

RICE UNIVERSITY

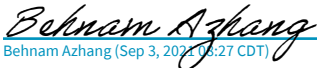
By

Romain Cosentino-Faugere

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

APPROVED, THESIS COMMITTEE



Behnam Aazhang (Sep 3, 2021 9:27 CDT)

Behnaam Aazhang



AnirvanMayukhSengupta (Sep 7, 2021 10:46 EDT)

Anirvan Sengupta



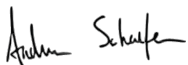
Richard Baraniuk (Sep 2, 2021 20:38 CDT)

Richard Baraniuk



Yanis Bahroun (Sep 3, 2021 12:20 EDT)

Yanis Bahroun



Andrew Schaefer



Hervé Glotin (Sep 3, 2021 11:10 GMT+2)

Herve Glotin

HOUSTON, TEXAS

September 2021

ABSTRACT

How to Group: from Time Series to Manifold

by

Romain Cosentino-Faugere

This thesis addresses the problem of data representation for pattern recognition by focusing on three fundamental properties: the efficiency, adaptivity, and interpretability of the representation. The century of progress in harmonic analysis led to the development of theoretically sounded and interpretable tools to decompose, analyze, and process signals. Nevertheless, these tools have now shown their limitation in terms of expressive power and flexibility. The last decade of research in pattern recognition has been revolutionized by the myriad of results that Deep Learning (DL) algorithms provided, helping us to understand better how one can build an efficient data representation. Among the intuitions that DL approaches provided, which most of them are yet to be proven, we will focus on the fact that an efficient representation of the data should be learned jointly with the task at hand. While DL provides the framework and practical tools that enable the efficiency and adaptivity of the representation, it lacks interpretability and theoretical guarantees. By intersecting harmonic analysis and deep learning, the work undertaken in this thesis explores the possibility of providing Deep Harmonic Learning tools, where the interpretability is driven by our profound knowledge of harmonic analysis techniques, and DL techniques drive the flexibility.

The first objective is to explore the generalization and learnability of the wavelet transform. Our approach decomposes this task by considering the wavelet transform as two building blocks: a mother wavelet and a group. We first propose to tackle the learnability of the mother wavelet exploiting the efficient representation of the mother wavelet using the Hermite cubic spline as a basis. This approach, both efficient and learnable, is used to replace the first layer of Deep Neural Networks (DNNs) and proved its performance on a large-scale pattern recognition task. Then, we consider the learnability of the group by which the mother wavelet is transformed to produce the filter bank. This approach allows for the learnability of intricate correlations that are often aligned with the symmetry of the data. Again, the replacement of the first layer of DNNs by these adaptive filters provides state-of-the-art results on various datasets.

The second objective of this thesis is to explore the approximation and quantization of manifolds by exploiting the assumption that the data manifold is governed by a symmetry group. Our approach is two-fold: firstly, we provide a quantizer of the image manifold that is based on the learnability of the non-rigid transformations governing the images. In particular, we build a metric aware of these intricate transformations, and that can adapt to the data at hand. The challenge of learning in an unsupervised fashion appropriate invariance is tackled by exploiting the intuitive parameterization that the Thin-Plate-Spline interpolation method offers. The resulting shallow clustering algorithm is fully interpretable and achieves performances comparable to its deep learning counterpart. Secondly, we provide a new approach to perform manifold approximation with generalization guarantees. This is achieved by exploiting the piecewise continuous approximation property of autoencoders which can be constrained to be equivariant to a group of transformations. Again, we consider the learnability of the group underlying the data to steer the equivariance. The equivariant autoencoder we propose achieves state-of-the-art results on a large number of datasets.

Acknowledgments

I would first like to thank my advisor, Dr. Behnaam Aazhang, for encouraging me to start this Ph.D., supporting me during my research, and most importantly, believing in my ideas. I would like to thank Dr. Richard Baraniuk for always sharing his cheerfulness and his french rhetoric. A particular thanks to Dr. Andrew Schaefer for his kindness and for supporting me during these years. Thanks to Cj, to have tried to introduce me to a part of the Texan culture. Thanks to my colleagues, Sudha, Negar, Fatima, Anton, Boqiang, for all the discussions and support. An immense thanks to Dr. Randall Balestrieri, with whom this adventure would not have been possible, thank you for pushing me to do "tractable" research, to have brought me on this continent, and for attempting to chew on whatever ideas we could come with.

A special thanks to Dr. Yanis Bahroun with whom I shared interesting and insightful discussions, for introducing me to the Flatiron CCB team, and for helping me shape this document. I am grateful to Dr. Arnivan Sengupta, for sharing his invaluable knowledge, always being extremely patient, friendly, and attempting to understand and add value to my ideas. I would like to thank Dr. Hervé Glotin, for his kindness. I would also like to sincerely thank Dr. Christine De Mol for introducing me to machine learning, pushing me from the beginning to explore this path, as well as for helping me to do so. A particular thanks to a rare Tarn breed, Dr. Léonard

Seydoux, for sharing his passion for his field, his knowledge, and his friendship.

I would like to thank each member of the jury for their help, advice, and time.

A great thanks to Dr. Harry Sevi, with whom I share many passions, values, and discussions. Thanks to Pierro, for being supportive and mentoring me since my childhood and always having the right advice.

I am extremely grateful to my mother, father, brother, and grandparents for all the values they shared with me, their patience, and eternal support. My deepest gratitude goes to my grandfather, Guy, for his genuinely humble and inspiring passion for learning.

Finally, I owe a deep sense of gratitude to my wife, H el ene, for being here and being her.

Contents

Abstract	ii
Acknowledgments	iv
1 Introduction	1
1.1 Motivation	1
1.2 Gaps and Research Questions	4
1.3 Contributions	7
1.4 Organisation & Publications	9
2 Background & Pre-requisites	14
2.1 Wavelet Transform	14
2.2 Group Transform	21
2.3 Lie Group Transformations	24
2.4 Thin Plate Spline Interpolation	26
3 Learnable Wavelet Transform	30
3.1 Outline of the Chapter and Contribution Summary	32
3.2 Related Work	33
3.3 Formalism	33
3.4 Properties	37
i Existence & Uniqueness	37
ii Space Dimensions	38
iii Filter-bank Derivation	39

3.5	Experiments	40
	i Task	40
	ii Filters Implementation	41
	iii Architecture Comparison	44
	iv Complexity & Parameters	49
	v Results	49
3.6	Conclusion	50
4	Learnable Group Transform	52
4.1	Outline of the Chapter and Contribution Summary	53
4.2	Related Work	54
4.3	Formalism	56
4.4	Properties	59
	i Recovering Standard Filter Banks	59
	ii Equivariance Properties of the learnable group transform	60
4.5	Experiments	63
	i Sampling the group	63
	ii Objective Function and Learning:	63
	iii Model Constraints to Reduce Aliasing	64
	iv Classification of chirp signals	66
	v Supervised Bird Detection Task	68
	vi Classification of haptics data	70
4.6	Conclusion	72
5	Learnable Invariant Distance	74
5.1	Outline of the Chapter and Contribution Summary	76
5.2	Related Work	77
5.3	Formalism	77

5.4	Image Transformations	79
5.5	Learning the Spatial Transformer K -means	81
5.6	Properties & Geometrical Aspects	83
	i Quasi-pseudo-semi Metric	83
	ii Invariance Property	85
	iii Convergence of the Spatial Transformer	
	K-means	86
	iv Geometrical Interpretation of the Similarity	
	Measure	87
	v Complexity & Parameters	89
5.7	Experiments	90
	i Evaluation Methods	93
	ii Cross-validation Settings	93
	iii Results	94
	iv Interpretability: Centroids Visualization	95
	v Interpretability: Embedding Visualization	96
5.8	Conclusion	97
6	Learnable Lie Group for Manifold Approximation	100
6.1	Outline of the Chapter and Contribution Summary	102
6.2	Related Work	103
6.3	Formalism	105
6.4	Properties & Geometrical Aspects	107
	i Reconstruction Guarantees	107
	ii Tangents and Hessians	109
	iii Interpretability of Regularization Techniques	111
	iv Lie Group Orbit Fitting	114
	v Regularizations for Continuous Piecewise	
	Affine Maps	116

	vi	Manifold Approximation Error	121
	vii	Complexity & Parameters	124
6.5		Experiments	126
	i	Parameters	126
	ii	Results	127
6.6		Conclusion	130
7		Discussion & Conclusion	131
	7.1	Discussion	131
	7.2	Conclusion & Future work	134
		7.2.1 Conclusion	134
		7.2.2 Future Work	136
		Bibliography	138

List of Figures

2.1	Morlet Wavelet	17
2.2	Gammatone Wavelet	19
2.3	Paul Wavelet	21
3.1	Hermite cubic spline: cubic polynomial on a close interval	42
3.2	Concatenation of Hermite cubic splines	43
3.3	Insuring continuity and smoothness of the filter	43
3.4	From the smooth filter to wavelet	44
3.5	From the mother wavelet to the filter-bank	44
3.6	Training accuracy w.r.t epochs for FreeField dataset	46
3.7	Learned filters: CNN vs Spline filter	47
3.8	First layer representation: MFSC vs CNN vs Spline filters	48
4.1	Time-frequency tilings	55
4.2	Learnable group transform - Summary	57
4.3	Transformation of a Morlet wavelet	58
4.4	Learnable group transform filters - artificial dataset	65
4.5	Artificial as/de-scending chirp	66
4.6	Learnable group transform visualizations	69
4.7	Learnable group transform filters - bird dataset	69
4.8	Learnable group transform filter - haptics dataset	71

5.1	Image transformations	80
5.2	Orbit of a hand-written digit 7 according to the group of rotation $SO(2)$	81
5.3	Centroids	90
5.4	t-SNE projections	91
5.5	Computational time	92
5.6	Accuracy vs distortion error	94
6.1	Autoencoder input space's partitioning	104
6.2	Autoencoder manifold	110
6.3	Number of data VS regions inside a ball	113
6.4	Test set reconstruction error	125
6.5	Group element parameters	129

List of Tables

3.1	Classification results - bird detection	50
4.1	Recovering well-known filters	59
4.2	Testing Accuracy for the Chirp Signals Classification Task	66
4.3	Testing AUC for the Bird Detection Task	68
4.4	Testing Accuracy for the Haptics Classification Task	72
5.1	Clustering accuracy	99
6.1	Autoencoders accuracy	127

1

Introduction

1.1 Motivation

The last decade of research in machine learning that this work is inscribed in might be described as one of the most experimental and prosperous periods of all time in artificial intelligence. The performances of recent machine learning algorithms are outperforming any of their ancestors in a large number of applications [1]. In particular, the capability of machines to perform perceptual tasks such as speech and image recognition are now at the level of human performances [2–4] if not superhuman for certain specific tasks [5, 6]. This huge step forward pushing further the boundaries of machine learning is mainly due to Deep Neural Networks (DNNs) extensive development. Behind their creation, their wish was to provide a DNN that would take inspiration from the brain’s functioning behind pattern recognition tasks and inform about its mechanism [7].

In particular, two major concepts that were known to neuroscientists as the key

elements to describe the representations that the brain might perform [8] have been digitalized: the equivariance and invariance of a representation with respect to specific transformations. In particular, convolutional neural networks [9] were conceived to integrate these concepts as part of the representation of the data they enable. Twenty years later, in [10], S. Mallat provided its harmonic analysis description along with a specific deep network architecture: the scattering network. This network architecture that looks, from far to a convolutional neural network and close to an intricate time-frequency based filtering representation, highlights the importance of two concepts that a salient representation of the data should yield, the equivariance and invariance with respect to groups. Whether it is from an experimental point of view or a more theoretical approach, these approaches reinforce the belief that machine learning algorithms' performances are mainly tied to the embedding that the data are transformed with. While the requirement for an appropriate embedding is now better than ever theoretically motivated, their use in signal processing has been there for decades and in human history for at least 73.000 years.

In our quest to understand what aspect of deep learning can be leveraged to develop harmonic analysis based algorithms that can be interpretable, efficient, and adaptive, we propose to consider the following approach to pattern recognition.

A pattern recognition task usually induces on the data a specific set of nuisances that should be removed to ease the detection of salient features by the classifier. For instance, in the context of image recognition, the pose of an object can be categorized as a nuisance, while in the case of pose estimation, the class is the nuisance. Similarly, in the case of time series classification, such as the diagnosis of heart disease via ECG recordings, the age of the patient and its sex are among the set of nuisances.

Therefore, one is interested in “observing” the data in a space where such nuisances

are aligned with some of the coordinate axes. The salient features of the data allowing to perform the pattern recognition task are then obtained by averaging along the nuisances axis. This simplified and intuitive idea depicts the importance of an appropriate change of basis, highlighting the different representations the data can take.

An important field of mathematics that has been used for decades to describe how an entity varies under the action of specific transformations, called a group, can be used to describe how one can use equivariant representations to better represent the data so that the invariance required to perform the pattern recognition task can be easily achieved. A group is a set that consists of all the elements that could produce a particular type of transformation, such as rotation, permutation, and translation. The question is now, how can one represent the data with respect to a basis that aligns with the different groups composing the symmetry of the data? An intuitive way of representing the data in accordance with this principle is to project the data onto the group underlying all the important transformations that the object undergoes without altering its identity. Such a projection enables the localization of the data on an interpretable manifold. One can easily assume that one or more of these manifolds exist for each class and that each of them has a specific set of transformations. For instance, one can consider all the images of the handwritten digit 4 as a manifold where the location of the sample on this manifold depends on the pose, lighting, and deformation relative to an implicitly defined canonical version of that entity [11]. Now, given the coordinate on this local manifold, one can easily produce an invariant representation by averaging over one coordinate axis.

The development of an operator that projects the data onto the group is therefore crucial. This operator would ease the computation of interpretable invariant

representations, which are now known to be crucial to perform pattern recognition.

1.2 Gaps and Research Questions

Two of the machine learning pillars helping us understand practically and theoretically these considerations are convolutional neural networks and the scattering network. While convolutional neural networks are fully adaptive and transformed the machine learning scene, the scattering network lacks expressive power but provides a fully interpretable representation of the data. The development of methods in between these approaches, trying to build an interpretable and efficient representation of the data driven by the task at hand, was almost absent from the literature at the commencement of this Ph.D.

In this section, we propose to briefly describe the main limitations and challenges that need to be addressed to learn an appropriate representation of the data for pattern recognition tasks. We are in particular interested in the learnability of appropriate representations for time-series as well as for manifold.

In the case of pattern recognition for time series data, the traditional wavelet transform remains one of the most used data representations because of its capability of decomposing the signal with respect to a structured basis that yields a sparse representation of a large number of signals [12]. Its wide application led to the development of various wavelets that often resonate well with specific applications [13]. Naturally, the selection of the mother filter, which required expert knowledge on the data at hand, has been progressively replaced by an automated search for the optimal filter among a selected list of mother wavelets [14–16]. However, these pre-processing techniques were derived for goals not necessarily aligned with the current tasks at hand (reconstruction, compression, classification), thus do not provide an adapted

solution. Later, attempts to provide end-to-end filter learning has been investigated. In particular, in [17,18] they consider the learning of parameters inherent to time-frequency representations such as Mel-Frequency Spectral Coefficients; however, their approaches were applied to datasets that did not contain external nuisances and did not provide a generalization of well known harmonic analysis methods.

Understanding the dataset as a whole and an attempt to understand its structure has been the center of manifold learning algorithms. We are particularly interested in two aspects, the approximation of the manifold and its quantization. Following that goal, various approaches have been developed: Kernel PCA [19], Isomap [20], Laplacian eigenmaps [21], Locally Linear Embedding [22]. Most of the approaches developed during that period were based on the idea that, using neighboring data, one can appropriately interpolate and approximate the curvature of the data manifold. These local methods and their drawbacks have been precisely pinpointed in [23]. In particular, we know that most of the data available to us are high-dimensional, and therefore the estimation of distance is challenging as per the curse of dimensionality and the large number of nuisances that need to be removed. To alleviate such drawbacks, one approach consists of augmenting the data by using the symmetry of the data [24,25]. Now, the problem is that this requires the knowledge of the symmetry of the data that are aligned with the task at hand. Considering and learning these groups is crucial to perform an accurate unsupervised approximation and quantization of the data manifold.

From time-series to manifold, finding an appropriate way to learn the representation of the data is the milestone of efficient machine learning algorithms. Following decades of development in harmonic analysis, statistics, and artificial intelligence, the question that this thesis attempt to tackle is

Is it possible to build an adaptive algorithm capable of representing the data efficiently such that: the representation is interpretable, theoretically grounded, and efficient?

In order to approach the answer to this question, we consider the following subquestions that will help us understand how such a goal can be achieved.

1. The wavelet transform of time series data has been used for decades for its efficiency to capture salient features, its interpretability, and its tractability; how can one provide a framework that allows their generalization and learnability while conserving the tractability and interpretability of the representation?
 - 1.1. Is it possible to appropriately learn a mother wavelet that characterizes the salient feature of the data?
 - 1.2. Is it possible to learn the group governing the plane onto which the signal is projected when performing a wavelet transform?
2. The K -means algorithm remains one of the most used clustering algorithms because of its interpretability and tractability; how can one equip it with a metric that alleviates the drawback of the Euclidean distance and appropriately capture the geometry of the data?
 - 2.1. Is it possible to learn an invariant metric with respect to non-rigid transformations in a fully unsupervised fashion?
 - 2.2. Does the theoretical convergence guarantees of the K -means algorithm hold when one replaces the Euclidean distance with an adaptive metric?
3. Autoencoders are state-of-the-art algorithms capable of encoding the underlying manifold of the data; how can one equip them with generalization guarantees?

- 3.1. Is there any formulation of autoencoders that enable us to characterize how they approximate the manifold, and what is their limitation?
- 3.2. Is it possible to enforce their generalization under the assumption that the data form a homogeneous space?
- 3.3. How can one learn the symmetry of the data and incorporate it into the manifold approximation performed by autoencoders?

1.3 Contributions

The research work carried out to accomplish the outlined research objectives has resulted in several original contributions to the field of machine learning. The contributions of this thesis are detailed in the following.

Contribution I: Learning the Mother Wavelet We propose to tackle the problem of end-to-end learning for raw waveform signals by introducing learnable continuous time-frequency atoms. The derivation of these filters is achieved by defining a functional space with a given smoothness order and boundary conditions. From this space, we derive the parametric analytical filters using cubic Hermite splines. Their differentiability property allows gradient-based optimization. As such, one can utilize any Deep Neural Network (DNN) with these filters. This enables us to tackle in a front-end fashion a large-scale bird detection task based on the freefield1010 dataset known to contain key challenges, such as the dimensionality of the input data ($> 100,000$) and the presence of additional noises: multiple sources and soundscapes. This contribution is reported in chapter 3.

Contribution II: Learnable Group Transform We propose a novel approach to filter bank learning for time-series by considering spectral decompositions of signals defined as a Group Transform. This framework allows us to generalize classical time-frequency transformations such as the Wavelet Transform and efficiently learn the representation of signals. While the creation of the wavelet transform filter-bank relies on affine transformations of a mother filter, our approach allows for non-linear transformations. The transformations induced by such maps enable us to span a larger class of signal representations, from wavelet to chirplet-like filters. We propose a parameterization of such a non-linear map such that its sampling can be optimized for a specific task and signal. The Learnable Group Transform can be cast into a Deep Neural Network. The experiments on diverse time-series datasets demonstrate the expressivity of this framework, which competes with state-of-the-art performances. This contribution is reported in chapter 4.

Contribution III: Learnable Invariant Distance K-means defines one of the most employed centroid-based clustering algorithms with performances tied to the data’s embedding. Intricate data embeddings have been designed to push K-means performances at the cost of reduced theoretical guarantees and interpretability of the results. Instead, we propose preserving the intrinsic data space and augment K-means with a similarity measure invariant to non-rigid transformations. This enables (i) the reduction of intrinsic nuisances associated with the data, reducing the complexity of the clustering task and increasing performances and producing state-of-the-art results, (ii) clustering in the input space of the data, leading to a fully interpretable clustering algorithm, and (iii) the benefit of convergence guarantees. This contribution is reported in chapter 5.

Contribution IV: Learnable Lie Group for Manifold Approximation A big mystery in deep learning continues to be the ability of methods to generalize when the number of model parameters is larger than the number of training examples. In this work, we take a step towards a better understanding of the underlying phenomena of Deep Autoencoders (AEs), a mainstream deep learning solution for learning compressed, interpretable, and structured data representations. In particular, we interpret how AEs approximate the data manifold by exploiting their continuous piecewise affine structure. Our reformulation of AEs provides new insights into their mapping, reconstruction guarantees, as well as an interpretation of commonly used regularization techniques. We leverage these findings to derive two new regularizations that enable AEs to capture the inherent symmetry in the data. Our regularizations leverage recent advances in the group of transformation learning to enable AEs to better approximate the data manifold without explicitly defining the group underlying the manifold. Under the assumption that the symmetry of the data can be explained by a Lie group, we prove that the regularizations ensure the generalization of the corresponding AEs. A range of experimental evaluations demonstrates that our methods outperform other state-of-the-art regularization techniques. The contribution is reported in chapter 6.

1.4 Organisation & Publications

The thesis is organized as follows. Chapter 2 covers the necessary background on wavelets, group transform, and other specific tools that we use in this thesis. For a review of machine learning and deep learning, the reader should refer to [26, 27]. In chapter 3 and 4, we explore the generalization and learnability of the wavelet transform. While chapter 3 focuses on the learnability of the mother wavelet, chapter 4 allows

for the learnability of another equivariant structure generalizing the assumed affine symmetry in the data.

Then, in chapter 5 we provide a metric that is invariant to learnable non-rigid deformations. That is, we address the problem of unsupervised invariance learning. That metric allows us to provide a fully interpretable and theoretically guaranteed clustering technique competing with state-of-the-art approaches.

Finally, in chapter 6, we propose to understand how autoencoders approximate manifold to develop a regularization enforcing generalization guarantees. In particular, such a regularization adapts the geometry of the manifold spanned by the autoencoder to the group of symmetry governing the data as to provide an adaptive equivariant interpolation method.

This manuscript is then concluded by a discussion in chapter 7.

Most of the work detailed and gathered in this thesis has led to work that was published in peer-reviewed publications.

- “Learnable Group Transform”, *Romain Cosentino, Behnaam Aazhang*, **ICML**.
- “Spline Filters For End-to-End Deep Learning”, *Randall Balestriero**, *Romain Cosentino**, *Hervé Glotin, Richard Baraniuk* (*: equal contribution), **ICML** .
- “Spatial Transformer K -means”, *Romain Cosentino, Randall Balestriero, Yanis Bahroun, Anirvan Sengupta, Richard Baraniuk, Behnaam Aazhang*, **submitted to ICML**
- “Deep Autoencoders: From Understanding to Generalization Guarantees”, *Romain Cosentino, Randall Balestriero, Richard Baraniuk, Behnaam Aazhang*, **MSML**.

- “Universal Frame Thresholding”, *Romain Cosentino, Randall Balestriero, Richard Baraniuk, Behnaam Aazhang*, **IEEE Signal Processing Letters**
- “Sparse Multi-Family Deep Scattering Network”, *Romain Cosentino, Randall Balestriero*, **arXiv**

List of Symbols

\mathbb{R}	Real numbers
\mathbb{N}	Natural numbers
\mathcal{D}_λ	Dilation operator
\mathcal{G}	Group
G	Generator of the group
$\mathcal{T}_I\mathcal{G}$	Lie algebra of the group \mathcal{G}
\odot	Group operation
GL	The general linear group
ρ	Group representation operator
\mathcal{O}	Orbit of a group
\circ	Composition operator
E	Encoder

D	Decoder
$\mathcal{T}_{(.)}$	Tangent space
γ	Smooth curve on manifold
$J[.]$	Jacobian matrix
$H[.]$	Hessian matrix
$d(,)$	Distance
L^2	Space of square integrable functions
C^n	Space of function with n continuous derivative
ψ	Mother wavelet in the time domain
$\hat{\psi}$	Mother wavelet in the frequency domain
J, Q	Number of octave and number of wavelet per octave
ω	Frequency axis
t	Time axis

2

Background & Pre-requisites

2.1 Wavelet Transform

”By oscillating it resembles a wave, but by being localized it is a wavelet”.

Yves Meyer

Wavelets were first introduced for high resolution seismology [28] [29] and then developed theoretically by Meyer et al. [30]. Formally, wavelet is a function $\psi \in \mathbb{L}^2$ such that:

$$\int \psi(t)dt = 0, \tag{2.1}$$

it is normalized such that $\|\psi\|_{\mathbb{L}^2} = 1$. There exist two categories of wavelets, the discrete wavelets and the continuous ones. The discrete wavelets transform are constructed based on a system of linear equation. These equations represent the atom’s property. These wavelet when scaled in a dyadic fashion form an orthonormal atom dictionary. Withal, the continuous wavelets have an explicit formulation and

build an over-complete dictionary when successively scaled. In this work, we will focus on the continuous wavelets as they provide a more complete tool for analysis of signals. In order to perform a time-frequency transform of a signal, we first build a filter bank based on the mother wavelet. This wavelet is names the mother wavelet since it will be dilated and translated in order to create the filters that will constitute the filter bank. Notice that wavelets have a constant-Q property, thereby the ratio bandwidth to center frequency of the children wavelets are identical to the one of the mother. Then, the more the wavelet atom is high frequency the more it will be localized in time. The usual dilation parameters follows a geometric progression and belongs to the following set:

$$\Lambda = \{2^{j/Q}, j = 0, \dots, J \times Q - 1\}$$

. Where the integers J and Q denote respectively the number of octaves, and the number of wavelets per octave.

Having selected a geometric progression ensemble, the dilated version of the mother wavelet in the time are computed as follows:

$$\psi_\lambda(t) = \frac{1}{\lambda} \psi\left(\frac{t}{\lambda}\right), \forall \lambda \in \Lambda$$

, and can be calculated in the Fourier domain as follows:

$$\hat{\psi}_\lambda(\omega) = \hat{\psi}(\lambda\omega), \forall \lambda \in \Lambda$$

.
Notice that in practice the wavelets are computed in the Fourier domain as the wavelet transform will be based on a convolution operation which can be achieved with more efficiency. By construction the children wavelets have the same properties than the mother one. As a result, in the Fourier domain:

$$\hat{\psi}_\lambda = 0, \forall \lambda \in \Lambda$$

. Thus, to create a filter bank that cover all the frequency support, one needs a function that captures the low frequencies contents. The function is called the scaling function and satisfies the following criteria:

$$\int \phi(t)dt = 1$$

. Among the continuous wavelets, different selection of mother wavelet is possible. Each one posses different properties, such as bandwidth, center frequency. This section is dedicated to the development of the families that are important for the analysis of diverse signals.

The Morlet wavelet The Morlet wavelet (Fig. 2.1) is built by modulating a complex exponential and a Gaussian window defined in the time domain by,

$$\psi^M(t) = \pi^{-\frac{1}{4}} e^{i\omega_0 t} e^{-\frac{t^2}{2}}, \quad (2.2)$$

where ω_0 defines the frequency plane. In the frequency domain, we denote it by $\hat{\psi}^M(t)$,

$$\hat{\psi}^M(\omega) = \pi^{-\frac{1}{4}} e^{-\frac{(\omega-\omega_0)^2}{2}}, \forall \omega \in \mathbb{R}^*, \quad (2.3)$$

thus, it is clear that ω_0 defines the center frequency of the mother wavelet.

With associated frequency center and standard deviation denoted respectively by $\omega_c^{\lambda_i}$ and $\Delta^{\lambda_i}\omega$, $\forall j \in \{0, \dots, JQ - 1\}$ are:

$$\omega_c^{\lambda_i} = \frac{\omega_0}{\lambda_i},$$

$$\Delta^{\lambda_i}\omega = \frac{1}{2\lambda_i^2}.$$

Notice that for the admissibility criteria $\omega_0 = 6$, however one can impose that zero-mean condition facilely in the Fourier domain. Usually, this parameter is assign to

the control of the center frequency of the mother wavelet. Then, we are able to vary the parameter ω_0 in order to have different support of Morlet wavelet.

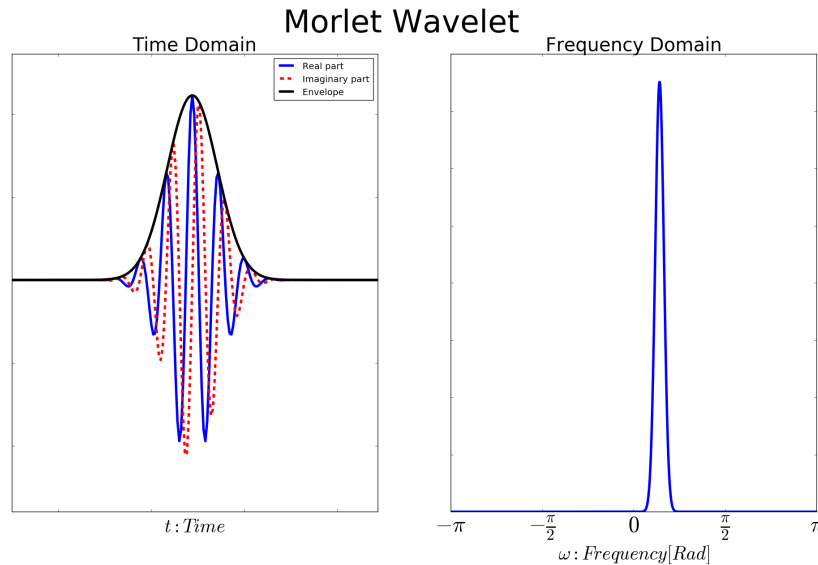


Figure 2.1 : On the left a Morlet wavelet in the time domain where the dashed line is the imaginary part, the solid line is the real part, and the black envelope is the complex modulus, on the right a Morlet wavelet in the frequency domain.

The Morlet wavelet, is optimal from the uncertainty principle point of view [31]. The uncertainty principle, when given a time-frequency atoms, is the area of the rectangle of its joint time-frequency resolution. In the case of wavelet, given the fact that their ratio bandwidth to center frequency is equal implies that this area is equal for the mother wavelets and its scaled versions. As a result, because of its time-frequency versatility this wavelet is widely used for biological signals such as bio-acoustic [32], seismic traces [33], EEG [34] data.

The Gammatone wavelet The Gammatone wavelet is a complex-valued wavelet that has been developed by [35] via a transformation of the real-valued Gammatone auditory filter which provides a good approximation of the basilar membrane filter [36].

Because of its origin and properties, this wavelet has been successfully applied for classification of acoustic scene [37]. The Gammatone wavelet (Fig. 2.2) is defined in the time domain by,

$$\psi^G(t) = (2\pi(i - \sigma)t^{m-1} + (m - 1)t^{m-2}) e^{-2\pi i \sigma t} e^{2\pi i i t}, \quad (2.4)$$

and in the frequency domain by,

$$\hat{\psi}^G(\omega) = \frac{i\omega(m - 1)!}{(\sigma + i(\omega - \sigma))^m}. \quad (2.5)$$

A precise work on this wavelet achieved by V. Lostalnen in [38] allows us to have an explicit formulation of the parameter σ such that the wavelet can be scaled while respecting the admissibility criteria:

$$\sigma^2 = \frac{r^{\frac{2}{m}}(1 - r^{\frac{2}{m}})m^2\xi^2}{2} \left(\sqrt{1 + \frac{B^2}{(1 - r^{\frac{2}{m}})^2 m^2 \xi^2}} - 1 \right),$$

where ξ is the center frequency and B is the bandwidth parameter. Notice that $B = (1 - 2^{-\frac{1}{Q}})\xi$ with $\xi = \frac{2\pi}{1+2^{\frac{1}{Q}}}$ induce a quasi orthogonal filter bank. The associated frequency center and standard deviation denoted respectively by $\omega_c^{\lambda_i}$ and $\Delta^{\lambda_i}\omega$, $\forall j \in \{0, \dots, JQ - 1\}$ are thus:

$$\omega_c^{\lambda_i} = \xi,$$

$$\Delta^{\lambda_i}\omega = B.$$

For this wavelet, thanks to the derivation in [38], we can manually select for each order m the center frequency and bandwidth of the mother wavelet, which ease the filter bank design. An important property that is directly related to the auditory response system is the asymmetric envelop, thereby the Gammatone wavelet is not invariant to time reversal to the contrary of the Morlet wavelet that behaves as a Gaussian function. Thus, for task such as sound classifications, this wavelet provides an efficient filter that

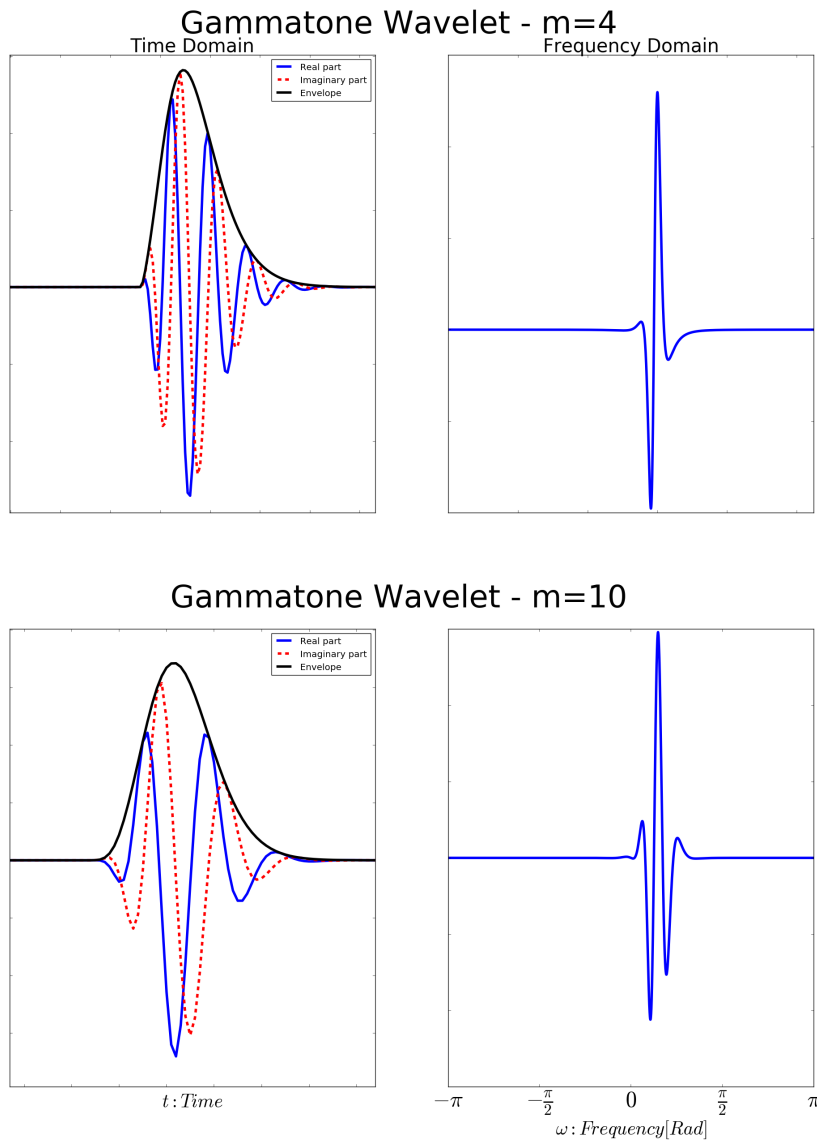


Figure 2.2 : On the upper (bottom) left a $m = 4$ ($m = 10$) Gammatone wavelet in the time domain where the dashed line is the imaginary part and the solid line is the real part, on the upper (bottom) right a $m = 4$ ($m = 10$) wavelet in the frequency domain.

will be prone to perceive the sound attack's. Beside this suitable property for specific analysis, this wavelet is near optimal with respect to the uncertainty principle. Notice that, when $m \rightarrow \infty$ it yields the Gabor wavelet [39]. Another interesting property of this wavelet is the causality, by taking into account only the previous and present information, there is no bias implied by some future information and thus it is suitable for real time signal analysis.

The Paul wavelet The Paul wavelet is a complex-valued wavelet which is highly localized in time, thereby has a poor frequency resolution. Because of its precision in the time domain, this wavelet is an ideal candidate to perform transient detection. The Paul wavelet of order m (Fig. 2.3) is defined in the time domain by,

$$\psi^P(t) = \frac{2^m i^m m!}{\sqrt{2m! \pi}} (1 - it)^{-(m+1)} \quad (2.6)$$

and in the frequency domain by,

$$\hat{\psi}^P(\omega) = \frac{2^m}{\sqrt{m(2m-1)!}} (\omega)^m e^{-\omega}, \forall \omega \in \mathbb{R}_+^*, \quad (2.7)$$

With associated frequency center and standard deviation denoted respectively by $\omega_c^{\lambda_j}$ and $\Delta^{\lambda_j \omega}$, $\forall j \in \{0, \dots, JQ - 1\}$ are:

$$\omega_c^{\lambda_j} = \frac{2m+1}{2\lambda_j},$$

$$\Delta^{\lambda_j \omega} = \frac{\sqrt{2m+1}}{2\lambda_j}.$$

In [13] they provide a clear and explicit formulation of some wavelet families applied the Paul wavelet in order to capture irregularly periodical variation in winds and sea surface temperatures over the tropical eastern Pacific Ocean . In addition, it directly represents the phase gradient from a single fringe pattern, yet providing a powerful tool in order to perform optical phase evaluation [40].

