that a p-chain c is a *p-cycle* if

$$\partial c = 0$$

or, equivalently, if $c \in \ker \partial$.

Example 2.1.5. We will see that geometrically, the p-cycles represent cycles in the simplicial complex. These, in turn, can be holes of dimension p. In Figure 2.11, the 1-cycle $[4,5] + [4,6] + [5,6]$ is shown in red, which is a hole. While blue represents the 1-cycle $[1,2] + [1,3] + [2,3]$, which is not a hole.

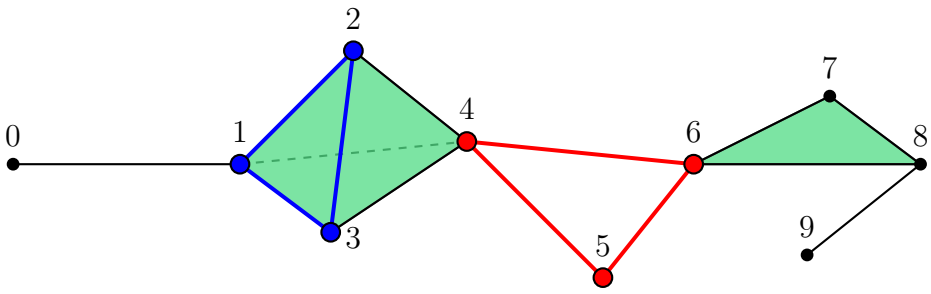


Figure 2.11: Example of 1-cycle

Definition 2.1.23. We say that a p-chain c is an *p-boundary* if there exists a $(p+1)$-chain c' such that

$$\partial c' = c$$

or, equivalently, if $c \in \operatorname{im} \partial_{p+1}$.

Example 2.1.6. The 1-cycle that we highlighted in blue in Figure 2.11 is a 1-boundary.

Remark. The sets of p-cycles $\mathrm{Z}_p = \ker \partial_p$, and p-boundaries $\mathrm{B}_p = \operatorname{im} \partial_{p+1}$ are linear subspaces of C_p.

We will prove that the p-boundaries are p-cycles, as in the previous example. For this, we will state the following lemma.

Lemma 2.1.4 (Fundamental lemma of homology [5, Chapter 4])**.** $\partial_p \partial_{p+1} c = 0$ *for every integer* p *and every* $(p+1)$*-chain* c*.*

It follows that B_p is a vector subspace of Z_p, that is, $\mathrm{B}_p \subset \mathrm{Z}_p$. Furthermore, we can define the *chain complex* associated with a simplicial complex K as the succession of chain groups connected by the boundary operators

$$\ldots \xrightarrow{\partial_{p+2}} \mathrm{C}_{p+1} \xrightarrow{\partial_{p+1}} \mathrm{C}_p \xrightarrow{\partial_p} \mathrm{C}_{p-1} \xrightarrow{\partial_{p-1}} \ldots$$

We can see in Figure 2.12 this relationship between the chain group C_p, the group of cycles Z_p and the group of boundaries B_p; and its connections generated by the boundary operator.

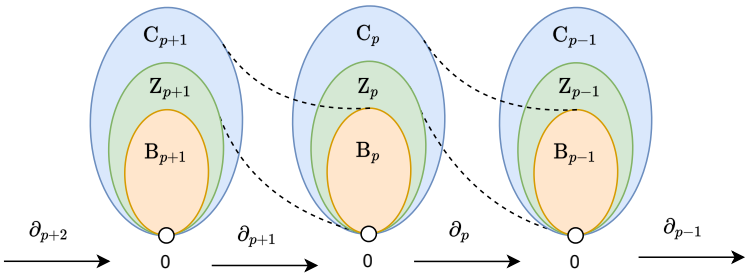


Figure 2.12: Chain complex representing the chain group, the cycle group, and the boundary group. Adapted from: [5]

Simplicial homology

The main idea of simplicial homology groups is to be able to find the holes by making use of the cycles. To do this, we will have to "discard" those cycles that are boundaries. This is why we will quotient the group of cycles by the group of boundaries since then, all the boundaries will be trivial in homology.

Definition 2.1.24. Given a simplicial complex K, its p*-dimensional simplicial homology group* is defined as the quotient space

$$\mathrm{H}_p(K) = \frac{\mathrm{Z}_p}{\mathrm{B}_p}.$$

The *p-dimensional Betti number* $\beta_p(K)$ is defined as the dimension of $\mathrm{H}_p(K)$.

Hence, the elements $z \in \mathrm{H}_p = \mathrm{H}_p(K)$ are of the form $z = c + \mathrm{B}_p$ with $c \in \mathrm{Z}_p$, where $c + \mathrm{B}_p$ is the *coset* of B_p in Z_p. Two cycles $c_1, c_2 \in \mathrm{Z}_p$ represent the same *homology class* $z \in \mathrm{H}_p$ if and only if $z = c_1 + \mathrm{B}_p = c_2 + \mathrm{B}_p$; which is equivalent to $(c_1 - c_2) \in \mathrm{B}_p$.

Definition 2.1.25. We say that two cycles $c_1, c_2 \in \mathrm{Z}_p$ are *homologous* if there exists $b \in \mathrm{B}_p$ such that

$$c_1 = c_2 + b\,.$$

Also, since Z_p, B_p, and H_p are vector spaces over $\mathbb{Z}_2$ it follows that

$$\beta_p = \dim \mathrm{H}_p = \dim \mathrm{Z}_p - \dim \mathrm{B}_p\,.$$

Topological properties

One of the most important values regarding homology groups is their corresponding Betti numbers, as these will give us a lot of information about the underlying space.

Theorem 2.1.5 ([15, Proposition 2.7])**.** *Let K be a simplicial complex. Then $\beta_0(K)$ matches the number of connected components of $|K|$.*

Corollary. *$|K|$ is connected if and only if $\beta_0(K) = 1$.*

We will see that our topological regularization will exploit the ability to study connectivity through homology. However, we can extract higher dimensional topological properties if we use higher dimensional homology groups. It can be shown [5, Chapter 5] that Betti numbers of a polyhedron contained in $\mathbb{R}^3$ can be interpreted in the following way:

- $\beta_0(K)$ tells us the number of connected components.
- $\beta_1(K)$ tells us the number of tunnels.
- $\beta_2(K)$ tells us the number of cavities.

In conclusion, the p-homology groups will represent p-dimensional holes in topological spaces.

2.1.4 Persistence

We will introduce the concept of persistence first for functions of one variable and then we will deepen in the case of simplicial complexes. In this section, I will use [17] as our main reference.

One-dimensional real functions

Let $f : \mathbb{R} \to \mathbb{R}$ be a smooth function. Remember that x is a *critical point* and $f(x)$ is a *critical value of* f if $f'(x) = 0$. Furthermore, a critical point x is *non-degenerate* if $f''(x) \neq 0$. So, suppose that f contains only non-degenerate critical points with distinct critical values.

Let the *sublevel set* $\mathbb{R}_t = f^{-1}(-\infty, t]$ for each $t \in \mathbb{R}$. Then we see that as we increase t, the number of connected components of $\mathbb{R}_t$ will remain constant until we pass through a t_0 critical value of f. As we can see in Figure 2.13, when we pass through a local minimum, a new connected component is created. When we pass through a local maximum, two connected components are combined into one.

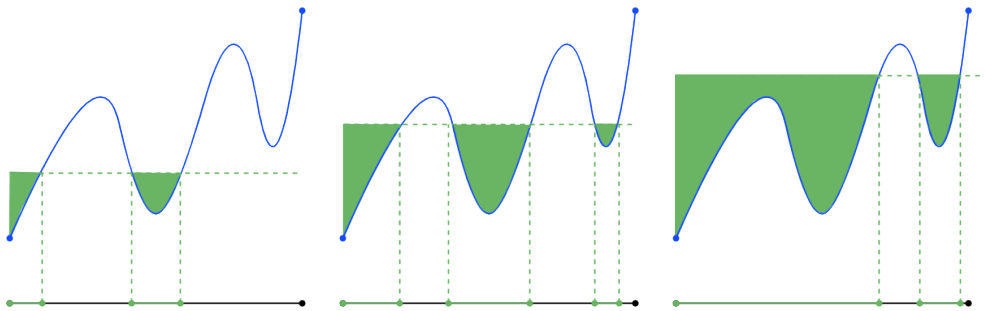


Figure 2.13: Connected components in $\mathbb{R}$ in the different leaks. Adapted from: [18]

The critical points of f are paired as follows:

1. When a new connected component appears, we say that the local minimum that created it *represents* that component.

2. When we pass through a local maximum and two components meet, we pair the maximum with the higher (youngest) of the two local minima that these components represent. The other minimum (the oldest) becomes the representative of the new component resulting from joining the two previous ones.

When the points x_1 and x_2 are paired following this method, we define the *persistence* of the pair as $f(x_2) - f(x_1)$. This persistence is coded through the *persistence diagram*, representing each pair with the point $(f(x_1), f(x_2))$, as can be seen in Figure 2.14. It can be seen that all points will lie above the

diagonal $y = x$ and that the persistence is the vertical distance from a point to the diagonal. For reasons that will be explained later, the points on the diagonal will be added to the persistence diagram.

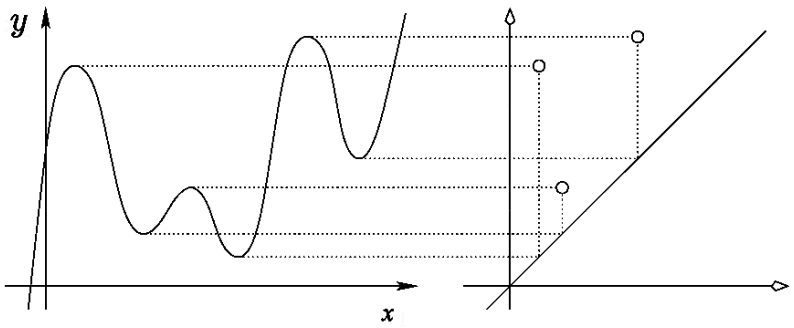


Figure 2.14: Pairing of the critical points of the function on the left represented as points in the persistence diagram on the right. Adapted from: [17]

Persistence on simplicial complexes

We will see that we can expand the concept of persistence seen for real functions to simplicial complexes. To do this, we will use the *filtrations* of a simplicial complex as sub-level sets, and we will use the *simplicial homology* as a homology theory.

Definition 2.1.26. Let K be a simplicial complex and $f : K \to \mathbb{R}$ a function. It is said that, f is *monotonous* if $f(\sigma) \leq f(\tau)$ when σ is a face of τ.

The monotony of f guarantees that for every $a \in \mathbb{R}$, the sub-level set $K(a) = f^{-1}(-\infty, a]$ is a subcomplex of K. As an example, we can define a Vietoris-Rips restricted to its 1-skeleton using a monotone function as follows:

Proposition 2.1.4 ([19]). *Let X be a subset of $\mathbb{R}^n$, $|X| = b$, and $\mathcal{V}(X) = \{\sigma \in \mathcal{P}([b]) \mid 1 \leq |\sigma| \leq 2\}$, and define*

$$f_X : \mathcal{V}(X) \to \mathbb{R},\ f_X(\sigma) = \begin{cases} 0 & \text{if } \sigma = \{i\} \\ \frac{1}{2}d(x_i, x_j) & \text{if } \sigma = \{i, j\} \end{cases}$$

Then the Vietoris-Rips complex w.r.t $r \geq 0$, restricted to its 1-skeleton is equal to $\mathrm{VR}(r)^{(1)} = f_X^{-1}(-\infty, r]$.

Definition 2.1.27. Let $a_1 < a_2 < \ldots < a_n$ the values that the function takes on the simplices and let $a_0 = -\infty$. Then f induces a *filtration*

$$\emptyset = K_0 \hookrightarrow K_1 \hookrightarrow \ldots \hookrightarrow K_n = K, \text{with } K_i = K(a_i)\,.$$

Since $K_i \subseteq K_j$ for all indices $i \leq j$, then the inclusions induce a linear function $f_p^{i,j} : \mathrm{H}_p(K_i) \to \mathrm{H}_p(K_j)$ for all p. We could say that this function sees on which homology class of K_j a cycle on K_i will belong. Hence we can formalize the idea of simplicial persistence, making use of these functions.

Definition 2.1.28. Let $f_p^{i,j} : \mathrm{H}_p(K_i) \to \mathrm{H}_p(K_j)$ be the linear mapping induced by the inclusion $K_i \subseteq K_j$. The *persistent homology groups* are defined as the image of $\mathrm{H}_p(K_i)$ in $\mathrm{H}_p(K_j)$ of the map $f_p^{i,j}$, that is ,

$$\mathrm{H}_p^{i,j} = \mathrm{im}\ f_p^{i,j} .$$

The corresponding *persistent Betti numbers* are defined as the dimension of these vector spaces, i.e., $\beta_p^{i,j} = \dim \mathrm{H}_p^{i,j}$.

Remark. If we analyze the maps $f_p^{i,j}$, we observe that the $\ker f_p^{i,j}$ are those elements $\gamma \in \mathrm{H}_p(K_i)$ such that $f_p^{i,j}(\gamma) = 0$. This means that if c is a cycle representing γ, then $c \in \mathrm{B}_p(K_j)$. Hence,

$$\ker f_p^{i,j} = \frac{\mathrm{Z}_p(K_i) \cap \mathrm{B}_p(K_j)}{\mathrm{B}_p(K_i)}$$

for each dimension p fixed. Therefore,

$$\mathrm{H}_p^{i,j} = \mathrm{im}\ f_p^{i,j} \cong \frac{\mathrm{H}_p(K_i)}{\ker f_p^{i,j}} = \frac{\frac{\mathrm{Z}_p(K_i)}{\mathrm{B}_p(K_i)}}{\frac{\mathrm{Z}_p(K_i) \cap \mathrm{B}_p(K_j)}{\mathrm{B}_p(K_i)}} \cong \frac{\mathrm{Z}_p(K_i)}{\mathrm{Z}_p(K_i) \cap \mathrm{B}_p(K_j)} ,$$

which means that the persistent homology group consists of the classes that were born before a_i and are still alive in a_j.

Comparing this case with the previously shown one in which we used a real function, the critical values of homology are the levels at which the homology of the sublevel sets changes. In this way, we will say that a homology class γ is born in K_i if it is not in the image of the function induced by the inclusion $K_{i-1} \subseteq K_i$. Also, a class γ that is born in K_i dies when entering K_j if the image of the function induced by $K_{i-1} \subseteq K_{j-1}$ does not contain the image of γ, but the image of the function induced by $K_{i-1} \subseteq K_j$ does. Which can be formally redefined as follows using persistent homology groups:

- A class $\gamma \in \mathrm{H}_p(K_i)$ *is born* in K_i if $\gamma \notin \mathrm{H}_p^{i-1,i}$.
- A class $\gamma \in \mathrm{H}_p(K_i)$ born in K_i *dies* on entering K_j if $f_p^{i,j-1}(\gamma) \notin \mathrm{H}_p^{i-1,j-1}$, but $f_p^{i,j}(\gamma) \in \mathrm{H}_p^{i-1,j}$.

Definition 2.1.29. Let γ be a homology class that is born in K_i and dies when it enters K_j. The *persistence* of γ is defined as $\text{pers}(\gamma) = a_j - a_i$. Likewise, the difference $j - i$ is called the *persistence index* of the class γ. If a class γ is born in K_i but never dies, then we say that its persistence, like its index, is infinite.

Following this notation, the multiplicity is defined as

$$\mu_p^{i,j} = (\beta_p^{i,j-1} - \beta_p^{i,j}) - (\beta_p^{i-1,j-1} - \beta_p^{i-1,j}) .$$

Where $\beta_p^{i,j}$ can be interpreted as the number of homology classes that are alive in K_i and still alive in K_j. Therefore, the first difference of the equality is interpreted as the number of independent classes that are alive at K_i and die at K_j, while the second difference is the number of independent classes that are born before K_i and die in K_j. In conclusion, the multiplicity, $\mu_p^{i,j}$, is interpreted as the number of homology classes that are born in K_i and die in K_j.

Definition 2.1.30. The *persistence diagram* $\text{Dgm}(f) \subset \overline{\mathbb{R}}^2$ of f is the multiset of points (a_i, a_j) with multiplicity $\mu_p^{i,j}$ for all $0 \leq i < j \leq n$, union of diagonal points, $\Delta = \{(x, y) \in \overline{\mathbb{R}}^2 \mid y = x\}$, with infinite multiplicity.

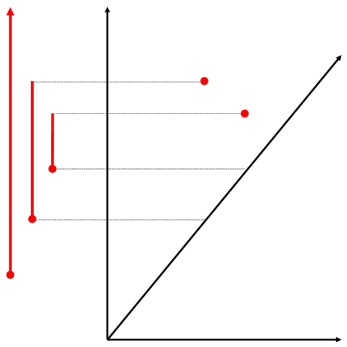


Figure 2.15: Barcode associated with a persistence diagram. Adapted from: [20]

Therefore, in a persistence diagram, each point (a_i, a_j) represents $\mu_p^{i,j}$ independent homology classes whose persistence coincides with the distance from point (a_i, a_j) to its vertical projection on the diagonal Δ. In addition to the persistence diagrams, we can encode information about persistent homology through so-called *barcodes*. These representations can be obtained from the persistence diagram by drawing for each point (a_i, a_j) with $a_i < a_j$

of said diagram $\mu_p^{i,j}$ half-open intervals $[a_i, a_j)$, as shown in Figure 2.15.

Finally, we will denote by $\#(A)$ the *total multiplicity* of a multiset A, which, by definition, is the sum of the multiplicities of the elements of A. Thus the total multiplicity of the persistence diagram minus the diagonal is

$$\#(\mathrm{Dgm}(f) \setminus \Delta) = \sum_{i<j} \mu_p^{i,j} .$$

This multiplicity is called the *size* of the persistence diagram.

We will denote the closed upper left quadrant with vertex at the point (a_i, a_j) as $Q_i^j = [-\infty, a_i] \times [a_j, \infty]$.

Lemma 2.1.6 (p-Triangle lemma [21])**.** *Let f be a monotonic function. Then, the total multiplicity of the persistence diagram in the upper left quadrant with vertex (x, y) is*

$$\#(\mathrm{Dgm}(f) \cap Q_i^j) = \beta_p^{i,j} .$$

This lemma guarantees us that the persistence diagram encodes all information about persistent homology groups [5, Chapter 7].

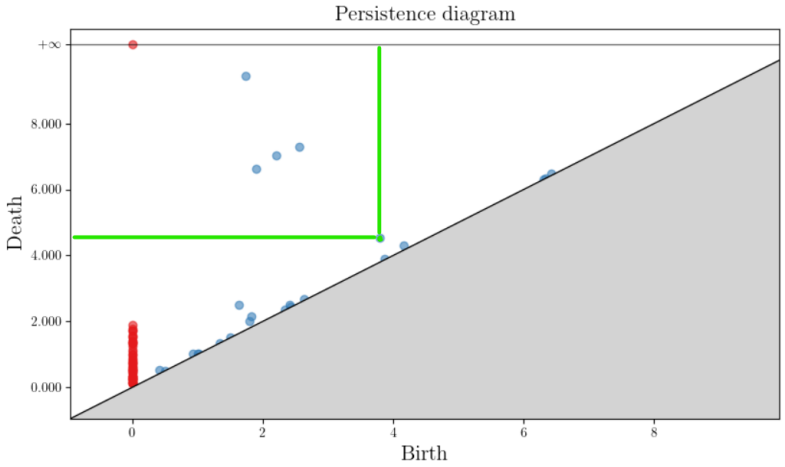


Figure 2.16: Persistence diagram, where 0-dimensional persistent homology is in red and 1-dimensional case is in blue. The 1-dimensional persistent Betti number corresponding to the green point is 5.

2.2 Related work

Upon reviewing the necessary mathematical background, we will now explore the relevant literature related to Topological Machine Learning (TopoML) and Representation Similarity.

2.2.1 Topological Machine Learning

In Section 2.1, we discussed the effectiveness of Persistent Homology in revealing potential topological characteristics of the underlying manifold from which our dataset is sampled. This tool has found extensive application in exploratory data analysis [22, 23, 24], providing a geometric perspective on data. Figure 2.17 illustrates a standard pipeline for such analysis.

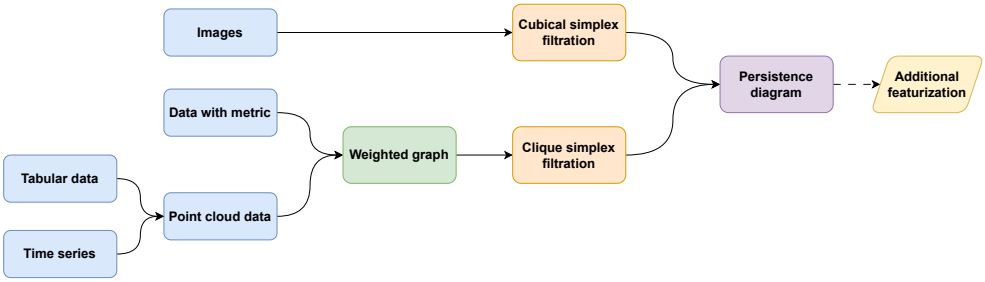


Figure 2.17: Standard TDA pipeline. Adapted from: [25]

Typically, our objective is to identify groups of data points sharing similar topological properties. To achieve this, we employ a neighborhood-based identification scheme, such as using fixed-radius balls centered at each point. More complex neighborhoods, like annuli or k-NN groupings, can also be considered. Once the identification function $\pi : X \to \mathcal{P}(X)$ is obtained, we compute the persistent homology for each subset $\pi(x) \subseteq X,\ x \in X$.

However, dealing with multisets poses a challenge when comparing topological characteristics. Consequently, various approaches have been proposed to handle these multisets as input for Machine Learning or Data Science methods [2]. The main approaches include:

- Encoding persistent homology information as piecewise constant functions, such as Stable Ranks [26] or Betti curves [2]. Another popular technique is the use of persistence landscapes [27], which are functional vector spaces (see Figure 2.18). These representations facilitate comparison using distances such as interleaving distance or L_p

norms. Moreover, we can use all our knowledge of calculus to obtain statistical analysis rewarding our newly obtained functions.

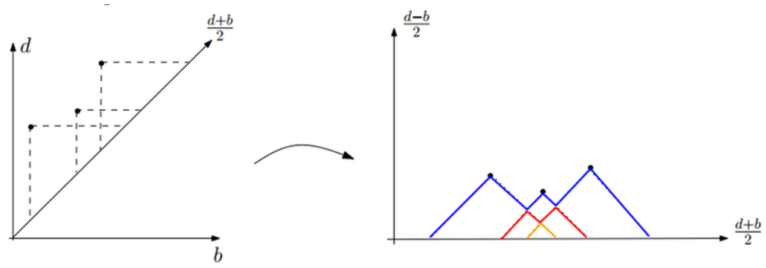
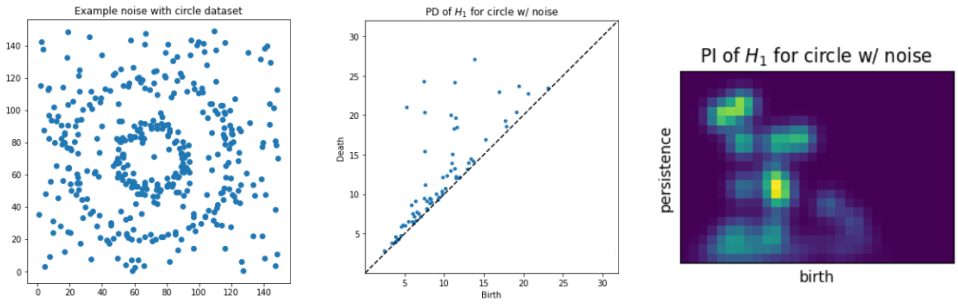


Figure 2.18: Persistence landscape construction. CC BY Source: [28]

From an ML perspective, kernel methods like SVMs can utilize the scalar product defined on these functions, or a signal processing setup can be simulated by sampling and quantizing to vectorize the functions over a certain domain.

- Transforming the persistence diagram into more common ML input formats, such as images or vectors. Persistence images [29] aim to capture both persistence and the relative importance of features in a "heat map" representation (see Figure 2.19). Alternatively, pairwise distances of points from the multiset can be encoded as a distance vector [30].



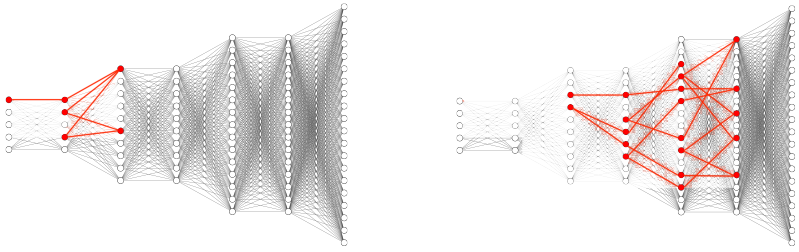
(a) Dataset (b) Persistence diagram (c) Persistence image

Figure 2.19: Persistence image construction. Adapted from: [31]

In the context of Deep Learning (DL), recent work has proposed novel approaches that focus on the network itself instead of postprocessing the persistence diagram. These approaches involve creating layers that learn to embed the persistent diagram or utilizing more complex network architectures

to handle multisets as input [2].

However, in this project, we will not focus our attention on using the persistent diagram as our input but on guiding the training of our standard classification tasks based on certain desired topological properties. But, before dealing with the main topic of topological regularization techniques, we will present an interesting use of TDA to asses certain properties of trained networks. It has been shown that we can make use of persistent homology in order to identify whether our network has been a victim of a trojan attack [32]. This kind of attack in a DL context means that there have been introduced some modified samples with a wrongly attached label during the training. Hence, the user, during testing, will not notice that the network has memorized the trojan pattern since the testing is done with the "clean samples".



