Foveal autoregressive neural time-series modeling

Mathieu Andreux

THÈSE DE DOCTORAT

de l'Université de recherche Paris Sciences et Lettres
PSL Research University

Préparée à l'École normale supérieure

Foveal Autoregressive Neural Time-Series Modeling

École doctorale n$^{\text{o}}$386

SCIENCES MATHÉMATIQUES DE PARIS CENTRE

Spécialité INFORMATIQUE

COMPOSITION DU JURY :

M. DUPOUX Emmanuel
ENS, Président du jury

M. TORRESANI Bruno
Aix-Marseille Université, Rapporteur

M. VINCENT Emmanuel
INRIA Grand Est, Rapporteur

M. LAGRANGE Mathieu
Centrale Nantes, Examinateur

M. WAINRIB Gilles
Owkin, Examinateur

M. MALLAT Stéphane
ENS, Directeur de thèse

Soutenue par **Mathieu ANDREUX**
le 12 novembre 2018

Dirigée par **Stéphane MALLAT**

FOVEAL AUTOREGRESSIVE NEURAL TIME-SERIES MODELING

MATHIEU ANDREUX

Département d’informatique
École normale supérieure, PSL Research University

Mathieu Andreux: *Foveal Autoregressive Neural Time-Series Modeling,*
This work is supported by the ERC InvariantClass grant 320959.

RÉSUMÉ

Cette thèse s'intéresse à la modélisation non-supervisée de séries temporelles univariées. Nous abordons tout d'abord le problème de prédiction linéaire des valeurs futures de séries temporelles gaussiennes sous hypothèse de longues dépendances, qui nécessitent de tenir compte d'un large passé. Nous introduisons une famille d'ondelettes fovéales et causales qui projettent les valeurs passées sur un sous-espace adapté au problème, réduisant ainsi la variance des estimateurs associés. Dans un deuxième temps, nous cherchons sous quelles conditions les prédicteurs non-linéaires sont plus performants que les méthodes linéaires. Les séries temporelles admettant une représentation parcimonieuse en temps-fréquence, comme celles issues de l'audio, réunissent ces conditions, et nous proposons un algorithme de prédiction utilisant une telle représentation. Le dernier problème que nous étudions est la synthèse de signaux audios. Nous proposons une nouvelle méthode de génération reposant sur un réseau de neurones convolutionnel profond, avec une architecture encodeur-décodeur, qui permet de synthétiser de nouveaux signaux réalistes. Contrairement à l'état de l'art, nous exploitons explicitement les propriétés temps-fréquence des sons pour définir un encodeur avec la transformée en scattering, tandis que le décodeur est entraîné pour résoudre un problème inverse dans une métrique adaptée.

ABSTRACT

This dissertation studies unsupervised time-series modeling. We first focus on the problem of linearly predicting future values of a time-series under the assumption of long-range dependencies, which requires to take into account a past of large duration. We introduce a family of causal and foveal wavelets which project past values on a subspace adapted to the problem, thereby reducing the variance of the associated estimators. We then investigate under which conditions non-linear predictors exhibit better performances than linear ones. Time-series which admit a sparse time-frequency representation, such as audio ones, satisfy these requirements, and we propose a prediction algorithm using such a representation. The last problem we tackle is audio time-series synthesis. We propose a new generation method relying on a deep convolutional neural network, with an encoder-decoder architecture, which allows to synthesize new realistic signals. Contrary to state-of-the-art methods, we explicitly use time-frequency properties of sounds to define an encoder with the scattering transform, while the decoder is trained to solve an inverse problem in an adapted metric.

REMERCIEMENTS

En premier lieu, je remercie mon directeur de thèse, Stéphane Mallat, pour le temps et l'énergie qu'il m'a consacrés, et dont j'ai énormément appris. Ses encouragements constants m'ont aidé à me frayer un chemin à travers la jungle tortueuse mais fascinante qu'est la recherche. Son exceptionnelle vision scientifique ainsi que la passion de la recherche qui l'anime continueront longtemps à m'inspirer.

Je remercie également les membres du jury de soutenance. Merci à Emmanuel Dupoux d'avoir accepté de présider cette soutenance qui a donné lieu à une discussion très stimulante. Merci à Emmanuel Vincent et Bruno Torrésani d'avoir accepté de rapporter ce manuscrit et en particulier pour leurs remarques très pertinentes. Merci enfin à Mathieu Lagrange et Gilles Wainrib d'avoir accepté d'être membres du jury et pour les échanges scientifiques que nous avons eus.

Je remercie Sira Ferradans, Sébastien Loustau, Camille Saumard, Aladin Virmaux, Bruno Torrésani et Stéphane Rivaud pour leurs invitations à présenter mes travaux de recherche.

Au cours de ces trois années de thèse, j'ai eu la chance de rencontrer des personnes absolument formidables au sein de l'équipe Data avec qui j'ai passé de très bons moments, qu'il s'agisse de collaborer scientifiquement, de refaire le monde autour d'un verre ou encore d'escalader des blocs : Tomás Angles, Antoine Brochard, Carmine Emanuele Cella, Samuel Chang, Ivan Dokmanić, Michael Eickenberg, Georgios Exarchakis, Sira Ferradans, Ravi Kiran, Roberto Leonarduzzi, Vincent Lostanlen, Chris Miller, Edouard Oyallon, Matthew Ricci, Gaspar Rochette, Alberto Romagnoni, Grégoire Sergeant-Perthuis, Amos Sironi, Louis Thiry, Gilles Wainrib, Irène Waldspurger, John Zarka, Sixin Zhang. Merci à eux. Mention spéciale aux personnes dont j'ai pu partager le bureau. Petit clin d'oeil également aux compagnons de cette aventure pédagogique qu'est le Challenge Data.

Un grand merci à Lise-Marie Bivard, Joëlle Isnard, Sophie Jaudon et Valérie Mongiat pour leur incroyable efficacité administrative, qui est très précieuse dans un monde de chercheurs. Merci également aux membres du SPI, notamment Jacques Beigbeder et Ludovic Ricardou, pour leur grande aide informatique, notamment dans les "deadlines" critiques.

Je remercie les enseignants qui ont stimulé mon goût des sciences et de l'écriture au cours de mes études, m'éveillant ainsi à de nouveaux mondes ; c'est notamment le cas de Frédéric Cuvellier, Thierry Meyer, Béatrice Stoll et Denis Hirson.

Pour leur relecture attentive du manuscrit malgré la canicule, un grand merci à John, Gaspar, et surtout Léopold.

Je remercie vivement mes amis pour les très bons moments que nous avons partagés, ainsi que pour leur soutien et leur affection. Merci tout d'abord à mes co-châtelains, Stanislas puis Julia, et à leurs associés, Anouk et Naoufal, ainsi qu'au noyau des nageurs, Antoine, Marie, Mathieu, Léopold et Louise. Merci ensuite aux glorieux anciens : ceux de Kléber, Jonas et Pierre-Édouard ; ceux du MVA, notamment Michel et Matthias. Enfin, merci au groupe des Alsaciens, en particulier Benjamin et Arnaud.

À tous les stades de mes longues études, j'ai eu la chance de pouvoir compter sur ma famille. Merci tout d'abord à mes parents, Luc et Patricia, pour m'avoir transmis le goût de l'effort, pour leur soutien de tous les jours, et surtout leur amour inconditionnel. Merci également à mes soeurs, Claire et Pénélope, ainsi qu'à leurs familles respectives, Antoine, Manu, Éléonore et Arthur : leurs encouragements m'ont grandement aidé dans les moments les plus difficiles. Merci enfin à mes grands-parents, tantes, oncles, cousines et cousins pour leur présence bienveillante.

Enfin, j'ai rencontré au cours de ce doctorat une personne qui a illuminé ma vie. Pour son soutien et son amour, merci à Juliette, sans qui rien n'aurait été possible.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

This dissertation falls into the realm of unsupervised learning. Unsupervised learning aims at inferring a model of the data studied from examples.

The data we consider consists in univariate real-valued time-series x of length T, sampled at discrete times $x = \{x(t)\}_{1 \leq t \leq T}$. For instance, such time-series may correspond to financial recordings (the daily valuations of a stock), audio signals (music) or astrophysical observations (the number of dark spots observed on the sun each month).

For each general category of time-series, we assume the existence of an underlying stochastic process X generating the data, so that each observation x is a realization of X. We also postulate that the probability distribution of X admits a probability density function p_X with respect to the Lebesgue measure. However, this probability density function p_X is unknown.

Modeling the time-series x consists in building an estimator $\widetilde{p}_X$ of the true probability density function p_X. The estimator is obtained with an algorithm which uses a finite dataset of observations $\{x_i\}_{1 \leq i \leq N}$. The choice of the estimating algorithm incorporates prior assumptions made by the modeler on the density $\widetilde{p}_X$.