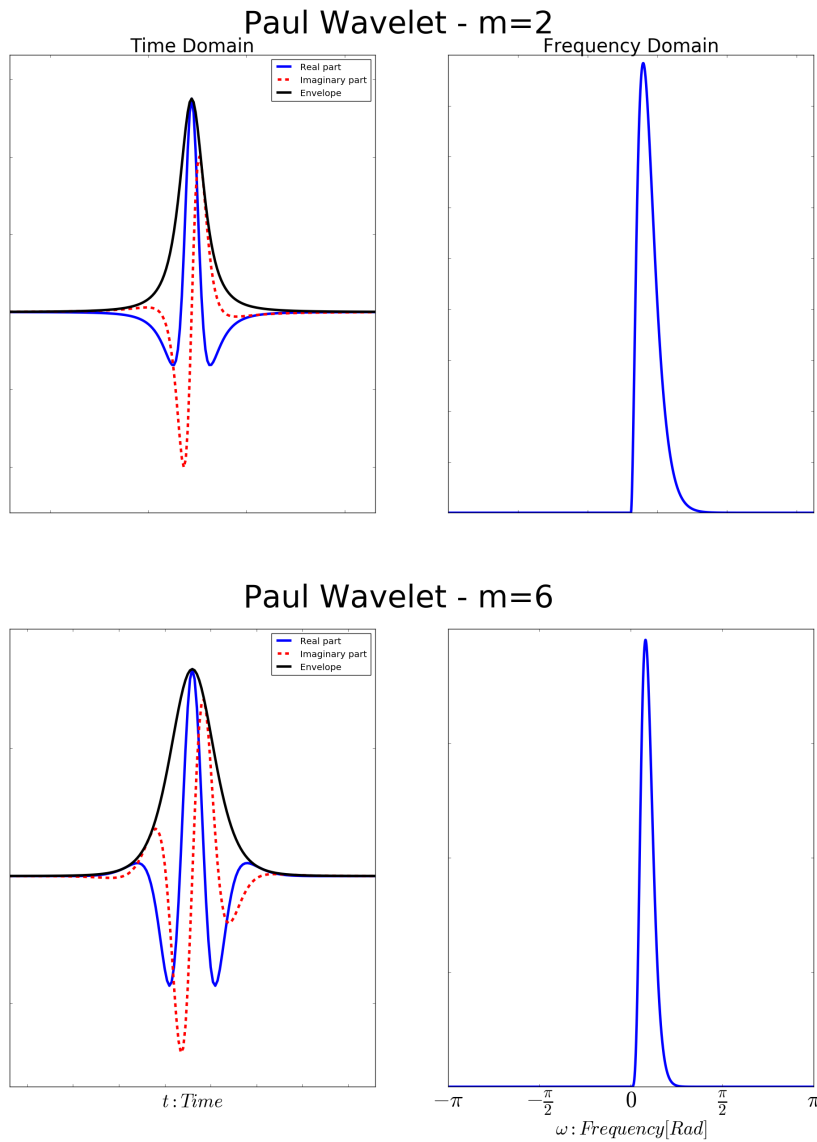


Figure 2.3 : On the upper (bottom) left a $m = 2$ ($m = 6$) Paul wavelet in the time domain where the dashed line is the imaginary part and the solid line is the real part, on the upper (bottom) right a $m = 2$ ($m = 6$) wavelet in the frequency domain.

2.2 Group Transform

In this section, we briefly describe the notion of group transform and its relation to the wavelet transform. For further details on the group theoretical aspects we presently

describe, the reader should refer to [41].

Definition 1. *A group is a set \mathcal{G} with a multiplicative operation \odot that respects enclosure, identity element, inverse element, and associativity.*

The representation of the group determines its action on a function space and bridges the gap between group theory and linear algebra, allowing to compute the transformation of a function following the rules induced by the specific group at hand. The representation of a group can be thought as a far-reaching generalization of the exponential function property, $\exp(x + y) = \exp(x) \exp(y), \forall x, y \in \mathbb{R}$ [42]. In fact, it is defined as,

Definition 2. *A linear continuous representation ρ of a group \mathcal{G} on the linear space \mathbb{H} is defined as*

$$\rho : \mathcal{G} \rightarrow GL(\mathbb{H}), \quad (2.8)$$

where $GL(\mathbb{H})$ is the the group of linear map in \mathbb{H} such that $\forall g, g' \in \mathcal{G}$

$$\rho(g \odot g') = \rho(g)\rho(g'). \quad (2.9)$$

For instance, let \mathbb{H} be a vector space such as \mathbb{R}^3 , the representation of the group is induced by 3×3 matrices. In this case, the operation on the right of (2.9) is a matrix multiplication, where each matrix depends on the group elements g and g' . This concept extends to linear operators acting on functional spaces.

As such, multiple transformations of a function by different elements of the group is equal to the representation of the combination of the group elements applied to the function.

This structure-preserving map defines the action of a group on elements of function spaces. Group transforms such as STFT and CWT can be expressed in such a way

by selecting a mother filter space and a group. The representation of the group in the mother filter space provides an operator that takes as input an element of the group and acts on the filter to transform it. A filter-bank can thus be created by iterating this process with different group elements. Therefore, the selected group carries the characteristics of the filter-bank and consequently, the group transform and its time-frequency tiling. Notice that further properties such as the invariant measure of the group and the resolution of the identity can be develop using the representation of the group.

As an example of group transform, we consider the creation of a wavelet filter-bank utilizing transformation group. Let's denote by \mathcal{G}_{aff} the affine group, the so called "ax + b" group, where the elements $(\lambda, \tau) \in \mathbb{R}_+^* \times \mathbb{R}$, where $\mathbb{R}_+^* = (0, +\infty)$, where the multiplicative operation of the group \odot is defined by

$$(\lambda, \tau) \odot (\lambda', \tau') = (\lambda\lambda', \tau + \lambda\tau') \quad (2.10)$$

Let's define by ρ_{aff} the representation of the affine group in $L^2(\mathbb{R})$, i.e., $\rho_{\text{aff}} : \mathcal{G}_{\text{aff}} \rightarrow GL(L^2(\mathbb{R}))$, such that ρ_{aff} is a homomorphism as per Definition 2. Its action on square integrable function ψ is defined as

$$[\rho_{\text{aff}}(g)\psi](t) = \frac{1}{\sqrt{\lambda}}\psi\left(\frac{t - \tau}{\lambda}\right), \quad t \in \mathbb{R}, \quad (2.11)$$

where (a, b) are respectively the dilation and translation parameters. The wavelet filter-bank is built by transforming a mother filter, ψ by the representation ρ_{aff} for specific elements of the group. A visualization of this approach for a Morlet wavelet filter can be seen in Figure (4.3). The wavelet transform of a signal $s_i \in L^2(\mathbb{R})$ is

achieved by

$$\mathcal{W}(s_i, \psi)(g_{(\lambda, \tau)}) = \langle s_i, \rho_{\text{aff}}(g_{(\lambda, \tau)})\bar{\psi} \rangle, \forall g_{(\lambda, \tau)} \in \mathcal{G}_{\text{aff}}, \quad (2.12)$$

$$= (s_i \star \rho_{\text{aff}}(g_{(\lambda, 0)})\psi), \forall g_{(\lambda, 0)} \in \mathcal{G}_{\text{aff}}, \quad (2.13)$$

where $\bar{\psi}(t) = \psi(-t)$, $\langle \cdot, \cdot \rangle$ denotes the inner product, \star the convolution, and $\rho_{\text{aff}}(g_{(\lambda, \tau)})\psi$ the action of the operator ρ_{aff} , evaluated at the group element $g_{(\lambda, \tau)}$, on the mother filter ψ as per (2.11). In practice, the filter-bank is generated by sampling a few elements of the group. For instance, in the case of the dyadic wavelet transform, the dilation parameters follow a geometric progression of common ratio equals to 2. In general, the translation parameter is sampled according to the scaling one [43]. Notice that in the convolution expression (2.13), the translation parameter $\tau = 0$, in fact the convolution operator \star acts as the translation one. In the case where the translation parameter depends on the scaling one, a specific stride is used to perform the discrete convolution.

Note that the STFT can be constructed similarly utilizing the Weyl-Heisenberg group [44], whose representation on $L^2(\mathbb{R})$ consists of frequency modulations and translations. More intricate group representations can be built as in [45] where the combination of the affine group and Weyl-Heisenberg group is considered.

2.3 Lie Group Transformations

The learnability of the group governing the data is crucial to our approach. We present here the major concepts that has been developed during the last decades regarding the approximation of Lie group. In [46–50], they propose methods capable of discovering the symmetry within the data alleviating the need for explicitly defining appropriate equivalence classes for the data. In fact, in a simple computer vision dataset such as

MNIST or in a music retrieval dataset such as GTZAN, there is more than translation and rotation to characterize efficiently the data [51]. This section is dedicated to the understanding of such approximation methods, which will be an important part of our work on manifold approximation with generalization guarantees.

The approximation of Lie groups has been introduced by [46] and later extended in [47, 52], and aims at learning the transformation operator underlying the data with the assumption that the dataset is the result of the action of a group on a sample. This framework has an essential place in neuroscience as there is evidence of an underlying network of neurons enabling the detection of a class of equivalence via transformation learning [49, 53, 54].

In the case of a Lie group, the dataset can be modeled according to the first-order Lie equation

$$\frac{d\mathbf{x}(\theta)}{d\theta} = G\mathbf{x}(\theta), \quad (2.14)$$

where $\mathbf{x}(\theta) \in \mathbb{R}^d$, θ is the coefficient governing the amount of transformation, and $G \in \mathbb{R}^{d \times d}$. This first-order differential equation indicates that the variation of the data is linear with respect to the data and depends on the infinitesimal operator $G \in \mathcal{T}_I\mathcal{G}$ where $\mathcal{T}_I\mathcal{G}$ denotes the Lie algebra of the group \mathcal{G} , i.e., the tangent of the group at the identity element. An introduction to group transformations can be found in [55]. The solution of Eq. 2.14 is given by $\mathbf{x}(\theta) = \exp(\theta G)\mathbf{x}(0)$.

One example of the orbit of a data with respect to a Lie group is the result of the rotation on an initial point $x(0) \in \mathbb{R}^2$, we have $x(\theta) = \exp(\theta G)x(0)$, $\theta \in \mathbb{R}$, $G = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$. In fact, where we recall that $\exp\left(\theta \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\right) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$. The infinitesimal operator G is thus encapsulating the group information. For more details regarding Lie group and the exponential map refer to [55].

While the learnability of the exponential map is tedious, one can exploit its Taylor series expansion to learn the infinitesimal operator. In fact, for a small ϵ we have

$$\mathbf{x}(\theta + \epsilon) \approx (I + \epsilon G)\mathbf{x}(\theta) \quad (2.15)$$

The operator G can thus be learned using data that are close to each other as they result from small transformations and thus follow this approximation. Without this form of supervision, the search for neighbor data is achieved by the nearest neighbor algorithm, as in [48]. Note that in our case, we will consider multiple transformations, each parametrized by a 1-dimensional Lie group, i.e. $\mathbf{x}(\theta) = \prod_{k=1}^h \exp(\theta_k G_k)\mathbf{x}(0)$, where $\theta \in \mathbb{R}^h$. In that case the first order approximation around the identity element of each group, as Eq. 2.15, becomes $\mathbf{x}(\theta + \epsilon) \approx (I + \sum_{k=1}^h \epsilon_k G_k)\mathbf{x}(\theta)$, where $\epsilon \in \mathbb{R}^h$ and with ϵ_k being the transformation parameter associated to infinitesimal operator G_k .

2.4 Thin Plate Spline Interpolation

While the formalism described above is appropriate for learning general Lie group, we thereby present an efficient parametrization of diffeomorphic-like transformations for images. Being capable of transforming images with respect to non-rigid transformation in an efficient way was crucial to perform image registrations. In our approach, we will leverage one of image registration favorite tool, the Thin Plate Spline Interpolation method,

Let's consider two set of landmarks, the source ones $\nu_s = \{u_i, v_i\}_{i=1}^{\ell}$ and the transformed $\nu_t = \{u'_i, v'_i\}_{i=1}^{\ell}$ where ℓ denotes the number of landmarks. The TPS aim at finding a mapping $F = (F_1, F_2)$, such that $F(u, v) = (F_1(u, v), F_2(u, v)) = (u', v')$, that is, the mapping between two set of landmarks. The particularity of the TPS is that it learns such a mapping by minimizing the interpolation term, and a regularization

that consists in penalizing the bending energy.

The TPS optimization problem is defined by

$$\min_F \sum_{i=1}^N \|(u'_i, v'_i) - F(u_i, v_i)\|^2 + \lambda \int \int \left[\left(\frac{\partial^2 F}{\partial u^2} \right)^2 + 2 \left(\frac{\partial^2 F}{\partial u \partial v} \right)^2 + \left(\frac{\partial^2 F}{\partial v^2} \right)^2 \right] dudv. \quad (2.16)$$

In our model, the source landmarks are considered to be the coordinates of a uniform grid. Also note that both the source landmarks and transformed ones are usually a subsets of the set of coordinate of the images. For instance, for the MNIST dataset of size 28×28 , the landmarks would be a grid of size $\ell \times \ell$, where $\ell < 28$. While the mapping is based on the landmark, it is then applied to the entire image coordinate. In fact, $F = (F_1, F_2)$ is mapping $\mathbb{R}^2 \rightarrow \mathbb{R}^2$, where F_1 (resp. F_2) corresponds to the mapping from (x, y) to the first dimension x' (resp. the second dimension y').

The solution of the TPS optimization problem, Eq. 2.16, provides the following analytical formula for F

$$F_1(u, v) = u' = a_1^{(1)} + a_u^{(1)}u + a_v^{(1)}v + \sum_{i=1}^{\ell} w_i^{(u)} U(|(u_i, v_i) - (u, v)|), \quad (2.17)$$

$$F_2(u, v) = v' = a_1^{(2)} + a_u^{(2)}u + a_v^{(2)}v + \sum_{i=1}^{\ell} w_i^{(v)} U(|(u_i, v_i) - (u, v)|), \quad (2.18)$$

where $|\cdot|$ is the L_1 -norm, a_1, a_u, a_v are the parameters governing the affine transformation, and w_i are parameters responsible for non-rigid transformations as they stand as a weight of the non-linear kernel U . The non-linear kernel U is expressed by $U(r) = r^2 \log(r^2), \forall r \in \mathbb{R}_+$.

Based on the landmarks ν_s and ν_t , we can obtain these parameters by solving a simple system of equation define by the following operations

$$\mathcal{L}^{-1}\mathcal{V} = \begin{bmatrix} (W^{(x)} | a_1^{(x)} a_x^{(x)} a_y^{(x)})^T \\ (W^{(x)} | a_1^{(y)} a_x^{(y)} a_y^{(y)})^T \end{bmatrix}. \quad (2.19)$$

where the matrix $\mathcal{L} \in \mathbb{R}^{(\ell+3) \times (\ell+3)}$, is defined as

$$\mathcal{L} = \left[\begin{array}{c|c} \mathcal{K} & \mathcal{P} \\ \hline \mathcal{P}^T & \mathcal{O} \end{array} \right], \mathcal{K} = \begin{bmatrix} 0 & U(r_{12}) & \dots & U(r_{1\ell}) \\ U(r_{21}) & 0 & \dots & U(r_{2\ell}) \\ \dots & \dots & \dots & \dots \\ U(r_{\ell 1}) & \dots & \dots & 0 \end{bmatrix}, \mathcal{P} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \dots & \dots & \dots \\ 1 & x_\ell & y_\ell \end{bmatrix}$$

where $r_{ij} = |(u_i, v_i) - (u_j, v_j)|$, $\mathcal{K} \in \mathbb{R}_+^{\ell \times \ell}$, and $\mathcal{V} = \begin{bmatrix} x'_1 & x'_2 & \dots & x'_\ell | 0 & 0 & 0 \\ y'_1 & y'_2 & \dots & y'_\ell | 0 & 0 & 0 \end{bmatrix}$.

Note that, since the matrix \mathcal{L} depends only on the source landmarks, and that in our case these are unchanged, its inverse can be computed only once. The only operation required to be computed for each data and each centroid is the matrix multiplication $\mathcal{L}^{-1}\mathcal{V}$ providing the parameters of the TPS transformation, as per Eq. 2.17, 2.18. Given these parameters, the mapping F can be applied to to each coordinate of the image.

Now in order to render the image, one can perform bilinear interpolation as it is achieved in . Besides, the bilinear interpolation will allow the propagation of the gradient through any differentiable loss function.

Given an image $x_1 \in \mathbb{R}^n$ where $n = W \times H$, W denotes the width and H the height of the image, and two sets of landmarks $\nu_s = \{u_i, v_i\}_{i=1}^\ell$, uniform grid coordinate of x_1 , and $\nu_t = \{u'_i, v'_i\}_{i=1}^\ell$, the transformation of the uniform grid, which are subset of the image coordinate, We are able to learn a mapping $F = (F_1, F_2)$ such that for each original pixel coordinate, we have their transformed coordinates. In fact, given any position (u, v) on the original image, the mapping F provides the new positions (u', v') as per Eq. 2.17, Eq. 2.18.

Now, from this transformed the coordinates space, we can render an image $x_2 \in \mathbb{R}^n$ using, as in [56], the bilinear interpolation function $\Gamma : \mathbb{R}^2 \times \mathbb{R}^n \rightarrow \mathbb{R}$ which takes as

input the original image x_1 and the transformed pixel coordinates (u', v') , and outputs the pixel value of the transformed image at a given pixel coordinate

$$\begin{aligned}
x_2(k, l) &= \Gamma[F(u_k, v_l), x_1] \\
&= \Gamma[(u'_k, v'_l), x_1] \\
&= \sum_{t, h \in \{0, 1\}} \sum_{i=1}^W \sum_{j=1}^H x_1(i, j) \delta(\lfloor u'_k + t \rfloor - i) \times \delta(\lfloor v'_l + h \rfloor - j) (u'_k - \lfloor u'_k \rfloor)^{\delta(t)} (v'_l - \lfloor v'_l \rfloor)^{\delta(h)} \\
&\quad \times (1 - (v'_l - \lfloor v'_l \rfloor))^{\delta(t-1)} (1 - (u'_k - \lfloor u'_k \rfloor))^{\delta(h-1)},
\end{aligned}$$

where δ is the Kronecker delta function and $\lfloor \cdot \rfloor$ is the floor function rounding the real coordinate to the closest pixel coordinate.

3

Learnable Wavelet Transform

Numerous learning tasks can be formed in a pattern recognition framework. Some of these applications are in speech, bioacoustic, and healthcare where the data have been exposed to different types of nuisances. For example, colored noises, multiple sources, measurements errors are a few to name. Recently, DNNs have provided an end-to-end learnable pipeline (from raw input data to the final prediction). In particular, convolutional-based DNNs are state-of-the-art in computer vision and other areas [1, 57, 58]. This approach reduces the need of designing hand-crafted features which involves an expert knowledge and a tedious search over the set of all possible features. Such paradigm shift opens the door to novel algorithms that encapsulate the learning of both, the features and the decision.

While providing a fully automated approach, DNNs' performances depend on the number of perturbations such as noise and inherent nuisances contained in the dataset. This is mainly due to the use of greedy optimization schemes applied on a very high-dimensional parametric model as well as the lack of explicit perturbation

modeling in DNNs [59]. This effect is amplified with the dimensionality of the input and the dimensionality of the filters.

Thus, it is particularly detrimental for time-series data, especially for bio-acoustic signals. In fact, those signals can be sampled at a high-frequency rate (up to 2,000 kHz). To add, the signals are recorded for long durations and exhibited non-stationary nuisances including sensor noise, background noise, and variant sources [60–62]. In addition, features of interest can lie at many different frequencies and in small time windows, adding complexity to the learning task.

Overall, current solutions to tackle bio-acoustic signals still rely on hand-crafted features providing representations that are input to the DNNs. Considered representations are often based on a time-frequency framework as they stretch and reveal crucial information embedded in the time-amplitude domain [63]. Moreover, decomposing signals in the time-frequency plane leverage the capability of Convolutional Neural Networks (CNNs). In fact, this feature is now considered as an image where CNNs are known to perform [2]. In addition, the design and selection of the filter enabling the time-frequency representation of the signal is directed by the prior knowledge on the feature of interest.

For instance, in the case of wavelet transform, one selects the most suitable wavelet family (i.e: Seismic data: Morlet wavelet, Speech: Gammatone wavelet [38,64]). Since the generalization capability of handcrafted features is only proportional to the amount of data witnessed by the designer. In [65,66], they developed algorithms that were able to automate the search for the optimal filter. However, these pre-processing techniques were derived for goals not necessarily aligned with the current tasks at hand (reconstruction, compression, classification) and thus do not provide a universal solution. In this work, we propose to alleviate the limitation of DNNs by proposing a

universal, learnable time-frequency representation that can be trained with respect to the application.

In particular, our solution learns the optimal time-frequency representation for the task and data at hand. This is done by learning time-frequency atoms with respect to the loss function (which can be of reconstruction, compression, anomaly detection, classification). The expression of these atoms corresponds to continuous filters analytically derived by spline functions. The filters can be constrained to inherit some pre-imposed properties such as smoothness and boundary conditions. Since the unique analytical expressions of the filters are differentiable with respect to their parameters, they can be optimized via first-order derivative methods such as gradient descent. As such, they can be cast in a DNN layer and learned by using backpropagation.

3.1 Outline of the Chapter and Contribution Summary

- We leverage hermite cubic spline interpolation to provide explicit expression of learnable continuous filters (Sec. 3.3).
- Derive the properties of the newly introduced learnable wavelet filter as to characterize the space and unicity of the basis the wavelets are built upon (Sec. 3.4).
- Allow to replace the filters of the first layer of Convolutional Neural Networks to provide a robust and interpretable representation of the signal showing state-of-the-art results on a bird detection task (Sec. 3.5).

3.2 Related Work

To provide flexible time-frequency representations and avoid the selection of hand-crafted filters, [17] proposed to learn the Mel-scale filters leading to Mel-Frequency Spectral Coefficients (MFSC). This approach concludes to learning the linear combination of the spectrogram frequency filters instead of using triangular windows. In this case, the underlying representation still relies on Fourier basis and thus inherits the problem of a pre-imposed basis. On the other hand, [18] proposed the use of a complex 1D convolutional layer followed by complex modulus and local averaging. This was motivated by stating that a Gabor scalogram followed by complex modulus and local averaging approximates MFSC coefficients [67]. Finally, with DNNs using the raw waveforms as input, [68–70] demonstrated that, with careful model design, one could reach results on parity with MFSC. Yet, the previously described work was applied onto datasets that are obtained from controlled experiments containing negligible noise and low-frequency sampling (leading to small length signals). As such, their results do not reflect the reliability and robustness of their methods for general real world-tasks.

3.3 Formalism

In this work, we propose to build continuous filters that can be extended to render time-frequency representation and specifically constant-Q transform [71]. This transformation renders the signal into a time-frequency plane where the frequency resolution decreases as the frequency increases. This transformation is directly related to the mapping performed by the human cochlea [72]. Our approach is general enough to produce any continuous filter as soon as a functional space to which they belong

exist. For sake of clarity, we will present the development of smooth locally supported oscillating filters, namely wavelet filters. As such, we provide the theoretical building blocks enabling one to build its own continuous filters depending on the application.

As we will show for the specific case of wavelet filters, our method is based on the definition of a functional space highlighting the properties of the wished filters. Given the latter, first, we will perform its discretization in the same manner as finite element methods for the variational problem of partial differential equations [73]. We build a discretization of the functional space such that as the number of knots grows, any continuous filter from the original functional space can be approximated arbitrarily closely. The filters are based on the linear combination of atoms that are basis elements of the discrete space, Hermite cubic splines in our case. It results in a filter that approximates a particular function in the infinite dimensional space. This filter, learned with respect to the data and the task, will describe a physical process underlying the signal while holding the properties of the functional space that it approximates. Thus, we create a framework enabling one to have theoretical guarantees based on the original functional space while being data and task driven.

Wavelets are square integrable localized wave functions [31]. Their ability to extract subtle patterns within non-stationary signals is inherited from their compact support [74]. In fact, wavelets are known to provide a robust time-frequency representation for non-stationary signals as it is localized both in time and frequency, and close to optimal from an uncertainty principle perspective with constant bandwidth to center frequency ratio [75]. In fact, the higher the frequency is, the higher the wavelet is precise in time (per contra, for low-frequency contents, wavelets are highly localized in frequency but wide in time). Besides, given the nature of the time-series data (e.g. non-stationary biological time-series), this embedding will encode the signal with only

a few activated wavelet atoms resulting in a sparse representation [76].

While we will leverage spline interpolation techniques to sample the filters from the functional space, our approach is independent of the spline wavelets setting. As a matter of fact, spline wavelets, well developed by [77] are constructed upon multiresolution analysis. These wavelets have an explicit expression in both the time and frequency domain hence facilitating their computation. Besides, they span a wide range of filter's smoothness order [77]. Despite the detachment between our framework and the one of spline wavelet, we can make an analogy between them. The ability of spline wavelets to provide an analytical formula for discrete wavelets is analogous to our proposal to provide the analytical continuous formula for the discrete filter-banks of convolutional networks.

In our case, we provide a theoretical framework enabling one to build through a data-driven process a continuous filter-bank spanning wavelet filters. Let define the space of wavelets be

$$\mathcal{V}_{L_c^2} = \left\{ \psi \in L_c^2(\mathbb{R}), \int \psi(t) dt = 0 \right\}, \quad (3.1)$$

where $L_c^2(\mathbb{R})$ defines the space of square integrable functions with compact support.

We direct the reader to a complete review of spline operators in [78]. In order to control the smoothness of the wavelets and thus of the sampled filters, we propose to restrict our study to the space of zero-mean functions with compact support belonging to $C_c^n(\mathbb{R})$

$$\mathcal{V}_{C_c^n} = \left\{ \psi \in C_c^n(\mathbb{R}), \int \psi(t) dt = 0 \right\}. \quad (3.2)$$

Since continuous and differentiable functions with compact support are square integrable, and a fortiori they belong to L_c^∞ , it is clear that $\mathcal{V}_{C_c^n} \subset \mathcal{V}_{L_c^2}$. Therefore, $\mathcal{V}_{C_c^n}$ is a space of function with compact support where the smoothness is described by the

order n . In this work, we will restrain our study to the space $\mathcal{V}_{C_c^1}$ which will provide an efficient trade-off between smoothness characterization and tractability. In order to build the discrete space denoted by V , we first proceed with the partition of the support of the function, denoted by the segment $[a, b]$, in $N + 1$ intervals of length $h = \frac{b-a}{N+1}$, we thus defined as $t_i = a + ih, \forall i \in \{0, \dots, N + 1\}$ the $N + 2$ points on the mesh, where in particular $t_0 = a$ and $t_{N+1} = b$. We define the discretization of the functional space $\mathcal{V}_{C_c^1}$ as

$$V = \left\{ \psi_h \in \bar{V}, \int \psi_h(t) dt = 0 \right\}, \quad (3.3)$$

where

$$\bar{V} = \left\{ \psi_h \in S_{C_c^1}, \psi_h(a) = \psi_h(b) = \frac{d\psi_h}{dt} = \frac{d\psi_h}{dt} = 0 \right\}, \quad (3.4)$$

and

$$S_{C_c^1} = \left\{ \psi_h \in C_c^1([a, b]), \psi_h|_{[t_i, t_{i+1}]} \in \mathcal{P}^3, i = 1, \dots, N \right\}, \quad (3.5)$$

where \mathcal{P}^3 defines the space of order 3 polynomials and $S_{C_c^1}$ the space of cubic and smooth splines.

Then, $\forall \psi_h \in \bar{V}$, we have

$$\psi_h = \sum_{i=1}^N \theta_{t_i} u^{(i)} + \sum_{i=1}^N \theta'_{t_i} v^{(i)}. \quad (3.6)$$

One can easily explicitly derived this basis via the following reference functions

$$u_0(t) = (1 + 2t)(1 - t)^2, u_1(t) = (2 - 2t)t^2, \quad (3.7)$$

$$v_0(t) = t(1 - t)^2, v_1(t) = -(1 - t)t^2, \quad (3.8)$$

then $\forall i \in \{1, \dots, N\}$ we have the following functions defined on their supports

$$u^{(i)}(t) = u_0\left(\frac{t - t_{i-1}}{h}\right), \forall t \in [t_{i-1}, t_i] \quad (3.9)$$

$$= u_1\left(\frac{t - t_i}{h}\right), \forall t \in [t_i, t_{i+1}], \quad (3.10)$$

and

$$v^{(i)}(t) = v_0\left(\frac{t - t_{i-1}}{h}\right)h, \quad \forall t \in [t_{i-1}, t_i] \quad (3.11)$$

$$= v_1\left(\frac{t - t_i}{h}\right)h, \quad \forall t \in [t_i, t_{i+1}]. \quad (3.12)$$

Finally, from \bar{V} to V , we require that the integral of the polynomial is null over the whole domain.

Note that the error of the approximation involved by the discretization of the space by mean of cubic Hermite splines is of the order $\mathcal{O}(h^4)$ [79]. As a matter of fact, the smaller the segment of the mesh is, the closer the approximant will be to the associated function in the functional space.

3.4 Properties

i Existence & Uniqueness

Lemma 1. *Any function in $S_{C_c^1}$ is entirely and uniquely defined by its values and its first order derivative values on each point of the mesh $t_i, \forall i \in \{0, \dots, N + 1\}$.*

Proof. Let $\psi_h \in S_{C_c^1}$, without loss of generality we focus on $\psi_{h_{[t_i, t_{i+1}]}}$. It is clear that given the fact that it is a polynomial of degree 3 on the interval $[t_i, t_{i+1}]$ it can be expressed as

$$\psi_{h_{[t_i, t_{i+1}]}} = a(t - t_i)^3 + b(t - t_i)^2 + c(t - t_i) + d. \quad (3.13)$$

Let show that the coefficients a, b, c, d of the polynom are uniquely determined by $\theta_{t_i}, \theta_{t_{i+1}}, \theta'_{t_i}, \theta'_{t_{i+1}}$. Naturally, $d = \theta_{t_i}$ and $\theta'_{t_i} = c$, then, the coefficient a, b are defined by the solution of the following problem

$$\begin{pmatrix} h^3 & h^2 \\ 3h^2 & 2h \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \theta_{t_{i+1}} - \theta'_{t_i}h - \theta_{t_i} \\ \theta'_{t_{i+1}} - \theta'_{t_i} \end{pmatrix}, \quad (3.14)$$

since $\det \begin{pmatrix} h^3 & h^2 \\ 3h^2 & 2h \end{pmatrix} = -h^4$, the system has a unique solution. \square

Theorem 1. *Let define $u^{(i)}$ and $v^{(i)}$ as functions belonging to $S_{C_c^1}$ such as $\forall i \in \{0, \dots, N+1\}$*

$$u^{(i)}(t_j) = \delta_{ij}, \quad u^{(i)'}(t_j) = 0, \quad (3.15)$$

$$v^{(i)}(t_j) = 0, \quad v^{(i)'}(t_j) = \delta_{ij}. \quad (3.16)$$

These functions form a basis of $S_{C_c^1}$, and for all $\psi_h \in S_{C_c^1}$, we have,

$$\psi_h = \sum_{i=0}^{N+1} (\theta_{t_i} u^{(i)} + \theta'_{t_i} v^{(i)}). \quad (3.17)$$

Proof. We first show that the space $S_{C_c^1}$ is spanned by such functions. Let ψ_h any function belonging to $S_{C_c^1}$, et let z defined such as

$$z = \sum_{i=0}^{N+1} (\theta_{t_i} u^{(i)} + \theta'_{t_i} v^{(i)}), \quad (3.18)$$

it is clear that z belongs to $S_{C_c^1}$ as a linear combination of functions belonging to $S_{C_c^1}$. Then, for all $j \in \{0, \dots, N+1\}$, we have $z(t_j) = \theta_{t_j}$ and $\frac{dz}{dt} = \theta'_{t_j}$. Thus z coincides with the function ψ_h on all the points of the mesh. From Lemma 1, we know that $z = \psi_h$, thus $u^{(i)}$ and $v^{(i)}$ span the space $S_{C_c^1}$. Let's now prove that this family is linearly independent. Let's assume $\psi_h = \sum_{i=0}^{N+1} (\lambda_i u^{(i)} + \mu_i v^{(i)}) = 0$, where λ_i, μ_i are scalar coefficients. Then, for all $j \in \{0, \dots, N+1\}$ we have $\theta_{t_j} = \lambda_j = 0$ and $\theta'_{t_j} = \mu_j = 0$. \square

Notice that the parameters $\theta_{t_i}, \theta'_{t_i}$, correspond respectively to the value of the function ψ_h and the derivative of the function ψ_h at the knot t_i .

ii Space Dimensions

Corollary 1. *The dimension of the space $S_{C_c^1}$ is $2(N+2)$.*

The proof is immediate given that its basis forms a $2(N + 2)$ functions as defined in the previous theorem. We have built a basis for the space $S_{C_c^1}$, it is simple to analyze the basis of its subspaces, namely \bar{V} and V , where we have $V \subset \bar{V} \subset S_{C_c^1}$. From the space $S_{C_c^1}$ to \bar{V} we add Dirichlet and Neumann boundary conditions. These conditions imply directly that any function in \bar{V} is $C^1(\mathbb{R})$ as the function in $S_{C_c^1}$ has a compact support, it is null out of its support, then imposing that both the derivative and the value on the boundary of the support is zeros implies the continuity and differentiability on \mathbb{R} .

Corollary 2. *The dimension of the space \bar{V} is $2N$.*

Proof. Imposing the boundaries conditions remove 4 degrees of freedom from the space $S_{C_c^1}$ as we only consider the internal part of the mesh. \square

Corollary 3.

$$V = \left\{ \psi_h \in \bar{V}, \exists j, \theta_{t_j} = - \sum_{i \neq j} \theta_{t_i} \right\}, \quad (3.19)$$

and the dimension of V is $2N - 1$.

Proof. While integrating $\psi_h \in \bar{V}$ and using Chasles' relation to split the integral over the mesh's segments, the C^1 property implies that the coefficients θ'_{t_i} cancel each other. Then the equality of the integral to zeros is equivalent to the condition following condition $\exists j \in \{1, \dots, N\}, \theta_{t_j} = - \sum_{i \neq j} \theta_{t_i}$, which proves the first part of the corollary. The dimension of the space is the dimension of \bar{V} minus one degree of freedom, which completes the proof. \square

iii Filter-bank Derivation Another advantage of analytical filters resides in the possibility to apply standard continuous operators such as time-dilation and frequency-shift. Applying such operators to the primitive filter yields the creation of

the filter-bank. From Lemma 1, it is clear that the set of parameters $\theta = \{(\theta_{t_i}, \theta'_{t_i}), \forall i \in \{1, \dots, N\}\}$ defines uniquely the spline filter. We now denote our discretized filter ψ_h by ψ_θ . For our experiments, we will consider the use of our filter formulation to derive a filter-bank. This is done by only learning a mother filter which is then dilated to build the collection of filters. Hence they all rely upon the same analytical form but are dilated versions of each other. Let's suppose we have a mother wavelet, $\psi_\theta \in \mathcal{V}_{L^2}$, we propose an operation, a dilation, that will provide the analytic expression of our redundant frame.

Let D_λ , a dilation operator defined by

$$\mathcal{D}_\lambda[\psi_\theta](t) := \frac{1}{\sqrt{\lambda}}\psi_\theta\left(\frac{t}{\lambda}\right). \quad (3.20)$$

The scale parameter $\lambda \in \mathbb{R}^+$ allows for time dilation and frequency-shift and follows a geometric progression for the case of wavelets. It is defined as $\lambda_i = 2^{\frac{i-1}{Q}}, i = 1, \dots, JQ$ where $J \in \mathbb{N}, Q \in \mathbb{N}$ define respectively the number of octave and the number of wavelets per octave. Taking $Q > 1$ yields a redundant frame, which can be more powerful for representation analysis [80]. We now denote this collection of scales as $\Lambda := \{\lambda_i, i = 1, \dots, JQ\}$. Note that, in this work, this parameter will not be learned but will be specified given a priori knowledge on the data.

3.5 Experiments

i Task In order to validate the proposed method in a supervised task, we provide experiments on a large scale bird detection application. The data set is composed of 7,000 field recording signals of 10 sec. sampled at 44 kHz from the Freesound [81] audio archive representing slightly less than 20 hours of audio signals. The audio waveforms are extracted from diverse scenes such as city, nature, train, voice, water,

etc., some of which include bird songs. In this paper, we will focus on the supervised bird detection task consisting of assigning the label 1 if the sound contains a bird song and 0 otherwise. The labels regarding the bird detection task can be found in freefield1010*. Due to the unbalanced distribution of the classes (3 for 1), the metric to evaluate these methods is the Area Under Curve (AUC) applied on a test set consisting of 33% of the overall dataset.

ii Filters Implementation In order to implement such filters, we leverage the Hermite cubic spline interpolation formula (3.6) between each of the knots of a specified domain to obtain the sampled filter’s chunk per region (between two knots). This takes the following form for a set of given filters

$$\begin{aligned} \psi_i(t) = & (2t^3 - 3t^2 + 1)\theta_{t_i} + (t^3 - 2t^2 + t)\theta'_{t_i} \\ & + (-2t^3 + 3t^2)\theta_{t_{i+1}} + (t^3 - t^2)\theta'_{t_{i+1}} \end{aligned} \quad (3.21)$$

$$\psi_\theta(t) = \sum_{i=0}^N \psi_i \left(\frac{t - t_i}{t_{i+1} - t_i} \right) 1_{\{t \in [t_i, t_{i+1}]\}}. \quad (3.22)$$

Then, one derives the filter bank by using the above equation with different time sampling according to the dilation from Λ . For each scale λ_i the time sample is refined as $t = \{t_0, t_0 + \frac{h}{\lambda_i}, \dots, t_N\}$. This process can be done independently for the calculation of the real and imaginary coefficients. For the time-dilation operation, it suffices to repeat this process with a finer or larger sampling grid where the Hermite cubic spline interpolation occurs.