(a) Clean Model + Trojaned Input (b) Trojaned Model + Trojaned Input

Figure 2.20: Trojan attacks. Adapted from: [32]

By analyzing the persistent homology of the neuron connectivity graph, it is possible to extend the Hebbian learning rule of *"fire together, wire together"* to higher-level connections, where wiring is not limited to adjacent nodes but also includes path-connected nodes. Hence, as we can see in Figure 2.20, a trojaned input on a trojaned network will generate a highly persistent cycle, which will connect the shallow and deep layers, short-cutting the proper classification. The reason we showcase this study is to remark on the fact that with some creativity and ingenuity, you can utilize persistent homology in more ways than to study the geometric characteristics of your dataset.

As mentioned briefly in Chapter 1, our focus is on applying TDA to assist in the training process of neural networks. As an example, Topological Autoencoders [33] leverage persistent homology to preserve the topological features of the dataset in the encoded latent space. By incorporating a penalty term based on the distance between two persistence diagrams, the network

learns the underlying manifold's topology.

In the context of classification tasks, we highlight the work of Chen *et al.,* [34], where they reduce generalization error by imposing a simpler topology on the decision boundary. This technique offers direct control over suppressing spurious topological structures, unlike other regularization methods that aim to simplify the decision boundary as a proxy for variance reduction.

Topologically Densified Distributions

The specific type of topological regularization we investigate in this project falls under the category of internal representation regularization. The goal of this regularization technique is to impose certain characteristics on our latent space that can benefit the classification task.

In a more traditional approach, we can consider techniques such as class-wise variance reduction (cw-VR) and class-wise correlation reduction (cw-CR) [35]. These methods aim to decrease the variance and correlation, respectively, of the latent space representation.

Let $(\mathcal{X}, P)$ be our input space with probability measure P, $\mathcal{Y} = [K] = \{1, ..., K\}$ the target space, and our network decomposed as $\gamma \circ \varphi : \mathcal{X} \to \mathcal{Y}$, where $\varphi : \mathcal{X} \to \mathcal{Z}$ is a feature extractor and $\gamma : \mathcal{Z} \to \mathcal{Y}$ is the classification head.

Additionally, since we are dealing with a classification task, we will define the function $c : \text{supp}(P) \to \mathcal{Y}$ that for each input it assigns its true label. We can restrict the domain of this function to the support of P without loss of generality due to the assumption of the manifold thesis [1].

Suppose our training set $S = \{x_1, ..., x_n\}$ is generated by n i.i.d draws from $X \sim P$, and $S_{x|k} = \{x \in S \mid c(x) = k\}$ represents the subset of samples in class k. Then, we can define the following regularization terms:

$$\Omega_{cw\text{-}CR} = \sum_k \sum_{i \neq j} \left(c_{i,j}^{(k)} \right)^2 , \text{ and } \Omega_{cw\text{-}VR} = \sum_k \sum_i v_i^{(k)}$$

where the class mean vector, covariance matrix, and variance vector are

defined as:

$$\mu_i^{(k)} = \frac{1}{\left|\varphi(S_{x|k})\right|} \sum_{z \in \varphi(S_{x|k})} z_i \,,$$

$$c_{i,j}^{(k)} = \frac{1}{\left|\varphi(S_{x|k})\right|} \sum_{z \in \varphi(S_{x|k})} (z_i - \mu_i^{(k)})(z_j - \mu_j^{(k)}) \,,$$

$$v_i^{(k)} = c_{i,i}^{(k)} \,.$$

Remark. The cw-VR and cw-CR regularization techniques bear similarities to the application of the KL divergence with a prior distribution on the latent space $\mathcal{Z} = \mathbb{R}^{m*}$ as seen in Variational Autoencoders [36]. However, the Bayesian setup in VAEs offers greater flexibility since the prior can incorporate more knowledge about the latent space. Nevertheless, it requires an explicit approximation of the push-forward probability Q induced on $\mathcal{Z}$ by the random variable $\varphi : \mathcal{X} \to \mathcal{Z}$ from P.

While statistical and probabilistic approaches have shown success, they often rely on empirical evidence to demonstrate their direct reduction of generalization error [3]. However, Hofer *et al.,* [3] propose a novel internal regularization method that has been mathematically proven to reduce generalization error under mild assumptions. To be precise, we define generalization error as follows:

Definition 2.2.1. For a classifier $h : (\mathcal{X}, P) \to \mathcal{Y}$, we define the *generalization error* as $\mathbb{E}_{X \sim P}[\mathbb{1}_{h,c}(X)]$, where

$$\mathbb{1}_{h,c}(x) = \begin{cases} 0 & \text{if } h(x) = c(x) \\ 1 & \text{otherwise} \end{cases} .$$

For this purpose, we will use the conditional push-forward probability on $\mathcal{Z}$, which can be defined as $Q_k(A) \equiv Q(A \mid C_k) = Q(A \cap C_k)/Q(C_k)$, for any class $k \in \mathcal{Y}$, event $A \in \mathfrak{B}$, and internal class representation $C_k = \varphi(c^{-1}(k))$. Hence, we will see that if we can ensure to set an upper bound to each of the conditional probabilities of misclassifying the decision region of

*For the topological densification technique, we can generalize the latent space to any metric space $(\mathcal{Z}, d)$ endowed with the push-forward probability measure Q induced by the random quantity $\varphi : \mathcal{X} \to \mathcal{Z}$, on the Borel σ-algebra $\mathfrak{B}$ defined by d on $\mathcal{Z}$ (i.e., for any event $A \in \mathfrak{B}$, $Q(A) = P(\varphi^{-1}(A))$). However, even in image classification, it is common to vectorize the latent space before feeding it to an MLP-based classification head γ. Therefore, assuming $\mathcal{Z} = \mathbb{R}^m$ is a reasonable assumption.

that given class, then we will reduce the generalization error. This statement is formalized in the following lemma:

Lemma 2.2.1 ([3]). *For any class $k \in \mathcal{Y} = [K]$, let $C_k = \varphi(c^{-1}(k))$ be its internal representation, and $D_k = \gamma^{-1}(k)$ be its decision region in $\mathcal{Z}$ w.r.t γ. If exists $\varepsilon > 0$ s.t.*

$$\forall k \; 1 - Q_k(D_k) \leq \varepsilon \,, \tag{2.1}$$

then

$$\mathbb{E}_{X \sim P}[\mathbb{1}_{\gamma \circ \varphi, c}(X)] \leq K\varepsilon \,.$$

In order to construct the regularization technique that increases those probabilities, we will observe this statement from a measure-theoretic probabilistic point of view*. In this context, we can reinterpret Eq. 2.1 of Lemma 2.2.1 as controlling how much probability mass of C_k is concentrated in D_k w.r.t. Q_k. Moreover, we will assume that our network perfectly classifies the training data set S without much structural risk, i.e., for every $z_i = \varphi(x_i)$ with $c(x_i) = k$, then there exists $r > 0$ (sufficiently large) with $\overline{B_r}(z_i) \subset D_k$. Consequently, increasing the mass concentration in $\overline{B_r}(z_i)$, i.e., $Q_k(\overline{B_r}(z_i))$, will subsequently reduce generalization error by increasing $Q_k(D_k)$.

Having understood the benefits of achieving mass concentration over a specific reference set $M \subseteq \mathcal{Z}$, we will see that we can provably enforce this property by applying some topological constraint on Q_k in a process called *topological densification*. To be precise, this topological constraint will give us a non-trivial lower bound on $Q_k(M_{l\cdot\varepsilon})$ in terms of $Q_k(M)$, where $M_{l\cdot\varepsilon}$ represents the $l \cdot \varepsilon$-extension of M for any $\varepsilon > 0$ and $l \in \mathbb{N}$, i.e.

$$M_{l\cdot\varepsilon} = \bigcup_{z \in M} \overline{B_{l\cdot\varepsilon}}(z) \,.$$

The topological constraint that we will use is the natural extension of the β-connectivity to random samples. Hence, we will create an indicator function (random variable) $\mathfrak{c}_b^\beta : \mathcal{Z}^b \to \{0, 1\}$ as

$$\mathfrak{c}_b^\beta(z_1, ..., z_b) = 1 \Leftrightarrow \{z_1, ..., z_b\} \text{ is } \beta\text{-connected} \,.$$

Definition 2.2.2. Let $\beta > 0$, $c_\beta \in [0, 1]$, $b \in \mathbb{N}$, and Q_k^b the product measure of Q_k. We call Q_k (b, c_β)-*connected* if $Q_k^b(\{\mathfrak{c}_b^\beta = 1\}) \geq c_\beta$.

*For a foundational understanding of probability theory from this perspective, we refer the reader to [37, Appendix B].

Hofer *et al.,* proved that: ***If a measure is (b, c_β)-connected, then mass attracts mass; the higher c_β, the stronger the effect*** [3]. In the original paper, they have an in-depth study on the properties of the lower bound of $Q_k(M_{l\cdot\varepsilon})$ in terms of a (b, c_β)-connected $Q_k(M)$, such as the minimum required mass to enforce proper mass condensation. As a broad idea, the way to obtain the lower bound is to study the properties of the trinomial distribution that characterizes certain events where the samples $\{z_1, ..., z_b\}$ are not β-connected.

Therefore, based on this property, we will enforce that for each batch $\mathcal{B}$, the subsets $\mathcal{B}_k = \{b \in \mathcal{B} \mid c(b) = k\}$ are β-connected. In order to measure how close they are to being β-connected and to have a numerical cost function penalizing these deviations, we will restate β-connectivity using persistent homology.

Definition 2.2.3. Let $\beta > 0$. A set $M \subseteq \mathcal{Z}$ is *β-connected* iff all 0-dimensional death-times of its Vietoris-Rips persistent homology, denoted as $\dagger(M)$, are in the open interval $(0, \beta]$.

We can clearly see that the two definitions are equivalent if we remember the proposition 2.1.2, stating:

Proposition 2.1.2. *For any distance d on X, and $\beta \geq 0$, the partition corresponding to the connected components of* $\mathrm{VR}(\beta)$*, denoted as* $\pi_0(\mathrm{VR}(\beta))$*, coincides with the β-components of X.*

Hence, the steps to enforce the *topological densification* are:

1. We construct each mini-batch, denoted as $\mathcal{B}$, as a collection of n sub-batches, i.e., $\mathcal{B} = (\mathcal{B}_1, ..., \mathcal{B}_n)$. Each sub-batch consists of b samples from the same class, resulting in an effective batch size of $n \cdot b$.

2. We incorporate the topological regularization cost function into our classification loss, given by

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda \mathcal{L}_\beta, \ \lambda > 0 \tag{2.2}$$

where,

$$\mathcal{L}_\beta = \sum_{i=1}^{n} \underbrace{\sum_{d \in \dagger(\mathcal{B}_i)} |d - \beta|}_{\widetilde{\mathcal{L}}_\beta(\mathcal{B}_i)} . \tag{2.3}$$

Lastly, we can prove the differentiability of this cost function by redefining $\tilde{\mathcal{L}}_\beta(\mathcal{B}_i)$ using a new indicator function, under the mild assumption that the pairwise distances of elements in $\mathcal{B}_i$ are unique. This assumption holds almost surely if we consider that Q is non-atomic, i.e., $Q(\{\varphi(x) = z\}) = 0$ for $x \in \mathcal{X}$, and $z \in \mathcal{Z}$ [19]. Furthermore, we will assume that $\mathcal{Z} = \mathbb{R}^m$ (i.e., we vectorized our latent space).

Theorem 2.2.2 (Topological regularization differentiability [19])**.** *Let $M \subseteq \mathbb{R}^m$ with $|M|$. Hence, $\widetilde{\mathcal{L}}_\beta(\mathcal{B}_i)$ as described on Eq. 2.3 can be rewritten as*

$$\widetilde{\mathcal{L}}_\beta(M) = \sum_{\{i,j\}\subset[b]} |\beta - d(z_i, z_j)| \cdot \mathbb{1}_{i,j}(z_1, ..., z_b)\,,$$

where

$$\mathbb{1}_{i,j}(z_1, ..., z_b) = \begin{cases} 1 & \text{if } d(z_i, z_j) \in \dagger(M) \\ 0 & \text{otherwise} \end{cases}.$$

Furthermore, for $1 \leq u \leq b$ and $1 \leq v \leq m$, the partial (sub-)derivative of $\widetilde{\mathcal{L}}_\beta(M)$ w.r.t. the v-coordinate of z_u exist and is given by

$$\frac{\partial \widetilde{\mathcal{L}}_\beta(M)}{\partial z_{u,v}} = \sum_{\{i,j\}\subset[b]} \frac{\partial\,|\beta - d(z_i, z_j)|}{\partial z_{u,v}} \cdot \mathbb{1}_{i,j}(z_1, ..., z_b)\,.$$

Figure 2.21 provides a visual representation of the mass attraction induced by the topological regularization when applied to our sub-batches during training.

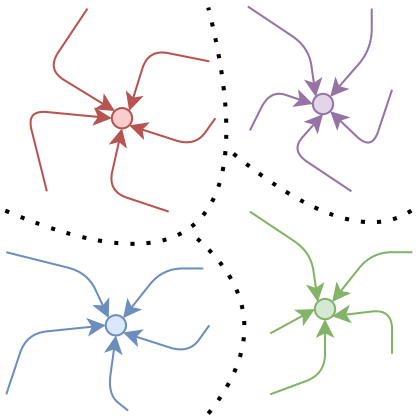


Figure 2.21: Densification process in each decision region due to the topological regularization. Inspired by: [38]

2.2.2 Representation Similarity & Model Stitching

In various domains within the field of ML, there has been significant interest in obtaining a function capable of extracting the most relevant features from input data. This pursuit of feature extraction predates the advent of DL and falls under the broader umbrella of Representation Learning. For example, in computer vision, numerous algorithms rely on the extraction of relevant features and descriptors, such as SIFT descriptors [39]. These "classical" feature extractors were manually crafted to suit specific domains (e.g., using Haar features for face detection [40]).

However, DL networks have introduced a new paradigm in representation learning, where they can be viewed as complex feature extractors composed of task-specific mappings [4]. Despite achieving state-of-the-art results, we still lack a comprehensive understanding of the limits and properties of these feature extractors. Several pertinent questions arise, such as: *How unique are the learned features in our models? How can we compare the latent spaces of different networks to assess their similarity? What do we mean when we say that two layers exhibit similar behavior?*

In this subsection, we will elucidate some of the prominent methods employed to address these questions and highlight the potential benefits of gaining insights into these aspects of our networks.

Given the nature of the aforementioned questions, our focus will be on the subfield of Representation Similarity, which exhibits significant overlaps with Representation Learning. In this context, we aim to compare layer L_1 of network f_A with layer L_2 of network f_B. It is important to note that the dimensionality of the two networks may differ completely. For simplicity, we assume, without loss of generality, that we can vectorize the output of both layers into $\mathbb{R}^{d_1}$ and $\mathbb{R}^{d_2}$, respectively, with $d_1 \leq d_2$.

A common practice when comparing layers from different networks is to examine the neuron activations for a test batch of n samples. Let $X \in \mathbb{R}^{d_1 \times n}$ and $Y \in \mathbb{R}^{d_2 \times n}$ be the matrices containing the activations of the two layers for the sample set of n points. We can define the following similarity metrics:

- **CCA:** a statistical way of comparing the activation layers is by using canonical correlation analysis (CCA). This method tries to find the best low-dimensional linear projections of X and Y, such as we maximize its correlation, i.e., we wish to find the vectors $\mathbf{w} \in \mathbb{R}^{d_1}$, $\mathbf{s} \in \mathbb{R}^{d_2}$ such

as we maximize

$$\rho = \text{corr}(\mathbf{w}^T X, \mathbf{s}^T Y) = \frac{\langle \mathbf{w}^T X, \mathbf{s}^T Y \rangle}{\|\mathbf{w}^T X\| \cdot \|\mathbf{s}^T Y\|} .$$

As we can see in [41], if we assume that the rows of X and Y are centered and letting $\Sigma_{X,X} = \text{cov}(X, X)$, $\Sigma_{Y,Y} = \text{cov}(Y, Y)$, and $\Sigma_{X,Y} = \text{cov}(X, Y)$, then we can reformulate the previous equation as

$$\begin{aligned} \max \rho &= \max_{\mathbf{w},\mathbf{s}} \frac{\mathbf{w}^T \Sigma_{X,Y} \mathbf{s}}{\sqrt{\mathbf{w}^T \Sigma_{X,X} \mathbf{w}} \sqrt{\mathbf{s}^T \Sigma_{Y,Y} \mathbf{s}}} \\ &= \max_{\mathbf{u},\mathbf{v}} \frac{\mathbf{u}^T \Sigma_{X,X}^{-1/2} \Sigma_{X,Y} \Sigma_{Y,Y}^{-1/2} \mathbf{v}}{\sqrt{\mathbf{u}^T \mathbf{u}} \sqrt{\mathbf{v}^T \mathbf{v}}} \qquad \left(\mathbf{w} = \Sigma_{X,X}^{-1/2} \mathbf{u},\ \mathbf{s} = \Sigma_{Y,Y}^{-1/2} \mathbf{v} \right) \end{aligned}$$

This maximization problem can be solved with singular value decomposition [41]: $U \Lambda V = \mathbf{u}^T \Sigma_{X,X}^{-1/2} \Sigma_{X,Y} \Sigma_{Y,Y}^{-1/2} \mathbf{v}$, being $\max \rho \equiv \rho_1$ the first singular value, and $\mathbf{u}^{(1)}, \mathbf{v}^{(1)}$ its corresponding left and right singular vectors. As we know from properties of the SVD, the second highest correlation under the constraint that $\langle \mathbf{u}^{(1)}, \mathbf{u}^{(2)} \rangle = 0$, $\langle \mathbf{v}^{(1)}, \mathbf{v}^{(2)} \rangle = 0$ is given by the second highest singular value ρ_2 and its corresponding left and right singular vectors. Hence, an intuitive way to assess the similarity between network representations is by the following summary statistics:

$$R_{CCA}^2 = \frac{\sum_{i=1}^{d_1} \rho_i^2}{d_1}, \text{ and } \bar{\rho}_{CCA} = \frac{\sum_{i=1}^{d_1} \rho_i}{d_1} .$$

However, it has been seen that these metrics are sensitive to perturbations when the condition number of X or Y is large [42]. Therefore, we have the following two variations that try to increase the robustness:

- **SVCCA:** a simple way to avoid these perturbations is to restrict the number of singular values used when computing the metric.
- **PWCCA:** a less harsh way to reduce the contribution of "less important" singular components is by having a weighted average, i.e.

$$\rho_{PW} = \frac{\sum_{i=1}^{d_1} \alpha_i \rho_i}{\sum_{i=1} \alpha_i}, \text{ with } \alpha_i = \sum_j |\langle \mathbf{h}_i, \mathbf{x}_j \rangle|$$

where $\mathbf{h}_i = (\mathbf{u}^{(i)})^T \Sigma_{X,X}^{-1/2} X$ and $\mathbf{x}_j$ is the j^{th} row of X. These weights, α_i, are based on the hypothesis that CCA vectors that account for a larger proportion of the original outputs are likely to be more important to the underlying representation. [41].

- **CKA**: this method proposes an alternative approach to comparing the covariance of the (projected) activations directly, as done in CCA. Instead, CKA suggests computing similarity matrices for the activations in each representation separately and comparing these matrices [42]. Hence, if we use the standard inner product, then our similarity metric between the layers L_1 and L_2 will be $\langle \text{vec}(X^T X), \text{vec}(Y^T Y) \rangle$. Furthermore, we will see that we can generalize this metric to any inner product in a reproducing kernel Hilbert Space (RKHS). For that, we will first review the following property of the Frobenious norm: let $A \in \mathbb{R}^{n \times m}$, then

$$\|A\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} A_{ij}^2} = \sqrt{\text{tr}(A^T A)}\,.$$

Moreover, if we have $A = YX^T$, then

$$\begin{aligned} \left\|YX^T\right\|_F^2 &= \text{tr}(XY^T YX^T) \\ &= \text{tr}(X^T XY^T Y) && (\text{tr}(AB) = \text{tr}(BA)) \\ &= \langle \text{vec}(X^T X), \text{vec}(Y^T Y) \rangle && (X^T X, Y^T Y \text{symmetric}) \end{aligned}$$

So if we further assume that the rows of X and Y are centered, then this last result states that the similarity between the similarity matrices of the activations of each layer is equal to the squared Frobenious norm of the covariance matrix between the activation matrices, i.e., $\|\Sigma_{X,Y}\|_F^2$.

The *Hilbert-Schmidt Independence Criterion* [43] will generalize the previous statement to any inner product in an RKHS as follows:

$$\text{HSIC}(K, L) = \frac{1}{(n-1)^2} \text{tr}(KHLH)\,,$$

where K and L are kernel matrices defined as $K_{ij} = k(\mathbf{x}_i, \mathbf{x}j)$ and $L_{ij} = l(\mathbf{y}_i, \mathbf{y}_j)$ for any kernels k and l, and H is the centering matrix defined as $H_n = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$.

Remark. If we use linear kernels $k(\mathbf{x}, \mathbf{y}) = l(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$, then $\text{HSIC}(X^T X, Y^T Y) = \|\Sigma_{X,Y}\|_F^2$

However, in order to make HSIC invariant to isotropic scaling, we will normalize it. This normalized index is known as centered kernel alignment [42]:

$$\text{CKA}(K, L) = \frac{\text{HSIC}(K, L)}{\sqrt{\text{HSIC}(K, K) \cdot \text{HSIC}(L, L)}}. \tag{2.4}$$

If we have a more in-depth look at CKA with a linear kernel, denoted as Linear CKA, we can see that it will resemble CCA weighted by the eigenvalues of the corresponding eigenvectors [42]. Hence, in this case, CKA encodes similar information to SVCCA and PWCCA, but we will not require computing any matrix decomposition.

Moreover, the fact that CKA is defined for any kernel in an RKHS gives us more flexibility than the previous metrics. For instance, if we use the RBF kernel $k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|_2^2 / (2\sigma^2))$, we can control the emphasis on similarity at small distances compared to large distances by varying the bandwidth parameter σ [42].

- **Others:** While CKA is the most commonly used similarity metric for comparing latent representations, there are other more complex methods available. These include the maximum matching similarity [44], which compares layers based on the subspaces generated by activation vectors of a subset of neurons, and the representation topology divergence (RTD) [45], which compares the topological characteristics of the latent spaces similar to the Topological Autoencoder with the input and latent space.

By employing these similarity metrics, we can investigate the circumstances under which our networks exhibit greater similarity to one another. Notably, wider networks and networks with low generalization error demonstrate a tendency to converge towards more similar solutions when trained on the same dataset [41]. Additionally, earlier layers, as opposed to later layers, tend to learn similar representations across different datasets [45]. Moreover, when dealing with networks comprising a large number of layers, training the same network with the same dataset but different weight initialization only guarantees significant functional similarities in the first and last layers of the network [44].

Model stitching & Relative representation

So far, we have seen model similarity from a representational point of view, where we obtain the activation matrices for the two layers and then compare them either by their statistical or geometric properties. However, there is another approach when comparing two networks which we can call *functional similarity*. The main difference between functional similarity and representational similarity is that in the latter, we focus on the shape and characteristics of the data embedding, while in the former, we want to compare the "functional utility" of these embeddings for each network, i.e., asses how well would the model B achieve its task using the representations of network A [46].

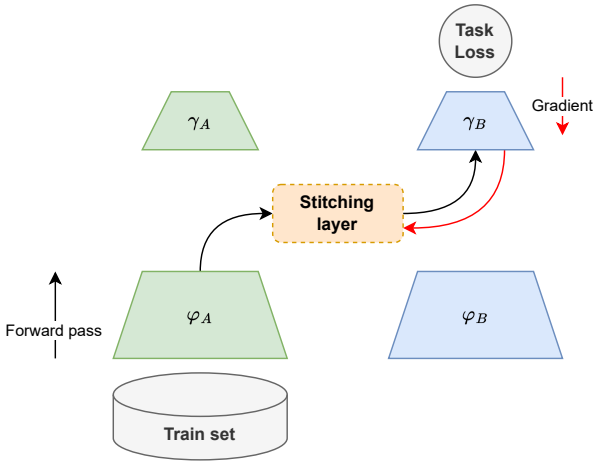


Figure 2.22: Model stitching using a stitching layer. Adapted from: [46]

To assess functional similarities between networks, researchers have proposed a technique called model stitching, which involves introducing an additional trainable stitching layer between the desired layers [46, 47, 48]. In this setup, we have two networks: $f_A = \gamma_A \circ \varphi_A$ and $f_B = \gamma_B \circ \varphi_B$, where they are decomposed into a *representation map* $\varphi_\bullet$ and a *task map* $\gamma_\bullet$. As depicted in Figure 2.22, we train a new layer l on our stitched model $f_{A;B} = \gamma_B \circ l \circ \varphi_A$. This can be achieved either by using the same task and labels employed for training network f_B, or by employing pseudo-labels generated by model B, similar to a teacher network in knowledge distillation.

Functional similarity can be quantified by comparing the relative accuracy achieved by $f_{A;B}$ and f_B. If the two networks exhibit high functional

similarity, we can conclude that their representations are compatible, allowing us to interchangeably use both $\gamma_B \circ l$ and γ_A as task maps for network A. This flexibility enables us to leverage cross-domain and cross-architecture model stitching, providing advantages in various applications.

Building upon the potential applications of model stitching in networks with compatible representations, a training setup has been proposed to enhance compatibility between networks [49]. In this setup, an additional auxiliary task head is incorporated, enabling regularization on the latent representation of both networks at corresponding layers. This regularization encourages higher functional similarity and is achieved through auxiliary tasks such as self-supervised prediction of input transformations and discrimination of common classes [49].

However, the existing methods mentioned above require training additional components to enhance representation compatibility, resulting in additional overhead. In this regard, we will present a technique called *relative latent representations*, which enables *zero-shot* model stitching. With this approach, we introduce a transformation in the latent spaces that increases compatibility without the need for training additional components.

In order to construct this transformation, we will assume that it is easier to obtain a compatible representation if we have higher representation similarity between the latent representations of the stitched layers*. Furthermore, there is evidence that some stochastic factors of our training, such as weight initialization, some hyperparameters, and data shuffling, will (most likely) result in *ε-similar representations* [4]†. Two representations $X, Y \subseteq \mathbb{R}^n$ are ε-similar if there exist a bijection $T : X \to Y$ st exists $\alpha \in \mathbb{R}^*$ for which $|d(T(x_1), T(x_2)) - \alpha \cdot d(x_1, x_2)| \leq \varepsilon$ for all $x_1, x_2 \in X$. Hence, these stochastic transformations will result in a relaxed version of an isometry up to a certain scale α.

