On Interpreting Sonar Waveforms via the Scattering Transform

By

DAVID S. WEBER
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Applied Mathematics

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Naoki Saito

James Bremer

Thomas Strohmer

Committee in Charge

2021

Contents

On Interpreting Sonar Waveforms via the Scattering Transform

Abstract

The Scattering Transform (ST) is a formalization of some potential properties that have made convolutional neural networks effective at a wide variety of image and signal processing problems. Classifying raw side angle sonar (SAS) data provides an interesting test case for the scattering transform, since in addition to being a worthwhile problem in its own right, it is possible to model explicitly and understand how changes in the parameters of the model effect the resulting signal. In this dissertation we both apply the scattering transform to real and synthetic sonar classification problems, attempt to deepen our understanding of the scattering transform, and then apply that to understanding the sonar classifiers.

We use several methods to interpret the ST coefficients; the principal one is creating signals which maximize the output of a particular coefficient, balanced against the norm of the signal, which we call a pseudo-inversion. This turns out to be a difficult optimization problem, which we solve using differential evolution. We also use the gradient, as this can provide local information about the coefficient maximization, and theoretical properties of wavelets with vanishing moments. As the number of vanishing moments corresponds to the order of the wavelet as a pseudo-differential operator, this allows us to frame the scattering transform as mixing various orders of derivatives.

To try to understand the role of nonlinearity, in 2D we construct the shearlet scattering (or shattering) transform. We extend the sparsity guarantees of the shearlet transform to the shattering transform, and to do so we need to place some constraints on the possible nonlinearities. These constraints end up explaining the variation in classification results on the MNIST and FashionMNIST datasets.

We create synthetic sonar signals by varying the target object's shape and internal wave speed, which corresponds to the material composition. We examine some simple geometric properties, such as the relation between the signal delays and the wave speed, as well as a characterization of both rotation and translation of the target object in terms of how they modify the signal. When we compute the pseudo-inverse of the classification of objects with varying wave speed, the signal delay appears as an important discrimination feature. On the other hand, discriminating the shape of the object with a fixed speed is a subtler problem, where the shape with higher variation in curvature has more variation in the behavior of the tail, relying on higher second layer frequency when using Morlet wavelets.

Acknowledgments

There are some clichés that are clichés because they express simple, nearly universal truths: I'd like to thank my parents, my sister, the rest of my family, and my friends for their consistent support. I am not a strong enough person to have written this without them.

I would like to thank to Professor Saito, for taking a chance on this project, and for his patient and tireless attention to detail, without which this manuscript may have remained on the border of comprehensibility. I would like to thank Professors James Bremer and Thomas Strohmer for being on my dissertation committee, and their patience with the delays.

I would like to thank Jeffery Steward for his help in finding alternative optimization methods, even as we went beyond gradient based methods with he was most comfortable with. I would like to thank Kaela Vogel, Linda Bond, and Wade Edwards for their company and support during the COVID-19 pandemic. It has been a wild year, and having their presence through this process has kept me sane.

I would like to thank Mark Goldman and Tim Lewis, for helping me find my way initially in graduate school. There is another version of the past 6 years where I would have written a dissertation with their advising.

As for funding, this research was supported in part by the grants from ONR N00014-12-1-0177, N00014-16-1-2255, and N00014-20-1-2381, and from NSF DMS-1418779, IIS-1631329, and CCF-1934568. I would like to thank Frank Crosby and Julia Gazagnaire of Naval Surface Warfare Center, Panama City, FL, for providing Professor Saito with the real BAYEX14 dataset, and Professor Saito's former intern Vincent Bodin (now at Sinequa) and postdoctoral researcher Ian Sammis (now at Google) who first generated the synthetic dataset, and along with Professor James Bremer for the development of the fast Helmholtz equation solver.

I would like to thank the whole Julia community for their spirit of open source development. There were many github issues raised in the process of writing this dissertation, and I certainly could not have solved all of them myself. Particularly relevant are the Flux.jl team, the GLMNet.jl team, the Wavelets.jl team, the BlackBoxOptim.jl team, and the Shearlab.jl team. Finally, I would like to thank the Fortran GLMNet team: Jerome Friedman, Trevor Hastie, Rob Tibshirani and Noah Simon (Stanford). I would like to thank Stephane Mallat and his group (ENS, France) for their ST codes.

And finally, I want to thank Sir Frederick Banting and J.J.R. Macleod, without whose discovery of insulin I would have been dead at 19. I am eternally grateful to them and their successors who have made living, and eventually thriving, with diabetes actually possible.

CHAPTER 1

Introduction

Deep neural networks, and convolutional neural networks in particular, have proven quite effective at discerning hierarchical patterns in large datasets [**49**]. Some examples include image classification [**46**], face recognition [**72**], and speech recognition [**21**], among many others. Clearly, something is going very right in the design of CNNs. However, the principles that account for this success are still somewhat elusive [**57**], as is the construction of systems that work well with few examples.

The scattering transform (ST) was created to remedy these issues. In 2012, Stéphane Mallat and Joan Bruna published both theoretical results [**56**] and numerical implementations [**10**] tying together convolutional neural networks and wavelet theory. They demonstrated that scattering transforms of wavelets and modulus nonlinearities, are translation invariant in the limit of infinite scale, and Lipschitz continuous under non-uniform translation, i.e. $T_\tau(f)(x) = f\big(x - \tau(x)\big)$ for τ with bounded gradient. Numerically, they achieved state of the art on image and texture classification problems.

More recent work from Wiatowski and Bölcskei have generalized the Lipschitz continuity result from wavelet transforms to frames, and more importantly, established that increasing the *depth* of the network also leads to translation invariant features [**77**]. We detail this approach to the scattering transform in Chapter 3. There have been a number of related papers, including a discrete version of Wiatowski's result [**78**], and a related method on graphs [**59**]. There have also been a number of papers using the scattering transform in such problems as fetal heart rate classification [**15**], age estimation from face images [**12**], and voice detection in the presence of transient noise [**25**].

Throughout this dissertation, we make extensive use of the Julia programming language[1] [**7**], which is designed as both a readable high-level language like Python or MATLAB that maintains the performance results of a lower level compiled language like C or Fortran. Using Julia and building off of the machine learning and automatic differentiation platform Flux[2] [**38**], we have constructed a differentiable scattering

[1] https://julialang.org/
[2] https://fluxml.ai/

transform ScatteringTransform.jl which allows for substituting various frames into the framework easily, along with supporting either multi-threaded or GPU based computing. This dissertation fits into the tradition of reproducible research, and the code to produce this dissertation, along with all of the figures and results can be found at this gitlab.com repository[3].

The filters used in the scattering transform have a rich set of interpretations, unlike a CNN. However, there is still some ambiguity introduced by the presence of the nonlinearities, subsampling, and averaging. Most previous work to specifically interpret what particular scattering transform output tells us about the input domain consists in determining the scattering transform's invertibility [**5**, **18**, **76**]. Working in a slightly different direction, in Chapter 4 we use both a pseudo-inverse in Section 4.2 and the theoretical properties of wavelets in Section 4.3 to develop more of an understanding of what information the ST coefficients are capturing.

Building off of Wiatowski and Bölcskei's result, which work for general frames and not just wavelets specifically, there have been extensions to Gabor systems, with specific proof of the decay properties [**19**]. There have also been some applied results building off of this work for music signals [**6**, **34**]. In a similar vein, in Chapter 5 we examine the utility and properties of a scattering transform based off of the shearlet transform [**47**]. In Section 5.2, we demonstrate that it has the same theoretical sparsity guarantees as the original shearlet transform, and in Section 5.3 we demonstrate that it performs at a comparable level to the 2D Morlet scattering transform implemented in Kymatio [**4**] on MNIST and FashionMNIST [**80**].

The problem of interpreting sonar signals is a difficult one, particularly for smaller objects such as unexploded ordinance (UXOs) [**43**]. Sonar signals offer a rich dataset which has a clear physical generation process that can be simulated [**8**] to understand the role of various parameters, such as shape and material properties; we describe the simulation in Section 6.2. This also allows us to show that the invariants of these classes fit within the framework of the scattering transform in Section 6.3. We test this empirically in Section 6.4, both on synthetic data and real data. Finally, we use the pseudo-inversion methods developed in Section 4.2 to interpret the ST coefficients used in a logistic regression classifier. But before any of that, we review the core tools of harmonic and wavelet analysis.

[3] https://gitlab.com/dsweber2/dissertation

CHAPTER 2

Linear Transforms

2.1. The Fourier Transform and Linear Feature Extractors

The Fourier transform is so well known that it hardly needs introduction. But for the sake of emphasis, we can think of the Fourier transform $\mathscr{F}$ as a linear operator that takes a function $f \in \mathscr{L}^1(\mathbb{R})$ and returns another function $\widehat{f} \in \mathscr{L}^\infty(\mathbb{R})$ through the transform

$$\widehat{f}(\omega) \coloneqq \mathscr{F}(f)(\omega) \coloneqq \int_{-\infty}^{\infty} f(x) \mathrm{e}^{-\mathrm{i}\omega x} \, \mathrm{d}x.$$

The resulting complex function $\widehat{f}(\omega) = a(\omega)\mathrm{e}^{\mathrm{i}\phi(\omega)}$, gives the amplitude a and phase ϕ of oscillations with frequency ω. When $\widehat{f}$ is also in $\mathscr{L}^1(\mathbb{R})$, then inverse is well defined and almost identical, up to a sign and a constant:

$$f(x) = \mathscr{F}^{-1}\big(\widehat{f}\big)(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(\omega) \mathrm{e}^{\mathrm{i}\omega x} \, \mathrm{d}\omega.$$

The Fourier transform simplifies some otherwise complicated operations: the convolution of two functions f and g becomes pointwise multiplication $\widehat{f \star g} = \widehat{f} \cdot \widehat{g}$, while differentiation becomes multiplication by a monomial $\widehat{\frac{\mathrm{d}}{\mathrm{d}x} f} = \mathrm{i}\omega \widehat{f}$, among others, see [**30**, **71**].

This last property ties together the smoothness of f and the decay rate of $\widehat{f}$. Specifically, if f is up to k times differentiable, then $\omega^k \widehat{f} \in \mathscr{L}^1(\mathbb{R})$ so $\widehat{f}$ must decay at least at a rate of $O\big(\ \omega\ ^{-k}\big)$. This means that smooth signals are quite sparse in frequency; see for example Fig. 2.1, where on a log scale the smooth signal rapidly decays, while an otherwise smooth function with 2 discontinuities decays quite slowly. A similar result holds for the Fourier series [**30**].

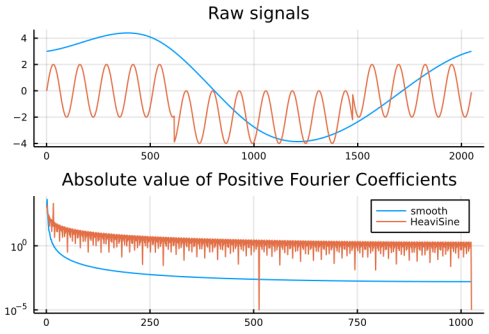


FIGURE 2.1. Comparing decay rates

It is also the canonical example in the family linear feature extractors that are *relatively invariant* to translation, meaning that for the translation operator $T_t(f)(x) \coloneqq f(x-t)$, the price for no longer applying the translation operator is multiplication by a function of translation distance t; to be precise, $\mathscr{F}\big(T_t(f)\big)(\omega) = \mathrm{e}^{\mathrm{i}t\cdot\omega}\,\widehat{f}(\omega)$ [**2**,**62**]. Any linear feature extractor that is relatively translation invariant can be considered as a linear sum over various frequencies. The same work by Amari and Otsu demonstrates that the only absolutely invariant *linear* transforms are the moments, which is a very restrictive class. This motivates our use of the absolute value of the Fourier transform (or AVFT) throughout this dissertation as the simplest example of a nonlinear (relatively) translation invariant transform.

Fig. 2.1 demonstrates one of the potential pitfalls of using just the Fourier transform; away from the two points of discontinuity, the HeaviSine is a single low frequency sinusoid, yet most of the Fourier domain is plagued by noise at all frequencies. The issue with the Fourier transform alone is that it only captures global frequency behavior, which is easily derailed by local aberrations. Capturing both local discontinuities and behavior at different scales is a reason for using wavelet transforms.

2.2. Continuous Wavelet Transforms

A *continuous wavelet transform* is generated through convolutions with scaled and translated copies of a *mother wavelet* ψ. Any function $\psi \in \mathscr{L}^2(\mathbb{R})$ which satisfies

$$\int_{-\infty}^{\infty} \psi(t)\,\mathrm{d}t = 0 \tag{2.1}$$

$$C_\psi = \int_{-\infty}^{\infty} \frac{\widehat{\psi}(\omega)^{\,2}}{\omega}\,\mathrm{d}\omega < \infty \tag{2.2}$$

can be used, though there are plenty of motivations for particular choices [**22**, **42**, **55**]. The corresponding wavelet transform $W[f](s,t)$ at scale $s \in \mathbb{R}^+$ and time $t \in \mathbb{R}$ is given by

$$W[f](s,t) \coloneqq f \star \psi_s(t) = \big(s\big)^{-p} \int_{-\infty}^{\infty} f(u)\psi\Big(\frac{u-t}{s}\Big)\,\mathrm{d}u \tag{2.3}$$

$$\psi_s(u) \coloneqq \big(s\big)^{-p}\psi\big(u/s\big),$$

with $p \in (1,\infty)$ and the convention that $p' = {}^{p}/_{p-1} = 1$ when $p = \infty$. Note that the choice of p determines which norm of ψ_s is preserved as the scale s varies [**42**, Section 3.1].[1] The norms preserved by the time and frequency domain are duals, since $\mathscr{F}\left[f({}^{t}/_{s})\right](\omega) = s\,\widehat{f}(s\omega)$, so we have that for any $s > 0$,

$$|\psi_s|_p = |\psi|_p,$$

$$|\widehat{\psi_s}|_{p'} = |\widehat{\psi}|_{p'}.$$

We will generally choose $p = 1$, which fixes the maximum value in the Fourier domain, as this prevents any one frequency from outweighing the others and makes inversion more stable. One of the nice properties of the Fourier transform outlined above was its relative translation invariance. While the continuous wavelet transform isn't relatively translation invariant, it is translation covariant, meaning that $W[T_a f](s,t) = T_a W[f](s,t)$. This is not as robust as relative invariance, but will serve as the basis for making an increasingly translation invariant transform with increasing depth in Chapter 3.

If f is a real-valued signal, $\widehat{f}(-\omega) = \overline{\widehat{f}(\omega)}$, so in principle we only need either the positive or negative frequencies to fully represent or reconstruct f. It is useful at this point to draw a distinction between two broad class of continuous wavelet transforms, depending on how the transform addresses this redundancy.

Real wavelets $\psi(t) \in \mathbb{R}$ address this by having exactly the same redundancy as the target signal class. Then C_ψ from Eq. (2.2) can be split into two equal halves coming from from the positive and negative halves of the integral, and then for any $f \in \mathscr{L}^2(\mathbb{R})$ the wavelet transform is invertible, and is given by

$$f(t) = \int_{-\infty}^{\infty}\int_{0}^{\infty} W[f](s,t)\psi\Big(\frac{u-t}{s}\Big)\,\frac{\mathrm{d}s\,\mathrm{d}u}{C_\psi s^{3-2p}}, \tag{2.4}$$

where equality is in a weak sense [**42**, Theorem 3.1]. Real wavelets are better adapted to characterizing discontinuous signals, as will be discussed more thoroughly in Section 2.3.1.

The second class is *analytic wavelets*, which are complex valued but only non-zero for non-negative frequencies. Then $W[f](s,t) = W[f_a](s,t)$, where f_a is the *analytic part* of a signal f, defined by $\widehat{f_a}(\omega) = \chi_{[0,\infty)}\widehat{f}(\omega)$, where χ_A is the characteristic function so that for $\omega \in A$, $\chi_A(\omega) = 1$ and $\chi_A(\omega) = 0$ otherwise. The analytic part of a signal can be uniquely[2] decomposed into an amplitude or *envelope* $f_a(t)$ and *instantaneous phase*

[1] note that we have chosen the opposite convention for p from ContinuousWavelets.jl for ease of presentation and to correspond to [**42**].

[2] There are ambiguities created when the amplitude passes through zero, see [**16**].

$\arg\big(f_a(t)\big)$. This allows for a unique definition of *instantaneous frequency* as the derivative of the instantaneous phase [**55**, Section 4.4.2] [**75**]. Analytic wavelets facilitate investigating the instantaneous frequency and amplitude of a function [**53**]. The reconstruction is quite similar to the real case:

$$f(t) = \int_{-\infty}^{\infty}\int_{0}^{\infty} \mathfrak{Re}\Big(W|f](s,t)\psi\Big(\frac{u-t}{s}\Big)\Big)\,\frac{\mathrm{d}s\,\mathrm{d}t}{C_\psi s^{3-2p}}, \tag{2.5}$$

from [**22**, Section 2.4.8]. There are some further complications dealing with f complex and variations on Eq. (2.2); for more on those, see either [**22**, Section 2.4] or [**42**, Section 3.2]. There is also a useful extension to the discrete case covered in [**42**, Chapter 6] or [**14**, Section 11.2].

When s and t are discretized, the resulting wavelet transform is an example of the more general class of frame transforms. A *frame* is a sequence of functions $\{f_k\}_{k=1}^{\infty} \subset \mathscr{L}^2(\mathbb{R})$ which satisfy the frame bounds

$$A|\,f\,|^2 \le \sum_{k=1}^{\infty} \langle f, f_k\rangle^{\,2} \le B|\,f\,|^2,$$

for some $A, B > 0$ and any function $f \in \mathscr{L}^2(\mathbb{R})$; we will reserve "the" frame bounds for the optimal such bounds [**14**, Chapter 6]. A frame can be thought of as a redundant basis which allows for multiple representations for a function. This allows for additional features, such as robustness to noise [**14**, Section 5.9], increased sparsity of the representation (see Section 2.3), and some invariance to transformations by bounded linear operators [**14**, Section 5.3.1]. They also allow for a diversity of inversion methods beyond the canonical dual frames found in Eq. (2.5); as we don't use the inversions in this dissertation, we will not go into that theory further, but it can be found in [**14**].

2.2.1. Examples. In Fig. 2.2 we have some example mother wavelets of more general wavelet families, as defined in the ContinuousWavelets.jl Julia package.[3] Only the first two families are (approximately) analytic, while the rest are real. The first three are described succinctly by Torrence and Compo [**74**]. In the time domain, the *Morlet wavelet family* is a Gaussian that is modulated by a complex exponential with frequency parameter ω_0:

$$\psi(t) \propto \mathrm{e}^{-t^2/2}\Big(\mathrm{e}^{i\omega_0 t} - \kappa\Big) \xrightarrow{\mathscr{F}} \widehat{\psi}(\omega) \propto \mathrm{e}^{-\frac{1}{2}(\omega-\omega_0)^2} - \kappa\mathrm{e}^{-\frac{1}{2}(\omega)^2}$$

[3] https://ucd4ids.github.io/ContinuousWavelets.jl/dev/

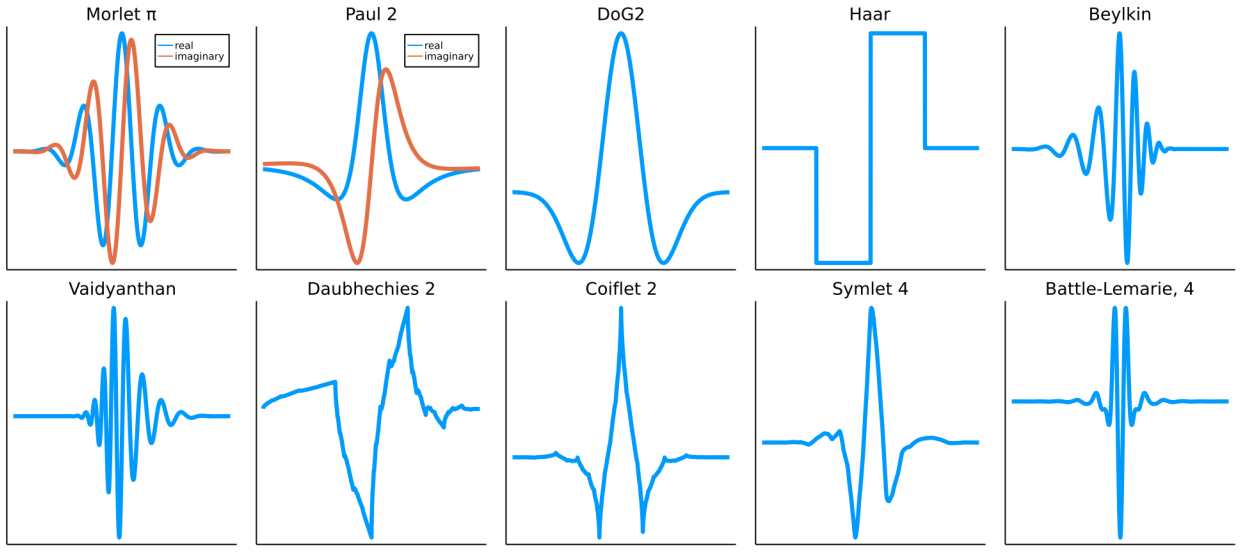


FIGURE 2.2. The various mother wavelets defined in ContinuousWavelets.jl[3]

where κ is chosen so that ψ satisfies the zero average condition Eq. (2.1). In Fig. 2.2, $\omega_0 = \pi$. ω_0 determines the trade-off between high spatial accuracy (ω_0 small) and high frequency accuracy (ω_0 large). We have defined $\psi(t)$ only up to a constant, since the constant will be chosen so that the entire wavelet family satisfies certain frame bounds (usually $B = 1$); this convention carries through for the rest of the wavelets. The Cauchy (aka Paul, or Klauder) wavelets are a classic example with a polynomial rate of decay m:

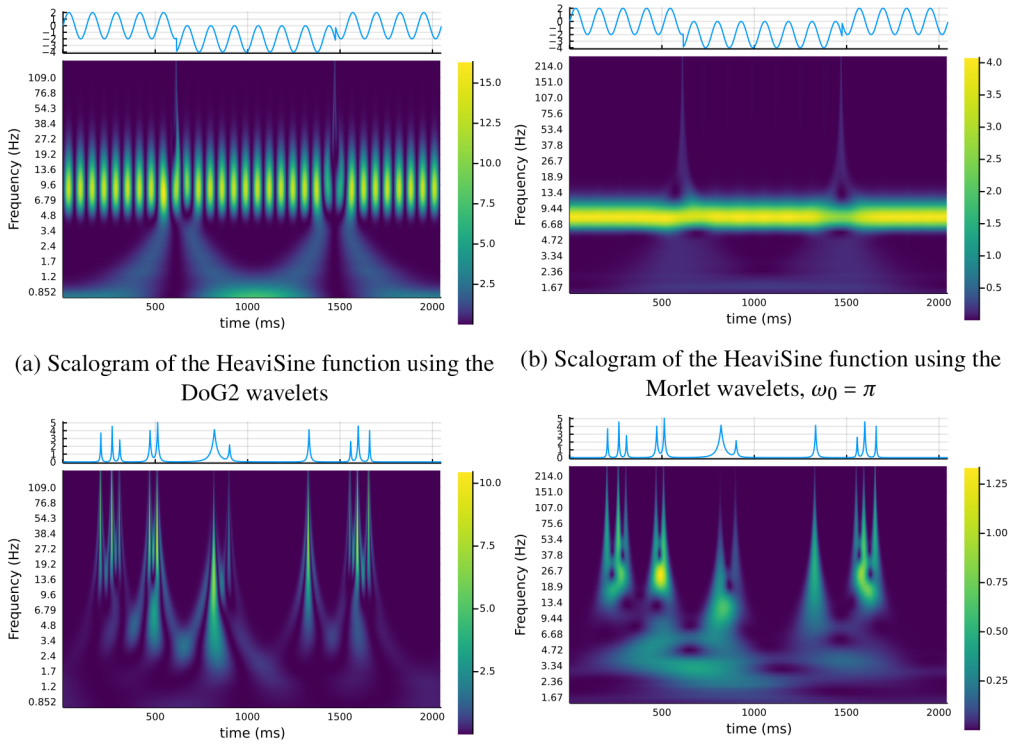
$$\psi(t) = \mathrm{i}^m \left(\frac{1}{1 - \mathrm{i}t} \right)^{m+1} \xrightarrow{\mathscr{F}} \widehat{\psi}(\omega) \propto H(\omega) \omega^m \mathrm{e}^{-\omega}$$

where $H(\omega) = \chi_{[0,\infty)}$ is the Heaviside function. Finally, the mth derivative of Gaussian (or DoGm) wavelets are simply that:

$$\psi(t) = -\left(-\frac{\mathrm{d}}{\mathrm{d}t} \right)^m \mathrm{e}^{-t^2/2} \xrightarrow{\mathscr{F}} \widehat{\psi}(\omega) \propto \left(\mathrm{i}\omega \right)^m \mathrm{e}^{-\omega^2/2}.$$

That they are derivatives of the Gaussian function will guarantee some nice properties for the decay rate for DoG wavelets, as will be discussed in Section 2.3. Both the Paul wavelets and a complex version of the DoG wavelets can be thought of as special cases of the analytic *Morse wavelets*, which we will not get into here, but are well described by Lilly and Olhede [**53**, **61**]. The other seven wavelets are continuous versions of classic orthonormal real wavelets, in this case generated using Wavelets.jl. There is no explicit formula for most of them (excluding the Haar wavelet), and they are derived by the Cascade Algorithm from

the corresponding quadrature mirror filters [**22**, Section 6.5]. The number in the titles of Fig. 2.2 for these refer to the number n of *vanishing moments*, that is for $m \leq n$ we have that $\langle t^m, \psi \rangle = 0$, a property further discussed in Section 2.3. For a more thorough explanation of each, see Daubechies's "Ten Lectures" [**22**], specifically Section 6.4 for the Daubechies, Haar and Symlet wavelets,[4] Section 8.2 for Coiflets, Section 5.4 for Battle–Lemarie wavelets. The Beylkin wavelets and Vaidyanathan wavelets can be found in [**79**].



(a) Scalogram of the HeaviSine function using the DoG2 wavelets

(b) Scalogram of the HeaviSine function using the Morlet wavelets, $\omega_0 = \pi$

(c) Scalogram of the Bumps function using the DoG2 wavelets

(d) Scalogram of the Bumps function using the Morlet wavelets, $\omega_0 = \pi$

FIGURE 2.3. Various continuous wavelet transforms

In Fig. 2.3 we have various *scalograms*, which are defined by $|W[f](s,t)|^2$, where the color is on a logarithmic scale. The y-axis, while labeled by increasing frequency, is uniform in the *sampled* scales, or inverse frequency. We will use this convention for scalograms throughout this dissertation. In Fig. 2.3a and Fig. 2.3b, we have the wavelet transform of the same HeaviSine function in Fig. 2.1 with the second derivative of Gaussian (DoG2) and Morlet wavelets, respectively. Both of these are much sparser than the Fourier transform of the same signal was, with the Morlet wavelet having a better concentration on the correct frequency of 8Hz.

[4]The Haar wavelets are exactly the Daubechies 1 wavelets

The discontinuity is visible in both, though very faint for the Morlet wavelet. On the other hand the bumps function, which is a collection of peaks with different rates of decay, has clearer "cones of influence" and spatial localization using the DoG2 wavelets in Fig. 2.3c than in Fig. 2.3d. Section 2.3.2 will discuss the relationship between the Lipschitz constant and the decay rate of the coefficients.

2.2.2. Effective Quality factor calculation. As we move from the ideal continuous case to something we can actually implement in a discrete case, we need to sample the scale parameter s in some manner. Throughout this section, the *wavelet index* $k \in \{1, \dots, K\}$ will denote the discrete set of wavelets used, with resulting scales $s_1, \dots, s_K$. As discussed in [**42**, Chapter 6], the default is to use a uniform sampling of the form $s_k = \sigma^k$ for $k \in \mathbb{Z}$; a common rephrasing of this is to consider octaves of the form $2^{k/Q}$ for some choice of *quality factor* Q. A reasonable default, especially for music signals, is to use $Q = 8$ [**3**].

Keeping a fixed number Q of wavelets per octave leads to a denser coverage of the low frequencies than is necessary in many applications, such as audio or sonar processing. Consequently, one needs to choose some method of reducing the number of such low frequency wavelets. One common choice is to choose a particular scale of wavelet and then translate this in the frequency domain. This leads to a collection of wavelets that are no longer a scaling of a single function.

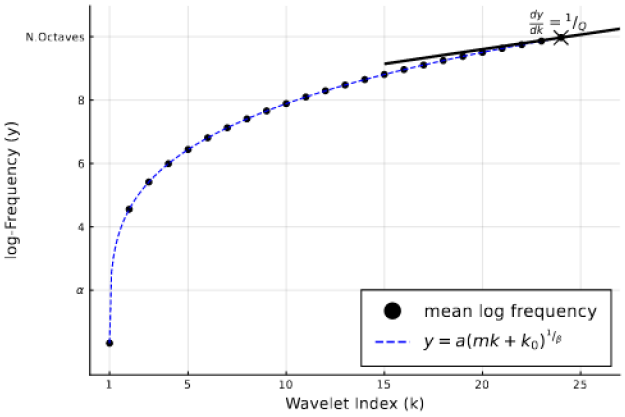


FIGURE 2.4. An example of the mean log-frequencies for some wavelets, where $\beta = 4$ and $Q = 8$

Instead of simply having scales distributed log-linearly as $2^{k/Q}$, we distribute them according to $s_k = 2^{a(mk+k_0)^{1/\beta}}$, or according to a βth-root in the log frequency, as in Fig. 2.4. If β is 1, then we have a linear relation between the index and the log-frequency, and Q gives exactly the number of wavelets per octave throughout. As β increases, the wavelets skew more and more heavily to high frequencies. To choose the parameters a, m and k_0 for a given frequency skew β, quality factor Q, and number of octaves α covered by the averaging function (see Section 2.2.3), there are a couple of criteria:

- The first wavelet is scaled by 2^α, so the curve $a(mk + k_0)^{1/\beta}$ goes through the point $(k, y) = (1, \alpha)$.

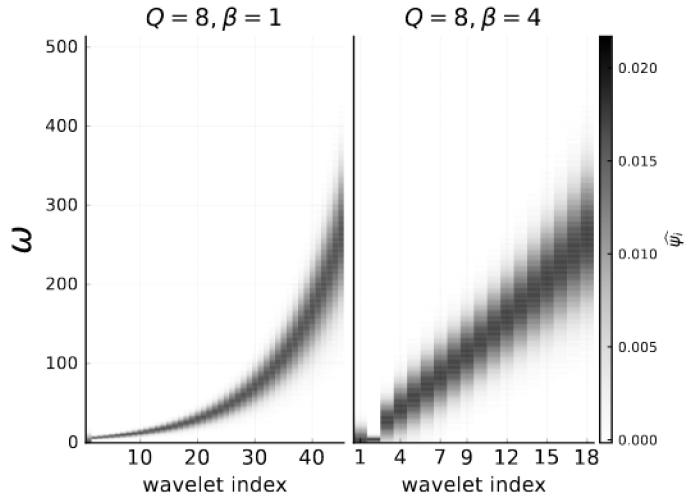


FIGURE 2.5. A comparison between the resulting Fourier domain wavelets for different values of β. Note that there are 25 filters with mean frequency below 50 in the first plot with $\beta = 1$, rather than 2 in the second where $\beta = 4$

- The derivative $\frac{\mathrm{d}y}{\mathrm{d}k}$ at the last point is $\frac{1}{Q}$, so the "instantaneous" number of wavelets k per octave y is Q. Each type of wavelet has a maximum scaling $2^{N_{Octaves}}$, so the final point N_w satisfies both $y(N_w) = N_{Octaves}$ and $y'(N_w) = 1/Q$.
- Finally, the spacing m is chosen so that there are exactly Q wavelets in the last octave.

As a simple example of the resulting wavelets, see Fig. 2.5; if $N = 2^9 = 512$, $Q = 8$, $a = 2$, and $\beta = 1$, we have 45 wavelets, whereas if $\beta = 4$, we have just 21.

2.2.3. The Averaging Function and Multiresolution Analysis. On their own, we need an infinite collection of wavelets to cover the coarse scale as $s \to \infty$ or $\omega \to 0$, and the mean (zero frequency) is completely uncovered even in the limit (this is part of the downside of the weak nature of the convergence). A way of remedying this is to introduce the *father*, or *averaging*, or *scaling wavelet* [**55**, Section 4.3.1]. This is specifically designed to cover the low frequency information missed by the mother ψ at scale 1:

$$\widehat{\phi}(\omega)^2 := \int_1^\infty \frac{\widehat{\psi}(s\omega)^2}{s^{3-2p}}\,\mathrm{d}s,$$

where we allowed to choose the phase with impunity (adapted from [**55**, Section 4.3.1]). This allows us to cap off s in either of the inversion formulas Eq. (2.4) or Eq. (2.5) to the interval $]0, s_0]$.

Through careful choices for the father ϕ and ψ and looking at the subspaces generated by their translates, one can create nested families of orthogonal bases, eventually spanning $\mathscr{L}^2(\mathbb{R})$, as discussed in Mallat's *Wavelet Tour of Signal Processing* [**55**, Chapter 7], Daubechies's *Ten Lectures on Wavelets* [**22**, Chapter 5], and Kaiser's *Friendly Guide to Wavelets* [**42**, Chapter 7]. These can be used to create a fast wavelet transform akin to the fast Fourier transform. Unfortunately, this beautiful theory is for the most part too far afield to spend much time exploring further. The principal problem with the orthogonal wavelet transforms for our purpose is that they are not even translation covariant. For finite signals, shifting the input by a single index results in wildly different wavelet coefficients [**66**], meaning that they cannot be used as coefficients for signal classification where translation invariance must be guaranteed.

2.3. Distributional Derivatives, Vanishing Moments, and Wavelet Decay

One way to extend the Fourier transform to functions f not in $\mathscr{L}^1(\mathbb{R})$ is to treat f as a tempered distribution, so that f itself is an operator on the nicer class of Schwartz functions $\mathscr{S}$ [**30**]. A Schwartz function $\varphi \in \mathscr{S}$ is an infinitely smooth function $\varphi \in \mathscr{C}^\infty(\mathbb{R})$, whose derivatives decay more quickly than any monomial, so $\sup_{x\in\mathbb{R}} x^\alpha \varphi^{(\beta)}(x) < \infty$. Schwartz functions are those functions nice enough so that the Fourier transform and its inverse maps the space to itself, so that $\varphi \in \mathscr{S} \Leftrightarrow \widehat{\varphi} \in \mathscr{S}$ [**30**]. This means that we can define a whole host of operations on our distribution f that may not be well defined on the corresponding function by moving that operation from f to φ. For the Fourier transform, a function $g \in \mathscr{L}^1(\mathbb{R})$, satisfies $\langle \widehat{g}, \varphi\rangle = \langle g, \widehat{\varphi}\rangle$, so for the case of the tempered distribution f with corresponding operator $F[\varphi] = \langle f, \varphi\rangle$, we define the Fourier transform of f so that $\widehat{F}[\varphi] = \langle \widehat{f}, \varphi\rangle \coloneqq \langle f, \widehat{\varphi}\rangle$. For this to be well behaved, we will need that F is of finite order, that is there's some $N \in \mathbb{N}$ and $C > 0$ so that

$$\langle f, \varphi\rangle \leq C \sum_{\alpha+\beta\leq N} \sup_x x^\alpha \varphi^{(\beta)}(x) ,$$

from [**30**]. We can use the same idea to take derivatives of general distributions using integration by parts, so the derivative of a distribution f is defined by the relation $\langle f', \varphi\rangle \coloneqq -\langle f, \varphi'\rangle$.[5] For example, the distributional derivative of the step function $\chi_{(0,\infty)}(x)$ is the delta "function", defined by $\int \delta(t)\varphi(t)\,\mathrm{d}t = \varphi(0)$. In fact, any operator G which has an adjoint G^*, meaning that $\langle Gf, g\rangle = \langle f, G^* g\rangle$, can effectively be defined to act on a distribution f by applying its adjoint to some class of test functions preserved by G. A couple of significantly

[5] This is well defined for a larger class of distributions than just tempered, and only needs the test functions to be smooth instead of Schwartz functions.

simpler examples than the above that are relevant for wavelets are translation T_t, which has an adjoint T_{-t}; scaling $S_s[f](t) = f(t/s)$, which has an adjoint $S_{1/s}$; convolution with a function $\star g$, which has an adjoint that is also convolution with $\overline{g(-t)}$; and finally multiplication by a smooth function g, which is self-adjoint.

2.3.1. Wavelets with Vanishing Moments. There is a whole class of wavelets which can be characterized as taking the distributional derivative of the input f they are transforming, that is

$$W[f](s,t) = (f \star \psi_s)(t) = (s\partial)^n (f \star \theta_s)(t) = \langle \partial^n f, \overline{\theta_s} \rangle.$$

To make this more precise, we need a couple of preliminary definitions. The first is fast decay: a function g is said to have *fast decay* if for every m, there is some C_m so that

$$g(t) \leq \frac{C_m}{1 + |t|^m}.$$

In words, g decays faster at infinity than any rational function. The second is having n vanishing moments: a mother wavelet ψ is said to have *n vanishing moments* if

$$\int_{-\infty}^{\infty} t^k \psi(t) \, \mathrm{d}t = 0$$

for every $k \in \{1, \ldots, n\}$. It turns out that for a fast decaying wavelet, having exactly n vanishing moments is equivalent to having a function θ such that

$$\psi(t) = \left(-\frac{\mathrm{d}}{\mathrm{d}t}\right)^n \theta(t),$$

where θ is a function with nonzero mean, $\int \theta \, \mathrm{d}t \neq 0$, and corresponds to the father wavelet described in Section 2.2.3 [**55**, Theorem 6.2]. Probably the most frequently used example in this family is the derivative of Gaussian (DoG) wavelets, where θ is a Gaussian. DoG2, the second derivative, is particularly common, as any resulting extreme values of the scalogram characterize regions of high curvature, while zeros are indicators of edges. For the entire family, the curves defined by $\partial_t(\psi \star f) = 0$ are guaranteed to be continuous as functions of s and vanish only as s increases [**81**].

2.3.2. Wavelet Decay. In Section 2.1 we have described the relation between the differentiability of f and the decay of the Fourier coefficients. For non-smooth functions it is more appropriate to discuss the

Lipschitz exponent $\alpha(f,\nu)$ at a point ν:

$$f(t) - f(\nu) \leq C_{\alpha,f,\nu} \ t - \nu \ ^{\alpha}$$

for t in a sufficiently small neighborhood of ν and some constant $C_{\alpha,f,\nu}$. If f is n times differentiable at ν, then $\alpha < n+1$, and the minimal constant $C_{\alpha,f,\nu}$ is exactly the αth derivative if α an integer. As might be expected from the properties of the derivative discussed in Section 2.1, if the Fourier transform of a function f satisfies

$$\int_{-\infty}^{\infty} \widehat{f}(\omega) \left(1 + \ \omega \ ^{\alpha}\right) \mathrm{d}\omega < \infty$$

for some $\alpha > 0$, then f is globally Lipschitz α. However, since it's global, $\alpha = \min_{\nu \in \mathbb{R}} \alpha(f,\nu)$, so there may only be a few points where α is small, while the rest of the signal is well behaved and $\alpha(f,\nu)$ is quite large. As promised, the wavelet transform localizes $\alpha(f,\nu)$ so that the spread of any irregular points is confined to a cone of influence. Suppose that the mother wavelet has n vanishing moments, and that f is Lipschitz $\alpha(f,\nu) \leq n$ at ν. Then there is some $A > 0$ so that

$$W|f](s,t) \leq A s^{\alpha+1/2} \left(1 + \ \frac{t-\nu}{s} \ ^{\alpha'}\right)$$

with $\alpha = \alpha'$ and conversely, if f satisfies this bound for some α' *strictly less than* α, then it is Lipschitz α at ν. This was originally proved in [**39**], though it can also be found as Theorem 10.1 in [**40**] or Theorem 6.4 in [**55**]. When combined with a father wavelet, which caps the scale to $s \in (0, s_0]$, this gives a characterization of the decay rate of the wavelet coefficients, since $W|f](s,\nu) \leq A s^{\alpha+1/2}$.

The set $\left\{(t,s) \quad \frac{t-\nu}{s} \leq C\right\}$ is known as the *cone of influence* of the point ν, and the magnitude of $W|f](s,t)$ within this cone is closely related to the regularity of ν, though the fact that α and α' are not equal in the converse above leaves room for exceptions. These cones are clearly visible in Fig. 2.3.

Having introduced our sparse and translation covariant transform, it is time to see if we can combine these features in a nonlinear feature extractor.

CHAPTER 3

Scattering Transform

3.1. General Scattering Transform

A *generalized scattering transform* (hereafter referred to simply as a 'scattering transform' or ST) [**56**,**77**], has an architecture that is a continuous operator inspired by the structure of a CNN. For a diagram, see Fig. 3.1. For each layer $m = 0, 1, \ldots$, we have with a family of generators $\Psi_m := \{\phi^{m-1}, \psi^m_{\lambda^m}\}_{\lambda^m \in \Lambda_m} \subset \mathscr{L}^1(\mathbb{R}^d) \bigcap \mathscr{L}^2(\mathbb{R}^d)$ for some translation invariant frame

$$\left\{ T_t \mathrm{In}\left[\phi^{m-1}\right], T_t \mathrm{In}\left[\psi^m_{\lambda^m_i}\right] \right\}_{t \in \mathbb{R}^d, \lambda^m \in \Lambda_m}$$

where T_t is the translation operator, $\mathrm{In}\left[f\right](x) = \overline{f(-x)}$ is the involution operator, and Λ_m is some countable discrete index set, such as $\mathbb{Z}^d$. This frame has bounds a_m and b_m, that is

$$a_m |\, f|\,^2_2 \leq \left| f \star \phi^{m-1} \right|^2_2 + \sum_{\lambda^m \in \Lambda_m} \left| f \star \psi^m_{\lambda^m} \right|^2_2 \leq b_m |\, f|\,^2_2,$$

for all $f \in \mathscr{L}^2(\mathbb{R}^d)$. The singled out element ϕ^{m-1} is a low frequency frame atom, typically the only frame element covering the zero frequency. The index set Λ_m needs to tile the frequency plane in some way, for example by indexing scales and rotations. The frame atoms $\psi^m_{\lambda^m}$ correspond to the receptive fields found in each layer of a CNN.

In addition to the frames, at layer m we define a 1-Lipschitz pointwise operator M_m with bound γ_m which satisfies $M_m f = 0 \Rightarrow f \equiv 0$. After these have been applied, the result is subsampled at a rate $r_m \geq 1$. Finally, the output at each layer is then generated by averaging with a specific atom from the next layer ϕ^{m-1}, no longer used to pass on to the next layer.

The scattering transform of Mallat [**56**] for $\mathbb{R}^2$ corresponds to choosing

$$\Lambda_m = \{(j, h)\}_{j > -J_m, h \in H_m} \tag{3.1}$$

for some finite rotation group H_m. The resulting frames are generated by scaling and rotating a single mother wavelet: if $\lambda^m = (j, h)$, the corresponding frame generator is $\psi^m_{\lambda^m} = 2^{2j/Q_m}\psi^m\left(2^{j/Q_m}h^{-1}x\right)$, for some mother wavelet ψ^m, such as the 2D Morlet wavelet, with the averaging function ϕ^{m-1} being its corresponding father wavelet. The Lipschitz nonlinearity is $M_m = |\cdot|$. However, it lacks a subsampling factor between layers, so $r_m = 1$.

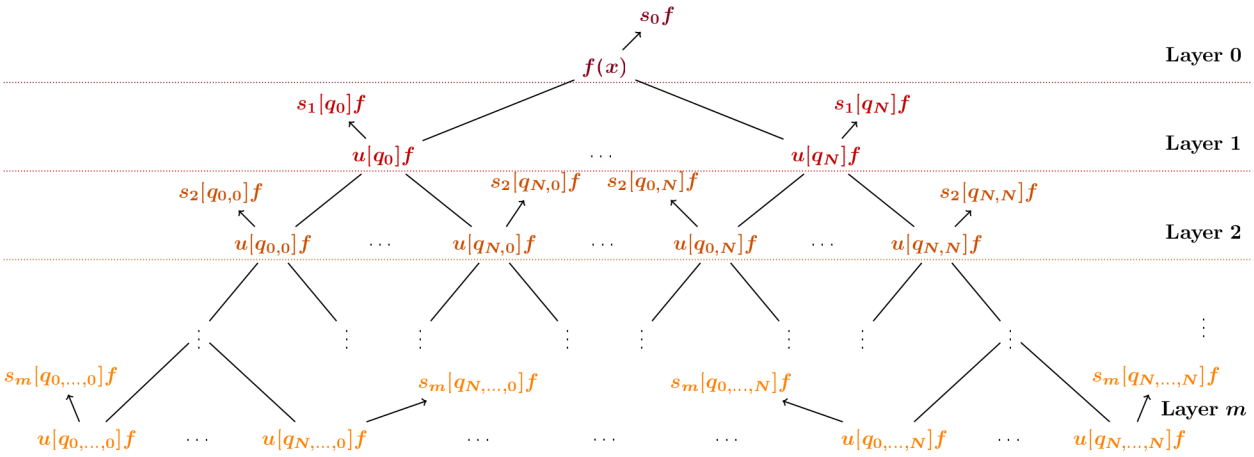


FIGURE 3.1. Generalized scattering transform. Here, $q^n_{k_n,\dots,k_0} = (\lambda^n_{k_n}, \dots, \lambda^1_{k_1})$ is an element of $\mathfrak{L}^n$, so $u\left|q^n_{k_n,\dots,k_0}\right]$ is as in Eq. (3.3), and $s\left|q^n_{k_n,\dots,k_0}\right] = \phi^n \star u_n\left|q^n_{k_n,\dots,k_0}\right]f$, i.e. an element of $\Phi_n|f]$

To get from layer $m-1$ to layer m, we define the function $u_m : \Lambda_m \times \mathscr{L}^2(\mathbb{R}^d) \to \mathscr{L}^2(\mathbb{R}^d)$ by

$$u_m|\lambda^m](f)(z) := M_m|f \star \psi^m_{\lambda^m_i}](r_m z). \tag{3.2}$$

Using this to define the value at a layer m, by selecting one index from each layer we have a *path* $q^m \in \mathfrak{L}^m := \Lambda_m \times \ldots \times \Lambda_1$, where $q^m = (\lambda^m, \lambda^{m-1}, \dots, \lambda^1)$ where $\lambda^i \in \Lambda_i$:

$$u|q^m](f) := u_m|\lambda^m]\,u_m|\lambda^{m-1}]\cdots u_1|\lambda^1](f). \tag{3.3}$$

As a base case, we set $u|q^0](f) \equiv f$. Choosing $d = 1$ corresponds to audio signals, such as sonar waveforms, while $d = 2$ would correspond to a sonar wavefield or images. Unlike a CNN, in a scattering transform each

layer has output, including just the average $s|\varnothing] = \phi^0 \star f$ (layer zero). For $m \in \mathbb{N}_0$, the output is

$$s\big|\, q^m\big] := \phi^m \star u\big|\, q^m\big](f)$$
$$\Phi_m\big|\, f\big] := \Big\{ s\big|\, q^m\big] \Big\}_{q^m \in \mathfrak{L}^m}.$$

The low frequency atom ϕ^m is the one paired with the frame in the *next* layer, so that it is capturing all of the information not passed on to further layers.