Many different tasks and applications are encompassed by this framework. Among them, time-series generation is an important one: it corresponds to sampling the original probability distribution. The resulting additional samples can then be used *e.g.* for data augmentation in a semi-supervised learning setting. Unsupervised learning for time-series also encompasses forecasting. Indeed, predicting future values of a time-series corresponds to estimating the conditional probability density $p_X(x(t+\Delta)|x(\leq t))$. We can also mention the anomaly detection problem, which consists in detecting outliers among a population. This requires to approximate the probability density function of the data points and identify samples with low probability.

The probability density estimation problem raises two fundamental questions. The first one deals with the ability to estimate p_X. What are the implications of different assumptions, or priors, for p_X regarding the quality of the estimator $\widetilde{p}_X$? To answer this question, one needs to create adapted notions of distance between the estimate $\widetilde{p}_X$ and the actual density p_X, in order to measure the quality of the estimation. The optimal sample complexity is a key issue: how many samples are required to reach some error level in the worst case? The second question concerns the actual algorithm used to perform the estimation. Which algorithms are suitable to estimate a probability distribution from samples? What is the runtime which is required to attain an error close to the optimal error given those samples?

In this chapter, we provide a short introduction to standard assumptions for the probability density function, such as stationarity, Gaussianity and Markov dependencies. These assumptions simplify the estimation problem. We then review the challenges to model probability distributions with less restricting priors. This allows us to introduce and put in perspective the contributions exposed in this work.

1.1 STANDARD PRIORS FOR p_X

We stress the difficulty of the estimation problem, which faces the curse of dimensionality, and review standard hypotheses which allow to make the problem tractable.

1.1.1 *Curse of dimensionality*

The probability density estimation problem for an arbitrary process X defined in a vector space of large dimension T is intractable due to the so-called curse of dimensionality: the number of samples to reach a given precision grows exponentially with the ambient dimension. We provide a simple example of this major difficulty with multivariate kernel density estimators, following [Här+04; Pow06].

Let us assume that p_X is three times differentiable and that we have access to N samples $\{x_i\}_{1\leq i\leq N}$. Let $\mathcal{K}$ be a kernel function of $L^1(\mathbb{R}^T)\cap L^2(\mathbb{R}^T)$, summing to one and whose first-order polynomial moment is zero. The kernel density estimator of bandwidth $h>0$ induced by $\mathcal{K}$ is defined as:

$$\widetilde{p}_X(x) = \frac{1}{Nh^T}\sum_{i=1}^{N}\mathcal{K}\left(\frac{x-x_i}{h}\right). \tag{1.1}$$

In other words, this estimator is the counting measure made of Dirac delta functions located at x_i smoothed by the kernel $\mathcal{K}$ dilated by h. A simple yet popular choice of kernel is the Gaussian isotropic one: $\mathcal{K}(x) = \kappa e^{-\|x\|^2/2}$, where $\kappa > 0$ is a normalizing constant. Insofar as $\int_u \mathcal{K}(u)du = 1$, $\widetilde{p}_X$ is a probability distribution summing to 1 over its domain.

As shown in [Här+04], it is possible to lower-bound the pointwise mean-square error $\mathbb{E}|\widetilde{p}_X(x) - p_X(x)|^2$. This can be done by decomposing the mean-square error into a bias and a variance term, and getting an asymptotic behavior for small values of h with a second-order Taylor expansion of p_X. By optimizing the sum of bias and variance with respect to h, one obtains the following lower-bound on the mean-square error:

$$\mathbb{E}|\widetilde{p}_X(x) - p_X(x)|^2 \geq C \left(\frac{1}{n}\right)^{4/(4+T)} , \tag{1.2}$$

where $C > 0$ is a constant depending on $\mathcal{K}$. From Equation (1.2), we see that in order to achieve an error level of ϵ, one needs $n(\epsilon)$ samples where

$$n(\epsilon) = O\left(\left(\frac{1}{\epsilon}\right)^{1+T/4}\right) . \tag{1.3}$$

The exponential dependency on the ambient dimension T makes it impossible to achieve this bound. For audio time-series, $T = 10^3$ is a small value, as it corresponds to 40 ms if the sampling rate is 22050 Hz. If one wants to achieve an error $\epsilon = 10^{-1}$, which is not very small with respect to typical values of $p_X(x)$, one would theoretically need about $n(10^{-1}) \approx 10^{251}$ independent samples. Since the estimated number of atoms in the observable universe is about 10^{90} [Gut03], it is completely impossible to have access to that many samples.

This simple result can be extended to more complex estimators, where the same problem appears [Här+04]. This so-called curse of dimensionality is fundamentally due to the fact that in high dimension, all samples are isolated from each other in general. It is necessary to make stronger assumptions on p_X in order to reduce the dimension on which it is defined.

1.1.2 *Stationarity*

Stationarity is an hypothesis of translation invariance of the probability distribution [BD91]. In its stronger mathematical formulation, it requires that the probability density functions of the original process $\{X(t)\}_t$ and of its translated version $\{X(t-1)\}_t$ are equal[1]. This means that for all time-series x,

$$p_X(\{x(t)\}_t) = p_X(\{x(t-1)\}_t) . \tag{1.4}$$

By recurrence and symmetry, Equation (1.4) will hold for all translated versions of x. There are weaker formulations of stationarity, for instance by requiring only first- and second-order moments to be invariant by translation. This latter formulation is useful in the Gaussian case, which is exposed in the next subsection.

The stationarity assumption makes it possible to dramatically augment the number of samples which are available. Indeed, each observed time-series x_i yields its translations $x_i(\cdot - u)$ for all possible steps u. However, this does not necessarily simplify the estimation

[1] We ignore boundary issues in the definition of this translation on $\mathbb{R}^T$, which is seen as a numerical restriction of $\mathbb{R}^{\mathbb{Z}}$.

problem, as these samples may be highly dependent. For instance, consider the toy process defined by $X(t) = B$ for all t, where B is a Bernoulli variable. This process is clearly stationary, but all its translations are perfectly correlated with itself so one realization of the time-series is not sufficient to estimate its probability density. The stationarity prior becomes more effective when used in conjunction with other assumptions on the probability distribution.

1.1.3 *Strong assumptions: Gaussian distribution and short-term dependence*

We now review two standard assumptions on stochastic processes which simplify the estimation of their probability distribution: the Gaussian prior and the existence of short-term, or Markovian, dependencies [BD91].

The Gaussian prior assumes that the stochastic process X is Gaussian, so that the probability density function p_X follows a multivariate Gaussian distribution. As a consequence, p_X is entirely characterized by the first-order moments $\mathbb{E}[X(t)]$ and second-order moments $\mathbb{E}[X(t)X(s)]$ for all t, s. Estimating p_X is thus simpler, as it is sufficient to estimate this restricted number of moments, which grows as $O(T^2)$. In the general case, one would have needed to estimate all higher-order moments $\mathbb{E}[X(t_1) \cdots X(t_k)]$ for $k > 2$ and all $t_1, \ldots, t_k$, whose number grows as $O(2^T)$.

Under the stationarity assumption, the first- and second-order moments admit a much simpler structure [BD91]. Indeed, because of the translation invariance, all $X(t)$ share the same mean, which is called the mean of the process. Up to a subtraction by the square of the mean, the second-order moments are characterized by the autocovariance function $\gamma_X(t) = \mathrm{Cov}(X(s), X(s+t))$, where s is a shadow variable. As a consequence, there are only $O(T)$ numbers to estimate in order to characterize p_X entirely, which greatly simplifies the problem.

The other important assumption which is often used in practice is the short-term dependence assumption, or Markov dependence assumption. This assumption posits that the variables $X(t)$ and $X(s)$ become less and less dependent as $|t - s|$ grows. Depending on the framework, this decreasing dependence can be formulated in different fashions. In the strict Markov sense, there exists a dependence horizon $\tau \ll T$ such that $X(t)$ is only conditionally dependent on $X(t-\tau), \ldots, X(t-1)$ and not on previous values:

$$p_X[x(t)|x(\leq t)] = p_X[x(t)|x(t-s), 0 \leq s \leq \tau] \tag{1.5}$$

If we further assume stationarity, there are two very important benefits for the estimation of p_X. First, in order to estimate p_X as a whole, it is sufficient to estimate $p_X(x(t-\tau), \ldots, x(t))$, where t is a shadow variable, which is defined in a vector space of much lower dimension than T. Second, a single time-series yields about $\lfloor T/\tau \rfloor$ independent samples.

In the Gaussian stationary case, it is possible to reformulate the Gaussian dependence with the autocovariance function γ_X. The new formulation assumes that correlations decrease exponentially quickly: there exists $v > 0$ such that

$$\gamma_X(t) = O(e^{-|t|/v}) \ . \tag{1.6}$$

As a consequence, dependencies admit a typical time-scale of $\tau = 5v$ beyond which correlations are negligible, so that the related variables are almost independent. This short-term dependence assumption provides similar benefits to the density estimation problem as the more general formulation [BD91].