We provide in this section a step-by-step construction of the proposed spline filters. First, in Fig. 3.1 we show the Hermite cubic spline that will be used as building blocks our filters. As can be seen, it is a cubic polynomial defined on a closed interval. Its

*<http://machine-listening.eecs.qmul.ac.uk/bird-audio-detection-challenge/>

parameters are uniquely defined by specifying the values of the polynomial at the boundaries as well as the values of the derivative of the polynomial at the boundaries. Then, in Fig. 3.2 we demonstrate how one leverages multiple Hermite cubic splines

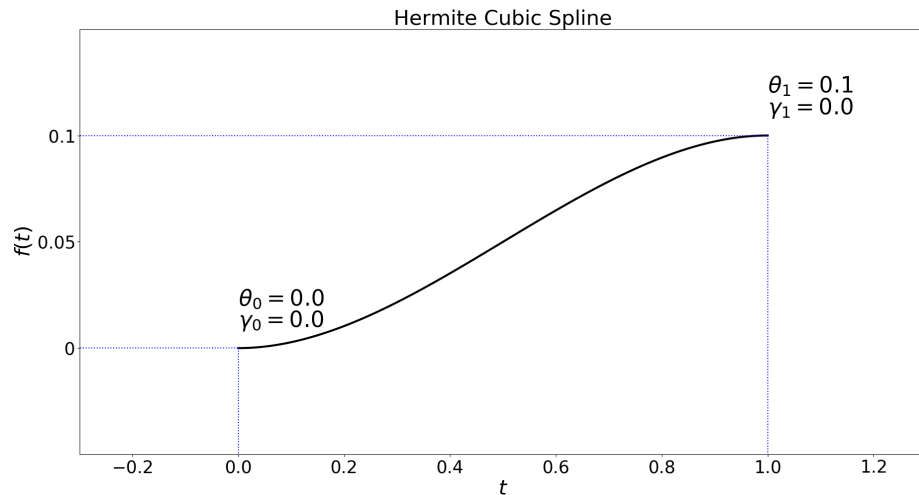


Figure 3.1 : Hermite cubic spline: cubic polynomial on a close interval

to construct the spline filters. The first step is to concatenate the Hermite cubic splines on a uniform partition of a closed interval. Each region leverages a Hermite cubic spline and we denote as spline filter the piecewise Hermite cubic spline function. In order to enforce the spline filter to be in the space of the considered filter (here wavelets), one has to impose continuity and smoothness by constraining the values that the Hermite cubic splines of each region, Fig. 3.3. In fact, by specifying that neighboring Hermite cubic splines have the same values at the shared boundary we reach smoothness. In addition, we require a localized and centered spline filter. This is imposed by constraining the values of the Hermite cubic splines as demonstrated in Fig. 3.4.

With the derived mother filter, it is now possible to sample the filter-bank that can

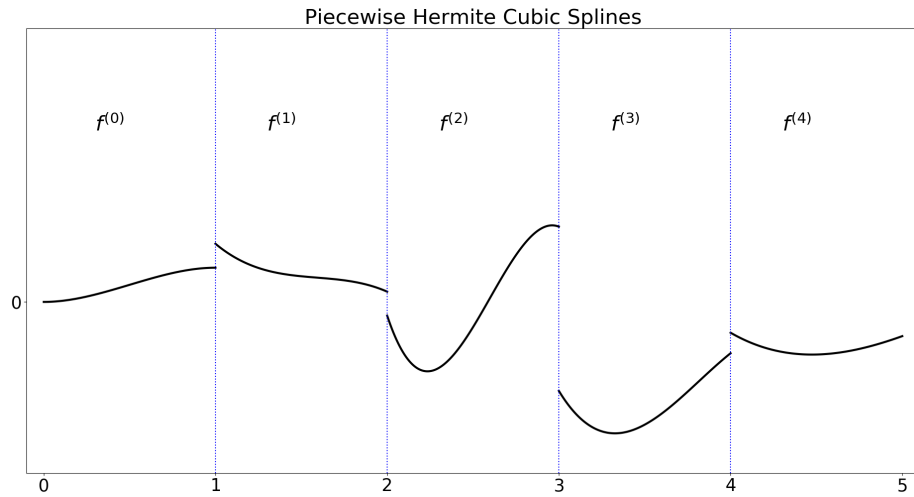


Figure 3.2 : Concatenation of Hermite cubic splines

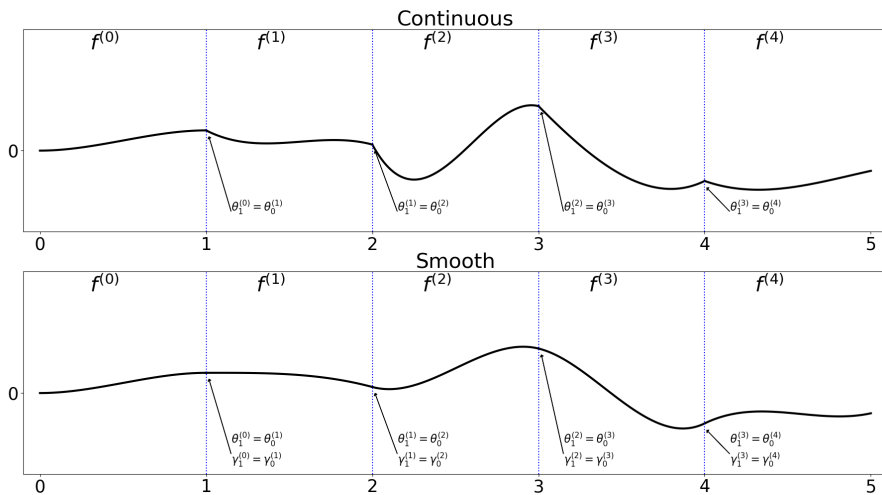


Figure 3.3 : Insuring continuity and smoothness of the filter

be used in place of standard filter of a convolutional layer of a deep network. To do so, the analytical expression is simply evaluated over a uniform sampling grid. Each grid will sample a filter and the filter-bank is sampled with different grids, each with different number of points. The more points in the grid the more dilated will be the filters.

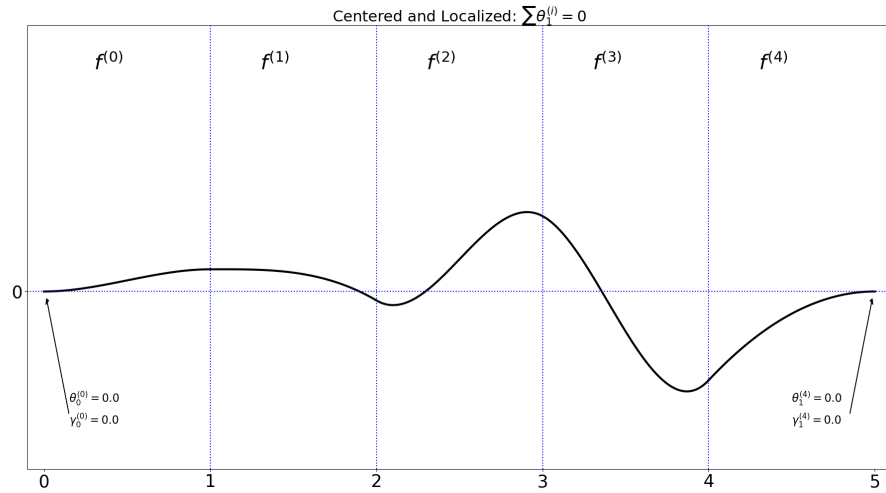


Figure 3.4 : From the smooth filter to wavelet

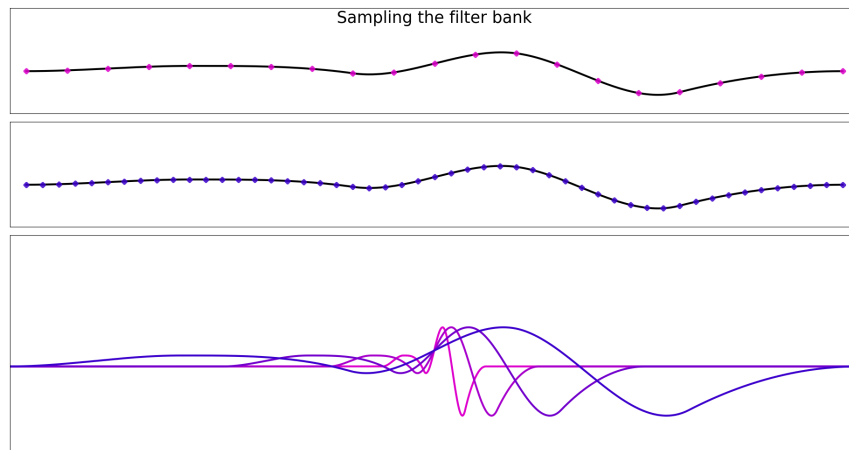


Figure 3.5 : From the mother wavelet to the filter-bank

iii Architecture Comparison To compare our method we propose different training settings. For all the trained methods, the signals are subsampled by 2, leading to a sampling rate of ≈ 22 kHz. The learning was set for 120 epochs with the batch size being 10 samples. The learning rate for each method has been cross-validated with respect to the following learning rate grid: $[0.0001, 0.005, 0.01, 0.05]$. We did not

perform data augmentation. We provide average and standard deviation for the AUC evaluation score over 10 independent runs.

For each run, all the topologies are trained and tested on the same training and testing set leading to a comparison of the different algorithms using the same data. The different methods we will apply correspond to variants of the state-of-the-art method proposed in [82]. The difference will lie in the first layer of the topology which corresponds to either an MFSC transform, an unconstrained complex $1D$ convolutional layer and finally the complex spline filters cast into the complex $1D$ convolutional layer. For all cases, the number and sizes of the filters are identical. Everything else in the DNN is kept identical between the methods. Also, both the Spline convolutional layer and the convolutional layer were tested with two filter initialization settings: random and Gabor.

Finally, due to the induced extra representation to store on GPU (namely $\mathcal{W}_{\psi_{\theta}}[x](\lambda, t)$) prior applying the mean-pooling, the required memory for the Spline convolutional and convolutional topologies is higher than the baseline which computes the MFSC on CPU a priori. As a result, the mean-pooling applied to these cases is chosen twice bigger for those topologies as opposed to the MFSC baseline, leading to a first layer representation twice smaller. We briefly describe the different methods and choice of parameters.

State-of-the-art method MFSC + ConvNet: The baseline and state-of-the-art method [82] is based on MFSC: spectrogram with window size of 1024 and 30% overlap, then mapped to the mel-scale by mean of 80 triangular filters from 50 Hz to 11 kHz. The MFSC are computed by applying a logarithm. This time-frequency representation is then fed to the following network: Conv2D. layer (16 filters 3×3), Pooling (3×3), Conv2D. layer (16 filters 3×3), Pooling (3×3), Conv2D. layer (16

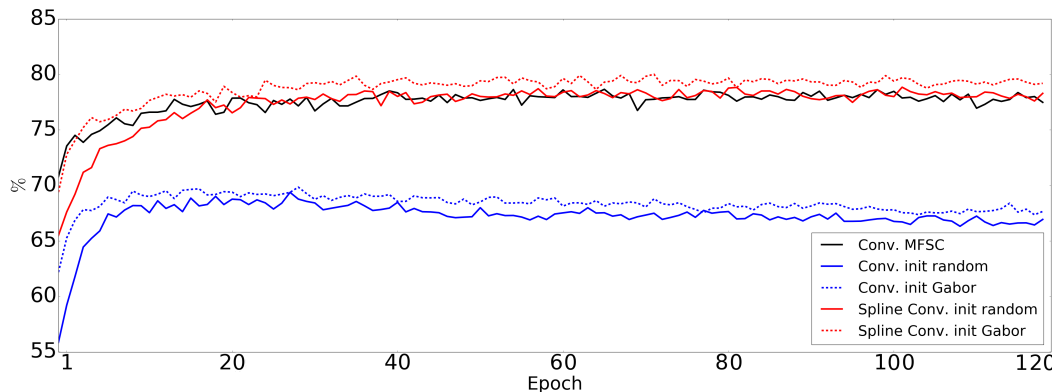


Figure 3.6 : **Training AUC on FreeField dataset** - Initializing the CNN filters with a Gabor filter-bank leads to increased performances as opposed to random initialization. Yet, the final performances remain around 10 percentage point below the other methods. The spline-based convolutional layer with random initialization is able to reach similar performances with the MFSC features after only 20 epochs. Finally, the Gabor initialized spline filter-bank starts on par with the MFSC features as can be seen for the first couple of epochs and is then able to overcome the MFSC feature to rapidly obtain about 2 point of percentage increased performances. Hence we can see the MFSC representation to be a satisfactory initializer yet not optimal.

filters 3×1), Pooling (3×1), Conv2D. layer (16 filters 3×1), Pooling (3×1), Dense layer (256), Dense layer (32), Dense layer (1 sigmoid). At each layer a leaky ReLU is applied following a batch-normalization. For the last three layers a 50% dropout is applied.

ConvNet: In this method, we keep the architecture of state-of-the-art solution, while replacing the deterministic MFSC by a regular complex convolutional layer, followed by a complex modulus, a logarithm operation, and an average pooling, providing as stated in [18] a learnable MFSC representation. The number of complex filters for the first layer is 80 leading to a representation at the first layer equivalent to the MFSC. We propose two initialization settings for the first layer of discrete filters: random and Gabor. The complex convolution is simply implemented as a two channel convolution corresponding to the real and imaginary part.

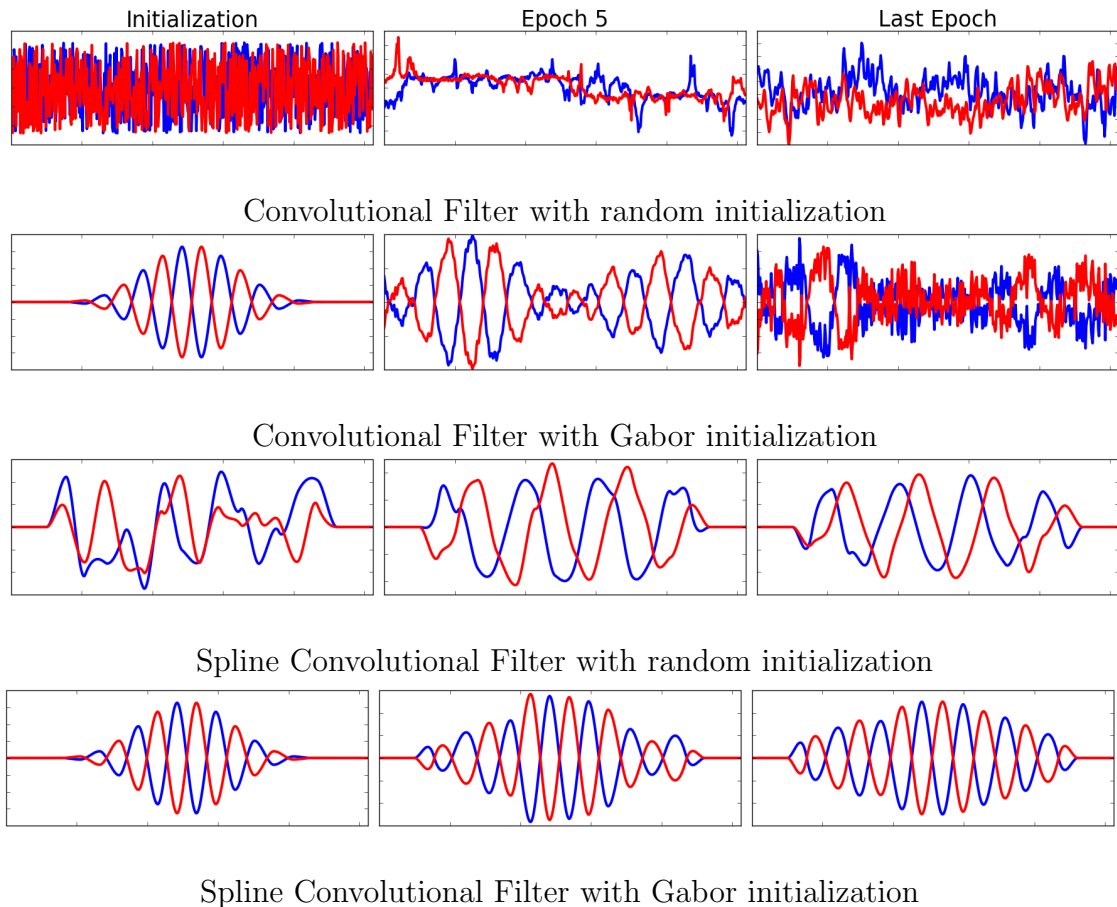


Figure 3.7 : **Filters extracted from the convolutional Layer and spline convolutional layer.** The red and blue lines correspond to the complex and real part respectively. Filters are presented in the left, middle, and right column respectively corresponding to the initialization, during learning, and after learning. As can be witnessed in the third row, even with random initialization, the smoothness and boundary conditions are able to prevent too erratic filters. Our Spline configuration initialized Gabor (the bottom row) through learning tends to a modified Gabor. In fact, while a Gabor is roughly a complex sine localized via a Gaussian window, the learned filter seems closer to a complex sine localized with a Welch window [83]. For the discrete convolutional filters, even with Gabor initialization (second row), the nuisances (noise, and other nonstationary class independent perturbations) are absorbed during learning even at early stages (middle column).

Spline Continuous Filter ConvNet: As for the Conv. Net model, we keep the same architecture but replace the first layer with the proposed method. In particular,

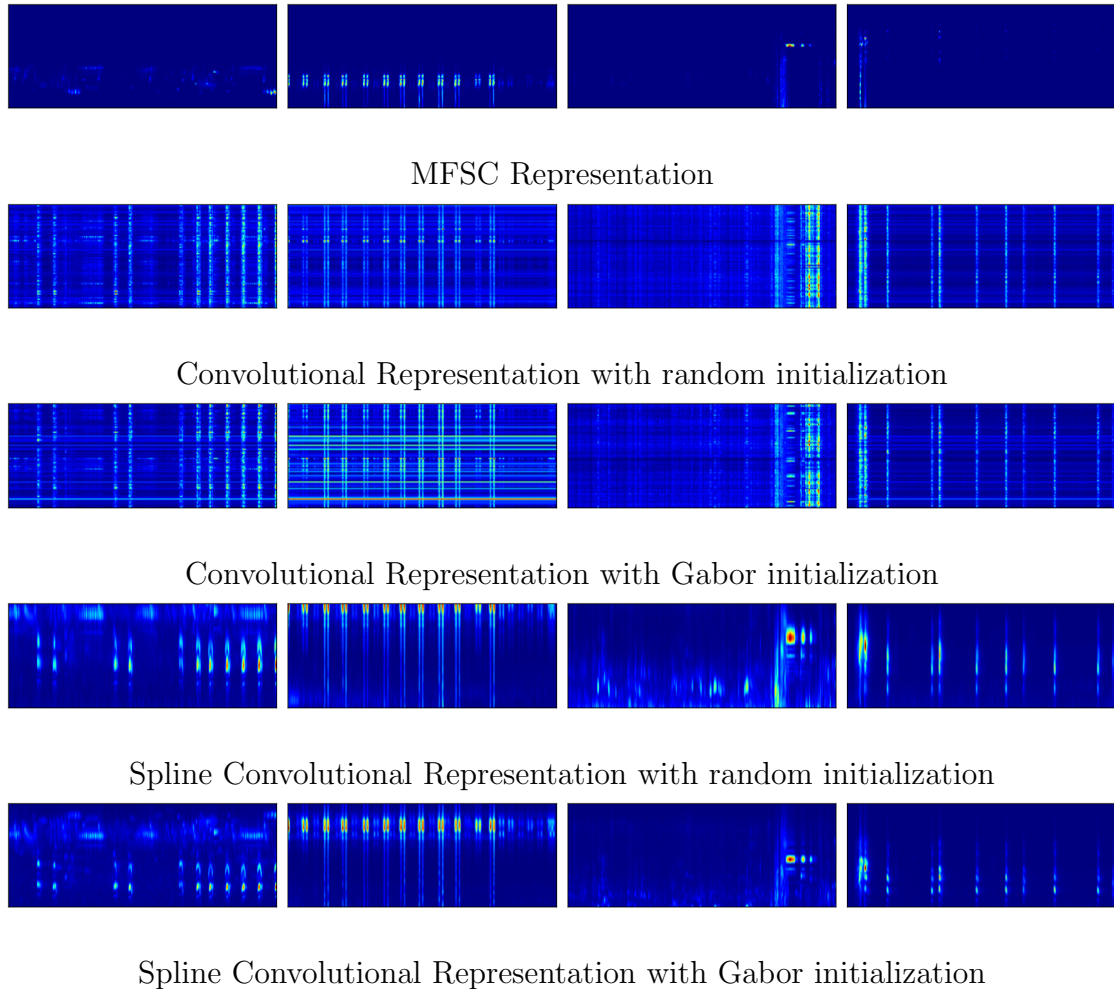


Figure 3.8 : **First layer representations** (*time-frequency plane*). Signals of class 1 (a bird is present). Each column depicts a different signal. Firstly, the amount of sparsity (the $L1$ -norm of the representation) often considered as a quality criterion can be seen to be conserved with the spline convolutional. In addition, the events are well localized in frequency as opposed to the convolutional representations depicting events covering the whole axis and/or time dimension. The detected events seem to be in accordance with all representations.

the first layer is a complex convolutional Layer with filters computed from our method. Given the dataset context, we naturally impose the functional space for the filters as the wavelet space. We use 80 filters based on the dilation operator developed in Eq. 3.20 with $J = 5$, $Q = 16$. This layer is followed by a complex modulus, a

logarithm operation, and an average pooling. We propose two initializations as for the previous method: random, and Gabor. For each filter, the number of cubic Hermite polynomials respective the boundary condition is 15 as 16 knots are used. Since the set of filters are derived by the dilation of one mother filter, the number of parameters for this layer is 56 (14×4).

iv Complexity & Parameters The number of parameters for the spline convolutional DNN is of 145,073. The computation time for one batch of 10 examples is 0.44 ± 0.009 sec. For the convolutional DNN, the number of parameters is 227,089 and the computation time for one batch is 0.42 ± 0.01 sec. In fact, given our current implementation, the Spline convolutional layer first has to interpolate and generate the filter-bank based on the parameters of the Hermite cubic spline and this filter-bank (for real and complex parts) is then used in a convolutional layer.

This extra computation time of interpolation and filter-bank derivation thus takes an additional 0.02 sec. per batch on average. Finally, for the state-of-the-art method, the number of parameters is 374,385 and the computation time for one batch is 0.01 ± 0.0004 sec. This comes from the input being directly the MFSC representation as opposed to the raw waveform. The increased number of degrees of freedom comes from having a time-frequency representation longer in time as opposed to the other two topologies having larger time-pooling for memory constraints.

v Results Table 5.1 displays the average over the last 20 epochs of the 10 runs for each method as shown in 3.6. We see that using classical discrete filters on the raw waveforms fail to generalize and is seen to overfit starting at epoch 50. However, performing MFSC representation drastically increases the accuracy. Finally, we see that our approach is capable of performing equivalent results than the state-of-the-art

Table 3.1 : Classification Results - Bird Detection - Area Under Curve metric (AUC)

<i>Model</i> (learning rate)	<i>AUC</i> (mean \pm std)
Conv. MFSC (0.01)	77.83 \pm 1.34
Conv. init. random (0.01)	66.77 \pm 1.04
Conv. init. Gabor (0.01)	67.67 \pm 0.98
Spline Conv. init. random (0.005)	78.17 \pm 1.48
Spline Conv. init. Gabor (0.01)	79.32 \pm 1.53

in the case of random initialization and increases score by nearly 2 points when initialized with Gabor filters.

3.6 Conclusion

In this work, we proposed a novel way to tackle end-to-end architecture for waveform analysis. As a matter of fact, while current work mainly focus on the architecture, we proposed to highlight the need of designing new learnable filters that can be used with any differentiable loss function and architecture. This approach showed its potential and robustness on a challenging audio scene dataset reaching new state-of-the-art results. One challenging issue that remains for both classical discrete filters as well as our filters is their size when one needs to capture low frequency components. As a matter of fact, in our case we reduced the number of degrees of freedom compared to classical filters, however the size of the filters remain the same leading to the same computational power needs. The problem of filter design becomes a problem of functional space design. While we exploited the finite impulse response filter type, the reduction of the dimension of filters capable of capturing low frequency component

can be achieved via infinite impulse response filters. Another point to be tackled is the learning of the dilation operator leading to the filter bank. As a matter of fact, as we have shown in Sec. 3.4 iii this operator can be written

$$\mathcal{D}_\lambda[\psi](t) := \frac{1}{\sqrt{\lambda}}\psi_\theta\left(\frac{t}{\lambda}\right),$$

where instead of constraining the parameter λ to follow a geometric progression we can learn it as it is a differentiable parameter. It would lead to a redundant dictionary that would be dilated according to the data and the task at hand, thus not capturing certain frequency bands that can be disregarded with respect to the task at hand. Finally, other operators can be introduced and learned. For instance, the frequency-chirp operator proposed by [84] and defined as,

$$\mathcal{C}_c[\psi_\theta](t) = \exp\left\{i2\pi\frac{c}{2}t^2\right\}\psi_\theta(t), c \in \mathbb{R},$$

allows to introduce variation, through time, of the frequency content. Thus leading to even more flexible filtering operations.

4

Learnable Group Transform

To this day, the front-end processing of time-series remains a keystone toward the improvement of a wealth of applications such as health-care [85], environmental sound [86, 87], and seismic data analysis [88]. The common denominator of the recorded signals in these fields is their undulatory behavior. While these signals share this common behavior, two significant factors imply the need of learning the representation: *(i)* time-series are intrinsically different because of their physical nature, *(ii)* the machine learning task can be different even within the same type of data. Therefore, the representation should be induced by both the signal and the task at hand.

A common approach to performing inference on time-series consists of building a Deep Neural Network (DNN) that operates on a spectral decomposition of the time-series such as wavelet transform (WT) or Mel Frequency Spectral Coefficients (MFSC). These decompositions represent the signal. While the use of these decompositions is extensive, we show in Section 4.2 their inherent biases and motivate the development of

a generalized framework. The selection of the judicious transform is either performed by an expert on the signal at hand, or by considering filter selection methods [89–91]. However, an inherent drawback is that the selection of the filters decomposing the signals is often achieved with criteria that do not align with the task. For instance, a selection based on the sparsity of the representation while the task is the classification of the signals. Besides, these selection methods and transformations require substantial cross-validations of a large number of hyperparameters such as mother filter family, number of octaves, number of wavelets per octave, size of the window [76, 92].

In this work, we alleviate these drawbacks by proposing a simple and efficient approach by considering the generalization of these spectral decompositions. They consist of taking the inner product between filters and the signals. From one decomposition to the other, only the filter bank differs. The filters of well-known spectral decompositions, such as the short-time Fourier transform (STFT) and the continuous wavelet transform (CWT) are built following a particular scheme. Each filter is the result of the action of a transformation map on a selected mother filter, e.g., a Gabor filter. If the transformation map is induced by a Group, the representation is called a Group Transform (GT), and both the group with the mother filter characterize the decomposition.

4.1 Outline of the Chapter and Contribution Summary

- We generalize common group transforms that are used to decompose signals in a multi-scale fashion replacing the commonly used affine group by a subgroup of the group of diffeomorphisms. (Sec. 4.3).
- Draw the connection between filters that can be learned by our framework and commonly observed waveform in biological time-series (Ref. 4.4 - i).

- Provide the equivariance properties of the representation and how it differs from the equivariance induced by the utilization of the affine group (Ref. 4.4 - vii).
- We propose an efficient way to learn the appropriate elements of the group of diffeomorphisms (Sec. 4.5 - vii).
- We show results competing with state-of-the-art methods on different datasets (Sec. 4.5 - v - vi).

4.2 Related Work

One approach to represent the data consists of building equivariant-invariant representations. For instance, in [10, 93] they propose a translation-invariant representation, the Scattering Transform, which is stable under the action of small diffeomorphisms. In [59, 94], they focus on equivariant-invariant representations for images, which reduces the sample complexity and endow DNN's layers with interpretability.

The closest work to ours consist of learning the filter bank in an end-to-end fashion. [17, 86, 95, 96] investigated the learnability of a mother filter such that it can be jointly optimized with the DNN. In order to build the filter bank, this learnable mother filter is transformed by deterministic affine maps. The representation of the signal is obtained by convolving the filter bank elements with the signals. Recently, [97] investigated the learnability of the affine transformations, that is, the sampling of the dilation parameter of the affine group inducing the wavelet filter bank. Optimized jointly with the DNN, their method allows for an adaptive transformation of the mother filter. Our work generalizes this approach and provide its theoretical properties and building blocks.

One of the main drawbacks of these approaches using time-frequency representation

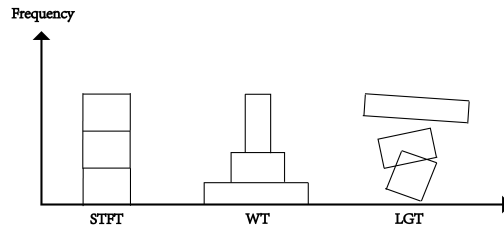


Figure 4.1 : **Time-Frequency Tilings** at a given time τ : (*left*) short-time Fourier transform, i.e., constant bandwidth, (*middle*) wavelet transform, i.e., proportional bandwidth, (*right*) Learnable Group Transform, i.e, adaptive bandwidth, the "tiling" is induced by the learned non-linear transformation underlying the filter bank decomposition.

is that the filter bank induces a bias that might not be adapted to the data. This bias can be understood by considering the time-frequency tiling of each GT. It is known that the spread of a filter and its Fourier transform are inversely proportional as per the Heisenberg uncertainty principle [98].

Following this principle, we can observe that in the case of STFT (respectively WT with a Gabor wavelet), at a given time τ , the signal is transformed by a window of constant bandwidth (respectively proportional bandwidth) modulated by complex exponential resulting in a uniform tiling (respectively proportional) on the frequency axis, Figure 4.1.

This implies that, for instance, in the case of WT, the precision in frequency degrades as the frequency increases while its precision in time increases [98]. Thus, WT is not adapted for fast-varying frequency signals [99]. In the case of STFT, the uniform tiling implies that the precision is constant along the frequency axis.

4.3 Formalism

Common time-frequency filter banks are built by transforming a mother filter that we denote by ψ . We consider the transformations of this mother filter defined as $\psi \circ g$, $g \in \mathcal{F}$, where \mathcal{F} defines the functional space of the transformation and $\psi \circ g$ denotes the function composition. Note that in signal processing, such a transformation is called warping [100,101]. Given a space \mathcal{F} , the filter bank with K filters is created by first, sampling K transformation maps from \mathcal{F} and then, by transforming the mother filter such as

$$\{\psi \circ g_1, \dots, \psi \circ g_K | g_1, \dots, g_K \in \mathcal{F}\}.$$

Now, let's denote a signal by $s \in L^2(\mathbb{R})$, we will consider the representation of the signal as the result of its convolution with the filter bank elements and denote it by

$$\mathcal{W}[s, \psi](\mathbf{g}, \cdot) = [\mathcal{W}[s, \psi](g_1, \cdot), \dots, \mathcal{W}[s, \psi](g_K, \cdot)]^T,$$

where

$$\mathcal{W}[s, \psi](g, \cdot) = s_i \star (\psi \circ g), \forall g \in \mathcal{F},$$

with \star the convolution operator and (\cdot) corresponds to the time axis.

Therefore, the properties of the representation are carried by the mother filter ψ , and space \mathcal{F} . In this work, we focus on the warping that generalizes common time-frequency decompositions as well as the properties carried by the associated filter bank, in particular we consider nonlinear warping. We provide a parameterization of such a warping and show how one can efficiently learn these parameters. The decomposition of the signal by this learned filter bank defines a Group Transform. The overall building blocks of the LGT, and its application on a signal is depicted in Figure 4.2.

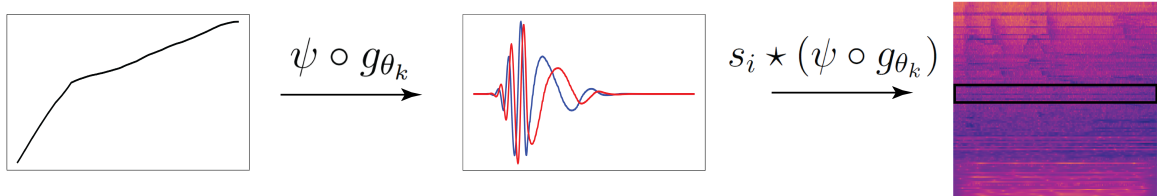


Figure 4.2 : **Learnable Group Transform:** (left) generating the strictly increasing continuous functions g_{θ_k} with parameters $\theta_k, \forall k \in \{1, \dots, K\}$, where K denotes the number of filters in the filter bank. The x -axis is the time variable and the y -axis the amplitude. (middle) The mother filter, ψ (presently a Morlet wavelet), is composed with each warping function g_{θ_k} , where the imaginary part is shown in red and the real part in blue. The x -axis represents the time and y -axis the amplitude of the filter. These transformations lead to the filter bank (only the k^{th} element is displayed). Then, the convolutions between the filter bank elements and the signal s_i lead to the LGT of the signal. The black box on the LGT representation (right) corresponds to the convolution of the k^{th} filter with the signal. In this figure, the horizontal axis corresponds to the time, each row corresponds to the convolution with a filter of the filter bank, and the color displays the amplitude of each inner product. Notice that a complex modulus has been applied to the LGT. The strictly increasing and continuous piecewise linear functions can be learned efficiently by back-propagating the error induced by the generated GT.

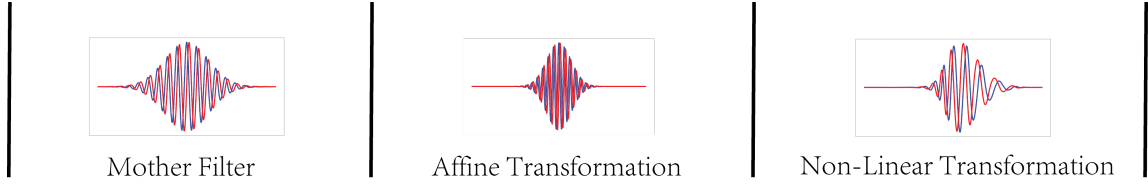


Figure 4.3 : **Transformation of a Morlet Wavelet:** For all the filters, the real part is shown in blue and the imaginary in red. (*left*) Morlet wavelet mother filter. (*middle*) Transformation of the mother filter with respect to an affine transform: the dilation parameter $0 < a < 1$, i.e., contraction, and translation $b = 0$, i.e., no translation. (*right*) Increasing and continuous transformation of the mother filter for some randomly generated function $g \in C_{\text{inc}}^0(\mathbb{R})$ leading to chirplet-like filter.

We propose to transform the mother filter by means of a subset of invertible maps on \mathbb{R} . Instead of the affine warping used in WT, we propose the use of a more general transformation map space \mathcal{F} . In particular, we will use the space of strictly increasing and continuous functions defined as

$$C_{\text{inc}}^0(\mathbb{R}) = \{g \in C^0(\mathbb{R}) | g \text{ is strictly increasing}\},$$

where $C^0(\mathbb{R})$ defines the space of continuous functions defined on \mathbb{R} . This set of functions is composed of invertible maps which is crucial in order to derive invariance properties as well as avoid artifacts in the transformed filters.

The transformation of a mother filter ψ is defined by the linear operator $\rho_{\text{inc}}(g)$ such as

$$\rho_{\text{inc}}(g)\psi = \psi \circ g, \quad g \in C_{\text{inc}}^0(\mathbb{R}),$$

By construction, this space allows for non-linear transformations of a mother filter. An example of such a warping can be visualized in Figure 4.3.

In the next paragraph, we introduce some filters that can be recovered using this transformation map. For some of these filters, the estimation of their parameter has been investigated [99, 102, 103], however, our method provides two benefits, first, the

Table 4.1 : Recovering well-known filters

$g \in C_{inc}^0(\mathbb{R})$	$\psi \circ g$
Affine	Wavelet
Quadratic Convex	Increasing Quadratic Chirplet
Quadratic Concave	Decreasing Quadratic Chirplet
Logarithmic	Logarithmic Chirplet
Exponential	Exponential Chirplet

generalization which alleviates the need of selecting a specific type of filter bank, second, the scalability of our method leading to a learnable filter bank.

4.4 Properties

i Recovering Standard Filter Banks The space $C_{inc}^0(\mathbb{R})$ allows us to span well known transformations. In particular, a filter can inherit a particular chirpyness* from nonlinear transformations belonging to $C_{inc}^0(\mathbb{R})$.

This property is interesting for the decomposition of non-stationary and fast-varying signals. In fact, various signals include such an intricate feature, such as bird song, speech, sonar system [105]. Among the possible transformations induced on a mother filter by the mapping $g \in C_{inc}^0(\mathbb{R})$, some of them correspond to well-known filters described in Table 4.1.

For instance, let's consider the case where \mathcal{F} is the space of linear function with positive slope and defined as $\forall g \in \mathcal{F}, g(t) = \frac{t}{\lambda}$, where λ is positive. In this case, we recover the transformation leading to the dilation or contraction of a wavelet

*Chirpyness is defined as the rate of change of the instantaneous frequency of the filter [104].

mother filter. The filter bank is then generated by sampling a few elements of the group. In the case of the dyadic wavelet transform, the dilation parameters follow a geometric progression of common ratio equals to 2, such as $\lambda_k = 2^{(k-1)/Q}$, $k = 1, \dots, K$, where $K = J \times Q$, with J and Q are the number of octaves and wavelets per octave, respectively. The filter bank obtained is $\left\{ \psi\left(\frac{t}{\lambda_1}\right), \dots, \psi\left(\frac{t}{\lambda_K}\right) \right\}$, and the representation of signal is obtained by convolutions between the filter bank elements and the signal. Equivalently, the space \mathcal{F} can be defined as affine, and the WT is achieved by inner products between the filters and the signal.

While the WT filter bank can easily be recovered, our modelization of the filter bank does not allow for elements with a number of oscillations that differ from the mother filter. To enable such a transformation, another function h with a number of oscillations that differs from the mother filter could be multiplied with the mother filter, such that $h \times \psi \circ g$ provides the elements of the filter bank. Therefore, STFT is not part of the representations that such a framework encompasses.

In this work, we also consider the case where the representation of the signal is performed by convolutions. This representation has equivariance properties that are induced by the convolutional operator as well as the space $C_{\text{inc}}^0(\mathbb{R})$.

ii Equivariance Properties of the learnable group transform The equivariance-invariance properties of signal representations play a crucial role in the efficiency of the algorithm at hand as they define how some variations in the signal may or may not be captured [106]. These properties can be intuitively explained and analyzed by considering the representation of the signal as a function of group elements. Details regarding the background of group theory and its link with wavelet analysis are provided in Sec. 2.2. Considering the mapping $\rho_{\text{inc}} = \psi \circ g$, $g \in C_{\text{inc}}^0(\mathbb{R})$, as a group

action on the space of the mother filter, i.e., $L^2(\mathbb{R})$, or more precisely, a representation of a group on $L^2(\mathbb{R})$, we can develop the equivariance properties of the LGT.