We define the output of the entire scattering transform to be

$$\Phi\big|\, f\big] := \bigcup_{m=0}^{\infty} \Phi_m\big|\, f\big].$$

The chief structural difference between a scattering transform and a CNN is the summation that occurs across filters in a CNN. In a scattering transform, each filter is only applied to a specific path of filters in the layer below, generating a tree-like structure; in contrast, in each layer of a CNN, it sums across different channels in the previous layer. This is one reason why the exponentially increasing number of paths limits the effective depth of a scattering transform, while CNNs have no such limitation.

3.2. Theoretical Guarantees

There are two additional conditions that restrict the various operators in a given layer simultaneously [**77**]. The first is the *weak admissibility condition*, which requires that the upper frame bound b_m be sufficiently small compared to the subsampling factor and Lipschitz constants:

$$\max\left\{ b_m, \frac{b_m \gamma_m^2}{r_m^2} \right\} \leq 1, \tag{3.4}$$

The second is that the nonlinearities must commute with the translation operator, so $M_m T_{\boldsymbol{x}} = T_{\boldsymbol{x}} M_m$. Most nonlinearities used for CNN's are pointwise, that is $M_m(\boldsymbol{x}) = \rho_m\big(f(\boldsymbol{x})\big)$ for some function ρ_m, so they certainly commute with $T_{\boldsymbol{x}}$. Given these constraints, we can now state the results of Wiatowski and Bölcskei [**77**] precisely; the first is that the resulting features Φ deform stably with respect to small frequency and space deformations:

THEOREM 3.2.1 (from [**77**]). *For frequency shift $\boldsymbol{\omega} \in \mathscr{C}^2\big(\mathbb{R}^d, \mathbb{R}^d\big)$ and space shift $\boldsymbol{\tau} \in \mathscr{C}^2\big(\mathbb{R}^d, \mathbb{R}^d\big)$, define the operator $F_{\boldsymbol{\tau},\boldsymbol{\omega}}\,\big|\, f\big]\,(\boldsymbol{x}) := \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{\omega}(x)\cdot \boldsymbol{x}} f\big(\boldsymbol{x} - \boldsymbol{\tau}(\boldsymbol{x})\big)$. If $|\, D\boldsymbol{\tau}|_\infty \leq \frac{1}{2d}$, then there exists a $C > 0$ independent of the*

choice of parameters for Φ *s.t. for all* $f \in \mathscr{L}^2_{\widehat{a}}(\mathbb{R}^d)$,

$$\left| \Phi\left| F_{\boldsymbol{\tau},\boldsymbol{\omega}} \right] - \Phi\left| f \right] \right|_2 \leq C\left(a|\,\boldsymbol{\tau}|_\infty + |\,\boldsymbol{\omega}|_\infty\right)|\,f|_2,$$

$$\left| \Phi\left| f \right] \right|_2 := \sum_{m=0}^{\infty} \sum_{q^m \in \mathfrak{Q}^m} \left| s\left| q^m \right](f) \right|_2$$

where $\mathscr{L}^2_{\widehat{a}}(\mathbb{R}^d)$ is the set of $\mathscr{L}^2(\mathbb{R}^d)$ functions whose Fourier transforms are band limited to $|-a, a]$. Mallat show a similar, tighter, bound bound for the specific case that $M_m = \cdot$ and $\psi^m_{\lambda^m_i}$ is generated by an admissible mother wavelet ψ^m with a number of conditions [**56**].

Their next result is that deeper layers of the scattering transform are more translation invariant:

THEOREM 3.2.2 (from [**77**]). *Given the conditions above, for* $f \in \mathscr{L}^2(\mathbb{R}^d)$ *the* m*th scattering layer output satisfies*

$$\Phi_m\left| T_{\boldsymbol{x}} f \right] = T_{x/r_m \cdots r_1}\left(\Phi_m\left| f \right]\right).$$

If there is also a global bound K *on the decay of the Fourier transforms of the averaging frame elements* ϕ^m*:*

$$\widehat{\phi^m}(\boldsymbol{\omega}) \leq K,$$

then we have the stronger result

$$\sum_{m=1}^{n} \left| \Phi_m\left| T_{\boldsymbol{x}} f \right] - \Phi_m\left| f \right] \right|_2 \leq \frac{2\pi\ \boldsymbol{x}\ K}{r_1 \cdots r_m} |\,f|_2.$$

To compare with the result from Mallat [**56**, Theorem 2.10], first define Φ^J_m to be the scattering transform with coarsest scale $-J$ in Eq. (3.1). Then for an admissible mother wavelet, the scattering transform achieves perfect translation invariance as the lower bound on the scale $-J$ goes to infinity:

$$\lim_{J \to \infty} \left| \Phi^J_m\left| T_{\boldsymbol{x}} f \right] - \Phi^J_m\left| f \right] \right|_2 = 0.$$

3.3. Implementation and Examples

Here we discuss the particulars of our implementation of the generalized scattering transform used for 1D sonar classification. The code can be found on github.com at ScatteringTransform.jl.[1] We have implemented

[1] https://github.com/dsweber2/ScatteringTransform.jl

all of the wavelets found in Section 2.2.1 from ContinuousWavelets.jl, which are all 1D transforms, as well as shearlets for the 2D case. For subsampling, we use a very straightforward average pooling, which if the subsampling rate is p/q, sums over windows of size q, keeping every p entries.

3.3.1. Discrete Output. One feature of our implementation that is not present in the continuous theory above is a final subsampling after averaging with the father wavelet. Because the father wavelet is typically approximately zero past some relatively low frequency (in Fig. 2.5, for example, the left-most column is the father wavelet, and it is approximately zero by ~40Hz out of the ~500Hz frequencies used), including all points in the output is typically highly redundant. We subsample the output $s\left|q^m\right](t)$ to reduce this redundancy; we will denote this further subsampling rate by r'_m.

In the discrete case, the input signal f must be evaluated at only a finite set of points t_j for $j = 0, \ldots, N-1$, so define $f_j = f[j] := f(t_j)$. Because of the subsampling, the points where either the internal layers $u\left|q^m\right]$ or external output $s\left|q^m\right]$ are evaluated will not simply be the t_j's. Instead, the discretized internal layers are given by the collection of vectors $U = \left\{\boldsymbol{u}\left|q^m\right]\right\}_{q^m \in \mathfrak{Q}^m, m \in \mathbb{Z}_0}$, with the entries of the vector $\boldsymbol{u}\left|q^m\right]$ given by $u_i\left|q^m\right] := u\left|q^m\right]\left(\tau_{i,m}\right)$, where the continuous output has been sampled at the points $\tau_{i,m} := t_{j_{i,m}}$ for some subset $t_{j_{i,m}}$ of the input points t_j which keeps every ~$\prod_{k=1}^{m} r_k$th point. Note that $\tau_{j,0} = t_j$. Similarly, the output is given by the collection of vectors $S = \left\{s_i\left|q^m\right] := s\left|q^m\right]\left(\tau'_{i,m}\right)\right\}_{q^m \in \mathfrak{Q}^m, m \in \mathbb{Z}_0}$, where the output sample points have been further subsampled, $\tau'_{i,m} := \tau_{k_i,m}$, where k_i keeps every ~r'_mth point. For the wavelets in layer m, we will denote the sampled vector equivalent as $\psi^m_{\lambda^m}[i] := \psi^m_{\lambda^m}\left(\tau_{i,m}\right)$.

We will refer to the *output location i/M (at path q^m)* as shorthand for ith value in the vector $\boldsymbol{s}\left|q^m\right]$, which is of length M. We will omit "at path q^m" when it is clear from context.

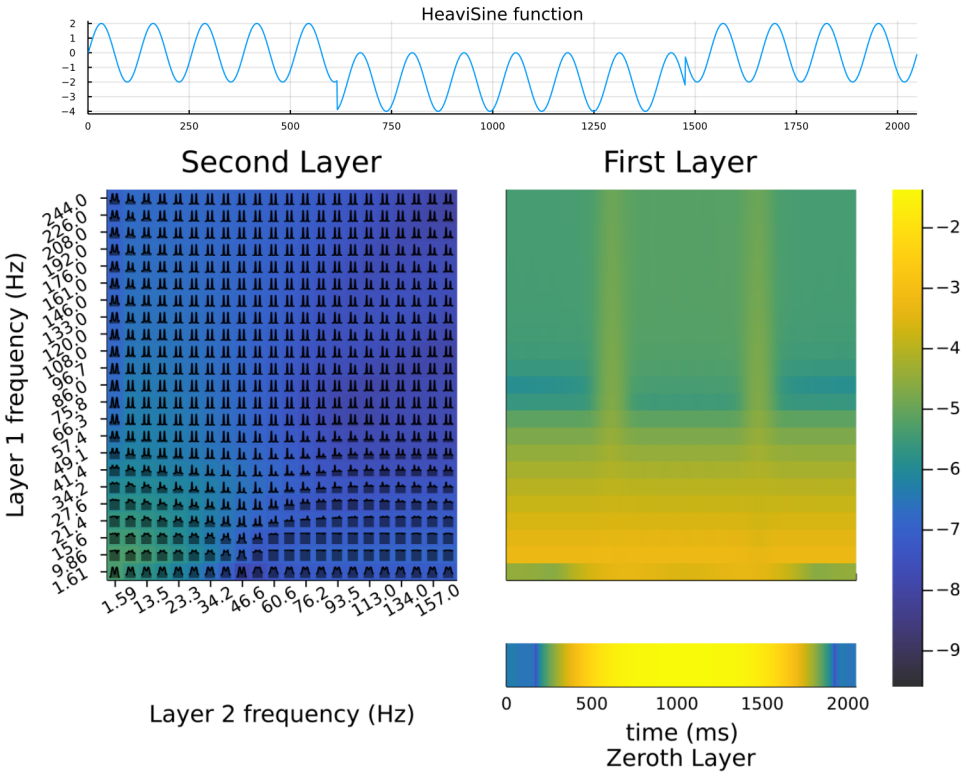
Unless otherwise noted, we will be using a sampling rate of 1000Hz, or a sample every millisecond.

3.3.2. Normalization. As observed in [**11**], the magnitude of the coefficients in each layer is frequently orders of magnitude different. This can be a major issue for convergence for some linear classification methods, which assume that the input is normally distributed and frequently mean zero [**73**]. The mean is usually accounted for through a bias term, but whitening is typically handled as a preprocessing step. To do this, we normalize each layer separately via $s\left|q^m\right](t) \cdot N_m / \left|\Phi_m\left|f\right]\right|_2$ where N_m is the total number of paths used in the mth layer. This only applies for classification however, since for interpretation purposes, normalization creates some odd distortions (see Chapter 4).

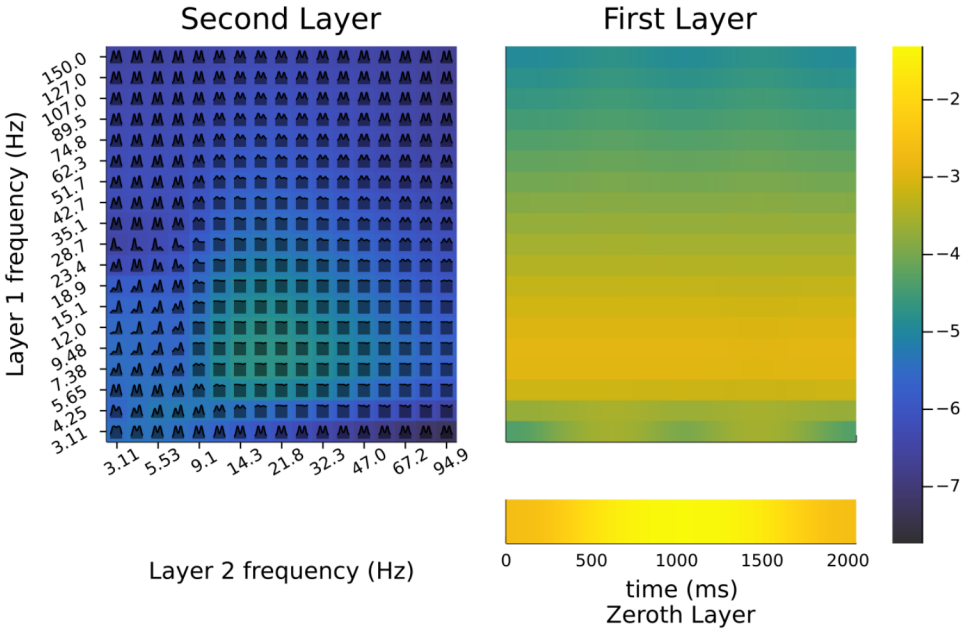
3.3.3. Example transforms. In Fig. 3.2a, Fig. 3.2b, and Fig. 3.3 we have a compressed representation of layers zero, one, and two for the scattering transform of various signals and wavelets. In all of the figures, the color represents the magnitude of the ST output in the logarithmic scale (base 10) at that path (for the second layer) or time (for the zeroth or first layers). For the zeroth layer, we have taken the absolute value to be able to plot on the same color scale as the other layers. The x-axis for the zeroth layer is the time, while there is no variation along the y-axis, since there is only the averaging function. It is a heatmap only to facilitate comparison with the output of the other layers. The first layer is quite similar to the scalograms in Section 2.2.1, although the magnitude is not squared to facilitate comparison, and it has been averaged and subsampled. The second layer is the most complicated to describe. There are two indices to express each path $q^m = (\lambda^2, \lambda^1)$, and then for each path there is a time component as well. To express this, the paths are given along the axes, while at a particular heatmap location, for example $(\lambda^2, \lambda^1) = (1.59\text{Hz}, 9.86\text{Hz})$, there is a small subplot where time varies along the x-axis, corresponding to the output $s\big|(\lambda^2, \lambda^1)\big](\tau'_{i,2})$. The color for the second layer refers to the log base 10 of the largest coefficient at that path. Given the layer of detail in each subplot, these figures are best examined on a digital copy, zooming in as needed.

For the HeaviSine function, the zeroth layer in either Fig. 3.2a or Fig. 3.2b captures the step function from 500–1500ms. As may be expected from averaging with the father wavelet, the edges in the first layer are blurred, but the pure sinusoid at 9Hz comes through clearly. In the second layer, the two discontinuities show up as peaks in either Fig. 3.2a or Fig. 3.2b at high first layer frequency, though the largest coefficients are still a result of paths with first layer frequency around 9Hz. These coefficients are roughly uniform in space on the paths near $(21.8, 9.48)$. On the other hand, there are some paths, such as $(5.53\text{Hz}, 15.1\text{Hz})$ or $(3.11\text{Hz}, 28.7\text{Hz})$ where only one of the two peaks shows up prominently. Attempting to understand what these coordinates mean will be the principal aim of Chapter 4.

For the bumps function in Fig. 3.3, the zeroth and first layers roughly indicate the locations of discontinuities, with somewhat more clarity as the scale becomes finer at high frequency. The first layer DoG2 wavelets in Fig. 3.4 correspond much more roughly to the scalogram in Fig. 2.3c, due to the blurring of the averaging function and the subsampling. For the DoG2 wavelets, looking at the individual subplots, the second layer paths can roughly be divided into two kinds. Paths such as $(3.11\text{Hz}, 9.48\text{Hz})$ which are low frequency in both the first and second layer wavelets, generally consist of a single peak. On the other hand, paths such as $(32.3\text{Hz}, 150.0\text{Hz})$ which is relatively high frequency in both, consist of two bumps, with the left higher than



(a) ST Coefficients of the HeaviSine using Morlet wavelets of mean frequency π



(b) ST Coefficients of the HeaviSine using DoG2 wavelets

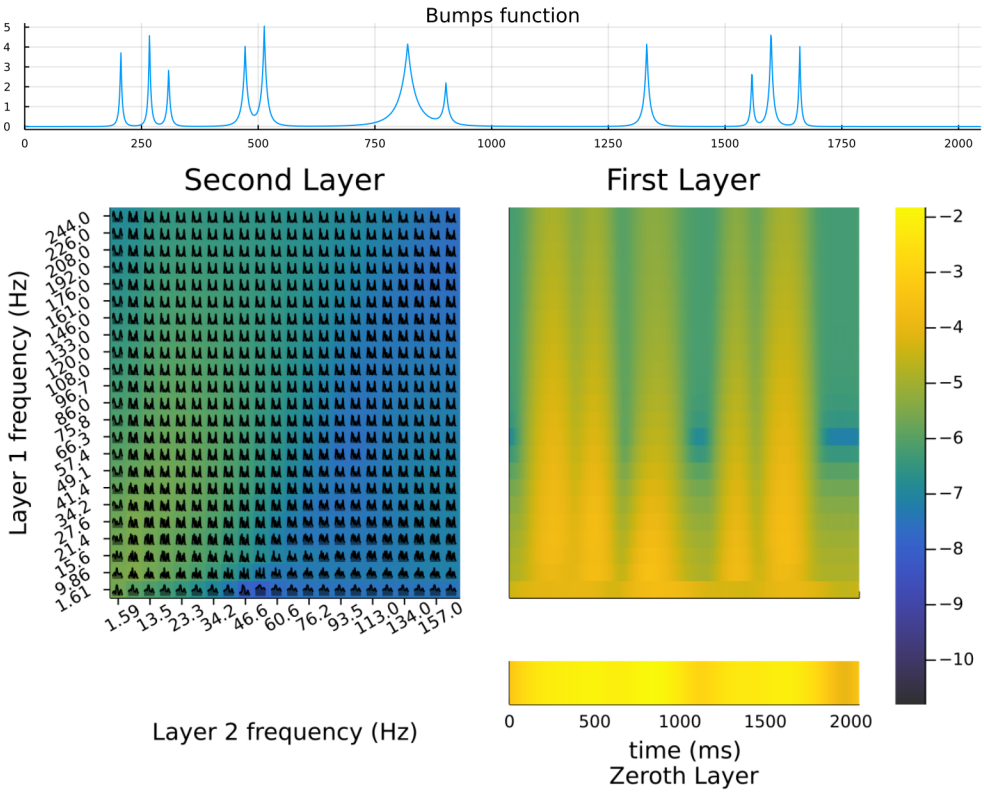


FIGURE 3.3. ST Coefficients of the bumps signal using Morlet wavelets of mean frequency π

the right. These roughly correspond to the collection of bumps between 400–900ms where there are 7 peaks, and the collection between 1300–1700ms, where there are only 4 peaks. Using Morlet wavelets in Fig. 3.3, the same two peaks show up for high frequency in both (such as path $(60.6\text{Hz}, 244.0\text{Hz})$). There is however much more noise in the signals.

In Fig. 3.2a and Fig. 3.3, we can see the concentration along decreasing paths where $\lambda^2 < \lambda^1$. Mallat proved this property occurs for a fairly restrictive class of wavelets in [**56**, Lemma 2.8] and it was empirically observed for Morlet wavelets by Bruna and Mallat in [**11**]. They observed that the only coefficients with large magnitude for scattering transforms using Morlet wavelets are those where the frequency in the second layer is less than the first layer, or $\lambda^2 < \lambda^1$. For these figures, this means that the majority of the energy is located in the upper left, above the diagonal line where the frequencies are equal; since the second layer is subsampled, this is slightly above the 45° line. Note however, that this is very much not the case for the real-valued DoG2 wavelets in Fig. 3.2b, where the largest second layer coefficients occur for paths around

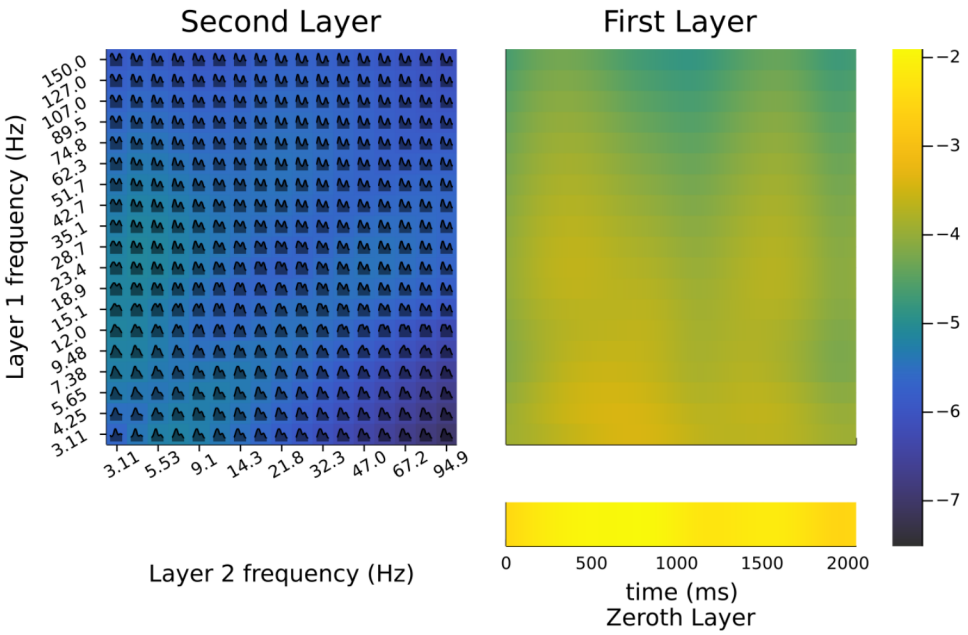


FIGURE 3.4. ST Coefficients of the bumps signal using DoG2 wavelets

$(21.8, 9.48)$. The effect is subtler in Fig. 3.4, but the response around path $(47.0\text{Hz}, 23.4\text{Hz})$ is of the same order of magnitude as that at path $(3.11\text{Hz}, 23.4\text{Hz})$.

CHAPTER 4

Time–frequency Interpretation of Scattering Transform Representation

As was discussed in Section 2.2 and Section 2.3, wavelet coefficients have a rich set of interpretations. For real wavelets, they characterize edges and other non-smooth points in a signal based on the Lipschitz regularity at those points. For analytic wavelets, they decompose a signal into an amplitude and phase locally in the time domain.

Unfortunately, we don't have as clean an interpretation of ST coefficients. Because the convolutional layer has well-understood filters makes it significantly easier to interpret than a generic CNN, yet there has been some difficulty in interpreting the coefficients. Most methods of interpretation to date are best considered as some form of inversion [**5**, **18**]. The work of Cotter and Kingsbury reconstructs the input using the dual frame in each layer with the expense of storing phase (or sign) information during the forward transform [**18**]. In [**5**], Angles and Mallat use a generative adversarial network very similar to that of Dosovitskiy et al. [**24**] to approximate the inverse of the scattering transform for a specific dataset. While of broader applicability than Cotter and Kingsbury's work, its focus is more on using ST coefficients to generate plausible images in a broader class, rather than interpreting the meaning of any particular scattering coefficient. Finally, Waldspurger's phase retrieval methods for recovering the original input from the modulus of the wavelet transform, while they would certainly form a reasonable basis from which to formulate an inversion method, we are unaware of any paper which does so [**76**].

In this chapter, we will use both theoretical and experimental methods to try to shed more light on the meaning of the ST coefficients. As with the original wavelets, there are marked differences between the interpretations of real and analytic wavelets, which recur in every interpretation method we use.

In Section 4.1, we examine the gradient of particular coordinates at various signals. The gradient is perhaps the simplest method of performing sensitivity analysis, and gives us local information about how to increase the response of a particular ST coefficient; an example of a work using the gradient in this manner for CNNs is by Simonyan et al. [**70**].

In Section 4.2 we use a more direct method of calculating the *pseudoinverse* than is found in either [**18**] or [**5**], and use this to create signals which maximize a particular ST coefficient. Previous work using this general framework in the context of CNNs includes [**27**], [**54**], and [**70**]. Mahendran and Vedaldi [**54**] seek to understand what information is retained by a neural network by "inverting" them, while [**70**] seeks to understand class labels specifically by finding examples which maximize the output for a specific class. In Section 4.2 we seek to find inputs that correspond to particular coordinates or combinations of coordinates being active.

In Section 4.3, we tackle interpreting the ST coefficients using the interpretations of the wavelet coefficients themselves described in Section 2.2 and Section 2.3. Finally, in Section 4.4 bringing all of this together, we apply the scattering transform to a classic signal classification problem discerning 3 classes, and interpret the LASSO coefficients used in the classification.

REMARK 4.0.1 (Normalization). *In Section 3.3.2 we describe our normalization process for classification. Throughout this chapter however, we will be working with the unnormalized scattering transform, as the normalization can introduce some spurious dependence between coordinates using either the gradient-based or pseudo-inversion methods. For the purposes of interpretation, the normalized and unnormalized coefficients are the same, since within a given layer the normalization is effectively multiplication by a constant. If we were to do Pseudo-inversion fitting paths from multiple layers simultaneously, then it would be important to include normalization (this will come up in Section 6.5). As we restrict our investigation to fitting paths from the same layer (and in large part a particular location) in this chapter, we can consider normalized and unnormalized coefficients to contain much the same information. Since this constant depends on the input signal however, both the gradient and the pseudoinverse introduce dependence. For examples of how this causes issues, see either Section 4.1.1, or Appendix A.1.*

4.1. Scattering Transform Coefficient Gradients

One potential meaning of interpreting a transform is demonstrate how sensitive the output is to a given change in the input: this is a principal concern behind the subject of *sensitivity analysis* [**69**]. A simple way of doing this analysis is using the gradient information, and was one of the reasons behind the development of *automatic differentiation*, a set of computational methods to compile a function and return its gradient [**33**]. We can write a linear regression as $g(\boldsymbol{x}) = a + \boldsymbol{\beta} \cdot \boldsymbol{x}$, and in this case the sensitivity of $g(\boldsymbol{x})$ with respect to coordinate x_j is simply β_j. The sign of the β_j tells us whether there is a positive or negative correlation with the output, while the relative magnitudes of the weights β_j tells us the relative importance of the x_j.

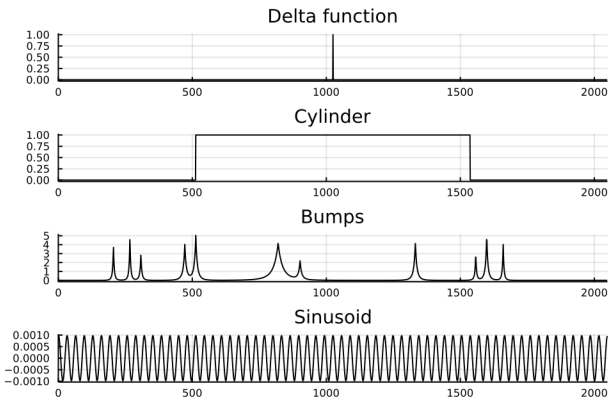


FIGURE 4.1. Examples

Similarly, the gradient of the discrete wavelet transform at a particular output time and scale is just a translated and scaled copy of the mother wavelet. Using notation from Section 3.3.1, the discrete equivalent of Eq. (2.3) with $p = 1$ for mother wavelet ψ at scales λ_k and output locations τ_i is given by

$$W|\boldsymbol{f}](\lambda_k, \tau_i) = \lambda_k^{-1} \sum_{l=0}^{N-1} f_l \psi\Big(\frac{t_l - \tau_i}{\lambda_k}\Big) \tag{4.1}$$

So the derivative with respect to f_j at scale λ_k and location τ_i, evaluated at the input function $\boldsymbol{g}$ is given by

$$\frac{\partial}{\partial f_j} W|\boldsymbol{f}](\lambda_k, \tau_i)\Big|_{\boldsymbol{f}=\boldsymbol{g}} = \frac{\partial}{\partial f_j} \lambda_k^{-1} \sum_{l=0}^{N-1} f_l \psi\Big(\frac{t_l - \tau_i}{\lambda_k}\Big)\Big|_{\boldsymbol{f}=\boldsymbol{g}} = \lambda_k^{-1} \psi\Big(\frac{t_j - \tau_i}{\lambda_k}\Big) = T_{\tau_i} \psi_{\lambda_k}(t_j), \tag{4.2}$$

which is simply a translated copy of the corresponding wavelet. This tells us that, given that we can add a vector $\boldsymbol{x}$ of fixed magnitude to the initial signal $\boldsymbol{f}$, the most efficient way of increasing the value of the wavelet coefficient $W|\boldsymbol{f}](\lambda_k, \tau_i)$ is to set $x_j = T_{\tau_i} \psi_{\lambda_k}(t_j)$, regardless of the signal $\boldsymbol{f}$.

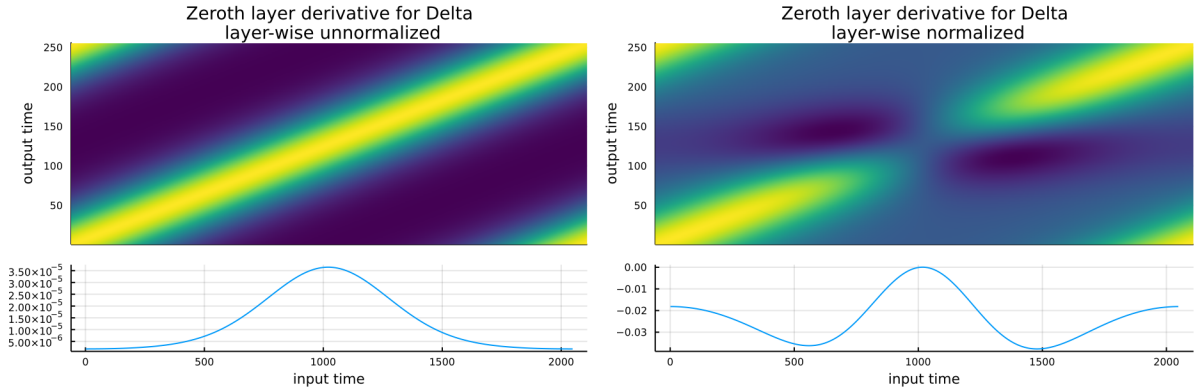
Moving on to the ST coefficients, as the ST is nonlinear, the gradient is no longer independent of the input signal. Instead of having output indexed just by scale and time, it is indexed by the layer m, then the path q^m, and finally the output location i/m (see Section 3.3.1 for a discussion of this notation). The derivative with

respect to changing f_j at output location i and path q^m, when evaluated at input g is the Jacobian

$$\frac{\partial}{\partial f_j} S_i \left| q^m \right] \Big|_{\boldsymbol{f}=\boldsymbol{g}},$$

though we will for the most part consider a fixed output coordinate i/m and a path q^m to get the gradient $\nabla_{\boldsymbol{f}} S_i \left| q^m \right] \Big|_{\boldsymbol{f}=\boldsymbol{g}}$. As noted above, the choice of input signal $\boldsymbol{f}$ matters for the case of the scattering transform, so we will look at a couple of different examples (see Fig. 4.1). Arguably the simplest example would be a constant function, but every path except the initial father wavelet is not dependent on the input signal's mean, so the gradient at a constant function for all other paths is simply zero. Hence, instead of the constant function, we will use: a delta spike, arguably the next simplest spatial signal; the "cylinder signal" function, which is just a characteristic function and will return in Section 4.4; the "bumps" function described in Section 2.2.1 and Section 3.3.3; a pure tone of 31.25Hz, which is a frequency delta spike; and finally white noise.

We find that as one may expect, as the depth increases, the degree of abstraction away from the used wavelets increases. The zeroth layer in Section 4.1.1 is simply the father wavelet. The first layer in Section 4.1.2 is still responsive to the wavelet at the output location and scale as in Eq. (4.2), but is also increasingly responsive to somewhat displaced copies of the wavelet.



(a) Unnormalized (b) Normalized

FIGURE 4.2. Comparing the zeroth layer derivative corresponding to the delta function in the normalized and unnormalized case. The horizontal axis varies with the gradient f_j, while the vertical axis in the heatmap corresponds to the output location i/m. The bottom line plot gives the derivative when $i/m = 128/256$.

4.1.1. Zeroth Layer. As the zeroth layer is a subsampled version of $\phi^0 \star f(x)$, one would expect (and get) just the father wavelet as a derivative $\nabla_{\boldsymbol{f}} S_i | \varnothing] \Big|_{\boldsymbol{f}=\boldsymbol{g}} = T_{\tau'_{i,0}} \phi^0(t_j)$, as in Fig. 4.2a, regardless of the target

function f. $\tau'_{i,0}$, as defined in section 3.3.1 is the ith sample at the subsampled rate of the zeroth layer output. For the DoG2 case shown here, $T_{i/m}\phi^0(t_j)$ is just a shifted Gaussian. Normalization as discussed in Section 3.3.2, while useful for classification, can cause some counter-intuitive results, as in Fig. 4.2b, where the derivative $\nabla_{\boldsymbol{f}} s_i|\varnothing]\ _{\boldsymbol{f}=\boldsymbol{e}_{128}}$ is not just a translation as we change j. The value of the output at the location of the delta spike (in Fig. 4.2b this is at input index $j = 1024$ and output index $i = 128$) will actually decrease if the signal increases at any other point, as the fixed total mass is distributed over a larger area. This results in the derivative at the center shown in the lower line graph of Fig. 4.2b being strictly non-positive. Any increase at a point other than $j = 1024$ results in some other output coordinate increasing in value, which pulls some of the fixed mass away from $s_{128}|\varnothing]$. For the remainder of this section we will not discuss normalized examples, as they primarily serve to complicate the narrative and introduce spurious dependence between the coordinates.

4.1.2. First Layer. For the first layer, we can directly compute the gradients, as the discrete output is $s_i|\lambda] = \frac{1}{2K+1}\sum_{k=-K}^{K} \phi^1 \star\ \psi_\lambda \star f\ |i-k]$, where our averaging subsampling has window width of $2K+1$. Thus, the derivative[1] is

$$\begin{aligned}
\frac{\partial}{\partial f_j} s_i|\lambda] &= \frac{\partial}{\partial f_j}\frac{1}{2K+1}\sum_{k=-K}^{K} \phi^1 \star\ \psi_\lambda \star f\ |i+k] \qquad (4.3)\\
&= \frac{\partial}{\partial f_j}\frac{1}{2K+1}\sum_{k=-K}^{K}\sum_{l_1=0}^{N-1}\phi^1|i+k-l_1]\sum_{l_2=0}^{N-1}\psi_\lambda|l_1-l_2]f|l_2]\\
&= \frac{1}{2K+1}\sum_{k=-K}^{K}\sum_{l_1=0}^{N-1}\phi^1|i+k-l_1]\mathrm{sgn}\big(\psi_\lambda\star f\big)|l_1]\psi_\lambda|l_1-j]\\
&= \sum_{l_1=0}^{N-1}\mathrm{sgn}\big(\psi_\lambda\star f\big)|l_1]\psi_\lambda|l_1-j]\frac{1}{2K+1}\sum_{k=-K}^{K}\phi^1|i+k-l_1]\\
&= \sum_{l_1=0}^{N-1}\mathrm{sgn}\big(\psi_\lambda\star f\big)|l_1]\psi_\lambda|l_1-j]\widetilde{\phi^1}|i-l_1]\\
&= \Big(\mathrm{sgn}\big(\psi_\lambda\star f\big)\psi_\lambda|\cdot-j]\Big)\star\widetilde{\phi^1}|i] \qquad (4.4)\\
&= \Big(\mathrm{sgn}\big(\psi_\lambda\star f\big)|-\cdot]\psi_\lambda|-\cdot]\star\mathrm{In}\Big|\widetilde{\phi^1}\Big|\ |i-\cdot]\Big)|j] \qquad (4.5)
\end{aligned}$$

[1] Points where $\psi_\lambda \star f = 0$ pose somewhat of a problem; for the numerical examples, we will choose the sgn function with $\mathrm{sgn}(0) = 0$.

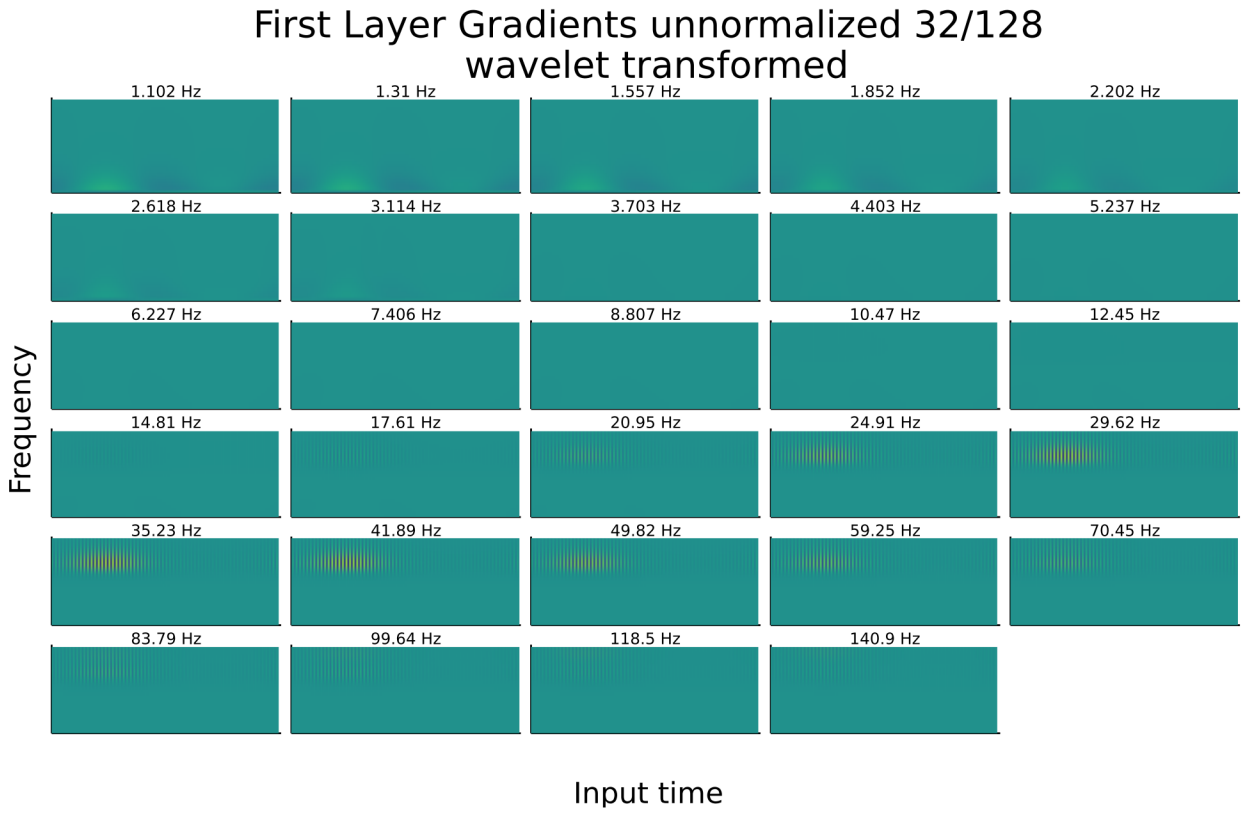


FIGURE 4.3. The scalogram using the first layer wavelets of the gradient $\nabla_f s_{32}|\lambda]\ _{f=\sin}$ of a sinusoid of frequency 31.25Hz, centered at $i/m = 32/128$.

where $\widetilde{\phi^1}|l] = \frac{1}{2K+1}\sum_{k=-K}^{K}\phi^1|l+k]$ is the local average of ϕ^1 around index l. Either Eq. (4.4) or Eq. (4.5) could be considered as a simplified form. Eq. (4.5) has the advantage of being evaluated at the index j of the gradient, while Eq. (4.4) is a simpler expression.

So the gradient is *a convolution of an averaged version of the involuted father wavelet centered at the output location with a possibly signed flipped version of the involuted target wavelet.* Relative to the sign function, the output index i determines the displacement of the averaging function, while the gradient index j determines the displacement of the wavelet itself. In the case of the input being a delta function at l_0, this is explicitly

$$\frac{\partial}{\partial f_j}s_i|\lambda] = \left(\operatorname{sgn}(\psi_\lambda)|\cdot - l_0]\psi_\lambda|\cdot - j]\right) \star \widetilde{\phi^1}|i] = \left(\operatorname{sgn}(\psi_\lambda)|l_0 - \cdot]\psi_\lambda \star \widetilde{\phi^1}|i-\cdot]\right)|j],$$

or the sign of the wavelet at one point, the value at another, and then smoothed.

While the gradient evaluated at the delta spike has the simplest explicit form, the simplest example to understand is a pure-tone sinusoid of frequency 31.25Hz, whose scalogram is in Fig. 4.3. For wavelets that don't overlap the relevant frequency, the gradient is (nearly) zero, while for those that do, the response is at the frequency of the input, with an envelope dictated by the averaging function centered around the output location.

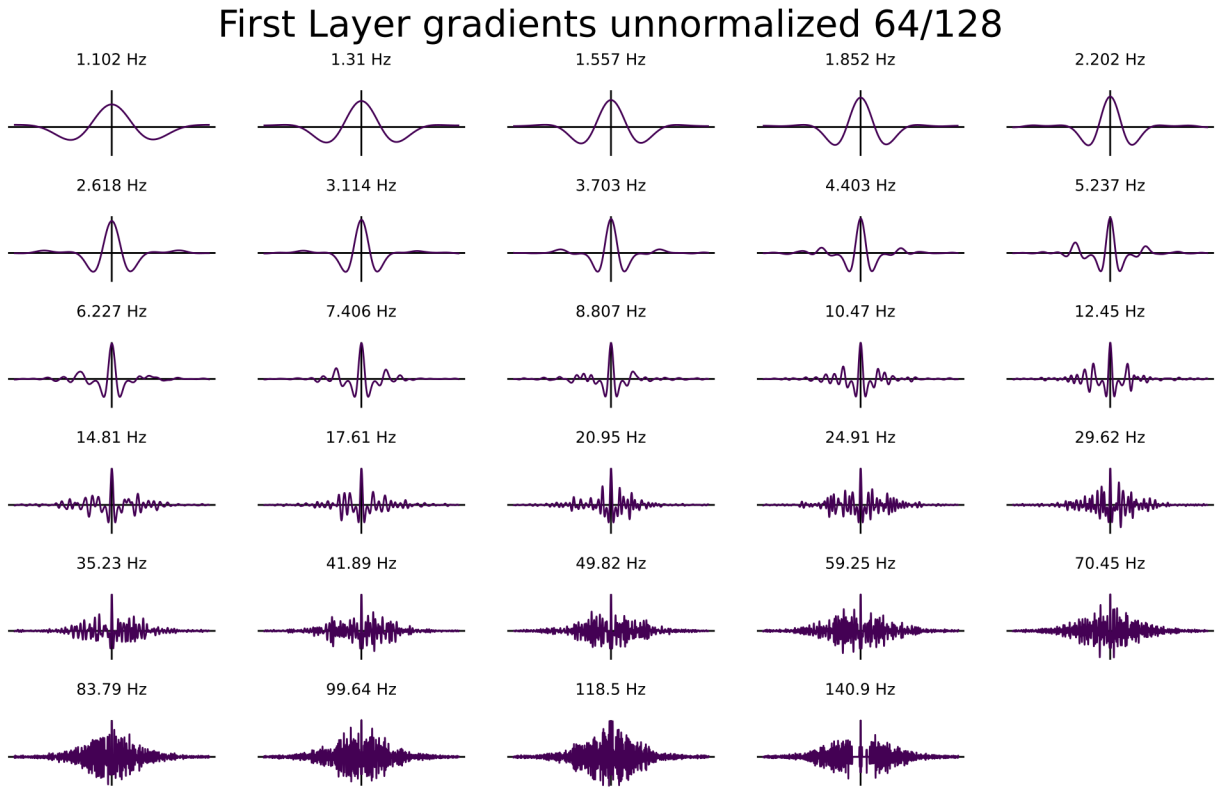
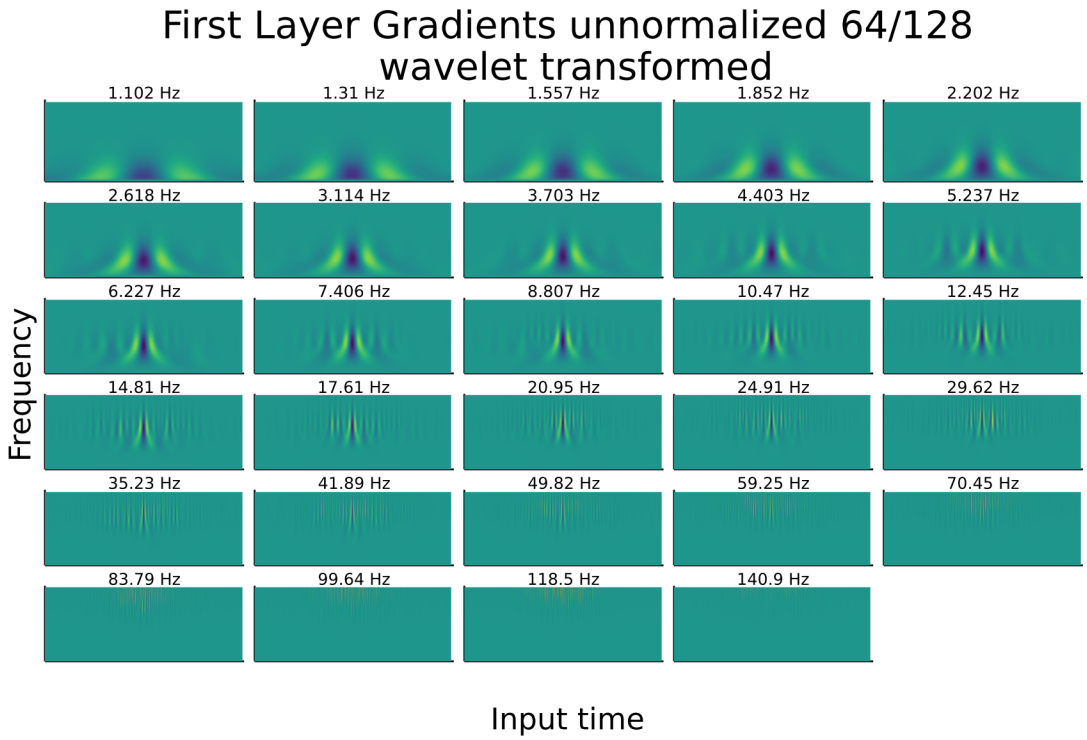
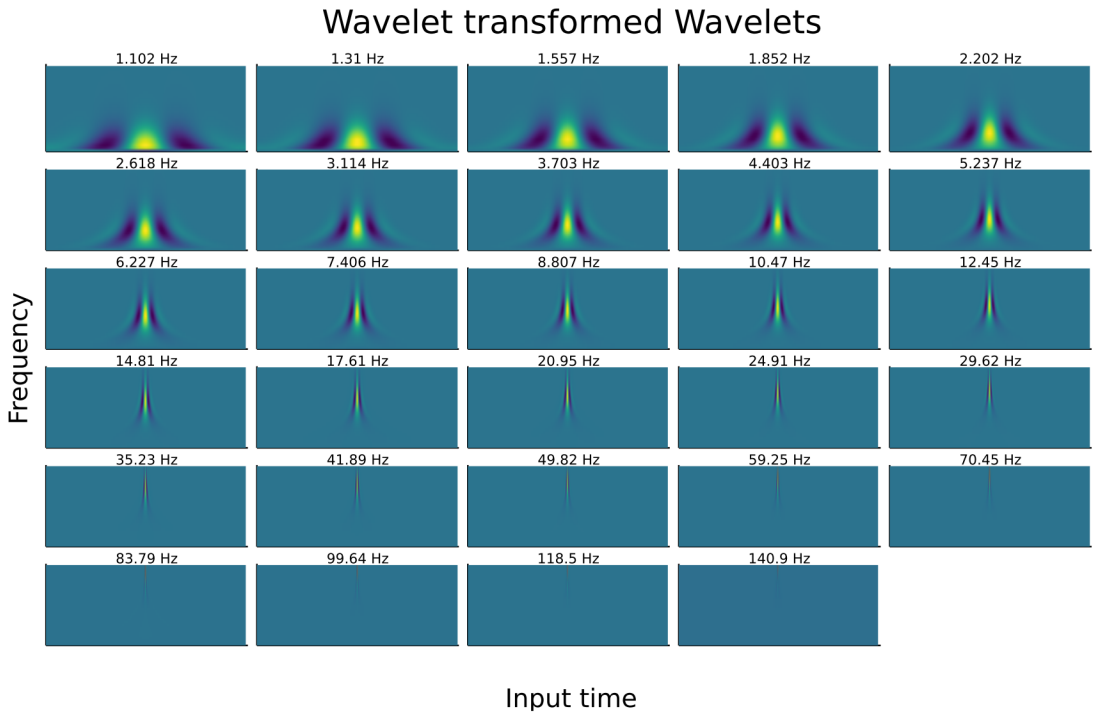


FIGURE 4.4. The gradient of the first layer for an input signal of a delta function as in Fig. 4.1 (located at 1024ms, halfway through the signal). The output location whose gradient we are taking is also halfway through the signal, at output location 64/128. Each separate sub-plot corresponds to a different path; in the first layer each path is uniquely identified by the corresponding frequency.