Many algorithms exploit the Gaussian and short-term dependencies hypotheses. Among them, the autoregressive moving-average (ARMA) models are very important: they allow to parametrize such distributions with very few coefficients. Further, these models allow to linearly predict future values of the time-series. In the Gaussian case, linear predictors are optimal in terms of mean-square error.

1.2 LOOKING FOR NEW PRIORS: CHALLENGES

The assumptions introduced in the previous section are sufficiently restrictive to allow the estimation of models, but limits their expressiveness. In this section, we stress the need for different priors which will allow to describe time-series with different properties, notably non-Gaussian distributions and long-range dependencies. Among these assumptions, the hypothesis induced by deep generative models allow to reproduce these characteristics. A major challenge lies in understanding what these deep priors imply for the modeled processes.

1.2.1 *Non-Gaussian distribution*

Many time-series of interest seem to follow a distribution for which the Gaussian assumption is ill-suited. We provide two examples of such series, stress the difficulties raised by these assumptions and review solutions which have been proposed to model these behaviors.

In finance, the well-documented phenomenon of "fat tails" is evidence of a deviation from the Gaussian model [BP09, Chapter 6]. Assume that X represents the daily returns of a given stock valuation, X being stationary. We consider the marginal probability distribution of $X(t)$. According to the Gaussian prior, $X(t)$ should follow a Gaussian distribution. Yet, empirical measurements show that extreme values are much more frequent than authorized by a Gaussian

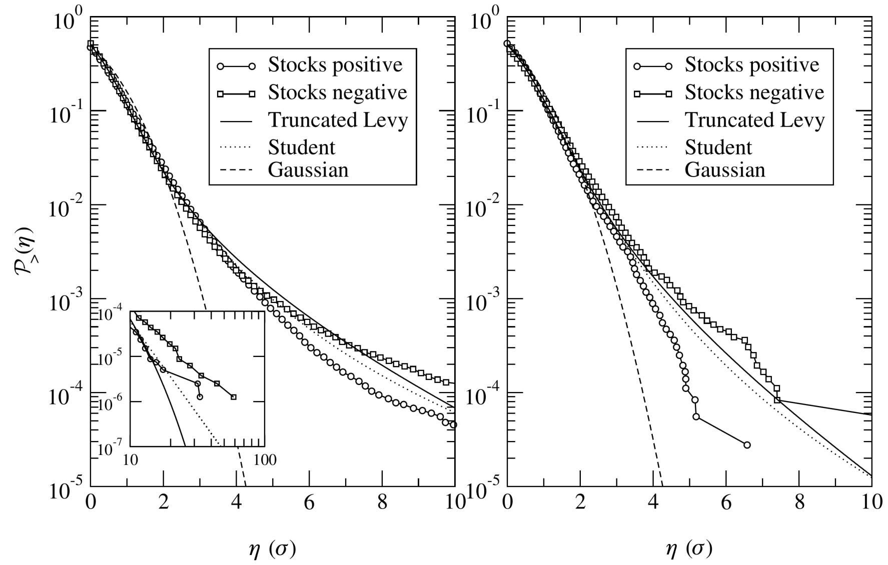

Figure 1.1: Fat tails in the daily (left) or monthly (right) distributions of returns in some US stocks. The horizontal axis represents thresholds η and the vertical axis represents the corresponding tails $\mathbb{P}\left[|X(t)| > \eta\right]$. Figure extracted from [BP09].

distribution, as displayed in Figure 1.1. The tails $\eta \mapsto \mathbb{P}[|X(t)| > \eta]$ decrease extremely slowly compared to the expected Gaussian behavior, typically following a power-law, hence the "fat-tail" denomination.

Many natural audio signals have characteristics which can hardly be modeled by Gaussian variables. In particular, due to physical constraints on the way they are produced, harmonic sounds such as speech or music admit sparse time-frequency decompositions: there exists a linear transform W such that for all $x \sim X$, Wx has very few non-zero coefficients [RS78; GME11]. This phenomenon is depicted in Figure 1.2 with a windowed Fourier transform. This sparsity is not compatible with a Gaussian prior, as it also leads to fat tails for the marginal probability distributions of each coefficient [Vin07]. Further, the resulting decompositions exhibit very particular structures. The non-zero coefficients tend to be activated in groups which form some patterns, such as formants for speech, attacks in music, etc [Mal08; MT05]. These very structured patterns are another evidence of a non-Gaussian behavior for these time-series.

As we have stressed in the previous section, it is very difficult to estimate non-Gaussian densities without other hypothesis. It is necessary to use assumptions which reduce the dimensionality of the problem in order to derive tractable estimators.

Several restrictive assumptions have been proposed to model these specific non-Gaussian behaviors. In finance, multiple closed-form models have been proposed so as to reproduce the empirical characteristics of the series. For instance, Lévy processes allow to model time-series with exponential tails instead of Gaussian ones [BP09; App04]. In

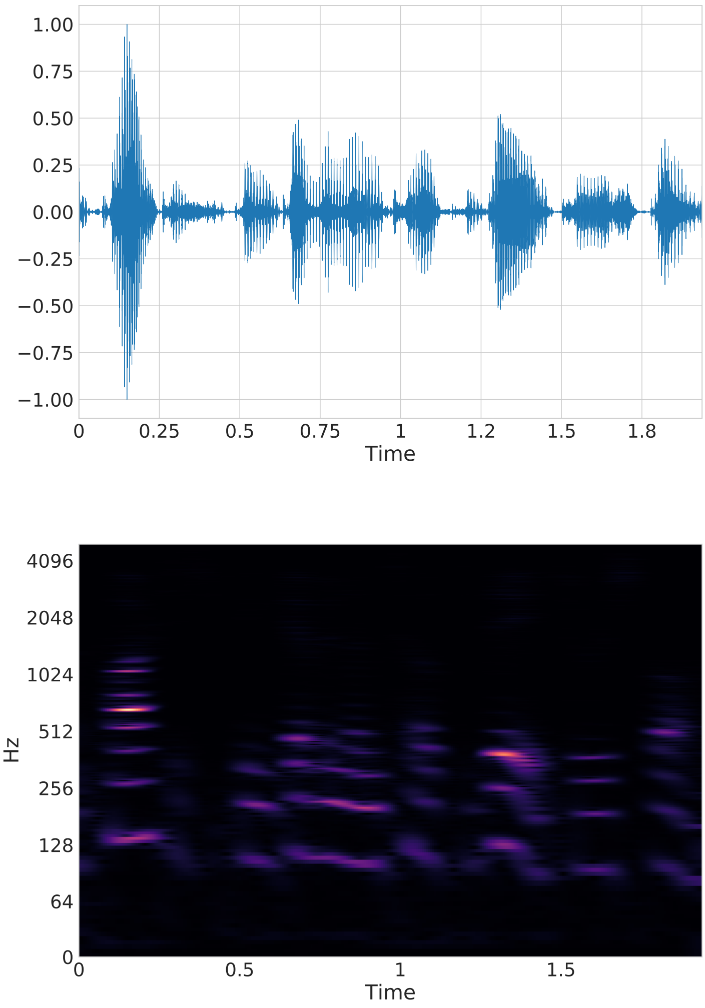

Figure 1.2: Speech signal from the VCTK dataset [Yam12]. **Top**: Waveform. **Bottom**: Corresponding spectrogram (black values are small). The signal is sparse in the time-frequency domain.

order to model the sudden jumps in the series, specific point processes such as Poisson [App04] or Hawkes processes [BMM15] have been introduced. These parametric models typically have few parameters which can be estimated from data. However, this lack of parameters restricts their expressivity.

In the case of music, closed-form parametric models relying on time-frequency decompositions have been proposed, see *e.g.* [MT05; KT06]. These models were designed to reproduce the characteristic patterns of speech and music signals, using Hidden Markov Models in order to create groups of activations. These models allow to derive tractable estimators, but they lack expressivity due to the small numbers of parameters they use.

In order to model non-Gaussian probability distributions in a non-parametric fashion, a generalized method of moments has recently been introduced [Bru+15]. It relies on the estimation of iterated wavelet transform moments. Contrary to polynomial moments, which characterize probability densities but suffer from variance explosion, these moments rely on contractive operators which prevent such an explosion [BM13a]. The probability distribution is then estimated as the maximum entropy distribution constrained by the estimated moments. It enables the capture of non-Gaussian behaviors such as intermittency and time-frequency sparsity.

1.2.2 *Long-range dependencies*

There is evidence that the short-term dependence assumption does not hold in many domains of interest.