Therefore we will build a transformation T_{rel} of our *(absolute) latent space representations* $\varphi_\bullet(x)$ so that our new encodings are invariant to ε-similarities. By doing so, we achieve higher representational and functional similarity.

*There is evidence of cases with low CKA and still having high functional similarity [46], i.e., we know that having high representation similarity it is not a necessary condition for high compatibility.

†The evidence is based on a few experimental results, so we can consider that this method takes this result almost as an additional assumption.

Let $\mathcal{X}$ denote our input space, and $\mathcal{A} = \{a_1, ..., a_k\}$ be a subset of $\mathcal{X}$ referred to as the set of *anchors*. For any similarity function sim, we define the relative representation of a point $x \in S$ with respect to $\mathcal{A}$ as follows:

Definition 2.2.4. Let $\varphi : \mathcal{X} \to \mathcal{Z}$ be our encoder. Then, the *relative representation* of $x \in \mathcal{X}$ w.r.t. $\mathcal{A}$ is

$$r = (\text{sim}(\varphi(x), \varphi(a_1)), \text{sim}(\varphi(x), \varphi(a_2)), ..., \text{sim}(\varphi(x), \varphi(a_k))) \in \mathbb{R}^k .$$

Remark. As we can see in Figure 2.23, this construction resembles representing a vector in coordinates of an orthonormal basis using the corresponding projection to each vector of the basis. However, in this case, we do not require the use of an inner product as a similarity function, nor do the anchors need to be linearly independent or orthogonal.

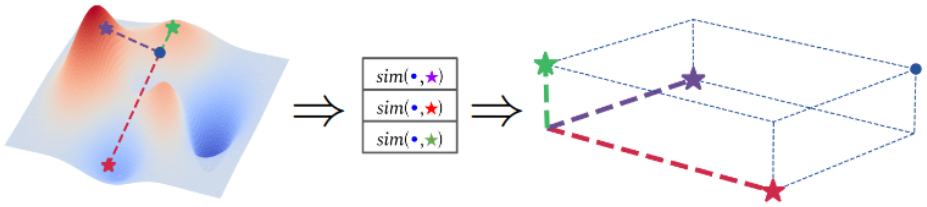
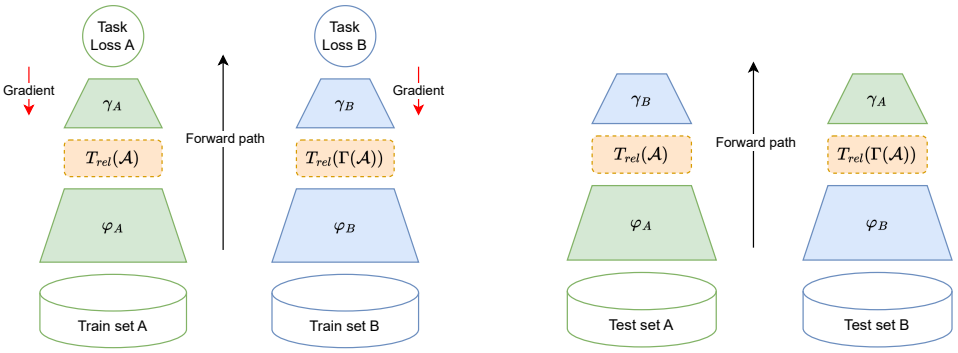


Figure 2.23: On the left, we have the anchors and input in the absolute representation, and on the right, we have the relative representation of the input. Source: [4]

Hence, if we use the cosine similarity, then this new latent space representation will be invariant to 0-similarity transformations. Furthermore, it has been observed that this representation not only increases network compatibility under the aforementioned stochastic factors but also enhances compatibility in the following scenarios:

- **Cross-domain model stitching:** We stitch two networks with the same architecture trained over two distinct datasets of the same semantic domain.
- **Cross-architecture model stitching:** We stitch two networks with different architectures trained over the same dataset.

Lastly, we can see in Figure 2.24 the standard training setup used in [4] for the case of cross-domain zero-shot model stitching. Here, we will use two pretrained encoders φ_A, φ_B, and train the networks $\tilde{f}_A = \gamma_A \circ T_{rel}(\mathcal{A}) \circ \varphi_A$



(a) Train w/ relative transformations. (b) Model stitching

Figure 2.24: Cross-domain model stitching training and testing

and $\tilde{f}_B = \gamma_B \circ T_{rel}(\Gamma(\mathcal{A})) \circ \varphi_B$ with both encoder weights frozen, using their corresponding domain-specific dataset. Note that in this case, where we have two distinct datasets, we fix the anchors in one of the networks (e.g., model A) and employ a partial correspondence mapping $\Gamma : \mathcal{P}(\mathcal{X}_A) \to \mathcal{P}(\mathcal{X}_B)$ to obtain the corresponding *parallel anchors* $\Gamma(\mathcal{A})$.

One potential application of this cross-domain model stitching procedure is in scenarios where multiple devices operate in different languages. By using only one trained classification head (e.g., trained on English corpus), we can achieve good performance when stitching this classification head with language-specific encoders. This can be really helpful from a computational perspective in scenarios where you would need to have frequent updates of your models since you would only need to do it for one of them.

2.3 Summary

In this chapter, we have first seen a brief introduction to Topological Data Analysis (TDA), where we have seen how we can use algebraic constructions such as Simplical Persistent Homology in order to asses some topological characteristics of the underlying manifold from which our dataset has been sampled from. Furthermore, in the specific case of n-dimensional simplicial homology, we have seen that the obtained topological features will indicate to us the number of n-dimensional holes.

Secondly, we have seen that recently there has been an increased interest in applying TDA in Machine Learning, a new field that we can call Topological

Machine Learning. One of the many ways that we can use our topological tools is to construct regularizers so that the network will obtain certain desired properties. Specifically, in this project, we will focus our attention on a topological regularization that will produce an increase in generalization capabilities by imposing a property called topological densification over our push-forward class-specific distributions [3].

Lastly, we have seen different techniques in order to assess the similarity between two layers of two (possibly distinct) networks. Furthermore, we observed that if our networks have high functional similarity, then we can do model stitching without a severe performance drop. Hence, there are several techniques in order to increase this type of similarity, thereby enhancing the compatibility of latent representations. In this project, we will focus on one called relative latent representation, which lets us do zero-shot stitching, i.e., it does not require us to train any additional parameters in order to either make our networks more compatible or to stitch them together [4].

Chapter 3

Methods

This chapter aims to provide an overview of the research methodology employed in this thesis.

3.1 Problem Formulation

The primary research question addressed in this study is as follows: *To what extent does the application of the topological densification technique [3] impact the classification performance of zero-shot stitching when utilizing relative latent representations [4]?*

The main objective, therefore, is to train multiple networks on a classification task using relative transformations and examine the potential advantages and disadvantages of incorporating the topological densification technique within this framework.

Prior to conducting this analysis, it was noted that the original paper on relative transformations [4] asserts that certain stochastic factors during training, such as weight initialization, hyperparameters, and data shuffling, are likely to produce ε-similar representations. However, the cited references in the original paper mostly just indicate that "well-performing" networks tend to exhibit similar latent representations based on metrics such as CCA and CKA*. Hence, an additional objective of this project has been to establish a theoretical explanation for why two networks could produce nearly isometric latent spaces. By achieving this, we can lend theoretical validity to the

*While a high CKA score can be obtained if we have isometric spaces, we might have cases with high CKA and not necessarily be ε-similar representations.

construction of the relative transformation and gain valuable insights into the latent space of our network.

3.2 Experiments

This section provides an overview of the experiments conducted to address the aforementioned objectives.

3.2.1 Latent Space Analysis

To assess whether two different initializations yield nearly isometric latent representations, we will explore both theoretical and empirical approaches.

Empirical analysis

Firstly, we will replicate the experiment outlined in the original paper on relative latent representations [4], where they demonstrate ε-similarity of the latent space in an autoencoder trained on MNIST. However, instead of relying solely on qualitative analysis through visual inspection of plots, we aim to evaluate the degree of similarity in the relative representations quantitatively.

For this purpose, we will employ the similarity metric known as Canonical Kernel Alignment (CKA), which, as previously discussed, is widely accepted as a standard measure for comparing latent spaces. In addition, since a high CKA score is not sufficient to guarantee ε-similarity, we will also utilize the following dissimilarity metric:

$$\text{minFrob}(A, B) = \min_{P \in \Pi} \left\| \frac{A}{\|A\|_F} - P \frac{B}{\|B\|_F} \right\|_F ,$$

where A and B represent the distance matrices of the respective latent spaces being compared, and Π denotes the set of permutation matrices. Notably, by normalizing the distance matrices, we achieve scale invariance. Furthermore, since having identical distance matrices up to permutation implies isometry between the spaces, obtaining a low value using this dissimilarity metric will suffice to indicate near ε-similarity.

Moreover, to enhance the qualitative analysis based on visual inspection of the alignment of both latent representation plots, we will incorporate Procustes Analysis. This method aims to identify the optimal linear transformation comprising translations, rotations, reflections, and uniform scaling, which

minimizes the L^2 distance (noted as Procustes error) between two shapes. Consequently, if we possess two ε-similar representations, we can determine the appropriate scale factor and orthogonal transformation that aligns the latent representations as closely as possible.

Remark. The orthogonal Procrustes problem can be efficiently solved by performing a singular value decomposition. Additionally, alternative versions of the Procrustes problem exist that incorporate different constraints, such as utilizing only permutation matrices or arbitrary linear maps.

The autoencoder network architecture comprises the following components:

- Encoder: It consists of a series of five convolutional layers, which enable the transformation of the initial 28x28x3 image size into a 4x4x56 feature vector. Subsequently, a flattening operation is applied, followed by a variable number of linear layers that will be adjusted during our analysis.
- Decoder: The decoder is designed to be symmetrical to the encoder architecture. It utilizes deconvolution layers to restore the image dimensions to their original form.

We will train this network for 300 epochs using an MSE reconstruction loss. Also, we will use an Adam optimizer with a learning rate of 10^{-3} and a one-cycle linear learning rate scheduler.

Lastly, since we want to use the relative transformation on a classification task, we will replicate the previous experiments using a simple CNN trained on the MNIST dataset. Hence, we will acknowledge if the results generalize to this specific task.

The network architecture for this purpose is identical to the encoder used in the autoencoder. In this case, the dimensions of the linear layers are fixed as 512-256-128-32-10. We will train this network for 20 epochs using Cross entropy loss and the same optimizer as before.

Theoretical analysis

In this subsection, we aim to provide a potential theoretical explanation for the occurrence of ε-similar representations when utilizing different initializations of the same network — an aspect that was not addressed in the original paper

on relative transformations.

As discussed in Section 2.2.2, utilizing the presented similarity metrics enables us to examine the scenarios in which our networks exhibit greater similarity. We recall that some of the most relevant properties are that wider networks and networks with lower generalization error tend to converge towards more similar solutions when trained on the same dataset [41]. These properties, along with others mentioned in Section 2.2.2, are used as justifications for the construction of the relative transformation. However, despite our understanding of achieving similar representations based on these metrics, the paper lacks a theoretical explanation for why we obtain ε-similar representations rather than equal representations under different linear or nonlinear mappings.

Instead, we will draw upon the insights presented in a recent work that investigates the symmetries in neural networks [50]. The central idea of this study revolves around the observation that certain activation functions induce similar symmetries in both weight space and latent representations. Specifically, we will focus on the symmetries generated by what we refer to as intertwiner groups.

Let $G_{\sigma_{n_i}}$ denote the set of invertible linear transformations that exhibit equivalent transformations before and after the nonlinear layer σ_{n_i}, i.e.,

$$G_{\sigma_{n_i}} \equiv \{A \in GL_{n_i}(\mathbb{R}) \mid \exists B \in GL_{n_i}(\mathbb{R}) \text{ s.t. } \sigma_{n_i} \circ A = B \circ \sigma_{n_i}\},$$

where $GL_n(\mathbb{R})$ represents the group of $n \times n$ invertible matrices [50].

We introduce the concept of an intertwiner group associated with the activation function σ_n through the following definition:

Definition 3.2.1. Let $\sigma(I_n)$ be invertible, then we call G_{σ_n} the *intertwiner group of the activation* σ_n.

Furthermore, we establish the existence of a homomorphism that corresponds to the equivalent invertible transformation after the activation function:

Lemma 3.2.1 ([50])**.** *Let $\sigma(I_n)$ be invertible, and for each $A \in GL_n(\mathbb{R})$ define $\phi_\sigma(A) = \sigma(A)\sigma(I_n)^{-1}$. Then G_{σ_n} is a group, $\phi_\sigma : G_{\sigma_n} \to GL_n(\mathbb{R})$ is a homomorphism such that $\sigma_n \circ A = \phi_\sigma(A) \circ \sigma_n$.*

Under mild assumptions on the activation function, it is shown in [50] that all elements of the intertwiner group can be represented as PD, where $P \in \Sigma_n$

and D is a diagonal matrix. Additionally, Table 3.1 illustrates that for widely used activation functions, the homomorphism ϕ_σ is equivalent to the identity.

Acivation	G_{σ_n}	$\phi_\sigma(A)$
$\sigma(x) = x$ (identity)	$GL_n(\mathbb{R})$	A
$\sigma(x) = \frac{e^x}{1+e^x}$	Σ_n	A
$\sigma(x) = \text{ReLU}(x)$	PD, w/ D positive entries	A
$\sigma(x) = \text{LeakyReLU}(x)$	As ReLU if negative slope $\neq 1$	A
$\sigma(x) = \text{GeLU}(x)$	Σ_n	A
$\sigma(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ (RBF)	PD, w/ D entries in $\{\pm 1\}$	$\text{abs}(A)$
$\sigma(x) = x^d$ (polynomial)	PD, w/ D non-zero entries	$A^{\odot d}$

Table 3.1: Explicit descriptions of G_{σ_n} and ϕ_σ for seven different activations. Here $P \in \Sigma_n$ is a permutation matrix, D is a diagonal matrix, and $A^{\odot d}$ denotes the entrywise dth power. CC BY Source: [50]

Hence, now we know that the activation functions have commutative properties w.r.t some transformations that resemble the 0-similarities: permutations are a subset of all possible isometries, and isotropic scaling is a case where all elements in the diagonal matrix are equal to the same scaling factor.

However, the importance of the intertwiner group will come from the following proposition, which, informally, tells us that it will produce symmetries in the weight space that propagates into symmetries in the latent representations [50].

Proposition 3.2.1 ([50])**.** *Suppose $A_i \in G_{\sigma_{n_i}}$ for $1 \leq i \leq k-1$, and let*

$$\widetilde{W} = (A_1 W_1, A_1 b_1, A_2 W_2 \phi_\sigma(A_1^{-1}), A_2 b_2, \ldots, W_k \phi_\sigma(A_{k-1}^{-1}), b_k)$$

Then, as functions, for each m

$$f_{\leq m}(x, \widetilde{W}) = \phi_\sigma(A_m) \circ f_{\leq m}(x, W),$$
$$f_{>m}(x, \widetilde{W}) = f_{>m}(x, W) \circ \phi_\sigma(A_m)^{-1},$$

where $f_{\leq m}$ and $f_{>m}$ represent the truncations of the network before and after layer m, respectively. In particular, $f(x, \widetilde{W}) = f(x, W)$ for all $x \in \mathbb{R}^{n_0}$.

Thus, we have identified a potential theoretical reason for having a variation of ε-similar representations when we have two random initializations of the same network. Nevertheless, we acknowledge that this property produced by the activation function is not the whole explanation of why we observe ε-similar representations, and further discussion regarding our hypothesis of the potential causes will be presented in Section 5.1.

Based on these findings, we will modify the relative transformation to make it invariant to the intertwiner group actions induced by common activation functions, as well as ε-similarities (with small ε). Thereof, we will know that if the symmetry comes from the activation function, we will produce a good composable representation [50, Theorem 4.2].

Let $\mathcal{X} = \mathbb{R}^d$ denote the input space, and $\mathcal{A} = \{a_1, ..., a_k\}$ be a subset of $\mathcal{X}$ referred to as the anchor set. For the cosine similarity, we define the *robust relative representation* of batch $\mathcal{B} = \{x_1, ..., x_n\} \subset \mathcal{X}$ with respect to $\mathcal{A}$ as follows:

Definition 3.2.2. Let $\varphi : \mathcal{X} \to \mathcal{Z} = \mathbb{R}^m$ be our encoder, and $\mathbb{A} \in \mathbb{R}^{d\times k}$, $\mathbb{B} \in \mathbb{R}^{d\times n}$ the matrix representation of $\mathcal{A}$ and $\mathcal{B}$. Then, the *robust relative representation* of $\mathcal{B} \subset \mathcal{X}$ w.r.t. $\mathcal{A}$ is

$$\hat{T}_\varphi(\mathcal{B}, \mathcal{A}) = \left(\widehat{\varphi(\mathbb{A})} D_{\mathbb{A}}\right)^T \left(\widehat{\varphi(\mathbb{B})} D_{\mathbb{B}}\right) \in \mathbb{R}^{k\times n},$$

where

$$D_{\mathbb{A}} = \text{Diag}\left(\frac{1}{\sum_{i=1}^m \widehat{\varphi(\mathbb{A})}_{i,1}^2}, ..., \frac{1}{\sum_{i=1}^m \widehat{\varphi(\mathbb{A})}_{i,k}^2}\right),$$
$$D_{\mathbb{B}} = \text{Diag}\left(\frac{1}{\sum_{i=1}^m \widehat{\varphi(\mathbb{B})}_{i,1}^2}, ..., \frac{1}{\sum_{i=1}^m \widehat{\varphi(\mathbb{B})}_{i,n}^2}\right),$$

and $\widehat{\varphi(\mathbb{A})}$ and $\widehat{\varphi(\mathbb{B})}$ represent the respective BatchNorm mean and variance standardizations of the anchor and batch images (without the learnable affine transformation). When the batch and the encoder are implied, we can denote this transformation by $\hat{T}_{rel}(\mathcal{A})$.

Note. To differentiate between the two "diag" operators, we will denote $\text{Diag}(x)$ as the map that generates a diagonal matrix from a vector, and $\text{diag}(X)$ as the map that extracts the diagonal of a matrix.

Proposition 3.2.2. *Let $A_i \in G_{\sigma_{n_i}}$ for $1 \leq i \leq k-1$, $\phi_\sigma = id$, and consider*

$$\widetilde{W} = (A_1 W_1, A_1 b_1, A_2 W_2 A_1^{-1}, A_2 b_2, \ldots, W_k A_{k-1}^{-1}, b_k)$$

For each m, we have

$$\hat{T}_{\tilde{f}_{\leq m}}(\mathcal{B}, \mathcal{A}) = \hat{T}_{f_{\leq m}}(\mathcal{B}, \mathcal{A}),$$

where $\tilde{f}_{\leq m} \equiv f_{\leq m}(x, \widetilde{W})$ and $f_{\leq m} \equiv f_{\leq m}(x, W)$.

Proof. First, let's consider how BatchNorm is affected by an intertwiner group action $PD \in G_{\sigma_m}$:

- Centering: $PDX - PDX\frac{\mathbf{1}\mathbf{1}^T}{n} = PD\left(X - X\frac{\mathbf{1}\mathbf{1}^T}{n}\right) = PD\tilde{X}$.
- Normalization:

$$\begin{aligned}
&\text{Diag}\left(\text{diag}(PD\tilde{X}\tilde{X}^T D P^T)^{-1/2}\right) PD\tilde{X} \\
&= \text{Diag}(PD^{-1}\text{diag}(\tilde{X}\tilde{X}^T)^{-1/2})PD\tilde{X} \\
&= \text{Diag}(P\text{diag}(\tilde{X}\tilde{X}^T)^{-1/2})P\tilde{X} \\
&= P\,\text{Diag}(\text{diag}(\tilde{X}\tilde{X}^T)^{-1/2})\tilde{X} = P\hat{X}.
\end{aligned}$$

By Proposition 3.2.1, we know that $f_{\leq m}(x, \widetilde{W}) = PDf_{\leq m}(x, W)$, so $\widehat{f_{\leq m}(x, \widetilde{W})} = P\widehat{f_{\leq m}(x, W)}$. Furthermore, since the L^2 norm is permutation invariant, we have

$$\text{Diag}\left(\frac{1}{\left\|P\widehat{f_{\leq m}(X)}_1\right\|}, \ldots, \frac{1}{\left\|P\widehat{f_{\leq m}(X)}_n\right\|}\right) = \text{Diag}\left(\frac{1}{\left\|\widehat{f_{\leq m}(X)}_1\right\|}, \ldots, \frac{1}{\left\|\widehat{f_{\leq m}(X)}_n\right\|}\right).$$

Hence,

$$\hat{T}_{\tilde{f}_{\leq m}}(\mathcal{B}, \mathcal{A}) = D_{\mathbb{A}}\widehat{f_{\leq m}(\mathbb{A})}^T P^T P \widehat{f_{\leq m}(\mathbb{B})} D_{\mathbb{B}} = \hat{T}_{f_{\leq m}}(\mathcal{B}, \mathcal{A}).$$

□

Therefore, according to the proposition above, the new relative transformation is invariant to the actions of the intertwiner groups induced by common activation functions (see Table 3.1). Furthermore, it will preserve the invariance to 0-similarities, as indicated by the following corollary.

Corollary. *If we use the standard relative transformation (without batch normalization), the equality* $\hat{T}_{\tilde{f}_{\leq m}}(\mathcal{B}, \mathcal{A}) = \hat{T}_{f_{\leq m}}(\mathcal{B}, \mathcal{A})$ *only holds when* $D = \text{Diag}(\lambda_1, ..., \lambda_d)$ *such that* $|\lambda_i| = |\lambda_j|$ *for all* i, j.

3.2.2 Topological regularization with relative transformation

Due to the selected topological regularization method [3], which requires a supervised classification setup for its application, it becomes necessary to fine-tune the pretrained encoder. This represents a deviation from the zero-shot setup described in [4], where only the decoder is fine-tuned using the relative representation while the encoder remains frozen. Therefore, this part of the research encompasses two primary objectives:

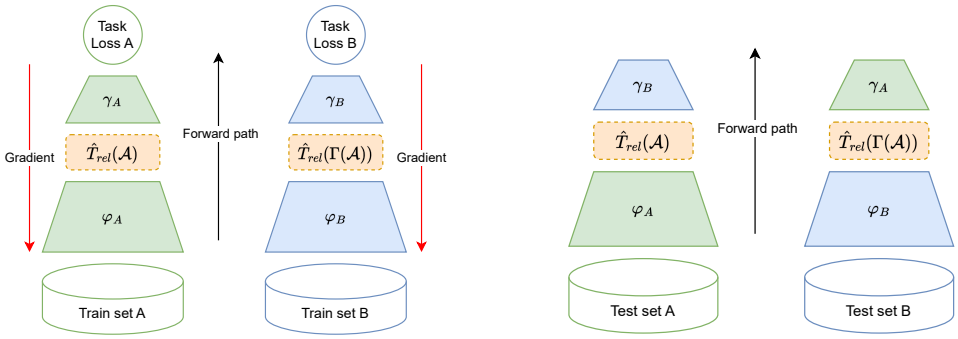
- Analyse the performance of zero-shot stitching after fine-tuning the encoder and decoder with relative representation.
- Analyze the performance when training with the topological regularization and the relative representation.

Full fine-tuning analysis

We saw in the literature review the standard training setup used in [4] for the case of cross-domain zero-shot model stitching. As we can see in Figure 2.24, they used two pretrained encoders φ_A, φ_B, and trained the networks $\tilde{f}_A = \gamma_A \circ T_{rel}(\mathcal{A}) \circ \varphi_A$ and $\tilde{f}_B = \gamma_B \circ T_{rel}(\Gamma(\mathcal{A})) \circ \varphi_B$ with both encoder weights frozen, using their corresponding domain-specific dataset.

However, in our proposed approach (depicted in Figure 3.1), we also allow the gradient to propagate through the encoder.

To be more precise, we will conduct an ablation study based on the experiment presented in [4] focusing on multi-lingual Amazon review classification. Due to time constraints, we will focus on cross-lingual stitching, leaving cross-architecture stitching as future work. As the method has been reported to be robust to anchor selection, we will utilize the "translated" modality for parallel anchors. In this modality, we first select a number of anchors from the English dataset equal to the dimensionality of the latent space (768 dimensions) and then translate them to other languages using the Google Translator API.



(a) Train w/ relative transformations. (b) Model stitching

Figure 3.1: New cross-domain model stitching training and testing

However, upon inspecting the provided code from the original relative representation project, we decided to modify the training procedure and the network architecture. This choice is motivated by the identification of several issues and potential methodological concerns:

- *(bug)* In the case of the 5-class dataset, the code utilized only four classes during both training and testing, despite the network output being five-dimensional. Consequently, a fair assessment of the training performance on the original dataset was not obtained.
- *(methodological)* When loading the Spanish model, the code failed to restore the dropout probability. This inconsistency in the regularization between the Spanish and other models could arise if the intention is not to freeze the encoder.
- *(methodological)* The standard Transformer library suggests adding a pooler layer to the classification head while disregarding the one provided in the pretrain model. However, the original code left the pooler layer in the encoder frozen.
- *(methodological)* An L^2 normalization is applied after the relative transformation. The authors' rationale for this step was that it improved result robustness. However, we believe that this normalization hinders the analysis of the ablation study of the transformation.
- *(methodological)* The classification head used SiLU activation and a modified version of InstanceNorm that acted as BatchNorm. Once again, the authors claimed that this architecture was selected based

on empirical results. Nevertheless, we suggest employing a more "standard" classification head to ensure that the claimed results can be achieved without relying on these less commonly used architectural components.