The gradient evaluated at the delta function in Fig. 4.4 introduces more complexity. At the low frequencies, up until ~3.114Hz, the gradient is simply the original wavelet. For higher frequencies however, additional oscillatory features begin to appear. The effect is most clear visually at 5.237Hz and 7.406Hz, where in addition to the central DoG2 wavelet, there are smaller copies surrounding it. Beyond this point, the frequency becomes too high for the eye to discern the pattern.



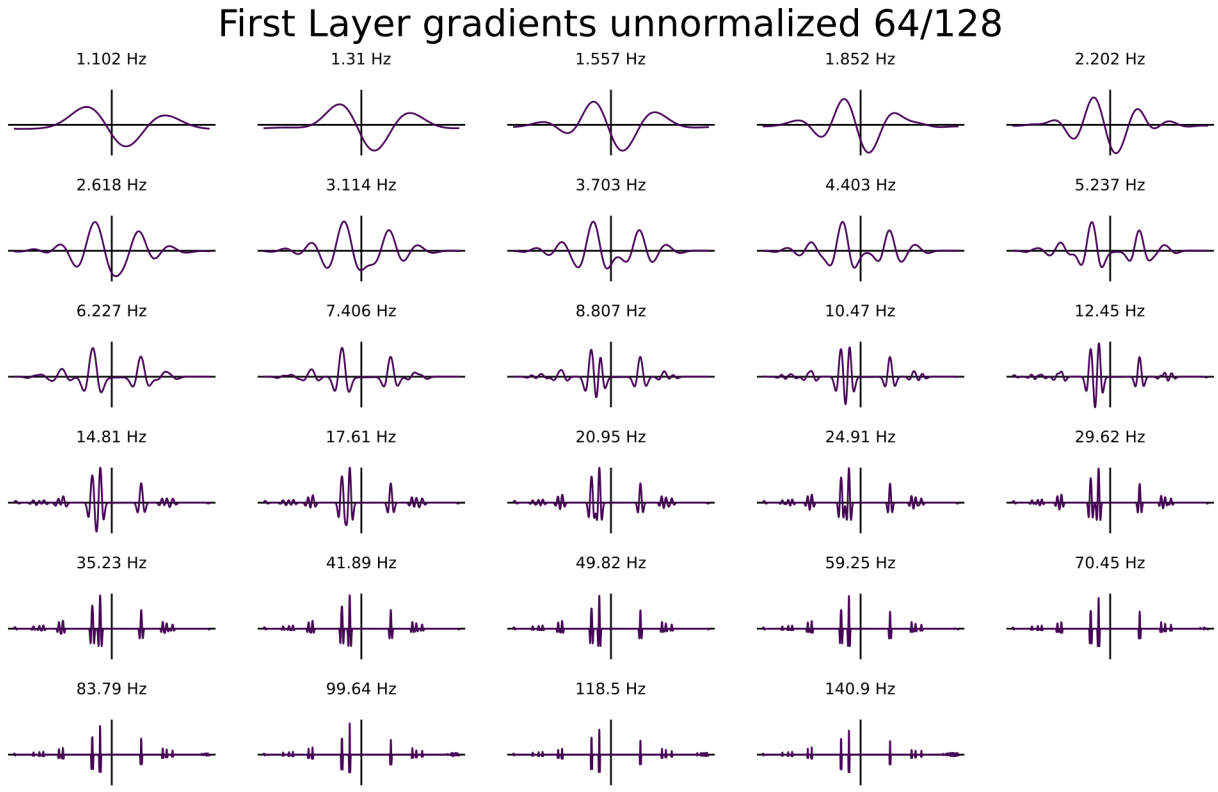
(a) Scalogram of the gradient of the first layer evaluated at the delta spike.



(b) Scalogram of the DoG2 wavelets for comparison.

FIGURE 4.5. Comparing Scalograms. The subfigures correspond to paths in the same manner as in Fig. 4.4.

To get a clearer look at this effect at high frequencies, it is most effective to compare the scalogram of the resulting signals as in Fig. 4.5a with the scalogram of the original wavelets in Fig. 4.5b. In addition to the central region matching the original wavelet, there are multiple fainter copies of this central wavelet in the immediate neighborhood of the main wavelet. This arises because of the averaging wavelet, since the output $s_{64}[\lambda]$ will increase if a copy of the wavelet is present anywhere within the support of ϕ, in proportion to the distance from the maximum. The interval where ϕ is greater than half maximum is $j = 703$ to $j = 1345$, out of 2048 entries, or approximately the center third of the signal.



(a) unnormalized

FIGURE 4.6. The gradient evaluated at the bumps function in Fig. 4.1

The gradient of the bumps function in Fig. 4.6 is generally much sparser than Fig. 4.4, especially at high frequency, suggesting that the response at the delta function is much more sensitive to any change at all, whereas the response at the bumps function is only dependent on the behavior in the neighborhood of the already existing peaks. As the frequency increases past ~10.47Hz, each peak contains a small copy of the corresponding wavelet, while as it decreases, the peaks continue with the expected joining of peaks as the

frequency passes below the distance between the bumps, until at 1.31Hz they are all covered by the same wavelet [**81**]. A feature that does carry over is that the magnitude depends on the distance from the output location $64/128$. This feature is somewhat obscured by the sparsity of the figure, but the magnitude of the wavelets at 24.91Hz, for example, are determined by the distance from the central line, rather than the magnitude of the original signal or wavelet transform (compare with Fig. 2.3c).

Finally, the gradient evaluated at the cylinder signal in Fig. 4.7 shares many similarities with the gradient at the delta function in Fig. 4.5a, in that there is a central spike with some activity around the edges. The difference is most prominent beyond 14.81Hz; for the cylinder signal, the strength of the support remains stronger throughout the supported center third of the signal than it is for the delta function. Increasing the response of a wavelet at either the beginning or the end of the interval is easier than for the delta function, which is 0 in the neighborhood of the corresponding locations.

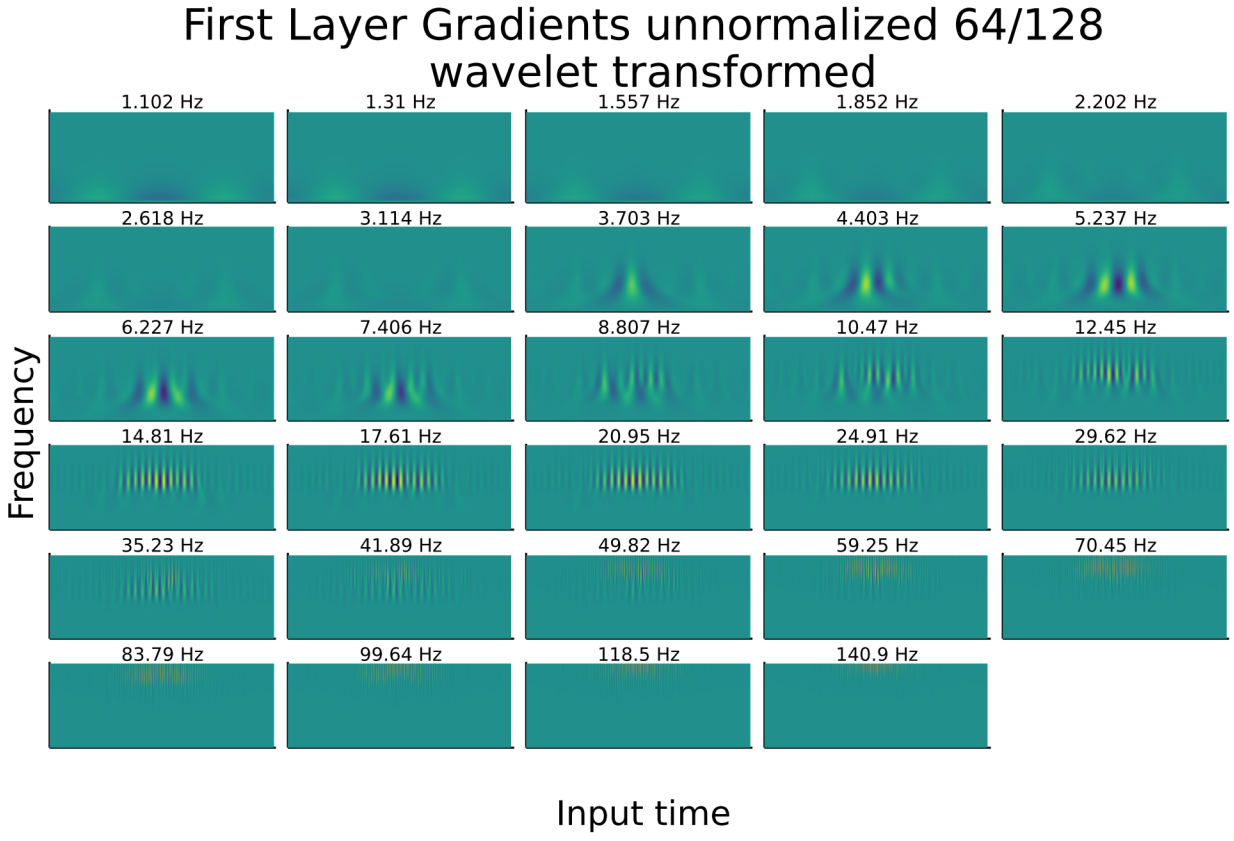


FIGURE 4.7. The scalogram of the gradient evaluated at the cylinder signal in Fig. 4.1

4.1.3. Second Layer. Following a similar calculation as the first layer, if we have a path $p = (\lambda^2, \lambda^1)$, then the derivative of the second layer is

$$\begin{aligned}\frac{\partial}{\partial f_j} s_i[p] &= \frac{\partial}{\partial f_j} \sum_{k_0=-K_0}^{K_0} \phi^2 \star \psi^2_{\lambda^2} \star \sum_{k_1=-K_1}^{K_1} \psi^1_{\lambda^1} \star f\,[\cdot + k_1]\,[i + k_0] \\ &= \widetilde{\phi^2} \star \Big(\operatorname{sgn}\Big(\psi^2_{\lambda^2} \star \sum_{k_1=-K_1}^{K_1} \psi^1_{\lambda^1} \star f[\cdot + k_1]\Big) \cdot \\ &\qquad \psi^2_{\lambda_2^2} \star \sum_{k_1=-K_1}^{K_1} \operatorname{sgn}\big(\psi^1_{\lambda^1} \star f\big)[\cdot + k_1]\,\psi^1_{\lambda^1}[\cdot + k_1 - j]\Big)[i]\end{aligned}$$

which is not terribly enlightening, though it tells us that the input function $\boldsymbol{f}$ comes through in two sign functions: the first is the sign of the internal second layer term, while the second is similar to that in the first layer Eq. (4.3).

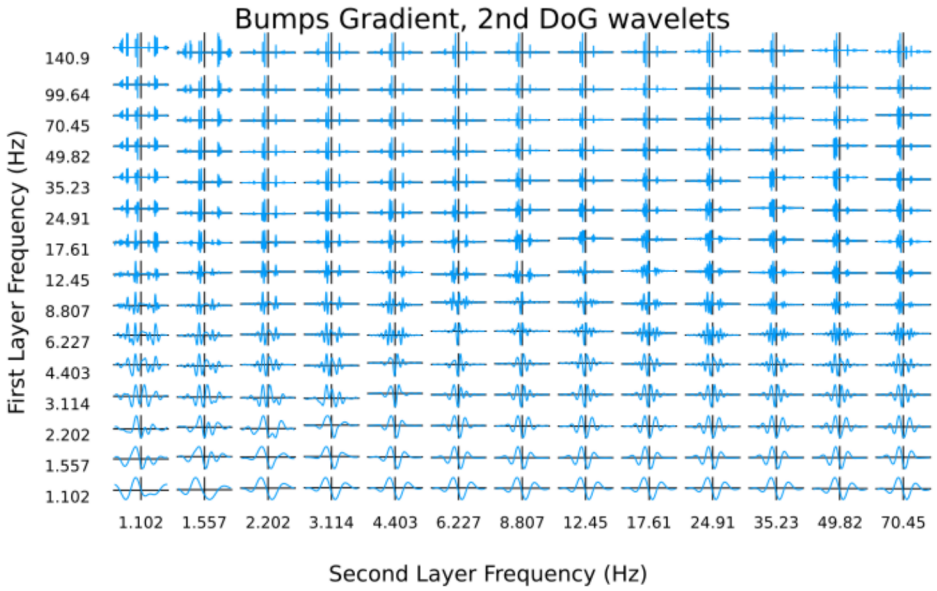


FIGURE 4.8. The gradients for each path, with the first layer varying along the vertical axis and the second layer varying along the horizontal axis. The output location is 16/32 for every path. This is specifically the Bumps signal

For the second layer gradients, there are both common features and some major differences between the fit results. The gradients of the bumps function in Fig. 4.8 are the most different from the remaining figures, with a similar set of copies of the wavelet at each spike as in Fig. 4.6a, however instead of uniformly decreasing in magnitude away from the center, the magnitude at each peak depends on the path, with the high/low

frequency paths (such as $(1.102\text{Hz}, 140.9\text{Hz})$[2] in the upper left, where this is most pronounced) having an oscillation in the magnitude of the response.

All of Fig. 4.9, Fig. 4.10, Fig. 4.11, and Fig. 4.12 on path $(1.102\text{Hz}, 99.64\text{Hz})$ have envelopes with a similar scale of variation and high internal frequency, though the exact shape of the envelope differs between them. The envelope consists of 3 lobes, with the center lobe larger than either of the side lobes. This is a reasonable description of the magnitude of the DoG2 wavelet itself. Also worth noting is that these envelopes all decrease in scale as the second layer frequency increases (so along the top row), as we would expect if they correspond to the second layer wavelet. A similar effect is happening in Fig. 4.8, but is somewhat obscured by the sparsity of the signal, so in effect we are only sampling from this envelope at the actual peaks in the signal.

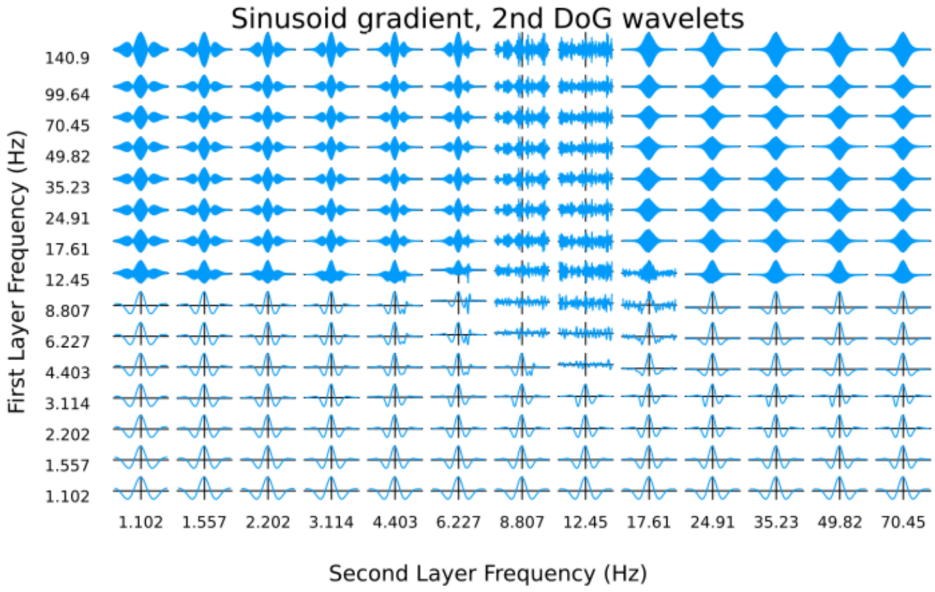


FIGURE 4.9. This is specifically for the 31.25Hz Sinusoid, same format as Fig. 4.8

In the cylinder signal the two edges show up as spikes along the paths in the 4th row from the bottom, with first layer frequency 3.114Hz. Somewhat more puzzling is the sign switch from the lowest frequency path $(1.102\text{Hz}, 1.102\text{Hz})$ to any of the other paths for the cylinder signal; at $(1.102\text{Hz}, 1.102\text{Hz})$, the central peak is positive, while for other low frequency paths where there is a central peak, it is negative.

The paths in the first row from the bottom, such as $(70.45\text{Hz}, 1.102\text{Hz})$, have almost identical responses to the first layer wavelets (compare Fig. 4.8 and the low frequencies of Fig. 4.6a, or Fig. 4.10 with Fig. 4.4).

[2]following the convention of (second layer frequency, first layer frequency) established in Section 3.3.3

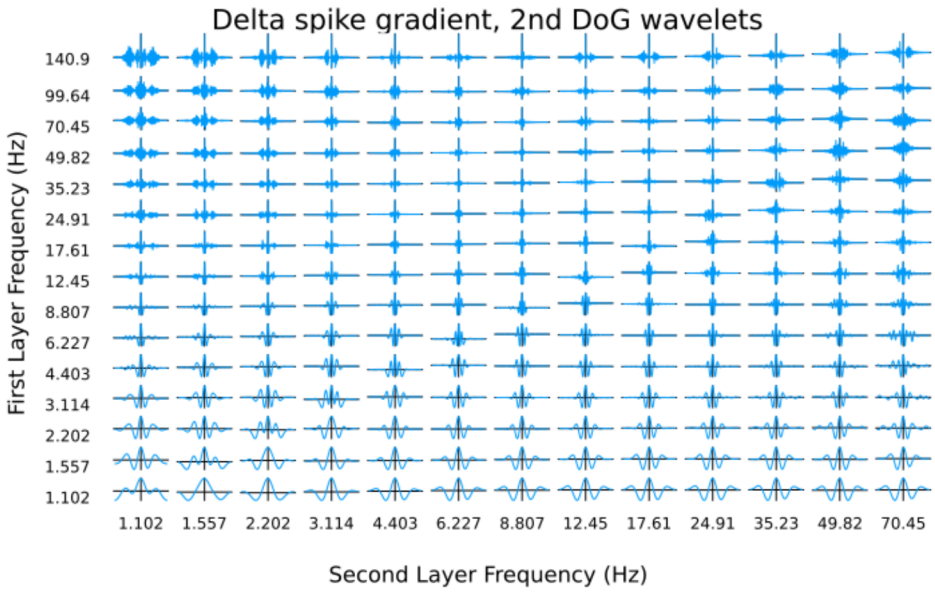


FIGURE 4.10. Delta function, same format as Fig. 4.8

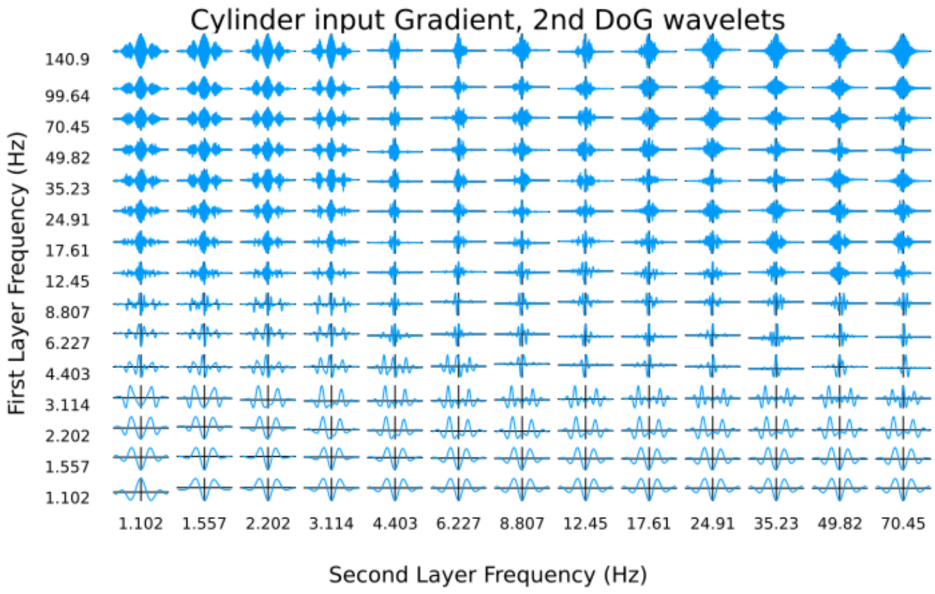


FIGURE 4.11. Cylinder Signal, same format as Fig. 4.8

4.1.4. Summary. So having examined the gradients from each layer, the kinds of signals that most increase the value of the ST coefficients in different layers differ dramatically. For the zeroth layer coefficients, the most effective way to increase those coefficients is to simply add the original averaging function translated to that output location. For the first layer coefficients, the most effective increase depends on the original

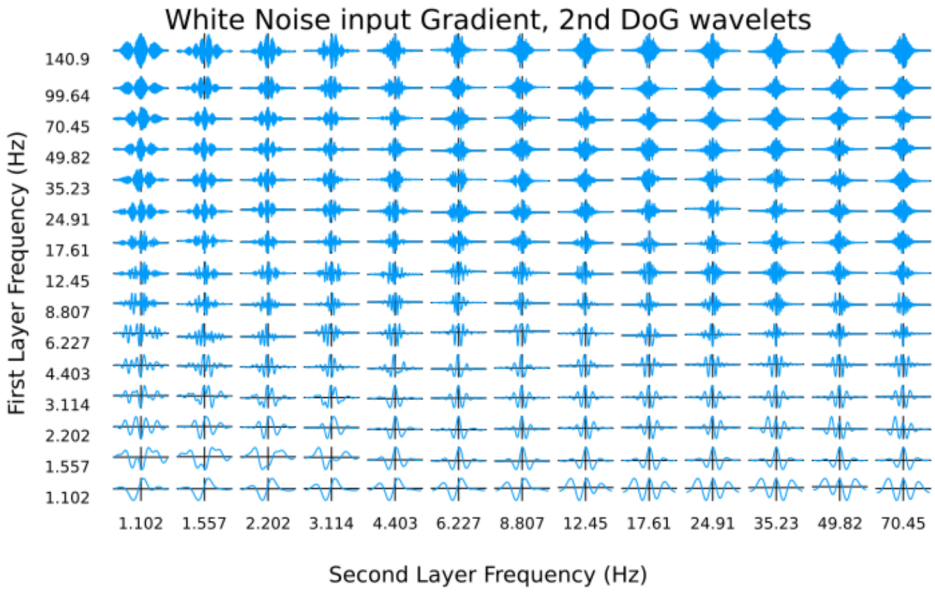


FIGURE 4.12. White noise, same format as Fig. 4.8

signal. It consists of adding copies of the original wavelets, located at or near the discontinuities already present in the signal, with the importance of different discontinuities modulated by the father wavelet located at the output location. For the second layer coefficients, in the case of paths with decreasing frequency, the most effective way to increase the coefficients is to repeat copies of the first layer wavelet, with magnitude distributed according to the second layer wavelet. It is difficult to interpret the paths with increasing frequency using the methods in this section. We now turn to pseudo-inversion, where instead of seeking local methods of increasing the output for a particular coordinate, we seek signals which maximize a particular coordinate.

4.2. Pseudo-inversion

While finding inputs which most activate a particular path requires some notion of inversion like the previous methods, it differs from them in that we expect the target to not actually be in the image of the scattering transform, so we will need to relax the notion of "inverse" somewhat. In the discrete case as discussed in Section 3.3.1, the image of $\Phi\left|\boldsymbol{f}\right]$ is a subset of collections of vectors of the form $Y := \left\{\boldsymbol{y}\left|q^m\right]\right\}_{q^m\in\mathfrak{L}^m, m\in\mathbb{Z}_0}$ where $\boldsymbol{y}\left|q^m\right]$ has the same shape and size as the output $\boldsymbol{s}\left|q^m\right]$. Some of these collections Y will correspond to the output of a particular input, but one can easily construct examples that aren't; for example, if any path in the first layer has a negative entry. For collections $Y = \Phi\left|\boldsymbol{f_0}\right]$ which are in the output, a problem where the

original $\boldsymbol{f}_0$ is an optimum is ℓ^2-minimization:

$$\min_{\boldsymbol{f}} \sum_{p\in P} \left| \boldsymbol{s}|\,p\right](\boldsymbol{f}) - \boldsymbol{y}|\,p\right] \Big|_2^2. \tag{4.6}$$

This will be true for arbitrary subsets $P \subseteq \bigcup_{m=0}^{\infty} \mathfrak{L}^m$, with more paths decreasing the possible range of alternative minima. See Appendix A.3 for some examples doing this for several simultaneous target paths.

This formulation of inversion naturally extends to Y not in the image of Φ, for which it is finding the closest point $\boldsymbol{s}|\,p\right](\boldsymbol{f})$ in the image of Φ and its inverse $\boldsymbol{f}$. One issue with doing just the ℓ^2-norm minimization is that if the zeroth layer is not in P, then for any $c \in \mathbb{R}$, $c + \boldsymbol{f_0}$ is also a minimizer, since for the wavelets of interest, we have that $\widehat{\psi}(0) = 0$. To deal with this, we impose a regularization term $\mu|\,\boldsymbol{f}\,|_2$:

$$\min_{\boldsymbol{f}} \left\{ \mu|\,\boldsymbol{f}\,|_2^2 + \sum_{p\in P} \left| \boldsymbol{s}|\,p\right](\boldsymbol{f}) - \boldsymbol{y}|\,p\right] \Big|_2^2 \right\}$$

A problem with an equivalent set of minima that that frames finding $\boldsymbol{s}|\,p\right]$ that most aligns with $\boldsymbol{y}|\,p\right]$ is

$$\min_{\boldsymbol{f}} \left\{ \mu \Big| \boldsymbol{f} \Big|_2^2 - \sum_{p\in P} \left\langle \boldsymbol{s}|\,p\right](\boldsymbol{f}), \boldsymbol{y}|\,p\right] \right\rangle \Big\}. \tag{4.7}$$

We will generally use this formulation of the problem, since it more directly ties the regularization parameter μ to the values of $\boldsymbol{s}|\,p\right](\boldsymbol{f})$. In the case of computing the reconstruction of real examples, we can choose μ by minimizing the difference between $\boldsymbol{f}$ and the original function $\boldsymbol{f}_0$. However, in the case of pseudo-inversion, it is not clear what the correct choice of μ should be. In the absence of a reason for a particular choice, we choose 1×10^{-5}, since this resulted in solutions that have values around 1.

4.2.1. Optimization methods. This is a form of regularized nonlinear least squares, where each residual is $r_{p,i}(\boldsymbol{f}) = s_i|\,p\right](\boldsymbol{f}) - y_i|\,p\right]$. The differentiability of $r_{p,i}$ comes down to that of $s|\,p\right](\boldsymbol{f})$, which is almost everywhere differentiable, since $\cdot$ is almost everywhere differentiable. However, in practice several variations on gradient descent or approximate gradient descent, such as BFGS [**60**], stochastic gradient descent [**26**], and PDFO [**64**], all failed to converge in reasonable time frames. This is most likely because across zero, the gradient flips sign. In combination with the highly oscillatory nature of the solutions, each gradient step flips the sign at a different time sample, leading to a combinatorial search despite using the gradient information. In practice this means that gradient or approximate gradient-based methods frequently get stuck around saddle nodes. As a result, non-gradient-based methods such as *differential evolution* have proven more

effective at solving the problem [**63**]. The specific version we use is the implementation of DE/rand/1/bin in BlackBoxOptim.jl[3], with some extra pre- and post-processing steps.

In general, to minimize a function h, evolutionary computation works by iteratively updating a population of candidate agents $i = 1, \ldots, P$ for each generation g using some update rule, and only keeping solutions which sufficiently improve the accuracy. Each candidate i during generation g consists of a candidate solution $\boldsymbol{X}^{i,g}$ and various associated parameters. DE/rand/1/bin's update rule, applied to each candidate $\boldsymbol{X}^{i_0,g}$ in the current population has two steps: first, we take three *other* randomly chosen current candidates $\boldsymbol{X}^{i_1,g}$, $\boldsymbol{X}^{i_2,g}$, and $\boldsymbol{X}^{i_3,g}$, and create an intermediate vector

$$\boldsymbol{V}^{i_0,g} = \boldsymbol{X}^{i_1,g} + F^{i_0,g} \cdot (\boldsymbol{X}^{i_2,g} - \boldsymbol{X}^{i_3,g}) \tag{4.8}$$

where $F^{i_0,g}$ is a parameter of the agent i_0. Next, with some crossover probability $r^{i_0,g}$ (which also depends on the agent i_0), each coordinate of the new candidate is

$$Y_k^{i_0,g} = \begin{cases} X_k^{i_0,g} & : \text{with probability } r^{i_0,g} \\ V_k^{i_0,g} & : \text{with probability } 1 - r^{i_0,g} \end{cases} \tag{4.9}$$

Finally, the actual objective function h comes into play: if $h(\boldsymbol{Y}^{i_0,g}) < h(\boldsymbol{X}^{i_0,g})$, then we replace $\boldsymbol{X}^{i_0,g+1} = \boldsymbol{Y}^{i_0,g}$. Otherwise, we keep the same candidate $\boldsymbol{X}^{i_0,g+1} = \boldsymbol{X}^{i_0,g}$, and reselect F^{i_0} and $r^{i_0,g}$ from the Cauchy distribution[4].

With this context, the somewhat opaque name DE/rand/1/bin refers to *D*ifferential *E*volution, with the base vector $\boldsymbol{X}^{i_0,g}$ in Eq. (4.8) chosen *rand*omly, with *1* difference added, and the final candidate $\boldsymbol{Y}^{i_0,g}$ chosen so the mutated locations follow a *bin*omial distribution.

We use a couple of adaptations that are particular to solving Eq. (4.7).

(1) We perform all updates to the candidates $\boldsymbol{X}^{i,g}$ after a discrete cosine transform (DCT). Applied directly to the space domain coefficients for Eq. (4.7), both Eq. (4.8) and Eq. (4.9) will frequently break any level of continuity in the candidate solutions, as there is no relation between coordinates. By representing the candidates in frequency, mutation will maintain the frequencies used throughout

[3] https://github.com/robertfeldt/BlackBoxOptim.jl

[4] The randomization of F^{i_0} and $r^{i_0,g}$ isn't strictly part of the DE/rand/1/bin algorithm, but is a common enough variant, e.g. [**13**].

the population. As one might expect for so stochastic a method, for any particular run, this may or may not improve convergence, but on average the DCT representation converges faster.

(2) We initialize the candidates non-uniformly in each coordinate. Given the power-law spacing of the wavelets in frequency, we initialize the candidates with a colored noise distribution, so if $\widehat{X}_k$ is the fast Fourier transform of $\boldsymbol{X}$, then for positive frequencies $k \geq 0$ the initial value is distributed as

$$\widehat{X_k^{i,1}} \sim \frac{\mathcal{N}(0,1) + \mathrm{i}\mathcal{N}(0,1)}{(k+1)^\alpha}$$

while for $k < 0$, $\widehat{X_k^{i,1}} = \overline{\widehat{X_{-k}^{i,1}}}$ to maintain $\boldsymbol{X}^{i,1}$ real. Here for a given signal α is chosen uniformly at random between .5 and 3 to allow for a wide distribution of smoothness. In practice, the frequently-used wavelets have minimal support in the very high frequencies, so noise in these frequencies takes a very long time to converge to zero. Since we know these should be less represented, we start with their values much smaller, but still non-zero and with varying relative magnitude.

(3) We perturb the worst fifth of the population using the same colored noise distribution every 4 minutes to guarantee that the process will eventually converge, in a manner similar to [**1**], though less frequent and independent of the agent. Differential evolution without some method of adding noise is also potentially susceptible to being trapped in local minima, which this modification avoids. Both 4 minutes and a fifth of the population were chosen heuristically, with 4 minutes approximately the time required for a single second layer target to converge. Any choice for the fraction perturbed will work, as long as the fraction of the population changed is not too large to disrupt convergence entirely, or not too small to escape local minima.

Finally, because black box optim cannot of some peculiarities of the implementation in BlackBoxOptim.jl, we adjust Eq. (4.7) to an equivalent form (with correct choice of μ)

$$\min_{\boldsymbol{f}} \exp\left(\log(1.1)\left(\mu|\,\boldsymbol{f}\,|_2^2 - \sum_{p\in P}\left\langle \boldsymbol{s}|\,p\right](\boldsymbol{f}), \boldsymbol{y}|\,p\right]\right\rangle\right)\right),$$

where the base 1.1 is chosen to avoid the objective function growing too quickly.

4.2.2. Fitting Single Coordinate. We start with the simplest possible version of Eq. (4.7): fitting a single coordinate with output location k/m at path p, or $\boldsymbol{y}\big|\, p\big] = \boldsymbol{e}_k$, and the value in every other path is ignored.[5] In the case of a wavelet transform, a pseudo-inversion targeting the wavelet output at time i and scale j results in the corresponding wavelet of scale j translated to the time i. The zeroth layer in Section 4.2.3 is the most straightforward, as it simply returns the averaging wavelet ϕ^m. One might expect the first layer, discussed in Section 4.2.4 to resemble the wavelet transform for the first layer, but the averaging by ϕ^m, absolute value, and subsampling cause some unexpected changes. The second layer in Section 4.2.5 is the most interesting.

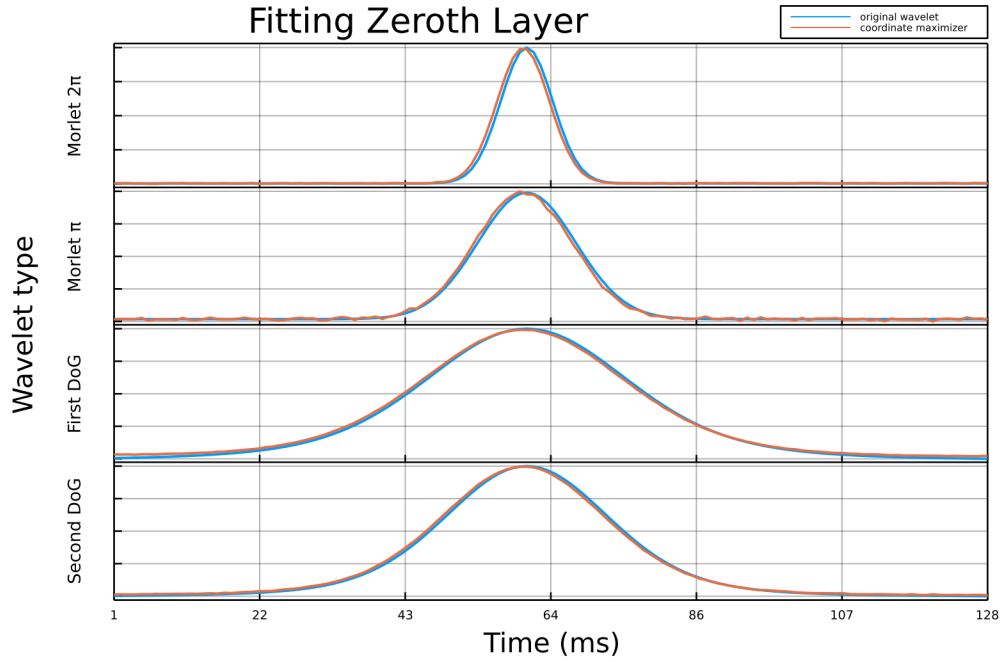


FIGURE 4.13. Comparing the fit solutions for the zeroth layer with the corresponding father wavelet.

4.2.3. Zeroth Layer Coordinate. The zeroth layer result is simply a subsampling of the averaging wavelet, so results in exactly recreating the father wavelet, as can be seen in figure Fig. 4.13. Here we are maximizing the output location $8/16$, which in the input space corresponds to 60ms, rather than the exact center 64ms, which isn't in the output space because the output grid has an even 16 coordinates.

4.2.4. First Layer Coordinate. As was noted in Section 4.1.2 for the gradient, the first layer differs from a continuous wavelet transform through the absolute value, averaging, and father wavelet. The results in

[5] We have also tried minimizing every other path simultaneously. However, if we apply the same pseudo-inversion technique to a wavelet transform, maximizing one output $W\big|f\big](\lambda_k, i/m)$ and minimizing the rest results in an extremely oscillatory solution not at all like the original wavelet. See Appendix A.1.1 for a discussion of why.

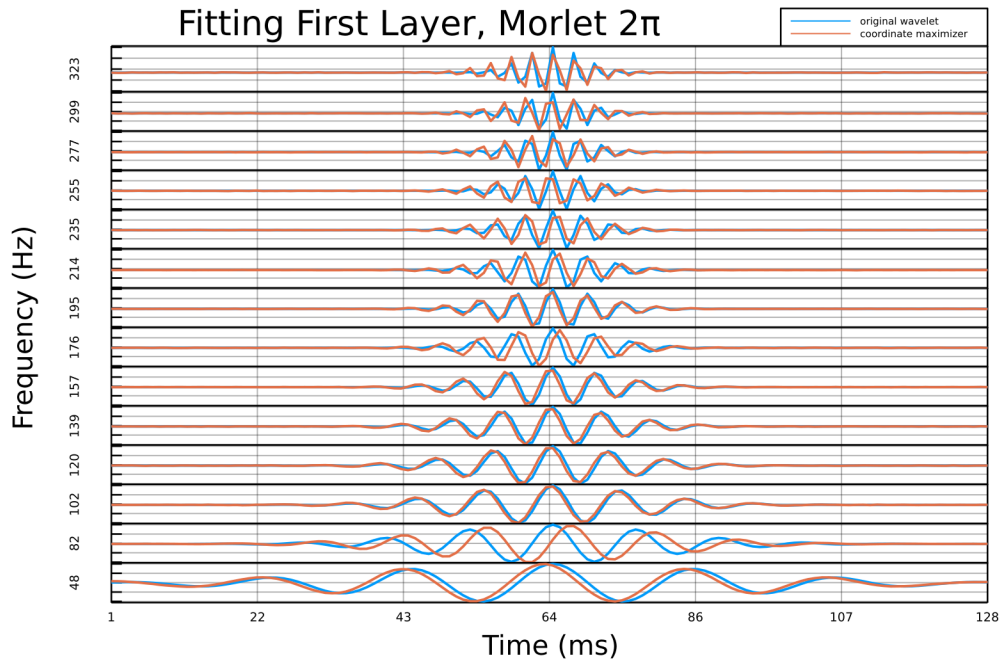


FIGURE 4.14. Comparing the fit solutions (target frequency along the y-axis) with the corresponding Morlet wavelet with mother wavelet mean frequency 2π translated to the input time matching $s_7\big|\lambda^1\big]$. The original wavelet is in blue while the fit solution is in orange.

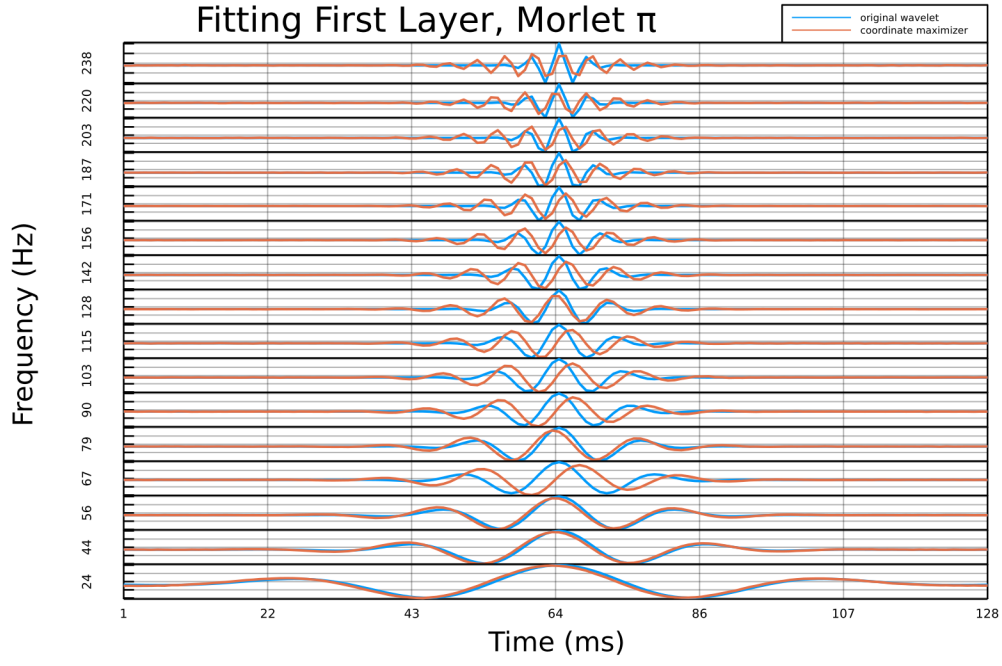


FIGURE 4.15. Similar figure to Fig. 4.14 but with a Morlet wavelet of mean frequency π instead.

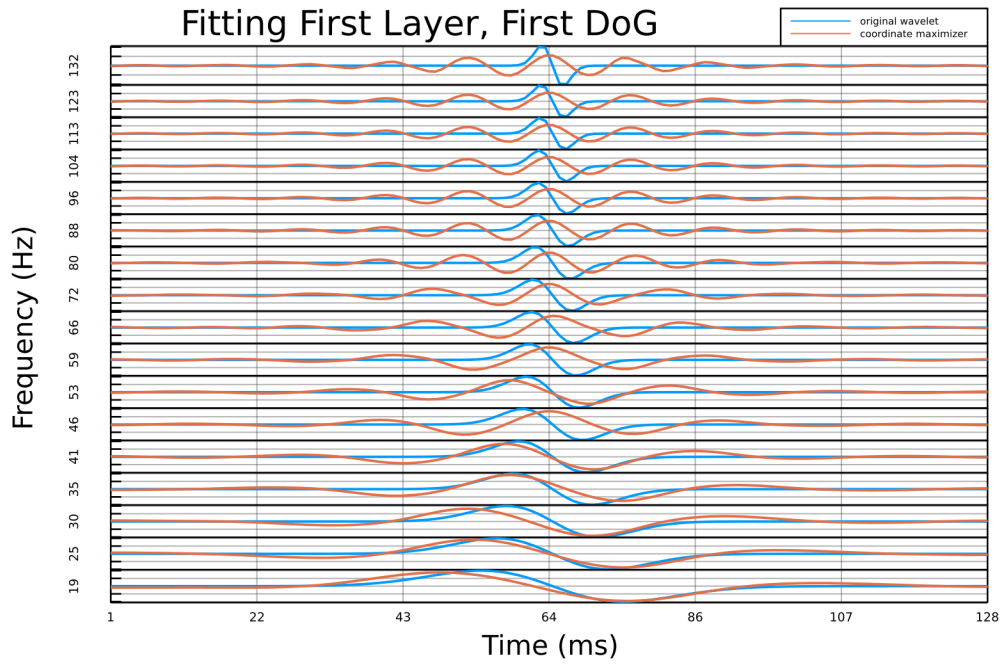


FIGURE 4.16. Similar figure to Fig. 4.14 but with a first derivative of a Gaussian wavelet instead.

that section suggest that we should expect the low frequency wavelets to simply maximize the corresponding wavelet coefficients, while as the frequency increases, and it is possible to fit more copies of the target wavelet within the envelope of the father wavelet, we should expect multiple, partially overlapping copies of the wavelet at the correct frequency.

In each of Fig. 4.14, Fig. 4.15, Fig. 4.16, and Fig. 4.17, we compare the (sign corrected) fit solution (orange) at a centrally located output time (specifically output location $7/11$ with varying paths) and at each scale with the original wavelet (blue). The first major difference from simply maximizing the corresponding wavelet coefficient is that absolute value eliminates the phase of the wavelet. The output is $\phi \star \psi_i^m \star f = \phi \star \psi_i^m \star -f$, the sign of the input is irrelevant to the error, so the signs on the actually fit solutions are arbitrary; we have multiplied by -1 whenever the peak of the solution doesn't match that of the target wavelet for easier comparison between the original wavelet and the fit solution.