Long-range dependent time-series were first observed in hydrology [Gra+17]. By looking at recordings of floods from the Nile river, Hurst [Hur51] observed very long correlations between past and current values. This phenomenon was then recognized in many different fields, be it finance or telecommunications [DOT03; Gra+17]. These time-series are modeled by Gaussian stationary processes X whose autocovariance γ_X has a power-law behavior:

$$\gamma_X(t) = O(|t|^{-\beta}) \; , \tag{1.7}$$

for $\beta > 0$. Contrary to short-term dependent processes, long-range dependent processes do not have a typical time-scale beyond which correlations can be neglected. In fact, these processes exhibit a scale-invariant behavior: a downsampling does not change the rate β at which the autocovariance decreases. As a consequence, correlations between $X(t)$ and $X(s)$ decrease slowly and neglecting them leads to large errors, even at long temporal intervals.

Some processes with a non-Gaussian distribution also exhibit long-range dependencies. This is notably the case of music signals, even though the exact nature of these dependencies is more difficult to formalize. Music signals are roughly defined according to a hierarchical structure which has at least three layers: the overall composition, individual notes, and the waveform $\{x(t)\}_t$ [LJ83; Cel11; Koe+13]. Due to choices of the composer, notably the use of repetitive patterns, the composition creates strong dependencies between notes at very long intervals, see *e.g.* [JPH07]. In turn, individual notes and corresponding portions of the waveform are extremely dependent. The formalization of this dependence is still an intense research topic [PMK10], but it is clear that the relationship involves more than mere second-order correlations. Overall, the hierarchical structure of music creates very long dependencies at the scale of the waveform, such that $X(t)$ and $X(s)$ are not independent even when $|t - s|$ is of the order of T. The same line of reasoning can be applied to speech signals to show the existence of long-range dependencies.

Both in the Gaussian and non-Gaussian cases, long-range dependencies increase the dimension of the domain of the probability density function, as it does not factorize into smaller terms. Estimating the problem's probability density function thus becomes much more challenging.

In the Gaussian case, the assumption of a power-law behavior for the autocovariance γ_X allows the derivation of tractable estimators through the wavelet transform. Indeed, under this assumption, which may be reformulated as a scale invariance, wavelet coefficients of a long-range dependent process have a short-term dependence behavior along the temporal axis [DOT03]. This short-term dependence allows the construction of tractable estimators of the decay parameter β, which parametrizes the probability distribution [AVF98].

In the non-Gaussian case, the assumption of a hierarchical structure allows the capture of some part of the long-range dependencies. For instance, it is possible to assume Markov dependencies over frames of speech signals [GME11]. The resulting Hidden Markov Models allow to capture part of the long-range dependencies. Refining the priors to cope with longer dependencies is still an active research topic [Ron+16; Per+18].

1.2.3 *Deep neural network priors*

Deep neural networks [LBH15] have recently provided very realistic models $\widetilde{p}_X$ for signals which are non-Gaussian and have long-range dependencies, such as speech or music [Oor+16; Meh+17]. However, the explicit assumptions made by these models are not enough to explain their success [Zha+17; Aro+17]. One of the current challenges in unsupervised learning lies in understanding what the implicit assumptions made in the design of these models, notably the choice of architecture, imply for the underlying processes.

A deep neural network is a parametric function $f_\theta : \mathbb{R}^d \mapsto \mathbb{R}^l$ between two vector spaces [GBC16]. In its most basic form, it can be described as a succession of affine mappings W_j followed by non-linearities ρ applied independently on each coordinate of input vectors:

$$f_\theta(v) = \rho\left(W_J \rho\left(\ldots \rho(W_1 v)\right)\right) \ . \tag{1.8}$$

The function f_θ is parametrized by θ, which is the collection of all the mappings $W_1, \ldots, W_J$. The number of parameters is typically larger than 10^6. The set of the networks $\{f_\theta\}_\theta$ is called an architecture: it only depends on the size of each layer and on choice of the types of connections and non-linearities between these layers. There are multiple variants or restrictions of the generic model (1.8). Among them, networks which only employ convolutional linear mappings [LB98] are of chief importance: they have allowed breakthroughs in both supervised [KSH12; AH+14] and unsupervised learning [RMC16; Oor+16].

Autoregressive neural networks [LM11] use neural networks to build a model $\widetilde{p}_X$. They use two explicit assumptions: stationarity, and Markov dependencies with a past size τ. This leads to a model of the form:

$$\widetilde{p}_X(x) = \prod_{t=1}^{T} \underbrace{\widetilde{p}_X(x(t)|x(t-\tau),\ldots,x(t-1))}_{f_\theta(x(t-\tau),\ldots,x(t))} \tag{1.9}$$

The parametrized function f_θ is a neural network estimating the conditional probability distributions, under the stationarity and Markov assumptions. It is usually chosen as a convolutional neural network [LB98] or as a recurrent one [HS97]. It can be fitted by maximizing the log-likelihood of the model over training data.

Autoregressive networks have lead to outstanding audio generation results, notably with the WaveNet [Oor+16; Oor+17] and SampleRNN [Meh+17] networks. Incorporated into larger hierarchical models, these networks have increased the state-of-the-art in text-to-speech applications. The same idea has also been applied to image [OKK16] and video [Kal+16] generation, with an equal success: it appears to be a powerful generic method to model non-Gaussian signals.

Deep neural networks have also been used to implicitly model the probability density function p_X, as depicted in Figure 1.3. These implicit models rely on a so-called latent space $\mathcal{Z}$ and an application

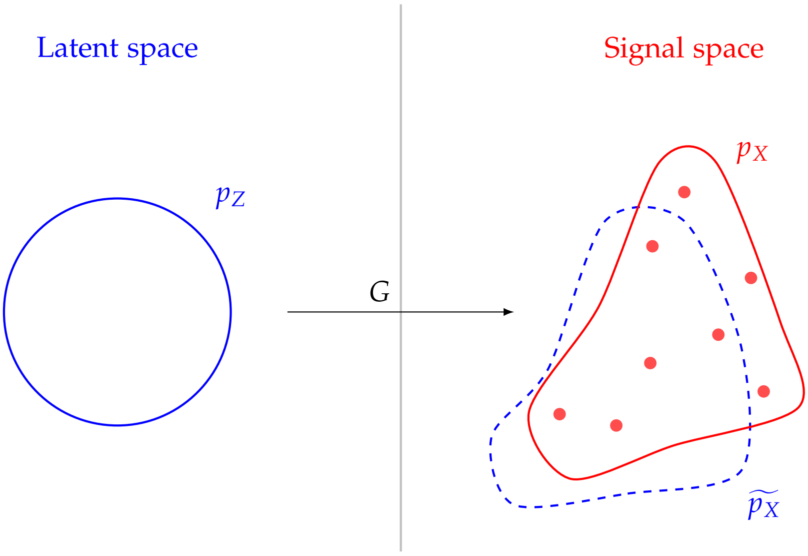

Figure 1.3: Implicit probability density modeling. A known probability distribution p_Z is mapped through G to a signal distribution $\widetilde{p}_X$ (1.10) approaching the true probability distribution p_X, only known from samples x_i.

$G : \mathcal{Z} \to \mathbb{R}^T$ mapping latent vectors z to signals x. A known distribution p_Z is assumed on the latent vectors and is push-forwarded by G on the signals, so that

$$\widetilde{p}_X = G_* p_Z. \tag{1.10}$$

The mapping G is usually chosen as a deep convolutional neural network. Generative Adversarial Networks (GANs) [Goo+14] and Variational Auto-Encoders (VAEs) [KW14] are two instances of such implicit generative networks, even though their training methods radically differ. Deep implicit generative networks have achieved very impressive results for image generation, and their improvement is an active topic of research [RMC16; ACB17; Tol+18]. In particular, these models are able to factorize the variabilities of images, so that linear interpolations in the latent space result in meaningful interpolations in the signal space.

The empirical success of deep generative networks in modeling non-Gaussian processes with long-range dependencies is not well understood [Aro+17; ARZ18]. Indeed, the hypotheses explicitely claimed by these models are insufficient to explain their success. In the case of autoregressive neural networks for instance, the past size S which is used is typically large, $S \sim 10^3$. As shown in the previous section, the probability density estimation is subject to the curse of dimensionality with such a large past value. In the case of GANs, no explicit assumption is done on the underlying signal.

As a consequence, there must be implicit priors in the definition of deep generative models which allow to circumvent the curse of dimensionality. Among these assumptions, the choices of architecture and training method are particularly important [GBC16]. One of the main challenges raised by these models is to understand what the modeler's assumptions on the signal are when making the choice of a deep neural architecture, and how these assumptions make the estimation problem tractable. The difficulty lies in the large complexity of these architectures, which involves tens of layers and millions of parameters, along with many tricks and hooks whose importance is not well understood.

1.3 CONTRIBUTIONS

This dissertation investigates three questions related to the probability density estimation problem with challenging characteristics.

In Chapter 2, we first tackle the problem of time-series forecasting in the Gaussian case, under long-range dependencies assumptions. We show that one can exploit priors on the autosimilarity of the autocovariance function in order to make the estimation tractable, by projecting the past of the time-series on adapted subspaces.