Hence, the proposed new methodology is:

- **Data:** We will retain the same dataset as the original paper, which involves randomly subsampling 25% of the Amazon Reviews dataset [51]. As mentioned earlier, we will use the "translated" setup to construct the parallel anchors. Training will be conducted on both the coarse-grained version (predicting whether a review is greater or smaller than 3 stars) and the fine-grained version (predicting ratings from 1 to 5 stars).
- **Architecture:** As in the original paper, we will use as encoders the following RoBERTa [52] models: "`roberta-base`" for English, "`PlanTL-GOB-ES/roberta-base-bne`" for Spanish, and "`ClassCat/roberta-base-french`" for French. Also, the pooler layer will be removed, and the dropout probability will be initialized to 0.1.

Listing 3.1: Decoder Pytorch class

```
pooler = nn.Sequential(
            nn.Linear(hidden_size, hidden_size),
            nn.Tanh(),
            nn.Dropout(hidden_dropout_prob)
        )

decoder = nn.Sequential(
            nn.Linear(hidden_size, hidden_size),
            nn.GELU(),
            nn.Dropout(hidden_dropout_prob),
            nn.Linear(hidden_size, hidden_size),
            nn.GELU(),
            nn.Dropout(hidden_dropout_prob),
            nn.Linear(hidden_size, num_labels)
        )
```

The architecture used for the classification head can be seen in Listing 3.1. The number of linear layers remains the same as in the original code, and the standard RoBERTa pooler layer from the Transformer repository is utilized. Additionally, we will employ the same dropout probability as used for the encoder.

Lastly, as shown in Figure 3.1, we will use the new *robust relative transformation* developed in the previous section.

- **Training:** Since we want to compare the performance between the fully fine-tuned model against the frozen encoder version, we will have two training setups. For the fully fine-tuned case, we will use layer-wise learning rate decay on the encoder parameters with an initial learning rate of $1.75 \cdot 10^{-4}/(num\_labels)$ and decay rate of 0.65. The reason is that we will want to apply the topological regularization in the subsequent experiments, and for this type of loss, we do not need to have significant changes in the initial layers.

  The classification head parameters will have an initial learning rate of $10^{-3}/(num\_labels)$ for both setups. Furthermore, we will use an AdamW optimizer with weight decay 0.01, along with the "`constant_schedule_with_warmup`" scheme, incorporating a warmup phase that spans 10% of the first epoch*. We will train the networks for five epochs in the fine-grained and three in the coarse-grained cases.

  Due to the size of the transformer models and the reviews, using large batch sizes will exceed the available VRAM. Therefore, we will address this issue using a common technique called gradient accumulation. This involves using a batch size of 16 but updating the network gradients only every num_labels steps. The corresponding gradients are accumulated during these steps, effectively simulating a batch size of $num\_labels \times 16$.

  Lastly, for the relative case, we will update the anchor embeddings every $100 \times num\_labels$ steps. This optimization reduces computational costs since the image of 768 samples does not need to be computed at every step. Furthermore, given the employment of layer-wise learning rate decay, we do not anticipate rapid changes in the encoder. Therefore, recomputing the anchor embeddings at every step is unnecessary†.

*These optimization strategies are considered standard practice when fine-tuning large Transformer models [53].

†Similar to DQN in RL [54] where we periodically freeze the target network, fixing the anchors' embedding every few steps can help stabilize the training. This is because a slight change in the anchor's image could result in critical differences in the geometry of the post-relative latent space.

- **Testing:** We will evaluate the performance of the stitching in both the absolute (no relative transformation during training) as well as the relative (using the relative transformation) cases. Therefore, as shown in Figure 3.1b, we will match the encoder language with the one used for the testing dataset, while the decoder can come from any language-specific trained network.

  Following the original paper, we will use Mean Absolute Error (MAE), FScore, and Accuracy (Acc) as our metrics.

Topological regularization analysis

As we have seen in Section 2.2.1, the original topological densification paper proposes the following batch construction: We construct each mini-batch, $\mathcal{B}$, as a collection of n sub-batches, i.e., $\mathcal{B} = (\mathcal{B}_1, ..., \mathcal{B}_b)$. Each sub-batch consists of n samples from the same class, so we end up having an effective batch size of $b \cdot n$.

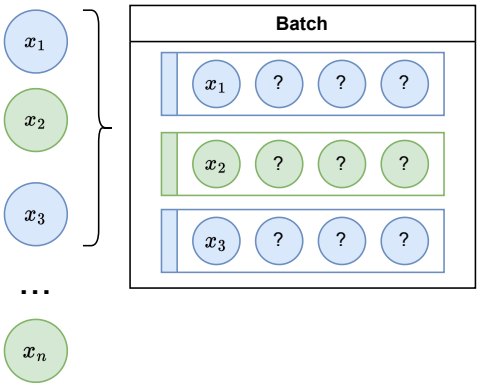


Figure 3.2: Hofer et al.'s original dataloader used for the topological regularization with $b = 3$ and $n = 4$.

However, when we analyzed the actual implementation, we observed that the batch construction was done as depicted in Figure 3.2:

1. We generate b-sized batches with random shuffling over the original dataset.

2. For each point x_i in the batch, we expand the sub-batches by sampling with replacement $n - 1$ elements of the same class as x_i.

We identified two major methodological issues with the utilized dataloader. Firstly, in the presence of significant class imbalances, there is

a risk of catastrophic forgetting, as prolonged training steps may occur without encountering the less-represented class. Secondly, this method is computationally intensive since training requires processing the original dataset n times in a single epoch.

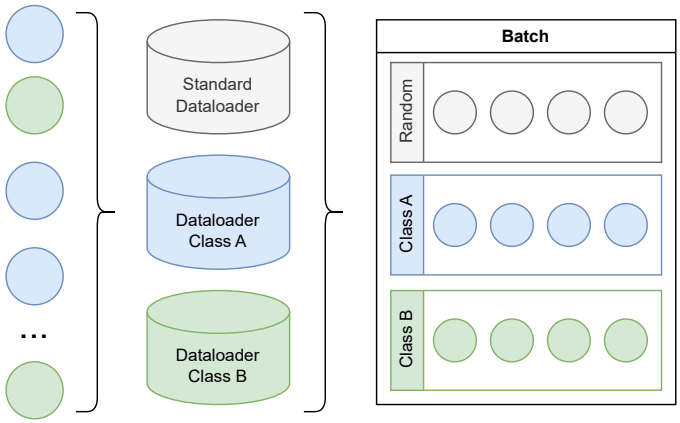


Figure 3.3: New dataloader used for the topological regularization with $b = 3$ and $n = 4$.

To address these concerns, we propose a new dataloader that follows the high-level description in the original paper while potentially mitigating the aforementioned issues. Figure 3.3 illustrates the construction of the new dataloader:

1. For each class, we create a separate dataloader containing all samples from that class. Additionally, we create a standard dataloader containing all the samples. These dataloaders utilize shuffling and have a batch size of n.

2. We aggregate one batch from each dataloader, resulting in a mini-batch consisting of $num\_classes + 1$ sub-batches. In the case of significant class imbalances, we can iterate through the less-represented class dataloaders until all batches from the most-represented class dataloader have been processed.

This new dataloader preserves the sub-class structure necessary for correctly applying the topological regularization loss. Furthermore, its computational overhead is comparable to training directly with the standard dataloader. However, we acknowledge that this new dataloader may potentially encourage the overfitting of the less-represented classes in

the presence of significant class imbalances. To counteract this, data augmentation techniques should be employed as a preventive measure. Nonetheless, as depicted in Figure 3.4, our datasets exhibit minimal class imbalances, thus minimizing concerns in this regard.

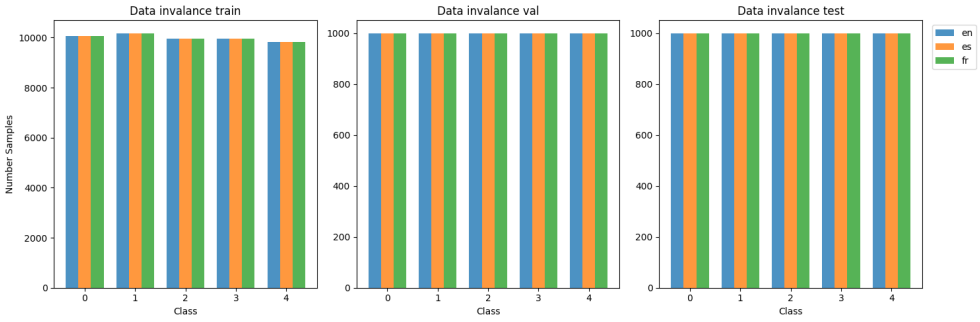


Figure 3.4: Class imbalance analysis of the fine-grained datasets w/ 25% subsampling on the training set.

In Figure 3.3, we introduce an additional random sub-batch in each mini-batch. This extra sub-batch serves as a solution to address an issue encountered when using gradient accumulation with the "biased" dataloaders proposed in the original paper.

Although we previously stated that gradient accumulation should simulate passing the entire mini-batch at once, this assumption only holds true when network components' performance does not depend on the "input quality". In our case, we observed that the mean and variance standardization of the robust relative transformation becomes biased with the new dataloader when using gradient accumulation, as we only pass elements from the same class. This issue was not identified in the original paper since the models used were significantly smaller than the RoBERTa models we employ, allowing them to pass the entire mini-batch in each step (without gradient accumulation).

To resolve this problem, we will take the following steps:

1. Freeze the parameters of the Linear and LayerNorm modules in the model and set BatchNorm1d and LayerNorm to training mode.

2. Perform a forward pass of the "random" mini-batch and compute the cross-entropy loss.

3. Unfreeze the parameters of the Linear and LayerNorm modules in the model and set BatchNorm1d and LayerNorm to eval mode.

4. Pass the remaining $num\_classes$ mini-batches and compute the cross-entropy loss and the topological regularization.

By adopting this approach, we can utilize the running averages of the mean and variance computed on unbiased sub-batches during the forward pass of the biased sub-batches.

In our ablation studies, we will utilize a classification head consisting of only one linear layer without even including a pooling layer. While there is no theoretical constraint on the classification head for applying the topological densification, we have two primary methodological justifications for this choice.

Firstly, we could see a more expressive classification head as an implicit regularization scheme. By using a simpler decoder, we intentionally reduce variance and increase bias, allowing the model to overfit the training data. Therefore, we establish a setup that requires the aid of explicit regularization techniques, such as topological densification, to reduce the generalization error.

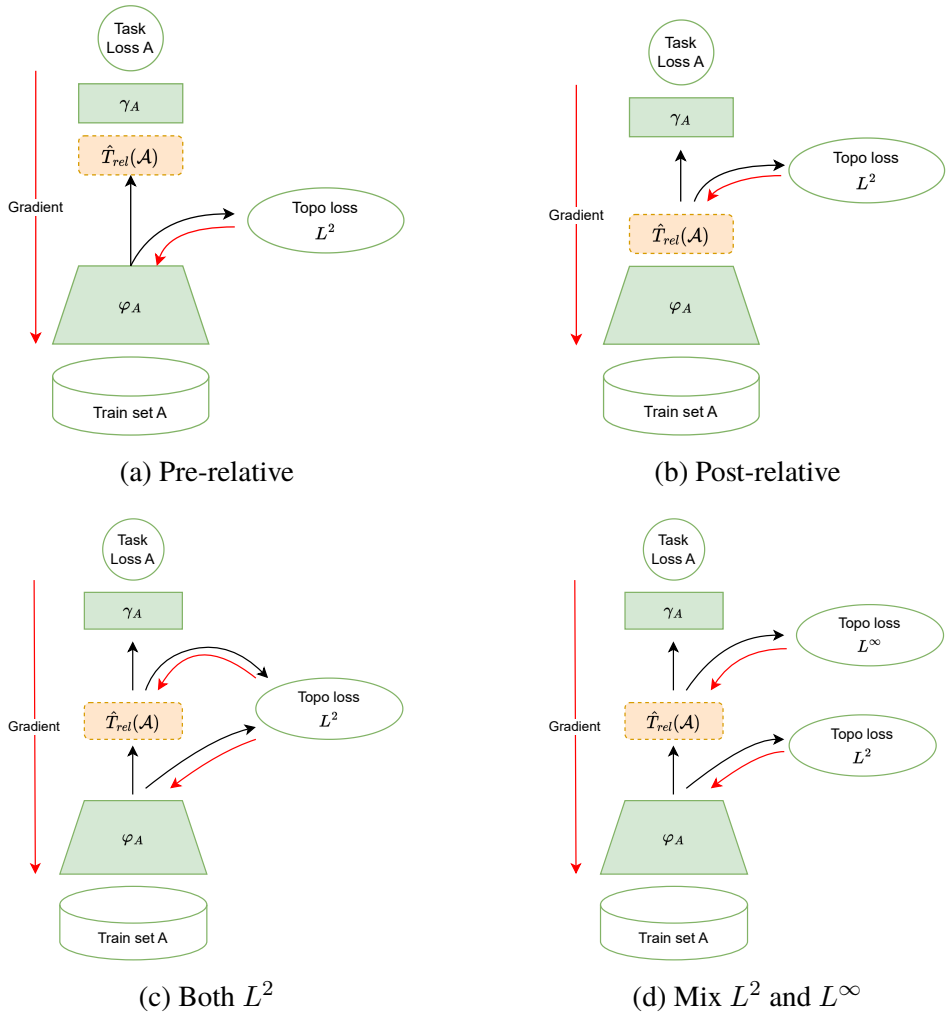
Secondly, the objective of the topological regularization loss is to encourage a more favorable configuration of the latent space for the classification task. However, if we employ a complex classification head, we may not require the assistance of these losses, as the model could potentially fit more intricate decision boundaries.

It is important to note that this setup is not unrealistic, as popular Transformer networks like Vision Transformers (ViTs) [55] often employ one or two linear layers (depending on the specific implementation) for their classification heads.

Figure 3.5 presents the different topological regularization setups we will explore when using the relative transformation. Firstly, we will analyze the case of applying topological regularization either before (pre-relative) or after (post-relative) the relative transformation. Additionally, based on empirical results discussed in Chapter 4, we will also apply L^2 topological regularization using the same β hyperparameter both before and after the relative transformation simultaneously.

Lastly, as shown in Figure 3.5d, we also use the L^∞ metric to construct the Vietorus-Rips filtration used for the post-relative topological regularization. There are two methodological justifications for using this metric:

1. TDA methods analyze and utilize the topological features of the



(a) Pre-relative

(b) Post-relative

(c) Both L^2

(d) Mix L^2 and L^∞

Figure 3.5: Different topological regularization setups while using relative transformation.

underlying data manifold. However, they are also geometric methods, and better results can be achieved by employing a distance metric that aligns with the geometry of the data manifold. Since the relative transformation codomain is $[-1, 1]^k$, where k is the number of anchors, L^∞ open balls are a better fit than L^2 balls.

2. Determining the optimal value of the β hyperparameter in a high-dimensional latent space can be challenging, even though we understand its meaning. In the original paper, the best β value was obtained through extensive hyperparameter tuning.

To provide an intuition for selecting β, we developed a new heuristic based on the empirical distribution of the 0-homology death times and the maximum death times of the test set*, as depicted in Figure 3.6. With this understanding of the topology without any imposed regularization, we can make more informed choices for different β values rather than relying on random selection.

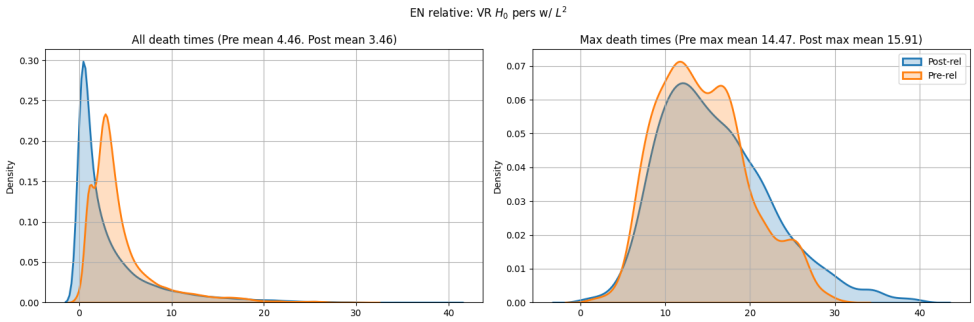


Figure 3.6: Example of the death time distributions.

Nevertheless, using the L^∞ metric allows for a more concise interpretation of this hyperparameter. In an optimal scenario where each cluster has only one anchor at its centroid, the maximum coordinate of the relative transformation corresponds to the cosine of the angle between a point and the center of its cluster. Therefore, we can argue that the β hyperparameter, when using L^∞, relates to the optimal spread of the clusters in terms of angle, given by $\beta = 1 - \cos(\pi\theta/180)$.

Lastly, we want to remark that due to time constraints, we will focus this analysis on a random 1% sub-sample of the fine-grained dataset. Furthermore, we will train our networks for 40 epochs to ensure that the topological regularization prevents overfitting. We will also use a "`cosine_schedule_with_warmup`" scheduler on the learning rate and a linear cyclic scheduler for the weight λ component of the regularization loss. This cyclic scheduler has been previously used in VAEs to obtain a better synergy between the reconstruction loss and the KL regularization [56].

*These plots might resemble the Persistence Images presented in Section 2.2.1. However, in this case, we do not have a 2D image because by using $\mathrm{H}_0(\mathrm{VR})$, all the points lie in the line $x = 0$. Furthermore, instead of a heat map, we plot the density estimation.

Writing Assistant Software

While writing this Master Thesis, we used Grammarly to analyze the grammar and style. Furthermore, we used ChatGPT to rewrite some fragments to improve the clarity and flow of the ***already written text***. Hence, Generative Models were used for writing assistance rather than to generate new text from scratch.

Chapter 4

Results and Analysis

In this chapter, we will present the project's main results and discuss them. The code, as well as additional results, can be found in the following GitHub repository: https://github.com/AGarciaCast/Topo_Reg_Relative_Rep

4.1 Latent space similarity analysis

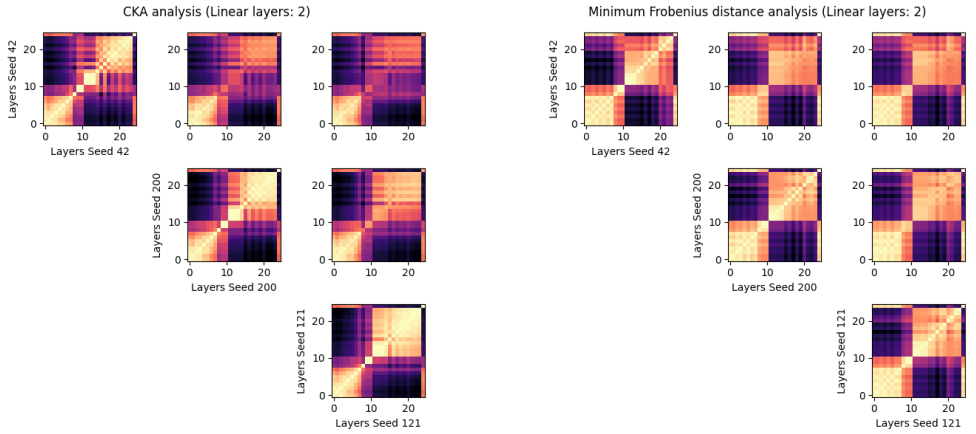
4.1.1 Autoencoder

One linear layer and two-dimensional latent space

In accordance with the experiment conducted in the original relative latent representation paper [4], we will set the dimension of the latent space to two in order to visualize it without dimensionality reduction. Initially, we will explore the case where just one linear layer is used to transition from the convolutional feature map to the two-dimensional space.

As described in Section 3.2.1, we will employ the CKA metric to compare the latent representations of this network across different initializations. Figure 4.1a displays the discrepancies between the representations generated by the encoder and decoder. Notably, there are instances, such as seed 42 vs. seed 121, where the differences in representations are more pronounced than those between seed 121 and seed 200. We attribute this phenomenon to the model's need to find the optimal linear transformation for reducing an 896-dimensional space to a 2-dimensional one. Considering that the optimal solution in terms of MSE corresponds to the PCA embedding, we believe that

the variation in representations may arise from using a different basis than the eigenvectors for dimensionality reduction. Consequently, when encoded using distinct approximations of the principal component, the decoder latent spaces will differ.



(a) CKA analysis

(b) Min. Frobenius norm

Figure 4.1: Numerical latent space analysis for the autoencoder with linear layers: 2

Figure 4.1b illustrates that the previously obtained results using CKA align with those derived from measuring the minimum Frobenius distance between the L^2 distance matrices.

Furthermore, we will employ Procrustes analysis to visually validate the similarity of these representations by examining their overlap after applying the optimal isometry up-to-scale transformation in terms of MSE. The interpretation of the Procrustes analysis presented in Figure 4.2 is as follows:

1. The diagonal elements represent the original 2-dimensional latent space generated by the encoder, with each class indicated by a distinct color.

2. Column-wise, we fix the original latent representation as our reference and plot it in the background in grey.

3. Row-wise, we apply the optimal Procrustes transformation to minimize the Procrustes error with respect to the fixed column reference. The transformed space is then colored.

4. We show the Procrustes error, as well as the Frobenius distance between the applied optimal orthogonal Procrustes transformation and

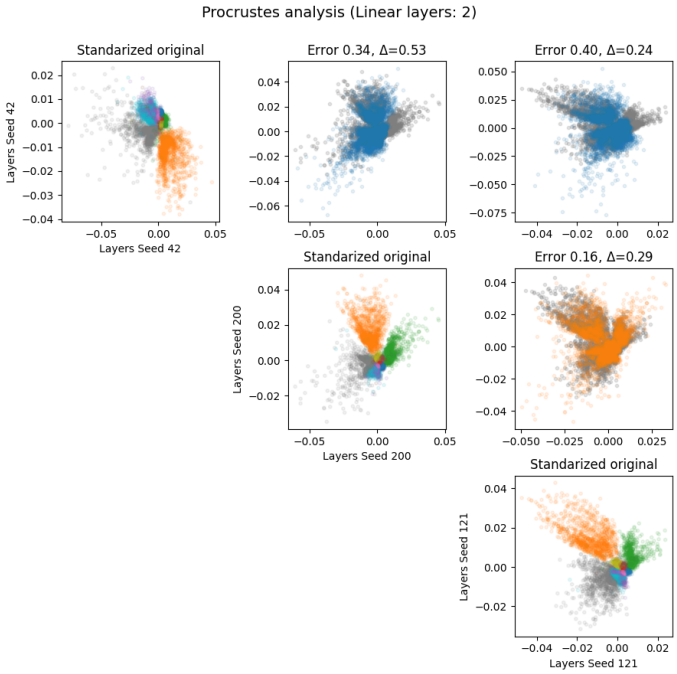


Figure 4.2: Procrustes analysis for the autoencoder with linear layers: 2

the optimal permutation Procrustes transformation (denoted by Δ). The latter indicates the degree to which we deviate from a theoretical intertwiner group action.

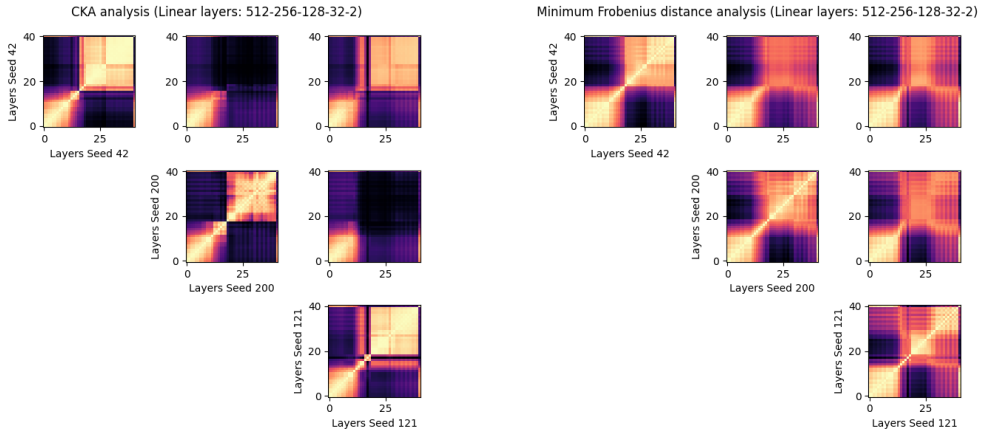
Hence, Figure 4.2 reveals that cases with higher numerical similarity correspond to smaller Procrustes errors. Even when higher Procrustes errors are observed, some similarities can still be observed between the two overlapping representations, suggesting the potential use of other metrics commonly employed in image segmentation, such as Jaccard or Dice coefficients.

Regarding the distance to the optimal permutation matrix, we lack sufficient evidence to confirm whether it comes from the intertwiner group or is simply an artifact resulting from finding sub-optimal approximations of principal components.

Note. In Appendix A.1, we present the numerical and Procrustes analysis results for two additional seeds for all the setups utilized in Section 4.1. It is important to note that these supplementary results align with those presented using only three seeds.

Multiple linear layers and two-dimensional latent space

In this case, we will employ the following Multi-Layer Perceptron (MLP) architecture to transition from the convolutional feature map to the 2-dimensional space: 512-256-128-32-2. This choice ensures that any similarities observed in the linear layers are not solely attributed to the inductive bias of the convolutional layers.



(a) CKA analysis

(b) Min. Frobenius norm

Figure 4.3: Numerical latent space analysis for the autoencoder with linear layers: 512-256-128-32-2

Figures 4.3a, A.4 show that in this case, we can obtain somewhat different representations in terms of CKA. However, the min Frobenius distance (Figures 4.3b, A.5) indicate that there might be a slight similarity. So this gives us the understanding that CKA can be more "pessimistic" than the Frobenius distance. Nevertheless, we still see a correlation between the results of the CKA plots and the Frobenius ones.

We believe that the reason for obtaining more distinct representations when employing this architecture lies in the multiple ways that exist to incorrectly map an 896-dimensional space to a 2-dimensional one using a highly expressive non-linear function. Consequently, Figure 4.4 illustrates collapsed class clusters and excessive spread in certain cases. Notably, some of these issues could potentially be mitigated by utilizing a Variational Autoencoder (VAE) since it applies proper regularization to the encoded space.