Proposition 1. ρ_{inc} is a group representation of \mathcal{G}_{inc} on $L^2(\mathbb{R})$.

Proof. Let $g, g' \in \mathcal{G}_{inc}$, then

$$\begin{aligned} [\rho_{inc}(g' \circledast g)\psi](t) &= \psi((g' \circledast g)(t)) \\ &= \psi(g'(g(t))) \end{aligned}$$

and,

$$\begin{aligned} [\rho_{inc}(g')\rho_{inc}(g)\psi](t) &= [\rho_{inc}(g')\psi](g(t)) \\ &= \psi(g'(g(t))) \end{aligned}$$

which verifies the homogeneity property. The linearity is implied by,

$$[\rho_{inc}(g)(\kappa\psi_1 + \psi_2)](t) = (\kappa\psi_1 + \psi_2)(g(t)) = \kappa\psi_1(g(t)) + \psi_2(g(t)), \forall t \in \mathbb{R}.$$

where $\psi_1, \psi_2 \in L^2(\mathbb{R})$ and $\kappa \in \mathbb{R}$. It is in fact a Koopman operator [107]. \square

We can consider the set $C_{inc}^0(\mathbb{R})$ with the operation \odot consisting of the composition of functions to form the group of strictly increasing and continuous maps denoted by \mathcal{G}_{inc} . This formulation eases the derivation of the equivariance properties of group transforms which can be defined for a group \mathcal{G} for all $g, g' \in \mathcal{G}$ by

$$\mathcal{W}[\rho(g')s_i, \psi](g, \cdot) = \mathcal{W}[s_i, \psi]((g')^{-1} \odot g, \cdot).$$

Transforming the signal with respect to the group \mathcal{G} and computing its representation is equal to computing the representation of the signal and then transforming the representation. If \mathcal{G} corresponds to the affine group, the associated group transform is

the WT, which is equivariant to scalings and translations. One can already notice that since $\mathcal{W}(\cdot, \cdot)$ employs convolution to decompose the signal, for any group \mathcal{G} , the LGT is translation equivariant. We now focus on more specific equivariance properties of the LGT by defining the local equivariance for all $g, g' \in \mathcal{G}$ by

$$\exists \tau \in \mathbb{R}, \mathcal{W}[\rho(g')s_i, \psi](g, \tau) = \mathcal{W}[s_i, \psi]((g')^{-1} \odot g, \tau).$$

That is, the representation of a local transformation of a signal in a window centered at τ is equal to the transformation of the representation at τ . The size of the window depends on the support of the filter. As a matter of fact, assuming that the representation of \mathcal{G}_{inc} is unitary, we have the following proposition.

Proposition 2. *The LGT is locally equivariant with respect to the action of the group \mathcal{G}_{inc} .*

Proof. Let $\tau \in \mathbb{R}$ and $g, g' \in \mathcal{G}_{\text{inc}}$,

$$\begin{aligned} \mathcal{W}[\rho_{\text{inc}}(g')s_i, \psi](g, \tau) &= \langle \rho_{\text{inc}}(g')s_i, \rho_{\text{inc}}(g)\overline{\psi_\tau} \rangle \\ &= \langle s_i, \rho_{\text{inc}}(g')^{-1}\rho_{\text{inc}}(g)\overline{\psi_\tau} \rangle \\ &= \langle s_i, \rho_{\text{inc}}(g'^{-1})\rho_{\text{inc}}(g)\overline{\psi_\tau} \rangle \\ &= \langle s_i, \rho_{\text{inc}}(g'^{-1} \odot g)\overline{\psi_\tau} \rangle \\ &= \mathcal{W}[s_i, \psi](g'^{-1} \odot g, \tau), \end{aligned}$$

where $\overline{\psi_\tau}(t) = \psi_\tau(-t)$ denotes the filter ψ centered at position τ . Then, there is not guarantee that this can be extrapolated to all $\tau \in \mathbb{R}$, i.e., in the convolution case, except in the affine case where the global transformation matches the iteration of a local one.

□

As we mentioned, a filter bank of K filters is created by sampling the space $C_{\text{inc}}^0(\mathbb{R})$. We now show how this sampling can be achieved efficiently by proposing a parametrization of functions belonging to such a space.

4.5 Experiments

i Sampling the group In this work, we are specifically interested in the learnability of such an increasing and continuous map. We provide a way to sample this space via its parameterization. We use piecewise affine functions constrained such that they belong to the class of strictly increasing and continuous functions, which can be efficiently performed by sorting the output of a 1-layer ReLU NN.

To implement the non-linear mapping induced by the representation of the piecewise affine group, we use the fact that a piecewise continuous function can be re-written as a 1-layer ReLU Neural Network [108, 109].

Besides the computational advantages of such relationships and the differentiable property of the weights of the NN, this model is a knot-free piecewise affine mapping, providing more flexibility regarding the warping function. The knot-free mapping implies that instead of having each affine piece of the function with uniform support, it can vary. As such, this flexibility induces better approximation property [110]. Then, the increasing constraint on the mapping is implemented by sorting the output of the NN. This operation has a $\mathcal{O}(n \log n)$ complexity and is applied on the warped time, which is usually of size $\approx 2^9$.

ii Objective Function and Learning: Let θ_k be the parameters of each increasing piecewise affine map computed by the NN and we denote by g_{θ_k} the sorted outputs

of the NN. The LGT filter bank has the following form

$$\{\psi \circ g_{\theta_1}, \dots, \psi \circ g_{\theta_K}\}.$$

Given a set of signals $\{s_i \in L^2(\mathbb{R})\}_{i=1}^N$ and given a task specific loss function L , we aim at solving the following optimization problem

$$\min_{\Theta} \sum_{i=1}^N L(F(\mathcal{W}[s_i, \psi](\mathbf{g}_{\Theta}, \cdot))),$$

where $\Theta = (\theta_1, \dots, \theta_K)$, N denotes the number of signals, K the number of filters, F represents a DNN, and we recall that

$$\mathcal{W}[s_i, \psi](\mathbf{g}_{\Theta}, \cdot) = [\mathcal{W}[s_i, \psi](g_{\theta_1}, \cdot), \dots, \mathcal{W}[s_i, \psi](g_{\theta_K}, \cdot)]^T.$$

Since, the g_{θ_k} are computed by sorting the output of the NN and the parameters can be learned by a gradient descent optimization jointly with the parameters of F .

iii Model Constraints to Reduce Aliasing The nonlinearity of the transformation might reduce the localization of the filter in the frequency domain, and produce aliasing. For some applications, the localization of each filter in the frequency domain is crucial, e.g., the bird detection task in Section v.

In order to limit the possible aliasing induced by the piecewise increasing mappings applied to the mother filter, we propose different settings. Besides, these constraints also impact the type of filter bank our method can reach.

First, we propose a normalization of the frequency of the transform filter (denoted in the result tables by nLGT). This normalization helps to reduce the aliasing induced by the filters. We propose to use \hat{f} , the normalized frequency f with respect to the maximum slope of the piecewise affine mapping. For instance, in the case of a Morlet wavelet, the normalization is as follows

$$(\psi \circ g_{\theta})(t) = \pi^{-\frac{1}{4}} \exp\left(2\pi j \hat{f} g_{\theta}(t)\right) \exp\left(-\frac{1}{2}(g_{\theta}(t)/\sigma)^2\right),$$

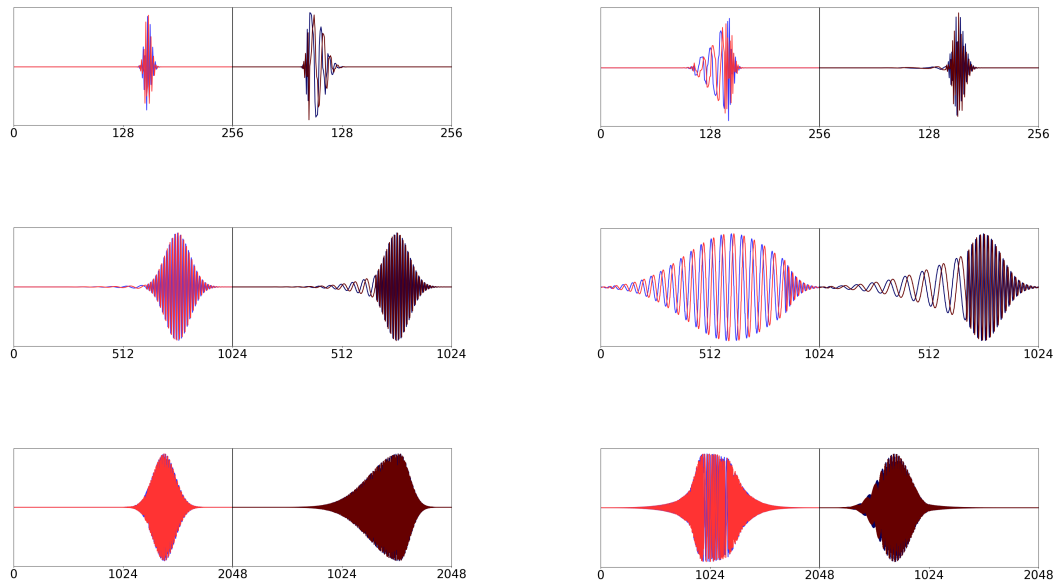


Figure 4.4 : **Learnable Group Transform Filters** for the Artificial Data - Each row displays two selected filters (left and right sub-figure) for different settings: (*from top to bottom*) nLGT, cLGT, cnLGT. For each subfigure, the left part corresponds to the filter before training and the right part to the filter after training. The blue and red denote the real and imaginary parts of the filters, respectively.

where $\hat{f} = f / \max_{l \in \{1, \dots, n\}} a_l$, where n denotes the number of pieces of the piecewise map, and a_l the slope of each piece, j is the imaginary unit, and σ is the width parameter defining the localization of the wavelet in time and frequency. This normalization will be performed for each sample of the group, and thus for each generated filter $k \in \{1, \dots, K\}$ of the filter bank.

Second, we constrain the domain of the piecewise affine map (denoted in the result tables by cLGT). In the following experiments, we propose a dyadic constraint of the domain as in the WT. The support of the filter is close to the support of a wavelet filter bank. However, the envelope of the filter and the instantaneous frequency still has a learned chirpyness.

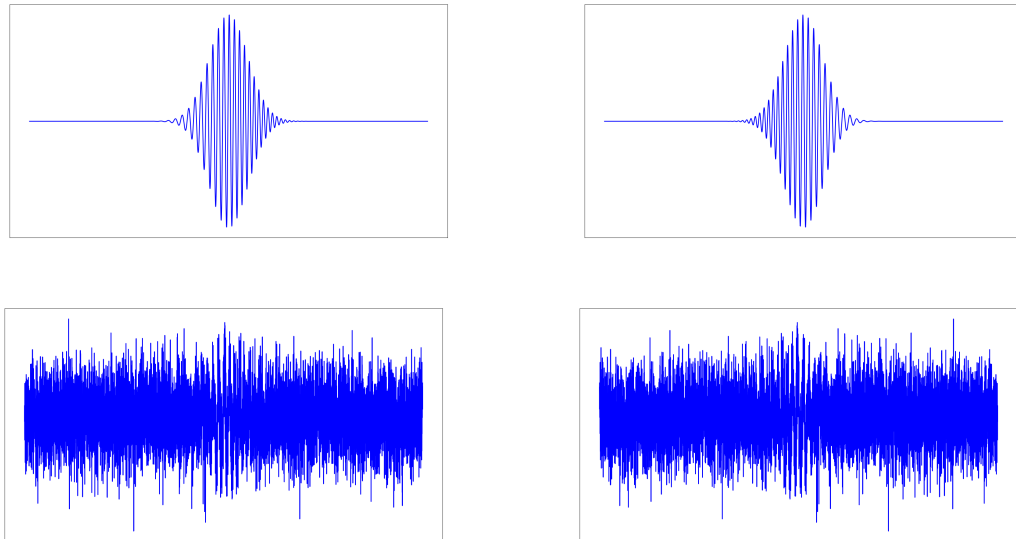


Figure 4.5 : **Artificial Dataset:** (*Top Left*) Ascending Chirp, (*Top Right*) Descending Chirp, i.e. class 0, (*Bottom Left*) Ascending Chirp plus Gaussian noise, (*Bottom Right*) Descending Chirp plus Gaussian noise, i.e., class 1. The samples contained in the training and testing set are higher in frequency and close to the Nyquist frequency.

Table 4.2 : Testing Accuracy for the Chirp Signals Classification Task

<i>Representation + Non-Linearity + Linear Classifier</i>	<i>Accuracy</i>
Wavelet Transform (64 Filters)	53.01 ± 5.1
Short-Time Fourier Transform (64 Filters)	65.1 ± 11.9
Short-Time Fourier Transform (128 Filters)	86.6 ± 9.8
Short-Time Fourier Transform (512 Filters)	100 ± 0.0
LGT (64 Filters)	92.9 ± 4.0
nLGT (64 Filters)	95.7 ± 3.3
cLGT (64 Filters)	56.8 ± 1.6
cnLGT (64 Filters)	100.0 ± 0.0

iv Classification of chirp signals We present an artificial dataset that demonstrates how a specific time-frequency tiling might not be adapted or would require cross-validations for a given task and data. To build the dataset, we generate one high frequency ascending chirp and one descending high-frequency chirp of size 8192 following the chirplet formula provided in [84]. Then for both chirp signals, we add Gaussian noise samples (100 times for each class), see Fig. 4.5. The task aims at being able to detect whether the chirp is ascending or descending. Both the training and test sets are composed of 50 instances of each class. For all models, set the batch size to 10, the number of epochs to 50. Each experiment was repeated 5 times with randomly sampled train and test set, and the accuracy was the result of the average over these 5 runs. Each GT is composed with a complex modulus, and the inference is performed by a linear classifier. For the case of WT and LGT, the size of the filters is 512.

As we can observe in Table 5.1, the WT, as well as the STFT with few numbers of filters, perform poorly on this dataset. The chirp signals to be analyzed are localized close to the Nyquist frequency, and in the case of WT, as illustrated in Figure 4.1, the wavelet filter bank has a poor frequency resolution in high frequency while benefiting from a high time resolution. In this experiment, we can see that this characteristic the WT time-frequency tiling implies that through time, the small frequency variations of the chirp are not efficiently captured. In the case of STFT, as the number of filters decreases, the frequency resolution is altered. Thus, this frequency variation is not captured. Using a large window for the STFT increases the frequency resolution of the tiling and thus enables to capture the difference between the two classes. In the LGT setting, the tiling has adapted to the task and produces good performances except for the cLGT model. In fact, the domain of the piecewise linear map is constrained to be

dyadic, and thus the adaptivity of the filter bank is reduced, which is not suitable for this specific task.

This experiment shows an example of signals that are not easily classified by neither the proportional-bandwidth nor the constant-bandwidth without considering cross-validation of hyperparameters.

Table 4.3 : Testing AUC for the Bird Detection Task

<i>Representation + Non-Linearity + Deep Network</i>	<i>AUC</i>
MFSC (80 Filters)	77.83 ± 1.34
Conv. Filter init. random (80 Filters)	66.77 ± 1.04
Conv. Filter init. Gabor (80 Filters)	67.67 ± 0.98
Spline Conv. init. random (80 Filters) [86]	78.17 ± 1.48
Spline Conv. init. Gabor (80 Filters) [86]	79.32 ± 1.52
LGT (80 Filters)	78.41 ± 1.38
nLGT (80 Filters)	75.50 ± 1.39
cLGT (80 Filters)	79.14 ± 0.83
cnLGT (80 Filters)	79.68 ± 1.35

v Supervised Bird Detection Task We now propose a large scale dataset to validate the suitability of our model in a noisy and realistic setting. The dataset is extracted from the Freesound audio archive [81]. This dataset contains about 7,000 field recording signals of 10 seconds sampled at 44 kHz, representing slightly less than 20 hours of audio signals. The content of these recordings varies from water sounds to city noises. Among these signals, some contain bird songs that are mixed with different background sounds having more energy than the bird song. The given task

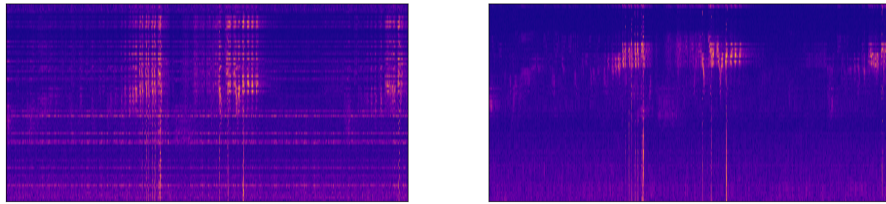


Figure 4.6 : **Learnable Group Transform** - Visualization of a sample containing a bird song (cLGT), where (*left*) at the initialization and (*right*) after learning. For each subfigure, the x -axis corresponds to time and the y -axis to the different filters. Notice that the y -axis usually corresponds to the scale or the center-frequency of the filters. We can observe that compared to the initialization, the learned representation is sparser and the SNR is increased. Besides, the representation is less redundant in the frequency axis.

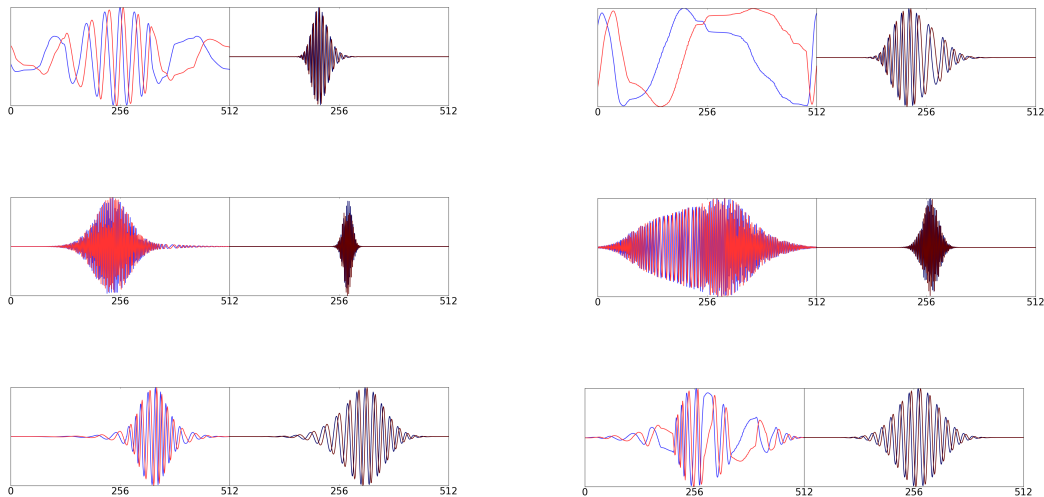


Figure 4.7 : **Learnable Group Transform Filters** for the Bird Detection Data - Each row displays two selected filters (left and right sub-figure) for different settings: (*from top to bottom*) LGT, nLGT, cLGT. For each subfigure, the left part corresponds to the filter before training and the right part to the filter after training. The blue and red denote the real and imaginary parts of the filters, respectively.

is a binary classification where one should predict the presence or absence of a bird song. As the dataset is unbalanced, we use the Area Under Curve (AUC) metric. The

results we propose for both the benchmarks and our models are evaluated on a test set consisting of 33% of the total dataset.

In order to compare with previously used methods, we use the same seeds to sample the train and test set, the batch size, i.e., 10, and the learning rate cross-validation grid as in [86]. For each model, the best hyperparameters are selected, and we train and evaluated randomly 10-times the models with early stopping, the results are shown in Table 4.3. While the first layer of the architecture has a model-dependent representation (i.e., MFSC, LGT, Conv. filters,...), we use the state-of-the-art architecture defined in [82]. Notice that this specific DNN architecture has been designed and optimized for MFSC representation.

As we can see in Table 4.3, the case without constraints (LGT) reaches better accuracy than the domain expert benchmark (MFSC). Besides, including more constraints on the model (cnLGT) reduces overfitting and further improve results to outperform the other benchmarks. One can also remark that both the LGT framework and learnable mother wavelet reach almost the same accuracy, while they both outperform the hand-crafted feature as well as the unconstrained convolutional filters. One can notice that all the learned filters in Figure 4.7 contain either an increasing chirp or a decreasing chirp, corresponding respectively to the convexity or concavity of the instantaneous phase of the filter and thus of the piecewise linear map. Such a feature is being used and is crucial in the detection and analysis of bird song [111].

vi Classification of haptics data The Haptics dataset is a classification problem with five classes and 155 training and 308 testing samples from the UCR Time Series Repository [116], where each time-series has 1092 time samples. As opposed to the bird dataset where features of interests are known, and competitive methods have

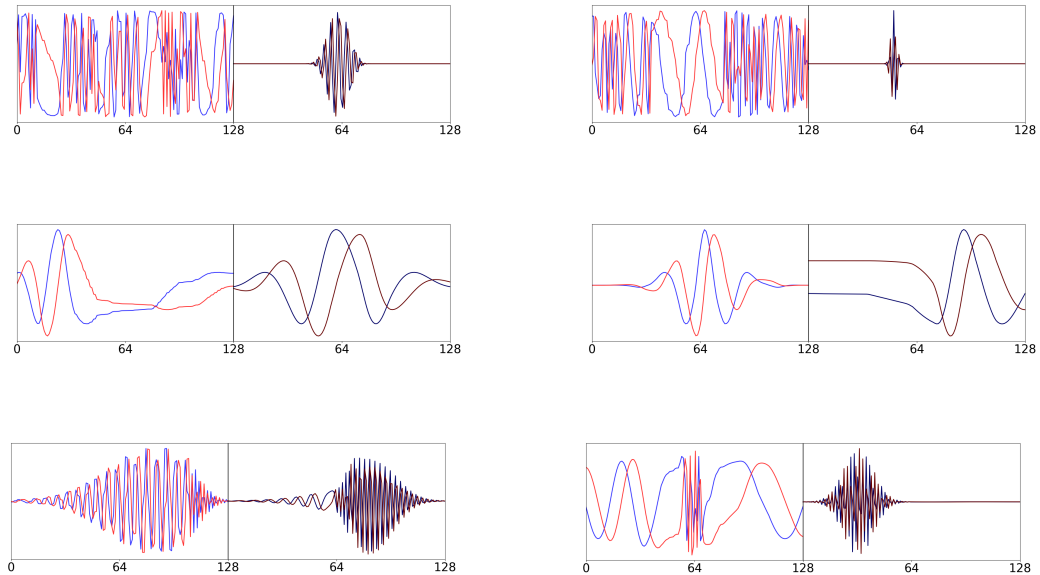


Figure 4.8 : **Learnable Group Transform Filters** for the Haptics Data - Each row displays two selected filters (left and right sub-figure) for different settings: (*from top to bottom*) nLGT, cLGT, cnLGT. For each subfigure, the left part corresponds to the filter before training and the right part to the filter after training. The blue and red denote the real and imaginary parts of the filters, respectively.

been established, there is no expert knowledge regarding the specific signal features (see Table 4.4). One can see that our method outperforms other approaches in the cLGT setting while performing the classification with a linear classifier as opposed to other methods using DNN algorithms. This demonstrates the capability of our method to transform the data efficiently while not requiring a further change of basis as well as knowledge on the features of interests. Besides, even in a small dataset regime, our approach is capable of learning an efficient transformation of the data.

We provide in Figure 4.8 the visualization of some sampled filters before and after learning. As opposed to the supervised bird dataset, we can see that the filters do not coincide with well-known filters that are commonly used in signal processing. This is

Table 4.4 : Testing Accuracy for the Haptics Classification Task

<i>Model</i>	<i>Accuracy</i>
DTW [112]	37.7
BOSS [113]	46.4
Residual NN [114]	50.5
COTE ([115]	51.2
Fully Convolutional NN [114]	55.1
WD + Convolutional NN [97]	57.5
LGT (96 Filters)+ Non-Linearity + Linear Classifier	53.5
nLGT (96 Filters)+ Non-Linearity + Linear Classifier	50.4
cLGT (96 Filters)+ Non-Linearity + Linear Classifier	58.2
cnLGT (96 Filters)+ Non-Linearity + Linear Classifier	54.3

an example of an application where the features of interest in the signals are unknown, and one requires a learnable representation.

4.6 Conclusion

We enable the learnability of Group Transform and generalize the wavelet transform by introducing non-linear transformations of a mother filter as well as an efficient way to sample this mapping. We establish the connections with well-known time-frequency filters that are common in diverse biological signals as well as the derivation of the equivariance properties of the LGT. Also, we have shown a tractable way to learn to sample these transformations using a 1-layer NN enabling an end-to-end approach. Our approach competes with state-of-the-art methods without a priori knowledge

on the signal power spectrum and outperforms classical hand-crafted time-frequency representations. Interestingly, in the bird detection experiment, we recover chirplet filters that are known to be crucial to their detection, while in the case of the haptic dataset where important features to be captured to perform the classification of the signals are unknown, the filters learned are very dissimilar to classical time-frequency filters and allow to outperform state-of-the-art methods with a linear classifier.

5

Learnable Invariant Distance

Clustering algorithms aim at discovering patterns in the data that enable their characterization, identification, and separation. The development of such a framework without any prior information regarding the data remains one of the milestones of machine learning that would assist clinicians, physicists, and data scientists, among others, with a better pattern discovery tool [117,118].

While supervised learning has been converging toward the almost exclusive use of Deep Neural Networks (DNN), avoiding the development of handcrafted features to provide the desired linearly separable embedding map, unsupervised clustering algorithms take various forms depending on the application at hand [119–121]. For instance, the usage of SIFT features combined with clustering algorithm for medical imaging [122], the extraction of DNNs embedding used as the input of the K -means algorithm for computer vision tasks [123], and the combination of signal-processing features extractors combined with Gaussian mixture model to understand the nature of the various seismic activities [124]. The important role of clustering algorithms in

assisting medical diagnoses as well as scientific discoveries highlight the importance of the development of an *interpretable* and *theoretically guaranteed* tool [125, 126].

In this work, we focus our attention on the K -means clustering algorithm [127] and its application to 2-dimensional signals, such as images or time-frequency representations. Well-known for its simplicity, efficiency, and interpretability, the K -means algorithm partitions the data space into K disjoint regions. Each region is represented by a centroid, and each datum is assigned to the closest centroid's region. The integral part in the design of a clustering algorithm is the choice of an appropriate distance, and the number of clusters [128–130]. While the Euclidean distance makes the design of the algorithm straightforward, this measure of similarity might omit the geometrical relationships between data points [131]. In fact, a small rigid perturbation of an image, such as rotation or translation, is enough to change the cluster assignment.

There are two major difficulties in constructing a distance for a clustering algorithm; on the one hand, the metric should take into account the geometry of the data, e.g., be invariant to rigid transformations for images, and on the other hand, the metric should be interpretable as it is tied to the interpretability of the algorithm [131].

In this work, we tackle these two difficulties by introducing in our similarity measure the spatial transformations inherent to the geometry of the data at hand. In particular, we: (i) formulate an interpretable and theoretically guaranteed K -means framework capable of exploiting the symmetry within the data, (ii) extend prior work on metrics invariant to rigid transformations to non-rigid transformations, thus taking into account a more realistic set of nuisances and (iii) allow the learnability of the symmetry underlying the data at hand, therefore enabling the exploration of data where the equivalence classes are yet to be determined.

To learn the symmetry in the data and perform their transformations, we will use

the spatial transformer framework, which was successfully introduced in [56]. This allows us to provide a learnable metric invariant to non-rigid transformations that is used as the K -means distortion error.

While many approaches to learn and estimate non-rigid transformations have been proposed, we will follow one of nowadays mainstream approaches developed in [56] where the Thin Plate Spline is used as a differentiable deformation model. Our attempt is, in fact, not to compare among deformation models but to consider a way to approach the learnability of invariances in an unsupervised setting such that it is effective, tractable, and interpretable.

5.1 Outline of the Chapter and Contribution Summary

- We propose a novel approach to tackle clustering using a novel adaptive similarity measure within the K -means framework that considers non-rigid transformations (Sec. 5.3).
- We derive an appropriate update rule for the centroids that drastically improves both the interpretability of the centroids and their quality (Sec. 5.5).
- We provide its invariance properties (Sec. 5.6 - ii), convergence guarantees (Sec. 5.6 - iii), and geometrical interpretations of our approach (Sec. 5.6 - iv).
- Finally, we show numerically that our unsupervised algorithm competes with state-of-the-art methods on various datasets while benefiting from interpretable results (Sec. 5.7).

5.2 Related Work

The development of measures invariant to specific deformations has been under investigation in the computer vision community for decades [132–134]. By considering affine transformations such as shearing, translation, and rotation of the data as being nuisances, these approaches propose a distance that reduces the variability intrinsic to high-dimensional images. These works are considered as appearance manifold-based framework; that is, the distance are quantified by taking into account geometric proximity [135–138].

While the development of affine invariant metrics is pretty standard, their extension to more general non-rigid transformations requires more attention. Recently, various deep learning methods proposed ways to learn diffeomorphic transformations [139–143]. Others adopt a more theoretically grounded approach based on group theory as in [144–147] as well as the statistical “pattern theory” approach developed in [148, 149].

5.3 Formalism

We recall that in this work we will consider 2-dimensional signals defined by their width and height, such as images and time-frequency representation of time-series. Given a set of 2-dimensional signals, $\{x_i\}_{i=1}^N$, with $x_i \in \mathbb{R}^n$, the K -means algorithm aims at grouping the data into K distinct clusters defining the partition $\mathcal{C} = \{C_k\}_{k=1}^K$, with $\cup_k C_k = \{x_i\}_{i=1}^N$ and $C_i \cap C_j = \emptyset, \forall i \neq j$. Each cluster C_k of the partition is represented by a centroid $\mu_k \in \mathbb{R}^n, \forall k \in \{1, \dots, K\}$.

As for the K -means algorithm, the goal of the ST K -means is to find centroids

minimizing the following distortion error

$$\min_{c, \mu_1, \dots, \mu_K} \sum_{k=1}^K \sum_{i: x_i \in C_k} d(x_i, \mu_k) \quad . \quad (5.1)$$

The assignment of a signal x_i to a cluster C_k is achieved through the evaluation of the similarity measure, d , between the signal and each centroid. A signal x_i belongs to the cluster C_l if and only if $l = \arg \min_k d(x_i, \mu_k)$. While the standard K -means algorithm makes use of the Euclidean distance, i.e., $d(x_i, \mu_k) = \|x_i - \mu_k\|_2^2$, we instead propose to use the following deformation invariant similarity measure

$$d(x_i, \mu_k) := \min_{\nu \in \mathbb{R}^{2\ell}} \|\mathcal{J}(x_i, \nu) - \mu_k\|_2^2 \quad , \quad (5.2)$$

where the transformer operator, denoted by \mathcal{J} , allows for non-rigid image transformations. It is based on the composition of two mappings; a deformation map and a sampling function. The deformation maps a uniform grid of 2-dimensional coordinates to provide its transformed version. The sampling function samples the signal with respect to a given grid of 2-dimensional coordinates.

This similarity measure represents the least-square distance between the centroids and the datum that has been fit to the centroid via the spatial transformer operator. Once this fitting is done for each centroid, the cluster assignment is done based on the argmin of those distances, i.e., the data x_i is assigned to $\arg \min_k d(x_i, \mu_k)$. Therefore, the underlying assumption of our approach is that the distance between the optimal transformation of a signal into a centroid belonging to the same "class" should be smaller than the distance between its optimal transformation into a centroid that does not. That is, let x_i be geometrically near μ_k , then $\min_{\nu \in \mathbb{R}^{2\ell}} \|\mathcal{J}(x_i, \nu) - \mu_k\|_2^2 < \min_{\nu \in \mathbb{R}^{2\ell}} \|\mathcal{J}(x_i, \nu) - \mu'_k\|_2^2$.

This measure requires solving a non-convex optimization problem. It can be achieved in practice by exploiting the spatial transformer's differentiability with

respect to the landmarks ν . As a result, we can learn the transformation by performing gradient-descent based optimization [150].

The crucial property of the measure we propose is its invariance to deformations that are spanned by the spatial transformer. This means that evaluating Eq. 5.2 with any datum that is transformed from the spatial transformer will produce the same value, as long as no information is lost.

5.4 Image Transformations

The mapping we select to enable the learnability of the transformation in the coordinate space of the 2-dimensional signal is the Thin-Plate-Spline (TPS) interpolation technique [151–153] which produces smooth surfaces from \mathbb{R}^2 to \mathbb{R}^2 [154]. We refer the reader to Sec. 2.4 for details regarding this method. We consider as learnable parameters of the TPS a set of 2-dimensional coordinates, called landmarks, and denoted by ν . Given a set of landmarks, the TPS provides the transformation map of a 2-dimensional grid. That is, the euclidean plane is bent according to the learned landmarks.

In Fig. 5.1, we show on the bottom right the grid associated with the $\ell = 6^2$ landmarks. Each grid corresponds to the spatial transformation applied to the handwritten digit 4. The transformation of the signal based on these new coordinates is produced by performing bilinear interpolation using the original signal (top left) and the new coordinates; the details are provided in Sec. 2.4.

The spatial transformer is the composition of these two maps and is defined as

$$\mathcal{T}(x, \nu) \quad , \quad (5.3)$$

where $x \in \mathbb{R}^n$ is the original 2-dimensional signal, $\nu \in \mathbb{R}^{2\ell}$ is the set of 2-dimensional transformed coordinate to be learned. Note that 2ℓ can be smaller than the dimension

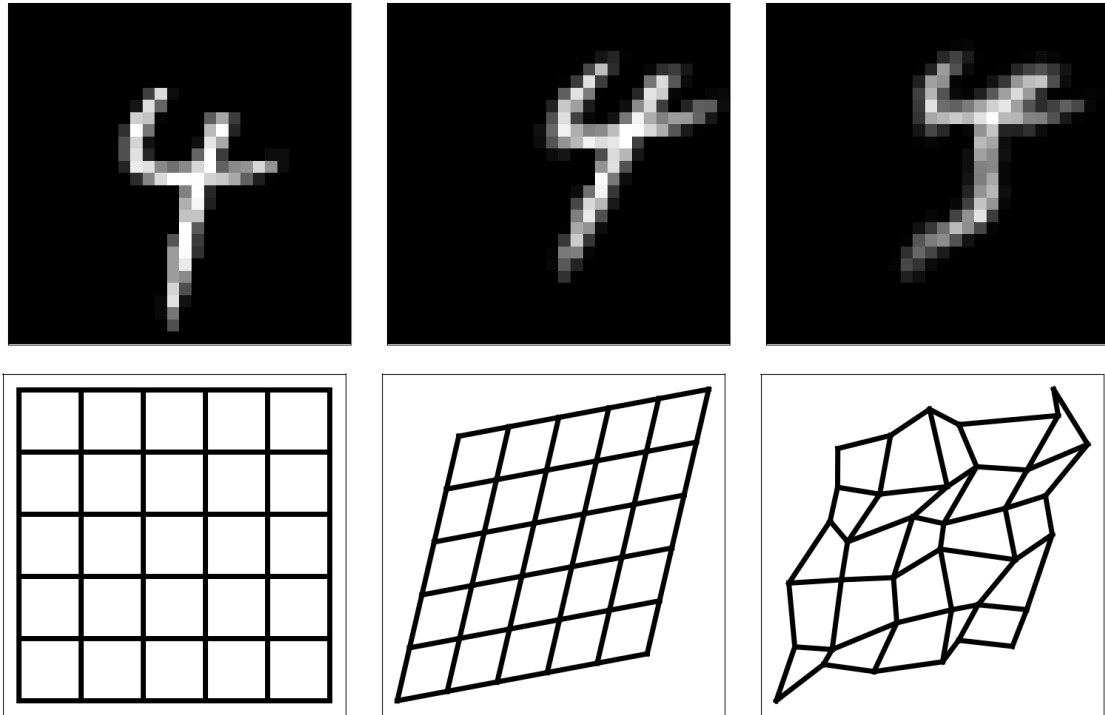


Figure 5.1 : **Spatial Transformations** - Visualizations of a sample taken from the MNIST dataset and its transformed versions. Each image results from the application of the spatial transformer that take as input the original signal (top left), and the grid displayed below its transformed version. (*Left*) we observe the original image and its associated original transformation grid, which corresponds to the identity transform. (*Middle*) the image has been transformed by the affine transformation induced by the associated grid. (*Right*) the image transformed by the non-rigid transformation using the TPS induced by the grid below it.

of the image as the TPS interpolates to re-scale the transformation to any size. Such a framework composing the TPS and bilinear interpolation has been defined as spatial transformer in [56]. However, in their work, the inference of the non-rigid transformations is performed using each datum as the input of a “localisation network”; instead, we directly learn the transformation parameters.



Figure 5.2 : Orbit of a hand-written digit 7 according to the group of rotation $SO(2)$.

5.5 Learning the Spatial Transformer K -means

Solving the optimization problem in Eq. 5.1, similarly to K -means, is an NP-hard problem. A popular tractable solution nonetheless exists and is known as the two-step Lloyd algorithm [155].

In the ST K -means, the first step of the Lloyd algorithm consists of assigning the data to the clusters using the newly defined measure of similarity in Eq. 5.2 . The second step is the update of the centroids using the previously determined cluster assignment. It corresponds to the result of the optimization problem: $\arg \min_{\mu_k} \sum_{i: x_i \in C_k} d(x_i, \mu_k)$, provided in following Proposition 3.

Proposition 3. *The centroids update of the ST K -means algorithm are given by*

$$\mu_k^* := \frac{1}{|C_k|} \sum_{i: x_i \in C_k} \mathcal{T}(x_i, \nu_{i,k}^*), \quad \forall k \quad (5.4)$$

where $|C_k|$ denotes the cardinal of the set C_k , $\nu_{i,k}^*$ is the set of parameters of the TPS that best transforms the signal x_i into the centroid μ_k , that is, $\nu_{i,k}^* = \arg \min_{\nu \in \mathbb{R}^{2l}} \|\mathcal{T}(x_i, \nu) - \mu_k\|_2^2$.