We then look for assumptions on signals which allow non-linear forecasting algorithms to beat linear forecasting methods. Time-frequency sparsity is identified as such a prior: we propose in Chapter 3 an algorithm exploiting this property and relate it to neural networks with one hidden layer.

Chapter 4 investigates the problem of generating time-series under the assumption of non-Gaussian sparse time-frequency distributions with long-range dependencies. We show that these assumptions allow to explicitly design a deep generative network tackling the problem in a tractable fashion.

Chapter 5 concludes this dissertations and discusses future perspectives.

1.3.1 *Foveal wavelets for long-range dependencies in the Gaussian case*

In Chapter 2, we first consider the problem of forecasting a stationary centered Gaussian process X. Given a future lag $\Delta > 0$, this boils down to the estimation of the conditional probability density $p_X[x(t+\Delta)|X(\leq t) = x(\leq t)]$. By stationarity, to simplify notations, we set the variable t at 0 in all subsequent equations.

Insofar as the time-series is stationary and Gaussian, it is sufficient to estimate the conditional mean $\mathbb{E}[X(\Delta)|X(\leq 0)]$. Further, linear estimators of this conditional mean are optimal in the mean-square error sense. Therefore, the forecasting problem consists in building a linear estimator $\widetilde{X}(\Delta) = \alpha^T X(\leq 0)$, where α is a vector. The quality of this estimator is measured by the mean-square error $\mathbb{E}|X(\Delta) - \widetilde{X}(\Delta)|^2$.

Under short-range dependence assumptions, the whole past $X(\leq 0)$ can be replaced with negligible error by the vector $[X(-S), \dots, X(0)]$ of size $S+1$, where S is the typical length of correlations in the time-series. Provided that the time-series is much larger than S or that one has access to multiple independent realizations of the process, the estimation problem becomes tractable.

Instead, we assume that the process X has long-range dependencies: there exists an exponent $\beta > 0$ such that the autocovariance $\gamma_X(t) = \mathrm{Cov}(X(0), X(t))$ behaves as a power-law at large times: $\gamma_X(t) = O(|t|^{-\beta})$. As a consequence, one cannot neglect correlations at long intervals. The fractional Gaussian Noise (fGN) is a well-studied closed-form model for these time-series [DOT03]. In this context, we investigate which linear representation of the past $X(\leq 0)$ allows us to make the estimation problem tractable while leading to low forecasting error.

Previous works have shown the adequation of wavelet representations for long-range dependent processes [AVF98]. A wavelet family $\{\psi_{j,n}\}_{j,n}$ is defined by the dilatations and translations of a single mother wavelet ψ of 0 mean:

$$\psi_{j,n}(t) = 2^{-j/2}\psi(2^{-j}t - n) \; . \tag{1.11}$$

Thanks to their vanishing polynomial moments, wavelets transform long-range dependences into short-range ones [DOT03]. It is therefore tempting to use wavelets to represent the past $X(\leq 0)$ as $\langle X(\leq 0), \psi_{j,n} \rangle$. However, save for the Haar wavelets, standard wavelets are smooth and the values at the borders of their support very close to 0. Since wavelets are not allowed to access future values, a wavelet representation of the past would lose the values closest to present $X(0), X(-1), \ldots$ Insofar as these values have the largest correlation with the value to predict, this leads to poor forecasting results.

The case of Haar wavelets reveals a key property of long-range dependent processes. The covariance between the target value $X(\Delta)$ and past values $X(\leq 0)$ admits a sparse representation in the Haar basis. The most important coefficients are those related to wavelets closest to the temporal singularity $X(0)$, at all scales. This is a consequence of the autosimilarity of the underlying stochastic process: a downsampling $X_{\downarrow}(t) = X(2t)$ does not change the exponent β such that $\gamma_{X_{\downarrow}}(t) = O(|t|^{-\beta})$.

We call the set of Haar wavelets closest to the present the foveal Haar family. Indeed, this family spans a set of functions with a resolution decreasing exponentially as one moves away from the present value. Such a behavior is observed in the retina, where the density of cones decreases exponentially around the center, called fovea, hence this name [Pur+01].

Our main contribution in Chapter 2 consists in introducing a new family of wavelets which are more flexible than the foveal Haar family: the foveal wavelets. These wavelets are inspired by a previous wavelet construction [Mal03], but they are simpler in their design and easier to implement, at the expense of orthogonality. These foveal wavelets are defined by the dilations of a family of functions at a single scale $\{\psi_0^m\}_m$:

$$\psi_j^m = 2^{-j/2} \psi_0^m (2^{-j} \cdot) \tag{1.12}$$

The family ψ_0^m consists in polynomials modulated by a fixed causal window. By construction, the set of the foveal wavelets at all scales $\{\psi_j^m\}_{j,m \leq M}$ captures the most important coefficients in the covariance structure of the process X, thereby leveraging the long-range dependent prior. In order to match complex covariance structures, it is possible to increase the number M of functions at each scale, thereby filling a larger space while keeping a similar temporal support.

Numerical results on synthetic and real time-series demonstrate an improvement in the estimation accuracy with respect to the linear baseline. Indeed, the approximation error of foveal wavelets is almost unchanged, but its estimation error reduces because there are fewer coefficients to estimate.

1.3.2 *Prediction of sparse time-frequency processes*

In Chapter 3, the problem of forecasting stationary time-series is considered. In the Gaussian case, on which focus has been given in the previous chapter, it is sufficient to consider linear estimators of future values. In many cases, it turns out that linear estimators are robust baselines very difficult to improve significantly. This is notably the case in finance, despite the non-Gaussian behaviors mentioned in 1.2.1. It is therefore legitimate to investigate which priors allow non-linear predictors to achieve a prediction error substantially lower than linear predictors.

In signal processing, the assumption of a sparse decomposition of the signal has allowed non-linear algorithms to improve over linear ones in many tasks, for instance denoising [Fev+08] or inpainting [Adl+12]. This sparse prior is empirically valid for audio signals with time-frequency dictionaries [Mal08].

Neural networks have recently attained state-of-the-art performances in modeling audio signals. The best performing networks, such as WaveNet [Oor+16], use an autoregressive formulation which focuses on modeling $p_X(x(t+1)|X(\leq t) = x(\leq t))$. Due to the sheer size and complexity of their architectures, understanding the operations performed by these networks is challenging.

In order to simplify these architectures, we focus on the simplest non-linear neural network, a multi-layer perceptron (MLP) with one hidden layer [GBC16]. We empirically show that this neural network is able to perform better than linear predictors when forecasting processes which are sparse in time-frequency dictionaries.

Based on this observation, we investigate how the sparse time-frequency prior can be harnessed for forecasting in a principled fashion. A detailed analysis of the weights of the MLP reveals that it actually performs a time-frequency decomposition of the past $X(\leq t)$ on time-frequency atoms, and recombines the coefficients of this decomposition for forecasting.

Using a simple cosine model, we explain this fact mathematically. A simple method to forecast a time-series made of a cosine function oscillating at a fixed frequency ω consists in extracting the local phase $\omega t + \phi$ and increasing the phase by $\omega\Delta$ in order to reach the adequate value: this last operation can be performed linearly by using complex numbers. In order to accurately estimate the local phase, one must take into account the causality constraint and therefore window the past. For a time-series containing a few frequential components, one can use this idea to obtain a non-parametric forecasting method. It consists in first decomposing the time-series onto a time-frequency dictionary, then using the resulting components which contain the local phases to get a forecast of each component, and eventually summing them to obtain a prediction by linearity.

We introduce a non-linear forecasting algorithm relying on this idea, with adapted time-frequency dictionaries, to predict sparse time-frequency processes. It is extended in a multiscale fashion, using multiple windows of the past, in order to accommodate processes which are more complex than a pure cosine function. The resulting predictor resembles a multi-layer perceptron, but it can be well understood. Numerical results on synthetic and real data show that this method reaches similar results as the perceptron, which validates our simplification.

1.3.3 *Sparse time-frequency time-series generation*

We consider the problem of audio time-series generation, which consists in building an estimate $\widetilde{p}_X$ of the probability density function p_X. The time-series we consider follow a non-Gaussian distribution, as they admit sparse time-frequency distributions, and have long-range dependencies, as explained in the previous section. Without any additional priors, this estimation problem is extremely difficult, due to the large dimension of the space on which the process is defined. Deep neural networks provide outstanding generation results to this problem [Oor+16; Meh+17], but it is unclear which assumptions on the data they actually make to circumvent the curse of dimensionality. Chapter 4 investigates which priors on audio signals can be used so as to estimate p_X in a tractable fashion.