Lastly, we can see in Figure 4.4 that the case where we obtained the best similarity in terms of CKA, we also obtain the smallest Procrustes error. Furthermore, we can observe that our findings based on the Frobenius norm were not entirely inaccurate, as there are local resemblances between the latent spaces (e.g., dispersion patterns in the blue and orange classes).

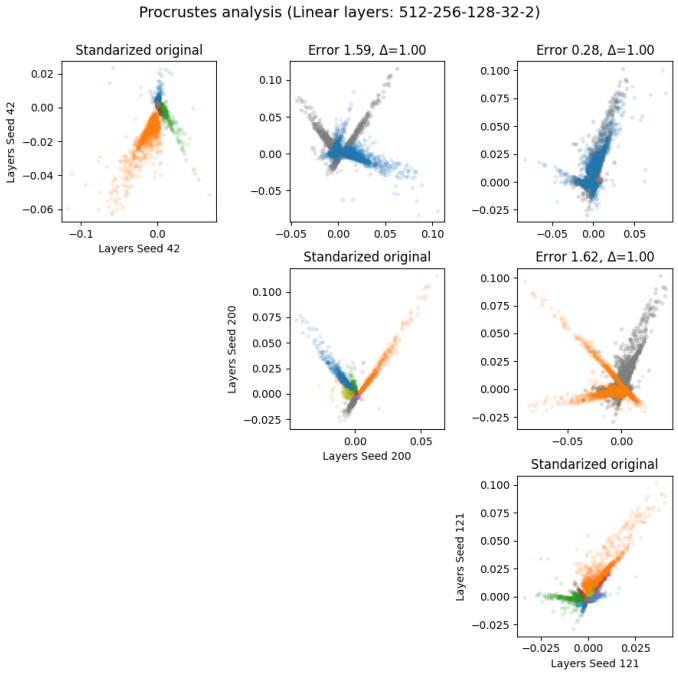


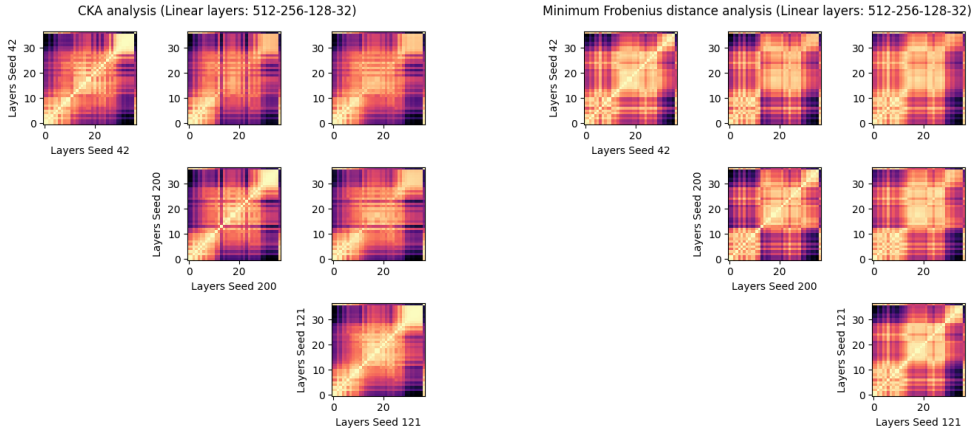
Figure 4.4: Procrustes analysis for the autoencoder with linear layers: 512-256-128-32-2

Multiple linear layers and high-dimensional latent space

We will explore the final configuration of the Autoencoder, which utilizes multiple linear layers to transition from the convolutional feature map to a 32-dimensional space: 512-256-128-32. This choice is motivated by the assertion made in [42] that "well-performing" networks yield more similar representations. Hence, we aim to investigate whether these networks also tend to produce representations that are ε-similar.

Figure 4.5 supports this claim, as we observe high similarities in terms of both CKA and the Frobenius norm. We believe that this outcome may be attributed to the fact that autoencoders with a higher-dimensional latent space

do not have such a strong information bottleneck.



(a) CKA analysis (b) Min. Frobenius norm

Figure 4.5: Numerical latent space analysis for the autoencoder with linear layers: 512-256-128-32

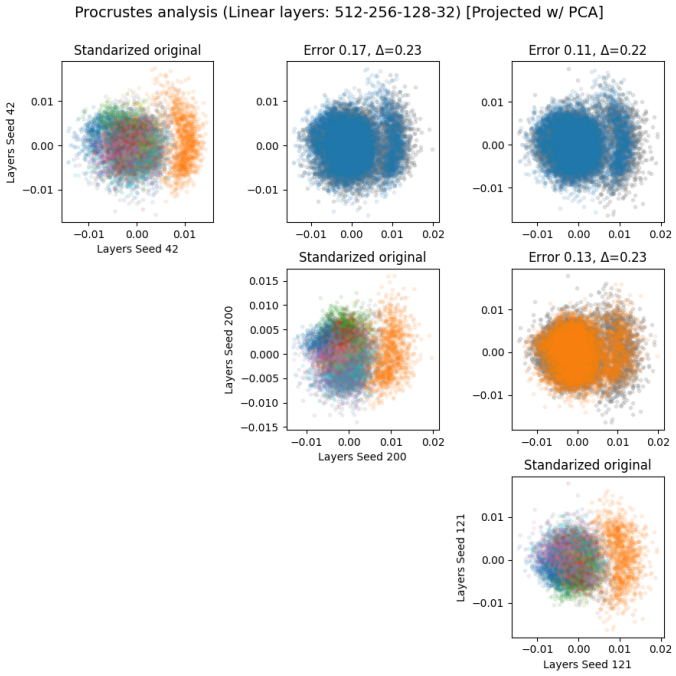
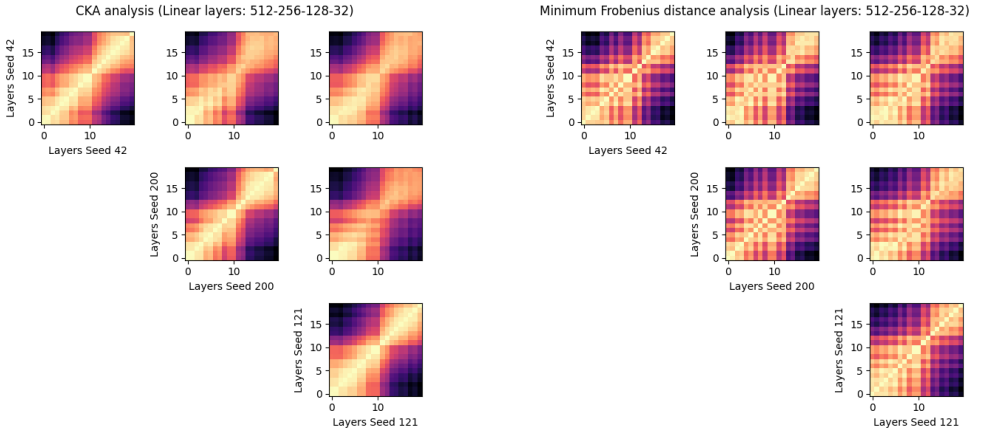


Figure 4.6: Procrustes analysis for the autoencoder with linear layers: 512-256-128-32

Since we now possess a high-dimensional latent space, additional dimensionality reduction techniques are necessary for visualization purposes. We have opted to employ Principal Component Analysis (PCA) because other methods, such as t-SNE, produce representations that it is not clear how they will be affected by the Procrustes transformation (see Figure A.9). However, even with PCA, Figure 4.6 reveals that the resulting visual representation is suboptimal for this task, as numerous clusters exhibit significant overlap in their projections.

4.1.2 Classifier

As previously described, we will repeat the previous experiment using now a simple CNN on a classification task.



(a) CKA analysis

(b) Min. Frobenius norm

Figure 4.7: Numerical latent space analysis for the CNN classifier

As depicted in Figure 4.7, we observe significant similarities in terms of both CKA and the Frobenius norm. Consequently, we have obtained empirical evidence supporting the existence of ε-similarities in the context of a classification task. It is important to note that the original paper just provided numerical analysis on word embeddings and visual examples on Autoencoders and ViTs..

Furthermore, we conducted a Procrustes analysis on the latent representations obtained from the vectorized convolutional feature map, positioned just before the classification head (see Figure 4.8). We chose this since this is

the position where we will apply the relative transformations and topological regularization in the next subsections.

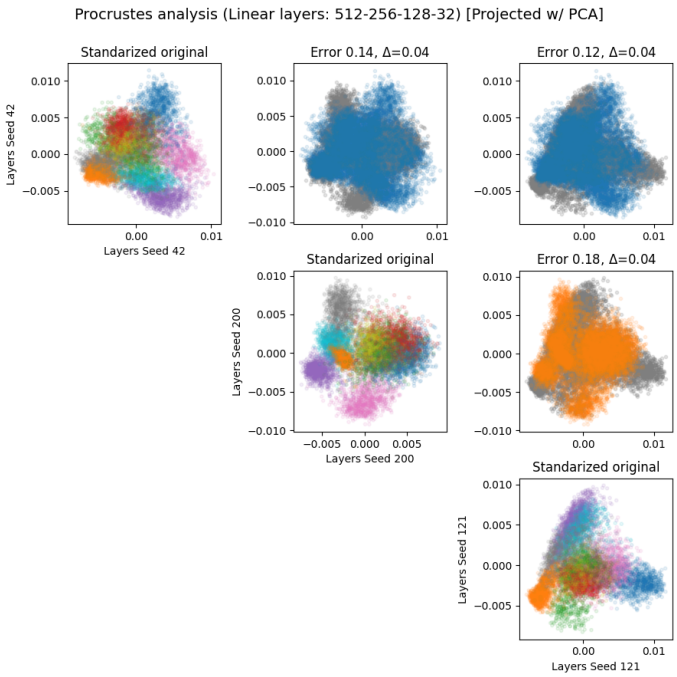


Figure 4.8: Procrustes analysis for the CNN classifier

In conclusion, these empirical findings, combined with the new theoretical framework presented in Section 3.2.1, contribute to a deeper understanding of the potential causes of ε-similarities — to which our relative transformations aim to be invariant. However, it is important to recognize that this is only a small piece of the larger puzzle that the field of Representation Similarity aims to assemble.

4.2 Multilingual model stitching

4.2.1 Full fine-tuning analysis

As discussed in Section 3.2.2, we have made some methodological changes from the original implementation of the following experiments. As a reference, the results presented in the original relative transformation paper are shown in Table 4.1.

		Absolute		Relative	
Decoder	Encoder	FScore × 100	MAE	FScore × 100	MAE
en	en	65.46 ± 2.89	0.38 ± 0.02	61.18 ± 1.92	0.44 ± 0.02
	es	22.70 ± 0.41	1.39 ± 0.03	51.67 ± 1.20	0.62 ± 0.01
	fr	30.75 ± 0.67	1.19 ± 0.02	49.18 ± 0.83	0.69 ± 0.02
es	en	21.24 ± 0.81	1.43 ± 0.07	51.02 ± 2.54	0.68 ± 0.05
	es	61.29 ± 3.04	0.43 ± 0.02	57.89 ± 3.80	0.48 ± 0.03
	fr	29.02 ± 0.85	1.26 ± 0.05	48.40 ± 1.02	0.71 ± 0.02
fr	en	27.39 ± 1.22	1.23 ± 0.06	45.55 ± 3.55	0.76 ± 0.09
	es	29.47 ± 3.68	1.18 ± 0.07	40.29 ± 1.72	0.90 ± 0.04
	fr	56.40 ± 1.89	0.51 ± 0.01	53.58 ± 0.70	0.57 ± 0.01

Table 4.1: Summary of the original results presented in [4] (over five random seeds)

As shown in Table 4.2, we observe slightly worst results when replicating the original training setup while incorporating our proposed methodological changes. However, as previously discussed, we will proceed with this new setup as it represents a more common methodology. Additionally, some of the introduced changes, such as employing gradient accumulation with smaller batch sizes, were necessary to train the models on our V100 GPU successfully. It would, however, be interesting to conduct further analysis to understand why the inclusion of post L^2 normalization on the relative transformation and the utilization of the "non-standard" classification head leads to improved performance.

		Absolute			Relative		
Decoder	Encoder	Acc × 100	FScore × 100	MAE × 100	Acc × 100	FScore × 100	MAE × 100
en	en	55.67 ± 0.58	54.96 ± 1.86	53.80 ± 2.55	55.06 ± 0.82	54.33 ± 1.13	55.69 ± 2.33
	es	18.80 ± 1.61	8.14 ± 2.06	191.49 ± 12.12	44.59 ± 0.16	42.44 ± 0.59	79.93 ± 0.27
	fr	21.30 ± 0.82	17.04 ± 3.28	143.92 ± 3.71	41.61 ± 1.09	38.86 ± 0.73	92.23 ± 3.69
es	en	18.64 ± 1.78	6.57 ± 0.11	207.14 ± 9.42	41.63 ± 1.43	39.98 ± 1.35	101.07 ± 7.57
	es	54.18 ± 0.51	53.80 ± 0.50	54.67 ± 1.23	51.46 ± 0.31	49.79 ± 0.27	61.15 ± 0.35
	fr	21.01 ± 0.41	12.19 ± 0.45	193.09 ± 2.28	39.88 ± 0.88	36.33 ± 2.00	99.27 ± 2.98
fr	en	20.16 ± 0.23	9.04 ± 3.36	131.96 ± 11.37	43.29 ± 1.94	42.66 ± 1.56	80.38 ± 2.09
	es	20.15 ± 0.27	9.77 ± 0.17	139.57 ± 5.44	40.95 ± 0.13	38.51 ± 0.08	92.40 ± 0.99
	fr	49.12 ± 1.24	48.86 ± 1.00	63.76 ± 0.28	46.39 ± 1.09	45.26 ± 1.98	74.25 ± 4.17

Table 4.2: Fine-grained: fine-tune (over two random seeds)

The results from Table 4.3 indicate that fully fine-tuning the model, without freezing the encoder, reduces the gap between our new methodology and the results reported in the original paper for the absolute case in the non-stitching modality.

However, we observed a decline in stitching performance in the absolute case. This decline can be attributed to the absence of additional constraints during the fine-tuning process. Without these constraints, the encoders tend to optimize the latent representations specifically for the given dataset, deviating from the pre-trained configuration known to exhibit high similarity among BERT models [57].

		Absolute			Relative		
Decoder	Encoder	Acc × 100	FScore × 100	MAE × 100	Acc × 100	FScore × 100	MAE × 100
en	en	65.33 ± 0.49	65.22 ± 0.10	39.49 ± 0.33	64.64 ± 0.42	64.67 ± 0.04	39.51 ± 0.52
	es	11.64 ± 10.30	8.65 ± 6.27	224.77 ± 37.46	59.15 ± 0.27	58.87 ± 0.32	45.68 ± 0.25
	fr	17.99 ± 2.81	11.89 ± 0.22	172.37 ± 12.57	56.44 ± 0.91	55.57 ± 1.92	50.25 ± 1.88
es	en	12.41 ± 2.93	10.94 ± 0.39	233.60 ± 5.69	62.43 ± 1.00	61.50 ± 1.66	43.19 ± 3.01
	es	59.42 ± 0.11	59.31 ± 0.01	44.76 ± 0.37	59.22 ± 0.62	58.65 ± 0.66	45.80 ± 0.71
	fr	18.37 ± 8.08	16.44 ± 10.91	164.08 ± 37.17	56.03 ± 0.41	54.30 ± 0.62	51.30 ± 0.40
fr	en	23.82 ± 0.76	21.22 ± 3.49	142.57 ± 24.03	64.77 ± 0.52	64.59 ± 0.18	39.56 ± 0.34
	es	13.85 ± 10.48	11.15 ± 6.95	188.00 ± 46.22	59.06 ± 0.17	58.90 ± 0.46	45.74 ± 0.40
	fr	56.33 ± 0.13	56.02 ± 0.56	50.22 ± 0.34	56.21 ± 0.44	55.72 ± 1.43	50.67 ± 1.77

Table 4.3: Fine-grained: full (over two random seeds)

On the other hand, we observed an improvement in performance in the relative case compared to both the original paper and our new baseline. In the non-stitching scenario, this improvement arises because the encoder is able to discover a better representation that, after the relative transformation, proves more beneficial for the classification task.

Interestingly, we found that fully fine-tuning the network leads to better stitching performance in the relative case. We speculate that the inclusion of relative transformations constructed using parallel anchors serves as a form of implicit regularization, enhancing the compatibility of the pre-relative latent space. Previous research (discussed in Section 2.2.2) has shown that specific auxiliary tasks, such as predicting transformations in the input (self-supervised) and discriminating common classes [49], can increase the compatibility of latent representations. In this case, however, we do not require an additional auxiliary classification head to apply this type of regularization.

Note. The same conclusions were obtained for the coarse-grained dataset (see Tables A.1, A.2).

4.2.2 Topological regularization

Testing new setup

As discussed in Section 3.2.2, we have once again made some methodological changes for this part of the project. One of the primary reasons for utilizing the freezing and unfreezing of the network with the new dataloader is to address a crucial issue: without this "fix," we are unable to train in the relative case. In fact, when this fix is not applied, we observe a significant performance drop, obtaining approximately 7.29 FScore $\times 100$ in English and 18.49 in French.

		Absolute			Relative		
Decoder	Encoder	Acc $\times$ 100	FScore $\times$ 100	MAE $\times$ 100	Acc $\times$ 100	FScore $\times$ 100	MAE $\times$ 100
en	en	60.36 ± 0.23	60.33 ± 0.10	46.32 ± 0.45	60.78 ± 0.34	60.71 ± 0.45	45.06 ± 0.00
	fr	35.63 ± 7.25	31.18 ± 9.77	98.44 ± 4.36	52.05 ± 1.12	52.03 ± 0.23	57.03 ± 0.33
fr	en	29.69 ± 5.16	27.30 ± 5.21	104.83 ± 6.89	60.45 ± 0.13	60.57 ± 0.08	45.12 ± 0.45
	fr	50.24 ± 1.98	50.63 ± 1.66	60.42 ± 2.12	52.21 ± 0.83	52.59 ± 0.33	56.53 ± 0.07

Table 4.4: Linear vanilla dataloader (over two random seeds)

One might argue that a simpler solution for using biased dataloaders in the relative case would be to employ the standard relative transformation instead of the new robust one. This approach would eliminate the need for debiasing the BatchNorm mean and variance estimates. However, we find that training in the relative case with a vanilla dataloader using the standard relative transformation leads to worse performance, resulting in approximately 38.52 FScore $\times 100$ in English and 30.87 in French. For reference, Table 4.4 presents the performance of training with the same setup described in the methodology but using a standard dataloader. It is important to note that this dataloader cannot be used when applying the topological densification technique.

Therefore, we now have both theoretical and empirical justifications for adopting this new relative transformation. Additionally, we notice that the addition of the BatchNorm prior to the relative transformation provides additional robustness to training by mitigating potential issues like vanishing gradients.

Furthermore, Table 4.5 demonstrates that our new baseline, against which we will compare the topological regularization, does not exhibit a significant change in performance compared to fully fine-tuning the network using a

Decoder	Encoder	Absolute			Relative		
		Acc × 100	FScore × 100	MAE × 100	Acc × 100	FScore × 100	MAE × 100
en	en	59.08 ± 0.20	59.08 ± 0.85	48.47 ± 0.64	61.30 ± 0.28	60.84 ± 0.77	44.87 ± 0.92
	fr	35.06 ± 4.36	31.39 ± 4.62	101.75 ± 4.26	48.48 ± 0.08	48.74 ± 0.20	59.26 ± 0.37
fr	en	27.04 ± 6.14	25.86 ± 5.75	115.04 ± 9.79	60.87 ± 1.15	60.25 ± 1.63	45.08 ± 1.87
	fr	48.74 ± 0.62	48.99 ± 0.06	62.53 ± 0.92	49.37 ± 0.30	50.07 ± 0.19	58.24 ± 0.79

Table 4.5: Linear biased dataloader (over two random seeds)

standard dataloader (see Table 4.4).

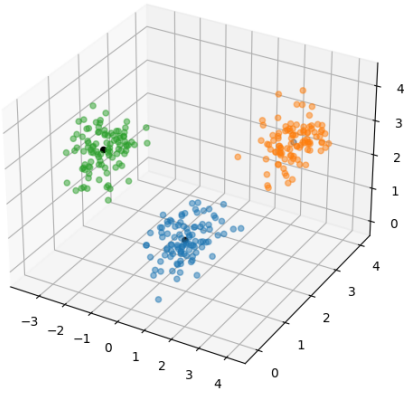
Lastly, it is intriguing that, in this case, the relative transformation tends to yield higher performance compared to the absolute case. One hypothesis we have is that the post-relative space may be more linearly separable. However, we currently lack supporting evidence for this claim, and we believe further research could be conducted to analyze the geometric properties of this new latent space.

Results

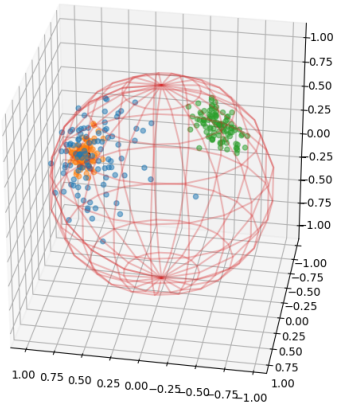
In Section 3.2.2, we explored four different approaches for applying the topological densification. Initially, we attempted the pre-relative case, but it did not achieve performance on par with the baseline presented in Table 4.5. One possible reason is that there are instances where the relative transformation fails to preserve the topology. An example of this can be observed in Figure 4.9, where we have three clusters, two of which possess colinear centroids, resulting in only two clusters after the transformation.

Therefore, we concluded that applying the topological densification to the post-relative space is crucial. However, despite an extensive hyperparameter search, we could still not improve the performance when we only applied the regularization to the post-relative space.

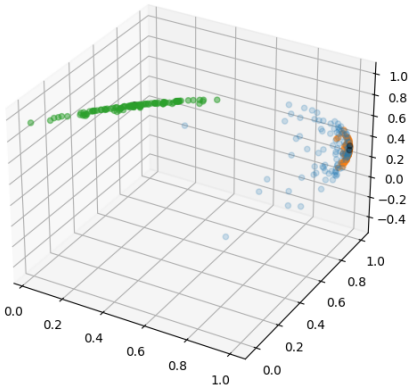
Upon closer examination, we observed that after training with this regularization, the death times distributions of the post-relative space exhibit a significantly smaller mean than the pre-relative space (see Figure 4.10). This implies that when we exclusively apply this loss to the post-relative case, the network discovers an anchor configuration that yields smaller clusters in the post-relative space while preserving the spread of clusters in the pre-relative space. We believe this configuration can lead to information bottlenecks, because compressing the clusters with this nonlinear transformation might



(a) Three clusters w/ an anchor in each centroid



(b) L^2 normalization



(c) Cosine similarity

Figure 4.9: Non-cluster-preserving relative transformation example

cause a loss of expressiveness in the latent space.

After further analysis, we discovered that when no regularization is applied, the death times distributions of the network tend to overlap* (see Figure 4.11). This insight led us to favor overlapped death times distributions to preserve H_0 homology through the relative transformation. Additionally, this approach addresses the issues we encountered when applying the regularization just before or after the relative transformation.

*Formally, two real probability density functions f_A and f_B overlap if their overlapping index $\eta(A, B) = \int_{\mathbb{R}^n} \min\{f_A(x), f_B(x)\}\, dx$ is close to one [58].

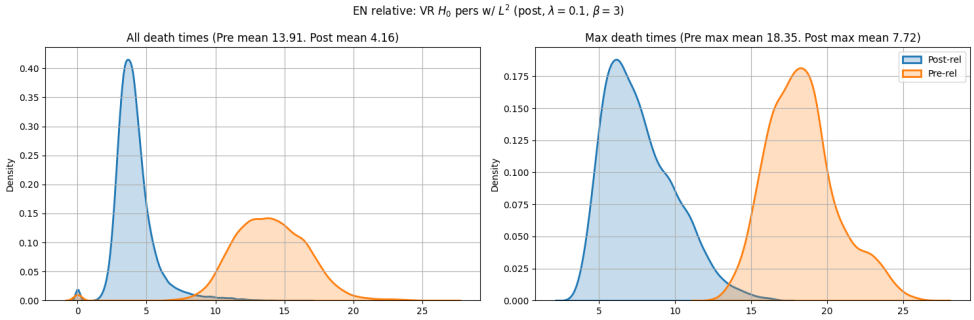


Figure 4.10: Death times distribution when we apply post-relative topological densification on the English dataset with $\beta = 3$.

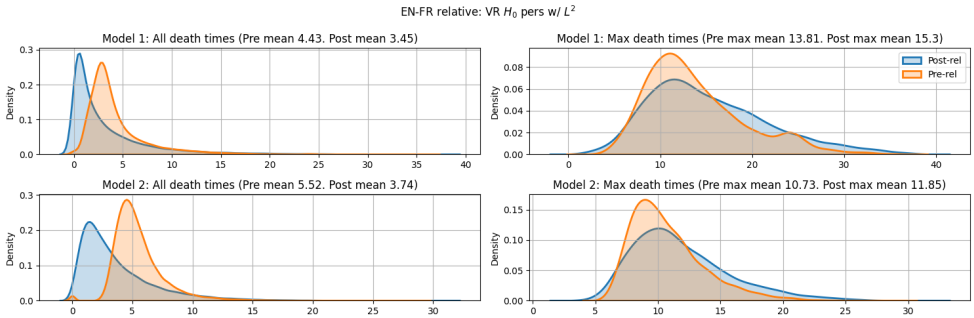


Figure 4.11: Death times distribution without the topological densification on the English dataset.

Following an exhaustive hyperparameter tuning process, we identified that the optimal setup for the absolute case is $\beta = 7$ and $\lambda = 0.1$ for both the English and French datasets. In the relative case, the best results were obtained by using a linear combination of pre-relative (0.1 weight) and post-relative (0.9 weight) regularization, with $\beta = 3$ and $\lambda = 0.02$ for the English case and $\beta = 4$ and $\lambda = 0.02$ for the French case*.