We consider the Fréchet mean of the centroid k to be the solution of the following optimization problem, $\arg \min_{\mu_k} \sum_{i: x_i \in C_k} d(x_i, \mu_k)$. Using our similarity measure, we obtain the following.

Proof. The Fréchet mean for the cluster C_k is defined as $\arg \min_{\mu_k} \sum_{i: x_i \in C_k} \|\mathcal{T}(x_i, \nu^*) - \mu_k\|_2^2$ since the optimization problem is convex in μ_k (as the result of the composition of the

identity map and a norm which are both convex) we have $\mu_k^* : \nabla_{\mu} \sum_{i:x_i \in C_k} \|\mathcal{T}(x_i, \nu^*) - \mu_k\|^2 = 0$. with,

$$\nabla_{\mu} \sum_{i:x_i \in C_k} \|\mathcal{T}(x_i, \nu^*) - \mu_k\|^2 = 2(|C_k|) \times \mu_k + 2 \sum_{i:x_i \in C_k} \mathcal{T}(x_i, \nu^*). \quad (5.5)$$

□

The averaging in Eq. 5.4 is performed on the transformed version of the signals. The ST K -means thus considers the topology of the signal's space. A pseudo-code of the centroid update Eq. 5.4 is presented in Algo. 1.

Algorithm 1 Centroids Updates of ST K -means

Input: Cluster C_k , TPS parameters $\{\nu_{i,k}^*\}_{i:x_i \in C_k}$

Output: Centroids update μ_k^*

- 1: Initialize $\mu_k = 0$
 - 2: **for** $i : x_i \in C_k$ **do**
 - 3: Compute $\mu_k = \mu_k + \mathcal{T}_{\ell}(x_i; \nu_{i,k}^*)$, Eq. 5.4
 - 4: $\mu_k^* = \frac{\mu_k}{|C_k|}$
-

The ST K -means, which aims to minimize the distortion error Eq. 5.1 is done by alternating between the two steps detailed above until convergence, as summarized in Algo. 2.

The update in Eq. 5.4, induced by our similarity measure, alleviates a fundamental limitation of the standard K -means. In fact, in the standard K -means, the average of the data belonging to a cluster C_k , $\frac{1}{|C_k|} \sum_{i:x_i \in C_k} x_i$, consists of an averaging of the signals without deforming them, which, as a result, does not account for the non-euclidean geometry of the signals [156, 157].

Algorithm 2 Spatial Transformer K -means

Input: Initial centroids μ_k , dataset $\{x_i\}_{i=1}^N$

Output: Cluster partition $\{C_k\}_{k=1}^K$

- 1: **repeat**
 - 2: **for** $i = 1$ to N **do**
 - 3: **for** $k = 1$ to K **do**
 - 4: Compute and store $d(x_i, \mu_k)$ by solving Eq. 5.2
 - 5: Assign x_i to C_l where $l = \arg \min_k d(x_i, \mu_k)$
 - 6: Update the centroid μ_k using Algo. 1
 - 7: **until** Convergence
-

5.6 Properties & Geometrical Aspects

i Quasi-pseudo-semi Metric

Proposition 4. *The similarity measure defined by $\min_{\nu \in \mathbb{R}^{2\ell}} \|\mathcal{T}(x, \nu) - \mu\|$ is a Quasipseu-dosemimetric.*

Proof. Let's first define the orbit of an image with respect to the TPS transformations. Note that, the TPS does not form a group as it is a piecewise mapping. However, we know that it approximate any diffeomorphism on \mathbb{R}^2 . Therefore, for sake of simplicity, we will make a slight notation abuse by considering the orbit, equivariance, and others group specific properties as being induced by the spatial transformer \mathcal{T} .

Definition 3. *We define the orbit an image x under the action the \mathcal{T} by*

$$\mathcal{O}(x) = \{\mathcal{T}(x, \nu) | \nu \in \mathbb{R}^{2\ell}\}. \quad (5.6)$$

Let's now consider each metric statement: 1) It is non-negative as per the use of a norm.

2) **Pseudo:** $\min_{\nu \in \mathbb{R}^{2\ell}} \|(\mathcal{J}(x, \nu) - \mu\| = 0 \Leftrightarrow \exists \nu \in \mathbb{R}^{2\ell}, s.t. x = \mathcal{J}(x, \nu) \Leftrightarrow x \sim_{\mathcal{J}} \mu$, that is, x and μ are equivariant with respect to the transformations induced by \mathcal{J} . Thus, $d(x, \mu) = 0$ for possibly distinct values x and μ , however, these are not distinct when we consider the data as any possible point on their orbit with respect to the group of diffeomorphism. In fact, the distance is equal to 0 if and only if, μ and x are equivariant.

3) **Quasi:** The asymmetry of the distance is due to the non-volume preserving deformations considered. In fact, we do not consider the Haar measure of the associated diffeomorphism group and consider the L_2 distance with respect to the Lebesgue measure. Although the asymmetry of d does not affect our algorithm or results, a symmetric metric can be built by normalizing the distance by the determinant of the Jacobian of the transformation. Such a normalization would make the metric volume-preserving and as a result make the distance symmetric.

4) **Semi:** If $x, x', x'' \in \mathcal{O}$, then $d(x, x'') = d(x, x') = d(x', x'') = 0$ as it exist a ν, ν', ν'' such that the TPS maps each data onto the other as per definition of the orbit, thus the triangular inequality holds. If $x, x'' \in \mathcal{O}$ and $x' \notin \mathcal{O}$, we have $d(x, x'') = 0 \leq d(x, x') + d(x', x'')$. If $x, x' \in \mathcal{O}$ and $x'' \notin \mathcal{O}$, we have $d(x', x'') = d(x, x'')$, and since $0 \leq d(x, x')$, the inequality is respected. However, if x, x', x'' belong to three different orbits, then we do not have the guarantee that then triangular inequality holds. In fact, it will depend on the distance between the orbits which is specific to each dataset. □

ii Invariance Property Motivated by the fact that small non-rigid transformations, usually, do not change nature of an image, we propose to exploit the invariance property of the similarity measure we proposed.

In this section, for sake of simplicity we will assume that the transformations belong to the group of diffeomorphism. In practice, the TPS can only approximate element of such group, and the constraint we impose on the transformation, e.g., number of landmark, also limit the type of diffeomorphism that can be approximated, therefore, we could instead consider that we approximate a subgroup of the diffeomorphism group.

Let's define an invariant similarity measure under the action of such group. That is, the similarity between two 2-dimensional signals remain the same under any diffeomorphic transformations. We propose to define the invariance in the framework of centroid-based clustering algorithm as follows.

Definition 4. *An invariant similarity measure with respect to $\text{diff}(\mathbb{R}^2)$ is defined as $d : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^+$ such that for all images $x \in \mathbb{R}^n$, all centroids $\mu \in \mathbb{R}^2$, and all group elements $\forall g \in \text{diff}(\mathbb{R}^2)$, we have*

$$d(x, \mu) = d(g \star x, \mu), \quad (5.7)$$

where $g \star x$ denotes the action of the group element g onto the image x .

The similarity used in Eq. 5.2 of the optimization problem is $\text{diff}(\mathbb{R}^2)$ -invariant as per Definition 4.

Proposition 5. *The similarity $\min_{g \in \text{diff}(\mathbb{R}^2)} \|g \star x - \mu\|$ is $\text{diff}(\mathbb{R}^2)$ -invariant.*

Proof. Let consider $g^\star = \arg \min_{g \in \text{diff}(\mathbb{R}^2)} \|g \star x - \mu\|$, we have $\arg \min_{g \in \text{diff}(\mathbb{R}^2)} \|g \cdot g' \star x - \mu\| = g^\star \cdot g'^{-1}$, where g'^{-1} is the inverse group element of g' . In fact, $\|g^\star \cdot g'^{-1} \cdot g' \star x - \mu\| =$

$\|g^* \star x - \mu\|$. Since for all $g' \in \text{diff}(\mathbb{R}^2)$, it exists an inverse element g'^{-1} , we have that $\forall g \in \text{diff}(\mathbb{R}^2)$, $d(g' \star x, \mu) = d(x, \mu)$.

That is, by definition of the group, there is always another element that minimizes the loss function by using the composition between the inverse element of the group that has just been added, g' , and the optimal element g^* . \square

iii Convergence of the Spatial Transformer K-means As we mentioned, our development is motivated by the interest in proposing a novel way to think about invariance in an unsupervised fashion while conserving the interpretability and theoretical guarantees of the K -means algorithm. We propose here to prove the convergence of the ST K -means algorithm following the generalization of clustering algorithms via the Bregman divergence as developed in [158]. In their work, they provide the class of distortion function that admits an iterative relocation scheme where a global objective function, such as the one in Eq. 5.1, is progressively decreased. We, therefore, prove that Algo. 2 monotonically decreases the distortion error of the ST K -means in Eq. 5.1 which in turn implies that Algo. 2 converges to a local optimal.

Proposition 6. *Under the assumption that the spatial transformation optimization problem in Eq. 5.2, reaches a global minimum, the ST K -means algorithm described in Algo. 2 terminates in a finite number of step at a partition that is locally optimal.*

Proof. Following the notation of Sec. 2.4, we can define the spatial transformer operator \mathcal{T} as the composition of the TPS and bilinear interpolation map. That is, $\mathcal{T}(x, \nu) = \Gamma[F(\nu), x]$. Now the aim is to prove that $\min_{\nu \in \mathbb{R}^{2l}} \|\mathcal{T}(x, \nu) - \mu\|_2^2$ defines a Bregman divergence measure as in [158]. In such a case, Algo. 2 defines a special case of the Bregman divergence hard-clustering algorithm again defined in [158] which is proven to converge.

Let's first start by making an assumption on the data x , we can without loss of generality assume that they are non-negative as we are dealing either with images or time-frequency representation where a modulus is applied to obtain the 2-dimensional real representation. Then, we also assume that the minimum over the transformation parameters ν reaches a global unique minimum, denoted by ν^* . Now,

$$\begin{aligned} \|\mathcal{T}(x, \nu^*) - \mu\|_2^2 &= \langle \mathcal{T}(x, \nu^*), \mathcal{T}(x, \nu^*) \rangle + \langle \mu, \mu \rangle - 2\langle \mathcal{T}(x, \nu^*), \mu \rangle \\ &= \langle \mathcal{T}(x, \nu^*), \mathcal{T}(x, \nu^*) \rangle - \langle \mu, \mu \rangle - \langle \mathcal{T}(x, \nu^*) - \mu, 2\mu \rangle \quad , \end{aligned}$$

Now it is clear that $\mu = \mathcal{T}(\mu, 0)$ which consists in the identity transform of the centroid μ . Then we denote by $\phi_{\nu^*}(y) = \langle \mathcal{T}(y, \nu^*), \mathcal{T}(y, \nu^*) \rangle$ where $y \in \mathbb{R}^n$, and obtain that,

$$\|\mathcal{T}(x, \nu^*) - \mu\|_2^2 = \phi_{\nu^*}(x) - \phi_0(\mu) - \langle \mathcal{T}(x, \nu^*) - \mathcal{T}(\mu, 0), \nabla \phi_0(\mu) \rangle \quad .$$

Now, we know that $\langle x, x \rangle$ is non-decreasing w.r.t each dimension since the image or time-frequency representation are positive real valued, and the inner product defines a strictly convex map. Then, we also know that $\mathcal{T}(x, \nu^*) = \Gamma[F(\nu^*), x]$ is defined as the composition of the TPS for the coordinate and the bilinear map for the image, which can be formulated as a linear transformation with respect to the data $x : Ax$, where A is a structured sparse matrix where each block denotes the dependency to nearby pixels. Therefore this mapping is convex. As a composition between a non-decreasing w.r.t each dimension and strictly convex function with a convex function, ϕ_{ν^*} is strictly convex, which complete the proof.

□

iv Geometrical Interpretation of the Similarity Measure One of the great benefit of the K -means algorithm is the interpretability of the regions composing its partitioning. In particular, they are related to Voronoi diagrams which are well

studied partitioning techniques [159, 160]. Following this framework, we propose now to highlight the regions defined by the ST K -means algorithm. This is achieved by analysing the following sets $\forall k \in \{1, \dots, K\}$

$$R_k = \{x \in \mathbb{R}^n \mid d(x, \mu_k) \leq d(x, \mu_j), \quad \forall j \neq k\}, \quad (5.8)$$

where we recall $d(x, \mu_k) = \min_{\nu \in \mathbb{R}^{2\ell}} \|\mathcal{T}(x, \nu) - \mu_k\|_2^2$. Such a partitioning falls in the framework of a special type of Voronoi diagram.

Proposition 7. *The partitioning induced by the ST K -means corresponds to a weighted Voronoi diagram where each region's size depends on the per data spatial transformations.*

Let's start by re-writing the similarity measure as to analytically express a metric tensor that would be the weight in the weighted Voronoi diagram the ST K -means defines. Using App. 2.4, we can re-write $d(x, \mu_k) = \min_{\nu \in \mathbb{R}^{2\ell}} \|\mathcal{T}(x, \nu) - \mu_k\|_2^2 = \min_{\nu \in \mathbb{R}^{2\ell}} \|A(\nu)x - \mu_k\|_2^2$, where $A(\nu)$ is bilinear in the coordinates that are induced by the TPS. In this formulation we can observe that ν defines the displacement vector w.r.t the original uniform grid of landmark. That is, if ν is the null vector, then $A(\nu)x = x$. Now, we assume that such linear operator is invertible, i.e., the TPS transformation is invertible (note that this is not always the case [161]). Then, we can re-write $d(x, \mu_k)$ as

$$\min_{\nu \in \mathbb{R}^{2\ell}} \|x - A(\nu_{x,k})^{-1} \mu_k\|_{A(\nu_{x,k})^T A(\nu_{x,k})}^2,$$

where $\|x\|_{A(\nu_{x,k})^T A(\nu_{x,k})} = x^T A(\nu_{x,k})^T A(\nu_{x,k}) x$, and $A(\nu_{x,k})^T A(\nu_{x,k})$ defines the metric tensor, and the notation $\nu_{x,k}$ indicates that the displacement vector ν depends on the centroid μ_k and the datum x . Also, note that while $A(\nu_{x,k})$ defines the transformation operator to map x onto μ_k , $A(\nu_{x,k})^{-1}$ is the inverse operator mapping the centroid μ_k to the datum x .

Now the tuple of cells $\{R_k\}_{k=1}^K$ defines such as

$$R_k = \left\{ x \in \mathbb{R}^n \mid \|x - A(\nu_{x,k})^{-1}\mu_k\|_{A(\nu_{x,k})^T A(\nu_{x,k})} \leq \|x - A(\nu_{x,j})^{-1}\mu_j\|_{A(\nu_{x,j})^T A(\nu_{x,j})}, \quad \forall j \neq k \right\},$$

defines a weighted Voronoi diagram [162, 163], where we observe that the metric tensor is dependant on all the spatial transformations.

While the Euclidean K -means induces a Voronoi diagram where each region is a polytope, the ST K -means does not impose such a constraint of its geometry. The similarity measure we propose adapts the geometry of each data to each centroid and thus induces a specific metric space for each data-centroid pair. In particular, for each data-centroid pair, the ST K -means has a particular metric that induces the boundary of the regions. In a more general setting, each region is defined as the orbit of the centroid with respect to the transformations induced by the spatial transformer, thus defining regions that depend on the orbit's shape instead of polytopal ones.

This geometric observation can lead to efficient initializations for the ST K -means [164], as well as the evaluation of its optimality [165]. Besides, one can perform in depth study to understand the shape of the regions spanned by our approach to understand the fail cases of the algorithm for a particular application [166, 167]. One can also compare the partitioning achieved in our approach with the one of DNN as in [168] to gain more insights into both models.

v Complexity & Parameters The time complexity of ST K -means is $O(NK(\ell^3 + \ell n))$. In fact, the ST K -means computes a TPS of computational complexity $O(\ell^3 + \ell n)$ for each sample of the N samples and each of the K centroids, as in Eq. 5.2. In practice, ℓ is of the order 2^6 . The number of parameters of the model is $2\ell \times N \times K$; it depends on the number of samples, clusters, and landmarks.

To speed up the computation, we (i) pre-compute the matrix inverse responsible

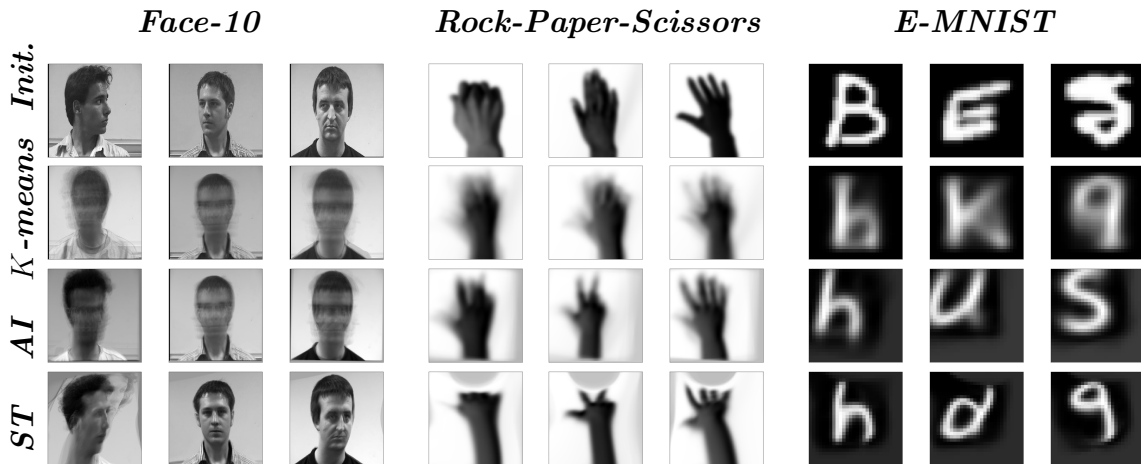


Figure 5.3 : **Centroids** - We depict some centroids for the different K -means algorithms. The centroid at initialization are displayed in the *1st* row. The centroids learned by K -means are shown in the *2nd* row, by the Affine invariant K -means in the *3rd* row, and by our ST K -means in the *4th* row. By comparing the results of the AI K -means (*3rd* row) with the standard K -means (*2nd* row), we can see that using only affine transformations slightly improves the K -means centroids and reduces the superposition issue that K -means suffers from. By comparing the results of our ST K -means (*4th* row) with the other methods, it is clear that using non-rigid transformations significantly improves the quality of the centroids, making them sharper and removing the issue related to the non-additiveness of images. Note that K -means iteratively updates the centroids and cluster assignments, as such, the class associated to a specific centroid usually changes during training.

for the dominating cubic term, see Sec. 2.4 for implementation details regarding the TPS, and (ii) implement ST K -means on GPU with SymJAX [169] where high parallelization renders the practical computation time near constant with respect to the number of landmarks as we depict in Fig. 5.5.

5.7 Experiments

In this section, we detail the experimental settings followed to evaluate the performances of our model. For all the experiments, the number of clusters is set to be the number of classes the dataset contains for all clustering algorithms.

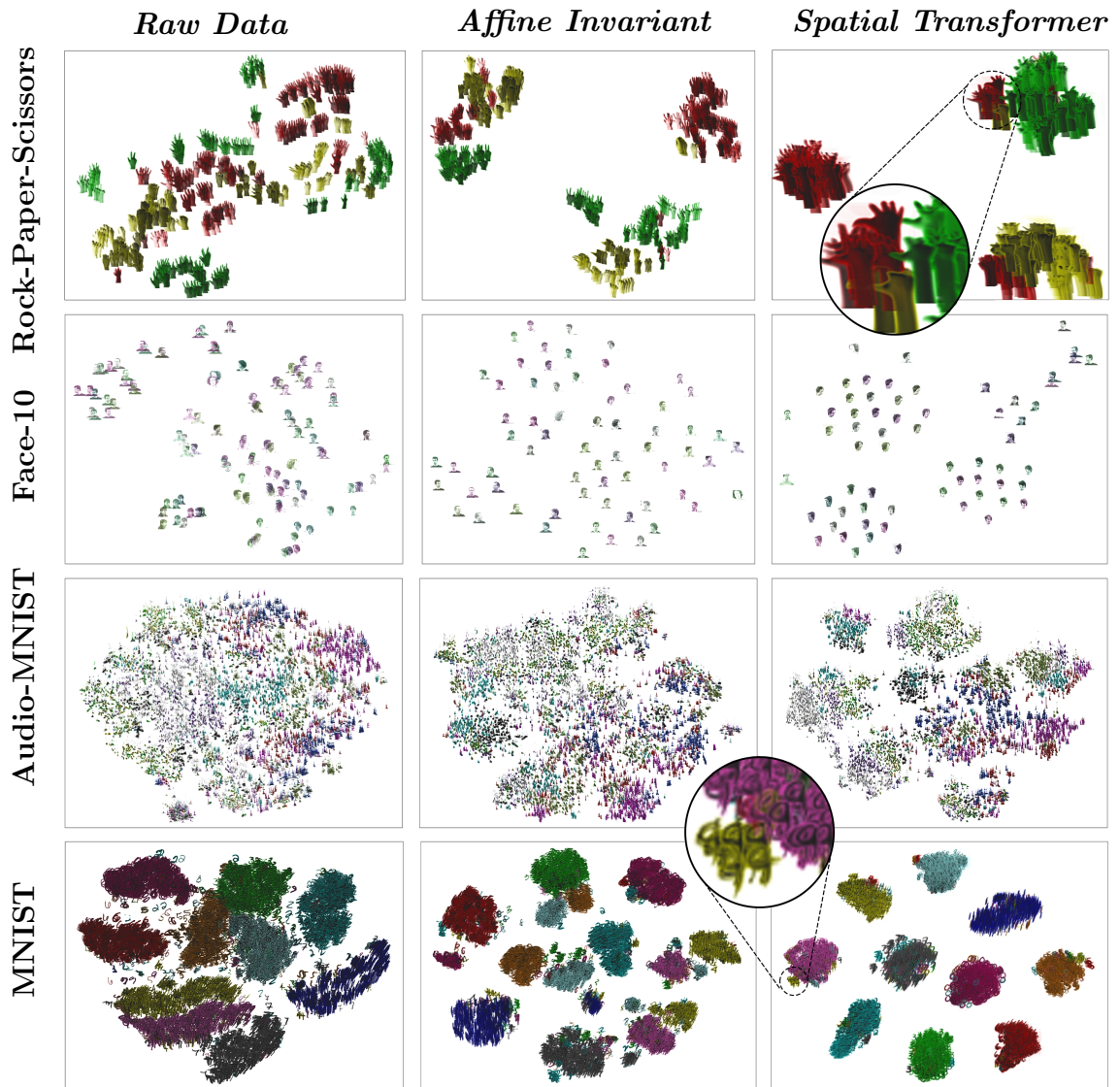


Figure 5.4 : t-SNE projections

2-dimensional t-SNE - (# denotes the number of clusters) - We suggest the reader to zoom in the plots to best appreciate the visualizations. - The raw data (*left column*), the affinely transformed data using the AI distance, i.e., we extract the best affine transformation of the data that corresponds to the centroid it was assigned and perform the t-SNE on these affinely transformed data, (*middle column*), the data transformed with respect to the TPS as per Eq. 5.2, i.e., the same process as previously mentioned but we consider the spatial transformer instead, and then perform the dimension reduction on these transformed data, (*right column*).

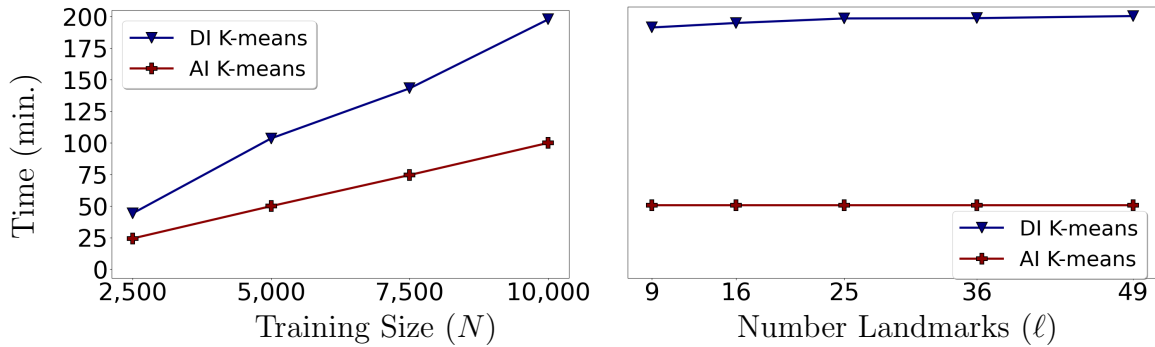


Figure 5.5 : Computational time

Computational Training Time - Comparison between our ST K -means and the Affine Invariant (AI) K -means computational times on the Arabic Characters dataset. The input pixel size is $n = 1024$. (*Left*) shows the computational time for varying training set sizes and $\ell = 7^2$. (*Right*) shows the computational time as a function of the number of landmarks, ℓ , for $N = 10,000$. Since the AI K -means does not use the TPS algorithm, its computational time is constant as a function of the number of landmarks. We can observe that our process to speed up the computation enables the tractability of the ST K -means.

i Evaluation Methods For all the experiments, the accuracy is calculated using the metric proposed in [170] and defined as

$$\text{Accuracy} = \max_m \frac{1}{N} \sum_{i=1}^N 1_{\{l_i=m(\hat{l}_i)\}} \quad , \quad (5.9)$$

where l_i is the ground-truth label, \hat{l}_i the cluster assignment and m all the possible one-to-one mappings between clusters and labels. The results in Table 5.1 are taken as the best score on the test set based on the ground truth labels among 10 runs as in [123]. We also provide on the same run the normalized mutual information (NMI) [171], and adjusted rand index (ARI) [172].

ii Cross-validation Settings Our model requires the cross-validation of hyper-parameters: the number of landmarks and the learning rate to learn the similarity measure in Eq. 5.2. However, the clustering framework does not allow the use of label information to perform the cross-validation of the parameters. We thus need to find a proxy for it to determine the optimal model parameters. Interestingly, the distortion error related used in the ST K -means, Eq. 5.1, appears to be negatively correlated to the accuracy, as displayed in Fig. 5.6. Note that the use of the distortion error is commonly used as a fitness measure in K -means, for example, when cross-validating the number of clusters.

We cross-validate the number of landmarks, ℓ , which defines the resolution of the transformation, which we optimize over the following grid, $[3^2, 4^2, 5^2, 6^2, 7^2, 8^2]$. Then, the learning of the landmarks, ν , is done via Adam optimizer. The learning rate is picked according to $[10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}]$. We train our method for 150 epochs for all the datasets, with batches of size 64. As for K -means and AI K -means, the centroids' initialization of the ST K -means is performed by the K -means++ algorithm. Importantly, the same procedure is applied to all datasets.

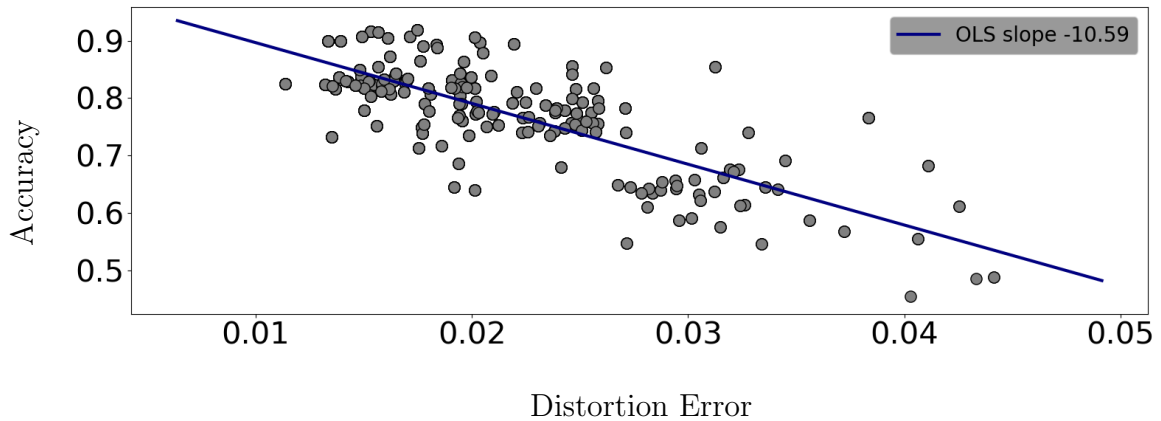


Figure 5.6 : Accuracy vs distortion error

Accuracy vs Distortion Error - Clustering accuracy, Eq. 5.9, of ST K -means algorithm on the MNIST dataset as a function of the distortion error, Eq. 5.1, using the similarity measure, Eq. 5.2. Each gray dot is associated with a specific set of hyper-parameters, e.g., the learning rate and the number of landmarks for the spatial transformer. The accuracy is negatively correlated to the distortion error (see the blue line corresponding to the ordinary least square fit), indicating that the distortion error is an appropriate metric to cross-validate the hyper-parameters of the ST K -means algorithm, which is crucial in an unsupervised setting as the labels are not available.

Note that during the training, both the similarity measure in Eq. 5.2 and the clustering update are performed, Eq. 5.9. During the algorithm’s testing phase, the centroids remain fixed, and only the similarity measure is performed to assign each testing datum to a cluster.

iii Results We report in Table 5.1 the accuracy of the different models considered on the different datasets. Our approach shows to outperform existing models on most datasets. Our model equals the performance of AI K -means on Affine MNIST and is

only outperformed by VaDE (MLP) on MNIST.

Whereas the various deep learning approaches perform well on datasets for which their architectures were developed, e.g., MNIST and its derivatives: E-MNIST, Arabic Characters, they show limited performance on higher resolution datasets with a small number of samples, such as Rock-Paper-Scissors, Face-10 as well as the two toy examples. In fact, they are composed of only 700 training data and 300 testing data. In the following sections, we interpret various visualizations of the K -means variants used in this work.

iv Interpretability: Centroids Visualization We propose in Fig. 5.3 to visualize the centroids obtained via K -means, AI K -means, and our ST K -means. For each dataset, the first row shows the clusters after initialization from K -means++. The three following rows show the centroids obtained via the K -means, AI K -means, and ST K -means algorithms, respectively.

We observe that, for all datasets, the K -means centroids are not lying on the data manifold as they are unrealistic images that could not occur naturally in the dataset. Besides, they appear to be blurry and hardly interpretable. These drawbacks are due to the update rule that consists in the average of the data belonging to each cluster in the pixel space. The AI K -means algorithm drastically reduces the centroids' blurriness induced by such an averaging as it considers the average of affinely transformed data. However, our ST K -means produces the crispest centroids and does not introduce any ambiguity in between the different clusters. In fact, the update of our method, Eq. 5.4, takes into account the non-linear structure of the manifold by taking the average over data transformed using a non-rigid transformation.

Interestingly, Fig. 5.3 shows that even if at initialization multiple centroids assigned

to the same class are attributed to different clusters, the ST K -means is able to recover this poor initialization thanks to its explicit manifold modeling and centroid averaging technique. For instance, in the Rock-Paper-Scissors dataset, although at initialization, two centroids correspond to the class paper, the ST K -means learns centroids of each of the three classes within this dataset. In the Face-10 dataset, some centroids learned correspond to the rotation of the initialization; even in such extreme change of pose, the centroids remain crisp in most cases.

v Interpretability: Embedding Visualization To get further insights into the disentangling capability of the ST K -means, we compare the 2-dimensional projections of the data using t-SNE [174], of the K -means, AI K -means and ST K -means.

The t-SNE visualizations, for both the AI and ST K -means, are obtained by extracting the optimal transformation that led to the assignment. Precisely, for each image x_i , we compute $l = \arg \min_k d(x_i, \mu_k)$ and extract the optimal parameter $\nu_{i,l}^*$ which is then used to obtain the transformed image fed as the input of the t-SNE.

We can observe in Fig. 5.4 that across datasets, both the affine transformations learned on the data and the non-rigid transformations help to define more localized clusters. One can observe that for the Face-10 dataset, while the dataset contains 13 clusters, we can see that the ST K -means induced transformations lead to a 2-dimensional space where the faces are clustered 3 majors orientations. The top left cluster corresponds to faces pointing left, the bottom one face pointing right, and the bottom right one face pointing front. We also propose to zoom-in two locations where the ambiguity in the transformation induced by the spatial transformer is noticeable. In particular, we show two cases where the non-rigid transformations are too large for certain samples leading to an erroneous clustering assignment, e.g., in the MNIST dataset, the yellow samples in the lense are initial instance of the class 4

that have been transformed into digit that geometrically resemble the centroid of the cluster 9, thus being assigned to the 9's cluster. The same concept is shown in the Rock-Paper-Scissors lense where some instance of the classes rock and paper are assigned to the class scissors.

Overall, the ST K -means drastically enhances the separability of the different clusters. When using ST K -means, the data are clustered based on macroscopically meaningful and interpretable parameters, making the model's performance possible to understand. For instance, for the Face-10 dataset, the t-SNE representation of the ST K -means clusters' shows that faces are grouped according to three significant orientations, left, right, and front. These three clusters are more easily observed in our ST K -means than in the affine invariant model. However, the 13 different orientations present in the dataset remain too subtle to be captured by the ST K -means.

For the MNIST dataset, the last row and column of Fig. 5.4, we observe that most of the incorrectly clustered images are almost indistinguishable from samples of the cluster they have been attributed. In particular, we highlight this by proposing to zoom-in into the cluster of hand-written 9 in Fig. 5.4. We can see that the yellow instances are samples from the class 4 that have been transformed such that they resemble the 9's centroid in Fig. 5.3. We also provide a zoom-in on one of the clusters obtained on the rock-paper-scissors dataset, first row and last column of Fig. 5.4. The incorrectly clustered data are the ones that, when transformed, easily fit the scissors shape.

5.8 Conclusion

Designing an unsupervised algorithm that is robust to non-rigid transformations remains challenging, despite the tremendous breakthrough in machine learning. The

problem lies in appropriately limiting the size of the transformations. We showed that the spatial transformer could achieve this as the number of landmarks allows the learnability of a coarse to fine grid of transformation. However, such a parameter controlling the size of the transformation should be designed as well as be learned per-cluster or per-sample. Besides this difficulty, we showed that we could conserve the interpretability of the K -means algorithm applied in the input data space while drastically improving its performances. Such a framework should be favored in clustering applications where the explainability of the decision is critical.

Table 5.1 : Clustering accuracy

Clustering results in % of the test set accuracy Eq. 5.9 - Following the benchmarks evaluation method, the best accuracy (ACC) over 10 runs are displayed - We also provide the associated normalized mutual information (NMI) - the number of clusters is denoted by # next to the dataset name and where (†): [123] and (‡): [173].

	<i>Deep Learning</i>	Aff. MNIST #10	Diffeo. MNIST #10	MNIST #10	Audio MNIST #10	E-MNIST #26	Rock-Paper-Sci. #3	Face-10 #13	Arabic Char. #28	Aff. MNIST #10	Diffeo. MNIST #10	MNIST #10	Audio MNIST #10	E-MNIST #26	Rock-Paper-Sci. #3	Face-10 #13	Arabic Char. #28
		ACC									NMI						
<i>K</i> -means	✗	68	61	53	10	39	40	20	19	-	-	50	1	39	5	18	27
AI <i>K</i> -means	✗	100	91	75	29	48	72	31	30	-	-	62	18	45	30	30	37
ST <i>K</i> -means	✗	100	99	92	41	65	86	45	51	-	-	82	26	63	63	53	61
AE + <i>K</i> -means	✓	72	60	66	13	41	48	37	23	-	-	64	1	40	9	27	33
DEC (MLP) (†)	✓	84	77	84	10	55	46	33	24	-	-	83	1	51	12	20	32
DEC (Conv)	✓	70	68	78	15	60	54	38	29	-	-	74	3	56	18	31	39
VaDE (MLP) (‡)	✓	68	65	94	11	20	50	36	26	-	-	89	1	12	16	27	30
VaDE (Conv)	✓	65	59	81	14	58	55	40	46	-	-	78	2	55	20	35	53

6

Learnable Lie Group for Manifold Approximation

Autoencoders (AEs) provide a rich and versatile framework that discovers the data’s salient features in an unsupervised manner. They are commonly leveraged to efficiently perform compression [175], denoising [176], data completion [177], as well as pre-training supervised DNs [178]. Solving these tasks is equivalent to discovering the data’s underlying manifold, a task becoming challenging in the high dimensional and the finite samples regime [179–182]. To overcome these challenges and improve the efficiency of AEs, various explicit or implicit regularizations have been proposed [183–186]. Despite these improvements, the underlying mechanisms and generalization capability of AEs are still poorly understood [187–189].

A compelling approach to understanding the inner mechanisms of DNs considers their capability at modeling the ubiquitous symmetries in the [106, 190]. Theoretically grounded data models such as the Deep Scattering Network and its derivatives have been derived in accordance with this principle [67, 191–193]. In [59, 194, 195]

they propose to explain the success of deep convolutional architectures through the development of an equivariance theory of DNs; in particular, they provide (i) an understanding and formalism behind the equivariance properties of DNs as well as their generalization, and (ii) reduce the sample complexity of DNs by exploiting well-known symmetry group inherent to the image manifold.

Besides explicitly imposing specific group of transformations, the studies of DNs through that lens mainly consider the properties of internal layers of DNs, e.g., *convolution, pooling, per-layer representation*. In this paper, we propose a global analysis by considering the DN from a geometrical standpoint. By global analysis, we consider the understanding of the output of a DN given its input in an end-to-end manner. Such analysis is presently performed by leveraging the analytical continuous piecewise affine (CPA) map formulation of DNs, as described in [196]. Such an approach has two significant advantages; it is agnostic of the architecture, e.g., type of layer, nonlinearities, number of layers, and it provides an analytical formula for the entire network mapping. These criteria are crucial since the understanding of AEs performed in this work has the goal of developing practical tools that are not tied to any specific AE architecture.

In the present work, the CPA formulation is leveraged to take a step into answering the following questions: (i) How an AE can effectively approximate a data manifold? (ii) How can one improve and guarantee the generalization of AEs exploiting the symmetry in the data?