In the case of Fig. 4.14 and Fig. 4.15, we are comparing the real part of the target wavelet with the fit result, which is a purely real input. This explains the solutions which are phase shifts of the corresponding wavelet, like 82Hz or 176Hz in Fig. 4.14, or most of the frequencies in Fig. 4.15. The effects of the averaging wavelet are more pronounced for the derivative of Gaussian wavelets in Fig. 4.16 and Fig. 4.17, where at high

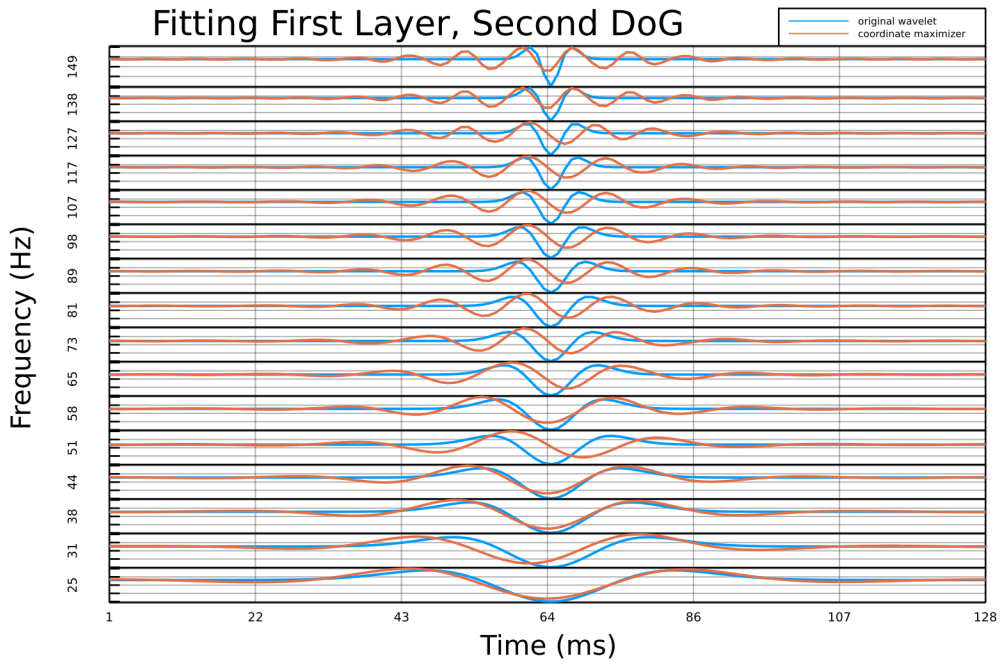


FIGURE 4.17. Similar figure to Fig. 4.14 but with a DoG2 wavelet instead.

frequency the fit solution is effectively an enveloped sinusoid. The difference in how much the high frequency fits spread away from the corresponding wavelet corresponds to the width of the averaging wavelets in Fig. 4.13, with appreciable deviation for both the First and Second DoG wavelets in Fig. 4.16 and Fig. 4.17, while both of the Morlet wavelets have significantly less spread.

Another somewhat unusual feature for the 2nd DoG wavelets is the offset of the peaks from the original peak. The solution at many of the frequencies in Fig. 4.17 is zero or nearly zero at 64 (where the central peak of the original wavelet is). As with the wider spatial support of the solution, this effect kicks in more dramatically as frequency increases. This is a second effect of the averaging wavelet, and may be caused by the ratio between the support of the wavelet at that scale and the averaging function. Depending on the ratio, more copies of the wavelet may be fit by adjusting how well they line up with the target frequency.

4.2.5. Second Layer Coordinate. The second layer fits are the first that differ quite fundamentally from the original wavelets. In Fig. 4.18 we have a collection of fit solutions using the Morlet wavelet with mean frequency π. The plots are laid out on a grid, where the plot's y-axis location corresponds to the frequency used in the first layer, while the x-axis location corresponds to the frequency in the second layer. To simplify interpretation, the output location is fixed to $3/7$ for all paths.

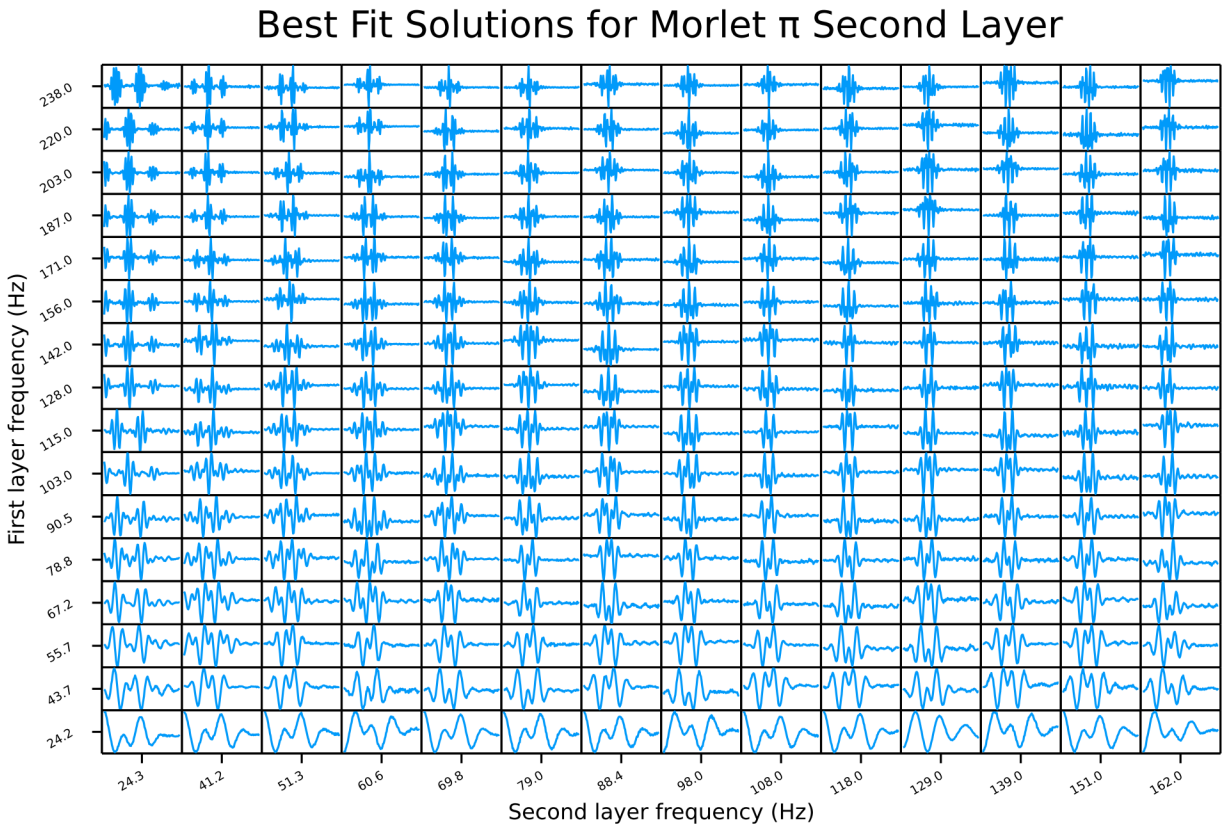


FIGURE 4.18. Fitting all paths using Morlet wavelets of mean frequency π. Each subplot is a separate fit, which is 1 at exactly the output location $3/7$ and zero everywhere else.

It is somewhat hard to discern what is going on in the spatial domain in Fig. 4.18, so instead we will look at scalograms with the same layout. The scalograms in Fig. 4.19 correspond to the plots in Fig. 4.18. We use the same wavelet transform used in the scattering transform we are examining in that figure, and square the absolute value of the coefficients as in a standard scalogram. For more examples of space domain plots, see Appendix A.

In Fig. 4.19 it is clearer to see what is happening. Consider the path with frequency 171Hz in the first layer and 88.4Hz in the second (on the middle right of the figure, it may help to zoom in quite a bit). At the particular target frequency and neighboring frequencies, the response is oscillatory. To more clearly see the effects of adjusting the frequency in either layer, compare the several figures in the upper left of the figure, with first layer frequencies between 203Hz and 238Hz, and second layer frequencies between 24.3 and 51.3. Each of these figures roughly consists of 3 peaks. As the first layer frequency increases (comparing plots vertically), the average frequency of the peaks increases (moves up *within* a given plot). As the second layer

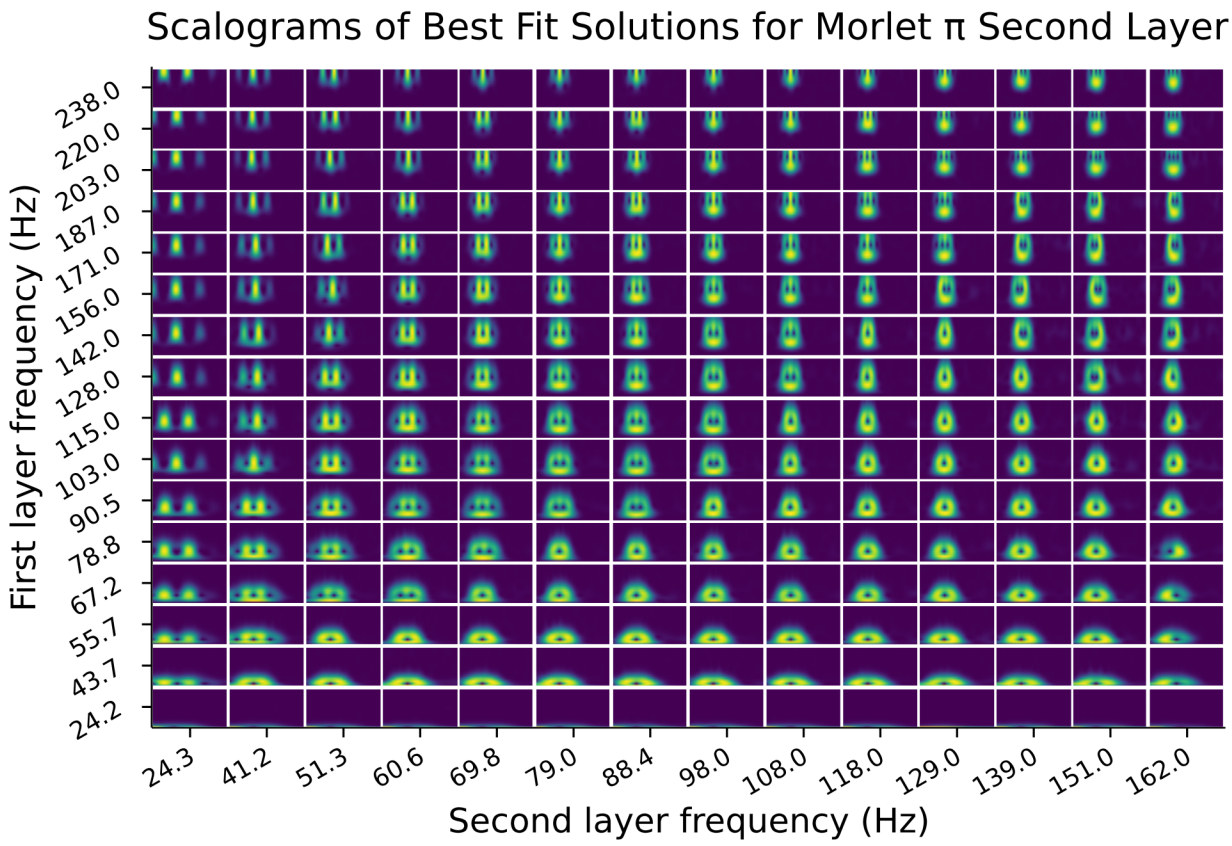


FIGURE 4.19. The scalograms corresponding to the signals in Fig. 4.18.

frequency increases (comparing plots horizontally), the distance between the peaks decreases. So the second layer frequency corresponds to oscillations in the envelope for a particular frequency in the first layer. The total number of oscillations increases as the second layer frequency does, e.g. the two plots at second layer frequency 118Hz.

An inconsistent feature of these scalograms is the amount of response at frequencies above and below the first layer target frequency. For Fig. 4.18, this happens consistently for first layer frequency above 24.2Hz and second layer frequency above 69.8Hz.

Moving on to the other wavelets, the higher frequency Morlet wavelet in Fig. 4.20 follows similar patterns, although as expected the envelope is higher frequency. This is particularly visible for the highest frequency in the first layer 323Hz and the lowest in the second layer. On the other hand, the real wavelets in Fig. 4.21 and Fig. 4.22 generally don't follow this pattern.

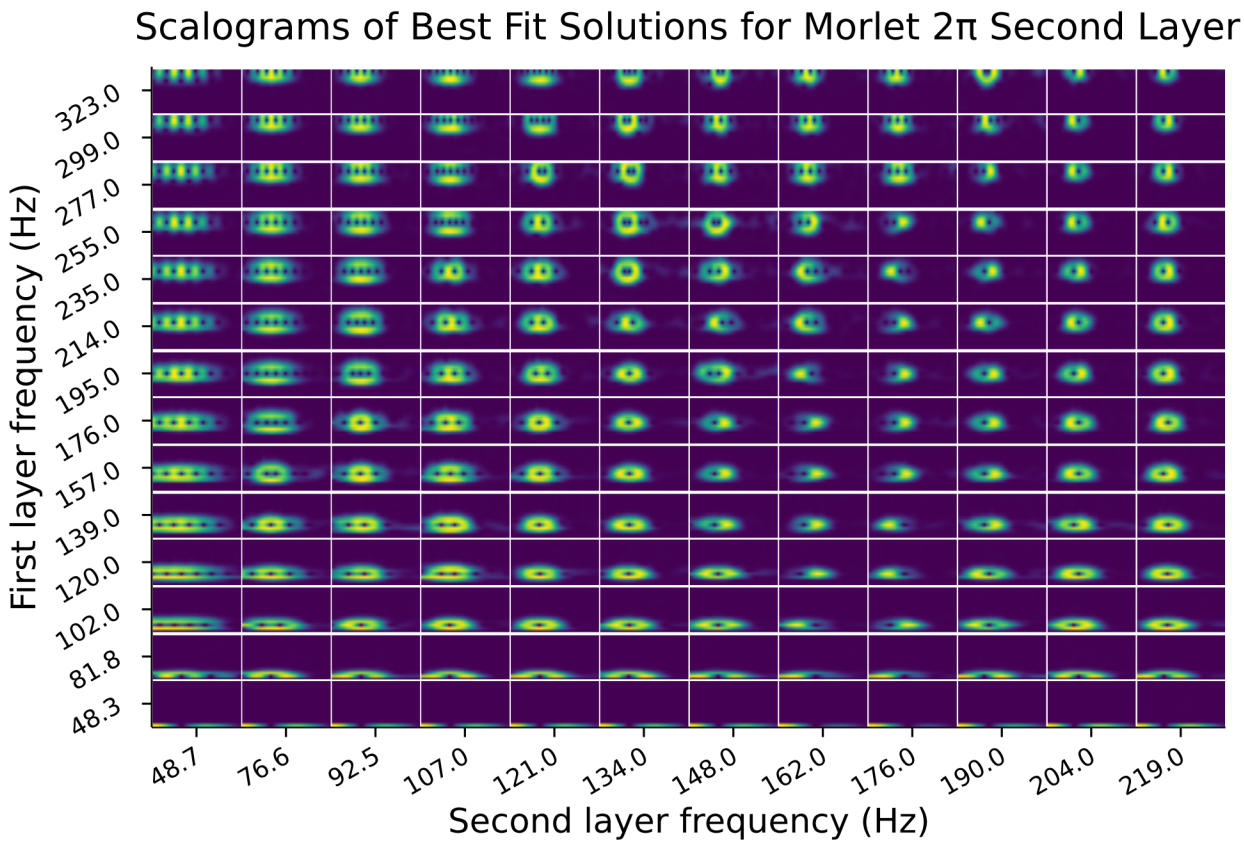


FIGURE 4.20. Scalograms for fit signals with a Morlet mean frequency of 2π.

For the case of the first DoG in Fig. 4.21, this is not surprising, as it is principally a first derivative detector. We should expect to see a large response at the target frequency to the left of the center, and then a region that has zero response to the right, since after taking an absolute value, there are no negative values left. This pattern is fairly clear in the first layer 24.7Hz second layer 23.9Hz figure, where there is even a secondary tail on the far right. As the first layer frequency increases, more edges are included to increase the overall response within the positive window provided by the left half of the second layer wavelet. Adjusting the second layer scale determines the width of response. The lack of negative valued inputs to the second layer gives these plots a particularly lopsided response, with most of the response happening on the left half of the plot.

The DoG2 in Fig. 4.22 is closer to the Morlet wavelet with mean frequency π in Fig. 4.19. The side lobes in the plots for the paths with first layer frequencies between 127–149Hz are significantly fainter than in the case for the morlet wavelets, but they are still clearly present, and for second layer frequency 75.5Hz

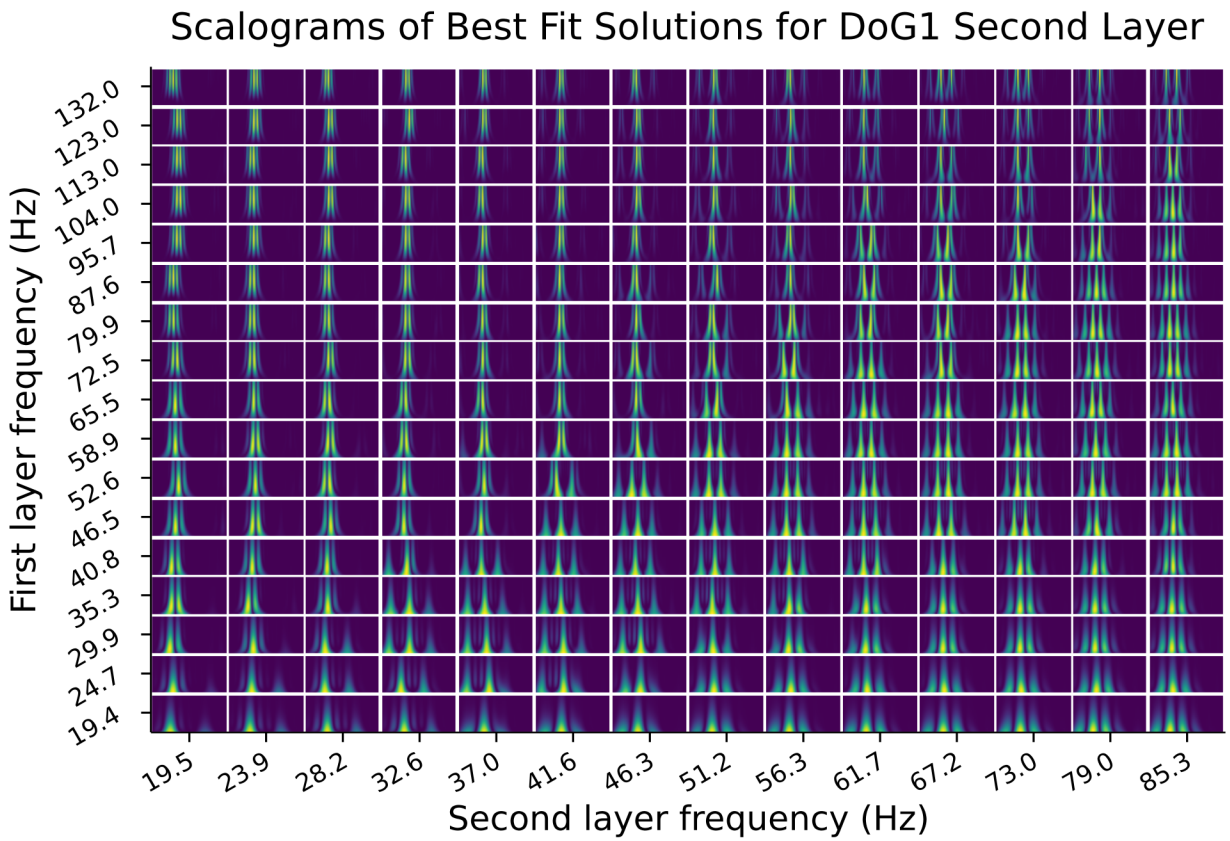


FIGURE 4.21. Scalograms for the first Derivative of Gaussian wavelets.

they are much more clearly visible. Unlike the case of the Morlet wavelets, there are no regions of high response surrounding the target wavelet; instead, the lines of wavelet maxima that are typical of wavelets with vanishing moments are much more clearly visible. Luckily, wavelets with n vanishing moments are more amenable to direct analysis, as will be discussed in the next section.

4.2.6. Summary. Much like in the case of the gradients, each of the layers has drastically different interpretation. The zeroth layer coordinates are maximized simply by the original father wavelet, shifted appropriately. The first layer coordinates overall maximizers most closely resemble the first layer gradient evaluated at the delta function; at low frequencies, they are simply the original wavelet, while as the scale decreases below the width of the father wavelet, they begin to produce many copies of the original wavelet, potentially with the sign flipped. The second layer coordinates maximizers depend more on which kind of wavelet is used. For the quasi-analytic Morlet wavelets, the maximizers are similar to the gradients; the overall envelope of the signal at the frequency of the first layer wavelet is the wavelet used in the second

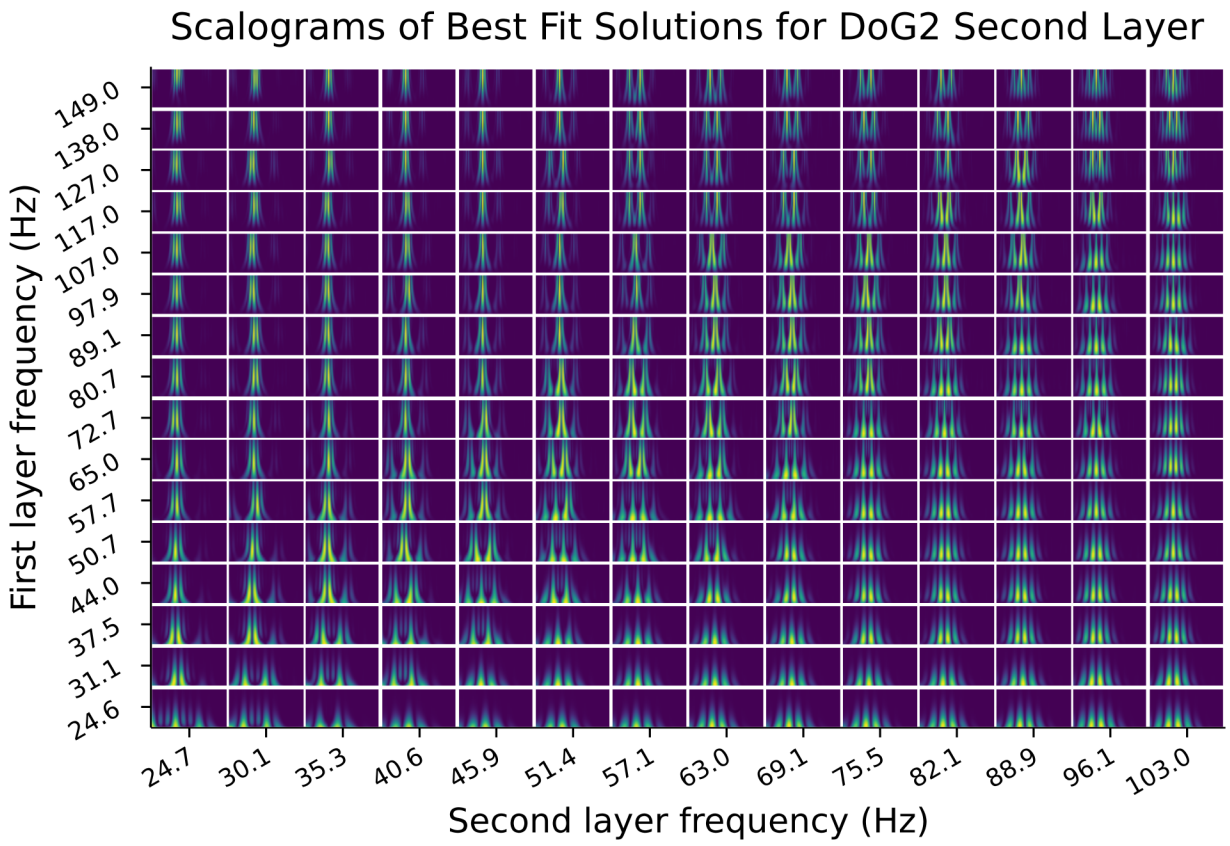


FIGURE 4.22. Scalograms for the DoG2 wavelets.

layer. For real wavelets, the sign of the wavelets used in the second layer becomes more important, with the negative portions of the second layer wavelet corresponding to zeros of the fit solution.

4.3. Theoretical Interpretation

One major advantage of real wavelets for the purpose of theoretical analysis is that they can be characterized as the nth gradient of some averaging function θ, where n gives the (maximum) number of vanishing moments, as we discussed in Section 2.3.1 [**55**, Theorem 6.2], which allows us to write $\psi^m = (-\partial)^{n_m}\theta_m$, where θ_m is an averaging function (so $\int \theta_m \neq 0$); in the case of the DoG wavelets, θ_m is just the corresponding father wavelet, but in general we don't have a guarantee of this. In this context, our frame index λ_i^m gives the ith scaling in layer m, so for convenience, define $\theta_m^{\lambda_i^m}(t) \coloneqq \theta_m\left(t/\lambda_i^m\right)$, and the kth member of this family as $\psi^{m,k_m} \coloneqq (-\partial)^{k_m}\theta_m$.To somewhat simplify our notation, we introduce the subsampling operator $R_{r_m} f(t) \coloneqq f(r_m t)$. If $q^2 = (\lambda_j^2, \lambda_i^1)$, we have for the second layer internal state that

$$
\begin{aligned}
u_2\left|q^2\right](f) &= \psi^2_{\lambda_j^2} \star R_{r_1}\rho\left(\psi^1_{\lambda_i^2} \star f\right) = (-\partial)^{n_2}\theta_2^{\lambda_2^j} \star R_{r_1}\rho\left((-\partial)^{n_1}\theta_1^{\lambda_i^1} \star f\right) \\
&= \theta_2^{\lambda_j^2} \star r_1^{-n_2} R_{r_1}(-\partial)^{n_2}\rho\left((-\partial)^{n_1}\theta_1^{\lambda_i^1} \star f\right) \\
&= \theta_2^{\lambda_j^2} \star \left(r_1^{-n_2}\sum_{k=1}^{n_2} R_{r_1}\rho^{(k)}\left((-\partial)^{n_1}\theta_1^{\lambda_i^1} \star f\right)\cdot\right. \\
&\qquad\qquad \left. R_{r_1} B_{n_2,k}\left(\theta_1^{\lambda_i^1} \star (-\partial)^{n_1+1} f, \ldots, \theta_1^{\lambda_i^1} \star (-\partial)^{n_1+n_2-k+1} f\right)\right) \\
&= \theta_2^{\lambda_j^2} \star \left(r_1^{-n_2}\sum_{k=1}^{n_2} R_{r_1}\rho^{(k)}\left(\psi^{1,n_1}_{\lambda_i^1} \star f\right)\cdot\right. \\
&\qquad\qquad \left. R_{r_1} B_{n_2,k}\left(\psi^{1,n_1+1}_{\lambda_i^1} \star f, \ldots, \psi^{1,n_1+n_2-k+1}_{\lambda_i^1} \star f\right)\right)
\end{aligned}
$$

where $B_{n,k}(x_1, \ldots, x_{n-k+1})$ are the partial (exponential) Bell polynomials, and we have used a variant of Faá di Bruno's formula for the chain rule of the n_2th derivative [**17**, **41**]. The derivative here is a distributional one, especially in the relevant cases of $\cdot$ or ReLU; for a proof that the chain rule applies as normally for distributions, see [**37**, Section 6.1]. We have also made use of the "only if" part of the vanishing moment theorem to substitute the $n_1 + n_2$ member of the wavelet family derived from θ_1. This means that $u_m\left|q^2\right](f)$ is a smoothing of a higher derivative wavelet transform at the scale λ_i^1 of the first layer times a term depending on the properties of the derivative of ρ.

For ρ being either $\cdot$ or ReLU, higher order derivatives turn these $\mathscr{C}^1$ functions into Heaviside functions and then into various derivatives of the delta function, which gives them a fairly natural interpretation as evaluating a blurred higher order derivative along the level sets of a lower order derivative. The divergent

case is when $n_2 = 1$, where we are only taking the first derivative of ρ; this is the sign function for $\cdot$ (derived in Appendix B.1) or the Heaviside function for ReLU. Explicitly, this is

$$\theta_2^{\lambda_j^2} \star r_1^{-1} R_{r_1}\Big(\operatorname{sgn}\Big(\psi_{\lambda_i^1}^{1,n_1} \star f\Big) \cdot (\psi_{\lambda_i^1}^{1,n_1+1} \star f)\Big).$$

Explicitly pulling the derivative out of the second layer wavelet has had two effects. The first is increasing the order of the wavelet from the first layer. The second is that we are multiplying this pseudo-differential operator by the sign for the original order derivative. All of this is subsampled and then blurred. Returning to Fig. 4.21, each path is maximizing the result for a second order DoG wavelet, while maintaining a positive sign for the first derivative at the same point.

For $n_2 = 2$, our terms begin proliferating:

$$\begin{aligned}\theta_2^{\lambda_j^2} \star r_1^{-2} R_{r_1}\Big(&\operatorname{sgn}\big(\psi_{\lambda_i^1}^{1,n_1} \star f\big) \cdot \Big(\psi_{\lambda_i^1}^{1,n_1+2} \star f\Big) \\ &+ \delta\Big(\psi_{\lambda_i^1}^{1,n_1} \star f\Big) \cdot \big(\psi_{\lambda_i^1}^{1,n_1+1} \star f\big)^2\Big), \\ \text{where} \qquad \delta(g) &\coloneqq \sum_{x_i \in \{x \quad g(x)=0\}} \frac{1}{g'(x_i)} \delta(x - x_i).\end{aligned} \tag{4.10}$$

This composed form of the delta function is derived in [**37**, Section 6.1] or [**32**, Section II.2.5]; thankfully g, being a convolution with a smooth wavelet, is sufficiently well-behaved for this formula to apply. The first term is much the same as for $n_2 = 1$, but the order of the first layer wavelet has been increased by 1. The second term is perhaps more interesting; it only returns a value in the neighborhood of the zero crossings of the n_1th derivative of f. Instead of evaluating the derivative at that point, however, we get the value of the $n_1 + 1$st derivative squared. Depending on the value of n_1, particularly if $n_1 = 2$, this closely resembles the more traditional singularity detection and characterization methods described in e.g. [**55**, Section 6.2].

For $n_2 = 3$ the coefficients from the Bell polynomials begin to increase in complexity, and we have the first δ' term:

$$\begin{aligned}\theta_2^{\lambda_j^2} \star r_1^{-3} R_{r_1}\Big(&\operatorname{sgn}\big(\psi_{\lambda_i^1}^{1,n_1} \star f\big) \cdot \psi_{\lambda_i^1}^{1,n_1+3} \star f + \\ &3\delta\big(\psi_{\lambda_i^1}^{1,n_1} \star f\big) \cdot \big(\psi_{\lambda_i^1}^{1,n_1+1} \star f\big)\big(\psi_{\lambda_i^1}^{1,n_1+2} \star f\big) + \\ &\delta'\big(\psi_{\lambda_i^1}^{1,n_1} \star f\big) \cdot 3\big(\psi_{\lambda_i^1}^{1,n_1+1} \star f\big)^2\big(\psi_{\lambda_i^1}^{1,n_1+2} \star f\big)\Big)\end{aligned}$$

which after some simplification, specifically using $\delta'(f)\cdot\varphi = \delta(f)\cdot\varphi'/f'$ (see [**32**, Section II.2.5]), becomes

$$\theta_2^{\lambda_j^2} \star r_1^{-3} R_{r_1}\Big(\operatorname{sgn}\big(\psi_{\lambda_i^1}^{1,n_1}\star f\big)\cdot\psi_{\lambda_i^1}^{1,n_1+3}\star f + 6\delta\big(\psi_{\lambda_i^1}^{1,n_1}\star f\big)\cdot\big(\psi_{\lambda_i^1}^{1,n_1+1}\star f\big)\big(\psi_{\lambda_i^1}^{1,n_1+2}\star f\big)\Big).$$

Through some lucky cancellation, we have reduced from three to two terms. The first resembles the sign term from the previous two, but with the derivative order increased yet again. The second term evaluates at the same points as the second term of Eq. (4.10), but the value now comes from both higher order derivatives. Finally for $n_2 \geq 2$, using the generalization of the derivative of a composition with the delta function in [**32**, Section II.2.5], we have

$$\theta_2^{\lambda_j^2} \star r_1^{-n_2} R_{r_1}\Big(\operatorname{sgn}\big(\psi_{\lambda_i^1}^{1,n_1}\star f\big)\cdot\big(\psi_{\lambda_i^1}^{1,n_1+n_2}\star f\big) + \delta\big(\psi_{\lambda_i^1}^{1,n_1}\star f\big)\sum_{k=2}^{n_2}\cdot\left(\frac{1}{(\psi_{\lambda_i^1}^{1,n_1}\star f)}\frac{\mathrm{d}}{\mathrm{d}x}\right)^{n_2-2} B_{n_2,k}\big(\psi_{\lambda_i^1}^{1,n_1+1}\star f,\ldots,\psi_{\lambda_i^1}^{1,n_1+n_2-k+1}\star f\big)\Big) \tag{4.11}$$

Which is, as may have been expected, rather a complicated formula. Away from the key points where the first layer $\psi_{\lambda_i^1}^{1,n_1}\star f$ is zero, however, the much simpler and more consistent first term predominates. This is just a smoothed version of the n_1+n_2th derivative with sign adjusted by the n_1th derivative. In the neighborhood of the zeros of $\psi_{\lambda_i^1}^{1,n_1}\star f$, the actual value has a complicated dependence on the rest of the derivatives up to n_1+n_2-1. If any of these are particularly large, as in the case of a loss of regularity, that term will dominate. So one possible interpretation of the second term is as an indicator of points where we have both that the n_1-1st derivative has an extrema, and that f loses regularity at some point before n_1+n_2-1.

In the case of real wavelets with vanishing moments, having a larger number of vanishing moments in the second layer leads to a broader set of derivatives that are picked up in the second layer coefficients. The second layer coefficients can be viewed as evaluating the derivatives between n_1 and n_1+2n_2-2 at the extrema of $f^{(n_1)}$, along with the n_1+n_2th derivative, with sign determined by that of the n_1th derivative.

4.4. Interpreting a Scattering Bell-Cylinder-Funnel Classifier

We will consider the simple signal shape separation problem, which consists of 3 prototype signals of varying lengths and shifts with added white noise. The classes are cylinder, bell, and funnel, which consist of signals that all start at some time t_0 and end at some time t_1 [**67**]. Between these two points, they are each characterized by different behaviors:

- Cylinder is just a characteristic function with height a of the interval $\lfloor t_0, t_1 \rceil$, or $a\chi_{\lfloor t_0, t_1 \rceil}$.
- Bell is a ramp starting at t_0 with a quick die off at t_1, ending with a height of a, or explicitly: $a(t - t_0) \cdot (t_1 - t_0)^{-1} \chi_{\lfloor t_0, t_1 \rceil}$.
- Funnel is the "inverse" of the bell class, starting with a jump of a at t_0 and slowly decaying to noise at t_1, or $a(t_1 - t) \cdot (t_1 - t_0)^{-1} \chi_{\lfloor t_0, t_1 \rceil}$.

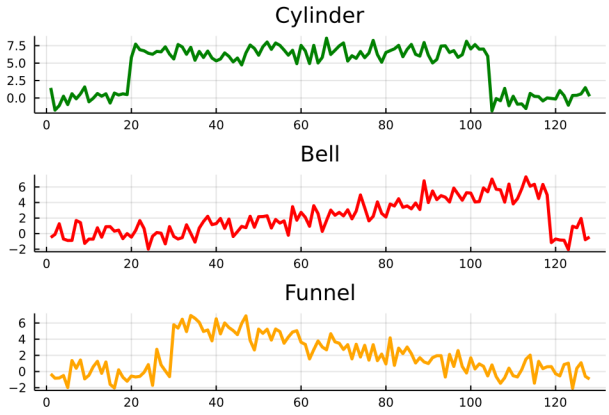


FIGURE 4.23. A couple of examples: $a = 6$ and the noise has $\sigma = 1$

A list of all classifiers on the problem can be found at the timeseriesclassification repository.[6] The best recorded accuracy on this problem in the time series classification website[7] with trained with 30 examples is 99.81%.

The classic version of this problem generates the signals in a way that correlates the time t_0 and the signal duration $t_1 - t_0$. Specifically, t_0 is uniformly random between 16 and 32, while t_1 is chosen between 32 and 96. So in addition to this classic dataset, we will consider a rotated version, which sets the starting time t_0 independently of the length $t_1 - t_0$ by circularly shifting the example by a uniformly chosen index after choosing t_0 and t_1 as above.

4.4.1. Classification Results. As a baseline reference, multinomial lasso regression on the raw coefficients of 300 examples yields a test accuracy of 93.6%, which puts it in a comparable class of problem difficulty to the MNIST dataset [**50**]. On the other hand, for the rotated version, a raw coefficient classifier

[6] https://timeseriesclassification.com/description.php?Dataset=CBF
[7] http://www.timeseriesclassification.com/dataset.php

Transform Method	Classic	Rotated	10 ex Classic	10 ex Rotated
Raw data	94.7%	47.1%	87.7%	40.3%
AVFT	57.1%	59.2%	33.5%	**56.8%**
St 1st DoG	99.5%	55.6%	**96.9%**	56.0%
St 1st then 2nd DoG	99.7%	51.8%	95.5%	52.9%
St 2nd DoG	**99.83%**	**71.2%**	94.2%	47.3%

TABLE 4.1. Percent accuracy on a test set of 1000 entries per type (so 3000 total). The first two columns use an input of 100 signals per class, while the last two use ten examples. The best classifier in each column is in **bold**.

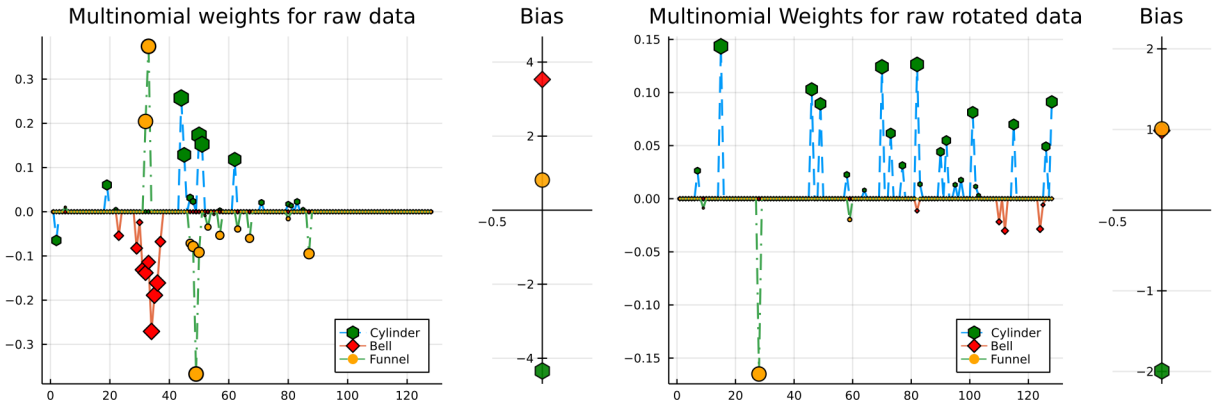
yields a test accuracy of 51.8%, which is appreciably greater than the random guessing of 33%, but because it isn't translation invariant, is still far from the best possible accuracy. For more classification results, see Table 4.1. We can use this rotated dataset to test for the role of translation invariance; unlike the raw signal classifier, the AVFT has roughly comparable accuracy on both the classic (59.5%) and rotated (57.0%) versions of the dataset. More interesting for our purpose, however, is understanding how a classifier separates these examples. As a brief reminder (and establishing some notation), multinomial lasso regression for a set of classes enumerated by $i = 1, \ldots, L$ chooses a set of weights $\boldsymbol{\beta}^i$ and offsets a_i for each class i and generates a probability distribution over the classes for an input vector $\boldsymbol{v}$ according to the soft-max

$$p_i = \frac{e^{a^i + \boldsymbol{\beta}^i \cdot \boldsymbol{v}}}{\sum_{j=1}^{L} e^{a^j + \boldsymbol{\beta}^j \cdot \boldsymbol{v}}}.$$

These regression weights $\boldsymbol{\beta}^i$ and the biases a^i are given in Fig. 4.24a. To get the prior probabilities (the classification of the null signal $\boldsymbol{v} \equiv 0$) we compute $e^{a^i}\left(\sum e^{a^j}\right)^{-1}$ which for Fig. 4.24a is 94% a bell, 6% a funnel, and only 4×10^{-4}% a cylinder.[8] Accordingly, the actual coefficients for the bell example tell us what it is *not*, while the coefficients for the cylinder generally tell us what it is. A bell is a signal that is *not* active near the front (specifically around 35), while a cylinder is a signal that is somewhat active throughout the middle section of the figure.

As expected, the coefficients for the rotated dataset in Fig. 4.24b are much less spatially concentrated; the cylinder coefficients are primarily picking up on the overall higher average value, whereas there is little to distinguish the bell from the funnel.

[8] Quite literally a rounding error.



(a) Classic bcf (b) rotated bcf

FIGURE 4.24. Regression weights for the bcf problem on raw data

4.4.2. Scattering Transform Setup. What we are looking for is to replicate a similar sort of analysis on the scattering coefficient weights and biases used to classify this data. The crux of the analysis is going from regression weights to an input example. Before that, let us examine the choice of parameters used to get Table 4.1.

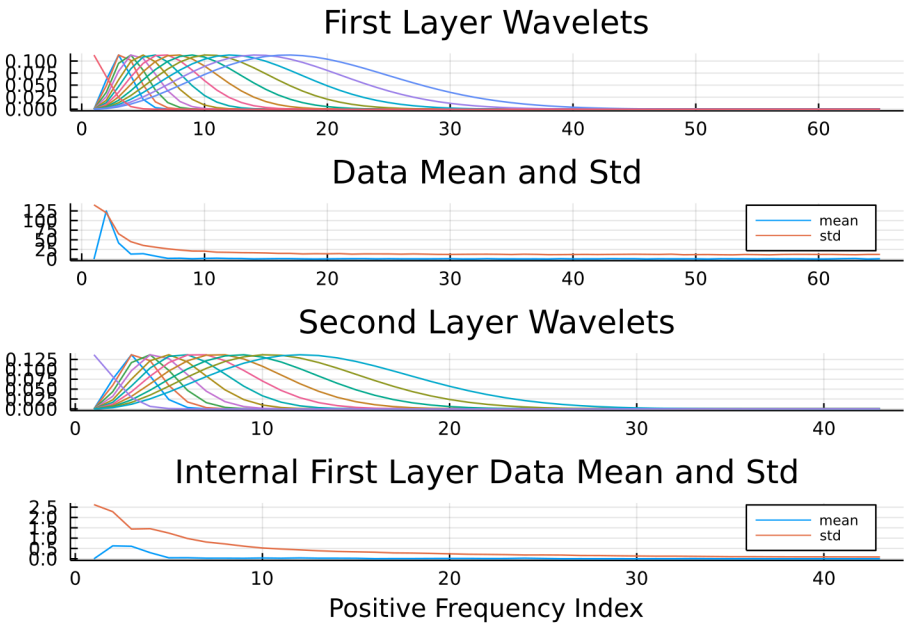


FIGURE 4.25. Choosing the wavelet frequencies to maximize the coverage of the data's mean in Eq. (4.12) and variation in Eq. (4.13). See most of Section 4.4.2 for a discussion.

As the signals can be characterized using their discontinuities, we are best off choosing some variety of real wavelet; the DoG2 performs the best, though only in the rotated 300 training examples did it stand out from the other scattering transform results in Table 4.1. To choose the proper wavelets, we first look at the positive frequency information in Fig. 4.25. The second plot gives the mean and standard deviation for each frequency across the entire 300 example training set, so if we have examples $\boldsymbol{f}^1, \ldots, \boldsymbol{f}^{300}$ from every class, then this is a plot of

$$b_k := \frac{1}{300} \sum_{j=1}^{300} \mathscr{F}\left(\boldsymbol{f} - \langle \boldsymbol{f}^i \rangle\right)_k \quad \text{for } k \geq 0 \tag{4.12}$$

$$o_k := \sigma\left(\mathscr{F}\left(\boldsymbol{f}\right)_k^i\right)_i \text{ for } k \geq 0 \tag{4.13}$$

where in this case by $\mathscr{F}$ we mean the discrete Fourier transform. The first is equivalent to the magnitude of the mean for every entry except b_0, which is zero, and the second is simply the standard deviation of $\widehat{\boldsymbol{f}}^i$ across the examples i (as opposed to the frequencies). We have subtracted out the zero frequency in order to see the variation in the other coordinates, as it is significantly larger than all other terms, and will be covered by the father wavelet no matter our other choices. Even after this, both the mean and standard deviation are quite low frequency, so we have set the averaging length as low as possible, and set the subsampling rate discussed in Section 2.2.2 to be $\beta = 1$, so there are exactly $Q = 4$ wavelets per octave. This maximizes the number of wavelets at frequencies where the data actually varies, while not choosing the hyper-parameters based on class information.

The fourth figure is the mean and standard deviation across all signals in the first layer u_1 using the settings from the first figure; it calculates the same quantities as Eq. (4.12) and Eq. (4.13), except across both the examples i and the different first layer frequencies, so the collection

$$\boldsymbol{u}\left[\lambda_1^1\right](\boldsymbol{f}^1), \ldots, \boldsymbol{u}\left[\lambda_N^1\right](\boldsymbol{f}^1), \ldots, \boldsymbol{u}\left[\lambda_1^1\right](\boldsymbol{f}^{300}), \ldots, \boldsymbol{u}\left[\lambda_N^1\right](\boldsymbol{f}^{300}).$$

is treated in the same way as the collection $\boldsymbol{f}^1, \ldots, \boldsymbol{f}^{300}$ was in the second subplot. The first layer internal coefficients have a more uniform spread of standard deviation, but still concentrates on the low frequency, so we also set the second layer $\beta = 1$, $Q = 4$. Explicitly, as used by ScatteringTransform.jl[9] the parameters are

[9] https://gitlab.com/Sonar-Scattering/ScatteringTransform-jl

Predict \ Actual	Bell	Funnel	Cylinder
Bell	998	0	1
Funnel	1	1000	2
Cylinder	1	0	997

(A) Classic

Predict \ Actual	Bell	Funnel	Cylinder
Bell	564	376	21
Funnel	393	603	11
Cylinder	43	21	968

(B) Rotated

TABLE 4.2. Confusion matrix for classifying the bcf problem using the DoG2 scattering transform. The accuracies are 99.83% and 71.2%

`cw=dcg2`, `Q=4`, β`=1`, `extraOctaves=0`, `averagingLength= -1 5,-1 5,2`, `cutputPccl=8`, `ccnvBcundary=Pericdic`, and `bcundary=PerBcundary`.

With these settings, using just the second layer on a new test set of 1000 examples per class, the confusion matrix is given in Table 4.2; on the classic problem it is nearly diagonal, while on the rotated version of the data in Table 4.2b, it effectively separates out the cylinder from the bell and funnel, but is much less effective at separating the bell and funnel from each other. On separating bell and funnel from cylinder, the accuracy is 96.8%, while for just signals classified as bell and funnel, the accuracy at distinguishing these is just 60.3%. We used a similar method to select the DoG1 and mixed scattering transform parameters.

4.4.3. Scattering Second layer Weights. Assured that the second layer of the scattering transform does in fact do well at this problem, what are the weight vectors responsible for this? In Fig. 4.26, we have a representation of the regression weights $\boldsymbol{\beta}^i$ as they correspond to the scattering domain, while in Fig. 4.27 we have the same plot, but for the rotated version. In the second layer, the weights have a spatial component as well as a two index path, so for example, the most negative cylinder weight for Fig. 4.26 is at path (λ^2, λ^1) = (17.4Hz, 20.7Hz) towards the front. Here the saturation of the color corresponds to the (signed) magnitude of the largest magnitude coefficient at that path.