Figure 4.12 demonstrates that we successfully encouraged overlapping death times distributions and reduced the maximum death times means. However, it is worth noting that moving the post-relative max distribution proved more challenging. This difficulty arises from the fact that the geometry of the post-relative transformation is influenced by the overall pre-relative latent space and, more significantly, by the images of the anchors. As

*The complete logs of the hyperparameter tuning results and their corresponding death times distributions can be found in the Thesis' GitHub repository.

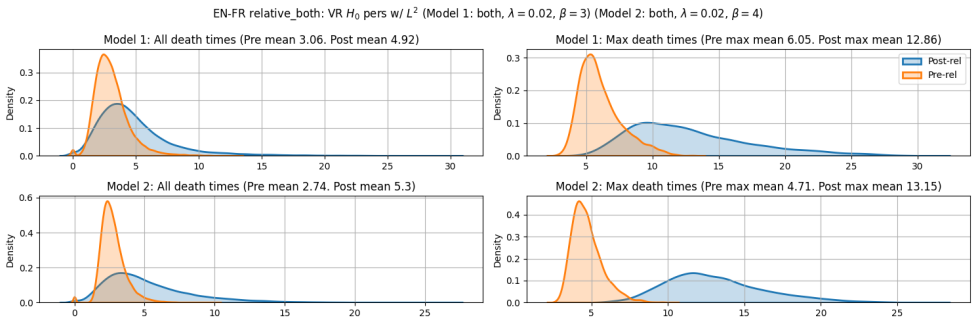


Figure 4.12: Death times distribution when we apply both pre and post-relative topological densification with $\beta = 3$ on the English dataset and $\beta = 4$ in the French dataset.

discussed in Section 3.2.2, we compute the anchors' images periodically and do not allow gradients to backpropagate through the anchors*. Therefore, effecting changes to the post-relative representation is more complex, as it involves adjusting the position of the anchors' images by analyzing the relative transformation of the batch.

As shown in Table 4.6, employing this topological regularization setup yields better performance in the non-stitching case compared to the new baseline in both the absolute and relative modalities. Furthermore, it outperforms the vanilla dataloader for the non-stitching English model in the relative modality and the French model in the absolute modality. Thus, it is reasonable to assume that with a GPU having much VRAM to accommodate the entire biased batch (without employing gradient accumulation and the debiasing trick), consistently superior results could be achieved compared to the vanilla baseline.

		Absolute			Relative		
Decoder	Encoder	Acc × 100	FScore × 100	MAE × 100	Acc × 100	FScore × 100	MAE × 100
en	en	60.20 ± 0.88	59.69 ± 0.37	46.33 ± 0.47	61.25 ± 0.24	61.37 ± 0.07	44.50 ± 0.17
	fr	30.04 ± 0.93	18.56 ± 1.73	121.52 ± 16.07	50.14 ± 0.76	50.55 ± 0.50	58.81 ± 0.16
fr	en	41.01 ± 5.53	29.78 ± 11.70	87.95 ± 7.62	60.49 ± 0.78	60.90 ± 0.54	44.96 ± 0.34
	fr	51.06 ± 0.00	51.81 ± 0.04	56.63 ± 0.01	51.27 ± 0.01	51.71 ± 0.19	57.94 ± 0.74

Table 4.6: Topological regularization (over two random seeds)

*We do not allow gradients to backprop through the anchors since it would require to virtual increase the sub-batch size by 768.

The above results demonstrate that with the appropriate setup, we can replicate the results reported in the original Hofer et. al.'s paper [3], even when using the relative transformation. However, the results of model stitching with topological regularization have not been previously reported. As shown in Table 4.6, we achieve improved stitching performance in all stitching combinations and modalities, except for the "en-fr abs" case, compared to the new baseline. Moreover, despite observing a performance drop when using the new dataloader with the fix, we surpass the results obtained with the vanilla dataset on stitching for the "fr-en" cases in both the relative and absolute modalities.

		Relative		
Decoder	Encoder	Acc × 100	FScore × 100	MAE × 100
en	en	61.25 ± 0.24	61.37 ± 0.07	44.50 ± 0.17
	fr	$\mathbf{50.90 \pm 0.65}$	$\mathbf{51.50 \pm 0.66}$	$\mathbf{57.27 \pm 0.07}$
fr	en	$\mathbf{60.87 \pm 0.95}$	$\mathbf{61.27 \pm 0.77}$	$\mathbf{44.56 \pm 0.71}$
	fr	50.11 ± 0.38	50.58 ± 0.79	57.78 ± 0.14

Table 4.7: Topological reg. w/ matched params (over two random seeds)

Additionally, Table 4.7 reveals that we can enhance the stitching performance by selecting the same β parameter for topological regularization (i.e., using $\beta = 4$ for the French case). However, this impacts the non-stitching performance, as the optimal hyperparameter choice of $\beta = 3$ is no longer used. This observation is logical since different hyperparameter choices will yield the best results for different datasets during network training. Nevertheless, for stitching purposes, it is preferable to impose consistent topological densification, increasing the possibility that class clusters reside within the decision boundary of the other classifier.

Note. The topological densification generally does not outperform early stopping in terms of non-stitching performance. However, in the "fr-en" stitching case, we achieve better results than early stopping (see Tables A.3, A.4).

The reason why we could not successfully apply the topological densification to the Spanish case as well as why the "en-fr" stitching case does not perform as well as "fr-en", will be discussed in Section 5.3.

As Section 3.2.2 outlines, utilizing L^∞ in the post-relative space offers certain advantages. In this section, we will demonstrate how it can benefit

hyperparameter selection. In previous experiments, the approach involved computing the death times distribution and conducting a grid search based on values such as: $0.3 \times$ post_mean, $0.5 \times$ post_mean, $0.6 \times$ post_mean, $0.3 \times$ post_max_mean, $0.5 \times$ post_max_mean, and $0.6 \times$ post_max_mean. However, a new approach can be adopted:

1. Compute the death times distributions using the L^2 in the pre-relative space and L^∞ in the post-relative space. For example, in Figure 4.13, we can see the resulting distributions for the English relative case.

2. Determine the approximate spread of the cluster using

$$\theta_{og} = 180 \arccos(1 - \text{post\_mean}).$$

   In the previous example, θ_{og} is approximately $36°$.

3. Conduct a grid search for $\theta \in [\tau_1\theta_{og}, \tau_2\theta_{og}]$, where $\tau_1 \geq 0$ and $\tau_2 > 0$, with a specified step size $\delta\theta$. Set $\beta_{post} = 1 - \cos(\pi\theta/180)$.

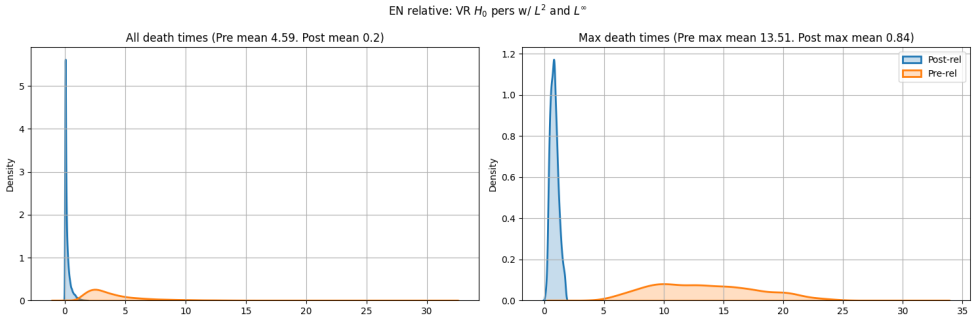


Figure 4.13: Death times distribution on the English dataset. L^2 is used in the pre-relative space and L^∞ in the post-relative space.

Therefore, we performed a grid search on $\theta \in [5°, 30°]$ with a step size of $\delta\theta = 5°$, while keeping β_{pre} fixed at 3. Surprisingly, for $\theta = 25°$, we achieved better performance compared to using L^2 with $\beta_{post} = 3$ (61.48 FScore $\times 100$). Hence, this result serves as a proof of concept for the potential exploration of different metrics when employing the topological densification technique*.

*Due to time limitations, we were not able to fully explore the L^∞ case to be able to produce a table with proper hyperparameter tuning as we did for the L^2 cases.

Chapter 5

Discussion

In this chapter, we will engage in further analysis, offering additional insights, opinions, and hypotheses concerning the methodology and the results obtained in this project.

5.1 Similarities beyond intertwiner group actions

As previously mentioned, we acknowledge that the observed ε-similarities are not exclusively generated by the actions of the intertwiner group. This claim is supported by two key points:

- In Section 4.1, we computed the Frobenius distance between the optimal orthogonal and optimal permutation Procrustes transformations. It is important to recall that we did not observe small distances between these matrices, which prevents us from conclusively attributing the source of the ε-similarity to the intertwiner group associated with the GeLU activation function used in RoBERTa models.
- In the original paper introducing the intertwiner group [50], a variant of the CKA metric called CKA-ReLU was developed. This variant is designed to yield high similarities between latent spaces if the differences arise from a ReLU intertwiner group's action. Using this new metric, the authors found that in later layers, the CKA-ReLU decreases while the standard CKA remains unchanged. This additional empirical evidence suggests that other factors may contribute to the observed ε-similarity.

However, despite these observations, we believe that establishing a theoretical foundation for the relative transformation is still crucial. We now understand that if the ε-similarity arises from the intertwiner group, our transformation will be invariant to it. Furthermore, even if the ε-similarity originates from another source, as mentioned in [4], our transformation will still maintain invariance, although lacking a rigorous theoretical proof.

5.2 Post-relative geometry

In the previous chapter, we observed that applying the topological densification in the post-relative space posed more challenges compared to the pre-relative space. One of the reasons identified was the significant influence of the anchors on the post-relative geometry. To illustrate the importance of anchors in supporting this claim, we provide the following proposition:

Proposition 5.2.1. *Let $\mathbb{A} \in \mathbb{R}^{d\times k}$ be the matrix representation of the (normalized) images of the anchor set $\mathcal{A}$, and let $z_1, z_2 \in \mathcal{Z}$ be the (normalized) images of samples $x_1, x_2 \in \mathcal{X}$. Then,*

$$\left\| \mathbb{A}^T \frac{z_1}{\|z_1\|} - \mathbb{A}^T \frac{z_2}{\|z_2\|} \right\|^2 \leq 2(\max \lambda - \min \sigma)\,,$$

where λ and σ are, respectively, the eigenvalues and singular values of $\mathbb{A}\mathbb{A}^T$.

Proof.

$$\begin{aligned}
\left\| \mathbb{A}^T \frac{z_1}{\|z_1\|} - \mathbb{A}^T \frac{z_2}{\|z_2\|} \right\|^2 &= \left(\frac{z_1^T}{\|z_1\|}\mathbb{A} - \frac{z_2^T}{\|z_2\|}\mathbb{A} \right) \left(\mathbb{A}^T \frac{z_1}{\|z_1\|} - \mathbb{A}^T \frac{z_2}{\|z_2\|} \right) \\
&= \frac{z_1^T \mathbb{A}\mathbb{A}^T z_1}{\|z_1\|^2} - 2\frac{z_1^T \mathbb{A}\mathbb{A}^T z_2}{\|z_1\|\,\|z_2\|} + \frac{z_2^T \mathbb{A}\mathbb{A}^T z_2}{\|z_2\|^2} \\
&\leq 2(\max \lambda - \min \sigma)\,.
\end{aligned}$$

The final step follows from the properties of Rayleigh's quotient. □

Hence, we can see that an upper bound on the diameter of the relative transformation is solely determined by the anchors.

5.3 Topological regularization overview

5.3.1 High computational complexity

As we have seen, the topological densification technique requires a specific dataloader which does not work as originally presented when we use gradient accumulation and the robust relative transformation. To address this issue, we employed the freezing/unfreezing technique, extensively discussed in Section 3.2.2. However, it is worth noting that this fix led to slightly inferior results compared to the vanilla case.

The scalability concern we encountered is not exclusive to the topological densification technique; several topological data analysis (TDA) methods face similar challenges. For instance, the standard algorithm for computing Persistent Homology has a complexity that scales cubically with the number of simplices [59]. Consequently, current practice often restricts the analysis to H_0 and H_1 since computing higher homologies on large datasets can take weeks. Nevertheless, due to the growing interest in TDA, recent research has aimed to enhance the scalability of various TDA methods [59, 60].

Therefore, we believe that if there starts to have an increase in the visibility of topological regularization techniques, it would be required to optimize existing methods or develop new ones that are better suited for modern large-scale neural network architectures.

5.3.2 Strong regularization

During the hyperparameter tuning phase, we made an interesting observation regarding the topological densification technique. It had a significant impact in mitigating overfitting, but the regularization effect was so pronounced that it led to inferior classification performance compared to not using the regularization. To address this issue, we employed the cyclical linear scheduler for the loss's weight component λ. This approach resulted in densified distributions that still yielded satisfactory performance for the classification task.

We believe that the reason why we could not successfully apply the topological densification to the Spanish case as well as why the "en-fr" stitching case does not perform as well as "fr-en", is that both the French

and Spanish models had less expressiveness than the English model. Hence applying a "demanding" regularization prevents the network from obtaining a proper latent space for the classification class. In other words, we encounter an imbalance in the bias-variance tradeoff, with excessive bias hampering the network's performance.

5.3.3 Beyond the Vietoris–Rips complex

As observed in our experiments, the use of different metrics for computing the 0-dimensional Vietoris–Rips persistent homology can have some benefits compared to using the standard L^2 metric.

In addition, we believe that employing diverse simplicial complexes for the computation of the topological densification loss can yield interesting insights. For example, utilizing Lazy Witness complexes appears to be well-suited for the post-relative space (see Definition 2.1.20). This choice is motivated by the fact that Witness complexes capture the data's topology from the perspective of the witnesses, while the relative transformations look at the geometry of the data through the anchors.

Furthermore, based on Proposition 2.1.3, we can know the relations between the H_0 homology of a Witness complex, and the ε-connectivity:

Corollary. *Let $X \subset \mathbb{R}^n$, and $\dagger(X)$ the H_0 persistent homology death times of a Witness complex filtration of X. Then,*

- *If* $\max \dagger(X) < \varepsilon \implies 2\varepsilon$*-connected.*
- *If* 2ε*-connected* $\not\implies \max \dagger(X) < \varepsilon$.
- *If* 2ε*-connected* $\implies \max \dagger(X) < 2\varepsilon$.

Chapter 6

Conclusions and Future work

6.1 Conclusions

In summary, this thesis provides novel insights regarding topological regulation techniques combined with model stitching methods, such as the relative transformation.

Firstly, our study led to the development of new theoretical foundations concerning the ε-similarities based on the intertwiner groups of activation functions. This advancement allowed for more effective construction of the relative transformation. Additionally, through empirical analysis, we gained a deeper understanding of the similarities within the latent space. As a result, we now have stronger theoretical and empirical support for utilizing the relative transformation in zero-shot model stitching.

Secondly, in the full fine-tuning analysis, we observed slightly worse results when replicating the original training setup while incorporating our proposed methodological changes. However, fully fine-tuning the model without freezing the encoder narrowed the performance gap between the new methodology and the results reported in the original paper for the absolute case. Furthermore, we observed that fully fine-tunning can also be beneficial for the relative case.

Thirdly, our exploration of topological regularization revealed that applying the regularization to both the pre-relative and post-relative spaces is crucial for achieving improved performance. We also found that encouraging overlapping of death times distributions resulted in better performance.

However, adjusting the post-relative representation proved to be more challenging due to the influence of the images of the anchors.

Lastly, the results of model stitching with topological regularization showed improved stitching performance in all stitching combinations and modalities compared to the new baseline, except for one case. Additionally, selecting the same hyperparameters for topological regularization enhanced stitching performance but impacted non-stitching performance.

Overall, our findings suggest that the proposed methodological changes and the application of topological regularization can potentially improve the performance of multilingual model stitching. However, further research is needed to explore the new latent space's geometric properties and investigate the use of different metrics for topological regularization.

6.2 Limitations

As mentioned previously, the training of the network using the topological densification technique, as originally proposed in [3], was hindered by the limited VRAM of the V100 GPU we utilized. Consequently, we resorted to employing the "debiasing trick," as described earlier, to accommodate this regularization technique. Unfortunately, this computational limitation imposed constraints on our ability to achieve optimal results.

Additionally, the unexpected issue with the topological densification led us to omit certain experiments, such as the analysis of representation similarity in multilingual model stitching.

Moreover, due to time constraints, we were unable to conduct an extensive literature study on the possible sources of the observed ε-similarity, as initially intended. We believe that conducting further research in this area could potentially unveil additional insights into the origins of the observed isometries.

Lastly, our multilingual stitching models employed a single linear layer as the classification head. Although this architecture aligns with the standard approach in certain state-of-the-art transformer models, we originally wanted to investigate the utility of this regularization technique when using a more expressive classification head. Unfortunately, due to time and computational limitations, we were unable to explore this avenue.

6.3 Future work

There are several potential directions for expanding the work presented in this thesis. Here, we highlight some of the most relevant ones:

- **Investigation of alternative simplicial complex constructions:** As we have seen in Section 5.3.3, it would be interesting to asses if our results depend on the specific construction to produce simplicial complexes in the TDA-based regularization. One good candidate for a new simplicial complex construction that could be used in the post-relative space is the Lazy Witness complex.

- **Further exploration of metric choices for Vietoris–Rips filtrations:** In Section 4.2.2, we demonstrated the potential benefits of using different metrics to compute the Vietoris–Rips filtrations. However, our analysis provided only a proof of concept. A more comprehensive investigation and analysis of different metrics would be advisable.

- **Analysis of representation similarity in multilingual model stitching:** Due to time constraints, we were unable to analyze the CKA in the multilingual model stitching experiments. Performing this representation similarity analysis would be valuable. To conduct such an analysis, a new dataset would need to be created, similar to the parallel anchor construction, where translations to other languages are available.

- **Testing topological regularization on large models with increased GPU VRAM:** It is expected that better performance could be achieved by using larger VRAM in the GPU, eliminating the need for the "debiasing fix" utilized in this work. Therefore, it would be interesting to evaluate the topological densification technique on large models while employing a more powerful GPU.

- **Exploring the combination of topological regularization with the relative transformation in a multimodal setup:** Recent studies have shown that the relative transformation can transform unimodal models into multimodal ones without additional training [61]. Investigating how to adapt the proposed methodology to combine the topological densification with the relative transformation in this new multimodal setup would be intriguing.

- **Exploring higher dimensional homology:** The topological densification only makes use of the 0-dimensional homology in order to impose β-connectivity. However, one advantage that TDA has over classical methods, such as Hierarchical clustering (precursor of H_0), is the ability to analyze higher dimensional holes (i.e., using H_n) [16]. Hence, it would be interesting to explore if there exist some properties regarding higher dimensional features that can be beneficial for model stitching.

6.4 Reflections

As discussed in Section 2.2.2, an application of the cross-domain model stitching procedure is a scenario where multiple devices exist, each operating in a specific language. In such cases, one can achieve satisfactory performance by utilizing a single trained classification head (e.g., trained on the English corpus) and stitching it with language-specific encoders. This computational approach proves advantageous in situations requiring frequent model updates, as it necessitates updating only one model instead of all.

The observation that specific regularization techniques, such as the topological densification, can enhance model stitching performance further supports their utilization. By employing these improved model stitching setups, we can mitigate computational costs associated with training and fine-tuning networks. It is important to note that this approach would yield not only economic benefits but also have positive environmental implications by reducing the immense carbon footprint associated with training large language models [62].

References

[1] C. Fefferman, S. Mitter, and H. Narayanan, "Testing the manifold hypothesis," *Journal of the American Mathematical Society*, vol. 29, no. 4, pp. 983–1049, Feb. 2016. doi: 10.1090/jams/852. [Online]. Available: https://www.ams.org/jams/2016-29-04/S0894-0347-2016-00852-4/ [Pages 1 and 32.]

[2] F. Hensel, M. Moor, and B. Rieck, "A Survey of Topological Machine Learning Methods," *Frontiers in Artificial Intelligence*, vol. 4, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frai.2021.681108 [Pages 1, 29, and 31.]

[3] C. D. Hofer, F. Graf, M. Niethammer, and R. Kwitt, "Topologically Densified Distributions," May 2021, arXiv:2002.04805 [cs, math, stat]. [Online]. Available: http://arxiv.org/abs/2002.04805 [Pages 1, 3, 7, 33, 34, 35, 45, 47, 54, 80, and 88.]

[4] L. Moschella, V. Maiorca, M. Fumero, A. Norelli, F. Locatello, and E. Rodolà, "Relative representations enable zero-shot latent space communication," Sep. 2022, arXiv:2209.15430 [cs]. [Online]. Available: http://arxiv.org/abs/2209.15430 [Pages x, xiii, 2, 3, 4, 5, 7, 37, 42, 43, 45, 47, 48, 54, 65, 73, and 84.]

[5] H. Edelsbrunner and J. Harer, *Computational topology: an introduction*. Providence, R.I: American Mathematical Society, 2010. ISBN 978-0-8218-4925-5 OCLC: ocn427757156. [Pages ix, 7, 10, 11, 12, 13, 14, 22, 23, and 28.]

[6] B. Doherty, "The Cech complex in Topological Data Analysis," *University of Western Ontario*, 2018. [Online]. Available: https://jdc.math.uwo.ca/TDA/Doherty-Cech-complex.pdf [Page 12.]

[7] ProboscideaRubber15, "Čech complex," Apr. 2023, page Version ID: 1149764739. [Online]. Available: https://commons.wikimedia.org/w/index.php?curid=69312114 [Pages ix and 13.]

[8] A. Choudhary, "Approximation algorithms for Vietoris-Rips and Čech filtrations," doctoralThesis, Saarländische Universitäts- und Landesbibliothek, 2017, accepted: 2017-12-14T11:38:24Z. [Online]. Available: https://publikationen.sulb.uni-saarland.de/handle/20.500.11880/26911 [Pages ix and 14.]

[9] V. Robins, "Computational Topology at Multiple Resolutions: Foundations and Applications to Fractals and Dynamics," 2000. [Pages 14 and 15.]

[10] V. D. Silva and G. Carlsson, "Topological estimation using witness complexes," *SPBG'04 Symposium on Point - Based Graphics 2004*, p. 10 pages, 2004. doi: 10.2312/SPBG/SPBG04/157-166 Artwork Size: 10 pages ISBN: 9783905673098 Publisher: The Eurographics Association. [Online]. Available: http://diglib.eg.org/handle/10.2312/SPBG.SPBG04.157-166 [Pages 15 and 16.]

[11] A. A. Medbouhi, "Towards topology-aware Variational Auto-Encoders : from InvMap-VAE to Witness Simplicial VAE," Ph.D. dissertation, KTH Royal Institute of Technology, 2022. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-309487 [Pages ix, 15, and 16.]

[12] "GUDHI: Witness complex." [Online]. Available: https://gudhi.inria.fr/doc/latest/group__witness__complex.html [Page 15.]

[13] C. H. Dowker, "Homology Groups of Relations," *Annals of Mathematics*, vol. 56, no. 1, pp. 84–95, 1952. doi: 10.2307/1969768 Publisher: Annals of Mathematics. [Online]. Available: https://www.jstor.org/stable/1969768 [Pages 15 and 16.]

[14] S. T. Schönenberger, A. Varava, V. Polianskii, J. J. Chung, D. Kragic, and R. Siegwart, "Witness Autoencoder: Shaping the Latent Space with Witness Complexes," in *Witness Autoencoder: Shaping the Latent Space with Witness Complexes*, Jul. 2022. [Page 16.]

[15] A. Hatcher, *Algebraic topology*. Cambridge ; New York: Cambridge University Press, 2002. ISBN 978-0-521-79160-1 978-0-521-79540-1 [Pages 17 and 23.]

[16] W. Chachólski, "SF2956 Topological Data Analysis lecture notes," 2022. [Online]. Available: https://www.kth.se/student/kurser/kurs/SF2956?l=en [Pages 17 and 90.]

[17] H. Edelsbrunner and J. Harer, "Persistent homology—a survey," in *Contemporary Mathematics*, J. E. Goodman, J. Pach, and R. Pollack, Eds. Providence, Rhode Island: American Mathematical Society, 2008, vol. 453, pp. 257–282. ISBN 978-0-8218-4239-3 978-0-8218-8132-3. [Online]. Available: http://www.ams.org/conm/453/ [Pages ix, 23, and 25.]

[18] J. Curry, "Counting Embedded Spheres with the same Persistence," Jun. 2020. [Online]. Available: http://www.fields.utoronto.ca/talks/Counting-Embedded-Spheres-same-Persistence [Pages ix and 24.]

[19] C. Hofer, R. Kwitt, M. Dixit, and M. Niethammer, "Connectivity-Optimized Representation Learning via Persistent Homology," Jun. 2019, arXiv:1906.09003 [cs, math, stat]. [Online]. Available: http://arxiv.org/abs/1906.09003 [Pages 25 and 36.]

[20] J. Curry, "The Fiber of the Persistence Map for Functions on the Interval," Jan. 2019, arXiv:1706.06059 [math]. [Online]. Available: http://arxiv.org/abs/1706.06059 [Pages ix and 27.]

[21] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, "Stability of Persistence Diagrams," *Discrete and Computational Geometry*, vol. 37, no. 1, pp. 103–120, Jan. 2007. doi: 10.1007/s00454-006-1276-5. [Online]. Available: http://link.springer.com/10.1007/s00454-006-1276-5 [Page 28.]

[22] L. Cheng, "The Application of Topological Data Analysis in Practice and Its Effectiveness," *E3S Web of Conferences*, vol. 214, p. 03034, 2020. doi: 10.1051/e3sconf/202021403034 Publisher: EDP Sciences. [Online]. Available: https://www.e3s-conferences.org/articles/e3sconf/abs/2020/74/e3sconf_ebldm2020_03034/e3sconf_ebldm2020_03034.html [Page 29.]

[23] C. M. Topaz, L. Ziegelmeier, and T. Halverson, "Topological Data Analysis of Biological Aggregation Models," *PLOS ONE*, vol. 10, no. 5, p. e0126383, May 2015. doi: 10.1371/journal.pone.0126383 Publisher: Public Library of Science. [Online]. Available: https:

//journals.plos.org/plosone/article?id=10.1371/journal.pone.0126383 [Page 29.]