We will execute this by considering the following two-fold approach: First, we

provide an analytical and interpretable formulation of the CPA representation of the manifold spanned by AEs. We make explicit some critical properties of AEs such as what type of function do they belong to, how standard regularization techniques affect the AE mapping, and how the encoder and decoder per region affine mappings are related. Second, we exploit the developed understanding of AEs to provide novel regularizations for AEs that capture the symmetry in the data. In particular, our regularizations constrain the global CPA surface spanned by AEs such that they adapt to the geometry of the data manifold modeled from as the orbit of a Lie group. We show that these regularizations constrain the entire surface even at locations in the manifold where data are missing, which is critical for the generalization of AEs. Besides, we show that these regularizations lead to generalization guarantees in the finite data regime.

6.1 Outline of the Chapter and Contribution Summary

- We demonstrate that AEs provide a CPA approximation of the data manifold. From this analytical characterization, we interpret the role of the encoder, decoder, layer parameters, and latent dimension (Sec. 6.3).
- Following these findings, we obtain reconstruction guarantees (Sec. 6.4 - i), interpretable formulas for the Jacobian and approximated Hessian of AEs (Sec. 6.4 - ii); and leverage them to provide insights into standard regularization techniques employed in AEs (Sec. 6.4 - iii).
- We demonstrate that when considering the symmetry of the data, we can impose constraints on an interpolation function, e.g., an AE, such that it approximates the data manifold driven by a Lie group (Sec. 6.4 - iv).

- We turn these constraints into regularizations adapted to AEs (Sec. 6.4 - v) and demonstrate their generalization guarantees under a finite data regime (Sec. 6.4 - vi).
- We provide experimental validations and computational complexity of the developed regularizations which compete with state-of-the-art methods on various datasets (Sec. 6.5).

6.2 Related Work

A compelling approach to understanding the inner mechanisms of DNs considers their capability at modeling the ubiquitous symmetries in the [106, 190]. Theoretically grounded data models such as the Deep Scattering Network and its derivatives have been derived in accordance with this principle [67, 191–193]. In [59, 194, 195] they propose to explain the success of deep convolutional architectures through the development of an equivariance theory of DNs; in particular, they provide (i) an understanding and formalism behind the equivariance properties of DNs as well as their generalization, and (ii) reduce the sample complexity of DNs by exploiting well-known symmetry group inherent to the image manifold.

In this work, we focus on autoencoders (AE), which aim at learning an identity mapping, also known as auto-association [197], on a given dataset with a bottleneck latent dimension. It has been implemented first for image compression [198], speech recognition [199], and dimensionality reduction [200]. It is composed of two nonlinear maps: an encoder, denoted by \mathbf{E} and a decoder, denoted by \mathbf{D} . The encoder maps an input $x \in \mathbb{R}^d$ to a hidden layer of dimension $h < d$, $\mathbf{E}(\mathbf{x})$, which encodes the salient features in the data [27] and defines its *code* or embedding. The decoder reconstructs the input from its *code*, thus the entire AE map is defined as $(\mathbf{D} \circ \mathbf{E})(\mathbf{x})$ with \circ

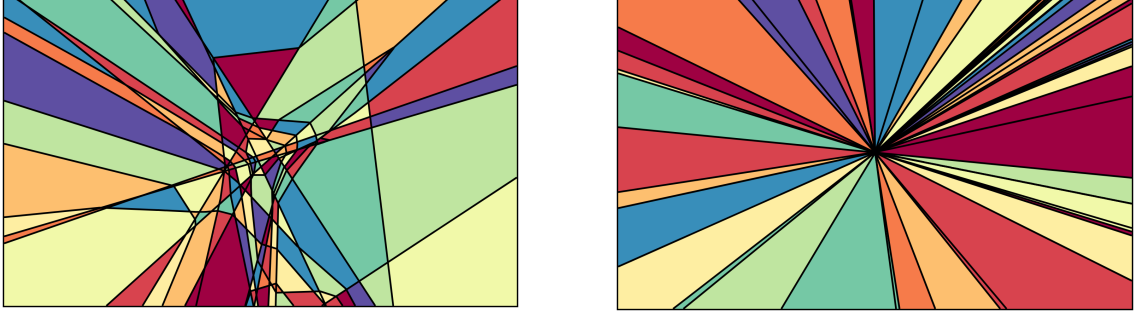


Figure 6.1 : 2-dimensional visualizations of the input space partitioning $\Omega^{E,D}$ induced by two randomly initialized AEs with bias (*left*) and zero bias (*right*). Each region, depicted by a particular color, bounded by the black lines has a set of CPA parameters $A_\omega^E, A_\omega^D, B_\omega^E, B_\omega^D$ described in Eq. 6.3 which depend on the per-layer affine parameters as well as the state of the nonlinearities of the region ω . To reconstruct its input, an AE achieves an affine map for each region; its output for a sample of a given region ω is provided by Eq. 6.2.

denoting the composition operator.

The weights of the AE are learned based on some flavors of reconstruction losses, e.g., the mean-square error for real data and the binary cross-entropy for binary data, between the output, $(\mathbf{D} \circ \mathbf{E})(\mathbf{x})$, and the input, \mathbf{x} . To improve generalization, some regularizations can complement the reconstruction loss [201] such as favoring sparsity of the *code* [185] or sparsity of the weights [202]. Other types of regularization include injecting noise in the input leading to Denoising AE known to increase the robustness to small input perturbations [183]. Closer to our work, [203] and [184] proposed to improve the robustness of the *code* to small input perturbations by penalizing the curvature of the encoder mapping by regularizing the Jacobian as well as the Hessian of \mathbf{E} .

6.3 Formalism

A DN is an operator \mathbf{f}_Θ with parameters Θ composing L intermediate *layer* mappings \mathbf{f}_ℓ , $\ell = 1, \dots, L$, that combine affine and simple nonlinear operators such as the *fully connected operator*, *convolution operator*, *activation operator* (applying a scalar nonlinearity such as the ubiquitous ReLU), or *pooling operator*.

A DN employing nonlinearities such as (leaky-)ReLU, absolute value, and max-pooling is a continuous piecewise linear operator and thus lives on a partition Ω of the input space. As such, the DN's CPA mapping of an input \mathbf{x} can be written as

$$\mathbf{f}_\Theta(\mathbf{x}) = \sum_{\omega \in \Omega} 1_{\{\mathbf{x} \in \omega\}} (A_\omega \mathbf{x} + B_\omega) \quad (6.1)$$

where 1 defines the indicator function, A_ω and B_ω the per region affine parameters involving the DN per layer affine parameters, $W^\ell, \mathbf{b}^\ell \in \Theta, \forall \ell$, and the nonlinearities state of the region $\omega \in \Omega$ [204]. The unit and layer input space partitioning can be rewritten as Power Diagrams, a generalization of Voronoi Diagrams [205]; composing layers produce a Power Diagram subdivision.

The output of a CPA DN is formed as per Eq. 6.1. An AE composing two CPA functions, the *encoder* and the *decoder*, the entire mapping remains a CPA with an input space partition and per region affine mappings. Because we can consider an AE as a network or as the composition of two networks, we will consider two different space partitioning. The partition of the input, i.e., data space, induced by the entire AE, and denoted by $\Omega^{E,D}$, as well as the partition of the decoder induced in the latent space, i.e., bottleneck layer, and denoted by Ω^D . Examples of the entire AE partitioning, i.e., $\Omega^{E,D}$, can be visualized in Fig. 6.1.

Now, let $\omega \in \Omega^{E,D}$ defines a region induced by the AE partitioning in the input

space as aforementioned. Given a d -dimensional sample $\mathbf{x} \in \omega$, the max affine spline formulation of the AE mapping is defined as

$$\mathbf{D} \circ \mathbf{E}(\mathbf{x}) = A_\omega^D A_\omega^E \mathbf{x} + A_\omega^D B_\omega^E + B_\omega^D, \quad (6.2)$$

where \circ is the composition operator, $A_\omega^D \in \mathbb{R}^{d \times h}$, $A_\omega^E \in \mathbb{R}^{h \times d}$, $B_\omega^E \in \mathbb{R}^h$ and $B_\omega^D \in \mathbb{R}^d$ with d being the dimension of the input space and h the bottleneck dimension.

The mapping from these global parameters to the per-layer ones is performed as follows. First, we denote by $W^\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$, $\mathbf{b}^\ell \in \mathbb{R}^{d_\ell}$ the affine parameters of each layer, where $\ell \in \{1, \dots, L\}$ defines the encoder indexes and $\ell \in \{L+1, \dots, L+P\}$ the decoder ones (with structure depending on the layer type), where L denotes the number of encoder layers, P the number of decoder layers, $d_{\ell-1}$ the input dimension of the layer ℓ and d_ℓ its output dimension. We have that $d_L = h$ the bottleneck dimension, $d_0 = d_{L+P} = d$ the input and output dimension. Then, we also denote by Q^ℓ the diagonal matrices encoding the region induced states of the nonlinearities, $(0, 1)$ for ReLU, $(-1, 1)$ for absolute value. Finally, the parameters of the max affine spline AE formulation described in Eq. 6.2 are defined as

$$A_\omega^E = W^L Q_\omega^{L-1} W^{L-1} \dots Q_\omega^1 W^1 \quad \text{and} \quad B_\omega^E = \mathbf{b}^L + \sum_{i=1}^{L-1} W^L Q_\omega^{L-1} W^{L-1} \dots Q_\omega^i \mathbf{b}^i. \quad (6.3)$$

A_ω^D and A_ω^D are defined similarly with $\ell \in \{L+1, \dots, L+P\}$. Therefore, there is a direct mapping from the intuitive piecewise affine parameterization of the network to the per-layer parametrization as it is commonly used in the literature.

Given these analytical maps, we now provide insights into the AE approximation.

Let's rewrite Eq. 6.2 as

$$\mathbf{D} \circ \mathbf{E}(\mathbf{x}) = \sum_{k=1}^h \left\langle \mathbf{a}_k^{E^T}[\omega], \mathbf{x} \right\rangle \mathbf{a}_k^D[\omega] + B_\omega^{E,D} = A_\omega^D \boldsymbol{\mu}_\mathbf{x} + B_\omega^{E,D}, \quad (6.4)$$

where $B_\omega^{E,D} = A_\omega^D B_\omega^E + B_\omega^D$, $\mathbf{a}_k^{E^T}[\omega]$ are the rows of A_ω^E , $\mathbf{a}_k^D[\omega]$ are the columns of A_ω^D . This is the shifted mapping of \mathbf{x} onto the subspace spanned by A_ω^D and with

coordinates driven by A_ω^E .

From Eq. 6.4, we deduce the per region role of the encoder and decoder. The samples of each region $\omega \in \Omega^{E,D}$, are expressed in the basis defined by the decoder region-dependent parameter A_ω^D , i.e., the per region parametric representation of the approximated manifold, and the coordinates of this sample in such a basis are induced by the region-dependent parameter A_ω^E , the whole mapping is then shifted according to both the encoder and decoder CPA parameters.

6.4 Properties & Geometrical Aspects

i Reconstruction Guarantees We now derive a necessary condition on the CPA parameters, A_ω^D, A_ω^E , such that the AE achieves perfect reconstruction on a given continuous piecewise linear surface in the case of zero bias as often used in practice [206].

Proposition 8. *A necessary condition for the zero-bias AE to reconstruct a continuous piecewise linear data surface is to be bi-orthogonal as per $\forall \mathbf{x} \in \omega$, $\mathbf{D} \circ \mathbf{E}(\mathbf{x}) = \mathbf{x} \implies \langle \mathbf{a}_k^D[\omega], \mathbf{a}_{k'}^E[\omega] \rangle = 1_{\{k=k'\}}$, where \mathcal{X} denotes the data surface.*

Proof. Perfect reconstruction $\implies: \forall \omega, \forall x \in \omega$, $x = \sum_{k=1}^h \langle x, \mathbf{a}_k^E \rangle \mathbf{a}_k^D$. We have $\forall \omega, \forall x \in \omega$

$$\begin{aligned} \sum_k \langle x, \mathbf{a}_k^E[\omega] \rangle \mathbf{a}_k^D[\omega] &= \sum_{k=1}^h \left\langle \sum_{k'=1}^h \langle x, \mathbf{a}_{k'}^E[\omega] \rangle \mathbf{a}_{k'}^D[\omega], \mathbf{a}_k^E[\omega] \right\rangle \mathbf{a}_k^D[\omega] \\ &= \sum_k \sum_{k'=1}^h \langle x, \mathbf{a}_{k'}^E[\omega] \rangle \langle \mathbf{a}_{k'}^D[\omega], \mathbf{a}_k^E[\omega] \rangle \mathbf{a}_k^D[\omega] \end{aligned}$$

$\iff A_\omega^D A_\omega^E x = A_\omega^D A_\omega^{D^T} A_\omega^{E^T} A_\omega^E x$ since $A_\omega^D A_\omega^E$ is injective on the region (as per perfect reconstruction condition) it implies that $A_\omega^{D^T} A_\omega^{E^T} = I_h$, where I_h is the identity matrix of dimension $h \times h$ □

That is, if a continuous piecewise linear surface is correctly approximated, we know

that the parameters of the MAS operator describing the encoder and decoder will be bi-orthogonal, i.e., the column vectors of A_ω^D and the row vectors of A_ω^E form a bi-orthogonal basis.

We now propose to give intuitions regarding this condition by utilizing the mapping between CPA parameters and layer weights as per Eq. 6.3. In fact, the following corollary provides the conditions for the bi-orthogonality to be fulfilled depending on the weights of the autoencoder, i.e., W^ℓ . For the sake of clarity, we consider the case of a 2-layer ReLU AE.

Corollary 4. *Let \mathbf{E} and \mathbf{D} be a 2-layer ReLU network with respective weights $W^1 \in \mathbb{R}^{h \times n}$ and $W^2 \in \mathbb{R}^{n \times h}$, as per Eq. 6.3. We denote by $W_{i,j}^1$ the i^{th} row and j^{th} column of the weight matrix W^1 . Now, $\forall x \in \mathcal{X}$, a necessary condition for bi-orthogonality is that, for each $k, k' \in \{1, \dots, h\}$, one of the following is fulfilled:*

- (i) $W_{k',.}^{1T} x \leq 0$.
- (ii) $\forall i \in \{1, \dots, d\}, W_{i,.}^{2T} \mathbf{E}(x) \leq 0$.
- (iii) $\forall i \in \{1, \dots, d\}, W_{i,.}^{2T} \mathbf{E}(x) > 0$ and $\langle W_{.,k}^2, W_{k',.}^1 \rangle = 0$.
- (iv) $\sum_{i=1}^d W_{i,k}^2 W_{k',i}^1 1_{\{W_{i,.}^{2T} \mathbf{E}(x) > 0\}} = 0$.

Proof. For a 2-layers ReLU autoencoder network, we have the following affine spline parameters $\forall x \in \omega$:

$$a_{k'}^E[\omega] = 1_{\{W_{k',.}^{1T} x > 0\}} W_{k',.}^1,$$

$$a_k^D[\omega] = \begin{pmatrix} 1_{\{W_{1,.}^{2T} z > 0\}} \\ \vdots \\ 1_{\{W_{d,.}^{2T} z > 0\}} \end{pmatrix} \cdot W_{.,k}^2$$

where \cdot defines here the elementwise vector multiplication. Now,

$$\begin{aligned}
\langle a_k^D[\omega], a_{k'}^E[\omega] \rangle &= \left\langle Q_\omega^2 W_{.,k}^2, 1_{\{W_{k',.}^{1T} x > 0\}} W_{k',.}^1 \right\rangle \\
&= 1_{\{W_{k',.}^{1T} x > 0\}} W_{.,k}^{2T} Q_\omega^2 W_{k',.}^1 \\
&= 1_{\{W_{k',.}^{1T} x > 0\}} W_{.,k}^{2T} \begin{pmatrix} 1_{\{W_{1,.}^{2T} \mathbf{E}(x) > 0\}} W_{k',1}^1 \\ \vdots \\ 1_{\{W_{n,.}^{2T} \mathbf{E}(x) > 0\}} W_{k',d}^1 \end{pmatrix} \\
&= 1_{\{W_{k',.}^{1T} x > 0\}} \left(\sum_{i=1}^d W_{i,k}^2 W_{k',i}^1 1_{\{W_{i,.}^{2T} \mathbf{E}(x) > 0\}} \right)
\end{aligned}$$

□

This results shows that the bi-orthogonality condition can be obtained via a combination of orthogonality conditions between the weights and/or nonlinearity activations.

For instance, the proposition (i) corresponds to the case where the input of the k' unit in the bottleneck layer is negative, condition (ii) is the case where the input of all output units is negative, condition (iii) corresponds to a linear decoder and orthogonality of the weights, and (iv) corresponds to an orthogonality condition between the k^{th} column of the decoder weight with the k^{th} row of the encoder weight modulo the activations of the decoder layer. Note that if (ii) and (iii) hold for multiple regions $\omega \in \Omega^{E,D}$ it implies that the decoder is linear with respect to the coordinate space and forms a linear manifold. Thus, these are not realistic conditions to have efficient AEs.

ii Tangents and Hessians From the CPA formulation, we observed that for each region $\omega \in \Omega^{E,D}$, $\mathbf{D} \circ \mathbf{E}$ defines a composition of two continuous piecewise affine functions, each defined respectively by the parameters A_ω^E , B_ω^E , and A_ω^D , B_ω^D . We can

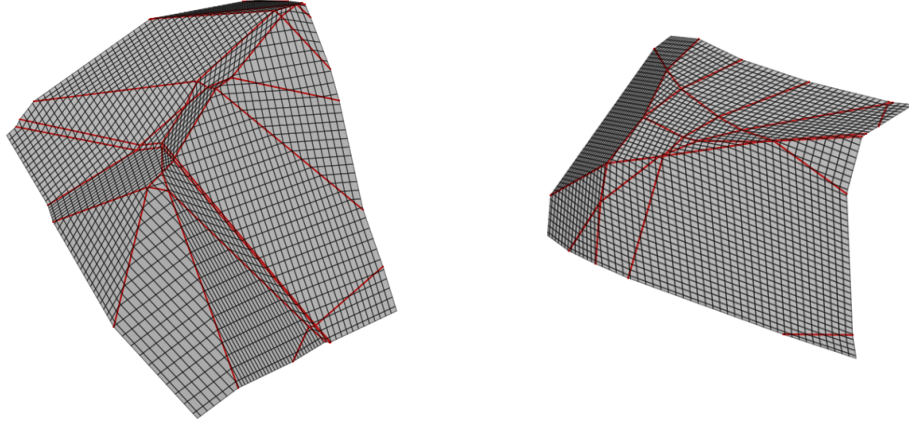


Figure 6.2 : Piecewise linear surfaces induced by two randomly initialized AE decoders and visualized in the ambient space of dimension $d = 3$ (latent dimension being $h = 2$). The gray denotes the regions, and the red lines their borders. As they correspond to the MAS surface induced by the decoder, each gray region has a slope characterized by the Jacobien of the decoder as in Eq. 6.8. Our work aims at developing a constraint on these surfaces via their per region tangent, such that they approximate the manifold defined by the orbit of a signal with respect to the action of a group.

thus derive simple analytical formulas for the per region Jacobian and approximated Hessian of the AE.

The Jacobian of the AE for a given region $\omega \in \Omega^{E,D}$ is given by

$$J_\omega[\mathbf{D} \circ \mathbf{E}] = A_\omega^D A_\omega^E. \quad (6.5)$$

In fact, let $[\mathbf{D} \circ \mathbf{E}(\cdot)]_i : \mathbb{R}^d \rightarrow \mathbb{R}$ be the i^{th} coordinate output of the AE, defined as $[\mathbf{D} \circ \mathbf{E}(x)]_i = [A_\omega^D]_{i,\cdot} A_\omega^E x + [A_\omega^D]_{i,\cdot} B_\omega^E + [B_\omega^D]_i$.

$$d[\mathbf{D} \circ \mathbf{E}(\cdot)]_i = [\mathbf{D} \circ \mathbf{E}(x + \epsilon)]_i - [(\mathbf{D} \circ \mathbf{E})(x)]_i = \left\langle A_\omega^{E^T} [A_\omega^D]_{i,\cdot}^T, \epsilon \right\rangle, \forall \epsilon \in \mathbb{R}^d. \quad (6.6)$$

As such, we directly obtain that

$$\nabla_x [\mathbf{D} \circ \mathbf{E}(\cdot)]_i = A_\omega^{E^T} [A_\omega^D]_{i,\cdot}^T, \quad (6.7)$$

which leads to Eq. 6.5.

It is also clear that the rank of the Jacobian is upper bounded by the latent dimension as $\text{rank}(J_\omega[\mathbf{D} \circ \mathbf{E}]) \leq h$, where h is the number of units of the bottleneck layer of the AE, and in general by the $\min_\ell d_\ell$. This dimension is directly related to the manifold’s dimension that one aims to approximate, assuming that all other layer widths are larger than h .

One can similarly obtain the per region tangent of the decoder, as it defines the per region parametric representation of the manifold, see Fig. 6.2. We recall that we denote by Ω^D the partition of the latent space induced by the decoder

$$\forall \omega \in \Omega^D, J_\omega[\mathbf{D}] = A_\omega^D, \quad (6.8)$$

where the columns of A_ω^D form the basis of the tangent space induced by \mathbf{D} .

The characterization of the curvature of the approximation of the data manifold can be done using the per region Hessian defined by $H_\omega, \forall \omega \in \Omega^D$, which in our case will be defined as the sum of the difference of neighboring tangent planes.

$$\forall \omega \in \Omega^D, \|H_\omega\|_F = \sum_{\omega' \in \mathcal{N}(\omega)} \|J_\omega[\mathbf{D}] - J_{\omega'}[\mathbf{D}]\|_F, \quad (6.9)$$

where $\mathcal{N}(\omega)$ denotes the set of neighbors of region ω and $\|\cdot\|_F$ is the Frobenius norm. This approach is based on the derivation described in [203]. In practice, we use a stochastic approximation of the sum by generating a small mini-batch of a few corrupted samples which induce neighboring regions.

iii Interpretability of Regularization Techniques We are now interested in leveraging these findings to analyze and interpret common AE regularizations.

- (i) **Higher-Order Contractive AE** [203]: This regularization penalizes the energy of the first and approximated second derivative the encoder map for any region containing a training sample, i.e., $\|A_\omega^E\|_F$ and $\sum_{\omega' \in \mathcal{N}(\omega)} \|A_\omega^E - A_{\omega'}^E\|_F$. In the case of a ReLU AE, we know from Eq. 6.2 and the submultiplicativity

of the Frobenius norm that the norm of the Jacobian is upper-bounded by $\|W^L\|_F \times \dots \times \|W^1\|_F$. Therefore adding a weight-decay penalty on the encoder weights induces the first-order contractive AE. The second-order induces the curvature of the piecewise linear map A^E to be small. Note that it is the per-region affine map induced by the encoder that is regularized, and that it depends on the region's activation codes, i.e., Q^i and $W^i \forall i \in \{1, \dots, L\}$. Thus, if two neighboring regions have only have few changes in their code, and that the associated weights are small, then, such a constraint does not affect the overall curvature. On the other hand, if between two regions, the code of a unit having a weight with large amplitude does not change, then the regularization does not affect the curvature either as we can see in the following toy example.

Let consider the case of a 1 hidden-layer encoder, follows by any depth encoder. In the second order regularization, one penalizes $\|A_\omega^E - A_{\omega'}^E\|_F$, where ω and ω' are neighboring regions. We know that $A_\omega^E = Q_\omega^1 W^1$, now let consider the case of a 3 ReLU-units encoder, that is, Q_ω^1 is a 3×3 diagonal matrix, and $W^1 \in \mathbb{R}^{3 \times n}$, where n is the input space dimension. A particular case we consider for our analysis is, $Q_\omega^1 = \text{Diag}(1, 0, 1)$, and $Q_{\omega'}^1 = \text{Diag}(1, 1, 1)$, i.e., the first region ω is encoded by 2 activated ReLUs and ω' by 3. The associated HOC penalization is

$$\left\| \begin{pmatrix} W_{1,:}^1 \\ 0 \\ W_{3,:}^1 \end{pmatrix} - \begin{pmatrix} W_{1,:}^1 \\ W_{2,:}^1 \\ W_{3,:}^1 \end{pmatrix} \right\|_F = \|W_{2,:}^1\|_2, \text{ where } W^1 = \begin{pmatrix} W_{1,:}^1 \\ W_{2,:}^1 \\ W_{3,:}^1 \end{pmatrix}.$$

Therefore we see even if $W_{1,:}^1$ or $W_{3,:}^1$ are large, they will not induce a penalization of the curvature between the region ω and ω' . Besides, if $W_{2,:}^1$ is small, even though it is associated with the changing unit between the two regions, the

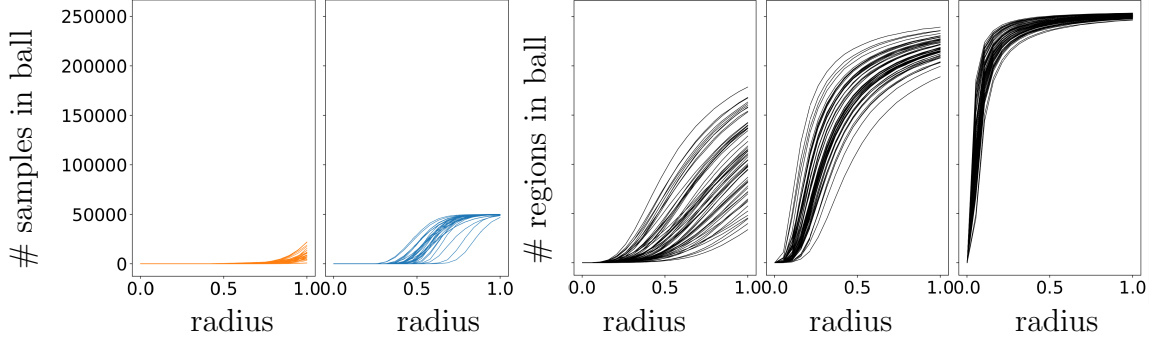


Figure 6.3 : The first and second figures (from left to right) represent the number of data points inside a ball of growing radius (*first to second*: CIFAR10, MNIST). From the third to the last figure (from left to right), we show the number of regions in the latent space of the AE inside the same ball of growing radius for different AE architectures (*third to fifth*: Small MLP, Large MLP, Convolutional). We observe that the number of regions induced by the AE partitioning of any DN architecture in any randomly sampled ball is much larger than the number of data for any radius.

curvature will not be penalized either.

- (ii) **Denoising AE** [183]: Denoising AE is known to have a similar effect than the weight-decay penalty on the DN architecture [207]. A penalty on the energy of W^ℓ induces a penalty on the energy of the A_ω^E and A_ω^D , $\forall \omega \in \Omega^{E,D}$. Therefore, it constrains each piece's slope to be as flat a possible, implying that the piecewise linear map focuses on approximating the low-frequency content in the data, which reinforces the learning bias of deep networks towards low-frequency information [208]. Thus, we see how denoising and Higher-Order Contractive are tied together.

Now that we understand autoencoders' different components and their underlying functionality, we propose to constrain the surface's geometry spanned by the CPA map. In fact, we can see in Fig. 6.3 that for a given ball positioned in the input data

space, the number of regions induced by the AE is much larger than the number of data. It is then clear that only a few of the regions contain data points. Thus, besides the implicit constraints of Deep Network, such as weight sharing on convolutional nets, and the continuity constraints of the mapping, there are no other structural constraints on the behavior of regions where no training data are available [209, 210]. *There is, therefore, a need to constrain all the regions of the CPA to guarantee the generalization capability of AEs.*

iv Lie Group Orbit Fitting For the remaining of the paper, we model the dataset as the orbit of a Lie group, that is, as per Eq. 2.14, $\mathbf{x}(\theta) = \exp(\theta G)\mathbf{x}(0)$, $\theta \in \mathbb{R}$, $G \in \mathcal{T}_I\mathcal{G}$, where $\mathcal{T}_I\mathcal{G}$ denotes the Lie algebra of the group \mathcal{G} . We also assume that $\forall \theta \in \mathbb{R}$, $x(\theta) \in \mathbb{R}^d \setminus \{0\}$ to avoid degenerated cases. Our aim is to provide a regularization that leads to generalization guarantees, i.e., the AE is equal to \mathbf{x} at any location of the manifold. In Sec. iv, we first provide such a regularization from a general point of view, that is, we consider the approximation of \mathbf{x} by a smooth interpolation function ($C^2(\mathbb{R}, \mathbb{R}^d)$). We then translate this condition for CPA operators (Sec. v) to apply it to any AE. We then demonstrate the generalization guarantees it yields (Sec. vi).

First, we want to understand under which condition a interpolation function $\mathbf{f} \in C^2(\mathbb{R}, \mathbb{R}^d)$ coincides with the orbit of $\mathbf{x}(0) \in \mathbb{R}^d \setminus \{0\}$ under the action of the group \mathcal{G} . In particular, we propose to exploit a regularization that induces an orbit of a Lie group, such as

$$\mathcal{R}_k(\mathbf{f}) \triangleq \int \left\| \frac{d^k \mathbf{f}(\theta)}{d\theta^k} - G \frac{d^{k-1} \mathbf{f}(\theta)}{d\theta^{k-1}} \right\| d\theta, \quad (6.10)$$

where $\frac{d^k \mathbf{f}(\theta)}{d\theta^k}$ denotes the k^{th} order derivative of \mathbf{f} .

This regularization constrains \mathbf{f} such that its k^{th} order derivative is a linear map of the $k - 1$ order. In the following theorem, we show that, for $k \in \{1, 2\}$, such regu-

larization coupled with an interpolation loss function leads to a perfect approximation of the data manifold x . That is, \mathbf{f} coincides with $\mathbf{x}(\theta) = \exp(\theta G)\mathbf{x}(0), \forall \theta \in \mathbb{R}$ if and only if $\frac{d^k \mathbf{f}(\theta)}{d\theta^k} = G \frac{d^{k-1} \mathbf{f}(\theta)}{d\theta^{k-1}}$ and it exists a certain number of θ_i , depending on the order k , such that $\mathbf{f}(\theta_i) = \mathbf{x}(\theta_i)$. Note that the restriction to the first two orders is natural as we will apply these results on continuous piecewise affine maps, in which the second-order can only be approximated using stochastic approximation as per Sec. ii.

Theorem 2. *For all $k \in \{1, 2\}$, assuming G is invertible, and that a function \mathbf{f} minimizes the regularization $\mathcal{R}_k(\mathbf{f})$ and it exists $\theta_i, i \in \{1, \dots, k\}$ such that $\mathbf{f}(\theta_i) = \mathbf{x}(\theta_i)$ then \mathbf{f} has perfect generalization as in*

$$\mathcal{R}_k(\mathbf{f}) = 0 \text{ and } \exists \theta_i \in \{1, \dots, k\} \text{ s.t. } \mathbf{f}(\theta_i) = \mathbf{x}(\theta_i) \iff \forall \theta, \mathbf{x}(\theta) = \mathbf{f}(\theta). \quad (6.11)$$

Proof. For both cases, we recall that we assume that $\forall \theta, x(\theta) \neq 0$. In fact, relaxing such assumption would lead to a degenerated case where the interpolant can be constant and equal to 0. In practice this assumption is more than realistic as the '0-datum' is usually not part of any dataset. Let's first consider the case $k = 1$.

We know that the solution of $\frac{d\mathbf{f}(\theta)}{d\theta} = G\mathbf{f}$ is $\mathbf{f}(\theta) = \exp(\theta G)\mathbf{f}(0)$. Now it is clear that if $\exists \theta_1$ such that $\mathbf{f}(\theta_1) = \mathbf{x}(\theta_1)$, then $\mathbf{f}(0) = \mathbf{x}(0)$, and therefore, $\mathbf{f}(\theta) = \exp(\theta G)\mathbf{x}(0) = \mathbf{x}(\theta), \forall \theta$.

Now for the case $k = 2$,

Let $\mathbf{y}(\theta) = \frac{d\mathbf{f}(\theta)}{d\theta}$, then we have

$$\frac{d\mathbf{y}(\theta)}{d\theta} = G\mathbf{y}(\theta),$$

which solution is

$$\mathbf{y}(\theta) = \exp(\theta G)\mathbf{y}(0).$$

Thus, $\frac{d\mathbf{f}}{d\theta} = \exp(\theta G)\frac{d\mathbf{f}(\theta)}{d\theta}|_{\theta=0}$. Now since

$$\exp(\theta G)G\frac{d\mathbf{f}(\theta)}{d\theta}|_{\theta=0} = \sum_{n \geq 0} \frac{G^n}{n!}G\frac{d\mathbf{f}(\theta)}{d\theta}|_{\theta=0} = \sum_{n \geq 0} G\frac{G^n}{n!}\frac{d\mathbf{f}(\theta)}{d\theta}|_{\theta=0} = G\exp(\theta G)\frac{d\mathbf{f}(\theta)}{d\theta}|_{\theta=0}$$

we have that,

$$\mathbf{f}(\theta) = \exp(\theta G)G^{-1}\frac{d\mathbf{f}(\theta)}{d\theta}|_{\theta=0} + c\mathbf{1},$$

where $c \in \mathbb{R}$ and $\mathbf{1}$ denotes the d -dimensional vector of 1. Let's now add the interpolation condition, that is

$$\exists \theta_1, \theta_2, \text{ s.t. } \mathbf{f}(\theta_1) = \mathbf{x}(\theta_1), \quad \mathbf{f}(\theta_2) = \mathbf{x}(\theta_2)$$

Which is equivalent to

$$\begin{cases} \exp(\theta_1 G)G^{-1}\frac{d\mathbf{f}(\theta)}{d\theta}|_{\theta=0} + c\mathbf{1} = \exp(\theta_1 G)x(0) \\ \exp(\theta_2 G)G^{-1}\frac{d\mathbf{f}(\theta)}{d\theta}|_{\theta=0} + c\mathbf{1} = \exp(\theta_2 G)x(0) \end{cases}$$

Which implies that, $\frac{d\mathbf{f}(\theta)}{d\theta}|_{\theta=0} = Gx(0)$ and that $c = 0$.

Therefore,

$$\mathbf{f}(\theta) = \exp(\theta G)x(0) = x(\theta), \forall \theta$$

□

Thus an interpolant \mathbf{f} , can approximate the orbit of a Lie group, utilizing two components, the aforementioned regularization with $k \in \{1, 2\}$, and a reconstruction error that force the interpolation function to coincide with k training samples.

v Regularizations for Continuous Piecewise Affine Maps The derived regularizations were based on a smooth interpolant \mathbf{f} and need to be adapted to the case of a CPA map. To do so, there are several crucial considerations:

- (i) For the sake of clarity, the previous section illustrated the case of a one-dimensional group. Here we propose to generalize such an approach to multiple groups of transformations. We, therefore, consider the case of h infinitesimal operators G_1, \dots, G_h each corresponding to a 1-dimensional group, as explained in Sec. 2.3.
- (ii) The second-order regularization requires constrains the Hessian of the CPA, which by definition, can only be approximated stochastically as explained in Sec. 6.4 - ii.
- (iii) The assumption on the data is that they are generated by h transformation groups. Thus, the intrinsic dimensionality of the data is at most h . Therefore, the size of the bottleneck layer, which corresponds to the maximum dimension of the manifold the autoencoder can generate is also h .

The case $k = 1$: The first-order regularization corresponds to the assumption that data that are generated by the decoder and that are close to each other result from small transformations of one to another. As per Eq. 2.15, we obtain

$$\mathcal{R}_1(\mathbf{D}) \triangleq \min_{G_1, \dots, G_h} \int_{\mathbb{R}^h} \int_{\mathcal{N}(\theta)} \min_{\epsilon_1, \dots, \epsilon_h} \left\| \mathbf{D}(\theta) - \left(I + \sum_{k=1}^h \epsilon_k G_k \right) \mathbf{D}(\theta') \right\|_2 d\theta' d\theta, \quad (6.12)$$

where $\mathcal{N}(\theta)$ denotes the neighborhood of $\theta \in \mathbb{R}^h$, the parameters $\epsilon_1, \dots, \epsilon_h$ are the scalars corresponding to the scale of the transformations, and the G_1, \dots, G_h the infinitesimal operators. The optimal parameters $\epsilon^* = [\epsilon_1, \dots, \epsilon_h]^T$ used during the training of the regularized AE are provided by the following proposition.

Proposition 9. *The ϵ of the first-order regularization defined in Eq. 6.12 is obtained*

as

$$\epsilon^* = \begin{pmatrix} \|G_1 \mathbf{D}(\theta')\|_2^2 & \dots & \langle G_h \mathbf{D}(\theta'), G_1 \mathbf{D}(\theta') \rangle \\ \vdots & \ddots & \vdots \\ \langle G_1 \mathbf{D}(\theta'), G_h \mathbf{D}(\theta') \rangle & \dots & \|G_h \mathbf{D}(\theta')\|_2^2 \end{pmatrix}^{-1} \begin{pmatrix} \langle \mathbf{D}(\theta) - \mathbf{D}(\theta'), G_1 \mathbf{D}(\theta') \rangle \\ \vdots \\ \langle \mathbf{D}(\theta) - \mathbf{D}(\theta'), G_h \mathbf{D}(\theta') \rangle \end{pmatrix}$$

where the matrix is always invertible ($\mathbf{D}(\theta') \neq 0$).