As we might expect with lasso, many paths are zero (and thus not plotted), and of those that are, most coefficients are zero. Somewhat unusually, the largest magnitude coefficients are all negative, indicating that each class is best thought of as neither of the other two. This will pose difficulties for interpreting the fit later. The bell coefficients have the distinction of having a presence across all frequencies, with most paths having a single negative coordinate towards the front of the signal, while most of the funnel coefficients occur towards the rear at any given path, many of which are also negative. The one strongly positive path for the bell, (98.2Hz, 69.4Hz), has a negative value followed by a positive value, which fits the general space features,

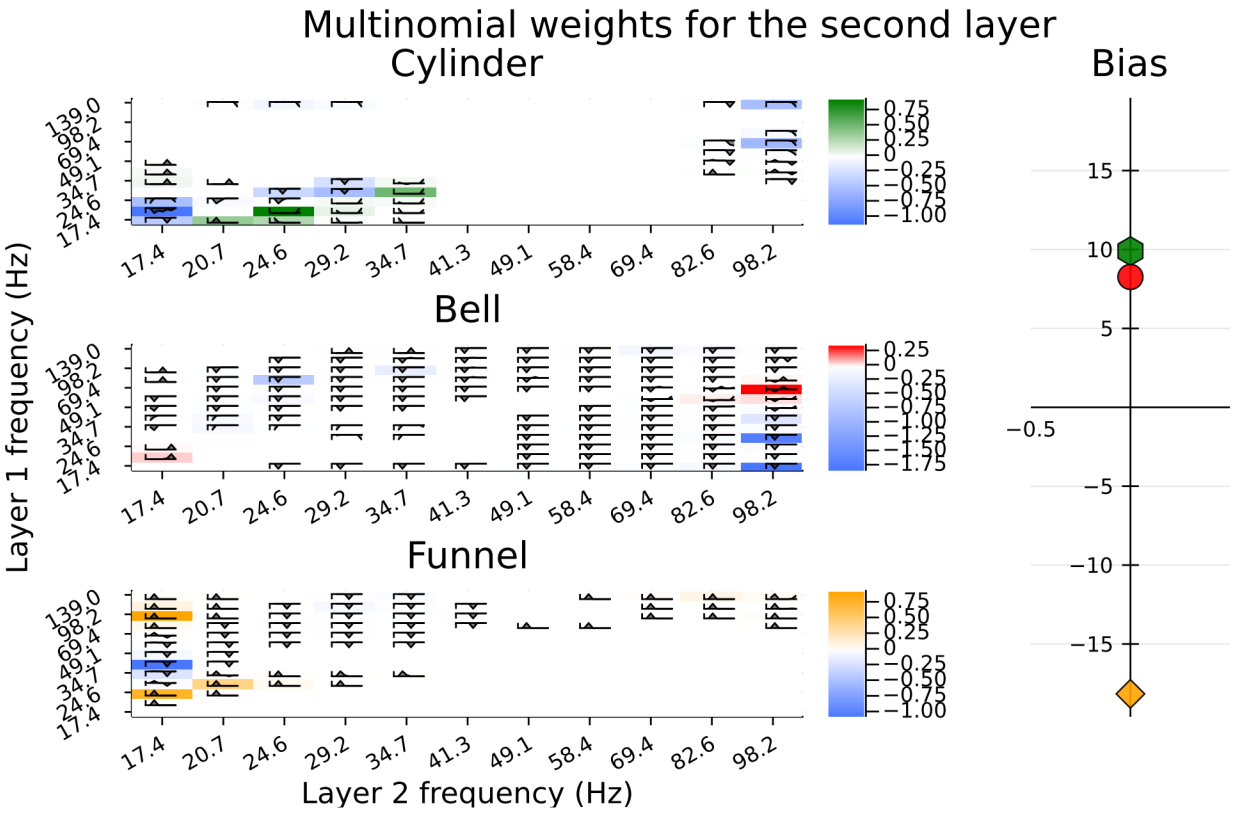


FIGURE 4.26. The weights used to classify the classic bcf problem using the second layer scattering transform, as wrapped in a set of ST coefficients. The second layer coefficients for each path consist of vectors of length 7, with the paths are along the x- and y- axis. The color represents the maximum value in that coordinate, with blue representing negative values in each plot, and each signal class having a unique positive value (green for the cylinder, red for the bell, and orange for the funnel). The colors in the bias correspond to the positive coefficient color. Only non-zero regions with at least one non-zero value have a plot of the vector.

while the largest values for the funnel occur towards the front. In the classic case, a bell then is mostly a signal that doesn't have a spike at the front, while a funnel is one that doesn't have a spike towards the back.

Such a characterization won't work for the rotated version of the problem, where the either peak could be located anywhere in the signal, so the classifier must use the difference in slopes. As we might expect, Fig. 4.27 shows less spatial consistency for the fit coefficients. The weight vectors are also considerably sparser, and the biases are almost exactly $-1/10$th as large (flipped and smaller). The paths and time locations of the larger negative coefficients for the cylinder are more consistent between the rotated and non-rotated problem, suggesting that these are responsible for the higher level of accuracy separating the cylinder in Table 4.1.

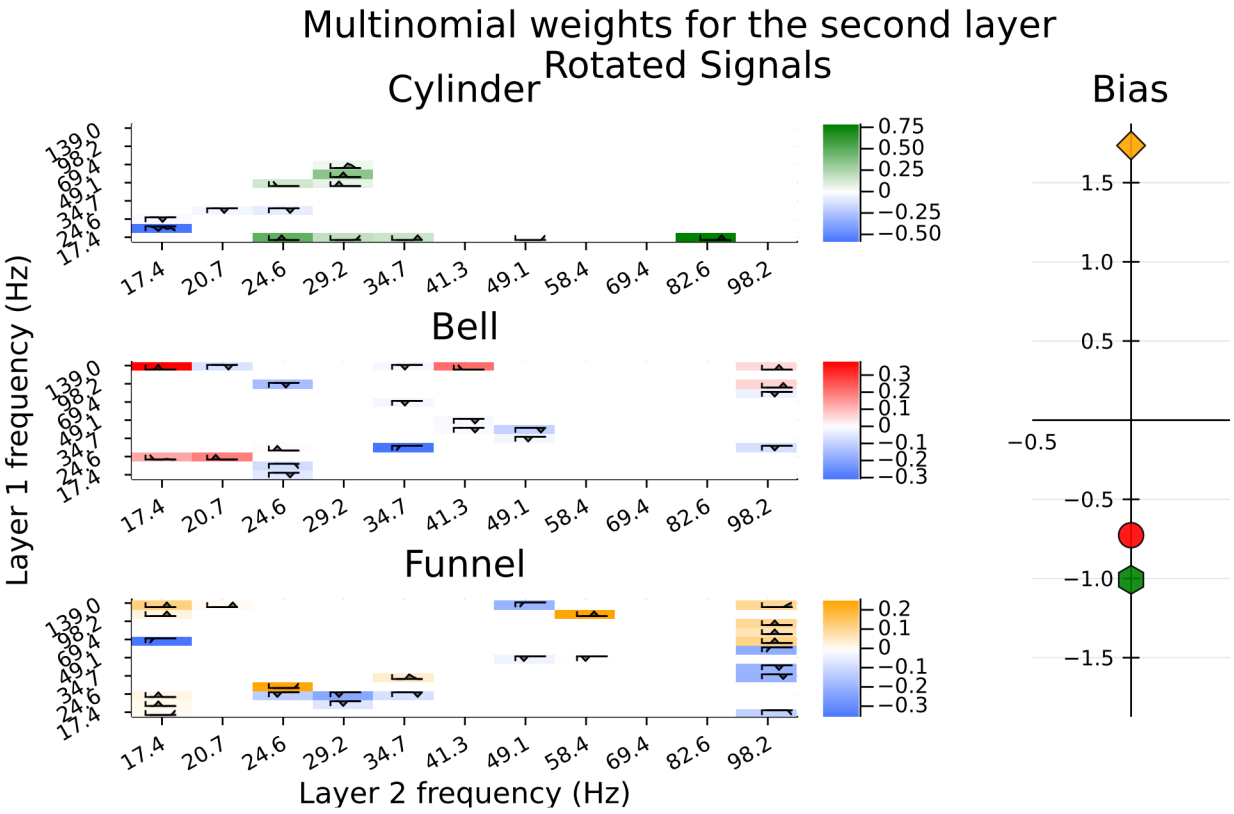


FIGURE 4.27. The weights used to classify the rotated bcf problem using the second layer scattering transform, as wrapped in a set of ST coefficients. See Fig. 4.26 for a description of the format.

On their own, these weight vectors aren't terribly informative, so we would like to see what kinds of inputs correspond most strongly to these regression vectors. Give that with the absolute value as the nonlinearity the second layer coefficients are purely non-negative, fitting these coefficients exactly would be impossible. That doesn't mean we can't find the point actually in the range nearest to the weight vector $\boldsymbol{\beta}$, however, as Eq. (4.7) finds the point $\boldsymbol{f}$ so that the output is close to $\boldsymbol{\beta}$ in a pointwise ℓ^2 sense. So explicitly, the problem we're optimizing is

$$\min_{\boldsymbol{f}} \exp\left(\log(1.1)\left(|\boldsymbol{f}|_2^2 - \sum_{q\in\mathfrak{L}^2}\langle s_x|q], \boldsymbol{\beta}^i\rangle\right)\right) \tag{4.14}$$

which is equivalent to Eq. (4.6) with $\boldsymbol{y}|q] = \boldsymbol{\beta}^i$ for all paths in the second layer, $q \in \mathfrak{L}^2$.

4.4.4. Fit Results. In Fig. 4.28 we have the pseudo-inverses $\boldsymbol{f}^i$ of the $\boldsymbol{\beta}^i$s in Fig. 4.26, where $\boldsymbol{f}^i$ is the solution to Eq. (4.14) using $\boldsymbol{\beta}^i$ as the target. It is fairly clear that the cylinder fit roughly corresponds to indicating the edges on both left and right, while the bell ramps up from left to right. As an alternate view, in Fig. 4.29 we have the scalograms of these signals. Here the cylinder clearly has large stretches of low frequency response with relatively sharp edges, while the funnel has more response towards the front. The bell has a more clearly visible edge at approximately the correct time. Both the bell and funnel fits have a more uniform high frequency response than might be ideal; this is most likely because setting `averagingLength= -1 5,-1 5,2` results in a very wide averaging function.

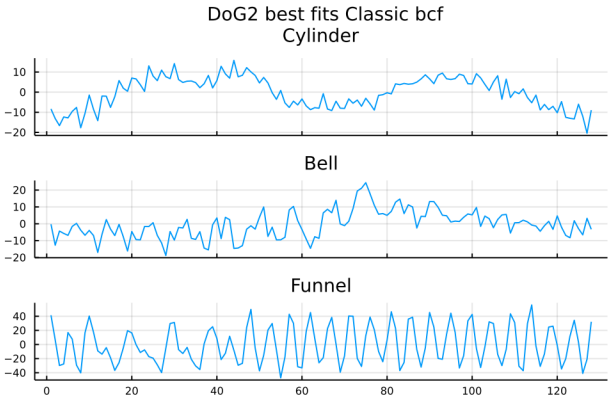


FIGURE 4.28. Pseudo-inverses of the lasso weight coefficients $\boldsymbol{\beta}^i$ for each class using only the second layer of a scattering transform using the 2nd DoG wavelets

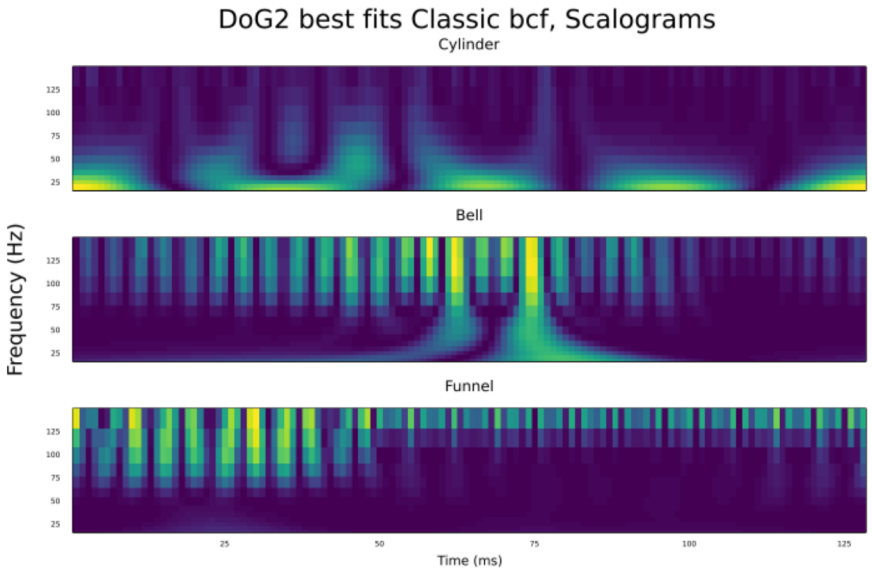
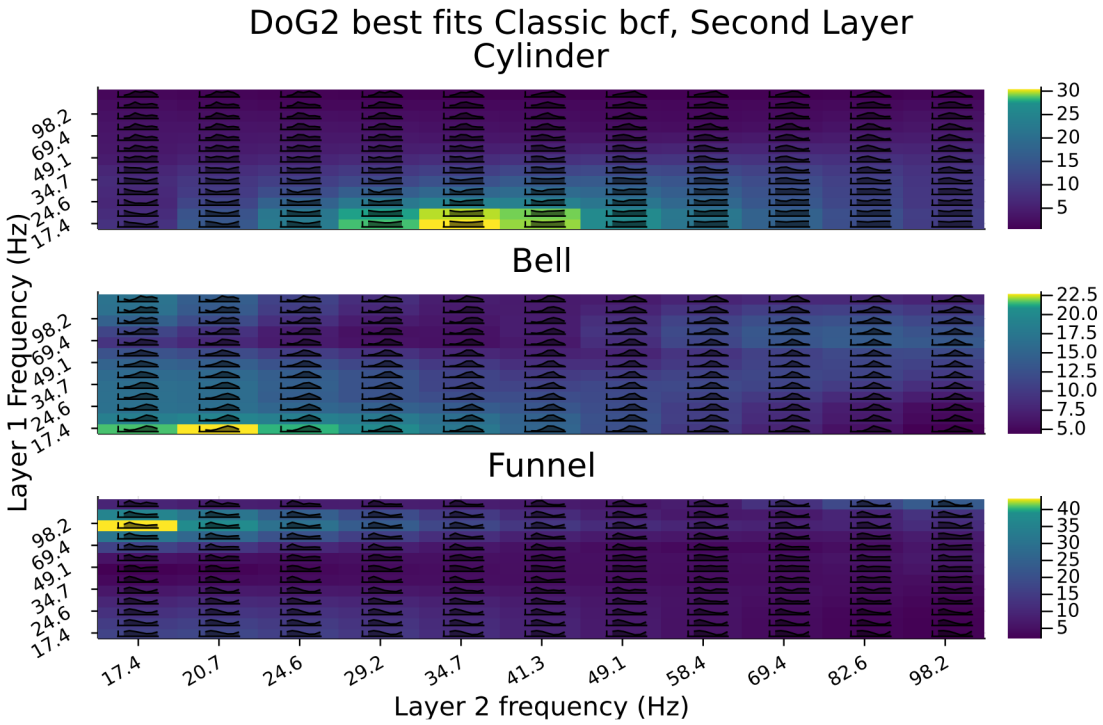
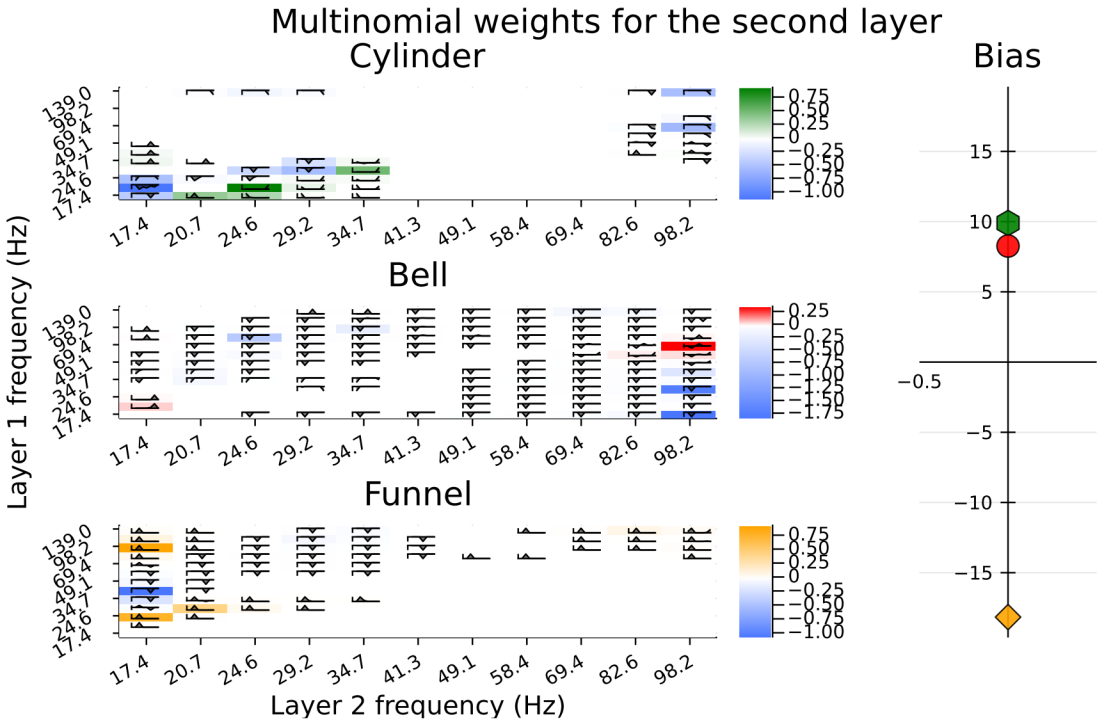


FIGURE 4.29. The scalograms of the Pseudo-inverses shown in Fig. 4.28



(a) Second layer transform of the fit coordinates, color is on a log scale



(b) Previously shown weights from Fig. 4.26 for ease of reference

FIGURE 4.30. Comparing fit with target second layer coefficients for the classic bcf problem using DoG2 wavelets

To see how well these fits match the target output, see Fig. 4.30, which has the second layer ST coefficients of the fit solutions $\boldsymbol{s}\big\lfloor q^2 \big\rceil \left(\boldsymbol{f}^i\right)$ on the left in Fig. 4.30a and the target $\boldsymbol{\beta}^i$s in Fig. 4.30b. The fit has roughly the right frequencies for the cylinder, although the time distribution is wider than at the target paths, and the negative values in $\boldsymbol{\beta}^i$ at $(17.4\text{Hz}, 20.7\text{Hz})$ caused the second layer frequencies in the resulting fit to skew higher than they otherwise had.

The bell coefficients only vaguely resemble the target. In part this is likely because the vast majority of coefficients are negative; even the path with the largest positive coefficient $(98.2\text{Hz}, 69.4\text{Hz})$ has a negative value. Consequently, while this path and its neighborhood are somewhat present, the fit skews towards the far lower frequency path $(17.4\text{Hz}, 20.7\text{Hz})$. The funnel coefficients most closely resemble the target, though there is some displacement caused by the negative frequencies again.

In Fig. 4.31 we have the pseudo-inverses of the $\boldsymbol{\beta}^i$s in Fig. 4.27, the rotate version of the problem, and similarly Fig. 4.32 gives the scalograms of these fit solutions. None of them are particularly clear in the time domain; in the scalogram, the cylinder fit has similar low frequency "bridges" as in Fig. 4.29, but like the bell and funnel has much higher frequency represented throughout. When comparing the second layer transformation of these coordinates in Fig. 4.33a with the targets in Fig. 4.33b, the fit is somewhat closer than it was in the previous layer, at least in part because the targets are much sparser. Roughly speaking, the bell signal is low in the first frequency and high in the second, while the funnel is high in both. That this only somewhat distinguishes the signals explains the confusion matrix in Table 4.1.

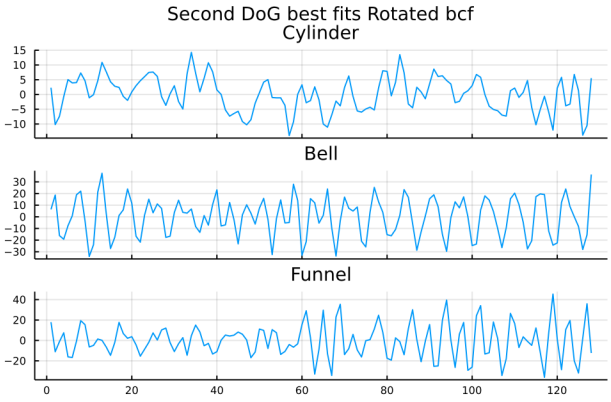


FIGURE 4.31. Rotated bcf problem fit solutions using a second layer DoG transform

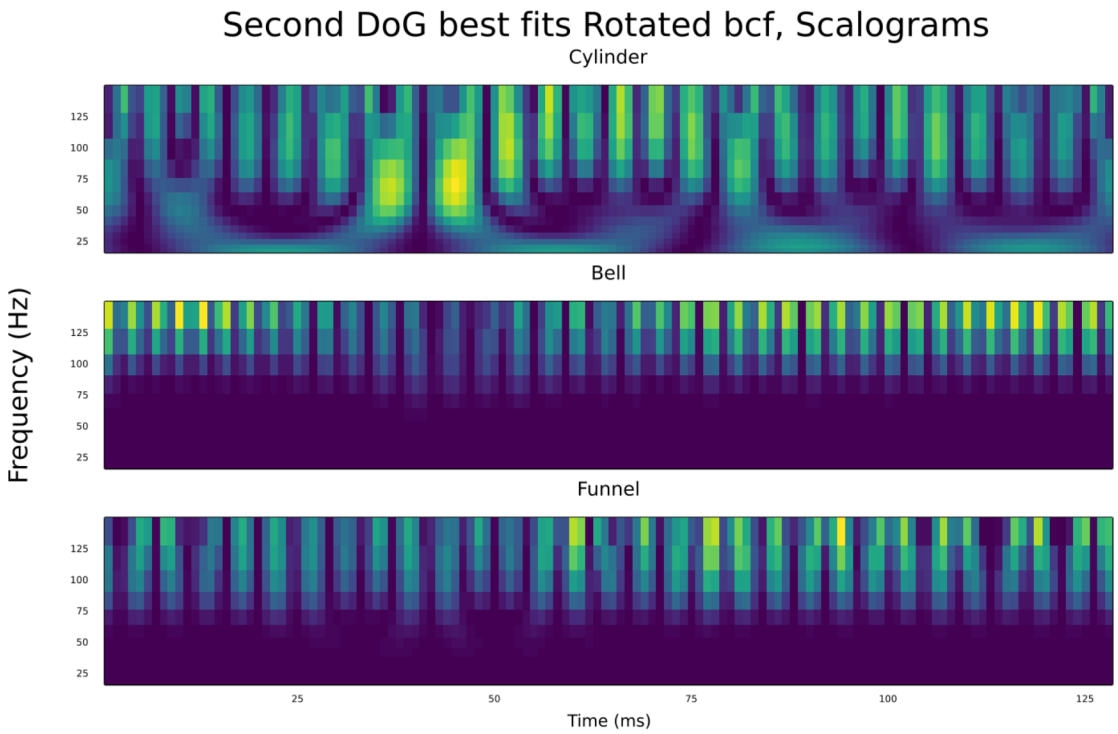
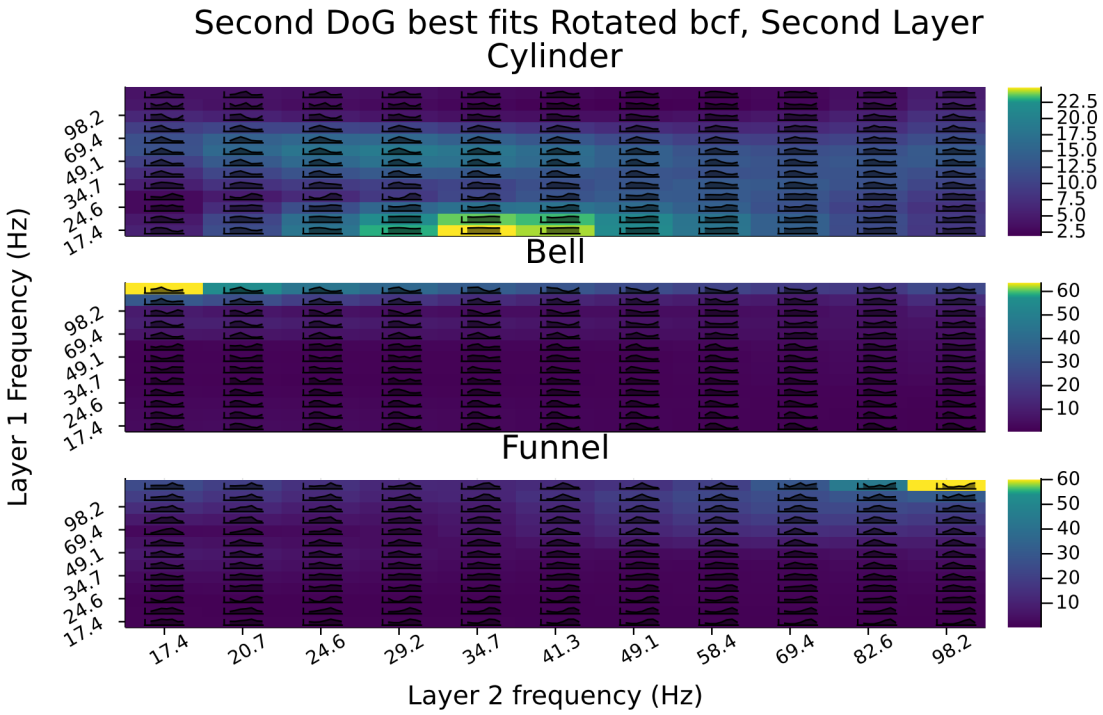
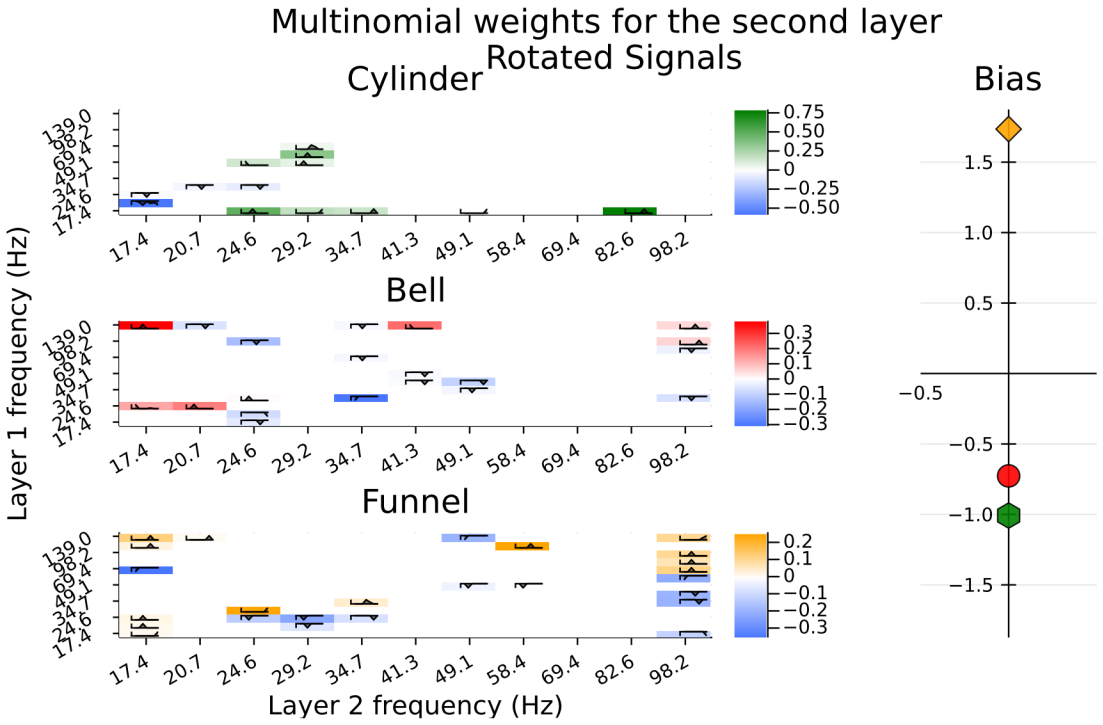


FIGURE 4.32. Scalograms of the signals in Fig. 4.31 (the rotated bcf problem)



(a) Second layer transform of the fit coordinates, color is on a log scale



(b) Previously shown weights from Fig. 4.27 for ease of reference

FIGURE 4.33. Comparing fit with target second layer coefficients for the rotated bcf problem using DoG2 wavelets

CHAPTER 5

Shattering Transform

One of the advantage of wavelets in one dimension is that they provide sparse representations of functions which are continuous except at a finite set of discontinuities. Extending this idea to two and higher dimensions requires some care, since in addition to point discontinuities there are curve discontinuities. Shearlets are a family of frame transforms designed to efficiently capture both point and curve singularities [**47**]. In this chapter, we implement a generalized scattering transform using the shearlet transform, called by the portmanteau *shattering transform*.

In Section 5.1 we detail the shearlet transform and its sparsity guarantees. In Section 5.2 we prove that the shattering transform maintains the sparsity guarantees of shearlets and satisfies the prerequisites for the results of the generalized scattering outlined in Section 3.2 [**77**]. Finally, in Section 5.3 we compare the shattering transform using various nonlinearities to the 2D Morlet wavelet transform as implemented in Kymatio[1] on the MNIST and Fashion MNIST datasets [**4**, **80**]. For the shearlet implementation we use Shearlab.jl[2], written by Héctor Loarca as a Julia implementation of Shearlab3D [**48**].

5.1. Shearlets

Shearlets generate optimally sparse linear representations of $f \in \mathscr{L}(\mathbb{R}^2)$. The Shearlet frame is formed by applying shearing matrices S_k with $k \in \mathbb{Z}$, and anisotropic scaling matrices $A_{j,\alpha}$ and $\widetilde{A}_{j,\alpha}$ to a generator ψ, which needs to have some anisotropic structure. These matrices are

$$S_k := \begin{pmatrix} 1 & ck \\ 0 & 1 \end{pmatrix}, \; A_j := \begin{pmatrix} 2^j & 0 \\ 0 & 2^{j\alpha/2} \end{pmatrix}, \text{ and } \widetilde{A}_j := \begin{pmatrix} 2^{j\alpha/2} & 0 \\ 0 & 2^j \end{pmatrix}.$$

Shearlab does so using a non-separable generator:

$$\widehat{\psi^{(1)}}(\xi_1, \xi_2) := P\left(\tfrac{\xi_1}{2}, \xi_2\right)\widehat{\psi_1 \otimes \phi_1}(\xi_1, \xi_2)$$

[1] https://www.kymat.io/

[2] https://arsenal9971.github.io/Shearlab.jl/

where P is a 2D directional filter[3], ϕ_1 is a 1D scaling (or averaging) function, ψ_1 is a 1D wavelet function, and $\otimes$ is the tensor product, so $\psi_1 \otimes \phi_1(x_1, x_2) = \psi_1(x_1)\phi_1(x_2)$. This function will have directional selectivity along ξ_1. To capture frequency information concentrated along ξ_2 with finitely many elements, it is necessary to introduce a second "cone", generated using $\widehat{\psi^{(2)}}(\xi_1, \xi_2) = \widehat{\psi^{(1)}}(\xi_2, \xi_1)$, which swaps the roles of ξ_1 and ξ_2. If we choose ψ_1 and ϕ_1 appropriately to define $\psi = \psi_1 \otimes \phi_1$, there is a corresponding averaging function ϕ and the resulting shearlet system and discrete shearlet transform $\mathscr{SH}$ are

$$\psi^1_{j,k}(\boldsymbol{l}, \boldsymbol{x}) := 2^{(\alpha+2)j/4}\psi\left(S_k A_j \boldsymbol{x} - [l_1 c_1 \;\; l_2 c_2]^\intercal\right)$$

$$\psi^2_{j,k}(\boldsymbol{l}, \boldsymbol{x}) := 2^{(\alpha+2)j/4}\psi\left(S_k^\intercal \widetilde{A}_j [x_2 \;\; x_1]^\intercal - [l_1 c_2 \;\; l_2 c_1]^\intercal\right)$$

$$\phi(\boldsymbol{l}, \boldsymbol{x}) := \phi\left(\boldsymbol{x} - [l_1 c_1 \;\; l_2 c_2]^\intercal\right)$$

$$k < \left\lceil 2^{j/2} \right\rceil, \quad [l_1 \;\; l_2] \in \mathbb{Z}^2, \quad j \geq 0$$

$$\mathscr{SH} f\left(\boldsymbol{l}^0, i, \boldsymbol{l}, j, k\right) := \left\{ \left\langle f(\cdot), \phi\left(\boldsymbol{l}^0, \cdot\right) \right\rangle, \left\langle f(\cdot), \psi^{(i)}_{j,k}\left(\boldsymbol{l}, \cdot\right) \right\rangle \right\}$$

where i is the cone, j is the scale, k is the shearing, $\boldsymbol{l}^0$ is the translation of the averaging function, $\boldsymbol{l}$ is the translation of the wavelet, and $\boldsymbol{c}$ is the sampling grid size. See the Shearlet textbook edited by Kutyniok and Labate [**47**] or the ShearLab3D paper [**48**] for further details.

5.1.1. Optimal Coefficient Decay rate. One of the nice properties of the shearlet transform is the coefficients are guaranteed to decay at (almost) the optimal rate for linear transforms of cartoon-like functions [**47**, Theorem 5.5]. By cartoon-like functions, we mean functions that are mostly smooth, other than jump discontinuities along smooth curves. More formally, the set of cartoon-like functions are the subset of $\mathscr{L}^2(\mathbb{R})$ such that $f = f_0 + f_1 \chi_B$ where $f_0, f_1 \in \mathscr{C}^2(\mathbb{R}^2)$, and ∂B is defined by some piecewise $\mathscr{C}^2$ curve Γ with bounded curvature.

By optimally sparse, we mean that shearlets can represent any cartoon-like function using (up to a log factor) as few coefficients as possible. For any linear system $G = \{g_i\}_{i=0}^\infty$ which spans the set of cartoon-like functions, we can write $f = \sum_{i=0}^{\infty} c_i g_i$. For this sum to converge, c_i must decay to zero; the rate at which these coefficients decay to zero is not guaranteed however. Donoho showed in 2001 that for any particular linear

[3]Specifically the third order polynomial defined in [**20**, Table II], as discussed in [**48**].

system[4], there is some cartoon-like function such that $(c_0, c_1, \ldots) \in \ell^p$ if and only if $p > \frac{2}{3}$ [**23**]. This means that the fastest that the coefficients c_i for a linear representation system can decay is $O\left(n^{-3/2}\right)$. There is no known linear system which achieves this bound exactly— however, shearlets, like curvelets, come within a log term of doing so [**47**, Chaper 5]. Denoting the shearlet coefficients of f reordered in decreasing order as c_n^*, we have that

$$c_n^* = O\left(n^{-3/2} \log(n)^{3/2}\right).$$

5.2. Extending Theoretical Properties

In this section our goals are guaranteeing that the results proved in other contexts relating to shearlets and the scattering transform apply and can be extended to the shattering transform. The first is showing that the conditions necessary to extend the results for the generalized scattering transform discussed in Section 3.1 apply for the shattering transform in particular. These conditions are quite lax, so the extension is quite straightforward.

Demonstrating that the sparsity just discussed in Section 5.1.1 extends to the case of the shattering transform in Theorem 5.2.1 is a bit more complicated; ultimately we need to add additional restrictions on the nonlinearity, discussed in Lemma 5.2.1. The typical nonlinearities, absolute value and ReLU, satisfy the necessary conditions. Additionally, the transition between continuous input to discrete output in the shearlet transform makes carrying through to the next layer awkward, but is alleviated by treating the input to the next layer and the output at the current layer differently, and is dealt with in Lemma 5.2.3. The averaging requires at least some care to deal with and is discussed in Lemma 5.2.2. After all this, we do arrive at the result one might intuitively expect: so long as the nonlinearity doesn't cause the magnitudes to increase at too rapid of a rate, the shattering coefficients have the same decay rate as the raw shearlet coefficients do.

5.2.1. Lipschitz and translation decay. This section addresses the *weak admissibility condition* given in Eq. (3.4). For the absolute value, the Lipschitz bound for the nonlinearity γ_m is 1, while for ReLU, tanh, and soft-max, $\gamma_m = 2$. Somewhat more difficult is the frame upper bound b_m on the shearlets. A theoretical upper bound could be constructed based off of estimates by Kittipoom et al. [**44**], however, as there are many

[4]To avoid pathological cases, we need the restriction that G can only be reordered up to a distance given by a fixed polynomial. Otherwise such pathological examples as countable dense subsets of $\mathscr{L}^2$ are allowed, which converge faster but are computationally useless.

configurations of the shearlet construction used by Shearlab, it is simpler to calculate it once the frame has been constructed. Since we are dealing with a frame of translates, the frame bounds are maxima of the frames in the Fourier domain:

$$\widehat{\varphi}(\omega) := \left|\widehat{\phi}(\omega)\right|^2 + \sum_{i,j,k} \left|\widehat{\psi}^i_{j,k}(\omega)\right|^2,$$

$$A \leq \widehat{\varphi}(\omega) \leq B.$$

This result can be found in Christensen [**14**, Theorem 7.2.3]. $\widehat{\varphi}$ is the dual frame weight function, which is important for inverting the shearlet transform, and so is already calculated as part of shearlab. For example, for images of size 500×500, with the default settings we have that $A \approx .0625$ and $B \approx 1$. However, smaller images pose a challenge for meeting the frame bound conditions: the MNIST dataset, with size 28×28, has quite large frame bounds $A \approx 4.43$ and $B \approx 22.90$. This may explain some of the experimental results in Section 5.3.

5.2.2. Sparsity Guarantee. The shattering transform is nonlinear, so the decay rate limits of Section 5.1.1 don't apply; however, because each layer is composed of shearlet transforms, as long as the nonlinearity and subsampling don't interfere, we will have that, the shattering coefficients also decay as $O\left(n^{-3/2}\log(n)^{3/2}\right)$.

In considering this, we need to account for the switch between continuous input f and discrete output c_n for the shearlet system Ψ_m. We arrive at a somewhat awkward construction: for the purposes of output, we discretize the internal layer operator u_m from Eq. (3.2) before averaging, while for passing to the next layer, we consider it continuous in space but not shearing or scale indices. In both cases the nonlinearity ρ_m introduces some additional constraints. As a continuous operator, u_m produces cartoon-like functions from cartoon-like functions, while as a discrete operator, we need to show that neither ρ_m nor averaging destroys the decay rate.

First, we deal with the nonlinearity for the discrete u_m. To maintain (or improve) the rate of decay, we need that for any two points s and t, for ρ_m to bring whichever one is closer to zero even closer. A motivating example of what we want to avoid is $\rho_m(t) = t^{1/2}$; applied to the sequence $\frac{1}{n^2}$, this nonlinearity ends up giving a decay rate of $\frac{1}{n}$.

LEMMA 5.2.1. *If ρ_m satisfies*

- *$\rho_m(t)$ is convex,*
- *if $t \le s$, with t and s having the same sign, then $\rho_m(t) \le \rho_m(s)$,*[5]
- *$\rho_m(0) = 0$*

and the sequence c_n decreases at rate $c_n \lesssim f(n)$, then the sequence $\rho_m(c_n)$ also decreases at least as fast: $\rho_m(c_n) \lesssim f(n)$. Here $a \lesssim b$ means $a = O(b)$.

PROOF. W.l.o.g., consider those c_n where we have exactly $c_n \le f(n)$. Any multiplicative constants needed to make this exact can be absorbed into f, while the finitely many possible exceptions can be ignored. Define $C := \frac{\rho_m\left(f(0)\right)}{f(0)}$, and choose $t := \frac{f(n)}{f(0)}$. Then $t \in [0, 1]$, since f is a decreasing function, and the expression $(1-t)0 + tf(0)$ simplifies to $f(n)$. We can use the convexity of $\rho_m(t)$ to say that

$$\begin{aligned} \rho_m\big(f(n)\big) &= \rho_m\Big((1-t)0 + tf(0)\Big) \le (1-t)\,\rho_m(0) + t\,\rho_m\big(f(0)\big) \\ &\le 0 + \frac{f(n)}{f(0)}\,\rho_m\big(f(0)\big) = Cf(n) \end{aligned} \tag{5.1}$$

where we have also used the third assumption that $\rho_m(0) = 0$. Then

$$\rho_m(c_n) \le \rho_m(f(n)) \le Cf(n).$$

The first step we have because of the second assumption, that $c_n \le Cf(n)$, and the second step is just Eq. (5.1). □

Some examples of functions that work are x^3, x, or ReLU; examples that will end up being relevant later that aren't allowed are tanh or other sigmoidal functions, as they all have concave portions (and must to reach a bounded value from below). It is worth noting that such sigmoidal functions were common in the neural network literature, but have since fallen out of common use.

Next, we deal with the somewhat more awkward case of the averaging. We show that so long as ϕ has compact support, every large coefficient can only change the magnitude of coefficients in its immediate neighborhood, which incurs at most a constant penalty $2L$.

[5] this isn't simply increasing, unfortunately. It is increasing on positive numbers t if $\rho_m(t)$ is positive, or decreasing if it's negative, while for $t < 0$ it is decreasing if $\rho_m(t)$ is positive, or increasing if it's negative.

LEMMA 5.2.2 (Decay after averaging). *Suppose we have an array $\boldsymbol{g}$ defined on $\mathbb{Z}^d$, and a rearrangement of the entries of $\boldsymbol{g}$ given by $\pi : \mathbb{N} \to \mathbb{Z}^d$ which orders $\boldsymbol{g}\big|\pi(n)\big]$ as a monotonically decreasing sequence, and further that they decay at a rate $\boldsymbol{g}\big|\pi(n)\big] \lesssim n^{(-3/2)}(\log n)^{3/2}$. Suppose we have some averaging array $\boldsymbol{\phi}$ which is supported on some finite box $|-L, L]^d$. Then $\boldsymbol{\phi} \star \boldsymbol{g}$ has a different rearrangement π^* so that it too also decays as $\boldsymbol{\phi} \star \boldsymbol{g}\big|\pi^*(n)\big] \lesssim n^{-3/2}\big(\log n\big)^{3/2}$.*

PROOF. First, some simplifying assumptions and definitions. W.l.o.g., assume that $|\boldsymbol{\phi}|_1 = 1$; if it isn't, simply consider the array $\frac{\boldsymbol{g}}{|\boldsymbol{\phi}|_1}$, which has the same coefficient order. Let the *neighborhood* $N(\boldsymbol{i})$ of $\boldsymbol{i} \in \mathbb{Z}^d$ be the set of tuples $N(\boldsymbol{i}) = \big\{\boldsymbol{k} \quad |\boldsymbol{i} - \boldsymbol{k}|_\infty \leq L\big\}$.

The core idea for constructing π^* from π is to enumerate every point in the neighborhood of $\pi(1)$, then every point in the neighborhood of $\pi(2)$ that wasn't already listed in the neighborhood of $\pi(1)$, and so on. This contains the interference from any coefficients increased in the neighborhood of $\pi(1)$ early in the list. To that end, let n_k be the number of points in the union $\bigcup_{j=1}^{k} N\big(\pi(j)\big)$ (the number of points in the neighborhoods of all the k largest points). Then π^* from 1 to n_2 is $N\big(\pi(1)\big)$, while π^* from $n_2 + 1$ to n_3 is $N\big(\pi(2)\big) \setminus N\big(\pi(1)\big)$. In general, π^* from $n_k + 1$ to n_{k+1} enumerates the set $N\big(\pi(k)\big) \setminus \bigcup_{j=1}^{k-1} N\big(\pi(j)\big)$ in order from largest to smallest element. By this construction, the number of points added $n_{k+1} - n_k$ is at most $(2L)^d$, since each neighborhood $N\big(\pi(k)\big)$ has this many points, and we may remove some. Note that in this construction, $\pi(k)$ may end up being in the neighborhood of an earlier point such as $\pi(k-5)$, and not in its own neighborhood This won't actually be a problem, as we simply want to guarantee that the elements in the neighborhood are smaller than that at $\pi(k)$.

To see that $\boldsymbol{g}_{\pi(k)}$ is larger than every point in the set $\{\boldsymbol{g}_{\boldsymbol{w}} \quad \boldsymbol{w} \in N\big(\pi^*(j)\big)$ for $j \in |n_k + 1, n_{k+1}]\}$ (the neighborhoods of every point in the range $|n_k + 1, n_{k+1}]$), assume that there's some point $\boldsymbol{i}$ so that $\boldsymbol{g}\big|\boldsymbol{i}\big] > \boldsymbol{g}\big|\pi(k)\big]$. This would mean that $\boldsymbol{i}$ must come before $\pi(k)$ in the list generated by the original π say as the jth largest coefficient, so $\boldsymbol{i} = \pi(j)$. But since every point which has $\boldsymbol{i}$ in its neighborhood is also in $\boldsymbol{i}$'s own neighborhood, $\pi(k)$ must have already been included as elements of $N\big(\pi(j)\big)$ somewhere between $n_j + 1$ and n_{j+1}. So in a proof by contradiction, $\boldsymbol{g}|\boldsymbol{w}] < \boldsymbol{g}|\pi(k)]$ for every $\boldsymbol{w} \in \{\boldsymbol{w} \quad \boldsymbol{w} \in N\big(\pi^*(i)\big)$ for $i \in |n_k + 1, n_{k+1}]\}$.

For convenience define $\boldsymbol{h}|\boldsymbol{i}] = (\boldsymbol{\phi} \star \boldsymbol{g})|\boldsymbol{i}]$. To get a bound on $\boldsymbol{h}|\boldsymbol{i}]$ using $\boldsymbol{g}$, we use a discrete version of Young's inequality with $p = q = 1$, $r = \infty$ and $f_{\boldsymbol{j}} = \delta_{\boldsymbol{i},\boldsymbol{j}}$: [**51**, Section 4.2]

$$h|\boldsymbol{i}] \;\leq \sum_{\boldsymbol{l} \in [-L,L]^d} \delta_{\boldsymbol{i},\boldsymbol{l}} \boldsymbol{\phi} \star \boldsymbol{g}\big|\boldsymbol{l}\big] \;\leq |\,\delta_{\boldsymbol{i},\boldsymbol{j}}\,|_1 |\,\boldsymbol{\phi}\,|_1 |\,\boldsymbol{g}\chi_{N(\boldsymbol{i})}\,|_\infty \leq |\,\boldsymbol{g}\chi_{N(\boldsymbol{i})}\,|_\infty$$

where $\chi_{N(\boldsymbol{i})}$ is the (discrete) characteristic function of the neighborhood of $\boldsymbol{i}$. If $\boldsymbol{i} = \pi^*(k)$ for k between $n_j + 1$ and n_{j+1}, then by the construction of π^* and n_j above, the largest magnitude coefficient in $N\big(\pi^*(k)\big)$ is $g_{\pi(j)}$. Thus $h_{\pi^*(k)} \leq g_{\pi(j)}$.