The solution we introduce adapts the framework proposed for images in [AM18b]. We propose an implicit generative model obtained by sampling a random latent variable Z with a white Gaussian distribution p_Z, and applying a non-linear transform G such that $\tilde{X} = G(Z)$ is a generated signal. Mathematically, the modeled probability density $\widetilde{p}_X$ is the push-forward of the latent probability density p_Z through the generator G: $\widetilde{p}_X = G_* p_Z$. We use an autoencoder architecture which relates the latent variable Z to the original signal X with an encoder Φ.

Since both Z and X are defined on large-dimensional vector spaces, the curse of dimensionality prevents the training of the encoder and the decoder to match the respective probability densities p_Z and p_X. Instead, we use priors on the signals to define a fixed encoder such that the signals X are mapped to approximately Gaussian white noise variables $\Phi(X)$. The existence of a sparse time-frequency decomposition, perceptual stability to time-frequency deformations and decorrelations at long scales are priors which are used for this purpose. The resulting encoder consists in a scattering transform, followed by a whitening operator which removes the covariance structure.

The generator G we propose is a deep causal convolutional neural network. In order to avoid using probability distributions, G is trained to invert the encoder on examples. This is a difficult problem for a network with a finite number of neurons. We propose to use an adequate scattering norm to solve this inverse problem, in order to focus on the content which is perceptually important. Further, in order to control the behavior of the network at generation time, we introduce a moment-matching loss which loosely controls the scattering moments of the generated samples.

The resulting deep generative network G is able to map latent Gaussian variables Z to new realistic signals of speech and music. We show that by crafting the encoder to be stable to time-frequency deformations, the resulting generator can transform low-level attributes of music, such as pitch, with a simple linear interpolation in the latent space. The samples do not reach the quality of state-of-the-art approaches, but these results pave the way for a simpler approach to sample complex probability distributions in a tractable and explicit fashion.

2 FOVEAL WAVELET LINEAR PREDICTION

In this dissertation, the first topic we tackle is time-series forecasting. In particular, this chapter is concerned with linear autoregressive time-series prediction, with a focus on long-range dependent time-series.

Long-range dependent time-series are time-series such that the correlation between two points decreases slowly, typically as a power-law, with respect to the temporal gap between these points. As such, it implies that one should take a look at a very large past in order to make a precise prediction of future values. However, in the context of a finite time-series, increasing the past size also leads to an increase in variance. It is thus necessary to consider a representation of the past of the time-series to circumvent these problems.

In this chapter, we investigate how to represent long-range dependent time-series for forecasting applications.

Linear autoregressive forecasting methods consider a past size truncated up to a certain point. This basic representation is suitable for many applications with short-range dependencies. On the contrary, in the case of long-range dependencies, multiple works have studied better suited representations. In particular, wavelet decompositions have emerged as an adequate representation for these series [DOT03]. Thanks to their vanishing moments, wavelets reduce long-range dependent processes to short-term dependent processes, thereby simplifying estimation procedures. However, these representations are not directly amenable to forecasting applications due to causality constraints.

Despite the apparent simplicity of the topic, this problem reveals simple principles which structure the problem: causality and foveality. Causality refers to the fact that we cannot access future values to perform the prediction, thus leading to a "singularity" at the present. Combined with the auto-similar behavior of long-range dependent time-series, this singularity will lead us to define a foveal representation, *i.e.* a multiscale representation centered around this point whose resolution diminishes as the distance from the singularity increases.

We can summarize the contributions of this chapter as follows. We introduce a general class of representations of the past of the time-series, the foveal wavelets. These foveal wavelets are adapted to the causality and long-range dependent constraints, and are thus suitable for forecasting applications.

This chapter is organized in four sections. In Section 2.1 we review and formalize the linear prediction task in a given representation for a long-range dependent process. Section 2.2 we review the wavelet tool used to represent long-range dependent processes, and highlight their limitations for forecasting applications. In Section 2.3, we develop the representation proposed, namely foveal wavelets. Section 2.4 experimentally validates the use of this representation for forecasting applications.

2.1 LINEAR FORECASTING

In this section, we formalize the notion of the representation of time-series in the linear forecasting case. Building on standard time-series and supervised learning tools, notably the Yule-Walker equations [BD91] and the bias-variance trade-off [HTF09], we express a criterion which measures how good a subspace is to represent the past of a time-series for forecasting.

2.1.1 *Linear estimation in a basis*

2.1.1.1 *Framework*

GAUSSIAN STOCHASTIC PROCESS Let $(X(t))_{t\in\mathbb{Z}}$ be a discrete real-valued stochastic process. We assume that X is stationary, has zero mean, and has a Gaussian distribution. Let us denote its autocovariance function as

$$\gamma_X(u) := \mathrm{Cov}(X(t), X(t+u)) = \mathbb{E}\left[X(t)X(t+u)\right] \ , \qquad (2.1)$$

where the right-hand-side quantity is independent of t by stationarity. To alleviate notations when the context is clear, we will drop the subscript X and simply write $\gamma_X = \gamma$.

FUTURE VALUE ESTIMATOR We consider the linear estimation of future values $X(t+\Delta)$ of the process from its past values

$$X(\leq t) := (X(s))_{t-\tau<s\leq t} = \begin{pmatrix} X(t) \\ X(t-1) \\ \vdots \\ X(t-\tau+1) \end{pmatrix} \in \mathbb{R}^{\tau} \ , \qquad (2.2)$$

restricted to a maximal lag $\tau \in \mathbb{N}^*$ to avoid considering infinite vectors. It will be convenient to view the collection $X(\leq t)$ as a vector in $\mathbb{R}^{\tau}$.

Let us denote the linear estimator of $X(t+\Delta)$ from its past with the following notation:

$$\alpha^T X(\leq t) = \sum_{u=0}^{\tau-1} \alpha(u) X(t-u) = X \star \alpha(t), \qquad (2.3)$$

for some $\alpha \in \mathbb{R}^{\tau}$. We ignore biases thanks to the zero-mean assumption.

Notice that α is interpreted both as a vector, with which we performed a dot product with the vector $X(\leq t)$, and as a causal filter $(\alpha(u))_{u\in\mathbb{Z}}$, with $\alpha(u) = 0$ if $u \notin [0, \tau - 1]$. The filter notation stems from the stationarity of the process X, but the dot product notation will be convenient to express restrictions to subspaces.

SUBSPACES OF THE PAST In this framework, we restrict the forecasting coefficients α to a vector subspace V of dimension p of the ambient space $\mathbb{R}^\tau$. Let us stress that we view V as included in $\mathbb{R}^\tau$ through the canonical injection. Therefore, any vector $\alpha \in V$ satisfies $\alpha \in \mathbb{R}^\tau$. In particular, the scalar product $\alpha^T X(\leq t)$ still makes sense for $\alpha \in V$, even if the dimension of V is lower than the ambient space.

In this chapter, V will always correspond to the span of a given matrix $D \in \mathbb{R}^{\tau\times p}$ with $p \leq \tau$, so that any vector $\alpha \in V$ can be written $\alpha = D\beta$ for some $\beta \in \mathbb{R}^p$. While this latter view is closer to the numerical aspects of this work, we prefer to focus on the subspace V than D spans and not D itself in order to avoid cumbersome matrix notations.

Instead of using matrices generating V, we will often use projectors on V to simplify equations. Let $P_V : \mathbb{R}^\tau \to \mathbb{R}^\tau$ denote the orthogonal projector on V with respect to the canonical scalar product. Note that because of the convention with respect to the canonical injection, P_V is self-adjoint,

$$P_V^T = P_V. \tag{2.4}$$

By definition of P_V, it holds that

$$\forall \alpha \in V,\ \alpha^T X(\leq t) = \alpha^T P_V X(\leq t). \tag{2.5}$$

Therefore, performing a prediction with coefficients $\alpha \in V$ implies that the knowledge of the past $X(\leq t)$ is restricted to the subspace V. The subspace V encodes a prior knowledge on the time-series about which part of the past is useful to predict future values.

We will often consider a family of such subspaces, $\{V_p\}_p$, indexed by their dimension p and partially ordered for the inclusion. The parameter p will allow to tune a trade-off between different types of errors.

As an example of such a family, let us introduce the autoregressive (AR) subspaces $\mathcal{A}_p$ for $1 \leq p \leq \tau$. They are generated by the Dirac delta functions δ_u located at u for $u = 0, 1, \ldots, p-1$:

$$\mathcal{A}_p = \text{Span}\left(\{\delta_n\}_{0\leq n<p}\right) \tag{2.6}$$

The family $\{\mathcal{A}_p\}_{1\leq p\leq\tau}$ is ordered, as $\mathcal{A}_p \subset \mathcal{A}_{p+1}$. Projecting $X(\leq t)$ over $\mathcal{A}_p$ is equivalent to keeping $X(t-s)$ for $0 \leq s < p$. In other words, this is an autoregressive forecast with p past values.