[24] E. J. Amézquita, M. Y. Quigley, T. Ophelders, E. Munch, and D. H. Chitwood, "The shape of things to come: Topological data analysis and biology, from molecules to organisms," *Developmental Dynamics*, vol. 249, no. 7, pp. 816–833, 2020. doi: 10.1002/dvdy.175 Https://onlinelibrary.wiley.com/doi/pdf/10.1002/dvdy.175. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/dvdy.175 [Page 29.]

[25] J. Berwald, "The Mathematics of Quantum-Enabled Applications on the D-Wave Quantum Computer," *Notices of the American Mathematical Society*, vol. 66, p. 1, Jun. 2019. doi: 10.1090/noti1893 [Pages x and 29.]

[26] J. Agerberg, R. Ramanujam, M. Scolamiero, and W. Chachólski, "Supervised Learning Using Homology Stable Rank Kernels," *Frontiers in Applied Mathematics and Statistics*, vol. 7, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fams.2021.668046 [Page 29.]

[27] P. Bubenik, "Statistical topological data analysis using persistence landscapes," Jan. 2015, arXiv:1207.6437 [cs, math, stat]. [Online]. Available: http://arxiv.org/abs/1207.6437 [Page 29.]

[28] G. Ma, "Using Topological Data Analysis to Process Time-series Data: A Persistent Homology Way," *Journal of Physics: Conference Series*, vol. 1550, p. 032082, May 2020. doi: 10.1088/1742-6596/1550/3/032082 [Pages x and 30.]

[29] H. Adams, S. Chepushtanova, T. Emerson, E. Hanson, M. Kirby, F. Motta, R. Neville, C. Peterson, P. Shipman, and L. Ziegelmeier, "Persistence Images: A Stable Vector Representation of Persistent Homology," Jul. 2016, arXiv:1507.06217 [cs, math, stat]. [Online]. Available: http://arxiv.org/abs/1507.06217 [Page 30.]

[30] M. Carrière, S. Y. Oudot, and M. Ovsjanikov, "Stable Topological Signatures for Points on 3D Shapes," *Computer Graphics Forum*, vol. 34, no. 5, pp. 1–12, 2015. doi: 10.1111/cgf.12692 Https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12692. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12692 [Page 30.]

[31] N. Saul, "Persistence Images in Classification — Persim 0.3.1 documentation," 2019. [Online]. Available: https://persim.scikit-tda.org/en/latest/notebooks/Classification%20with%20persistence%20images.html [Pages x and 30.]

[32] S. Zheng, Y. Zhang, H. Wagner, M. Goswami, and C. Chen, "Topological Detection of Trojaned Neural Networks," in *Topological Detection of Trojaned Neural Networks*, Jan. 2022. [Online]. Available: https://openreview.net/forum?id=1r2EannVuIA [Pages x and 31.]

[33] M. Moor, M. Horn, B. Rieck, and K. Borgwardt, "Topological Autoencoders," May 2021, arXiv:1906.00722 [cs, math, stat]. [Online]. Available: http://arxiv.org/abs/1906.00722 [Page 31.]

[34] C. Chen, X. Ni, Q. Bai, and Y. Wang, "A Topological Regularizer for Classifiers via Persistent Homology," Oct. 2018, arXiv:1806.10714 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1806.10714 [Page 32.]

[35] D. Choi and W. Rhee, "Utilizing Class Information for Deep Network Representation Shaping," Feb. 2019, arXiv:1809.09307 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1809.09307 [Page 32.]

[36] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," Dec. 2022, arXiv:1312.6114 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1312.6114 [Page 33.]

[37] M. J. Schervish, *Theory of Statistics*, ser. Springer Series in Statistics. New York, NY: Springer, 1995. ISBN 978-1-4612-8708-7 978-1-4612-4250-5. [Online]. Available: http://link.springer.com/10.1007/978-1-4612-4250-5 [Page 34.]

[38] J. J. Torres, "Hopfield Network," in *Encyclopedia of Computational Neuroscience*, D. Jaeger and R. Jung, Eds. New York, NY: Springer, 2019, pp. 1–3. ISBN 978-1-4614-7320-6 [Pages x and 36.]

[39] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2002. ISBN 978-0-201-18075-6 Google-Books-ID: 738oAQAAMAAJ. [Page 37.]

[40] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR*

2001, vol. 1. Kauai, HI, USA: IEEE Comput. Soc, 2001. doi: 10.1109/CVPR.2001.990517. ISBN 978-0-7695-1272-3 pp. I–511–I–518. [Online]. Available: http://ieeexplore.ieee.org/document/990517/ [Page 37.]

[41] A. Morcos, M. Raghu, and S. Bengio, "Insights on representational similarity in neural networks with canonical correlation," in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/a7a3d70c6d17a73140918996d03c014f-Abstract.html [Pages 38, 39, 40, and 50.]

[42] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of Neural Network Representations Revisited," in *Proceedings of the 36th International Conference on Machine Learning*. PMLR, May 2019, pp. 3519–3529, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v97/kornblith19a.html [Pages 38, 39, 40, and 69.]

[43] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring Statistical Dependence with Hilbert-Schmidt Norms," in *Algorithmic Learning Theory*, ser. Lecture Notes in Computer Science, S. Jain, H. U. Simon, and E. Tomita, Eds. Berlin, Heidelberg: Springer, 2005. ISBN 978-3-540-31696-1 pp. 63–77. [Page 39.]

[44] L. Wang, L. Hu, J. Gu, Z. Hu, Y. Wu, K. He, and J. Hopcroft, "Towards Understanding Learning Representations: To What Extent Do Different Neural Networks Learn the Same Representation," in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/5fc34ed307aac159a30d81181c99847e-Abstract.html [Page 40.]

[45] S. Barannikov, I. Trofimov, N. Balabin, and E. Burnaev, "Representation Topology Divergence: A Method for Comparing Neural Network Representations." in *Proceedings of the 39th International Conference on Machine Learning*. PMLR, Jun. 2022, pp. 1607–1626, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v162/barannikov22a.html [Page 40.]

[46] A. Csiszárik, P. Kőrösi-Szabó, . K. Matszangosz, G. Papp, and D. Varga, "Similarity and Matching of Neural Network Representations," Oct.

2021, arXiv:2110.14633 [cs]. [Online]. Available: http://arxiv.org/abs/2110.14633 [Pages x, 41, and 42.]

[47] Y. Bansal, P. Nakkiran, and B. Barak, "Revisiting Model Stitching to Compare Neural Representations," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 225–236. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/01ded4259d101feb739b06c399e9cd9c-Abstract.html [Page 41.]

[48] Y. Li, J. Yosinski, J. Clune, H. Lipson, and J. Hopcroft, "Convergent Learning: Do different neural networks learn the same representations?" Feb. 2016, arXiv:1511.07543 [cs]. [Online]. Available: http://arxiv.org/abs/1511.07543 [Page 41.]

[49] M. Gygli, J. Uijlings, and V. Ferrari, "Towards Reusable Network Components by Learning Compatible Representations," Dec. 2020, arXiv:2004.03898 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2004.03898 [Pages 42 and 74.]

[50] C. Godfrey, D. Brown, T. Emerson, and H. Kvinge, "On the Symmetries of Deep Learning Models and their Internal Representations," Mar. 2023, arXiv:2205.14258 [cs]. [Online]. Available: http://arxiv.org/abs/2205.14258 [Pages xiii, 50, 51, 52, and 83.]

[51] P. Keung, Y. Lu, G. Szarvas, and N. A. Smith, "The Multilingual Amazon Reviews Corpus," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020. doi: 10.18653/v1/2020.emnlp-main.369 pp. 4563–4568. [Online]. Available: https://aclanthology.org/2020.emnlp-main.369 [Page 56.]

[52] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," Jul. 2019, arXiv:1907.11692 [cs]. [Online]. Available: http://arxiv.org/abs/1907.11692 [Page 56.]

[53] P. Chang, "Advanced Techniques for Fine-tuning Transformers," Nov. 2021. [Online]. Available: https://towardsdatascience.com/advanced-techniques-for-fine-tuning-transformers-82e4e61e16e [Page 57.]

[54] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement

Learning," Dec. 2013, arXiv:1312.5602 [cs] version: 1. [Online]. Available: http://arxiv.org/abs/1312.5602 [Page 57.]

[55] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," Jun. 2021, arXiv:2010.11929 [cs]. [Online]. Available: http://arxiv.org/abs/2010.11929 [Page 61.]

[56] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin, "Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing," Jun. 2019, arXiv:1903.10145 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1903.10145 [Page 63.]

[57] G. Roeder, L. Metz, and D. P. Kingma, "On Linear Identifiability of Learned Representations," Jul. 2020, arXiv:2007.00810 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2007.00810 [Page 74.]

[58] M. Pastore and A. Calcagnì, "Measuring Distribution Similarities Between Samples: A Distribution-Free Overlapping Index," *Frontiers in Psychology*, vol. 10, 2019. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpsyg.2019.01089 [Page 77.]

[59] C. G. Akcora, M. Kantarcioglu, Y. R. Gel, and B. Coskunuzer, "Reduction Algorithms for Persistence Diagrams of Networks: CoralTDA and PrunIT," Nov. 2022, arXiv:2211.13708 [cs, math]. [Online]. Available: http://arxiv.org/abs/2211.13708 [Page 85.]

[60] V. Polianskii, "Breaking the Dimensionality Curse of Voronoi Tessellations," Ph.D. dissertation, KTH Royal Institute of Technology, Stockholm, 2022, publisher: KTH Royal Institute of Technology. [Online]. Available: https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-319984 [Page 85.]

[61] A. Norelli, M. Fumero, V. Maiorca, L. Moschella, E. Rodolà, and F. Locatello, "ASIF: Coupled Data Turns Unimodal Models to Multimodal Without Training," Oct. 2022. [Online]. Available: https://arxiv.org/abs/2210.01738v2 [Page 89.]

[62] E. Strubell, A. Ganesh, and A. McCallum, "Energy and Policy Considerations for Deep Learning in NLP," Jun. 2019, arXiv:1906.02243 [cs]. [Online]. Available: http://arxiv.org/abs/1906.02243 [Page 90.]

Appendix A

Additional results

A.1 Latent space similarity analysis

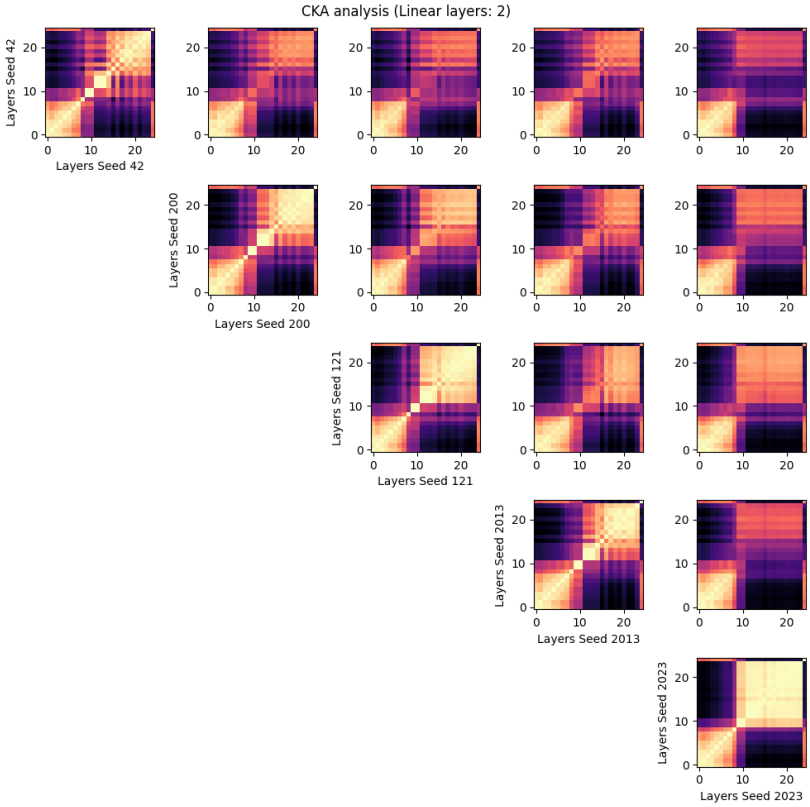


Figure A.1: Additional CKA analysis for the autoencoder with linear layers: 2

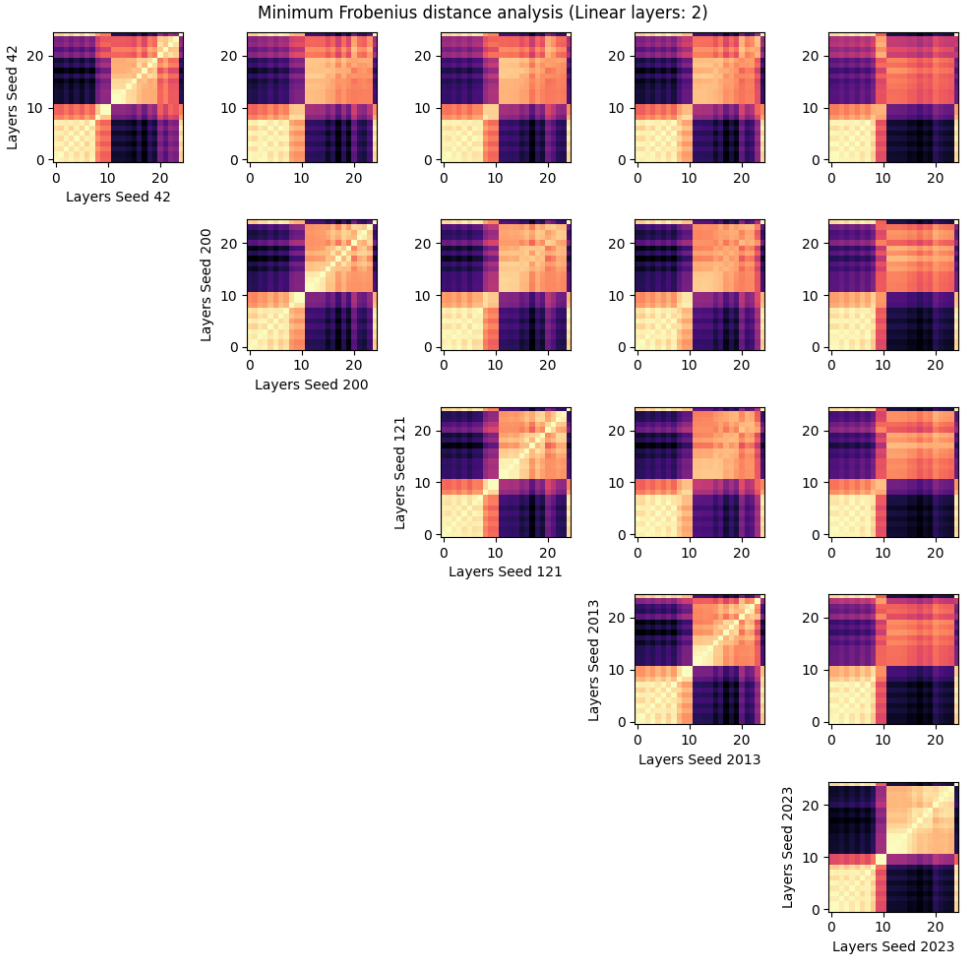


Figure A.2: Additional Min. Frobenius norm analysis for the autoencoder with linear layers: 2

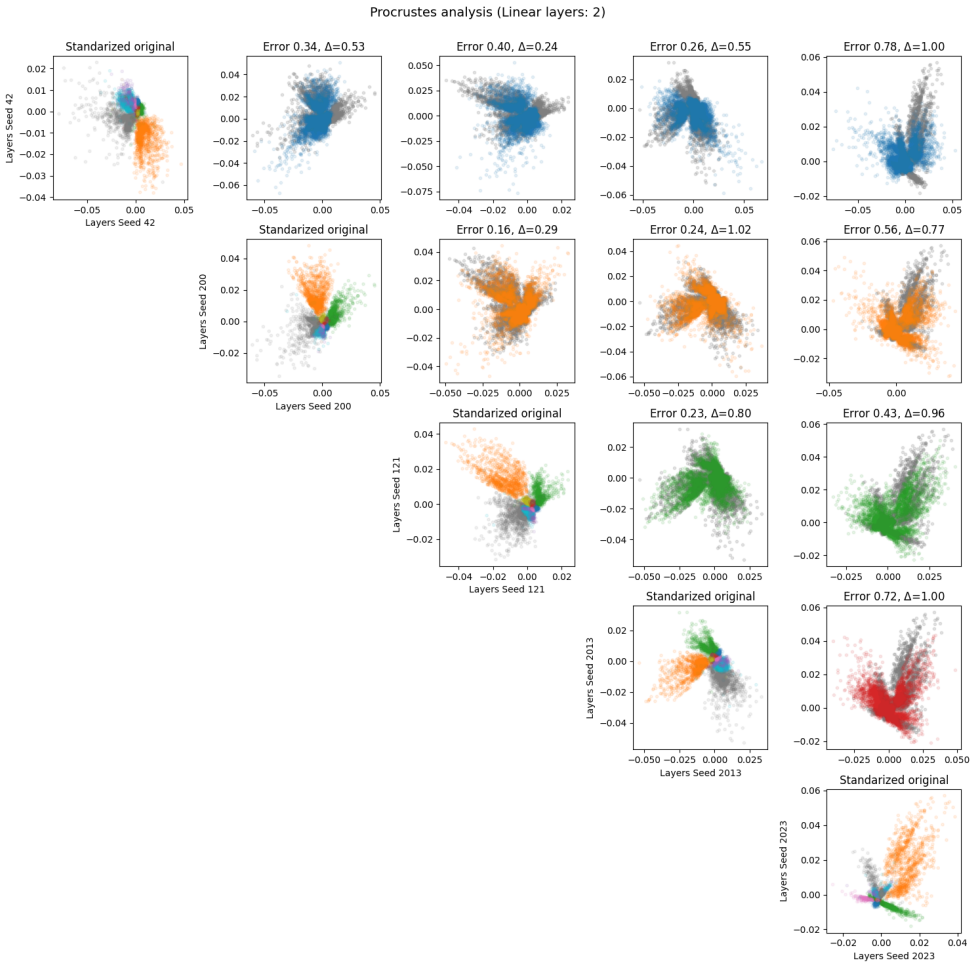


Figure A.3: Additional Procrustes analysis for the autoencoder with linear layers: 2

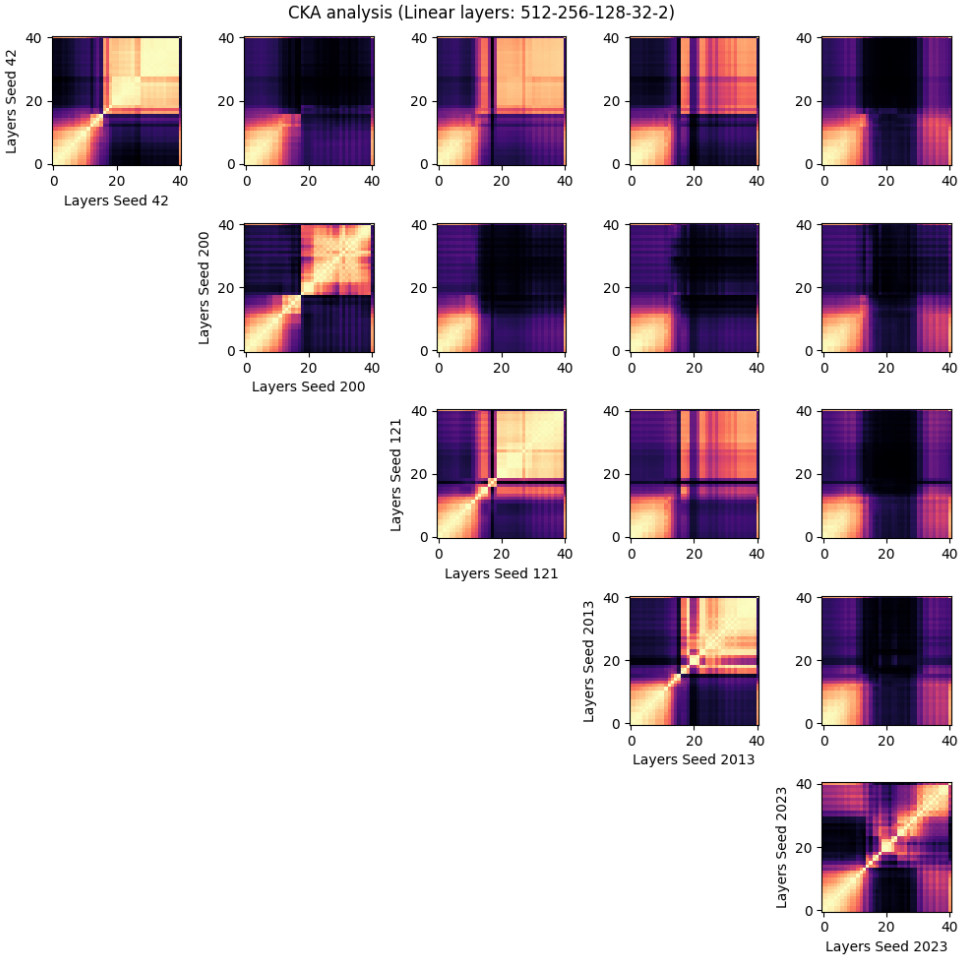


Figure A.4: Additional CKA analysis for the autoencoder with linear layers: 512-256-128-32-2

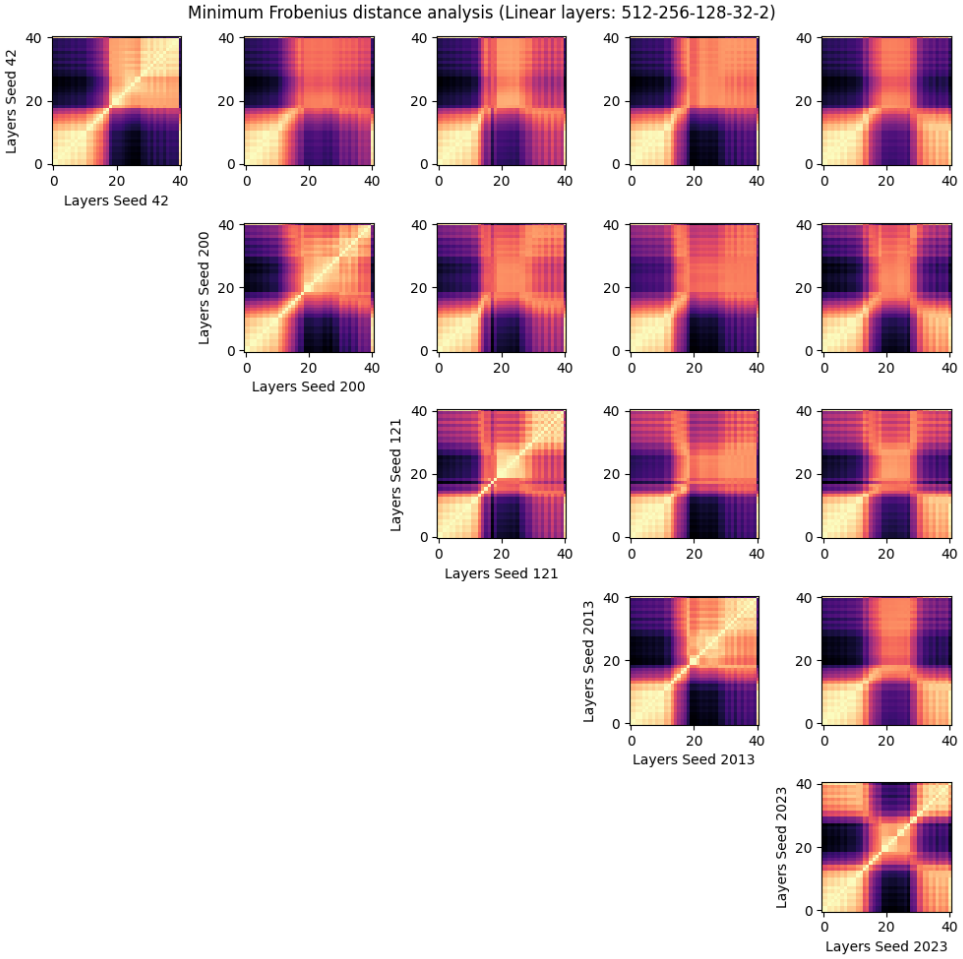


Figure A.5: Additional Min. Frobenius norm analysis for the autoencoder with linear layers: 512-256-128-32-2

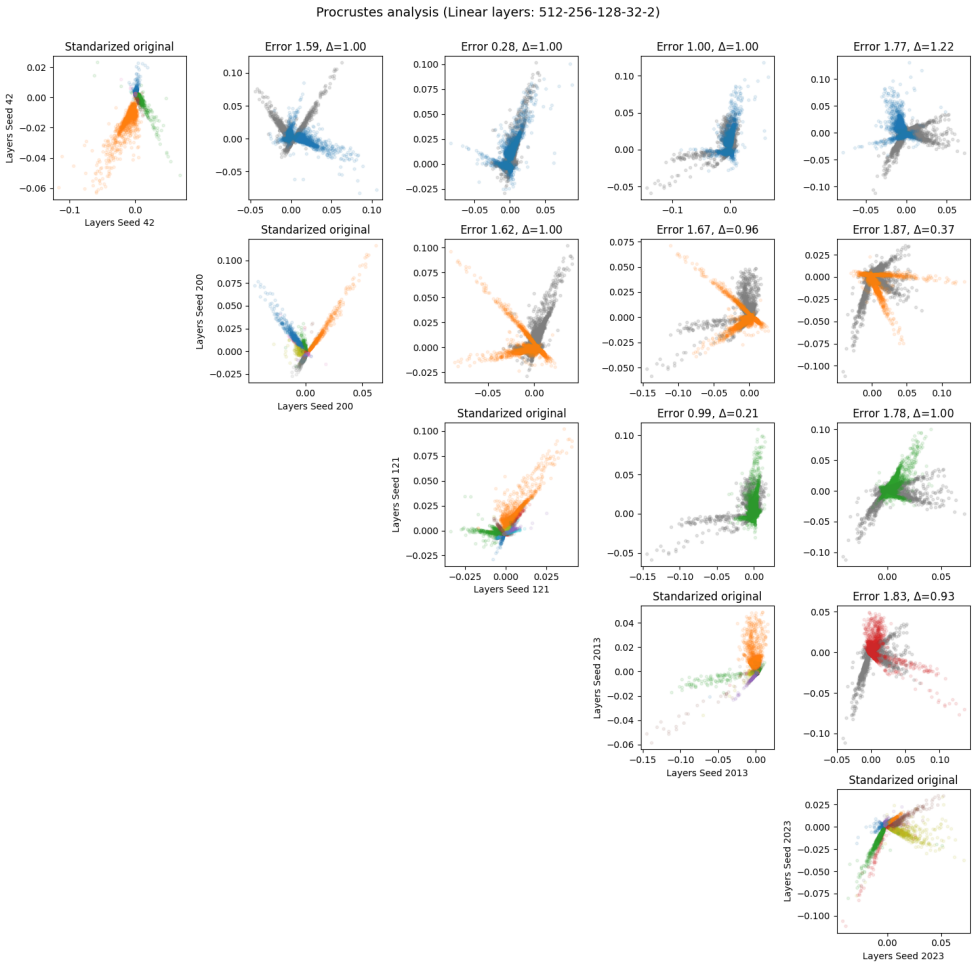


Figure A.6: Additional Procrustes analysis for the autoencoder with linear layers: 512-256-128-32-2