Because each neighborhood has at most $(2L)^d$ elements, asymptotically we can relate j and k through $(2L)^d j \leq k \leq (2L)^d (j+1)$, so we have that $k \approx (2L)^d j$, or $j \approx (2L)^{-d} k$. Using this and the result from the previous paragraph,

$$h_{\pi^*(k)} \leq g_{\pi(j)} \leq C\, j^{-3/2} (\log j)^{3/2} = C \frac{\Big(\log(k(2L)^{-d})\Big)^{3/2}}{\big(k(2L)^{-d}\big)^{3/2}}$$

$$= C(2L)^{2d/3} \frac{\Big(\log(k) - d\log 2L\Big)^{3/2}}{\big(k\big)^{3/2}} \lesssim k^{(-3/2)} (\log k)^{3/2}$$

□

Note that while the proof is for the case of a single array $\boldsymbol{g}$, since $\boldsymbol{\phi}$ only interacts with a single $\boldsymbol{g}$ at a time, it can be extended to the case of a collection of arrays $\{\boldsymbol{g}^\lambda\}_{\lambda \in \Lambda}$ at the cost of more notation overhead. It is worth noting that this rearrangement is most likely not monotonically decreasing, so in practice the additional $(2L)^d$ factor may not be necessary if $\boldsymbol{\phi}$ has the shape of an actual averaging function like e.g., B-splines, Welch, Hann, etc. We could also extend this result to averaging functions which decay sufficiently quickly, rather than have finite support, but do not do so as a matter of time, as there are shearlet constructions where the averaging wavelet does have compact support.

Finally, we deal with the internal case of continuous u_m. It will turn out that the $u_m(f)$ is actually a slight generalization of a cartoon-like function that involves the potential intersection of several boundary curves.

LEMMA 5.2.3. *If ρ_m is piecewise smooth and f is a cartoon-like function, then the shearlet transform of $u_m(f)$ decays as $O(n^{-3/2}(\log(n))^{3/2})$.*

PROOF. Convolution gives the output the smoothness of the smoother of the two functions; consequently, $\psi^m_{\lambda^m_i} \star f$ for f being cartoon-like results in a smooth function, since the $\psi^m_{\lambda^m_i}$ for shearlets are smooth. The only major difficulty is what happens with ρ_m. Because it is piecewise smooth, the only difficulties arise on the level sets of $h(\boldsymbol{x}) = \rho_m(\psi^m_{\lambda^m_i} \star f)(\boldsymbol{x})$ for the points of discontinuity of ρ_m (suppose one is at t_0: the same ideas work for any additional level-sets). By the inverse function theorem [**65**, Theorem 9.24], we have that the level set $h(\boldsymbol{x}) = t_0$ is at least $\mathscr{C}^2$ away from points where $\nabla h \neq \mathbf{0}$, and further that at those points that it is the intersection of finitely many $\mathscr{C}^2$ curves. Thankfully, the case of piecewise $\mathscr{C}^2$ curves is covered in [**47**, Section 5.1.7]. The proof that not an unreasonable number of coefficients become too large at the point of discontinuity in the curve applies equally well when there are multiple curves intersecting instead of a single curve having a cusp. For brevity, we don't recapitulate the proof here. □

Finally, putting together Lemma 5.2.1, Lemma 5.2.2, and Lemma 5.2.3, we have the actual result of this section:

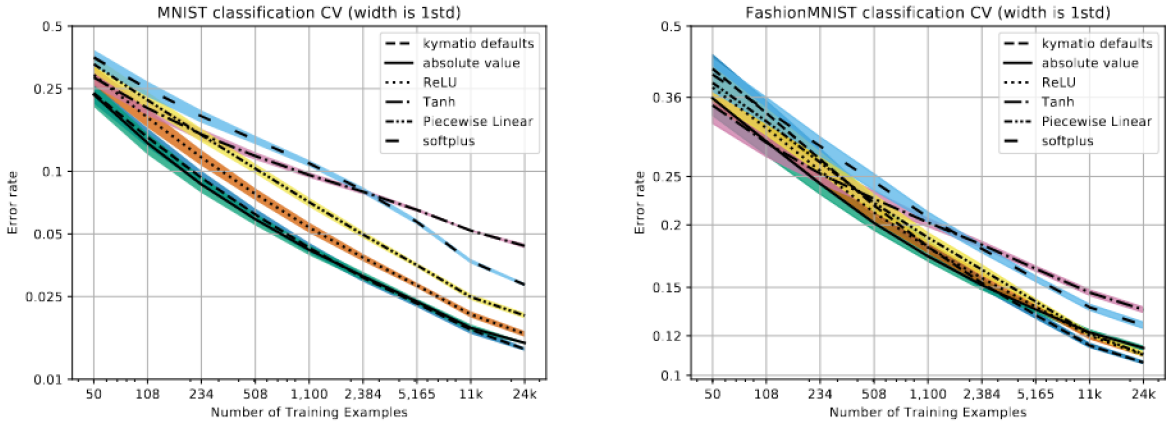
THEOREM 5.2.1 (Scattering Sparsity). *If the nonlinearities* $\{\rho_m\}_{m\in\mathbb{N}}$ *satisfy*

- ρ_m *is piecewise smooth*
- $\rho_m(t)$ *is convex,*
- *for* $t \leq s$, $\rho_m(t) \leq \rho_m(t)$,
- $\rho_m(0) = 0$

and the averaging functions are compactly supported, then for cartoon functions f*, the shattering transform output* $\Phi\big|f\big]$ *decays at the rate* $O(n^{-3/2}(\log(n))^{3/2})$.

5.3. Experiments

The primary classification results can be found in Fig. 5.1. We compare the efficacy of various nonlinearities in transform along with the 2D scattering transform as implemented in Kymatio [**4**] at classification on the MNIST dataset and the Fashion MNIST datasets with an increasing amounts of data [**80**]. For classification we used a linear support vector machine. The final data point in each is trained on 525k examples and the reported test is on the validation set. Both graphs are on a log-log plot, with the colored regions indicating the standard deviation of the cross validation. The linear dependence between accuracy and amount of training data fits the rates found for neural networks [**36**]. The nonlinearities used are in Table 5.1.



(a) MNIST dataset.

(b) Fashion MNIST dataset

FIGURE 5.1. Relation between the error rate using various nonlinearities and the number of training examples. Both the x- and y-axes are logarithmic. The colored bands surrounding each curve represent the cross validation variance. As one may expect, the variance is higher with fewer examples.

The shattering transform with the absolute value is on par with the kymatio transform for the MNIST classifier; somewhat surprisingly, all other nonlinearities lead to a significantly worse classification rate. On the Fashion MNIST dataset, all of the shattering techniques are significantly more effective when given only 50 examples, but are surpassed by 5000 examples.

First, we may have expected the sparsity of the transform to lead to better classification performance as more features would be concentrated in fewer coefficients. However, as was mentioned in Section 5.2.1, both MNIST and FashionMNIST are only 28×28 pixels. This led to very large upper and lower frame bounds, and in general a poorly constructed version of the shearlet transform. Effectively, at this scale, there is very little difference in construction between the shearlets and the 2D Morlet wavelets. On a larger dataset, we might see the benefit provided by the sparsity of the shearlets more concretely.

Nonlinearity Name	Function
absolute Value	$\rho_m(t) = \quad t$
ReLU	$\rho_m(t) = \max\{t, 0\}$
Tanh	$\rho_m(t) = \frac{e^t - e^{-t}}{e^t - e^{-t}}$
Piecewise Linear	$\rho_m(t) = \begin{cases} t & : t \geq 0 \\ \frac{t}{2} & : t < 0 \end{cases}$
Softplus	$\rho_m(t) = \log(1 + e^t)$

TABLE 5.1. The nonlinearities used in the Shattering transform in this section

It is worth considering why we have such different results for different nonlinearities. We postulate that it is primarily due to bounded nature of the tanh, ReLU, and soft-plus. In the case of ReLU, the value of the function in all negative regions is erased completely. The absolute value only discards the sign information. While tanh and soft-plus are theoretically invertible, and so should lose nothing, in practice numerical precision rounds any sufficiently negative output to 0 (in practice, this seems to be around -16 for soft-plus and ± 8 for tanh). So practically, tanh discards both positive and negative values outside of $[-8, 8]$, explaining its somewhat abysmal asymptotic performance.

Why then does ReLU perform well in the case of convolutional neural networks? A possible reason for this is that CNNs use affine, not linear transforms, meaning that the input to the ReLU can be shifted so that informative regions are on the positive side of the zero hyperplane. In contrast, unmodified scattering transforms aren't affine, and so benefit more from nonlinearities that include information from the negative regions.

CHAPTER 6

Sonar Scattering

6.1. Problem Overview

In this chapter, we begin the second half of the study promised in the title: sonar signal classification. The bulk of the work was published previously in [**68**]. In the wake of various conflicts, notably WWII, large quantities of unexploded ordinance have made their way to the seabed [**45**]. In shallow waters, these explosives pose a risk to any ships traveling through the region. However, the area to be surveyed is vast, so a relatively novel strategy involves deploying unmanned underwater vehicles (UUV) equipped with synthetic aperture sonar (SAS) to detect unexploded ordinance, such as the top left 3 items in Fig. 6.1.

FIGURE 6.1. Example targets on shore and in the target environment [**43**]

The standard approach to analyzing such SAS data is to use beamforming techniques to create an image that can be interpreted by the human eye [**82**, Section 6.4]. However, in the case of UXO detection, the size of the objects is both too small and noisy (as these objects are often embedded in mud/sand/etc.) for the

beamformed images to be interpretable. Even if that were solved, there is also a problem of scale, as training is required to interpret beamformed images.

Thus begins the search to find an automated method of detecting UXOs directly from the raw signals. This dissertation is not the first effort on the part of Professor Saito and his group towards detecting UXOs directly from sonar data [**52**,**58**]. However, sonar data is naturally well suited to the benefits of the scattering transform. When moving directly towards or away from an object, sonar data is translation invariant, and because of the underlying generation mechanism (see Section 6.2 and Section 6.3), we expect that as we move around an object the variations that occur can be characterized as the space/frequency deformation of Theorem 3.2.1. So long as morphing via $f\big(t-\tau(t)\big)$ from one class to another requires a τ with large derivative, then the classes will be well separated. Accordingly, we use a linear classifier on the output of the scattering transform. Additionally, because the scattering transform concentrates energy at coarser scales and wavelets in general encourage sparsity for smooth signals with singularities, we use a sparse version of logistic regression as our linear classifier, using LASSO [**35**]. Specifically, we use a Julia wrapper around GLMNET [**31**].

As a baseline classifier, we use LASSO on the absolute value of the Fourier transform (AVFT) of the signal. One strong advantage of the absolute value of the Fourier transform is its complete invariance to translations, and is the prototype of every linear filter which is invariant to translations, as discussed back in Section 2.1. In addition to this invariant, the close ties between frequency and the speed of sound suggest that it should be sensitive to changes in the material. This will be examined in more depth in Section 6.3.

For this problem, we have both real and synthetic examples. The real examples, collected in the BAYEX14 dataset [**43**], consist of 14 partially buried objects at various distances and rotations in a shallow mud layer on top of a sand ocean bed at a depth of ~ 8m. To model a UUV, they then set up a sensor/emitter on a rail like the leftmost figure of Fig. 6.2, and pinged the field of objects at various rotations. So for each rotation of the object relative to the rail, there is a 2D wavefield.

Additionally, we can generate synthetic waveforms using a fast solver for the Helmholtz equation in two regions with differing speed of sound, provided by Ian Sammis and James Bremer [**8**, **9**]. We use this to examine more closely the dependence of both classification and the output of the scattering transform on both material properties and shape variations. The setup for the synthetic case is made to replicate the real case, and is in Fig. 6.2. Further discussion of this process is in Section 6.2.

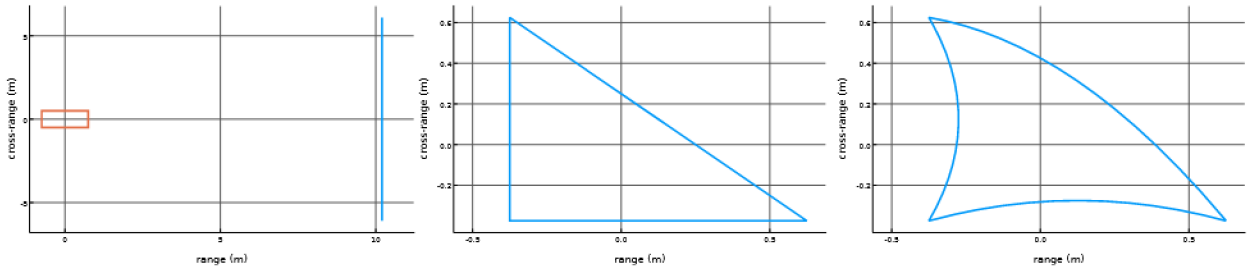


FIGURE 6.2. The three shapes of Ω used in the synthetic setting. The rectangle on the left includes the observation rail without rotation. The range is approximately 10m, while the observation rail itself is 12m in total. The triangle has side lengths of 1m, while the shark fin is deformed from the same triangle. The rectangle has side lengths of 1m and 1.5m.

Section 6.4.1 gives the results of applying the ST and AVFT to the dataset to performing binary classification on shape and material in the synthetic case, while Section 6.4.2 gives the results on a similar binary UXO/non-UXO classification.

6.2. Signal Synthesis via the Helmholtz Equation

The synthetic examples come from considering the 2D Helmholtz equation in regions Ω and Ω^c with differing speed of sound, coupled by enforcing $\mathscr{C}^2$ equality with the source wave g evaluated on the boundary [**8**]:

$$\begin{aligned} \Delta u + (k_1)^2 u &= 0 \quad \text{in } \Omega \\ \Delta v + (k_2)^2 v &= 0 \quad \text{in } \Omega^c \\ u - v &= g \quad \text{on } \partial\Omega \\ \partial_\nu u - \partial_\nu v &= \partial_\nu g \quad \text{on } \partial\Omega \\ \sqrt{\boldsymbol{x}}\left(\partial_{\boldsymbol{x}} - \mathrm{i}k_2\right) v(\boldsymbol{x}) &\to 0 \text{ as } \boldsymbol{x} \to \infty, \end{aligned}$$

The solution u gives the response to a sinusoidal signal with frequency ω inside on an object Ω with wave speed $k_1 = \omega / c_1$, where c_1 is the speed of sound in the material, while v gives the solution outside the object. c_i ranges from 343m/s in air, to 1503m/s in water, to 5100m/s in aluminum. To find u and v, we use a fast solver created by Ian Sammis and James Bremer to synthesize a set of examples [**8**], where we can more explicitly test the dependence of the scattering transform on the material properties (corresponding to changing the speed of sound) and geometry. The initial method used to generate these examples was written by Vincent Bodin, a former summer intern supervised by Professor Saito. It is worth noting that this is idealized in several ways:

- the model we use is 2D, rather than 3D
- the material is modeled as a fluid with only one layer, instead of a solid with multiple different components
- there is no representation of the ocean floor itself.

If the signal emitted by the UUV were a pure sinusoid, then the response at location $\boldsymbol{x}$ and time t would simply be $\mathfrak{Re}\left(v(\boldsymbol{x})\mathrm{e}^{\mathrm{i}\omega t}\right)$. However, they are instead short pings as given in Fig. 6.3. One can approximate the response to such multi-frequency signals (e.g., Gabor functions or chirps) by integrating across frequencies. Let the input signal be fixed as $s(t) \in \mathscr{L}^1(\mathbb{R}) \bigcap \mathscr{L}^2(\mathbb{R})$.The ideal reconstruction of the response $f(t, \boldsymbol{x})$ to s, for a transmitter and receiver located at $\boldsymbol{x} \in \Omega^c$ is

$$f(t, \boldsymbol{x}) = \int_{-\infty}^{\infty} \hat{s}(\omega)\, v(\omega, \boldsymbol{x}) \mathrm{e}^{-\mathrm{i}\omega\left(t - P(\omega, \boldsymbol{x})\right)}\, \mathrm{d}\omega \tag{6.1}$$

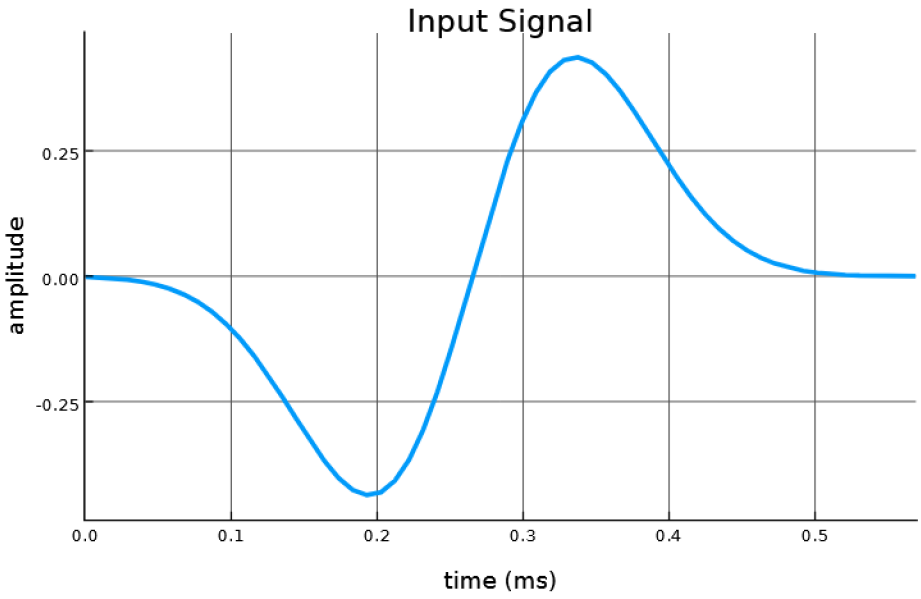


FIGURE 6.3. The non-zero portion of the source signal $s(t)$

where v is the solution to the Helmholtz equation in Ω^c and $\hat{s}$ denotes the Fourier transform of s. To actually simulate this, it is necessary to zero pad so that the periodic version generated via a FFT is approximately the same as the finite support version.

To define the observation rail, first we define an unrotated observation point, $\boldsymbol{x}_0(r) = (x, r)$ for $r \in [-y, y]$ Then, a rotation of the object by an angle $-\theta$ is equivalent to rotating the rail by θ, so define $\boldsymbol{x}_\theta(r) = R_\theta \boldsymbol{x}_0(r)$ where R_θ is the appropriate rotation matrix. In this setup, $x = 10$m, and $y = 6$m.

6.3. Geometric Properties

Ideally the invariants discussed above would apply to transformations in the *object* domain rather than to signals. But translations of the object (or equivalently, the observation rail), have a more complicated effect on the signal than simply translating the observation: even translating away from the object will cause a decay in signal amplitude in addition to delaying the response. The changes in the object domain we seek to understand are changes in object material, translations and rotations of the object/rail, and changes in geometry. A classifier for this problem should be invariant to translation and rotation, but sensitive to the geometry of the object and the material. In this section we examine what can be said about the correspondence

between object variations and scattering coefficient variations. We begin with the what happens as we vary the speed of sound.

6.3.1. Effects of the speed of sound. To determine the behavior of a fixed location $\boldsymbol{x}_\theta(r)$ as we change the speed of sound, we use common acoustic properties such as reflection coefficients and Snell's law [**68**, Section 3.3]. The crudest possible assumption that still gives meaningful results is that internal angles are irrelevant, and only refraction, reflection, and average internal distance matters. Going from Ω^c with speed of sound c_2 to Ω with speed of sound c_1, the reflection coefficient is given by $V_{2,1} = \frac{Z_2 - Z_1}{Z_1 + Z_2}$, while the refraction coefficient is $W_{2,1} = 1 - V_{2,1} = \frac{2Z_1}{Z_1 + Z_2}$, where the impedance is $Z_i = \rho_i c_i$, with ρ_i the density of the material. The distance from the center to a given point $x_\theta(r)$ on the line is just given by the Pythagorean theorem, $\sqrt{x^2 + r^2}$. This means the initial peak occurs at $\frac{1}{c_2}\sqrt{x^2 + r^2}$. If the input peak has

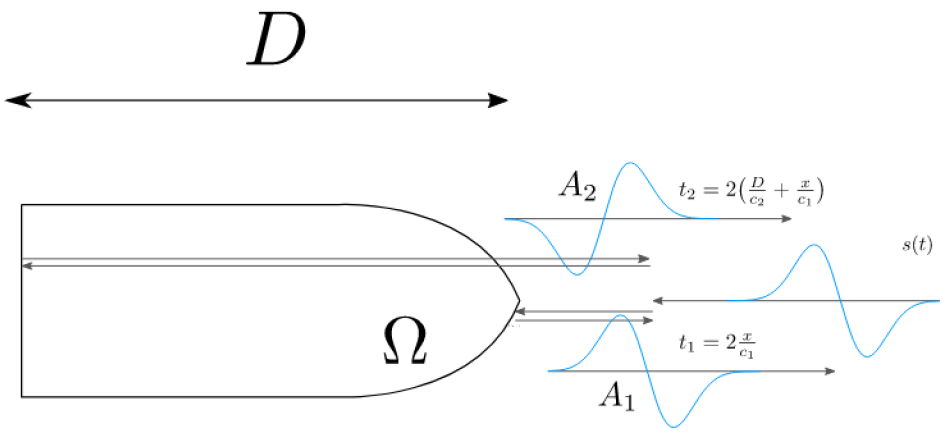


FIGURE 6.4. A motivating diagram for determining the prototypical response as a function of speed. t_1 and t_2 give the time until the first and second peaks arrive back at the receiver. Note that the direction of time is flipped between the signal going towards the object and towards the receiver.

magnitude A_0, then the first return peak should be approximately $A_1 = V_{2,1}\frac{A_0}{x^2+r^2} = \frac{(Z_2 - Z_1)}{(x^2+r^2)(Z_2+Z_1)}A_0$; since $Z_2 > Z_1$ for most relevant examples, this is positive. Setting $\mathrm{diam}(\Omega) = D$, the next peak can be approximated as $A_2 = W_{2,1}V_{1,2}W_{1,2}\frac{A_0}{(x^2+r^2)d} = \frac{4Z_2Z_1(Z_1 - Z_2)}{(x^2+r^2)D(Z_1+Z_2)^3}A_0$, and the sign of this second peak will flip. The third peak is $A_3 = W_{2,1}V_{1,2}^2W_{1,2}\frac{A_0}{(x^2+r^2)D^2}$ and will flip sign again. Similarly $A_n = W_{2,1}V_{1,2}^nW_{1,2}\frac{A_0}{(x^2+r^2)D^n}$.

From this, we should expect that the decay from the first peak to the second peak, $\frac{4Z_1Z_2}{D(Z_1+Z_2)^2}$, is larger than that between any further consecutive peaks, $\frac{2}{(Z_1+Z_2)D}\frac{Z_1 - Z_2}{}$. One can see the sign flip clearly in Fig. 6.5 (take care that the input signal (Fig. 6.3) is a positive spike followed by a negative spike, so the second peak begins at $\sim$ .75ms for speed of sound 4000m/s).

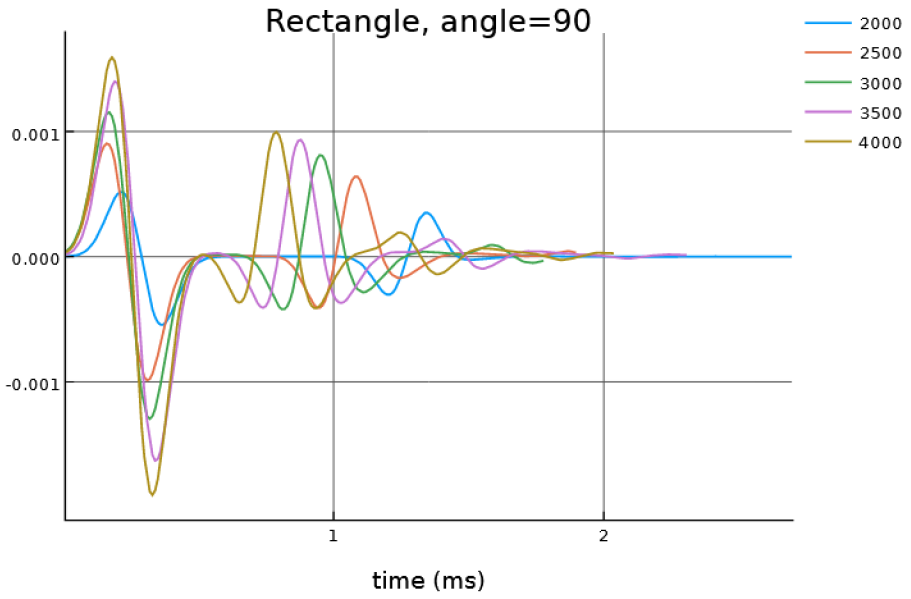


FIGURE 6.5. Non-zero portion of the rectangle signal for varying speed of sound, when facing the long edge.

For classification purposes, for a fixed angle and position on the rail, the reflection of every peak but the first depends on both D and the speed of sound c_1. Further, the time between peaks should be $2D/c_1$, so the scale of the solutions should provide a strong indicator of the speed of sound in the material. We will indeed see that the absolute value of a Fourier transform (AVFT), which has access to scale information (as its inverse, frequency), is reasonably effective at separating shapes based the material speed of sound.

6.3.2. Effects of rotation. The previous section almost completely ignored the internal geometry of Ω. For a rectangle, when facing the longer side, this approximation is reasonable. For other shapes or angles, however, Snell's law will both change the magnitude and the angle of the signal, and the problem of ray tracing in a region Ω becomes non-trivial in its own right, requiring averaging over all paths.

To avoid this, instead of trying to directly construct properties of the observations, we can derive how they will change under rotation and translation. We can use the far field approximation to do so [**28**, Chapter 4]. Since $\text{diam}(\Omega) \approx 1$ and the center frequency of s is $\omega_0 = 2500$Hz, we have that the observation rail at a range of 10m is much farther than $\frac{c_1}{\omega_0} \approx .6$m, the condition for far-field. Here we take the solution as approximately separable, so $v(\omega, \boldsymbol{x})e^{i\omega P(\omega,\boldsymbol{x})} \approx R(\omega, r)\Theta(\omega, \theta)$. The solution to the radial Helmholtz equation is a Bessel function of the first kind, and so to zeroth order is approximately $R(\omega, r) = J_0(k_2 r) \approx \frac{1}{\sqrt{k_2 r}} e^{ik_2 r - \pi/4}$.

Using this, we can examine the effect of rotation of the object (or the rail about the object). Suppose we know the solution at angles θ_{-1} and θ_1, and we want to determine the solution at an angle θ_0 between these. Every point in $\boldsymbol{x}_0(q)$ will have the same angle as a point on either $\boldsymbol{x}_1(r)$ or $\boldsymbol{x}_{-1}(p)$ (or possibly both) if θ_{-1} and θ_1 are close enough that the paths cross, such as the case in Fig. 6.6. This happens when $\tan\left(\frac{1}{2}(\theta_1 - \theta_{-1})\right) < y/x$, or in the synthetic dataset, $\theta_1 - \theta_{-1} \leq \pi/6$). Some geometry gives that the point $\boldsymbol{x}_0(q)$ has the same angle as $\boldsymbol{x}_1(r)$ if

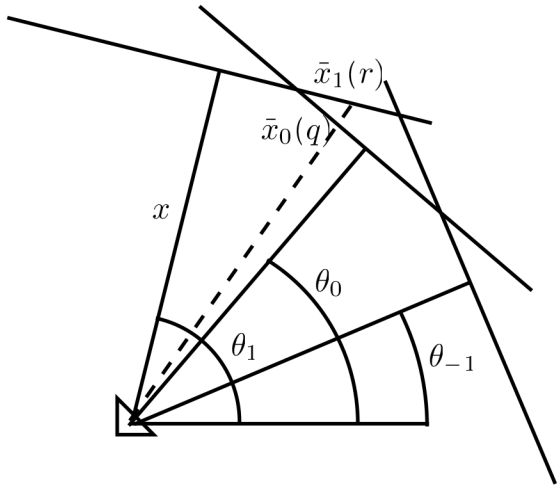


FIGURE 6.6. Composing a rail observation along $\boldsymbol{x}_0(q)$ from that of $\boldsymbol{x}_{-1}(p)$ and $\boldsymbol{x}_1(r)$.

$$T(q) = -x\tan\left(\theta_1 - \theta_0 - \arctan(q/x)\right) \tag{6.2}$$

for $q > 0$; similar reasoning works for $q < 0$ and $\boldsymbol{x}_{-1}$ with flipped signs. The distance changes from $\sqrt{x^2+q^2}$ to $\sqrt{x^2+T(q)^2}$, while the angle remains fixed, so plugging the zeroth order approximation to the Bessel function $J_0(k_2 r)$ above into Eq. (6.1),

$$f(t, \boldsymbol{x}_0(q)) \approx \int_{-\infty}^{\infty} \widehat{s}(\omega) \frac{J_0\left(k_2\sqrt{q^2+x^2}\right)}{J_0\left(k_2\sqrt{T(q)^2+x^2}\right)} \nu\left(\omega, \boldsymbol{x}_1\left(T(q)\right)\right) \mathrm{e}^{-\mathrm{i}\omega(t - P(\omega, \boldsymbol{x}_1(T(q))))}\, \mathrm{d}\omega \tag{6.3}$$

$$f(t, \boldsymbol{x}_0(q)) \approx \frac{\sqrt[4]{T(q)^2+x^2}}{\sqrt[4]{q^2+x^2}} \int_{-\infty}^{\infty} \widehat{s}(\omega) \nu\left(\omega, \boldsymbol{x}_1\left(T(q)\right)\right) \mathrm{e}^{-\mathrm{i}\omega\left(t + h(q) - P(\omega, \boldsymbol{x}_1(T(q)))\right)}\, \mathrm{d}\omega$$

where $h(q) = -\frac{1}{c_2}(\sqrt{q^2+x^2} - \sqrt{T(q)^2+x^2})$. So there are two effects on f in the zeroth order approximation. The first is a phase shift by $h(q)$, under which both the AVFT and the ST are invariant. The second is a small amplitude modulation (for $\theta_1 - \theta_0 = \pi/6$, this ranges from .94 to 1.077). However, since the error in this approximation is $\left(\frac{1}{kx}\right)^{1/2} \approx .24$, we should only roughly expect this to hold, since most of the signals have amplitudes on the order of .001. In the full case of Eq. (6.3), we have a ratio of Bessel functions $A(q,\omega)e^{i\kappa(q,\omega)} := \frac{J_0\left(k_2\sqrt{q^2+x^2}\right)}{J_0\left(k_2\sqrt{T(q)^2+x^2}\right)}$ whose argument depends linearly on the frequency ω. If $A(q,\omega) \leq 1$, then this can definitely be written in the form of a non-constant frequency shift $\omega(t)$ as in Theorem 3.2.1.

6.3.3. Effects of translation. Increasing the distance x to the object from x_1 to x_2 is a similar transformation to rotation, since every point on the new rail corresponds to a point on the old rail, but with increased radius. If r is the location on the new rail then the point with the same angle is just $T(r) = \frac{x_1}{x_2}r$; since $x_2 > x_1$, this is smaller than y. Then we have the same sort of derivation as above, with a strictly linear function instead of Eq. (6.2).

For translation along a given rail, the dependence on $\Theta(\theta)$ is unavoidable. For a given point $x_\theta(r)$, the radius is simply $\sqrt{r^2+x^2}$, while the angle is $\varphi(r) = \theta + \arctan(r/x)$.

6.4. Classification Results

As a baseline to compare against, we use the same GLMNET logistic classifier on the absolute value of the Fourier transform (AVFT), which is a simple classification technique that is translation invariant and sensitive to frequency shifts. To understand the generalization ability of the techniques, we split the data 10 times into two halves, one half training set and one half test set, uniformly at random, i.e. 10-fold cross validation. For the synthetic dataset, we also normalized the signals and added uniform Gaussian white noise so the SNR is 5dB.

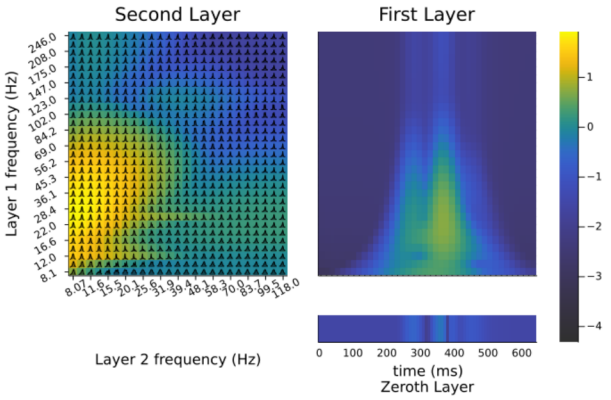


FIGURE 6.7. The ST coefficients for the synthetic triangle with speed of sound is $c_1 = 2000\text{m/s}$, while the ambient speed of sound is $c_2 = 1503\text{m/s}$. We use morlet wavelets, the output subsampling is only 10 instead of the 80 actually used in the classifier.

For the synthetic data, there are two primary problems of interest. The first is determining the effects of varying shape on the scattering transform. An example

of the 1D scattering transform for a triangle is in Fig. 6.7. Since the energy at each layer decays exponentially with layer index, layers 1-3 have been scaled to match the intensity of the first layer. Note that only the zeroth layer has negative values; this is because the nonlinearity used by the scattering transform is an absolute value. In the figure, one can clearly see a time concentrated portion of the signal in layers 0, 1, and 2.

For the real dataset, the problem of interest is somewhat more ambiguous. In addition to a set of UXO's and a set of arbitrary objects, there are some UXO replicas, not all of which are made of the same material. As we will see in the synthetic case, the difference in the speed of sound has a much clearer effect on classification accuracy than shape, so it is somewhat ambiguous how to treat these. We should expect that correctly classifying non-UXO's to be more difficult, since as a class they do not have much in common– a SCUBA tank is much more similar, in both material and shape, to a UXO than to a rock.

6.4.1. Synthetic Classification. For the synthetic dataset, we compare three transforms. The first is the absolute value of the Fourier transform (AVFT). The second, which we will call the coarser ST, is a two layer scattering transform using Morlet Wavelets, with different quality factors rates in each layer: $Q_1 = Q_2 = Q_3 = 1$. Increasing quality factor corresponds to decreasing the rate of scaling of the mother wavelet, and thus gives more coefficients for the same frequency regime. The third, which we will call the finer ST, is another three layer scattering transform with increased quality in all 3 layers: $Q_1 = 8, Q_2 = 4, Q_3 = 4$. For each of these, we use 10-fold cross validation to check the generalization of our results.

We investigate two of the problems from Section 6.3: shape and material discrimination. For the shape discrimination, we compare the triangle and the shark fin, with the material speed of sound fixed at $c_1 = 2000\text{m/s}$, while for the material discrimination, we fix Ω to be a triangle, and compare $c_1 = 2000\text{m/s}$ with $c_1' = 2500\text{m/s}$. To do this comparison, we use the received operator characteristic (ROC) curve, which compares the trade off between false positives and true positives as we change the classification threshold; since it strictly concerns one class, it is insensitive to skewed class sizes [**29**]. A way of summarizing the ROC is the area under the curve (AUC), which simply integrates the total area underneath the curve; we use the trapezoidal approximation. This varies from .5 for random guessing[1] to 1 for the ideal classifier, which doesn't misclassify.

The results for material discrimination are in Fig. 6.8; the corresponding AUCs are .99284 for the AVFT, .97778 for the coarser ST, and .99994 for the finer ST. Fitting with the basic derivation in the geometry

[1] Strictly speaking, values below .5 are possible for particularly terrible classifiers.

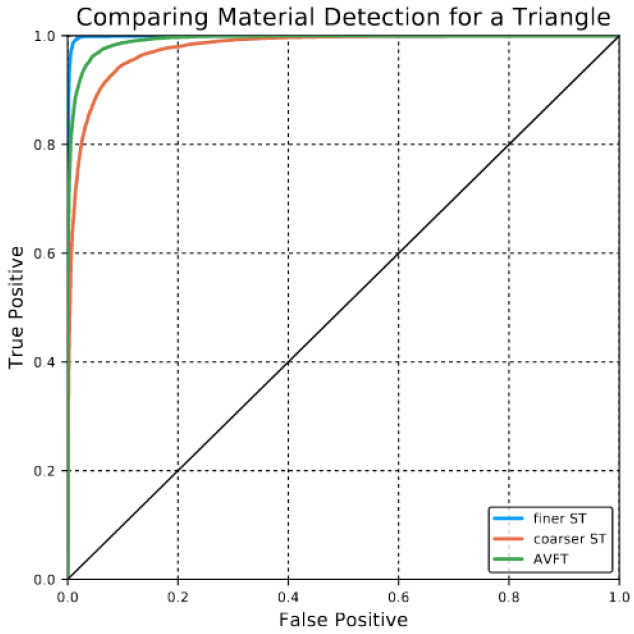


FIGURE 6.8. The ROC curve for detecting the material difference in a triangle, for speeds of sound $c_1 = 2000\text{m/s}$ and $c_1 = 2500\text{m/s}$. Note that the finer ST curve is an ideal classifier, completely in the upper left. The diagonal line is equivalent to random guessing.

TABLE 6.1. The objects in each class for the real dataset

UXO-like	Other Objects
155mm Howitzer with collar	55-gallon drum, filled with water
152mm TP-T	rock
155mm Howitzer w/o collar	2ft aluminum pipe
aluminum UXO replica	Scuba tank, water filled
steel UXO replica	
small Bullet	
DEU trainer (mine-like object)	

Section 6.3, even the AVFT is capable of discriminating material effectively. Somewhat surprisingly, the coarser ST performs worse than the AVFT. This is likely because of insufficient frequency resolution, which the finer ST is able to achieve.

The results for shape discrimination are in Fig. 6.9, and are more definitive in demonstrating the effectiveness of the scattering transform. The coarser ST, with an AUC of .886, outperforms the AVFT with an AUC of .775. But the finer ST clearly outperforms both of these, with an AUC of .998, on par with the classification rates for the speed of sound problem.

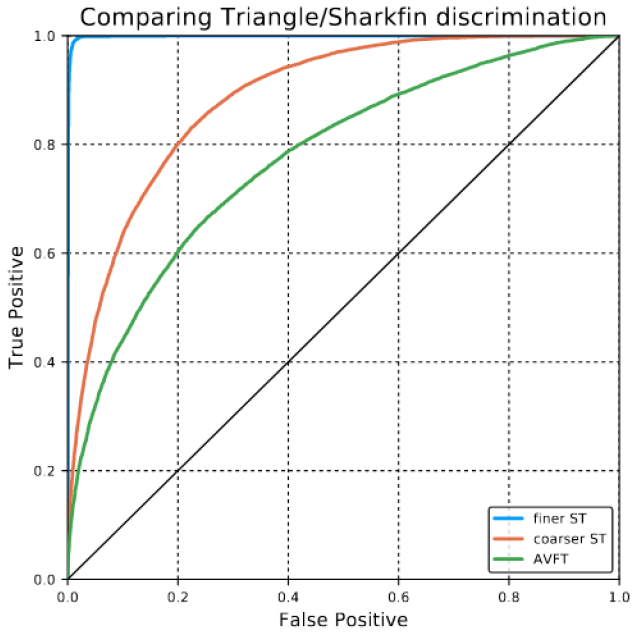


FIGURE 6.9. The ROC curve for discriminating a shark fin from a triangle where both have a speed of sound fixed at 2000m/s.

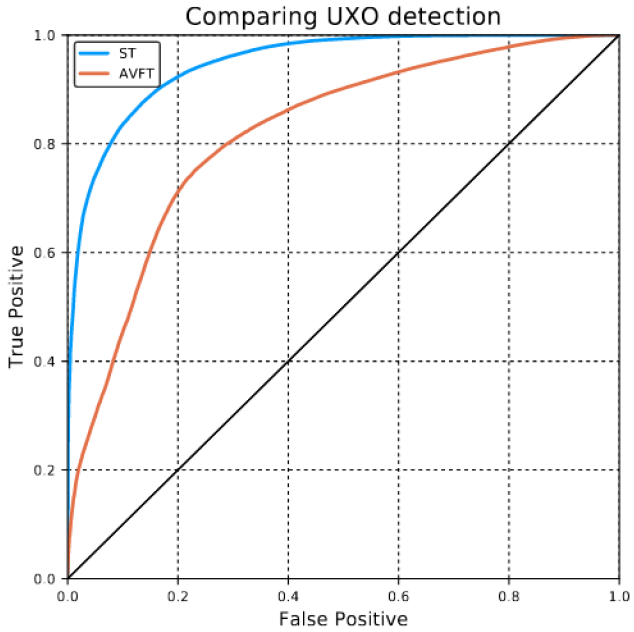


FIGURE 6.10. The ROC curve for detecting UXOs.

6.4.2. Real Classification. For the real dataset, we compare two transforms, the AVFT and a two layer scattering transform with $Q_1 = 8, Q_2 = 1$. We have split the data into a set of objects that are either UXOs or

replicas, and a set of the other objects in the dataset, as listed in Table 6.1. In both classes, there are a variety of materials and shapes. Between classes, there are no similar shapes (as the shape is what determines if it is a replica rather than a UXO), but there are two with the same material (aluminum UXO replica vs aluminum pipe). The ROC curves are in Fig. 6.10. The ST has an AUC of .9487, while the AVFT has an AUC of .8186. The AVFT actually did better on this problem than it had on the shape detection problem, suggesting that it is primarily the material properties of the UXOs that distinguish them.

6.5. Interpretation of Classifiers

In this section we compute the pseudo-inversion of the discrimination weights $\boldsymbol{\beta}$ used in a LASSO logistic regression for two layer classifiers for both speed and shape. Because we have two class classifiers, there is only one vector $\boldsymbol{\beta}$, so unlike in Section 4.4.4, maximizing the inner product corresponds to one class, while minimizing corresponds to the other. For the speed classifier, for example, maximizing $\sum_q \left\langle \boldsymbol{\beta}, \boldsymbol{s}\middle|q\right](\boldsymbol{x})\Big\rangle$ emphasizes features that signals coming from the 2000m/s class have that those coming from the 2500m/s don't have, while minimizing it emphasizes features corresponding to 2500m/s.

6.5.1. Fitting the Speed Classifier. In Fig. 6.11 we have the weights used to distinguish 2000m/s signals from 2500m/s ones. The second layer weights are larger than either the first or zeroth layer weights, and somewhat surprisingly, the increasing paths such as $(31.0\text{Hz}, 12.3\text{Hz})$ and $(31.0\text{Hz}, 9.78\text{Hz})$ are more important for classification than the decreasing paths.

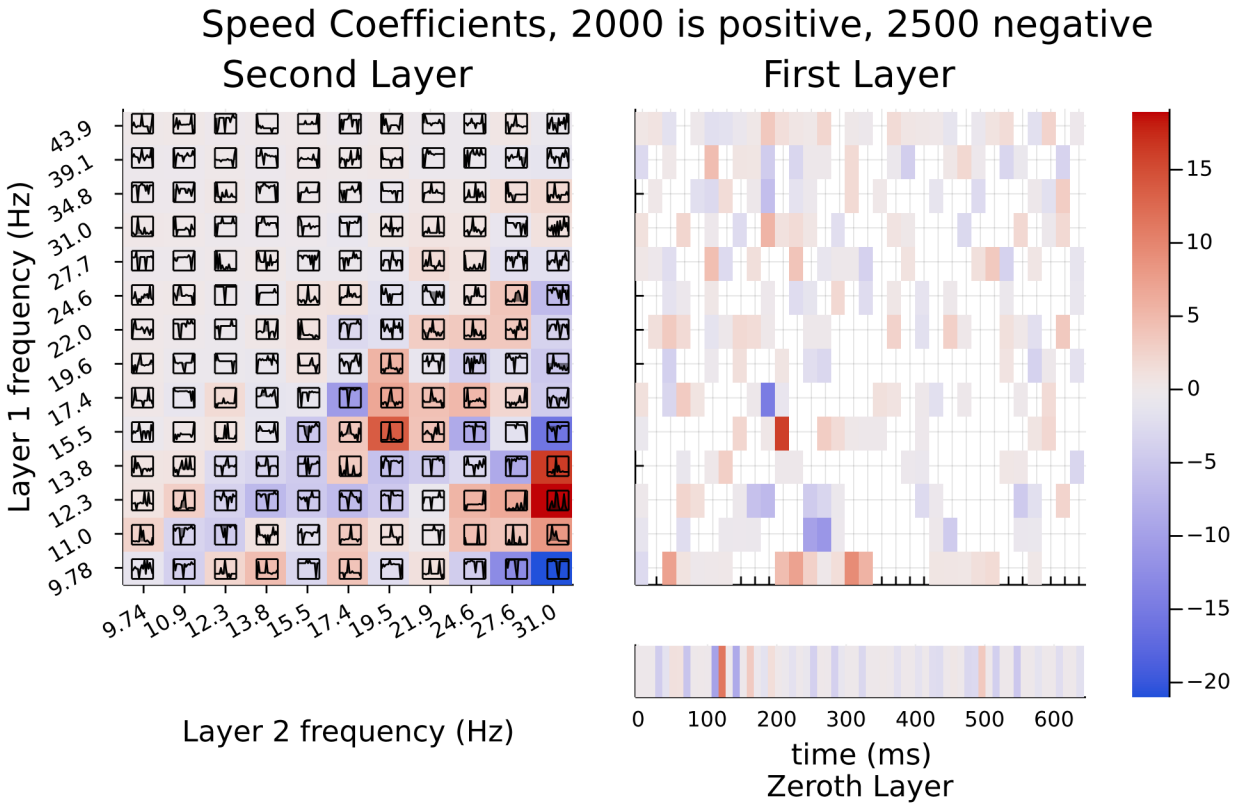
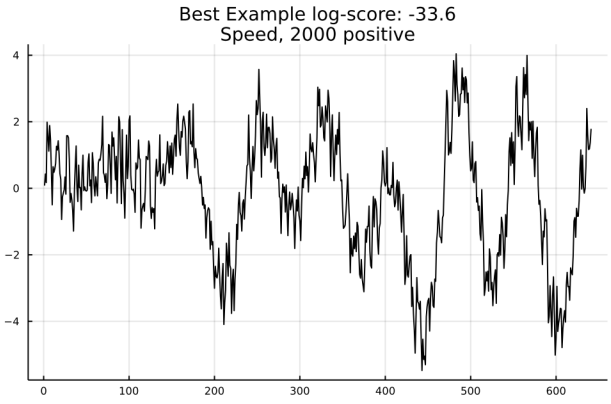


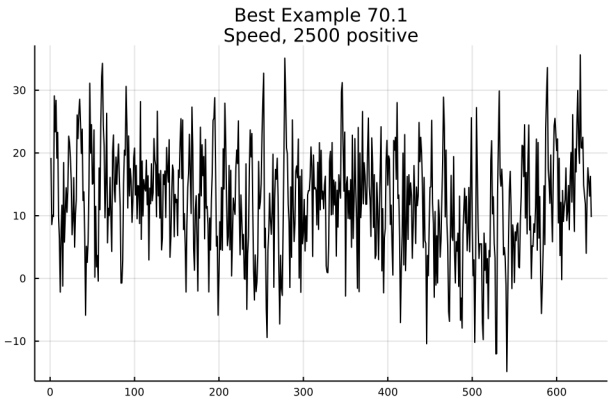
FIGURE 6.11. The Lasso logistic regression weights being fit for speed discrimination. The bias is −2.05, so the prior is ~88.1% that the signal has speed 2500m/s.