The autoregressive subspaces $\mathcal{A}_p$ will serve as a baseline for linear forecasting. Indeed, any process with a continuous spectral density can be approximated arbitrary well by an autoregressive process, provided p is large enough [BD91, Chapter 4].

Finally, let us introduce the notion of a temporal support of a subspace $V \subset \mathbb{R}^\tau$. This temporal support corresponds to the minimal integer τ_s such that all vectors of V are supported in $[0, \dots, \tau_s)$:

$$\tau_s = \min\{q \in \mathbb{N} | \forall \alpha \in V, \forall u > q, \alpha(u) = 0\}. \tag{2.7}$$

In general, the temporal support is different from the dimension of a subspace. For instance, consider the subspace generated by a single vector $\mathbf{1} \in \mathbb{R}^\tau$:

$$V = \text{Span}\left(\sum_{u=0}^{\tau-1} \delta_u\right), \tag{2.8}$$

where δ_u stands for the Dirac delta function. In this case, the temporal support of V is τ, while the dimension of V is equal to 1.

2.1.1.2 *Forecasting in a subspace*

We formulate the mean-square estimation problem of $X(t+\Delta)$ from the projection of its past lags $X(\leq t)$ on a subspace V.

YULE-WALKER EQUATION Let P_V denote the orthogonal projector on V with respect to the canonical scalar product. The optimal forecasting coefficients corresponding to this linear estimation are solution of the optimal mean-square error (MSE) forecasting problem:

$$\alpha_V^* = \arg\min_{\alpha \in V} \mathbb{E}\left|X(t+\Delta) - \alpha^T P_V X(\leq t)\right|^2 . \tag{2.9}$$

α_V^* is the vector of V for which the differential of this quadratic form is 0. Since one can explicitly compute its gradient as

$$\begin{aligned}\tfrac{1}{2}\nabla_\alpha\left(\mathbb{E}\left|X(t+\Delta) - \alpha^T P_V X(\leq t)\right|^2\right) &= P_V \mathbb{E}\left[X(\leq t) X(\leq t)^T\right] P_V^T \alpha \\ &\quad - P_V \mathbb{E}\left[X(t+\Delta) X(\leq t)\right],\end{aligned} \tag{2.10}$$

the minimizer α_V^* is characterized by the relationship

$$P_V \Gamma P_V^T \alpha_V^* = P_V \gamma_\Delta , \tag{2.11}$$

where

$$\Gamma = \{\gamma(u-v)\}_{0 \leq u,v < \tau} \in \mathbb{R}^{\tau \times \tau} \tag{2.12}$$

is the covariance matrix of the past $X(\leq t)$ and

$$\gamma_\Delta = \{\gamma(u+\Delta)\}_{0 \leq u < \tau} \in \mathbb{R}^\tau \tag{2.13}$$

is the covariance vector of the past X_t and the future $X(t+\Delta)$. Equation (2.11) is called the Yule-Walker equation [BD91].

When V is equal to the ambient past space $\mathbb{R}^\tau$, we simply write

$$\alpha^* := \alpha_{\mathbb{R}^\tau}^* \tag{2.14}$$

to denote the optimal solution using all the past of size τ.

CHARACTERIZATION WITH A PROJECTION Assuming that α^* is known, one can derive an alternative characterization of the restricted solution α^*_V which will be useful to bound the mean-square error of the associated estimator. To this end, we use the following result.

Proposition 2.1.1. *For any $\alpha \in \mathbb{R}^\tau$, the mean-square error of the estimator $\alpha^T X(\leq t)$ is related to the mean-square error of $(\alpha^*)^T X(\leq t)$ through the relationship:*

$$\mathbb{E}|X(t+\Delta) - \alpha^T X(\leq t)|^2 = \mathbb{E}|X(t+\Delta) - (\alpha^*)^T X(\leq t)|^2 + \|\alpha^* - \alpha\|^2_\Gamma \tag{2.15}$$

where $\|\cdot\|_\Gamma$ is the norm induced by the scalar product $\langle u, v\rangle_\Gamma = u^T \Gamma v$.

Proof. Let us decompose the left-hand side of (2.15):

$$\begin{aligned}\mathbb{E}|X(t+\Delta) - \alpha^T X(\leq t)|^2 &= \mathbb{E}|X(t+\Delta)|^2 - 2\alpha^T \mathbb{E}[X(t+\Delta)X(\leq t)] \\ &\quad + \alpha^T \mathbb{E}[X(\leq t)X(\leq t)^T]\alpha \\ &= \mathbb{E}|X(t+\Delta)|^2 - 2\alpha^T \gamma_\Delta + \alpha^T \Gamma \alpha \qquad (2.16)\end{aligned}$$

From this last equation, we get in the particular case of α^*:

$$\begin{aligned}\mathbb{E}|X(t+\Delta) - (\alpha^*)^T X_t|^2 &= \mathbb{E}|X(t+\Delta)|^2 - 2(\alpha^*)^T \gamma_\Delta \\ &\quad + (\alpha^*)^T \Gamma \alpha^* \\ &\overset{\Gamma\alpha^* = \gamma_\Delta}{=} \mathbb{E}|X(t+\Delta)|^2 - (\alpha^*)^T \gamma_\Delta \qquad (2.17)\end{aligned}$$

Using the same argument,

$$\|\alpha^* - \alpha\|^2_\Gamma = (\alpha^*)\gamma_\Delta - 2\alpha^T \gamma_\Delta + \alpha^T \Gamma \alpha \tag{2.18}$$

Summing (2.17) and (2.18) leads to (2.16), which proves (2.15). □

Thanks to Proposition 2.1.1, α^*_V is equivalently solution of the problem

$$\alpha^*_V = \arg\min_{\alpha \in V} \|\alpha^* - \alpha\|^2_\Gamma \tag{2.19}$$

As a consequence, α^*_V is the orthogonal projection of α^* on V with respect to the scalar product induced by Γ, which we note

$$\alpha^*_V = P^\Gamma_V \alpha^* \tag{2.20}$$

where P^Γ_V stands for the projector on V with respect to $\langle \cdot, \cdot \rangle_\Gamma$, which is defined as:

$$P^\Gamma_V z = \arg\min_{v \in V} \|z - v\|_\Gamma. \tag{2.21}$$

This quadratic problem can be solved in closed-form, and the operator P^Γ_V is linear with respect to its input.

By orthogonality, we get a corollary which expresses the MSE of an arbitrary vector $\alpha \in V$ with respect to the MSE of the optimal coefficients α^* defined by Equation (2.14).

Corollary 2.1.2. *For any $\alpha \in V$, the mean-square error of the estimator $\alpha^T X_t$ is related to the mean-square errors of $(\alpha^*)^T X_t$ and $(\alpha_V^*)^T$ through the relationship:*

$$\begin{aligned} \mathbb{E}|X(t+\Delta) - \alpha^T X(\leq t)|^2 &= \mathbb{E}|X(t+\Delta) - (\alpha^*)^T X(\leq t)|^2 \\ &\quad + \|\alpha^* - \alpha_V^*\|_\Gamma^2 + \|\alpha_V^* - \alpha\|_\Gamma^2 \end{aligned} \tag{2.22}$$

where $\|\cdot\|_\Gamma$ is the norm induced by the scalar product $\langle u, v\rangle_\Gamma = u^T \Gamma v$.

2.1.2 *Empirical estimation problem*

We formulate the estimation problem in a subspace V in the empirical setting, where the autocovariance function γ is unknown.

Let us assume that a single time-series $\{X_o(t)\}_{1 \leq t \leq T}$ of finite size T, which is a realization of the original process X, is observed. Let τ_s denote the temporal support of V defined in Equation (2.7), which corresponds to the maximal temporal lag spanned by V.