Proof.

$$\begin{aligned} \left\| \mathbf{D}(\theta) - \left(I + \sum_{k=1}^h \epsilon_k G_k \right) \mathbf{D}(\theta') \right\|_2^2 &= \left\| \mathbf{D}(\theta) - \mathbf{D}(\theta') - \sum_{k=1}^h \epsilon_k G_k \mathbf{D}(\theta') \right\|_2^2 \\ &= \langle \mathbf{D}(\theta) - \mathbf{D}(\theta'), \mathbf{D}(\theta) - \mathbf{D}(\theta') \rangle \\ &\quad - 2 \left\langle \mathbf{D}(\theta) - \mathbf{D}(\theta'), \sum_{k=1}^h \epsilon_k G_k \mathbf{D}(\theta') \right\rangle \\ &\quad + \left\langle \sum_{k=1}^h \epsilon_k G_k \mathbf{D}(\theta'), \sum_{k=1}^h \epsilon_k G_k \mathbf{D}(\theta') \right\rangle, \end{aligned}$$

Now, $\forall j \in \{1, \dots, h\}$

$$\frac{\delta \left\| \mathbf{D}(\theta) - \mathbf{D}(\theta') - \sum_{k=1}^h \epsilon_k G_k \mathbf{D}(\theta') \right\|_2^2}{\delta \epsilon_j} = -2(\mathbf{D}(\theta) - \mathbf{D}(\theta'))^T G_j \mathbf{D}(\theta') + 2 \sum_{k=1}^h \epsilon_k \mathbf{D}(\theta')^T G_k^T G_j \mathbf{D}(\theta'),$$

setting $\frac{\delta \left\| \mathbf{D}(\theta) - \mathbf{D}(\theta') - \sum_{k=1}^h \epsilon_k G_k \mathbf{D}(\theta') \right\|_2^2}{\delta \epsilon_j} = 0$, for all j we obtain

$$\epsilon^* = \begin{pmatrix} \|G_1 \mathbf{D}(\theta')\|_2^2 & \dots & \mathbf{D}(\theta')^T G_h^T G_1 \mathbf{D}(\theta') \\ \vdots & \ddots & \vdots \\ \mathbf{D}(\theta')^T G_1^T G_h \mathbf{D}(\theta') & \dots & \|G_h \mathbf{D}(\theta')\|_2^2 \end{pmatrix}^{-1} \begin{pmatrix} (\mathbf{D}(\theta) - \mathbf{D}(\theta'))^T G_1 \mathbf{D}(\theta') \\ \vdots \\ (\mathbf{D}(\theta) - \mathbf{D}(\theta'))^T G_h \mathbf{D}(\theta') \end{pmatrix}$$

, and we have that

$$\begin{pmatrix} \|G_1 \mathbf{D}(\theta')\|_2^2 & \dots & \mathbf{D}(\theta')^T G_h^T G_1 \mathbf{D}(\theta') \\ \vdots & \ddots & \vdots \\ \mathbf{D}(\theta')^T G_1^T G_h \mathbf{D}(\theta') & \dots & \|G_h \mathbf{D}(\theta')\|_2^2 \end{pmatrix} = \begin{pmatrix} G_1 \mathbf{D}(\theta') \\ \vdots \\ G_h \mathbf{D}(\theta') \end{pmatrix}^T \begin{pmatrix} G_1 \mathbf{D}(\theta') \\ \vdots \\ G_h \mathbf{D}(\theta') \end{pmatrix}$$

which is thus a positive definite matrix. \square

Note that the infinitesimal operators are learned using stochastic gradient descent and that the approximation of the integrals in Eq. 6.12 is developed in Sec. 6.5 - ii.

The case $k = 2$: We know from Sec. ii that for each region $\omega \in \Omega^D$, the decoder is characterized by its tangent plane, A_ω^D . The second-order regularization imposes that each tangent plane of the AE is related to their neighboring tangents plane by small transformations. Again, considering the linearized exponential maps and exploiting the definition of the Hessian in Eq. 6.9 we obtain the following second-order regularization on the CPA

$$\mathcal{R}_2(\mathbf{D}) \triangleq \min_{G_1, \dots, G_h} \int_{\mathbb{R}^h} \int_{\mathcal{N}(\omega)} \min_{\epsilon_1, \dots, \epsilon_h} \left\| J_\omega[\mathbf{D}] - \left(I + \sum_{k=1}^h \epsilon_k G_k \right) J_{\omega'}[\mathbf{D}] \right\|_F d\omega' d\omega, \quad (6.13)$$

where $\mathcal{N}(\omega)$ denotes the set of neighbors of region ω and $\|\cdot\|_F$ is the Frobenius norm and we recall that $J_{\omega'}[\mathbf{D}] = A_{\omega'}^D$ and $J_\omega[\mathbf{D}] = A_\omega^D$. The implementation regarding the sampling of neighboring regions is detailed Sec. 6.5 - ii. In this case also, the optimal parameters $\epsilon^* = [\epsilon_1, \dots, \epsilon_h]^T$ used during the training of the regularized AE are provided in the following proposition.

Proposition 10. *The ϵ of the second-order regularization defined in Eq. 6.13 is obtained by*

$$\epsilon^* = \begin{pmatrix} \Sigma_i \|G_1[A_\omega^D]_{\cdot, i}\|_2^2 & \dots & \Sigma_i \langle G_1[A_\omega^D]_{\cdot, i}, G_h[A_\omega^D]_{\cdot, i} \rangle \\ \vdots & \ddots & \vdots \\ \Sigma_i \langle G_h[A_\omega^D]_{\cdot, i}, G_1[A_\omega^D]_{\cdot, i} \rangle & \dots & \Sigma_i \|G_h[A_\omega^D]_{\cdot, i}\|_2^2 \end{pmatrix}^{-1} \begin{pmatrix} \Sigma_i \langle G_1[A_\omega^D]_{\cdot, i}, [A_\omega^D]_{\cdot, i} - [A_{\omega'}^D]_{\cdot, i} \rangle \\ \vdots \\ \Sigma_i \langle G_h[A_\omega^D]_{\cdot, i}, [A_\omega^D]_{\cdot, i} - [A_{\omega'}^D]_{\cdot, i} \rangle \end{pmatrix},$$

where the matrix is invertible ($A_\omega^D \neq 0$).

Proof. Given $J_{\omega'}[\mathbf{D}] = A_{\omega'}^D$ and $J_{\omega}[\mathbf{D}] = A_{\omega}^D$, we have

$$\begin{aligned} \left\| A_{\omega'}^D - A_{\omega}^D - \sum_{k=1}^h \epsilon_k G_k A_{\omega}^D \right\|_F^2 &= \text{Tr}(A_{\omega'}^D \odot A_{\omega'}^D - A_{\omega'}^D \odot A_{\omega}^D - A_{\omega}^D \odot A_{\omega'}^D + A_{\omega}^D \odot A_{\omega}^D \\ &\quad + A_{\omega}^D \odot A_{\omega}^D - A_{\omega}^D \odot A_{\omega'}^D + A_{\omega'}^D \odot \left(\sum_{h=1}^k \epsilon_k G_k A_{\omega}^D \right) \\ &\quad - \left(\sum_{h=1}^k \epsilon_k G_k A_{\omega}^D \right) \odot A_{\omega'}^D + \left(\sum_{h=1}^k \epsilon_k G_k A_{\omega}^D \right) \odot A_{\omega}^D \\ &\quad + \left(\sum_{h=1}^k \epsilon_k G_k A_{\omega}^D \right) \odot \left(\sum_{h=1}^k \epsilon_k G_k A_{\omega}^D \right)) 11^T). \end{aligned}$$

Now, $\forall j \in \{1, \dots, h\}$

$$\begin{aligned} \frac{\delta \left\| A_{\omega'}^D - A_{\omega}^D - \sum_{k=1}^h \epsilon_k G_k A_{\omega}^D \right\|_F^2}{\delta \epsilon_j} &= 2 \text{Tr} \left((G_j A_{\omega}^D) \odot (A_{\omega}^D - A_{\omega'}^D + \sum_{k=1}^h G_k A_{\omega}^D 11^T) \right) \\ &= 2 \text{Tr}(G_j A_{\omega}^D \odot (A_{\omega}^D - A_{\omega'}^D) 11^T) \\ &\quad + 2 \sum_{k=1}^h \epsilon_k \text{Tr}((G_j A_{\omega}^D \odot G_k A_{\omega}^D) 11^T), \end{aligned}$$

setting $\frac{\delta \left\| A_{\omega'}^D - A_{\omega}^D - \sum_{k=1}^h \epsilon_k G_k A_{\omega}^D \right\|_F^2}{\delta \epsilon_j} = 0$ for all j and rearranging in matrix form gives

$$\epsilon^* = \begin{pmatrix} \sum_i \|G_1[A_{\omega}^D]_{\cdot, i}\|_2^2 & \dots & \sum_i \langle G_1[A_{\omega}^D]_{\cdot, i}, G_h[A_{\omega}^D]_{\cdot, i} \rangle \\ \vdots & \ddots & \vdots \\ \sum_i \langle G_h[A_{\omega}^D]_{\cdot, i}, G_1[A_{\omega}^D]_{\cdot, i} \rangle & \dots & \sum_i \|G_h[A_{\omega}^D]_{\cdot, i}\|_2^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum_i \langle G_1[A_{\omega}^D]_{\cdot, i}, [A_{\omega'}^D]_{\cdot, i} - [A_{\omega}^D]_{\cdot, i} \rangle \\ \vdots \\ \sum_i \langle G_h[A_{\omega}^D]_{\cdot, i}, [A_{\omega'}^D]_{\cdot, i} - [A_{\omega}^D]_{\cdot, i} \rangle \end{pmatrix},$$

and we have that

$$\begin{pmatrix} \sum_i \|G_1[A_{\omega}^D]_{\cdot, i}\|_2^2 & \dots & \sum_i \langle G_1[A_{\omega}^D]_{\cdot, i}, G_h[A_{\omega}^D]_{\cdot, i} \rangle \\ \vdots & \ddots & \vdots \\ \sum_i \langle G_h[A_{\omega}^D]_{\cdot, i}, G_1[A_{\omega}^D]_{\cdot, i} \rangle & \dots & \sum_i \|G_h[A_{\omega}^D]_{\cdot, i}\|_2^2 \end{pmatrix} = \sum_{i=1}^h \begin{bmatrix} G_1[A_{\omega}^D]_{\cdot, i} \\ \vdots \\ G_h[A_{\omega}^D]_{\cdot, i} \end{bmatrix}^T \begin{bmatrix} G_1[A_{\omega}^D]_{\cdot, i} \\ \vdots \\ G_h[A_{\omega}^D]_{\cdot, i} \end{bmatrix},$$

therefore it is the sum of positive definite matrices.

For the case $h=1$, we have that

$$\begin{aligned} \|a_{\omega'}^D - a_{\omega}^D - \epsilon G a_{\omega}^D\|^2 &= \langle a_{\omega'}^D, a_{\omega'}^D \rangle - 2 \langle a_{\omega'}^D, a_{\omega}^D \rangle + \langle a_{\omega}^D, a_{\omega}^D \rangle \\ &\quad + 2 \langle \epsilon G a_{\omega}^D, a_{\omega}^D - a_{\omega'}^D \rangle + \langle \epsilon G a_{\omega}^D, \epsilon G a_{\omega}^D \rangle, \end{aligned}$$

thus,

$$\frac{\delta \|a_{\omega'}^D - a_{\omega}^D - \epsilon G a_{\omega}^D\|^2}{\delta \epsilon} = a_{\omega}^{D^T} G^T (a_{\omega}^D - a_{\omega'}^D) + \epsilon a_{\omega}^{D^T} G^T G a_{\omega}^D$$

□

Let us now provide interpretations regarding the Lie group regularizations we developed. While the first-order regularization constrains the AE mapping, the second-order constrains the AE's tangent plane of each region. In the first-order case the distance between $(I + \sum_{k=1}^h \epsilon_k G_k) \mathbf{D}(\theta')$, which corresponds to small transformations of the sample generated by the decoder, and $D(\theta)$ is minimized. Thus, such a regularization constrains the AE mapping to approximate the orbit induced by the infinitesimal generators. Then, the second-order regularization aims at minimizing the distance between $(I + \sum_{k=1}^h \epsilon_k G_k) J_{\omega'}[\mathbf{D}]$, which is the small transformation of the tangent plane of region ω' , and $J_{\omega}[\mathbf{D}]$. This means that the second-order regularization constrains the Hessian of the decoder, which defines the angle between neighboring piecewise linear maps, to approximating the angle of the data manifold. Therefore, this penalization enforces the curvature of the piecewise linear map to fit the curvature of the orbit. Besides, as opposed to the Higher-Order Contractive AE [203], these regularizations constrain all the piecewise affine regions whether they contain training data or not as they do not rely on samples from the dataset. This is crucial to provide generalization guarantees in a finite data regime.

vi Manifold Approximation Error In Sec. 6.4 iv, we showed that if the regularization defined in Eq. 6.10 is equal to zero for any given $k \in \{1, 2\}$, and if the interpolation function \mathbf{f} coincides with the data manifold defined by \mathbf{x} on k points, then \mathbf{f} coincides with \mathbf{x} . We now derive the generalization guarantees in the particular case where \mathbf{f} is a CPA approximant.

Based on the assumption that (i) a region of the real manifold is correctly approximated, (ii) one of the regularizations defined in Eq. 6.12, 6.13 is minimized, and that (iii) the infinitesimal operator G obtained from the regularization coincides with the infinitesimal operator of the group governing the data, we obtain the following bound on the approximation of the data manifold.

Theorem 3. *If on a region $\omega' \in \Omega^D$ the matrix $A_{\omega'}^D$ forms a basis of the manifold tangent space on this region, and it exists $k \in \{1, 2\}$ such that $\mathcal{R}_k(\mathbf{D}) = 0$ then for all regions $\omega \in \Omega^D$ the basis vectors of A_{ω}^D are the basis vector of the tangent of the data manifold and the distance between the continuous piecewise affine map and the data manifold is upper bounded by the radius of the regions as per*

$$d(\cup_{\omega \in \Omega^D} \mathcal{T}_{AE}(\omega), \mathcal{X}) \leq \sum_{\omega_i \in \Omega^D} \text{Rad}(\omega_i),$$

where $\mathcal{T}_{AE}(\omega)$ the tangent space of the AE for the region ω , \mathcal{X} denotes the data manifold, d defines the 2-norm distance, and $\text{Rad}(\omega_i)$ the radius of the region ω_i .

For the following proofs, we will denote by $T : \mathbb{R}^d \times \mathbb{R}^h \rightarrow \mathbb{R}^d$, the transformation operator taking as input a datum and a group parameter, and giving as output the transformed datum. As we used a Lie group, we can define this operator analytically as $T(x, \theta) = \exp(\theta G)x$. We will use the notation $\mathcal{T}_{\mathcal{X}}(\omega)$ as the tangent space of the manifold described by the data \mathcal{X} for the data in the region ω , and by $\mathcal{T}_{AE}(\omega)$ the tangent space of the AE for the region ω . We show that if these two tangent space coincides for a given region, i.e., if the tangent space of the AE coincides with the tangent space of the manifold for a specific position, then they coincide everywhere.

Proof. By assumption, we know that $\{a_1^D(\omega'), \dots, a_h^D[\omega']\}$ form a basis of $\mathcal{T}_{\mathcal{X}}[\omega']$. If the regularization is satisfied, we also know that the tangent induced by the AE at position ω , denoted by $\mathcal{T}_{AE}(\omega)$, is equal to $T(\mathcal{T}_{\mathcal{X}}(\omega'), \theta)$. In fact, for the order $k = 2$

the regularization imposes that the tangent (induced by the AE) of the different regions are transformed version of each other by the transformation operator T . Now for the order one, we know that if $\frac{d\mathbf{f}(\theta)}{d\theta} = G\mathbf{f}(\theta)$, then $\frac{d^2\mathbf{f}(\theta)}{d\theta^2} = G\frac{d\mathbf{f}(\theta)}{d\theta}$. Which means that if the outputs of the interpolant \mathbf{f} are connected by the transformation group T , then the tangents of such interpolant are also connected by the same group of transformation.

Note that the operator T forms a Lie group action operator, it is a diffeomorphism from the orbit of the group to the orbit of the group. Therefore, $\forall\omega$, it exists θ such that $T(\mathcal{T}_x(\omega'), \theta) = \mathcal{T}_x(\omega)$. Per assumption, the tangent of the region ω' , i.e. $\mathcal{T}_{AE}(\omega')$ is actually tangent to \mathcal{X} as its basis coincides with $\mathcal{T}_x(\omega')$. Denote by $x \in \mathcal{X}$ the point at which $\mathcal{T}_x(\omega')$ and \mathcal{X} intersects. Let's first first prove that for $\epsilon' = \arg \max_{\epsilon} x + \epsilon h \in \omega$, where $h \in \mathcal{T}_x(\omega')$, that is, $x + \epsilon' h$ lies at the boundary of the region ω' . We further assume that $\|h\| = 1$ such that $\epsilon' = \text{Rad}(\omega')$. Let's define a smooth curve on the manifold $\gamma : \mathbb{R} \rightarrow \mathcal{X}$ such that $\gamma(0) = x$ and $\gamma'(0) = h$. Now,

$$\begin{aligned} d(x + \epsilon' h, \mathcal{X}) &\leq d(x + \epsilon' h, \gamma(\epsilon')) \\ &= \|\gamma(\epsilon') - \gamma(0) - \epsilon' \gamma'(0)\|. \end{aligned}$$

Since, $\lim_{\epsilon' \rightarrow 0} \frac{\gamma(\epsilon') - \gamma(0)}{\epsilon'} = \gamma'(0)$, we have that $\frac{d(x + \epsilon' h, \gamma(\epsilon'))}{\epsilon'} = o(\text{Rad}(\omega'))$. Then, since the $\omega_i \forall i \in \{1, \dots, |\Omega|\}$ form a partition of Ω and that by Proposition 3 we know that since one tangent of the AE coincides with the tangent of the manifold at the point x then any tangent of the AE coincides with a tangent of the manifold. Thus, we have that $d(\cup_{\omega \in \Omega} \mathcal{T}_{AE}(\omega), \mathcal{X}) = \sum_{i=1}^{|\Omega|} d(\mathcal{T}_{AE}(\omega_i), \mathcal{X}) \leq \sum_{i=1}^{|\Omega|} \text{Rad}(\omega_i)$. \square

The previous statement shows that if the number of pieces of the piecewise affine map, which depends on the number of neurons in the DN architecture (see Fig. 6.3 and refer to [211] for more details) and the type of nonlinearity, goes to infinity, then

the decoder would coincide with the data manifold. In a practical setting, it tells us that the higher the number of regions is, the higher is the degrees of freedom of the CPA, and that under this regularization, these degrees of freedom are controlled while not requiring more training points.

vii Complexity & Parameters Recall that in the proposed second-order regularization, one should have the knowledge of the decoder latent space partition. In practice, and for large networks, the discovery of the partition would not be feasible. We thus propose to approximate the regularization by only sampling some of the regions and some of their respective neighbors. This sampling is done by first randomly sampling some vectors in the AE latent space. As for each sample, the associated per region map is automatically formed during the forward pass of the decoder, the per region parameters can be obtained by computing the affine mapping induced by the samples. To compute the neighbors of those sample regions, we use a simple dichotomic search. That is, for each of the sampled regions, we sample another (nearby) vector and keep pushing this new sample toward the first sample until one obtains the closest sample that remains in a different region. With the above, one now has the knowledge of some regions and one neighboring region for each of those regions. We leverage this approach and perform the search of a single neighbor; for a better approximation of the regularization, one can repeat this sampling process and accumulate the obtained regions and neighbors. For the first-order term, we propose a similar approximation where we approximate the integral by sampling a latent space vector θ (at each mini-batch).

Let us now consider the computational complexity induced by the regularizations omitting the computational cost of a pass through the AE as it is shared across

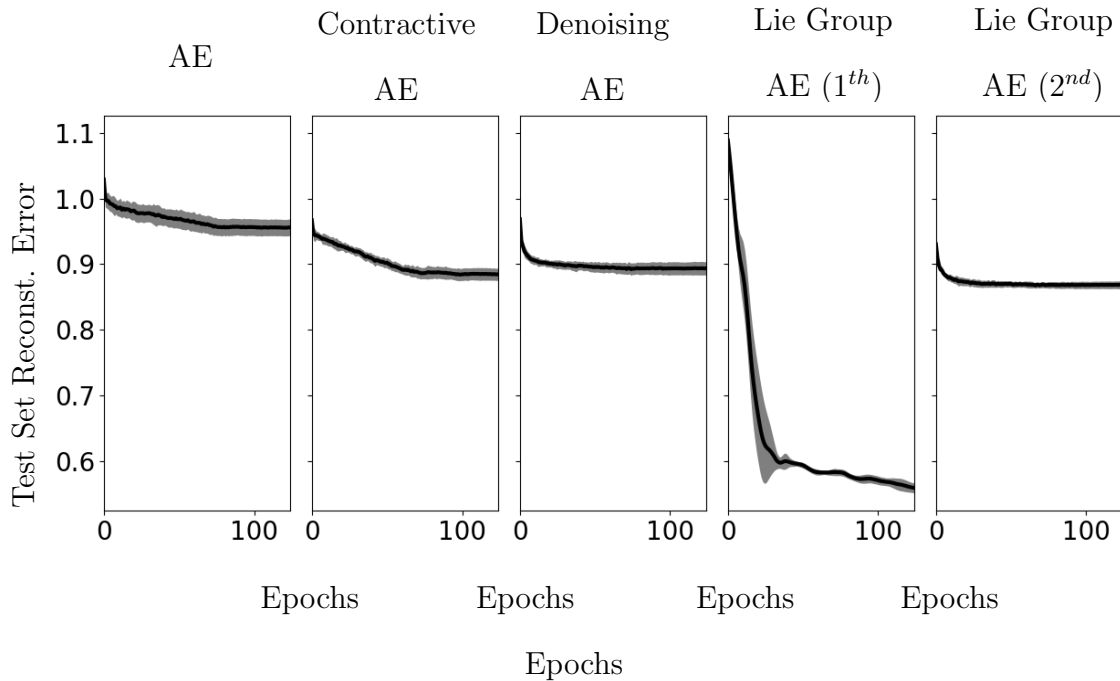


Figure 6.4 : Test set reconstruction error on the SyntheticControl dataset evaluated on the best set of parameters for different AEs (from left to right): AE, Higher Order Contractive AE, Denoising AE, Lie Group AE (first-order), and Lie Group AE (second-order). For each model, the mean over 10 runs is reported in black, and the gray area corresponds to its standard deviation. We observe that the first order regularization performs much better than the second order one, which is close to the higher-order contactive AE error. In fact, this dataset contains six classes of time-series trends (upward, downward, normal,...), which can be easily related by a linear transformations, that is, the first order regularization can be easily optimized. The second order is harder to train and is more sensitive to the sampling of the regions, therefore is less reliable and harder to interpret.

all techniques. The optimal coordinates ϵ^* are obtained by solving a linear system of h equations in both cases for each sampled datum (first-order) or each sampled region (second-order). This equation has to be solved for each sample region or latent space vector; we denote this by N as in our case, we sample in each mini-batch as many vectors/regions as the size of the mini-batch. We obtain the time complexity $\mathcal{O}(h^2N + d^2hN)$ for the first-order, and $\mathcal{O}(h^2N + d^2h^2N)$ for the second-order, and

a space complexity of $\mathcal{O}(d^2h)$ in both cases being driven by the need to retain the matrices G_1, \dots, G_h . The current bottleneck is the storage of those matrices, which limits the size of the AE bottleneck and output dimension.

6.5 Experiments

In this section, we discuss some practical aspects of the proposed regularizations as well as provide the experimental validations. In particular, how the parameters of the regularizations are learned as well as how the sampling required in both regularizations is performed along with their parameters, and finally the experimental validations.

i Parameters The degrees of freedom of our regularized AE comprise the usual AE parameters (per layer affine transformations) and the parameters of each regularization. The ϵ values are found from the analytical form given by Propositions 9 and 10. We learn the matrices $G_k, \forall k \in \{1, \dots, h\}$ with gradient descent based optimizer [150] and thus our method introduces hd^2 additional parameters, where d is the dimension of the input data. Note that a priori knowledge on the structure of the G_k such as low-rank or skew-symmetric, i.e., Lie algebra of the special orthogonal group, can be imposed to reduce the number of parameters; we do not explore this in our study while it could be considered to speed up the computations and improve the regularization tractability. The regularizations themselves depend on the AE to find the optimal ϵ and adapt the matrices G_k . The dimension of each G_k is quadratic in the dimension of the data. As such, for a high-dimensional datasets, the number of learnable parameters is large. Hence the optimization of the G_k matrices remains the current bottleneck of the method. We propose to apply the regularization term during training starting from the random initialization. More advanced strategies such as scheduled alternating

Table 6.1 : Comparison of the testing reconstruction errors ($\times 10^{-2} \pm \text{std} \times 10^{-2}$) for each AE (columns) and dataset (rows). The methods denoted by **Lie G.** (1^{th}) and **Lie G.** (2^{nd}) correspond respectively to the first-order and second-order Lie group regularizations we developed. **H.O.C. AE** denotes the Higher-Order contractive AE, and **Den. AE** denoising AE.

<i>Data\Model</i>	AE	Den. AE	H.O.C. AE	Lie G. (1^{th})	Lie G. (2^{nd})
CIFAR10	5.6 ± 0.05	5.0 ± 0.05	-	4.9 ± 0.07	-
MNIST	12.01 ± 0.003	12.01 ± 0.004	12.01 ± 0.004	6.3 ± 0.1	10.13 ± 0.1
CBF	62.38 ± 0.74	52.66 ± 0.76	51.09 ± 0.54	43.99 ± 1.2	49.73 ± 0.31
Yoga	33.76 ± 0.81	33.29 ± 0.72	32.08 ± 0.42	20.28 ± 1.1	30.78 ± 1.2
Trace	13.95 ± 0.45	11.28 ± 0.57	12.57 ± 0.21	13.23 ± 0.4	10.91 ± 0.45
Wine	63.06 ± 0.02	59.34 ± 0.02	49.94 ± 0.02	19.01 ± 0.02	49.94 ± 0.01
ShapesAll	67.98 ± 3.0	58.67 ± 1.4	61.42 ± 5.5	52.97 ± 1.9	57.80 ± 1.2
FiftyWords	64.91 ± 1.7	60.91 ± 1.0	60.92 ± 0.7	71.84 ± 3.4	57.89 ± 1.0
WordSyn	70.95 ± 1.5	66.02 ± 0.8	66.52 ± 0.5	68.21 ± 2.7	62.22 ± 1.1
Insect	51.86 ± 0.6	40.24 ± 0.8	41.93 ± 0.6	38.11 ± 0.9	38.22 ± 0.3
ECG5000	21.92 ± 0.75	20.31 ± 0.39	20.31 ± 0.36	18.06 ± 0.9	20.29 ± 0.4
Earthquakes	56.23 ± 4.1	54.62 ± 4.1	51.79 ± 1.0	99.41 ± 0.2	50.20 ± 0.5
Haptics	37.25 ± 0.2	36.02 ± 1.8	27.21 ± 0.5	16.94 ± 3.4	26.06 ± 0.9
FaceFour	49.82 ± 1.0	48.51 ± 0.8	48.52 ± 0.7	48.60 ± 1.9	46.00 ± 0.6
Synthetic	95.61 ± 1.3	89.37 ± 1.0	88.47 ± 0.9	55.87 ± 0.8	86.83 ± 0.6

minimization or employing a warm-up phase could be leveraged and result in further improvement in performance.

ii Results We evaluate our framework on diverse datasets, including images and time-series data including speech, medical as well as seismic recordings. For each model and each hyperparameter, we perform 10 runs for 125 epochs with batch size 16. The results are reported in Table 6.1. In this table, the statistics reported correspond to the average over the 10 runs, each run using the test set performances based on the best validation set measure. Note that for CIFAR10, the computational burden of both the second-order Lie group regularization and the higher-order contractive one is too high. Thus only the AE, denoising AE and the first-order Lie group regularization are evaluated.

We propose, in particular, to visualize the test set reconstruction for the different AE models during training in Fig. 6.4, where we can see that both Lie Group AEs are robust to the DN initialization and do not overfit. Besides, we can observe that while the first order on this dataset outperforms all the other regularizations, its variance at the beginning of the learning phase is more volatile than other approaches.

The hyperparameter responsible for the variance of the noise added to the data in the Denoising AE case also corresponding to the noise added to the data to sample Jacobian of nearby regions in Higher-Order Contractive AE parameter is evaluated for the values $\{0.001, 0.01, 0.1, 1\}$. Another hyperparameter is the regularization trade-off parameter for both the Higher-Order Contractive AE and Lie Group AEs, the following values are tested for both models $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. All the models were trained using the same AE with 3 fully connected encoder layers with ReLU with bottleneck dimension $h = 10$, and 3 fully connected decoder layer with ReLU and 1 linear fully connected output layer.

We can observe in Table 6.1 that the Lie group regularizations are usually outperforming the other methods the different datasets we evaluated. While the second-order

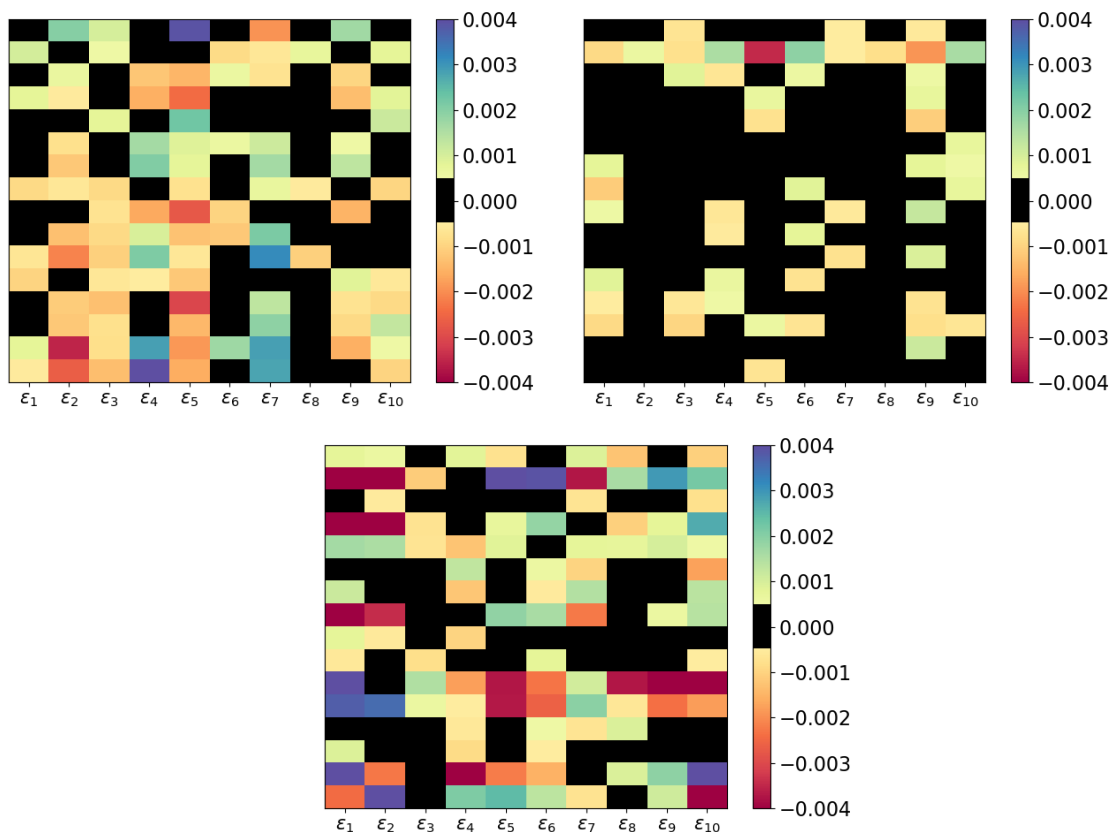


Figure 6.5 : Visualisation of the value (after learning) of the group strength parameter ϵ , for the (*Top left*) SyntheticControl, (*Top right*) Haptics, and (*Bottom*) FaceFour datasets, in the case of the first order regularization. For each row, we observe the value of the parameter ϵ_i for a given sample, $D(\theta)$ and its neighbor $D(\theta')$ as per Eq. 6.12, that is, what is the amplitude of the transformation associated to the generator G_i needed to map one onto the other. Note that we display the epsilons for 16 sampled neighboring pairs.

regularization is more computationally demanding, it appears to be more stable and robust to the change of parameters. Besides, we can see that the first-order regularization might be more sensitive to how close to a Lie group the dataset under evaluation is. In fact, both the first and second-order depends on the learned infinitesimal operator, however, while the second-order use such matrix to constrain the overall curvature of the CPA the first-order use it to constrains its mapping.

In Fig. 6.5, we show, for three datasets, the value of the parameters $\epsilon_i, \forall i \in \{1, \dots, h\}$ obtained by minimizing Eq. 6.12. This observation is important as the framework we propose assumes that the dimension of the orbit, hence the number of group transformations, is the same as the dimension of the bottleneck layer of the autoencoder. In particular, we propose to highlight the sparsity of the parameters, i.e., if for various sampled pairs, the value a particular strength parameter, ϵ_i , is close to 0. If it is, then the associated group of transformation is not being used to map any $\mathbf{D}(\theta)$ to a close sample $\mathbf{D}(\theta')$. Therefore, the number of group of transformations selected is too large, e.g., in the top right subplot, corresponding to the Haptics dataset, we observe that, the 10 transformations are not required.

6.6 Conclusion

We analyzed AEs from a geometrical standpoint and provided insights into how AEs are approximating the data manifold. In particular, we provided analytical formulas of the per region map that AEs are performing using its continuous piecewise affine formulation. This approach’s strength lies in its interpretability power, as for a given region in the input space, the DN mapping is a simple affine map. Leveraging these key features, we proposed to enhance and guarantee the generalization capability of AEs by proposing two regularizations that capture the symmetry in the data. These regularizations constrain the piecewise continuous surface spanned by the decoder to approximate the orbit of a Lie group. Besides, inspired by the theory of learning Lie group transformations, we alleviated the need to explicitly define a group of symmetry underlying the data and propose to learn the group’s generator. In fact, the generator of a Lie group lives in a vector space, thus enabling common matrix manipulations required to perform its update.

7

Discussion & Conclusion

7.1 Discussion

We recall that this thesis aimed to evaluate the possibility of providing suitable and interpretable data embeddings by taking inspiration from the success of DNNs. The first part of this work was particularly inscribed in the field of time series analysis and addressed the possibility of learning and generalizing wavelet transform. Then, we proposed approaching the problem of designing adaptive metrics that take into account the image manifold. Finally, we considered how to provide a manifold approximator with generalization guarantees. In the following, we propose to discuss the answer to each specific question and the limitations of our approaches.

Learnable Wavelet Transform In this work, the mother wavelet is learned, thus replacing the need of expert knowledge on the data to pick the appropriate mother wavelet. An essential aspect of our contribution is the efficient parametrization of

functions based on Hermite cubic splines. In particular, our construction of an appropriate space leads to localized filters that integrate to zeros. Such a parametrization drastically reduces the number of samples that the first layer of the convolutional neural network filters require to converge to a local minimum and provide state-of-the-art results.

Limitations of the Proposed Approach While the adaptivity of the filter and generalization of the mother wavelet has been tackled, the problem lies in the computational complexity. In fact, as opposed to DNN filters used on images of size 5×5 or 3×3 , the wavelet 1-dimensional filters can be large. By construction, the more the filter is low frequency, the larger its size will be. Therefore, the cost computation of the convolution and the back-propagation can be large. Now, given this information and the effectiveness of the method, it appears that the expressive power of DNN can cope with an inappropriate choice of mother wavelet. We believe that this type of filter can provide flexible and efficient shallow models, but in conjunction of a deep network, the computational cost is usually not worth the marginal accuracy gain.

Learnable Group Transform The understanding and decomposition of the wavelet transform via the group transform allow for a nice generalization. One can, in fact, replace the affine group currently used by the wavelet transform with another group to reflect the symmetry in the data. To provide a generalization of the wavelet transform, we opted for the diffeomorphism group. An important part of the contribution was finding an efficient parametrization of this group, which was achieved exploiting 1-hidden layer ReLU network. This type of filter can also replace the first layer of DNN. We obtained state-of-the-art results on various applications and allowed to find a new type of filter adapted to Haptics data.

Limitations of the Proposed Approach As we mentioned for the learnable wavelet transform, the complexity is an issue, and it takes a substantial amount of time to train a DNN with this type of filter bank in the first layer. Another aspect here is how to constrain the diffeomorphic transformations such that the training is stable and no artifacts are introduced in the filter bank. Overall, this approach could be advantageous in an exploratory setting or for shallow models.

Learning Invariant Distances Understanding the topology of the data in an unsupervised manner is a crucial problem that, even with the advances in DNNs the last decade, remains open. Our approach takes advantage of the Thin-plate-spline model that can capture diffeomorphic transformations. The generalization of affine invariant distance and the learnability of the diffeomorphism are crucial, and a large improvement over other approaches is noticed. The full interpretability of the model is a crucial advantage for various applications.

Limitations of the Proposed Approach The major difficulty when working with transformations, and in particular non-rigid transformations in the data space, is the requirement of background removal. In our approach, we selected data without background as not to complicate the approach. Another difficulty is the control over the diffeomorphism as the intensity of the transformation should be tied to the data. In our approach, the Thin-plate-spline allows this parametrization, but its per-data adaptation was not explored.

Learnable Lie Group for Manifold Approximation Understanding and proving the generalization guarantee of an algorithm is one of the milestones of machine learning research. In this work, we propose a regularization that encodes

the notion of equivariance with respect to learnable transformations in the very definition of autoencoders. To do so, we attempted to describe the inner mechanisms of autoencoders as well as how they approximate manifolds. The understanding and interpretability of commonly used variants of autoencoder have been provided. The generalization guarantee has been proven and a generalization bound has been derived, expressing the link between approximation and the region of the piecewise linear autoencoder surface.

Limitations of the Proposed Approach One of the major limitations of the group is the assumption that the data space is homogeneous; that is, the data lie on the orbit of a group. A more realistic approach would consider different orbits, as the data manifold is, almost surely, a collection of “pancakes”. Another consideration is the sampling of the region and the learnability of the Lie algebra basis, both presenting challenges that have not been adequately explored in the literature.

7.2 Conclusion & Future work

This section presents the summary of the main results and future work.

7.2.1 Conclusion

Overall, we shown that it is possible to provide suitable and interpretable data representations based upon the projection onto carefully designed spaces that can compete with Deep Learning methods. To achieve this, we exploited tools from harmonic analysis allowing us to inject prior knowledge into the data representations.