The proximity of the paths, frequencies, and times used to distinguish the two will prove somewhat difficult for creating a pseudo-inverse of the weight vector. For example, in the first layer, the most important weight for the speed 2000m/s is at ~220ms and 15.5Hz while for the speed 2500m/s, the most important weight is at ~190ms and 16.4Hz, just 1 index in both space and frequency away. Emphasizing the later and lower frequency response is exactly what we should expect to distinguish the signals generated from the 2000m/s class from the 2500m/s one based on our discussion in Section 6.3.1.

The direct time display of both the pseudo-inversion of $\boldsymbol{\beta}$ and that of $-\boldsymbol{\beta}$ are in Fig. 6.12a and Fig. 6.12b. Fitting $\boldsymbol{\beta}$ in Fig. 6.12a corresponds to the coefficients with best discriminative power in favor of 2000m/s, while Fig. 6.12b corresponds to the coefficients in favor of 2500m/s. As might be expected, Fig. 6.12a has much more prominent lower frequency, and even roughly has the expected shape of a typical example with either speed; initially noise, with a pulse, followed by a gap, and then trailing oscillations. Unlike actual examples signals in, e.g., Fig. 6.13, the oscillations continue until the end of the signal. This is most likely due to the high variation in where the actual signal occurs. The pseudo-inversion of $-\boldsymbol{\beta}$ contains too much high frequency energy for the time domain version to be particularly informative.



(a) Pseudo-inversion of $\boldsymbol{\beta}$



(b) Pseudo-inversion of $-\boldsymbol{\beta}$

FIGURE 6.12. Time domain pseudo-inverses of the speed classifier weights

In Fig. 6.14a and Fig. 6.14b, we have the scalograms using the same wavelets of both fits. Here, the represented frequencies of the pseudo-inverse of $-\boldsymbol{\beta}$ are more clearly visible, along with some spatial variation. Particularly, there is a gap around 190Hz of Fig. 6.14b. The oscillations in the scalogram that characterize the single coordinate fits in Section 4.2.5 are clearly visible in Fig. 6.14b.

The scattering transform of the pseudo-inverses can be found in Fig. 6.15a and Fig. 6.15b, where the color corresponds to the power on a log scale. As expected, the first layer coefficients roughly correspond to

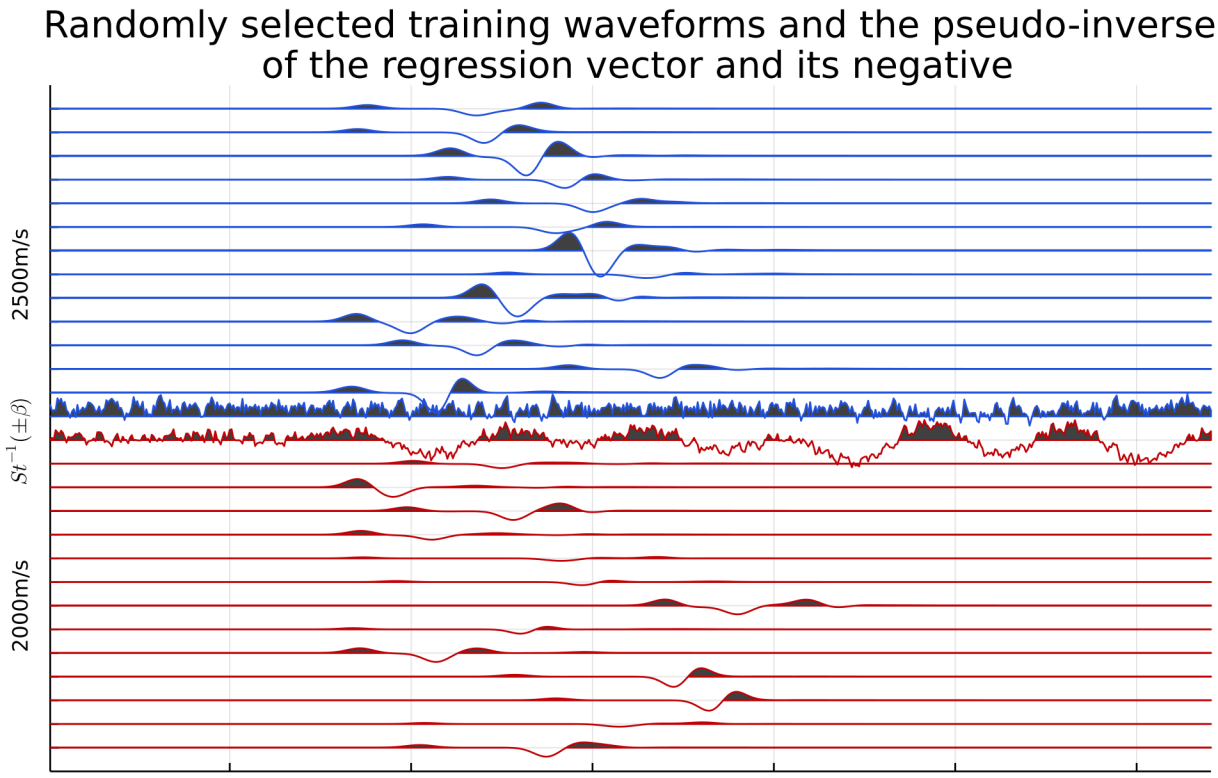
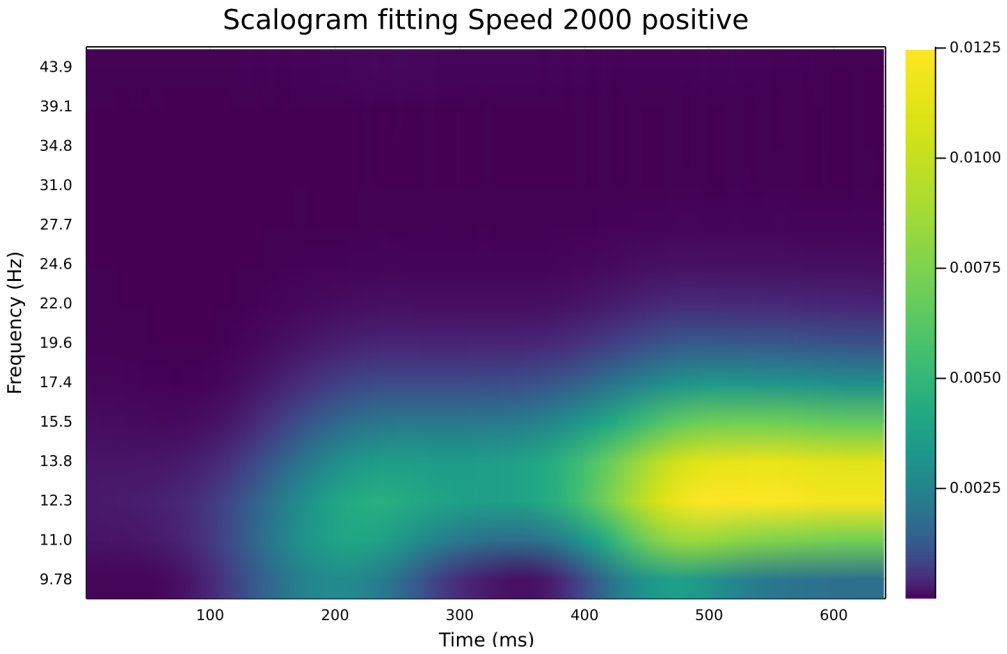


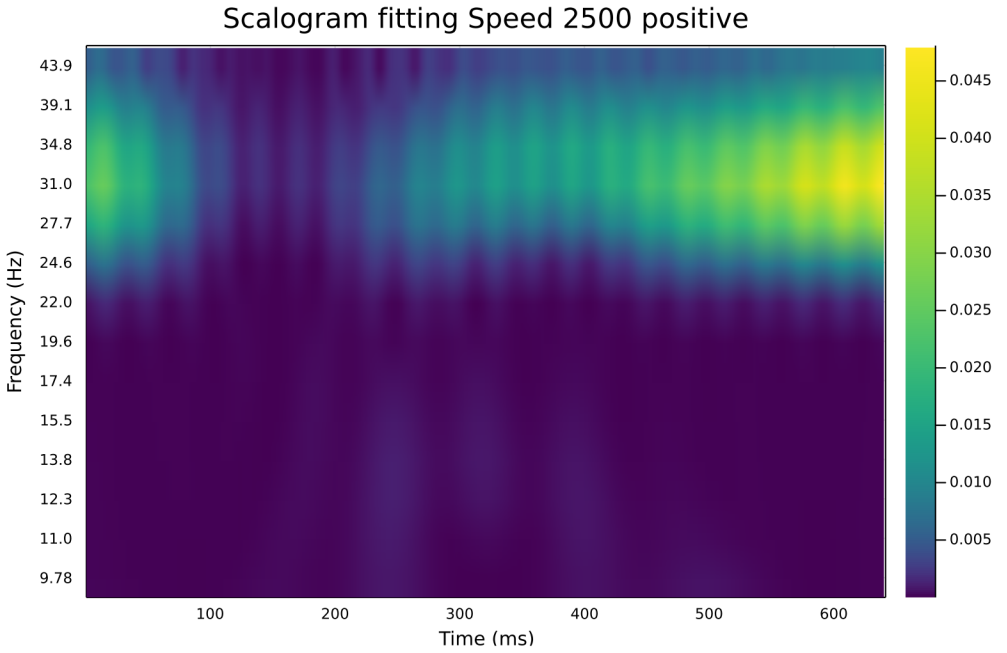
FIGURE 6.13. Comparing randomly selected signals from each speed with the pseudo-inverses emphasizing each respective speed. The first 13 blue signals are random examples at speed 2500m/s, while the last 13 red signals are random examples at speed 2000m/s. The first red signal is the pseudo-inverse of $\boldsymbol{\beta}$ in Fig. 6.11, which corresponds to maximizing the 2000m/s class, while the last blue signal is the pseudo-inverse for $-\boldsymbol{\beta}$, which corresponds to maximizing the 2500m/s class.

the scalograms above, but with a lower time resolution. Comparing the zeroth layer in Fig. 6.15a and Fig. 6.15b shows that the pseudo-inverse of $\boldsymbol{\beta}$ has significantly more variation in the average value than the pseudo-inverse of $-\boldsymbol{\beta}$. Comparing either with the target weights in Fig. 6.11 demonstrates the difficulty in maximizing coefficients for some paths while minimizing coefficients from adjacent paths, particularly when those paths are frequently quite small to begin with. Doing so frequently means maximizing the value at a distant path. To maximize the result at $(31.0\text{Hz}, 12.3\text{Hz})$ and $(31.0\text{Hz}, 13.8\text{Hz})$ in Fig. 6.15a, the paths with the same first layer frequency and lowest second layer frequency, that is $(9.74\text{Hz}, 12.3\text{Hz})$ and $(9.74\text{Hz}, 13.8\text{Hz})$ have the actually largest values, even though the weights in Fig. 6.11 at these paths is zero.

For the speed classifier, the primary difference is one of frequency, along with more variation in the later signal for the speed 2500m/s signals.

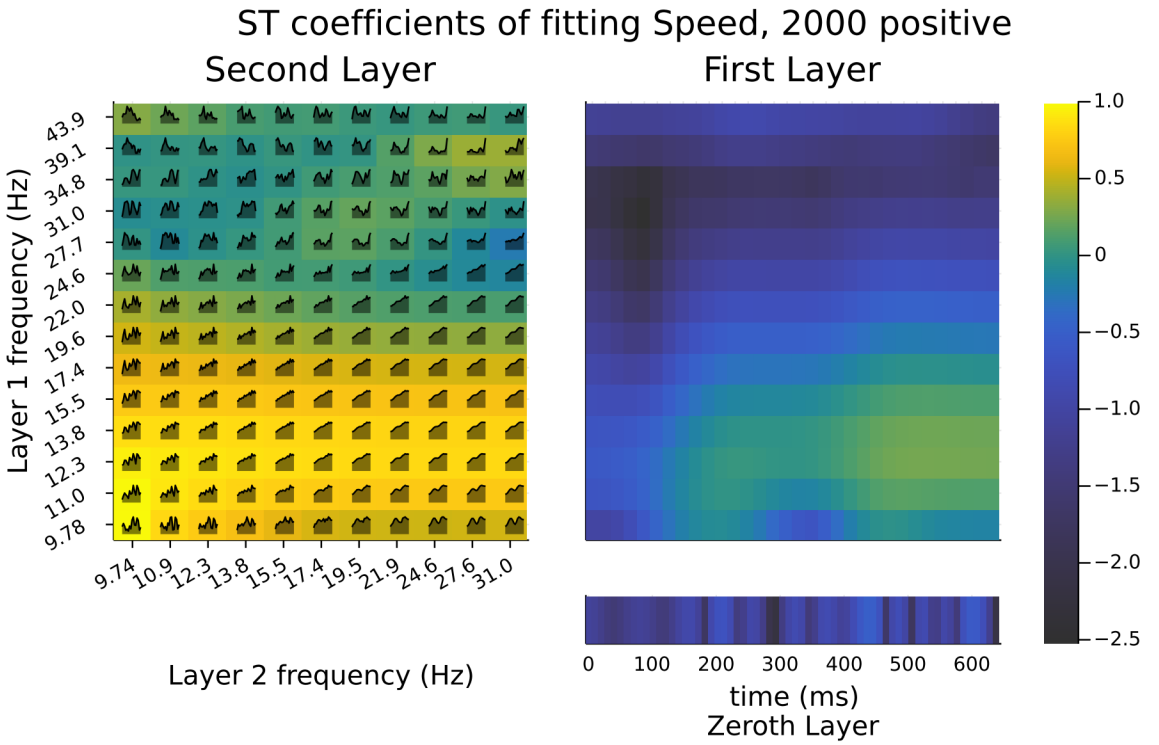


(a) Scalogram of the pseudo-inverse of $\boldsymbol{\beta}$ (maximizing 2000m/s)

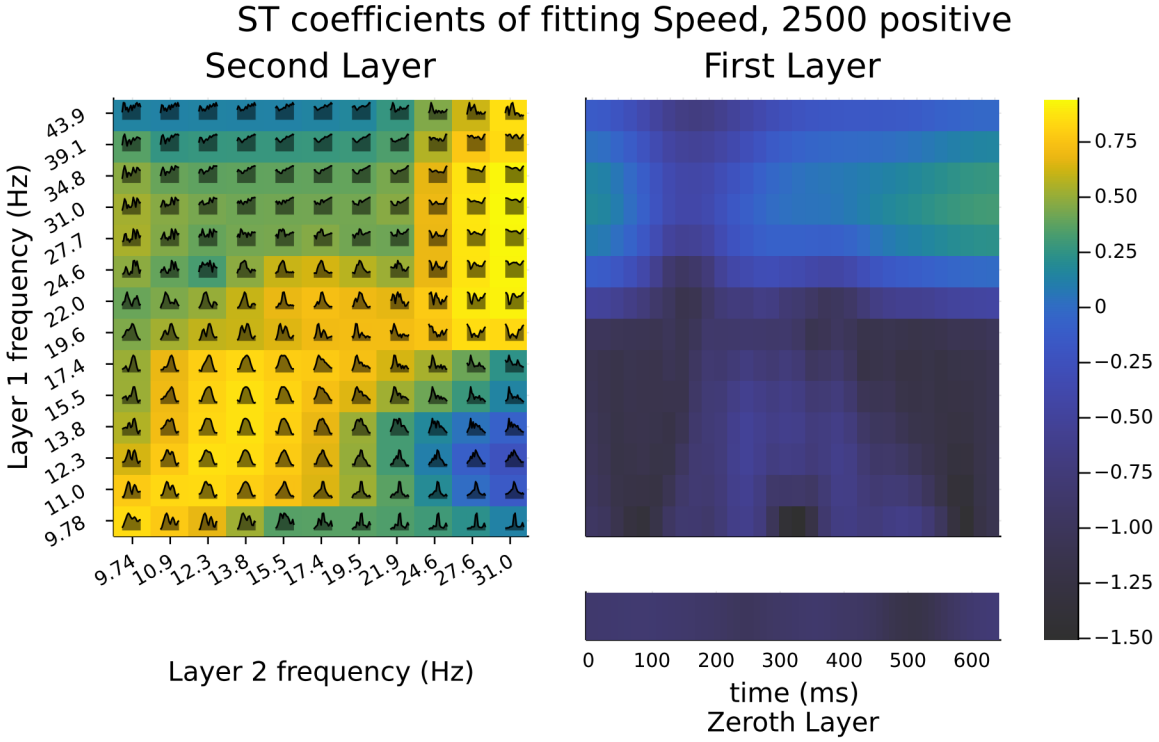


(b) Scalogram of the pseudo-inverse of $-\boldsymbol{\beta}$ (maximizing 2500m/s)

FIGURE 6.14. Scalograms of the pseudo-inverses for the speed problem.



(a) ST coefficients of the pseudo-inverse of $\boldsymbol{\beta}$ (so maximizing the red paths and minimizing the blue paths in Fig. 6.11).



(b) ST coefficients of the pseudo-inverse of $-\boldsymbol{\beta}$ (so maximizing the blue paths and minimizing the red paths in Fig. 6.11)

FIGURE 6.15. ST coefficients of the pseudo-inversion of the speed classifier weights

6.5.2. Fitting the Shape Classifier. The coefficients used for the shape classifier are in Fig. 6.16. Compared to the speed case, there is more emphasis on the second layer coefficients. As in the speed case, the weights are skewed towards increasing paths. Perhaps the strongest difference is that the paths $(13.8\text{Hz}, 12.3\text{Hz})$ and $(15.5\text{Hz}, 12.3\text{Hz})$ are both quite large in magnitude, which may cause some difficulties in finding the pseudo-inverse of this classifier.

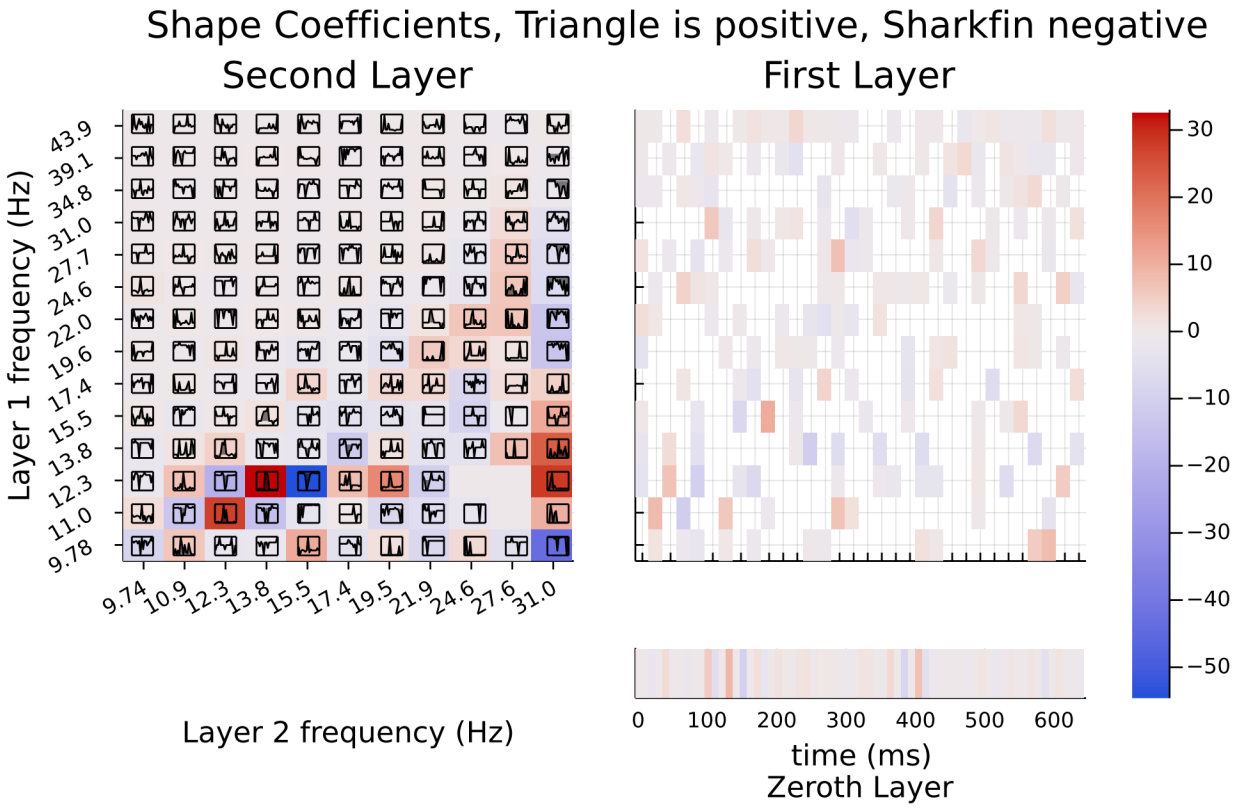
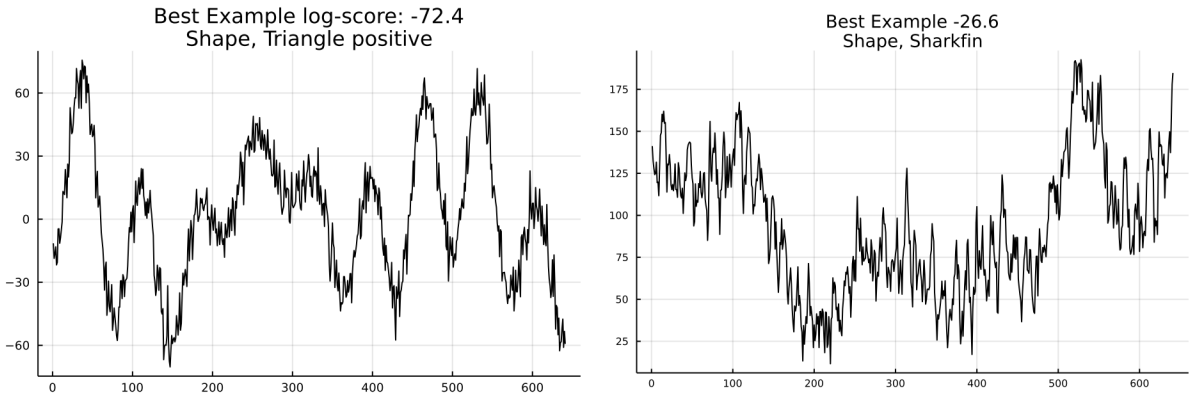


FIGURE 6.16. The Lasso weights being fit for shape discrimination. The bias is 0.67, so the prior is ~66.1% that the signal is a triangle.

The resulting pseudo-inverses are in Fig. 6.17a and Fig. 6.17b. A couple of features to note: the sharkfin pseudo-inverse is strictly positive, with a depressed period roughly in the region where the signals tend to be active, between ~190 – 500ms. While it is consistently oscillatory, the triangle has a lower frequency response in a subset of that range, ~190 – 350ms. The most striking feature of the sharkfin pseudo-inverse is the peak at ~500ms. We can take this as some indication that the sharkfin response tends to both be more positive on average, and have sharper transitions. We can somewhat see this in Fig. 6.18, where the period where the Sharkfin is negative is generally of shorter duration and sharper.



(a) Pseudo-inversion of $\boldsymbol{\beta}$

(b) Pseudo-inversion of $-\boldsymbol{\beta}$

FIGURE 6.17. Single space domain example of the best Pseudo-inversion of $\pm\boldsymbol{\beta}$

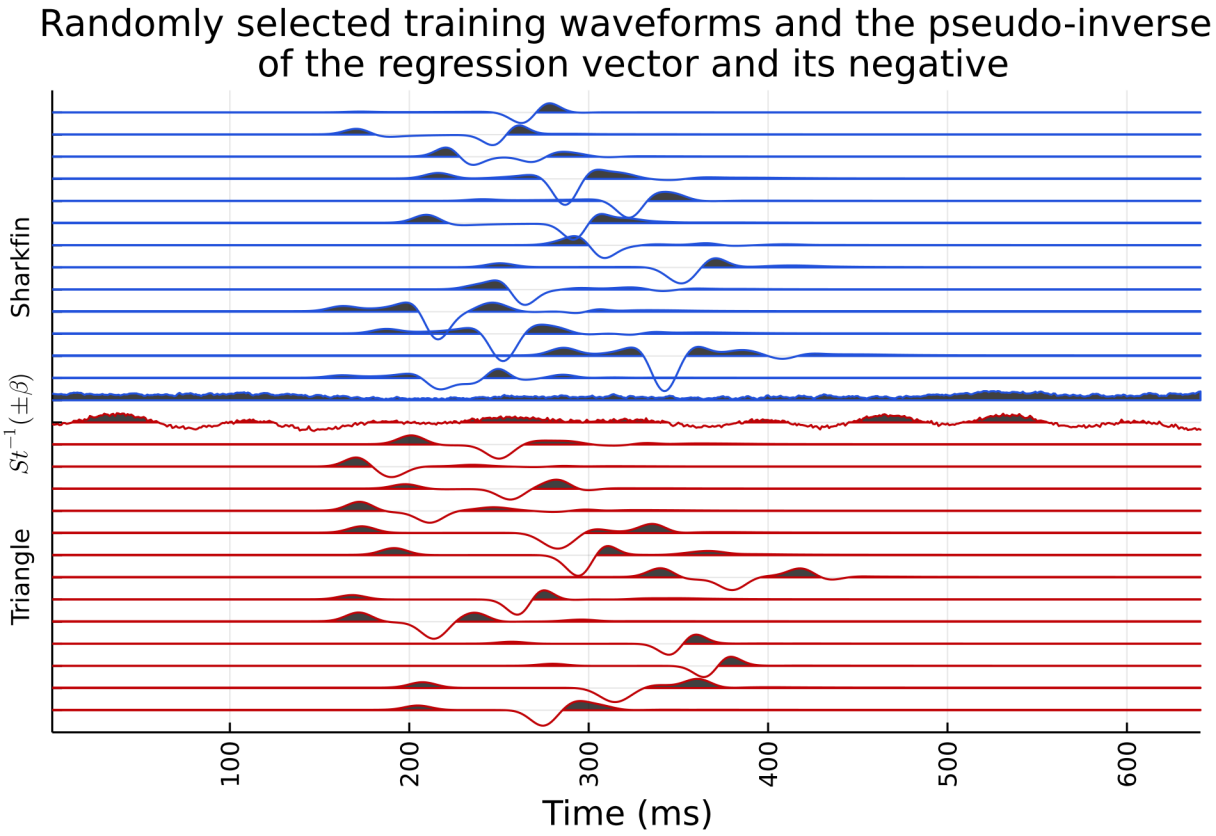
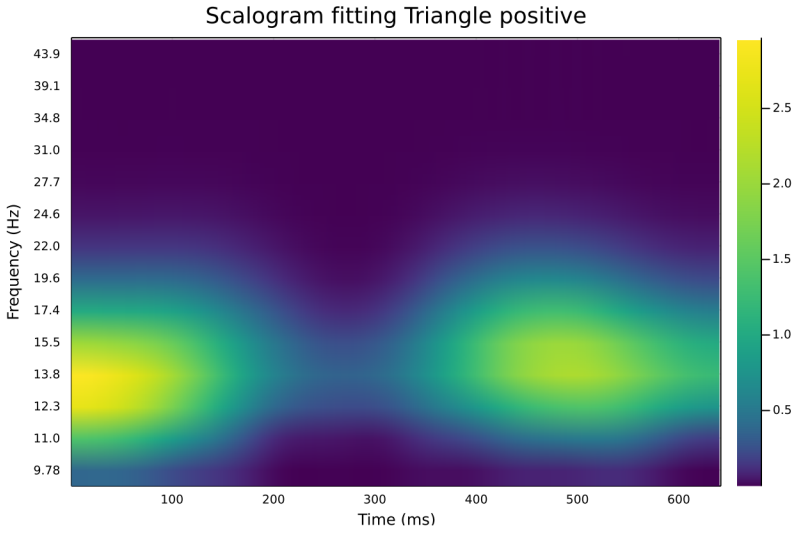
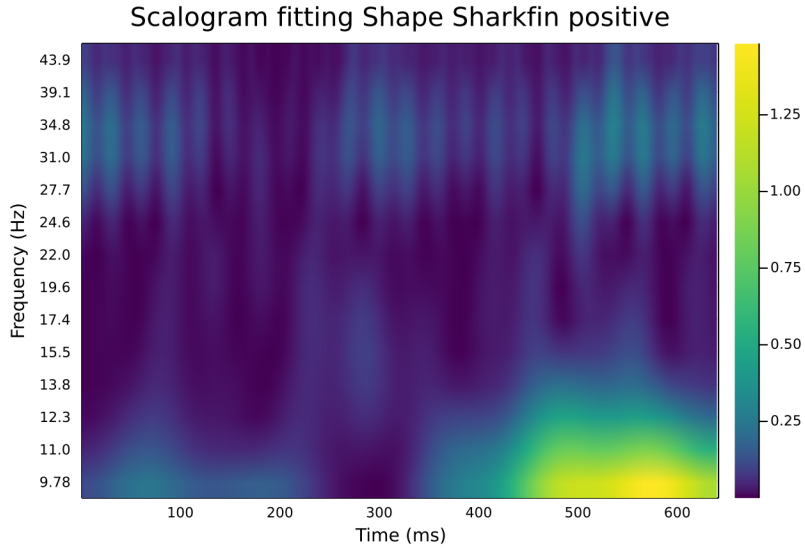


FIGURE 6.18. Comparing randomly selected signals from each shape with the pseudo-inverses emphasizing each respective shape. The first 13 blue signals are random sharkfin examples, while the last 13 red signals are random triangle examples. The first red signal is the pseudo-inverse of $\boldsymbol{\beta}$ in Fig. 6.16, which corresponds to maximizing the triangle class, while the last blue signal is the pseudo-inverse for $-\boldsymbol{\beta}$, which corresponds to maximizing the sharkfin class.

In Fig. 6.19a and Fig. 6.19b we have scalograms of the pseudo-inverses. The triangle fit in Fig. 6.19a is concentrated around frequency 13.8ms, with a spatial gap at around ~250ms. There is more activity at all frequencies for the sharkfin in Fig. 6.19b. The oscillations in the scalogram that are characteristic of high (first layer) frequency second layer coefficients is visible at 34.8Hz throughout, though with a distinct gap again at ~200ms. The jump at ~500ms is part of a low frequency response that continues in that region. One conclusion that we can draw from this is that the sharkfin has more variation at both high and low frequency across different examples.



(a) Scalogram of the pseudo-inverse of $\boldsymbol{\beta}$ (maximizing the triangle).

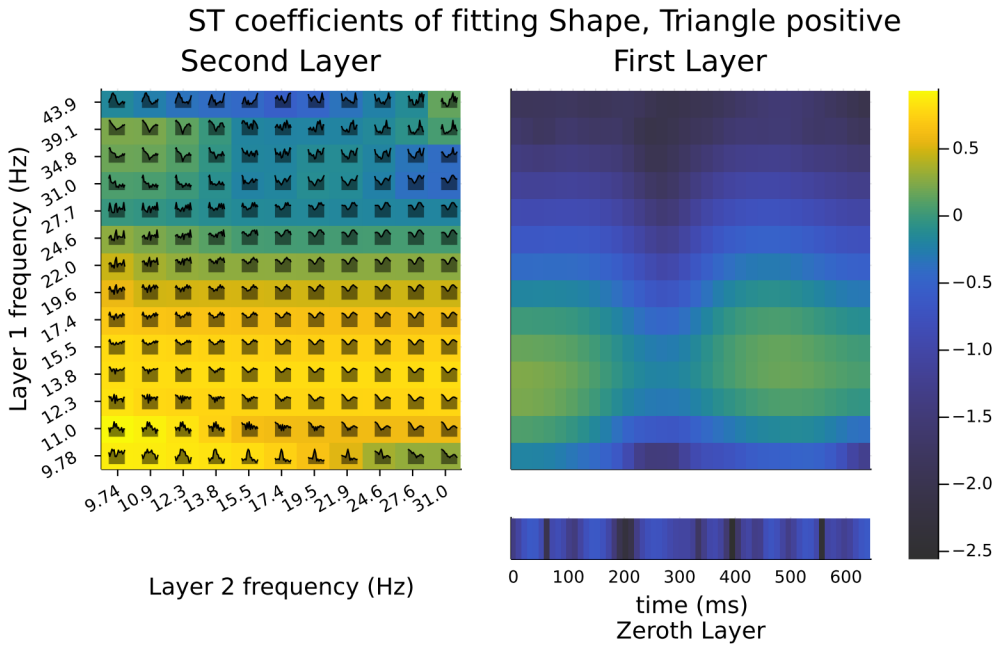


(b) Scalogram of the pseudo-inverse of $-\boldsymbol{\beta}$ (maximizing the sharkfin).

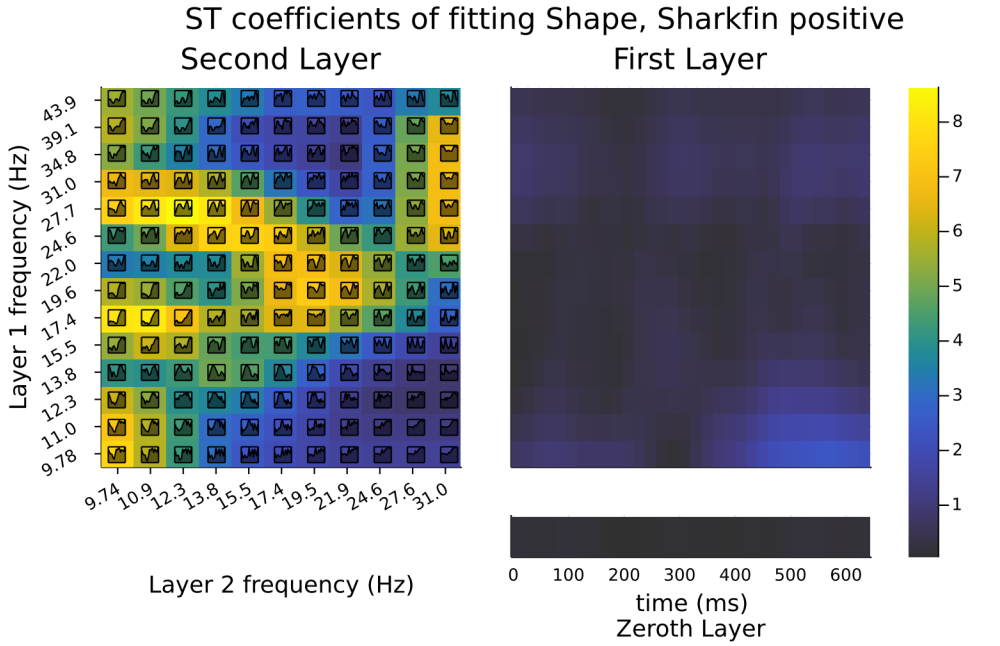
FIGURE 6.19. Scalograms of the pseudo-inverses for the shape problem.

The scattering transform of the pseudo-inverses can be found in Fig. 6.20a and Fig. 6.20b. As may have been expected by the lower importance of the first layer coefficients for the classifier, the resulting ST coefficients for both directions have a stronger response in the second layer, with the sharkfin having a significantly stronger response in the second layer, to the point that it is difficult to actually see the difference in magnitude. There is significant variation between different paths in the second layer for the sharkfin. In contrast, the triangle has an almost uniform second layer response when the first layer frequency is 15.5Hz. Comparing the actual smallest and largest locations in either Fig. 6.20a and Fig. 6.20b with the target Fig. 6.16 shows that in order to maximize these particular locations resulted in completely different coordinates also increasing. For Fig. 6.20a, maximizing both $(13.8\text{Hz}, 12.3\text{Hz})$ and $(12.3\text{Hz}, 11.0\text{Hz})$ while minimizing $(15.5\text{Hz}, 12.3\text{Hz})$ results in a large portion of the bottom left corner having a relatively large value.

6.5.3. Summary. As might be expected based on the target coefficients being distributed across all layers, the pseudo-inverses of both the speed and shape classifiers have features present in fitting the single coordinates in the zeroth, first and second layers, such as the large scale variation of the zeroth layer coefficients, or the oscillations in the scalogram characteristic of the second layer coefficients. For the pseudo-inverse of the 2000m/s speed classifier, this results in a signal that bears some resemblance to the target class signals. More often, the resulting pseudo-inverse has exaggerated versions of the distinguishing features for each class, such as the relative frequency difference that discriminates the speed of sound amplified in Fig. 6.19. This is a natural consequence of using logistic regression, which focuses on coefficients which best and only characterize the difference between the two classes. A reasonable next step would be finding the pseudo-inverses for linear discriminant analysis weights. In the speed case, the distinguishing feature is principally the separation between the initial peak and the following peaks, and is characterized by which of the low frequency second layer paths are largest. Additionally, the tail of second, third, etc., signals have significantly higher frequency for 2500m/s signals. The pseudo-inverse corresponding to the sharkfin is more positive overall, but the troughs are sharper.



(a) ST coefficients of the pseudo-inverse of $\boldsymbol{\beta}$ (so maximizing the red paths and minimizing the blue paths in Fig. 6.16).



(b) ST coefficients of the pseudo-inverse of $-\boldsymbol{\beta}$ (so maximizing the blue paths and minimizing the red paths in Fig. 6.16)

FIGURE 6.20. St coefficients of the pseudo-inversion of the shape discrimination weights.

CHAPTER 7

Conclusion

Throughout this dissertation we have focused on providing interpretations of scattering transform coefficients. This is necessarily a somewhat subjective process.

As one might hope, the meaning of the scattering transform coefficients is quite dependent on the kind of frame used in their construction. There is also great variability in the meaning of the different layers, so that very different features can be characterized all within the same transform. For either kind of wavelet, the zeroth layer coefficients are simply the local average value, while the first layer coefficients correspond directly to averages of the wavelet magnitudes. The second layer begins to get novel behavior. For the analytic wavelets, in the case of paths with increasing frequency, the first layer frequency determines the signal instantaneous frequency, while the second layer frequency determines the envelope. The exact shape of the envelope depends on the wavelet used. The ST coefficients with real-valued wavelets are somewhat more complex, with the second layer coefficients representing a mixture of several orders of distributional derivatives, all evaluated at the level sets of the first layer derivative (in its full detail, this is Eq. (4.11)). In addition to the pseudo-inverses of the particular ST coefficients and the distributional derivative formulation, the gradients for the second layer coefficients in Section 4.1 increase a given coefficient in the second layer by inducing a signal that locally maximizes the response at that frequency, modulated by the second layer wavelet. This is similar to the pseudo-inverse of the coordinate for the analytic case, which is somewhat surprising, given that gradient-based methods weren't well suited to actually solving the pseudo-inversion since the sign change induced by the absolute value meant the gradient frequently changes abruptly.

Both our experiments and theoretical work in Chapter 5 begin to shed some light on the relevance of the non-linearity chosen for the sparsity of the resulting scattering transform. Nonlinearities which preserve the decay rate of coefficients as per Theorem 5.2.1 were significantly better at classifying than those which didn't.

The features revealed in the previous chapters allow us to build classifiers for sonar data that represent meaningful features of the input domain. Using geometric arguments and the synthetic dataset, we have

demonstrated that material detection is a considerably simpler problem than shape detection, and that the scattering transform is capable of solving both problems. For the case of classifying object with varying internal speed of sound, the pseudo-inversion of the depth two classifier emphasizes the difference in frequency as well as the gap between the first and second peaks. For the case of classifying objects with varying shape, the increased complexity of the sharkfin geometry relative to the triangle leads to higher levels of variability in the tail of the response.

In some respects, the work in this dissertation is just the beginning of an examination into the possible uses of the scattering transform in understanding sonar data. The dependence on angle, distance, and subtler geometric features are only briefly touched on, and the real signal objects could be de-aggregated into separate classes and the differences between them better characterized. Further, the third layer and deeper ST coefficients have not been thoroughly examined; if the results for the second layer are any indication, the third layer would nest the second layer signals in another layer of oscillations.

APPENDIX A

Extra Pseudo-inverse Figures

A.1. Pseudo-inversion of Wavelet Coefficients

This appendix serves as a sanity check of the methods developed in Section 4.2 to confirm that they do actually return the original wavelets when fitting the wavelet coefficients. Specifically, we are trying to find the input signal that maximizes the discrete wavelet transform defined in Eq. (4.1). We will discuss a couple of alternatives that demonstrate why we eventually settled on Eq. (4.7). The first is the analog of Eq. (4.7):

$$\min_{\boldsymbol{f}}\left\{\mu\left|\ \boldsymbol{f}\right|^2 - W|\boldsymbol{f}](\lambda_k,\tau_i)\right\} \tag{A.1}$$

where $W|\boldsymbol{f}](\lambda_k,\tau_i)$ is defined in Eq. (4.1), and $\boldsymbol{y}|\lambda_k] = \boldsymbol{e}_{\tau_i}$, where λ_k, τ_i are the target output location τ_i and scale λ_k. We will specifically be working with the case of the Morlet wavelets with mean frequency π.

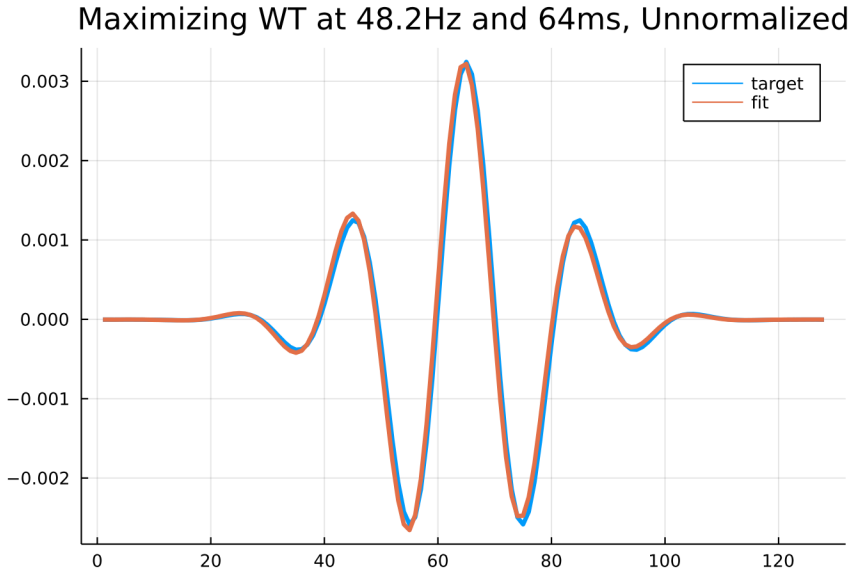


FIGURE A.1. The result from fitting Eq. (A.1), which is nearly identitical to the original wavelet.

One somewhat counter-intuitive result is depending on the kind of normalization imposed on the wavelets themselves (the p from Section 2.2), solving Eq. (A.1) can lead to a different coordinate actually being

larger. The reason for this is that the inner product between the wavelets, $\langle \psi^m_{\lambda_k}, \psi^m_{\lambda_{k'}} \rangle$ is determined by the $\ell^2(\mathbb{Z}^d)$ norm, rather than the other p norms, so by maintaining the $\ell^1(\mathbb{Z}^d)$ norm, the $\ell^2(\mathbb{Z}^d)$ norm decreases with scale. simply because the net $\ell^2(\mathbb{Z}^d)$ mass is larger, which dictates the inner product between the two wavelets. This comes through somewhat in the case of $p = 1$ in the scalogram Fig. A.2, as the scale $\lambda_k = 57.4$Hz actually has some larger coefficients than $\lambda_k = 48.2$Hz itself, the target frequency.

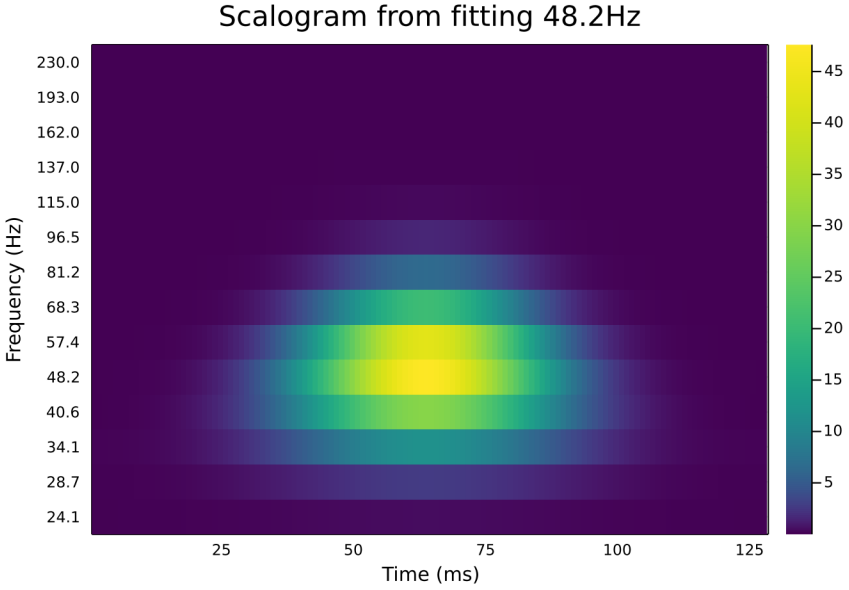


FIGURE A.2. The Scalogram of the fit result in Fig. A.1. Note that this is also effectively the scalogram of the original wavelet, which is clearly non-orthogonal with the other wavelet scales.

A.1.1. Maximizing one coordinate while minimizing others. Because both the CWT and the scattering transform are redundant, non-orthogonal transforms, maximizing one location while minimizing all others can lead to some unexpected behavior. Explicitly, if we are optimizing

$$\min_{\boldsymbol{f}} \left\{ \mu \left| \left| \boldsymbol{f} \right| \right|^2 - \; W|\boldsymbol{f}](\lambda_k, \tau_i) \; + \alpha \sum_{\tau_{i'} \neq \tau_i, \lambda_{k'} \neq \lambda_k} W|\boldsymbol{f}](\lambda_{k'}, \tau_{i'}) \right\} \tag{A.2}$$

for some value of α and μ, then several things can potentially go wrong. If α is chosen too large, then the resulting fit will select frequencies with the narrowest support. Even with α chosen approximately correctly in Fig. A.3 and the corresponding scalogram in Fig. A.4, the wavelet is significantly distorted to minimize the other coefficients.