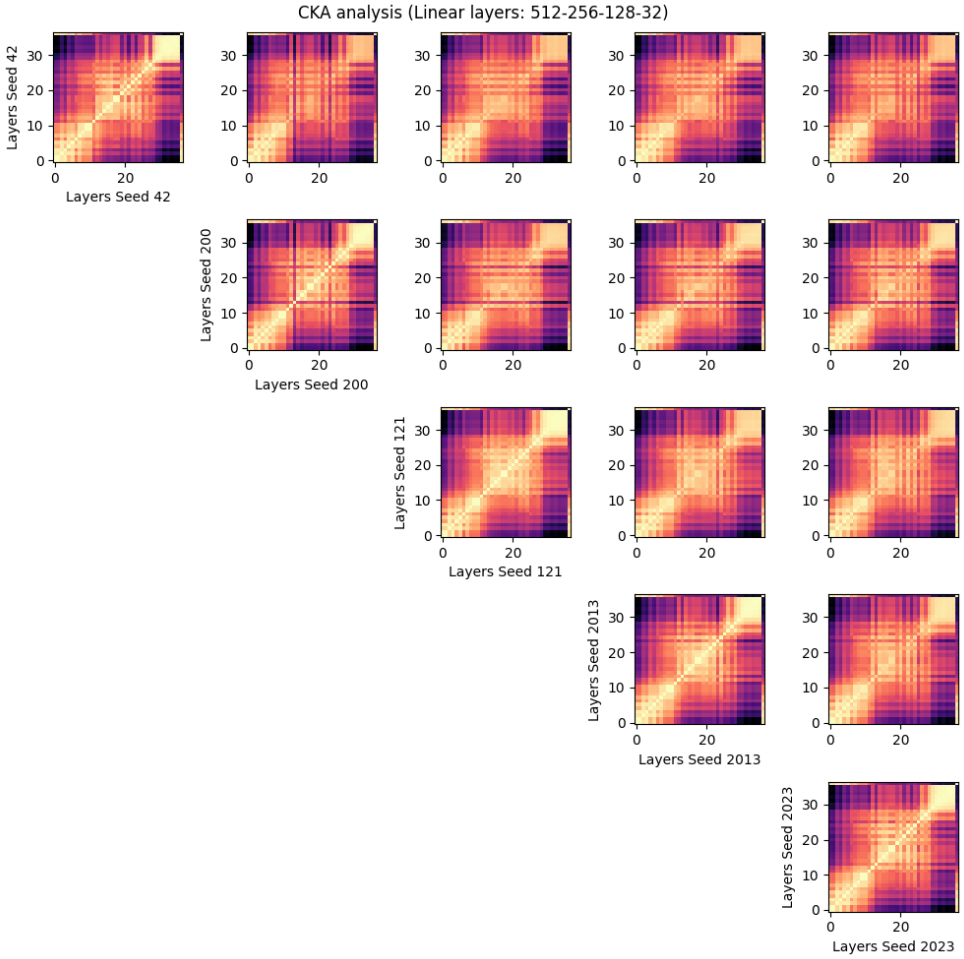


Figure A.7: Additional CKA analysis for the autoencoder with linear layers: 512-256-128-32

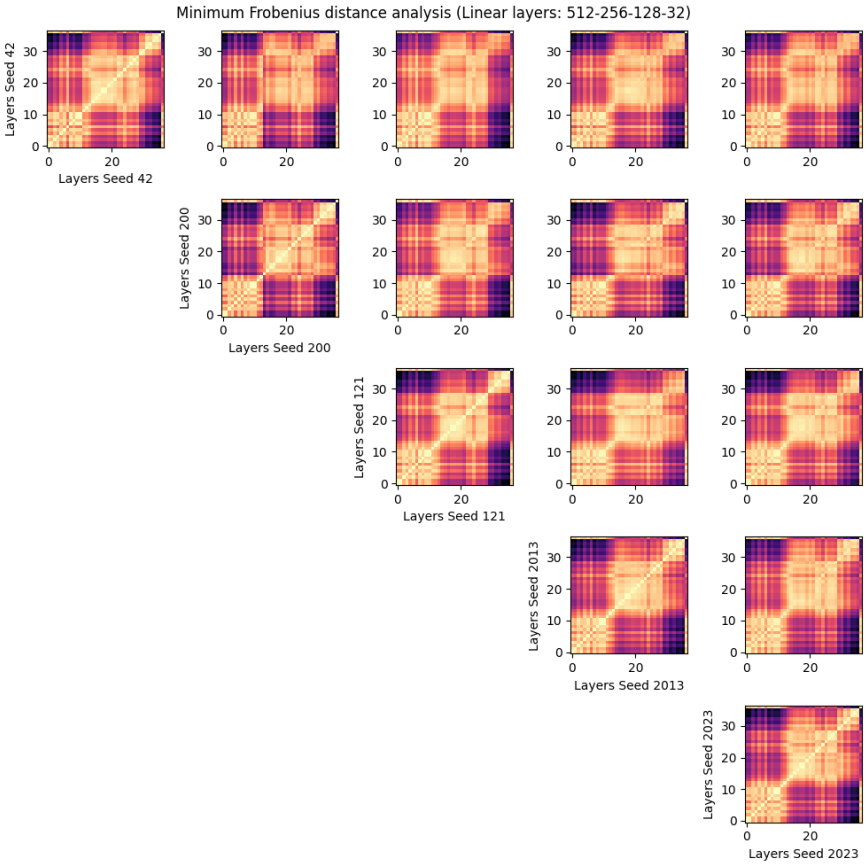


Figure A.8: Additional Min. Frobenius norm analysis for the autoencoder with linear layers: 512-256-128-32

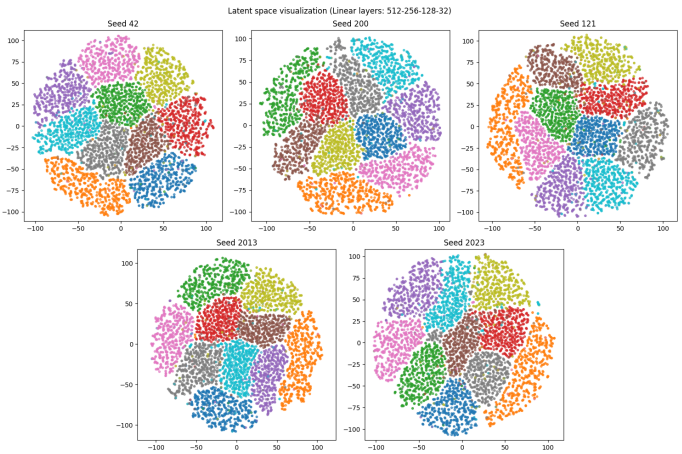


Figure A.9: T-SNE of the encoded latent space for the autoencoder with linear layers: 512-256-128-32

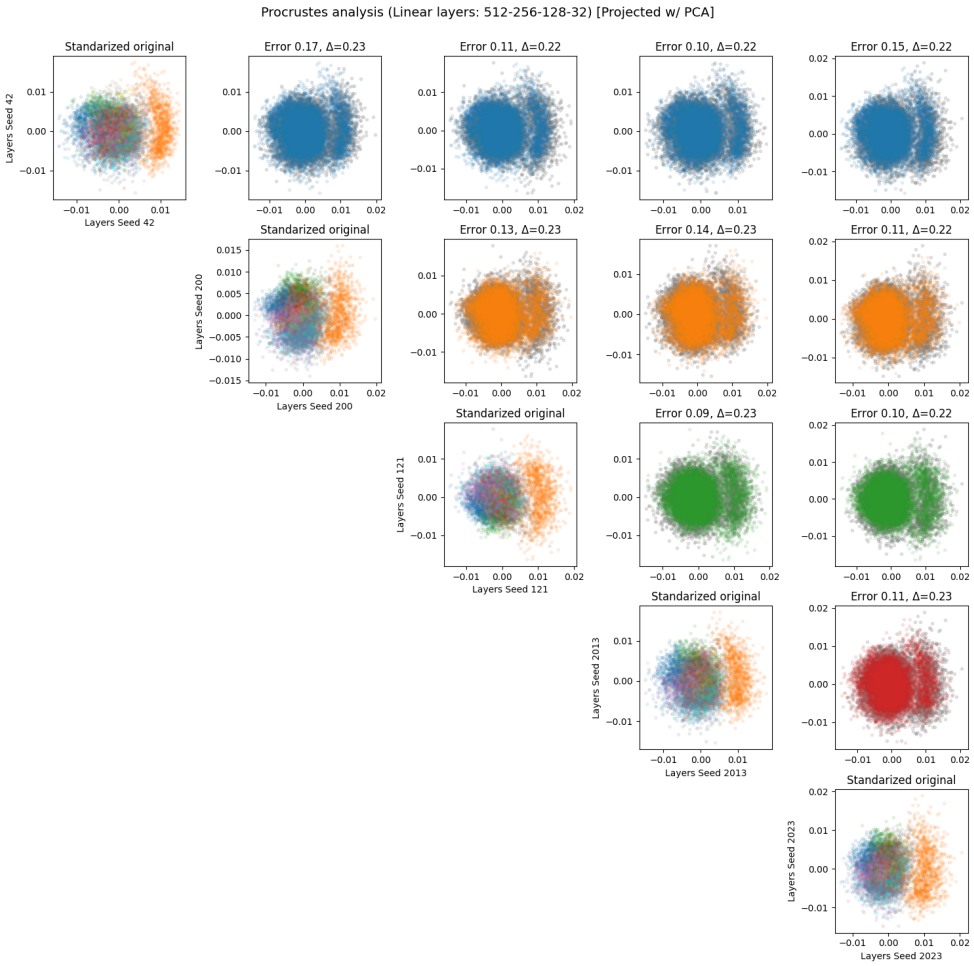


Figure A.10: Additional Procrustes analysis for the autoencoder with linear layers: 512-256-128-32

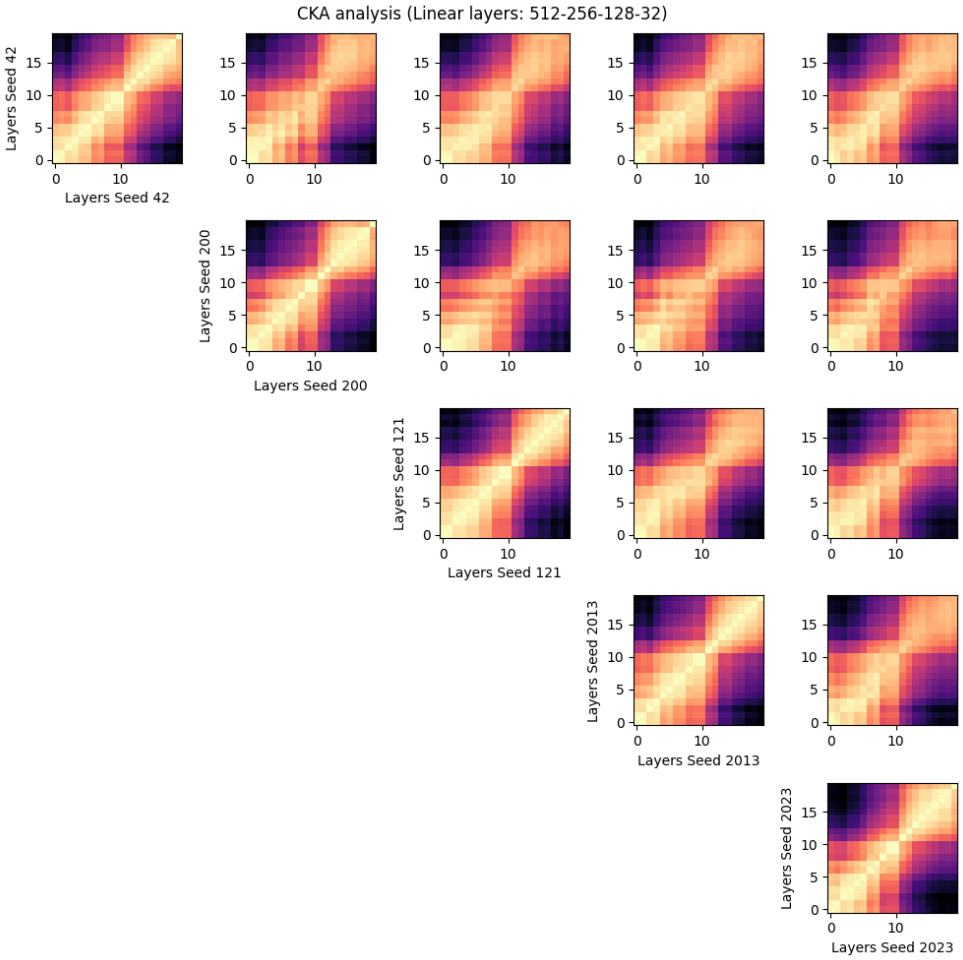


Figure A.11: Additional CKA analysis for the CNN classifier

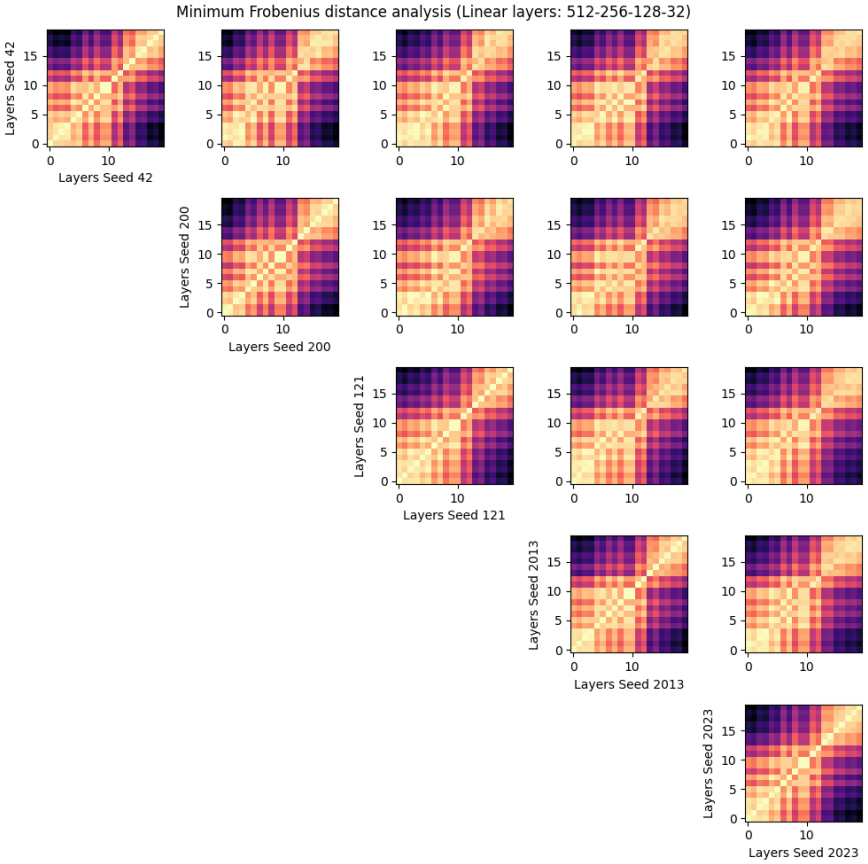


Figure A.12: Additional Min. Frobenius norm analysis for the CNN classifier

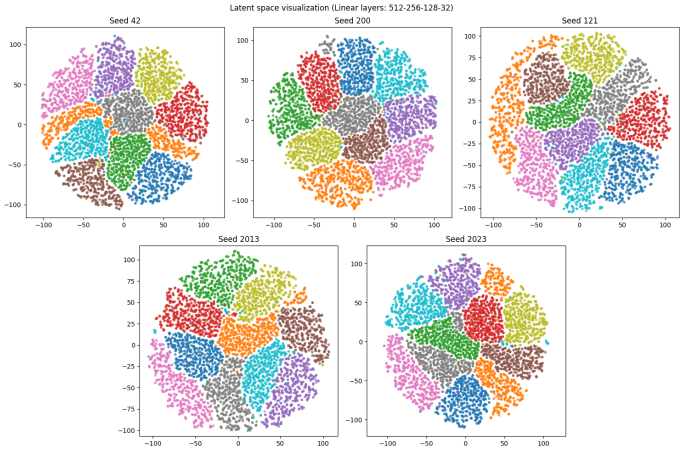


Figure A.13: T-SNE of the encoded latent space for the CNN classifier

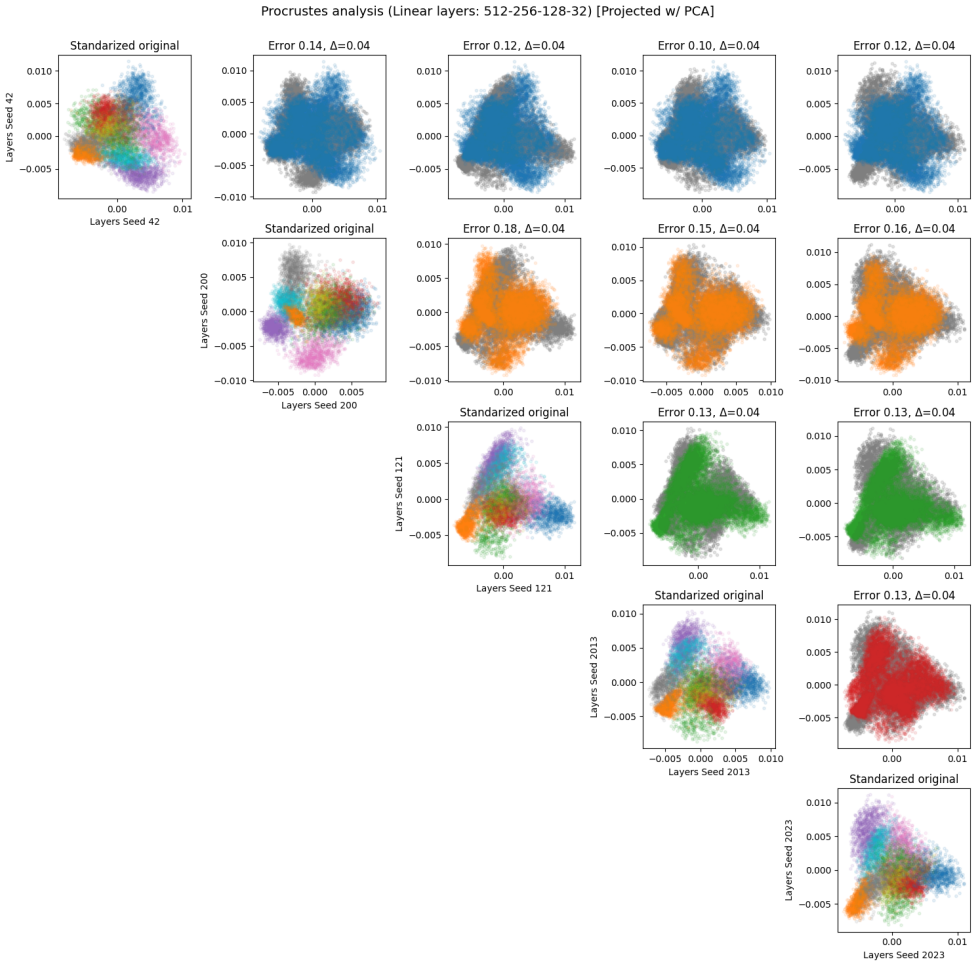


Figure A.14: Additional Procrustes analysis for the CNN classifier

A.2 Multilingual model stitching

		Absolute			Relative		
Decoder	Encoder	Acc × 100	FScore × 100	MAE × 100	Acc × 100	FScore × 100	MAE × 100
en	en	92.00 ± 0.07	92.00 ± 0.07	8.00 ± 0.07	89.24 ± 1.79	89.24 ± 1.79	10.76 ± 1.79
	es	49.91 ± 0.02	33.32 ± 0.02	50.09 ± 0.02	83.70 ± 1.48	83.66 ± 1.52	16.30 ± 1.48
	fr	52.95 ± 0.14	42.81 ± 0.83	47.05 ± 0.14	78.85 ± 0.49	78.81 ± 0.54	21.15 ± 0.49
es	en	41.15 ± 5.76	31.48 ± 1.03	58.85 ± 5.76	76.21 ± 3.02	76.10 ± 3.01	23.79 ± 3.02
	es	91.74 ± 0.94	91.72 ± 0.95	8.26 ± 0.94	90.36 ± 0.02	90.35 ± 0.02	9.64 ± 0.02
	fr	47.81 ± 1.15	43.70 ± 2.93	52.19 ± 1.15	77.26 ± 0.90	77.22 ± 0.92	22.74 ± 0.90
fr	en	50.28 ± 0.28	35.07 ± 2.22	49.73 ± 0.28	81.80 ± 1.66	81.79 ± 1.66	18.20 ± 1.66
	es	50.34 ± 0.05	34.59 ± 0.47	49.66 ± 0.05	81.01 ± 0.94	80.85 ± 0.95	18.99 ± 0.94
	fr	87.62 ± 0.85	87.60 ± 0.89	12.38 ± 0.85	85.61 ± 0.62	85.60 ± 0.62	14.39 ± 0.62

Table A.1: Coarse grained: finetune (over two random seeds)

		Absolute			Relative		
Decoder	Encoder	Acc × 100	FScore × 100	MAE × 100	Acc × 100	FScore × 100	MAE × 100
en	en	93.33 ± 1.80	93.33 ± 1.80	6.67 ± 1.80	91.56 ± 2.81	91.56 ± 2.80	8.43 ± 2.81
	es	54.41 ± 6.15	44.39 ± 15.17	45.59 ± 6.15	88.21 ± 5.76	88.19 ± 5.78	11.79 ± 5.76
	fr	41.82 ± 15.29	34.55 ± 11.59	58.18 ± 15.29	85.00 ± 8.59	84.97 ± 8.62	15.00 ± 8.59
es	en	45.30 ± 8.25	37.10 ± 8.05	54.69 ± 8.25	83.22 ± 10.71	83.15 ± 10.77	16.78 ± 10.71
	es	93.20 ± 1.79	93.20 ± 1.80	6.79 ± 1.79	92.01 ± 2.25	92.00 ± 2.25	7.99 ± 2.25
	fr	41.22 ± 9.42	37.15 ± 8.19	58.78 ± 9.42	83.88 ± 9.37	83.85 ± 9.39	16.12 ± 9.37
fr	en	50.20 ± 0.26	34.69 ± 1.79	49.80 ± 0.26	87.02 ± 6.69	87.01 ± 6.69	12.98 ± 6.69
	es	52.56 ± 3.07	40.35 ± 8.05	47.44 ± 3.07	86.18 ± 7.40	86.08 ± 7.49	13.82 ± 7.40
	fr	90.42 ± 3.58	90.40 ± 3.59	9.58 ± 3.58	89.14 ± 4.64	89.13 ± 4.64	10.86 ± 4.64

Table A.2: Coarse grained: full (over two random seeds)

		Absolute			Relative		
Decoder	Encoder	Acc × 100	FScore × 100	MAE × 100	Acc × 100	FScore × 100	MAE × 100
en	en	61.01 ± 0.16	60.68 ± 0.36	46.34 ± 0.14	61.19 ± 0.58	61.14 ± 0.68	45.08 ± 0.42
	fr	35.17 ± 6.07	26.32 ± 6.43	92.28 ± 4.38	52.63 ± 0.04	52.14 ± 1.03	56.55 ± 1.97
fr	en	29.16 ± 2.57	27.71 ± 3.72	112.68 ± 8.97	60.35 ± 0.55	60.40 ± 0.20	45.67 ± 0.44
	fr	52.63 ± 0.35	52.29 ± 0.88	56.60 ± 0.40	52.72 ± 0.11	52.90 ± 0.10	55.88 ± 0.99

Table A.3: Linear vanilla dataloader w/ early stopping (over two random seeds)

		Absolute			Relative		
Decoder	Encoder	Acc $\times$ 100	FScore $\times$ 100	MAE $\times$ 100	Acc $\times$ 100	FScore $\times$ 100	MAE $\times$ 100
en	en	59.78 ± 0.45	59.04 ± 0.04	48.31 ± 0.04	61.60 ± 0.57	61.43 ± 0.52	44.02 ± 0.25
	fr	31.03 ± 0.83	22.50 ± 0.07	109.80 ± 14.93	51.34 ± 0.23	51.62 ± 0.51	56.32 ± 0.08
fr	en	30.43 ± 4.91	27.09 ± 4.91	112.66 ± 1.95	60.91 ± 0.13	60.52 ± 0.13	44.75 ± 0.38
	fr	51.67 ± 0.81	52.17 ± 0.56	57.09 ± 0.07	51.59 ± 0.58	51.60 ± 0.35	56.75 ± 0.38

Table A.4: Linear biased dataloader w/ early stopping (over two random seeds)

TRITA-EECS-EX- 2023:0000