Representation of Time Series The problem of time series representation has been tackled under the frame of time-frequency analysis. While large numbers of ap-

proaches are possible to represent time series efficiently, time-frequency representations appeared to be close to optimal from an empirical standpoint. In fact, when trained on a large number of signals, the first layer of convolutional neural networks converge toward Gabor-like filters. The localization of particular patterns in the time-frequency plane is both particularly interpretable and has been used by engineers and scientists for decades to understand our world better. This representation has been driving the first part of this thesis, as our aim was to provide their modernization. The wavelet transform can be divided into two building blocks, using the group transform formulation, which enables us to understand how we can provide its generalization and learnability. Our approach to achieving these objectives considers the sampling of both the mother wavelet and the group by appropriately sampling carefully designed functional spaces. In both cases, we reach state-of-the-art performances, and our approach can provide insights into the data at hand because of its interpretability. We believe these tools can help scientists extract meaningful information from their time-series data as the learning of their parameters can be combined with any differentiable loss function.

Manifold Interpolation & Quantization In order to provide novel approaches to manifold interpolation and their quantization, we started by exploiting the capability of the K -means algorithm, known for its "simplicity" and efficiency. While the current form of K -means algorithms usually adopts pre-defined metrics, we attempt to modernize it by bringing some flavor of Deep Learning. In fact, we exploited the Thin-plate-spline as a transformation framework that can be learned by back-propagating an error induced by our metric. Together with its particular update rule, our novel approach to K -means is particularly effective for images and can, in fact, characterize

the data manifold and quantize it efficiently. Key results are the tiling of the manifold via orbits and not via convex regions and the understanding of the convergence of the algorithm.

From the image manifold, we extended our characterization of data orbit to any data manifold via the Lie group autoencoder proposed. Enforcing the structure of an orbit to the manifold as well as learning its underlying group was shown to be one way to guarantee generalization. Being capable of learning the symmetry of the data as well as imposing this structural constraint on manifold learning algorithm is key to future manifold approximation work. In particular, to provide more robust and stable algorithms. In fact, this approach replaces the traditional data augmentation techniques, which require lots of design, knowledge on the data and often blur the understanding of the algorithm. In our work, the regularization is explicit and part of the training; thus can be analyzed as part of the entire learning framework.

7.2.2 Future Work

Representation of Time Series We believe that our work was achieved parallelly to other approaches having the same goal and that the limitations have been partially achieved. In particular, the accuracy and predictive power reached their maximum capability given the current methods, and the main contribution would be to reduce the computational burden. We also believe that the variety of time series and their different statistics can hardly be described by a single framework, while in the case of computer vision, most images share the same statistics, particularly their decay in the frequency spectrum. This difference and variety in the data require the development of specialized solutions, which are usually in the hand of the expert of datasets.

Now, a lot remains to be done on irregular domains like graphs. The same approach can be considered to replace the graph convolution used in deep networks by carefully designed learnable wavelet graph filters. Again, this would fit the specific application that appropriately resonates with wavelet analysis.

Manifold Interpolation & Quantization This idea that the data lies on a low-dimensional space and that our high-dimensional observations are the result of nuisances is a powerful idea that can help further improve manifold learning algorithms. Our approach that focuses on the learnability of the group underlying the entire data manifold is just at its first fruits, and there is a lot to be done. The learnability of the group underlying the data needs to be improved from all aspects, and a formulation that assumes multiple orbits within the data is required. Also, a parametrization of diffeomorphism and its theoretical analysis needs to be developed. While GAN provided a fantastic toolbox to be trained on a huge quantity of dataset, we believe that its lack of stability, guarantee, and interpretability create the needs for approaches like ours that describe the data more analytically. This part of our work requires the development of theoretical and practical tools that can help us create a novel type of interpretable, guaranteed, and efficient algorithm.

Bibliography

- [1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
- [3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.
- [4] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” *arXiv preprint arXiv:1409.3215*, 2014.
- [5] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv:1312.5602*, 2013.

- [6] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, “Mastering the game of go with deep neural networks and tree search,” *Nature*, vol. 529, pp. 484–503, 2016.
- [7] K. Fukushima and S. Miyake, “Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition,” in *Competition and cooperation in neural nets*, pp. 267–285, Springer, 1982.
- [8] W. Pitts and W. S. McCulloch, “How we know universals the perception of auditory and visual forms,” *The Bulletin of Mathematical Biophysics*, vol. 9, no. 3, pp. 127–147, 1947.
- [9] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, “Handwritten digit recognition with a back-propagation network,” in *Advances in Neural Information Processing Systems* (D. Touretzky, ed.), vol. 2, Morgan-Kaufmann, 1990.
- [10] S. Mallat, “Group invariant scattering,” *Communications on Pure and Applied Mathematics*, vol. 65, no. 10, pp. 1331–1398, 2012.
- [11] G. E. Hinton, A. Krizhevsky, and S. D. Wang, “Transforming auto-encoders,” in *International conference on artificial neural networks*, pp. 44–51, Springer, 2011.
- [12] P. Flandrin, *Time-frequency/time-scale analysis*. Academic press, 1998.
- [13] C. Torrence and G. P. Compo, “A practical guide to wavelet analysis,” *Bulletin of the American Meteorological society*, vol. 79, no. 1, pp. 61–78, 1998.

- [14] R. Coifman and M. Wickerhauser, “Entropy-based algorithms for best basis selection,” *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 713–718, 1992.
- [15] R. Cosentino, R. Balestriero, and B. Aazhang, “Best basis selection using sparsity driven multi-family wavelet transform,” in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 252–256, 2016.
- [16] A. Megahed, A. Monem Moussa, H. B. Elrefaie, and Y. Marghany, “Selection of a suitable mother wavelet for analyzing power system fault transients,” in *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pp. 1–7, 2008.
- [17] E. Cakir, E. C. Ozan, and T. Virtanen, “Filterbank learning for deep neural network based polyphonic sound event detection,” in *Neural Networks (IJCNN), 2016 International Joint Conference on*, pp. 3399–3406, IEEE, 2016.
- [18] N. Z., N. U., I. K., T. S., G. S., and E. D., “Learning filterbanks from raw speech for phone recognition,” *CoRR*, vol. abs/1711.01161, 2017.
- [19] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch, “Kernel pca and de-noising in feature spaces.,” in *NIPS*, vol. 11, pp. 536–542, 1998.
- [20] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [21] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction

- and data representation,” *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.
- [22] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [23] Y. Bengio and M. Monperrus, “Non-local manifold tangent learning,” *Advances in Neural Information Processing Systems*, vol. 17, no. 1, pp. 129–136, 2005.
- [24] T. DeVries and G. W. Taylor, “Improved regularization of convolutional neural networks with cutout,” *arXiv preprint arXiv:1708.04552*, 2017.
- [25] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” *arXiv preprint arXiv:1710.09412*, 2017.
- [26] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [28] P. Goupillaud, A. Grossmann, and J. Morlet, “Cycle-octave and related transforms in seismic signal analysis,” *Geoexploration*, vol. 23, no. 1, pp. 85–102, 1984.
- [29] A. Grossmann and J. Morlet, “Decomposition of hardy functions into square integrable wavelets of constant shape,” *SIAM journal on mathematical analysis*, vol. 15, no. 4, pp. 723–736, 1984.
- [30] I. Daubechies, A. Grossmann, and Y. Meyer, “Painless nonorthogonal expansions,” *Journal of Mathematical Physics*, vol. 27, no. 5, pp. 1271–1283, 1986.
- [31] S. Mallat, *A wavelet tour of signal processing*. Academic press, 1999.

- [32] R. Balestriero and H. Glotin, “Scattering decomposition for massive signal classification: from theory to fast algorithm and implementation with validation on international bioacoustic benchmark,” in *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*, pp. 753–761, IEEE, 2015.
- [33] S. Chopra and K. J. Marfurt, “Choice of mother wavelets in cwt spectral decomposition,” in *SEG Technical Program Expanded Abstracts 2015*, pp. 2957–2961, Society of Exploration Geophysicists, 2015.
- [34] C. D’Avanzoa, V. Tarantinob, P. Bisiacchib, and G. Sparacinoa, “A wavelet methodology for eeg time-frequency analysis in a time discrimination task,”
- [35] A. Venkitaraman, A. Adiga, and C. S. Seelamantula, “Auditory-motivated gammatone wavelet transform,” *Signal Processing*, vol. 94, pp. 608–619, 2014.
- [36] J. L. Flanagan, “Models for approximating basilar membrane displacement,” *Bell Labs Technical Journal*, vol. 39, no. 5, pp. 1163–1191, 1960.
- [37] V. Lostanlen and J. Andén, “Binaural scene classification with wavelet scattering,”
- [38] V. Lostanlen, *Opérateurs convolutionnels dans le plan temps-fréquence*. PhD thesis, Paris Sciences et Lettres, 2017.
- [39] L. Cohen, *Time-frequency Analysis: Theory and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1995.
- [40] M. Affi, M. Fassi-Fihri, A. Nassim, and R. Sidki, “Paul wavelet-based algorithm for optical phase distribution evaluation,” *Optics communications*, vol. 211, no. 1, pp. 47–51, 2002.

- [41] N. Y. Vilenkin, *Special Functions and the Theory of Group Representations*, vol. 22. American Mathematical Soc., 1978.
- [42] R. G. Baraniuk, “Shear madness: signal-dependent and metaplectic time-frequency representations.,” 1993.
- [43] I. Daubechies, *Ten Lectures on Wavelets*, vol. 61. Siam, 1992.
- [44] H. G. Feichtinger, W. Kozek, and F. Luef, “Gabor analysis over finite abelian groups,” *Applied and Computational Harmonic Analysis*, vol. 26, no. 2, pp. 230–248, 2009.
- [45] B. Torr sani, “Wavelets associated with representations of the affine weyl–heisenberg group,” *Journal of Mathematical Physics*, vol. 32, no. 5, pp. 1273–1279, 1991.
- [46] R. Rao and D. L. Ruderman, “Learning lie groups for invariant visual perception,” in *Advances in neural information processing systems*, pp. 810–816, 1999.
- [47] J. Sohl-Dickstein, C. M. Wang, and B. A. Olshausen, “An unsupervised algorithm for learning lie group transformations,” *arXiv preprint arXiv:1001.1027*, 2010.
- [48] T. B. Hashimoto, P. S. Liang, and J. C. Duchi, “Unsupervised transformation learning via convex relaxations,” in *Advances in Neural Information Processing Systems*, pp. 6875–6883, 2017.
- [49] Y. Bahroun, D. Chklovskii, and A. Sengupta, “A similarity-preserving network trained on transformed images recapitulates salient features of the fly motion detection circuit,” in *Advances in Neural Information Processing Systems*, pp. 14178–14189, 2019.

- [50] R. Cosentino and B. Aazhang, “Learnable group transform for time-series,” in *International Conference on Machine Learning*, vol. 119 of *Proceedings of Machine Learning Research*, (Virtual), pp. 2164–2173, PMLR, 13–18 Jul 2020.
- [51] C. Ick and V. Lostanlen, “Learning a lie algebra from unlabeled data pairs,” 2020.
- [52] C. M. Wang, J. Shol-Dickstein, I. Tasic, and B. A. Olshausen, “Lie group transformation models for predictive video coding,” in *2011 Data Compression Conference*, pp. 83–92, IEEE, 2011.
- [53] N. Kriegeskorte, M. Mur, D. A. Ruff, R. Kiani, J. Bodurka, H. Esteky, K. Tanaka, and P. A. Bandettini, “Matching categorical object representations in inferior temporal cortex of man and monkey,” *Neuron*, vol. 60, no. 6, pp. 1126–1141, 2008.
- [54] A. Sengupta, C. Pehlevan, M. Tepper, A. Genkin, and D. Chklovskii, “Manifold-tiling localized receptive fields are optimal in similarity-preserving neural networks,” in *Advances in Neural Information Processing Systems*, pp. 7080–7090, 2018.
- [55] B. Hall, *Lie Groups, Lie Algebras, and Representations: an Elementary Introduction*, vol. 222. Springer, 2015.
- [56] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial transformer networks,” in *Advances in Neural Information Processing Systems*, pp. 2017–2025, 2015.
- [57] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern*

- Recognition (CVPR)*, pp. 770–778, 2016.
- [58] M. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, “Deep learning of the tissue-regulated splicing code,” *Bioinformatics*, vol. 30, no. 12, pp. 121–129, 2014.
- [59] T. Cohen and M. Welling, “Group equivariant convolutional networks,” in *International Conference on Machine Learning*, pp. 2990–2999, 2016.
- [60] D. A. Ramli and H. Jaafar, “Peak finding algorithm to improve syllable segmentation for noisy bioacoustic sound signals,” *Procedia Computer Science*, vol. 96, pp. 100–109, 2016.
- [61] H. Glotin, J. Ricard, and R. Balestrieri, “Fast chirplet transform injects priors in deep learning of animal calls and speech,” 2017.
- [62] M. Trone, H. Glotin, R. Balestrieri, and D. E. Bonnett, “Enhanced feature extraction using the morlet transform on 1 mhz recordings reveals the complex nature of amazon river dolphin (*inia geoffrensis*) clicks,” *The Journal of the Acoustical Society of America*, vol. 138, no. 3, pp. 1904–1904, 2015.
- [63] S. Jaffard, Y. Meyer, and R. Ryan, *Wavelets: Tools for Science and Technology*. Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics, 2001.
- [64] R. Serizel, V. Bisot, S. Essid, and G. Richard, “Acoustic features for environmental sound analysis,” in *Computational Analysis of Sound Scenes and Events*, pp. 71–101, Springer, 2018.

- [65] R. Cosentino, R. Balestrieri, and B. Aazhang, “Best basis selection using sparsity driven multi-family wavelet transform,” in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 252–256, Dec 2016.
- [66] A. Megahed, A. M. Moussa, H. Elrefaie, and Y. Marghany, “Selection of a suitable mother wavelet for analyzing power system fault transients,” in *Power, Energy Society General Meeting-Conversion, Delivery of Electrical Energy in the 21st Century*, pp. 1–7, IEEE, 2008.
- [67] J. Andén and S. Mallat, “Deep scattering spectrum,” *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4114–4128, 2014.
- [68] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, “Learning the speech front-end with raw waveform cldnns,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
- [69] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, “Very deep convolutional neural networks for raw waveforms,” in *Acoustics, Speech and Signal Processing (ICASSP)*, pp. 421–425, IEEE, 2017.
- [70] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou, “Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network,” in *Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5200–5204, IEEE, 2016.
- [71] J. C. Brown, “Calculation of a constant q spectral transform,” *Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.
- [72] C. A. Shera, J. J. Guinan, and A. J. Oxenham, “Revised estimates of human

- cochlear tuning from otoacoustic and behavioral measurements,” *Proceedings of the National Academy of Sciences*, vol. 99, no. 5, pp. 3318–3323, 2002.
- [73] R. W. Clough, “Original formulation of the finite element method,” *Finite Elements in Analysis and Design*, vol. 7, no. 2, pp. 89–101, 1990.
- [74] C. Xu, C. Wang, and W. Liu, “Nonstationary vibration signal analysis using wavelet-based time–frequency filter and Wigner-Ville distribution,” *Journal of Vibration and Acoustics*, vol. 138, no. 5, p. 051009, 2016.
- [75] Y. Meyer, “Wavelets-algorithms and applications,” *Wavelets-Algorithms and applications Society for Industrial and Applied Mathematics Translation.*, 142 p., vol. 1, 1993.
- [76] R. Cosentino, R. Balestriero, R. Baraniuk, and A. Patel, “Overcomplete frame thresholding for acoustic scene analysis,” *arXiv preprint arXiv:1712.09117*, 2017.
- [77] M. A. Unser, “Ten good reasons for using spline wavelets,” in *Wavelet Applications in Signal and Image Processing V*, vol. 3169, pp. 422–432, International Society for Optics and Photonics, 1997.
- [78] I. J. Schoenberg, “On interpolation by spline functions and its minimal properties,” in *On Approximation Theory*, pp. 109–129, Springer, 1964.
- [79] C. A. Hall and W. W. Meyer, “Optimal error bounds for cubic spline interpolation,” *Journal of Approximation Theory*, vol. 16, no. 2, pp. 105–122, 1976.
- [80] B. A. Olshausen and D. J. Field, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images,” *Nature*, vol. 381, no. 6583, p. 607, 1996.

- [81] D. Stowell and M. D. Plumbley, “An open dataset for research on audio field recording archives: freefield1010,” *CoRR*, vol. abs/1309.5275, 2013.
- [82] T. Grill and J. Schlüter, “Two convolutional neural networks for bird detection in audio signals,” in *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*, (Kos Island, Greece), Aug. 2017.
- [83] F. J. Harris, “On the use of windows for harmonic analysis with the discrete Fourier transform,” *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978.
- [84] R. G. Baraniuk and D. L. Jones, “Wigner-based formulation of the chirplet transform,” *IEEE Transactions on signal processing*, vol. 44, no. 12, pp. 3129–3135, 1996.
- [85] C. Saritha, V. Sukanya, and Y. N. Murthy, “Ecg signal analysis using wavelet transforms,” *Bulg. J. Phys*, vol. 35, no. 1, pp. 68–77, 2008.
- [86] R. Balestriero*, R. Cosentino, H. Glotin, and R. Baraniuk, “Spline filters for end-to-end deep learning,” in *International Conference on Machine Learning*, pp. 364–373, 2018.
- [87] F. Lelandais and H. Glotin, “Mallat’s matching pursuit of sperm whale clicks in real-time using daubechies 15 wavelets,” in *New Trends for Environmental Monitoring Using Passive Systems, 2008*, pp. 1–5, IEEE, 2008.
- [88] L. Seydoux, N. M. Shapiro, J. de Rosny, F. Brenguier, and M. Landès, “Detecting seismic activity with a covariance matrix analysis of data recorded on seismic arrays,” *Geophysical Journal International*, vol. 204, no. 3, pp. 1430–1442, 2016.

- [89] R. R. Coifman and M. V. Wickerhauser, “Entropy-based algorithms for best basis selection,” *Information Theory, IEEE Transactions on*, vol. 38, no. 2, pp. 713–718, 1992.
- [90] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [91] R. Gribonval and E. Bacry, “Harmonic decomposition of audio signals with matching pursuit,” *IEEE Transactions on Signal Processing*, vol. 51, no. 1, pp. 101–111, 2003.
- [92] T.-P. Le and P. Argoul, “Continuous wavelet transform for modal identification using free decay response,” *Journal of sound and vibration*, vol. 277, no. 1-2, pp. 73–100, 2004.
- [93] J. Bruna, *Scattering representations for recognition*. PhD thesis, 2013.
- [94] E. Oyallon, S. Zagoruyko, G. Huang, N. Komodakis, S. Lacoste-Julien, M. B. Blaschko, and E. Belilovsky, “Scattering networks for hybrid representation learning,” *IEEE transactions on pattern analysis and machine intelligence*, 2018.
- [95] M. Ravanelli and Y. Bengio, “Interpretable convolutional filters with sincnet,” *arXiv preprint arXiv:1811.09725*, 2018.
- [96] N. Zeghidour, N. Usunier, I. Kokkinos, T. Schaiz, G. Synnaeve, and E. Dupoux, “Learning filterbanks from raw speech for phone recognition,” in *2018 IEEE international conference on acoustics, speech and signal Processing (ICASSP)*, pp. 5509–5513, IEEE, 2018.

- [97] H. Khan and B. Yener, “Learning filter widths of spectral decompositions with wavelets,” in *Advances in Neural Information Processing Systems*, pp. 4601–4612, 2018.
- [98] S. Mallat, *A Wavelet Tour of Signal Processing*. Elsevier, 1999.
- [99] C. Xu, C. Wang, and J. Gao, “Instantaneous frequency identification using adaptive linear chirplet transform and matching pursuit,” *Shock and Vibration*, vol. 2016, 2016.
- [100] S. Goldenstein and J. Gomes, “Time warping of audio signals,” in *cgi*, p. 52, IEEE, 1999.
- [101] G. Kerkycharian, D. Picard, *et al.*, “Regression in random design and warped wavelets,” *Bernoulli*, vol. 10, no. 6, pp. 1053–1105, 2004.
- [102] R. Gribonval, “Fast matching pursuit with a multiscale dictionary of gaussian chirps,” *IEEE Transactions on signal Processing*, vol. 49, no. 5, pp. 994–1001, 2001.
- [103] Y. Wang and Y.-C. Jiang, “Modified adaptive chirplet decomposition with application in isar imaging of maneuvering targets,” *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, p. 456598, 2008.
- [104] S. Mann and S. Haykin, “The chirplet transform: Physical considerations,” *IEEE Transactions on Signal Processing*, vol. 43, no. 11, pp. 2745–2761, 1995.
- [105] P. Flandrin, “Time frequency and chirps,” in *Wavelet Applications VIII*, vol. 4391, pp. 161–175, International Society for Optics and Photonics, 2001.

- [106] S. Mallat, “Understanding deep convolutional networks,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150203, 2016.
- [107] M. Korda and I. Mezić, “Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control,” *Automatica*, vol. 93, pp. 149–160, 2018.
- [108] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, “Understanding deep neural networks with rectified linear units,” *arXiv preprint arXiv:1611.01491*, 2016.
- [109] D. Yarotsky, “Error bounds for approximations with deep relu networks,” *Neural Networks*, vol. 94, pp. 103–114, 2017.
- [110] D. L. B. Jupp, “Approximation to data by splines with free knots,” *SIAM Journal on Numerical Analysis*, vol. 15, no. 2, pp. 328–343, 1978.
- [111] D. Stowell and M. D. Plumbley, “Framewise heterodyne chirp analysis of bird-song,” in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pp. 2694–2698, IEEE, 2012.
- [112] G. Al-Naymat, S. Chawla, and J. Taheri, “Sparsedtw: A novel approach to speed up dynamic time warping,” in *Proceedings of the Eighth Australasian Data Mining Conference-Volume 101*, pp. 117–127, Australian Computer Society, Inc., 2009.
- [113] P. Schäfer, “The boss is concerned with time series classification in the presence of noise,” *Data Mining and Knowledge Discovery*, vol. 29, no. 6, pp. 1505–1530, 2015.

- [114] Z. Wang, W. Yan, and T. Oates, “Time series classification from scratch with deep neural networks: A strong baseline,” in *2017 international joint conference on neural networks (IJCNN)*, pp. 1578–1585, IEEE, 2017.
- [115] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, “Time-series classification with cote: the collective of transformation-based ensembles,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2522–2535, 2015.
- [116] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, “The ucr time series classification archive,” July 2015.
- [117] D. Bertsimas, A. Orfanoudaki, and H. Wiberg, “Interpretable clustering: an optimization approach,” *Machine Learning*, pp. 1–50, 2020.
- [118] D. Greene and P. Cunningham, “Producing accurate interpretable clusters from high-dimensional data,” in *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 486–494, Springer, 2005.
- [119] Q. Ma, J. Zheng, S. Li, and G. W. Cottrell, “Learning representations for time series clustering,” *Advances in neural information processing systems*, vol. 32, pp. 3781–3791, 2019.
- [120] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, “Constrained k-means clustering with background knowledge,” 2001.
- [121] V. Estivill-Castro, “Why so many clustering algorithms: a position paper,” *ACM SIGKDD explorations newsletter*, vol. 4, no. 1, pp. 65–75, 2002.
- [122] J. E. Nam, M. Maurer, and K. Mueller, “A high-dimensional feature clustering approach to support knowledge-assisted visualization,” *Computers & Graphics*,

- vol. 33, no. 5, pp. 607–615, 2009.
- [123] J. Xie, R. Girshick, and A. Farhadi, “Unsupervised deep embedding for clustering analysis,” in *International Conference on Machine Learning*, pp. 478–487, 2016.
- [124] L. Seydoux, R. Balestriero, P. Poli, M. De Hoop, M. Campillo, and R. Baraniuk, “Clustering earthquake signals and background noises in continuous seismic data with unsupervised deep learning,” *Nature communications*, vol. 11, no. 1, pp. 1–12, 2020.
- [125] S. Dolnicar, “Using cluster analysis for market segmentation—typical misconceptions, established methodological weaknesses and some recommendations for improvement,” *Australasian Journal of Market Research*, vol. 11, no. 2, pp. 5–12, 2003.
- [126] R. Xu and D. C. Wunsch, “Clustering algorithms in biomedical research: a review,” *IEEE Reviews in Biomedical Engineering*, vol. 3, pp. 120–154, 2010.
- [127] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, (Berkeley, Calif.), pp. 281–297, University of California Press, 1967.
- [128] K. He, F. Wen, and J. Sun, “K-means hashing: An affinity-preserving quantization method for learning binary compact codes,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2938–2945, 2013.
- [129] B. J. Frey and N. Jojic, “Fast, large-scale transformation-invariant clustering,” in *Advances in Neural Information Processing Systems*, pp. 721–727, 2002.

- [130] B. Raytchev and H. Murase, “Unsupervised face recognition from image sequences based on clustering with attraction and repulsion,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, vol. 2, pp. II–II, IEEE, 2001.
- [131] M. Steinbach, L. Ertöz, and V. Kumar, “The challenges of clustering high dimensional data,” in *New Directions in Statistical Physics*, pp. 273–309, Springer, 2004.
- [132] A. Fitzgibbon and A. Zisserman, “On affine invariant clustering and automatic cast listing in movies,” in *European Conference on Computer Vision*, pp. 304–320, Springer, 2002.
- [133] P. Y. Simard, Y. A. LeCun, J. S. Denker, and B. Victorri, “Transformation invariance in pattern recognition—tangent distance and tangent propagation,” in *Neural networks: tricks of the trade*, pp. 235–269, Springer, 2012.
- [134] J. Lim, J. Ho, M.-H. Yang, K.-c. Lee, and D. Kriegman, “Image clustering with metric, local linear structure, and affine symmetry,” in *European Conference On Computer Vision*, pp. 456–468, Springer, 2004.
- [135] H. Murase and S. K. Nayar, “Learning and recognition of 3d objects from appearance,” in *Proceedings IEEE Workshop on Qualitative Vision*, pp. 39–50, 1993.
- [136] R. Basri, D. Roth, and D. Jacobs, “Clustering appearances of 3d objects,” in *Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 414–420, IEEE, 1998.

- [137] M.-C. Su and C.-H. Chou, “A Modified Version of the K-means Algorithm with a Distance Based on Cluster Symmetry,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 674–680, 2001.
- [138] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, “Clustering appearances of objects under varying illumination conditions,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, vol. 1, pp. I–I, IEEE, 2003.
- [139] N. S. Detlefsen, O. Freifeld, and S. Hauberg, “Deep diffeomorphic transformer networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4403–4412, 2018.
- [140] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, “An unsupervised learning model for deformable medical image registration,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9252–9260, 2018.
- [141] S. Lohit, Q. Wang, and P. Turaga, “Temporal transformer networks: Joint learning of invariant and discriminative time warping,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12426–12435, 2019.
- [142] A. V. Dalca, M. Rakic, J. Guttag, and M. R. Sabuncu, “Learning conditional deformable templates with convolutional networks,” *arXiv preprint arXiv:1908.02738*, 2019.
- [143] R. A. Shapira Weber, M. Eyal, N. Skafté, O. Shriki, and O. Freifeld, “Diffeomorphic temporal alignment nets,” in *Advances in Neural Information Processing*

- Systems*, vol. 32, pp. 6574–6585, Curran Associates, Inc., 2019.
- [144] M. Zhang and P. T. Fletcher, “Finite-dimensional lie algebras for fast diffeomorphic image registration,” in *International conference on information processing in medical imaging*, pp. 249–260, Springer, 2015.
- [145] O. Freifeld, S. Hauberg, K. Batmanghelich, and J. W. Fisher, “Highly-expressive spaces of well-behaved transformations: Keeping it simple,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2911–2919, 2015.
- [146] S. Durrleman, S. Allasonnière, and S. Joshi, “Sparse adaptive parameterization of variability in image ensembles,” *International Journal of Computer Vision*, vol. 101, no. 1, pp. 161–183, 2013.
- [147] S. Allasonnière, S. Durrleman, and E. Kuhn, “Bayesian mixed effect atlas estimation with a diffeomorphic deformation model,” *SIAM Journal on Imaging Sciences*, vol. 8, no. 3, pp. 1367–1395, 2015.
- [148] U. Grenander and E. Grenander, *General Pattern Theory: A Mathematical Study of Regular Structures*. Oxford Mathematical Monographs, Clarendon Press, 1993.
- [149] P. Dupuis, U. Grenander, and M. I. Miller, “Variational problems on flows of diffeomorphisms for image matching,” *Quarterly of applied mathematics*, pp. 587–600, 1998.
- [150] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

- [151] J. Duchon, “Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces,” *Revue Française d’Automatique, Informatique, Recherche Opérationnelle. Analyse Numérique*, vol. 10, no. R3, pp. 5–12, 1976.
- [152] F. L. Bookstein, “Principal warps: Thin-plate splines and the decomposition of deformations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 567–585, 1989.
- [153] M. Nejati, R. Amirfattahi, and S. Sadri, “A fast hybrid approach for approximating a thin-plate spline surface,” in *2010 18th Iranian Conference on Electrical Engineering*, pp. 204–208, IEEE, 2010.
- [154] B. S. Morse, T. S. Yoo, P. Rheingans, D. T. Chen, and K. R. Subramanian, “Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions,” in *ACM SIGGRAPH*, pp. 78–es, 2005.
- [155] S. Lloyd, “Least squares quantization in pcm,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [156] E. Klassen, A. Srivastava, M. Mio, and S. H. Joshi, “Analysis of planar shapes using geodesic paths on shape spaces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 372–383, 2004.
- [157] A. Srivastava, S. H. Joshi, W. Mio, and X. Liu, “Statistical shape analysis: Clustering, learning, and testing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 590–602, 2005.
- [158] A. Banerjee, S. Merugu, I. S. Dhillon, J. Ghosh, and J. Lafferty, “Clustering with bregman divergences,” *Journal of machine learning research*, vol. 6, no. 10, 2005.

- [159] F. Aurenhammer, “Voronoi diagrams—a survey of a fundamental geometric data structure,” *ACM Computing Surveys (CSUR)*, vol. 23, no. 3, pp. 345–405, 1991.
- [160] F. Aurenhammer, R. Klein, and D. Lee, *Voronoi diagrams and Delaunay triangulations*. World Scientific Publishing Company, 2013.
- [161] H. J. Johnson and G. E. Christensen, “Landmark and intensity-based, consistent thin-plate spline image registration,” in *Biennial International Conference on Information Processing in Medical Imaging*, pp. 329–343, Springer, 2001.
- [162] D. Letscher, “Vector weighted voronoi diagrams and delaunay triangulations,”
- [163] M. Inaba, N. Katoh, and H. Imai, “Applications of weighted voronoi diagrams and randomization to variance-based k-clustering,” in *Proceedings of the tenth annual symposium on Computational geometry*, pp. 332–339, 1994.
- [164] D. Arthur and S. Vassilvitskii, “k-means++: The advantages of careful seeding,” tech. rep., 2006.
- [165] A. Bhattacharya, R. Jaiswal, and N. Ailon, “Tight lower bound instances for k-means++ in two dimensions,” *Theoretical Computer Science*, vol. 634, pp. 55–66, 2016.
- [166] S. Har-Peled and B. Raichel, “On the complexity of randomly weighted voronoi diagrams,” in *Proceedings of the thirtieth annual symposium on Computational geometry*, pp. 232–241, 2014.
- [167] M. Xia and S. Aïssa, “Unified analytical volume distribution of poisson-delaunay simplex and its application to coordinated multi-point transmission,” *IEEE*

- Transactions on Wireless Communications*, vol. 17, no. 7, pp. 4912–4921, 2018.
- [168] R. Balestriero, R. Cosentino, B. Aazhang, and R. Baraniuk, “The geometry of deep networks: Power diagram subdivision,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [169] R. Balestriero, “Symjax: symbolic cpu/gpu/tpu programming,” *arXiv preprint arXiv:2005.10635*, 2020.
- [170] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang, “Image clustering using local discriminant models and global integration,” *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2761–2773, 2010.
- [171] S. Romano, J. Bailey, V. Nguyen, and K. Verspoor, “Standardized mutual information for clustering comparisons: one step further in adjustment for chance,” in *International Conference on Machine Learning*, pp. 1143–1151, PMLR, 2014.
- [172] L. Hubert and P. Arabie, “Comparing partitions,” *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.
- [173] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, “Variational deep embedding: An unsupervised and generative approach to clustering,” *arXiv preprint arXiv:1611.05148*, 2016.
- [174] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [175] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, “Deep convolutional autoencoder-based lossy image compression,” in *2018 Picture Coding Symposium (PCS)*,

- pp. 253–257, IEEE, 2018.
- [176] G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis, “Single-cell rna-seq denoising using a deep count autoencoder,” *Nature communications*, vol. 10, no. 1, pp. 1–14, 2019.
- [177] L. Tran, X. Liu, J. Zhou, and R. Jin, “Missing modalities imputation via cascaded residual autoencoder,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1405–1414, 2017.
- [178] D. Erhan, Y. Bengio, A. Courville, P. A. Manzagol, P. Vincent, and S. Bengio, “Why does unsupervised pre-training help deep learning?,” *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 625–660, 2010.
- [179] J. B. Tenenbaum, V. De Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [180] W. Wang, Y. Huang, Y. Wang, and L. Wang, “Generalized autoencoder: A neural network framework for dimensionality reduction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 490–497, 2014.
- [181] Y. Wang, H. Yao, and S. Zhao, “Auto-encoder based dimensionality reduction,” *Neurocomputing*, vol. 184, pp. 232–242, 2016.
- [182] E. Facco, M. d’Errico, A. Rodriguez, and A. Laio, “Estimating the intrinsic dimension of datasets by a minimal neighborhood information,” *Scientific reports*, vol. 7, no. 1, pp. 1–8, 2017.

- [183] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.
- [184] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, “Contractive autoencoders: Explicit invariance during feature extraction,” 2011.
- [185] A. Makhzani and B. Frey, “K-sparse autoencoders,” *arXiv preprint arXiv:1312.5663*, 2013.
- [186] L. Falorsi, P. de Haan, T. R. Davidson, N. De Cao, M. Weiler, P. Forré, and T. S. Cohen, “Explorations in homeomorphic variational auto-encoding,” *arXiv preprint arXiv:1807.04689*, 2018.
- [187] P. Li and P. M. Nguyen, “On random deep weight-tied autoencoders: Exact asymptotic analysis, phase transitions, and implications to training,” in *International Conference on Learning Representations*, 2019.
- [188] T. V. Nguyen, R. K. W. Wong, and C. Hegde, “On the dynamics of gradient descent for autoencoders,” in *Proceedings of Machine Learning Research*, vol. 89, pp. 2858–2867, PMLR, Apr 2019.
- [189] N. Lei, D. An, Y. Guo, K. Su, S. Liu, Z. Luo, S. Yau, and X. Gu, “A geometric understanding of deep learning,” *Engineering*, 2020.
- [190] A. Paul and S. Venkatasubramanian, “Why does deep learning work? a perspective from group theory,” *arXiv preprint arXiv:1412.6621*, 2014.
- [191] J. Bruna and S. Mallat, “Invariant scattering convolution networks,” *IEEE*

- transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.
- [192] X. Chen, X. Cheng, and S. Mallat, “Unsupervised deep haar scattering on graphs,” in *Advances in Neural Information Processing Systems*, pp. 1709–1717, 2014.
- [193] J. Andén, V. Lostanlen, and S. Mallat, “Joint time-frequency scattering for audio classification,” *CoRR*, vol. abs/1512.02125, 2015.
- [194] T. Cohen, M. Geiger, J. Köhler, and M. Welling, “Spherical CNNs,” *CoRR*, vol. abs/1801.10130, 2018.
- [195] R. Kondor and S. Trivedi, “On the generalization of equivariance and convolution in neural networks to the action of compact groups,” *arXiv preprint arXiv:1802.03690*, 2018.
- [196] R. Balestrierio and R. G. Baraniuk, “Mad max: Affine spline insights into deep learning,” *arXiv preprint arXiv:1805.06576*, 2018.
- [197] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, “A learning algorithm for boltzmann machines,” *Cognitive science*, vol. 9, no. 1, pp. 147–169, 1985.
- [198] G. Cottrell, P. Munro, and D. Zipser, “Image compression by back propagation: An example of extensional programming,” *ICS Report*, no. 8702, 1987.
- [199] J. L. Elman and D. Zipser, “Learning the hidden structure of speech,” *The Journal of the Acoustical Society of America*, vol. 83, no. 4, pp. 1615–1626, 1988.
- [200] P. Baldi and K. Hornik, “Neural networks and principal component analysis: Learning from examples without local minima,” *Neural networks*, vol. 2, no. 1,

- pp. 53–58, 1989.
- [201] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [202] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?,” in *2009 IEEE 12th international conference on computer vision*, pp. 2146–2153, IEEE.
- [203] S. Rifai, G. Mesnil, P. Vincent, X. Muller, Y. Bengio, Y. Dauphin, and X. Glorot, “Higher order contractive auto-encoder,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 645–660, Springer, 2011.
- [204] R. Balestriero and R. Baraniuk, “A spline theory of deep learning,” in *Proceedings of the 35th International Conference on Machine Learning*, pp. 374–383, 2018.
- [205] R. Balestriero, R. Cosentino, B. Aazhang, and R. Baraniuk, “The geometry of deep networks: Power diagram subdivision,” in *Advances in Neural Information Processing Systems*, pp. 15832–15841, 2019.
- [206] S. Mohan, Z. Kadkhodaie, E. P. Simoncelli, and C. Fernandez-Granda, “Robust and interpretable blind image denoising via bias-free convolutional neural networks,” *arXiv preprint arXiv:1906.05478*, 2020.
- [207] S. Wager, S. Wang, and P. S. Liang, “Dropout training as adaptive regularization,” in *Advances in neural information processing systems*, pp. 351–359, 2013.

- [208] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. A. Hamprecht, Y. Bengio, and A. Courville, “On the spectral bias of neural networks,” *arXiv preprint arXiv:1806.08734*, 2018.
- [209] M. Gamba, S. Carlsson, H. Azizpour, and M. Björkman, “Hyperplane arrangements of trained convnets are biased,” 2020.
- [210] T. Ergen and M. Pilanci, “Convex geometry of two-layer relu networks: Implicit autoencoding and interpretable models,” in *International Conference on Artificial Intelligence and Statistics*, pp. 4024–4033, 2020.
- [211] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, “On the number of linear regions of deep neural networks,” *Advances in neural information processing systems*, vol. 27, pp. 2924–2932, 2014.