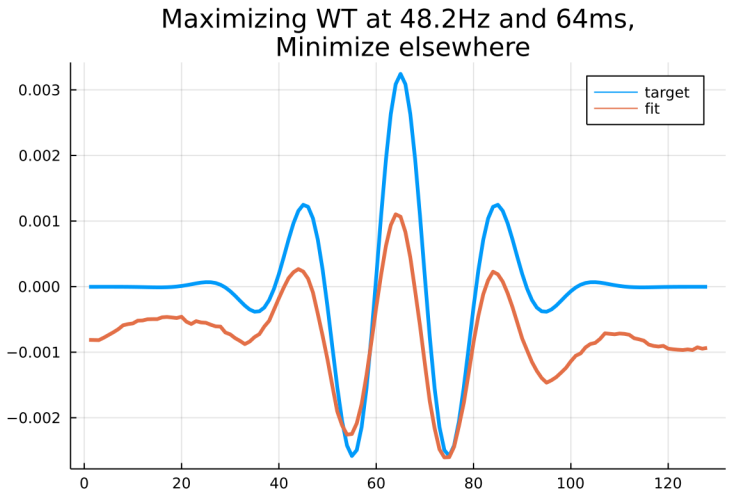


FIGURE A.3. The result of fitting the maximized and minimized Eq. (A.2) with $\mu = 3.3 \times 10^{-8}$ and $\alpha = 3.33 \times 10^{-3}$.

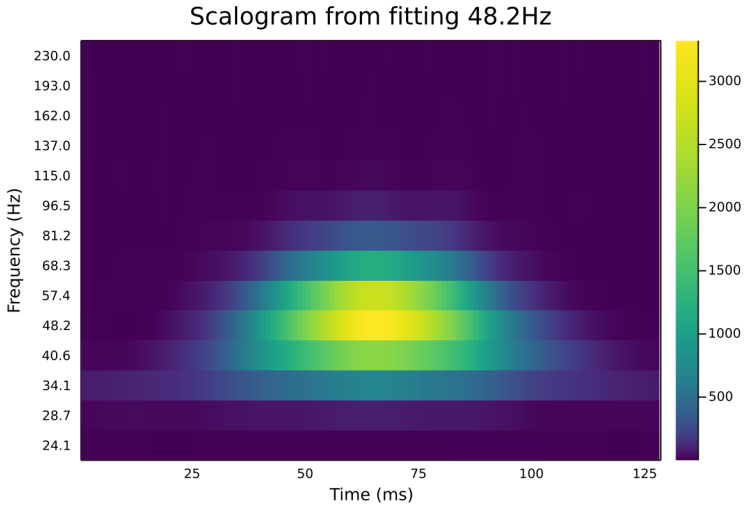


FIGURE A.4. The scalogram corresponding to Fig. A.3. Compared to Fig. A.2, it is slightly more concentrated around the target location. However because of the non-orthogonal nature of the CWT, this results in a worse fit.

A.1.2. Fitting the Normalized Coordinate. While useful for classification, the normalization of the ST coefficients discussed in Section 3.3.2 causes the fit to distort far away from the target wavelet. This is because maximizing the target location necessitates minimizing the other locations, as there is a fixed mass shared between all locations. Unlike the previous section, it has the advantage of not having an extra parameter α to tune, but it suffers a similar problem of over-optimizing for the exact location of a frequency.

Fig. A.6 is even more concentrated at the target frequency and location than either Fig. A.2 or Fig. A.5, but the resulting fit bears no resemblance to the target.

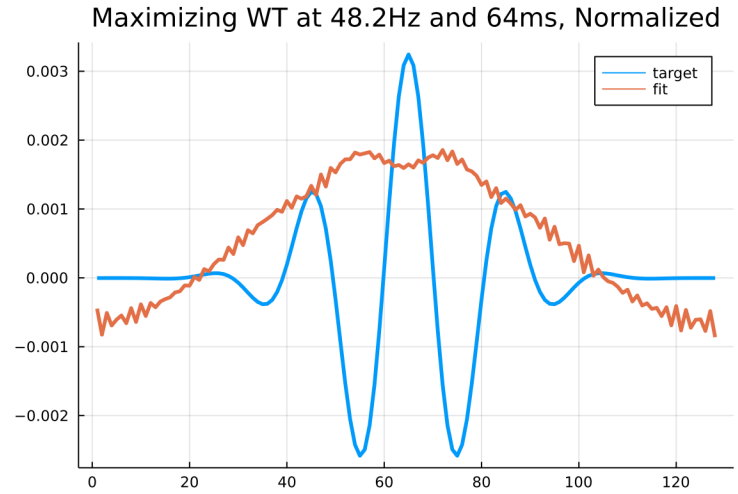


FIGURE A.5. The result of fitting the normalized version of Eq. (A.1). There is very little resemblance between the fit solution and the original wavelets.

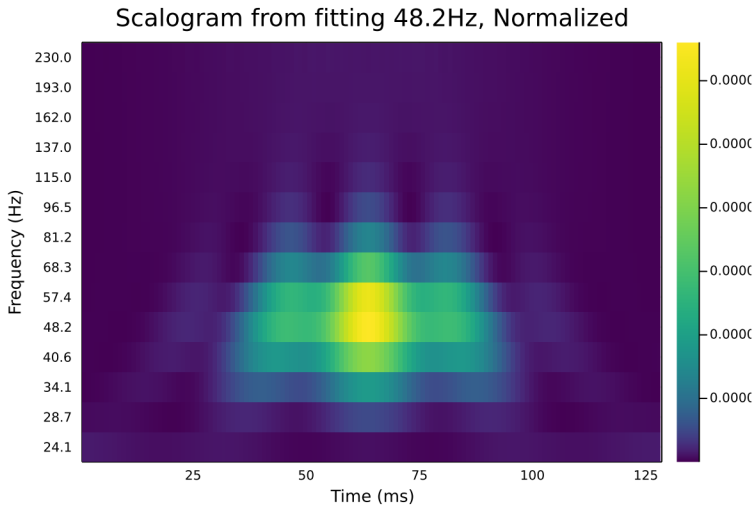
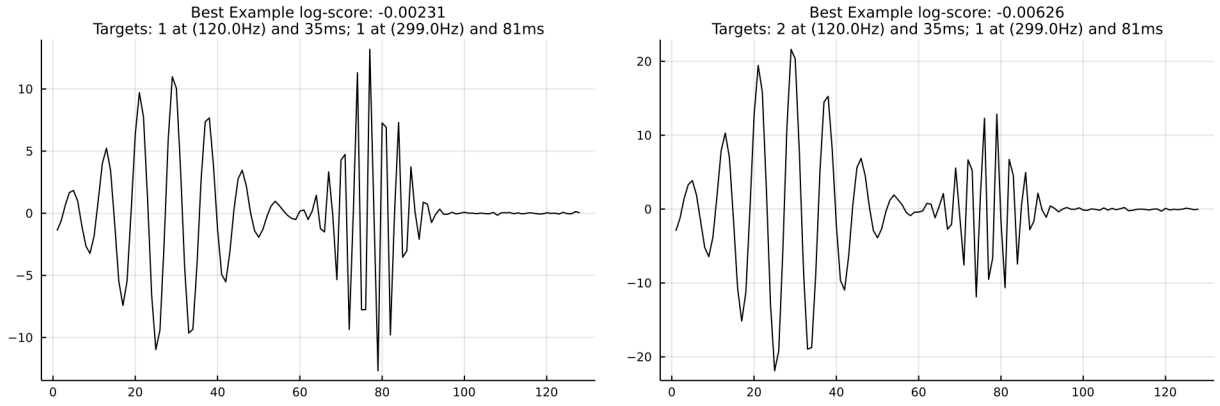


FIGURE A.6. The scalogram

A.2. Penalizing Multiple Paths Simultaneously

In Section 4.2.2 we focused on finding the pseudo-inverse of single coordinates, where $\boldsymbol{y}\big|p\big] = \boldsymbol{e}_k$ for a single p, while in either Section 4.4.4 or Section 6.5 we focus on penalizing entire layers or transforms. This short appendix provides some examples of fitting just a couple of locations, so $\boldsymbol{y}\big|p_i\big] = \boldsymbol{e}_{k_i}$ for $i = 1, 2$

for two paths p simultaneously. In Fig. A.7 we are fitting in the first layer, and have that p_1 = 120Hz with corresponding output time 35ms and p_2 = 299Hz with corresponding output time 81ms. These two paths are well separated in both time and frequency, and they are clearly separated in the resulting space figure. In Fig. A.7a, the two are weighted equally, while in Fig. A.7b we have doubled the weight on the first location, so $y_{35\text{ms}}|120\text{Hz}] = 2$.

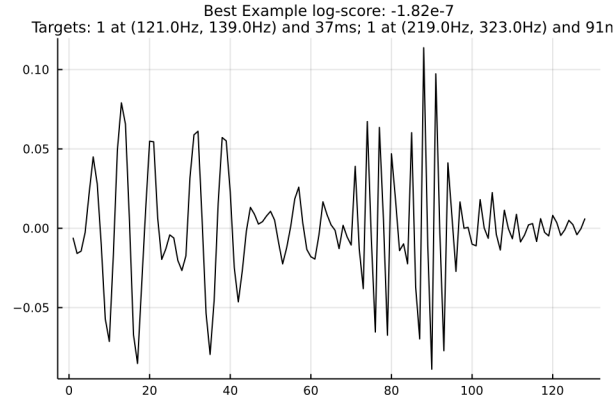


(a) Fitting in the first layer with equal weights and two distant paths

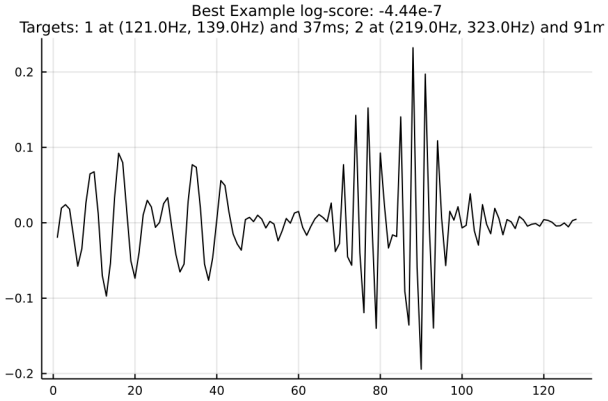
(b) Fitting in the first layer with unequal weights and two distant paths

FIGURE A.7. Fitting multiple paths in the first layer

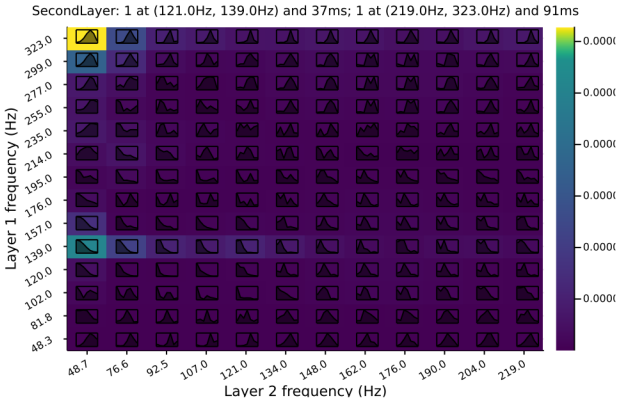
In Fig. A.8 we are fitting in the second layer, and have that $p_1 = (121\text{Hz}, 139\text{Hz})$ with corresponding output time 37ms and $p_2 = (219\text{Hz}, 323\text{Hz})$ with corresponding output time 91ms. These two paths are well separated in both time and frequency, and they are clearly separated in the resulting space figure. In Fig. A.7a, the two are weighted equally, while in Fig. A.7b we have doubled the weight on the first location, so $y_{91\text{ms}}|(219.0\text{Hz}, 323.0\text{Hz})] = 2$. As was the case with maximizing single coordinates, another coordinate ends up being larger in the process of maximizing these locations. The relative weights are less clearly doubled as they were in the first layer.



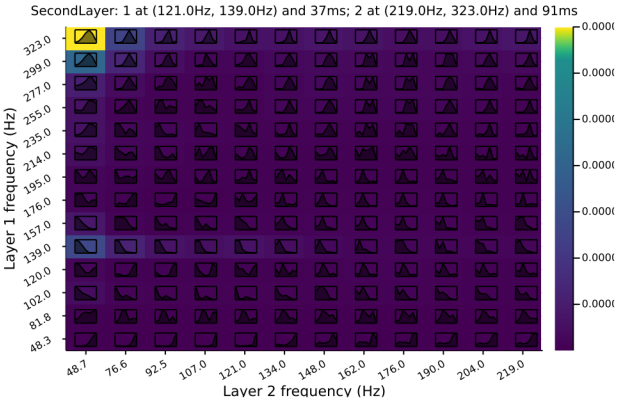
(a) Fitting in the second layer with equal weights and two distant paths



(b) Fitting in the second layer with unequal weights and two distant paths



(c) The second layer coefficients of Fig. A.8a



(d) The second layer coefficients of Fig. A.8b

FIGURE A.8. Fitting multiple paths in the second layer

A.3. Space Domain Second Layer coefficient pseudo-inversion

This appendix is the space domain representation of the pseudo-inverses of the second layer coefficients, mostly displayed in scalograms in the main text.

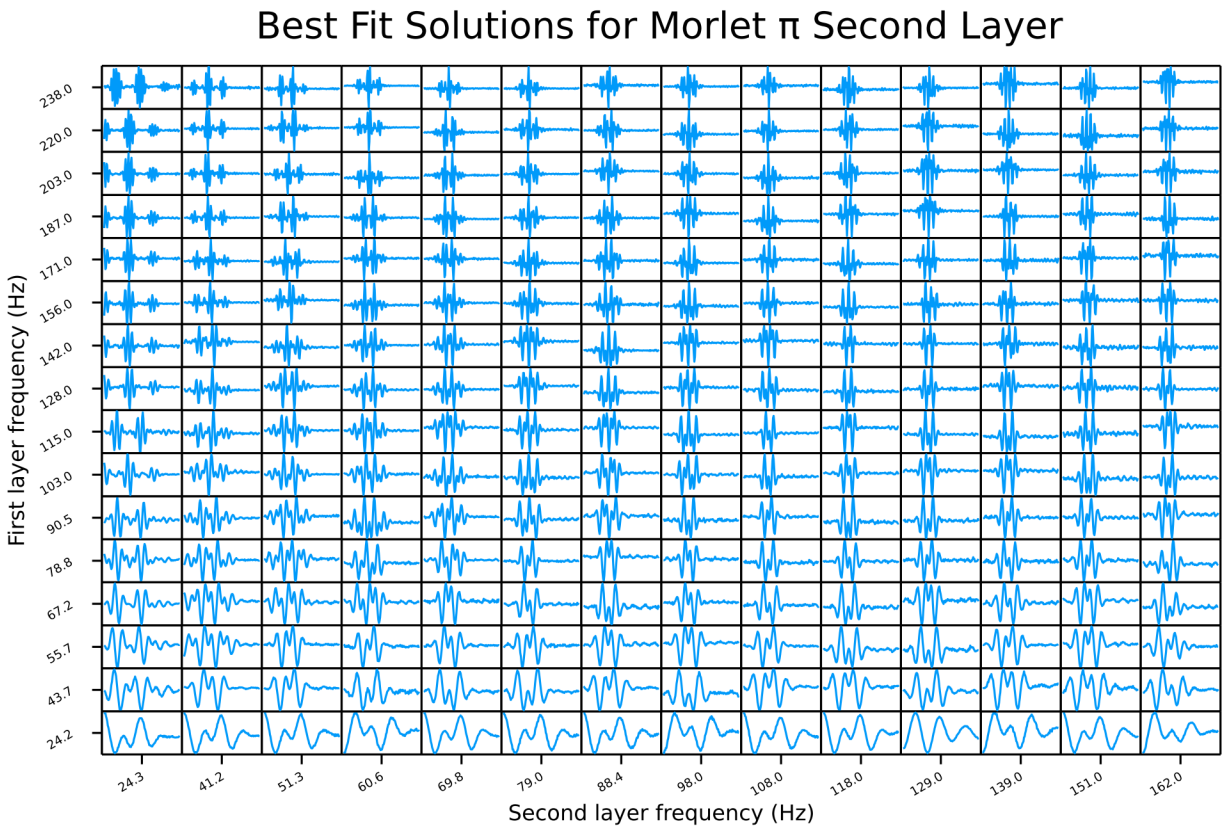


FIGURE A.9. Morlet mean π

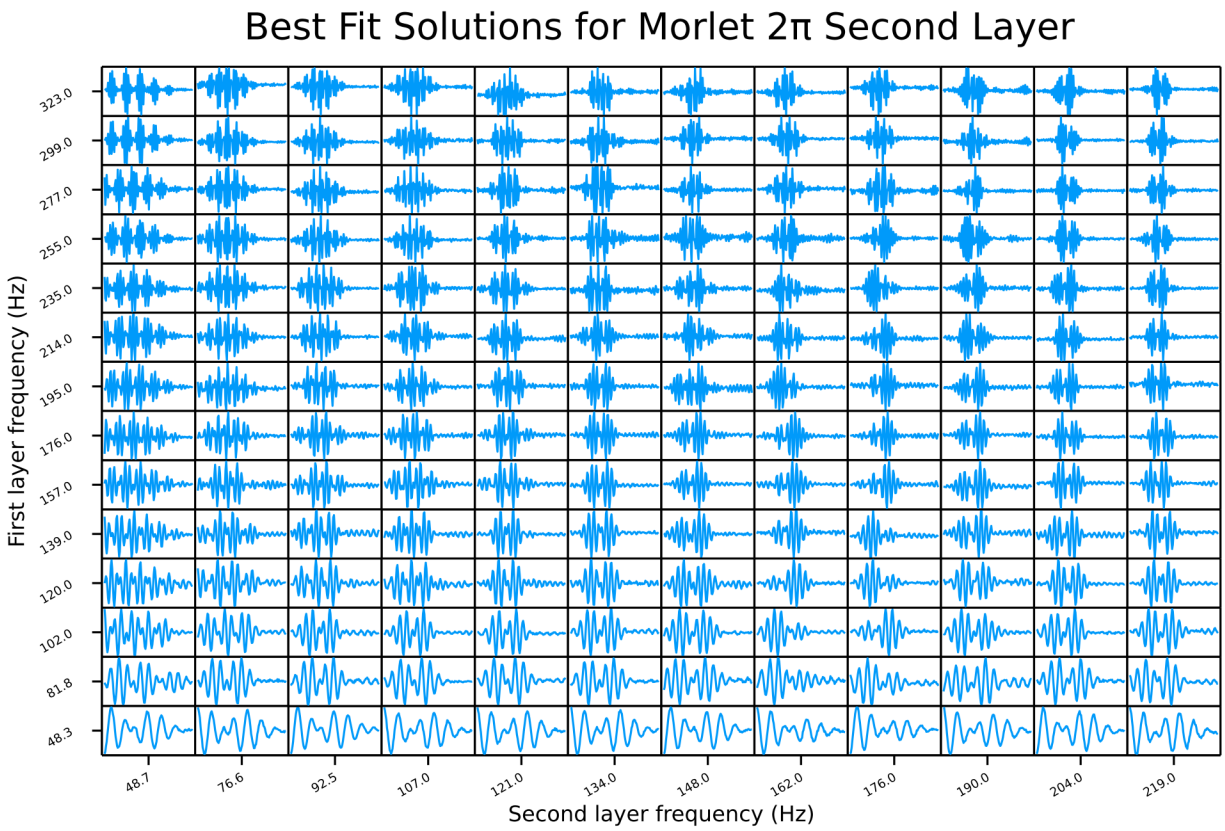


FIGURE A.10. Morlet mean 2π

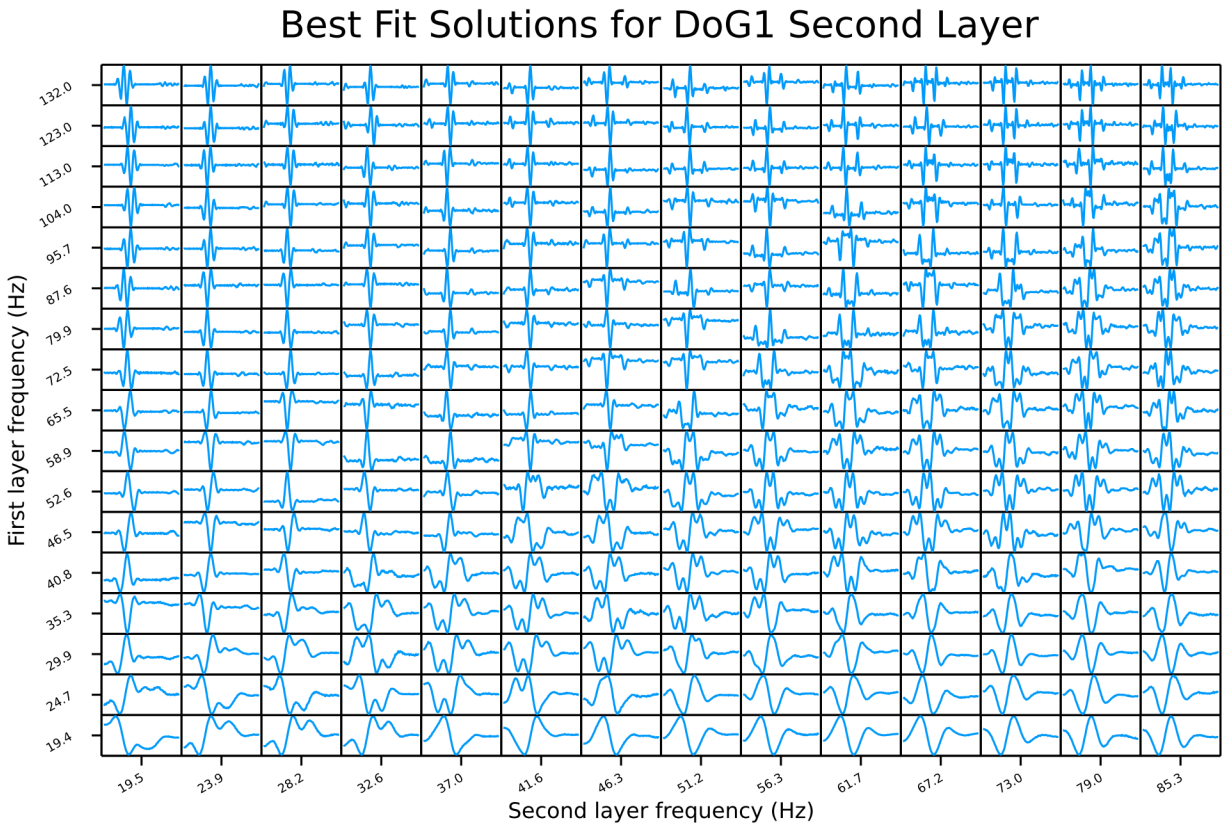


FIGURE A.11. DoG1

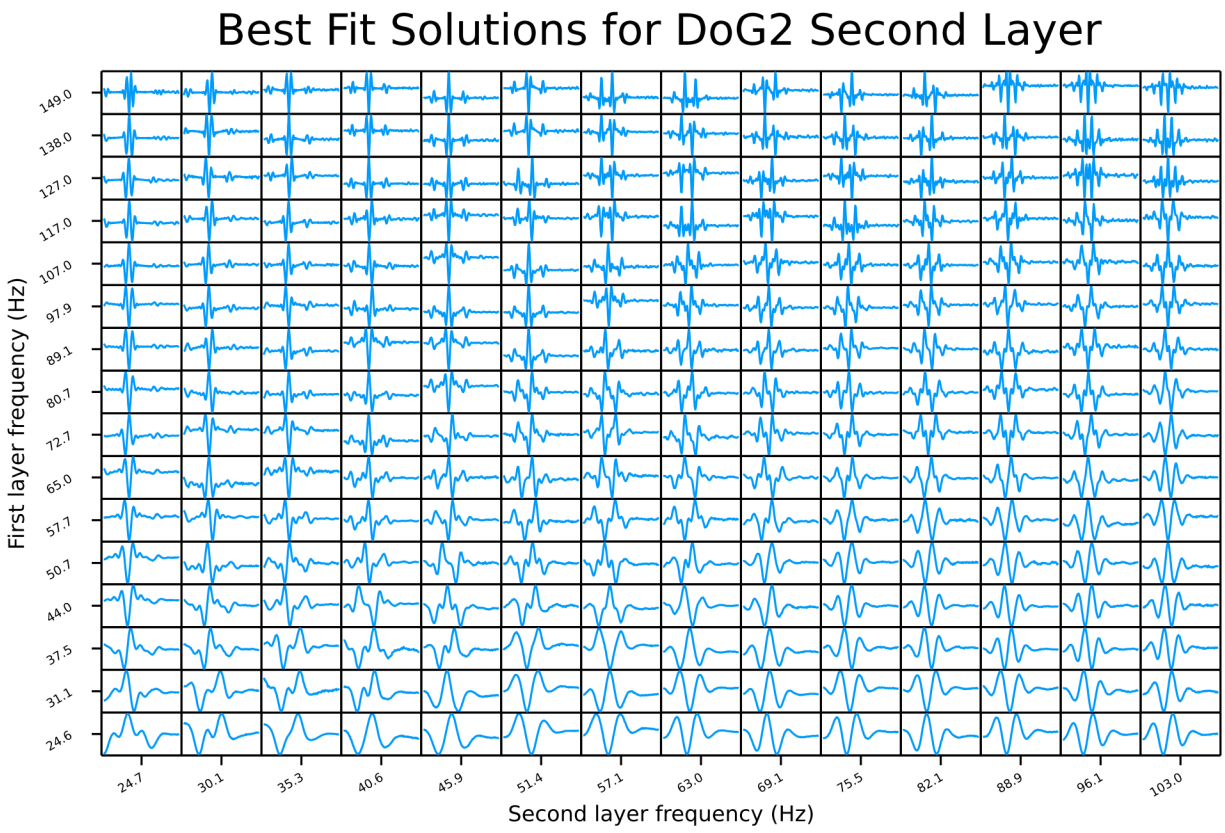


FIGURE A.12. DoG2

APPENDIX B

Extra derivations

B.1. Distributional Chain-rule derivation for absolute value and its derivatives

In this appendix we find the derivatives of $f(x)$, treated as a distribution, as used in Section 4.2. Throughout, let φ be a test function, let f a piecewise infinitely differentiable function which has finitely many roots $Z_f = \{x \quad f(x) = 0\}$ and points of discontinuity D_f, and let $F_f = Z_f \bigcup D_f \bigcup \{-\infty, \infty\}$. Then for x, the distributional derivative is defined via integration by parts:

$$\begin{aligned}\int \frac{\mathrm{d}}{\mathrm{d}x} f\ (x)\varphi(x)\ \mathrm{d}x &:= -\int f\ (x)\varphi'(x)\ \mathrm{d}x = \sum_{x_i \in F_f} -\int_{x_i}^{x_{i+1}} \operatorname{sgn}(f)(x) f(x)\varphi'(x)\ \mathrm{d}x \\ &= \sum_{x_i \in F_f} \left(f(x_{i+1}-)\varphi(x_{i+1}) - f(x_i+)\varphi(x_i) + \int_{x_i}^{x_{i+1}} \operatorname{sgn}(f) f'(x)\varphi(x)\ \mathrm{d}x \right) \\ &= \int \operatorname{sgn}(f) f'(x)\varphi(x)\ \mathrm{d}x + \sum_{x_i \in D_f} f(x_i+)\varphi(x_i) - f(x_i-)\varphi(x_i)\end{aligned}$$

where we have used that $\operatorname{sgn}(f)(x)$ is a constant between x_i and x_{i+1}, since outside of these points it is a non-zero smooth function, the fact that at the points $x_i \in Z_f \setminus D_f$ (so zero but not also discontinuous), $f(x_i)$ is just zero, and the fact that $\varphi(\pm\infty) = 0$ because it is a test function. So in a distributional sense, $\frac{\mathrm{d}}{\mathrm{d}x} f = \operatorname{sgn}(f)(x) f'$, since the distributional derivative of f includes the sum over points of discontinuities [**30**, Theorem 9.1]. For sgn, we know that $\frac{\mathrm{d}}{\mathrm{d}x}\operatorname{sgn}(x) = 2\delta(x)$, so

$$\begin{aligned}\int \frac{\mathrm{d}}{\mathrm{d}x} \operatorname{sgn} f(x)\varphi(x)\ \mathrm{d}x &= \int \operatorname{sgn}(f(x))\varphi'(x)\ \mathrm{d}x = \sum_{x_i \in F_f} \int_{x_i}^{x_{i+1}} \operatorname{sgn}(f(x))\varphi'(x)\ \mathrm{d}x \\ &= \sum_{x_i \in F_f} \operatorname{sgn}(f)(x_{i+1}-)\varphi(x_{i+1}) - \operatorname{sgn}(f)(x_i+)\varphi(x_i)\end{aligned}$$

To go further, we will need to decompose $Z_f \bigcup D_f$ into 3 sets: let

- $N_f = \{x \in Z_f \bigcup D_f \quad \operatorname{sgn}(f)(x-) > \operatorname{sgn}(f)(x+)\}$ (where the sign becomes negative),
- $P_f = \{x \in Z_f \bigcup D_f \quad \operatorname{sgn}(f)(x-) < \operatorname{sgn}(f)(x+)\}$ (where the sign becomes positive), and
- $S_f = \{x \in Z_f \bigcup D_f \quad \operatorname{sgn}(f)(x-) = \operatorname{sgn}(f)(x+)\}$ the sign is the same.

Only the first two are really needed, as both $\operatorname{sgn}(f)(x_i-)\varphi(x_{i+1})$ and $-\operatorname{sgn}(f)(x_i-)\varphi(x_{i+1})$ are equal and opposite for any $x_i \in S_f$. Using these and the fact that $\varphi(\pm\infty) = 0$, we can rearrange the sum to give

$$\int \frac{\mathrm{d}}{\mathrm{d}x}\operatorname{sgn} f(x)\varphi(x)\,\mathrm{d}x = \sum_{x_i \in P_f} 2\varphi(x_i) - \sum_{x_i \in N_f} 2\varphi(x_i)$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\operatorname{sgn} f = \sum_{x_i \in P_f} 2\delta(x - x_i) - \sum_{x_i \in N_f} 2\delta(x - x_i)$$

All further derivatives of $f(x)$ are now derivatives of delta functions, which are unique distributions that evaluate the derivative of the test function they are operating at, e.g. $\delta'\varphi(x) = -\varphi'(0)$ and so on.

Bibliography

[1] M. ALI AND M. PANT, *Improving the performance of differential evolution algorithm using Cauchy mutation*, Soft Comput, 15 (2011), pp. 991–1007.

[2] S. AMARI, *Invariant structures of signal and feature space in pattern recognition problems*, RAAG Memoirs, 4 (1968), pp. 553–566.

[3] J. ANDÉN AND S. MALLAT, *Deep Scattering Spectrum*, IEEE Trans. Signal Process., 62 (2014), pp. 4114–4128.

[4] M. ANDREUX, T. ANGLES, G. EXARCHAKIS, R. LEONARDUZZI, G. ROCHETTE, L. THIRY, J. ZARKA, S. MALLAT, J. ANDÉN, E. BELILOVSKY, J. BRUNA, V. LOSTANLEN, M. J. HIRN, E. OYALLON, S. ZHANG, C. CELLA, AND M. EICKENBERG, *Kymatio: Scattering Transforms in Python*, arXiv:1812.11214 [cs, eess, stat], (2018).

[5] T. ANGLES AND S. MALLAT, *Generative networks as inverse problems with Scattering transforms*, in International Conference on Learning Representations, ICLR 2018, Vancouver, BC, CA, May 2018.

[6] R. BAMMER AND M. DORFLER, *Invariance and stability of Gabor scattering for music signals*, in 2017 International Conference on Sampling Theory and Applications (SampTA), Tallin, Estonia, July 2017, IEEE, pp. 299–302.

[7] J. BEZANSON, A. EDELMAN, S. KARPINSKI, AND V. B. SHAH, *Julia: A Fresh Approach to Numerical Computing*, SIAM Rev., 59 (2017), pp. 65–98.

[8] J. BREMER, *A fast direct solver for the integral equations of scattering theory on planar curves with corners*, J. Comput. Phys., 231 (2012), pp. 1879–1899.

[9] J. BREMER AND Z. GIMBUTAS, *On the numerical evaluation of the singular integrals of scattering theory*, J. Comput. Phys., 251 (2013), pp. 327–343.

[10] J. BRUNA AND S. MALLAT, *Classification with scattering operators*, in Computer Vision and Pattern Recognition (CVPR) 2011, June 2011, pp. 1561–1566.

[11] J. BRUNA AND S. MALLAT, *Invariant scattering convolution networks*, IEEE Trans. Pattern Anal. Mach. Intell., 35 (2013), pp. 1872–1886.

[12] K.-Y. CHANG AND C.-S. CHEN, *A learning framework for age rank estimation based on face images with scattering transform*, IEEE Trans. Image Process., 24 (2015), pp. 785–798.

[13] T. J. CHOI, C. W. AHN, AND J. AN, *An Adaptive Cauchy Differential Evolution Algorithm for Global Numerical Optimization*, Sci World J, 2013 (2013), pp. 1–12.

[14] O. CHRISTENSEN, *An Introduction to Frames and Riesz Bases*, Applied and Numerical Harmonic Analysis, Springer International Publishing, second ed., 2016.

[15] V. CHUDÁČEK, J. ANDÉN, S. MALLAT, P. ABRY, AND M. DORET, *Scattering Transform for Intrapartum Fetal Heart Rate Variability Fractal Analysis: A Case-Control Study*, IEEE T Bio-Med Eng, 61 (2014), pp. 1100–1108.

[16] L. COHEN, P. LOUGHLIN, AND D. VAKMAN, *On an ambiguity in the definition of the amplitude and phase of a signal*, Signal Process, 79 (1999), pp. 301–307.

[17] L. COMTET, *Advanced combinatorics: the art of finite and infinite expansions*, D. Reidel Publishing Company, Dordrecht, [translated from the french by j. w. nienhuys] revised and enlarged ed., 1974.

[18] F. COTTER AND N. KINGSBURY, *Visualizing and Improving Scattering Networks*, in Machine Learning for Signal Processing 2017, Tokyo, Japan, Sept. 2017, IEEE, pp. 1–6.

[19] W. CZAJA AND W. LI, *Analysis of time-frequency scattering transforms*, Appl. Comput. Harmon. Anal., 47 (2019), pp. 149–171.

[20] A. DA CUNHA, J. ZHOU, AND M. DO, *The Nonsubsampled Contourlet Transform: Theory, Design, and Applications*, IEEE Trans. on Image Process., 15 (2006), pp. 3089–3101.

[21] G. E. DAHL, T. N. SAINATH, AND G. E. HINTON, *Improving deep neural networks for LVCSR using rectified linear units and dropout*, in Acoustics, Speech and Signal Processing (ICASSP) 2013, May 2013, pp. 8609–8613.

[22] I. DAUBECHIES, *Ten Lectures on Wavelets*, vol. 61 of Regional Conference Series in Aplied Mathematics, SIAM, Philadelphia, PA, USA, Jan. 1992.

[23] D. L. DONOHO, *Sparse Components of Images and Optimal Atomic Decompositions*, Constr. Approx., 17 (2001), pp. 353–382.

[24] A. DOSOVITSKIY AND T. BROX, *Inverting Visual Representations with Convolutional Networks*, in Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 2015, IEEE Computer Society, pp. 4829–4837.

[25] D. DOV AND I. COHEN, *Voice activity detection in presence of transients using the scattering transform*, in Convention of Electrical Electronics Engineers in Israel (IEEEI), IEEE, Dec. 2014, pp. 1–5.

[26] S. S. DU, C. JIN, J. D. LEE, M. I. JORDAN, A. SINGH, AND B. PÓCZOS, *Gradient Descent Can Take Exponential Time to Escape Saddle Points*, in Advances in Neural Information Processing Systems (NeurIPS), 2017, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., Long Beach, CA, 2017, Curran Associates, Inc., pp. 1067–1077.

[27] D. ERHAN, Y. BENGIO, A. COURVILLE, AND P. VINCENT, *Visualizing Higher-Layer Features of a Deep Network*, Technical Report 1341, University of Montreal, Montreal, QC, Canada, June 2009.

[28] P. C. ETTER, *Underwater Acoustic Modeling and Simulation*, CRC Press, Boca Raton, fifth ed., Apr. 2018.

[29] T. FAWCETT, *An introduction to ROC analysis*, Pattern Recogn Lett, 27 (2006), pp. 861–874.

[30] G. B. FOLLAND, *Fourier Analysis and Its Applications*, no. 4 in Pure and Applied Undergraduate Texts, American Mathematical Society, Belmont, CA, fourth ed., 1992.

[31] J. FRIEDMAN, T. HASTIE, AND R. TIBSHIRANI, *Regularization Paths for Generalized Linear Models via Coordinate Descent*, J Stat Softw, 33 (2010).

[32] I. M. GEL'FAND AND G. E. SHILOV, *Generalized Functions: Properties and Operations*, vol. 1, Academic Press, Inc., New York ; London, 1964.

[33] A. GRIEWANK AND A. WALTHER, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics, Jan. 2008.

[34] D. HAIDER AND P. BALAZS, *Extraction of Rhythmical Features with the Gabor Scattering Transform*, in Proceedings of the 14th International Symposium on Computer Music Multidisciplinary Research (CMMR), Marseille, France, Oct. 14-18, Dec. 2019, pp. 916–923.

[35] T. HASTIE, R. TIBSHIRANI, AND M. WAINWRIGHT, *Statistical Learning with Sparsity - The Lasso and Generalizations*, no. 143 in Monographs on Statistics and Applied Probability, Chapman & Hall/CRC Press, Boca Raton, May 2015.

[36] J. HESTNESS, S. NARANG, N. ARDALANI, G. DIAMOS, H. JUN, H. KIANINEJAD, M. M. A. PATWARY, Y. YANG, AND Y. ZHOU, *Deep Learning Scaling is Predictable, Empirically*, arXiv:1712.00409 [cs, stat], (2017).

[37] L. HÖRMANDER, *The Analysis of Linear Partial Differential Operators I: Distribution Theory and Fourier Analysis*, vol. 1 of A Series of Comprehensive Studies in Mathematics, Springer-Verlag, New York, 1998.

[38] M. INNES, *Flux: Elegant machine learning with Julia*, Journal of Open Source Software, 3 (2018), p. 602.

[39] S. JAFFARD, *Pointwise smoothness, two-microlocalization and wavelet coefficients*, Publ Mat, 35 (1991), pp. 155–168.

[40] S. JAFFARD, *Wavelets: Tools for Science & Technology*, Society for Industrial and Applied Mathematics SIAM Market Street, Floor 6, Philadelphia, PA 19104, 2001.

[41] W. P. JOHNSON, *The Curious History of Faà di Bruno s Formula*, Am Math Mon, 109 (2002), pp. 217–234.

[42] G. KAISER, *A Friendly Guide to Wavelets*, Modern Birkäuser Classics, Birkhäuser Boston, Boston, 1994.

[43] S. KARGL, *Acoustic Response of Underwater Munitions near a Sediment Interface: Measurement Model Comparisons and Classification Schemes*, Tech. Rep. MR-2231, SERDP and University of Washington Seattle Applied Physics Lab, 2015.

[44] P. KITTIPOOM, G. KUTYNIOK, AND W.-Q. LIM, *Construction of Compactly Supported Shearlet Frames*, Constr Approx, 35 (2012), pp. 21–72.

[45] J. KÖLBEL AND F. SEUBRING, *Dealing with UXO (Unexploded Ordnance): Detection, Identification, Disposal and Awareness*, Terra et Aqua, 141 (2015), p. 5.

[46] A. KRIZHEVSKY, I. SUTSKEVER, AND G. E. HINTON, *ImageNet Classification with Deep Convolutional Neural Networks*, in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds., Curran Associates, Inc., 2012, pp. 1097–1105.

[47] G. KUTYNIOK AND D. LABATE, eds., *Shearlets - Multiscale Analysis for Multivariate Data*, Applied and Numerical Harmonic Analysis, Birkhäuser Basel, New York, 2012.

[48] G. KUTYNIOK, W.-Q. LIM, AND R. REISENHOFER, *ShearLab 3D: Faithful Digital Shearlet Transforms Based on Compactly Supported Shearlets*, ACM Trans Math Softw, 42 (2014), pp. 1–42.

[49] Y. LECUN, Y. BENGIO, AND G. HINTON, *Deep learning*, Nature, 521 (2015), pp. 436–444.

[50] Y. LECUN, L. BOTTOU, Y. BENGIO, AND P. HAFFNER, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278–2324.

[51] E. H. LIEB AND M. LOSS, *Analysis*, no. 14 in Graduate Studies in Mathematics, American Mathematical Society, second ed., 2010.

[52] L. LIEU AND N. SAITO, *Signal Ensemble Classification Using Low-Dimensional Embeddings and Earth Mover s Distance*, in Wavelets and Multiscale Analysis, J. Cohen and A. I. Zayed, eds., Birkhäuser Boston, Boston, 2011, pp. 227–256.

[53] J. M. LILLY AND S. C. OLHEDE, *On the Analytic Wavelet Transform*, IEEE T Infom Theory, 56 (2010), pp. 4135–4156.

[54] A. MAHENDRAN AND A. VEDALDI, *Understanding deep image representations by inverting them*, in Computer Vision and Pattern Recognition, CVPR 2015, IEEE, June 2015, pp. 5188–5196.

[55] S. MALLAT, *A Wavelet Tour of Signal Processing: The Sparse Way*, Academic Press, Burlington, MA, third ed., 2009.

[56] S. MALLAT, *Group Invariant Scattering*, Commun Pur Appl Math, 65 (2012), pp. 1331–1398.

[57] ———, *Understanding deep convolutional networks*, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374 (2016), p. 20150203.

[58] B. MARCHAND, N. SAITO, AND H. XIAO, *Classification of Objects in Synthetic Aperture Sonar Images*, in Workshop on Statistical Signal Processing 2007, Aug. 2007, pp. 433–437.

[59] P. A. MEROLLA, J. V. ARTHUR, R. ALVAREZ-ICAZA, A. S. CASSIDY, J. SAWADA, F. AKOPYAN, B. L. JACKSON, N. IMAM, C. GUO, Y. NAKAMURA, B. BREZZO, I. VO, S. K. ESSER, R. APPUSWAMY, B. TABA, A. AMIR, M. D. FLICKNER, W. P. RISK, R. MANOHAR, AND D. S. MODHA, *A million spiking-neuron integrated circuit with a scalable communication network and interface*, Science, 345 (2014), pp. 668–673.

[60] J. NOCEDAL AND S. WRIGHT, *Numerical Optimization*, Springer Series in Operations Research, Springer, second ed., Dec. 2006.

[61] S. OLHEDE AND A. WALDEN, *Generalized Morse wavelets*, IEEE T Signal Proces, 50 (2002), pp. 2661–2670.

[62] O. OTSU, *An invariant theory of Linear Functionals as Linear Feature Extractors*, Bulletin of the Electrotechnical Laboratory, 37 (1973), pp. 893–913.

[63] K. V. PRICE, R. M. STORN, AND J. A. LAMPINEN, *Differential Evolution: A Practical Approach to Global Optimization*, Natural Computing Series, Springer, Berlin ; New York, 2005.

[64] T. M. RAGONNEAU AND Z. ZHANG, *PDFO: Cross-Platform Interfaces for Powell s Derivative-Free Optimization Solvers*. Zenodo, June 2020.

[65] W. RUDIN, *Principles of Mathematical Analysis*, International Series in Pure and Applied Mathematics, McGraw-Hill, New York, third ed., 1976.

[66] N. SAITO AND G. BEYLKIN, *Multiresolution representations using the autocorrelation functions of compactly supported wavelets*, IEEE T Signal Proces, 41 (1993), pp. 3584–3590.

[67] N. SAITO AND R. R. COIFMAN, *Local discriminant bases and their applications*, Journal of Mathematical Imaging and Vision, 5 (1995), pp. 337–358.

[68] N. SAITO AND D. S. WEBER, *Underwater object classification using scattering transform of sonar signals*, in Wavelets and Sparsity XVII, vol. 10394, International Society for Optics and Photonics, Aug. 2017, p. 103940K.

[69] A. SALTELLI, *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*, no. 20859 in EUR, Wiley, Hoboken, NJ, 2004.

[70] K. SIMONYAN, A. VEDALDI, AND A. ZISSERMAN, *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*, in International Conference on Learning Representations, ICLR 2014, Banff, AB, CA, Apr. 2014.

[71] E. M. STEIN AND R. SHAKARCHI, *Fourier Analysis: An Introduction*, vol. 1 of Princeton Lectures in Analysis, Princeton University Press, Princeton, NJ, Apr. 2003.

[72] Y. SUN, X. WANG, AND X. TANG, *Deep Convolutional Network Cascade for Facial Point Detection*, in Conference on Computer Vision and Pattern Recognition, CVPR 2013, IEEE, June 2013, pp. 3476–3483.

[73] R. TIBSHIRANI, J. FRIEDMAN, AND T. HASTIE, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics, Springer, New York, NY, second ed., 2009.

[74] C. TORRENCE AND G. P. COMPO, *A Practical Guide to Wavelet Analysis*, Bulletin of the American Meteorological Society, 79 (1998), pp. 61–78.

[75] D. VAKMAN, *On the analytic signal, the Teager-Kaiser energy algorithm, and other methods for defining amplitude and frequency*, IEEE T Signal Proces, 44 (1996), pp. 791–797.

[76] I. WALDSPURGER, *Phase Retrieval for Wavelet Transforms*, IEEE Transactions on Information Theory, 63 (2017), pp. 2993–3009.

[77] T. WIATOWSKI AND H. BOLCSKEI, *A Mathematical Theory of Deep Convolutional Neural Networks for Feature Extraction*, IEEE Transactions on Information Theory, 64 (2018), pp. 1845–1866.

[78] T. WIATOWSKI, M. TSCHANNEN, A. STANIC, P. GROHS, AND H. BOLCSKEI, *Discrete Deep Feature Extraction: A Theory and New Architectures*, Proc. of International Conference on Machine Learning (ICML), (2016).

[79] M. V. WICKERHAUSER, *Adapted Wavelet Analysis from Theory to Software*, CRC Press, Wellesley, MA, 1994.

[80] H. XIAO, K. RASUL, AND R. VOLLGRAF, *Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms*, arXiv:1708.07747 [cs, stat], (2017).

[81] A. L. YUILLE AND T. A. POGGIO, *Scaling theorems for zero crossings*, IEEE Trans Pattern Anal Mach Intell, 8 (1986), pp. 15–25.

[82] L. J. ZIOMEK, *An Introduction to Sonar Systems Engineering*, CRC Press, Boca Raton, FL, Dec. 2016.