With this notation, the empirical estimation of (2.9) exploiting all the available data points is:

$$\tilde{\alpha}_V = \arg\min_{\alpha \in V} \sum_{t=\tau_s}^{T-\Delta} \left| X_o(t+\Delta) - \alpha^T P_V X_o(\leq t) \right|^2 \tag{2.23}$$

Similarly to (2.11), the minimizer is characterized by $\tilde{\alpha}_V \in V$ and the empirical Yule-Walker equation:

$$P_V \widetilde{\Gamma} P_V^T \tilde{\alpha}_V = P_V \widetilde{\gamma_\Delta} \tag{2.24}$$

where the estimates $\widetilde{\Gamma}$ and $\widetilde{\gamma_\Delta}$ are given by:

$$\widetilde{\Gamma}(u,v) = \begin{cases} \frac{1}{T-\Delta-\tau_s+1} \sum_{t=\tau_s}^{T-\Delta} X_o(t-u) X_o(t-v) & \text{if } 0 \leq u, v \leq \tau_s - 1, \\ 0 & \text{otherwise.} \end{cases} \tag{2.25}$$

$$\widetilde{\gamma_\Delta}(u) = \begin{cases} \frac{1}{T-\Delta-\tau_s+1} \sum_{t=\tau_s}^{T-\Delta} X_o(t+\Delta) X_o(t-u) & \text{if } 0 \leq u < \tau_s, \\ 0 & \text{otherwise.} \end{cases} \tag{2.26}$$

Let us note that the optimal coefficients $\tilde{\alpha}_V$ are a random vector which depends on the observation $(X_o(t))_{1 \leq t \leq T}$. Conditionally on this observation, Corollary 2.1.2 allows to express the out-of-sample mean-square error of the estimator $\tilde{\alpha_V}^T X(\leq t)$, *i.e.* the MSE for a new realization independent of the observations, as:

$$\begin{aligned} \mathbb{E}\left[|X(t+\Delta) - \tilde{\alpha}_V^T X(\leq t)|^2 \middle| X_o \right] &= \mathbb{E}|X(t+\Delta) - (\alpha^*)^T X(\leq t)|^2 \\ &\quad + \|\alpha^* - \alpha_V^*\|_\Gamma^2 + \|\alpha_V^* - \tilde{\alpha}_V\|_\Gamma^2 . \end{aligned} \tag{2.27}$$

Taking the expectation over the random observations X_o, the complete MSE of $\tilde{\alpha}_V^T X_t$ has the following decomposition:

$$\mathbb{E}\left[|X(t+\Delta) - \tilde{\alpha}_V^T X(\leq t)|^2\right] = \underbrace{\mathbb{E}|X(t+\Delta) - (\alpha^*)^T X(\leq t)|^2}_{\text{Minimal error}} + \underbrace{\|\alpha^* - \alpha_V^*\|_\Gamma^2}_{\text{Approximation error}} + \underbrace{\mathbb{E}\|\alpha_V^* - \tilde{\alpha}_V\|_\Gamma^2}_{\text{Estimation error}} \tag{2.28}$$

The minimal error is the error attained in the full information setting. The approximation error comes from the restriction to the subspace V. The estimation error is a deviation from the optimal predictor in the subspace, which comes from the randomness of the observations.

2.1.3 *Approximation and estimation control*

We now explain how to control the approximation and estimation error terms in decomposition (2.28) with the subspace family $\{V_p\}_p$. On the one hand, we introduce a criterion $\mathcal{C}(V)$ which bounds the approximation error. On the other hand, the dimension p of the subspace controls the estimation error, provided the temporal support of the subspace is small compared to the total length of the time-series T.

2.1.3.1 *Approximation error*

We show that the approximation error diminishes when the dimension of the subspace V grows. We bound the approximation error with the approximation error of the covariance vector γ_Δ in V with respect to the scalar product induced by Γ.

MONOTONY Let $V_p, V_{p'}$ be subspaces such that $p \leq p'$ and $V_p \subset V_{p'}$. Because of the inclusion, we have that

$$V_{p'}^\perp \subset V_p^\perp \tag{2.29}$$

where $V^\perp$ denotes the orthogonal subspace of V with respect to $\langle \cdot, \cdot \rangle_\Gamma$. Equation (2.20) and Proposition 2.1.1 imply that

$$\|\alpha^* - \alpha_{V_{p'}}^*\|_\Gamma^2 \leq \|\alpha^* - \alpha_{V_p}^*\|_\Gamma^2 \tag{2.30}$$

Thus, the approximation error diminishes with the dimension p.

UPPER BOUND Let us now consider a single subspace V. Let us introduce the orthogonal projector $P_{V^\perp}^\Gamma$ on the orthogonal of V with respect to the scalar product $\langle \cdot, \cdot \rangle_\Gamma$, defined as

$$P_{V^\perp}^\Gamma = Id - P_V^\Gamma, \tag{2.31}$$

where P_V^Γ is defined by Equation (2.21). This operator allows to upper bound the approximation error, as stated in the following proposition.

Proposition 2.1.3. *For any subspace V, autocovariance γ and future lag Δ, let us define the quantity*

$$\mathcal{C}(V) = \|\alpha^*\| \|(P^{\Gamma}_{V^\perp})^T \gamma_\Delta\|, \tag{2.32}$$

where α^* is the optimal forecasting coefficient in the ambient vector space $\mathbb{R}^\tau$. Then the following upper-bound holds:

$$\|\alpha^* - \alpha^*_V\|^2_\Gamma \leq \mathcal{C}(V). \tag{2.33}$$

Notice that $\mathcal{C}(V)$ implicitly depends on γ and Δ, but we drop these dependencies for simplicity. The quantity $\|(P^{\Gamma}_{V^\perp})^T \gamma_\Delta\|$ measures the residual of the future coefficient in the orthogonal of V, with respect to the scalar product of V. This quantity appears naturally given the structure of the Yule-Walker equation (2.11).

Let us note that the upper bound on the approximation relies on the standard ℓ^2 norm $\|\alpha^*\|$ of the optimal prediction coefficients. It is reasonable to assume that this quantity is bounded. Otherwise, we can add an additional regularization parameter to control it if necessary.

Proof. Let us decompose in detail the approximation error:

$$\begin{aligned} \|\alpha^* - \alpha^*_V\|^2_\Gamma &= (\alpha^* - \alpha^*_V)^T \Gamma (\alpha^* - \alpha^*_V), && (2.34)\\ &= (\alpha^* - \alpha^*_V)^T (\gamma_\Delta - \Gamma \alpha^*_V)), && (2.35)\end{aligned}$$

where we have exploited $\Gamma\alpha^* = \gamma_\Delta$.

Since α^*_V is equivalently characterized by (2.20), it holds that $\alpha^* - \alpha^*_V$ belongs to the orthogonal subspace $V^\perp$ of V with respect to the scalar product induced by Γ. As $\alpha^*_V \in V$, by orthogonality it holds that

$$(\alpha^* - \alpha^*_V)^T \Gamma \alpha^*_V = \langle \alpha^* - \alpha^*_V, \alpha^*_V \rangle_\Gamma = 0. \tag{2.36}$$

Further, Equations (2.20) and (2.31) imply

$$\alpha^* - \alpha^*_V = P^{\Gamma}_{V^\perp} \alpha^*. \tag{2.37}$$

Combining these last two equations, we finally obtain

$$\|\alpha^* - \alpha^*_V\|^2_\Gamma = (\alpha^*)^T (P^{\Gamma}_{V^\perp})^T \gamma_\Delta. \tag{2.38}$$

Thanks to the Cauchy-Schwartz inequality in the canonical scalar product, we finally derive the desired bound:

$$\|\alpha^* - \alpha^*_V\|^2_\Gamma \leq \underbrace{\|\alpha^*\| \|(P^{\Gamma}_{V^\perp})^T \gamma_\Delta\|}_{\mathcal{C}(V)}. \tag{2.39}$$

□

2.1.3.2 *Estimation error*

When it comes to the estimation error, one knows that in the case of autoregressive moving-average (ARMA) processes [BD91], asymptotically as $N \to +\infty$,

$$\mathbb{E}\|\tilde{\alpha}_V - \alpha_V\|_\Gamma \to N^{-1}\mathrm{Tr}(P_V \Gamma P_V) \tag{2.40}$$

This last quantity is non-decreasing with respect to p, the dimension of V, hence the asymptotic growth of the estimation error with p. Therefore, p represents a trade-off between approximation error (small p) and estimation error (large p).

In the remaining of this chapter, we will be interested in long-range dependent processes, which, as explained later, exhibit very particular statistical properties. In this case, it becomes much more difficult to derive similar asymptotic estimations, see *e.g.* [FT; KS12].

As a rule of thumb, we therefore control the estimation error by bounding the dimension p of the subspace V.

2.2 WAVELET BASES

In this section, we introduce wavelet bases. We analyze the subspaces they define with respect to the forecasting criterion derived in the previous section. Because of the autosimilarity of the covariance function of long-range processes, a particular subspace emerges: the so-called foveal subspace, spanned by all the dilations of the wavelet closest to the present time t, without translations.

The number of vanishing polynomial moments of a wavelet is an important quantity which controls the quality of the approximation in this subspace. We stress that for forecasting purposes, the causality constraint forces to use discontinuous wavelets, in other words practically nothing but the Haar family. This will motivate our construction of a richer foveal family in the remaining part of this paper.

2.2.1 *Wavelets and Long-range dependent processes*

2.2.1.1 *Wavelets*

Wavelets are designed to provide a multiscale representation of a signal [Mal08]. They are defined thanks to a mother function ψ by dyadic dilations and translations:

$$\psi_{j,n}(u) = \frac{1}{\sqrt{2^j}}\psi\left(\frac{u - n2^j}{2^j}\right) . \tag{2.41}$$

A low-pass function ϕ is associated to ψ, and we consider similar dilations and translations $\phi_{j,n}$ of this function. The low-pass function ϕ spans a given scale, while the wavelets characterize the transitions between different scales.

The regularity of a wavelet ψ is measured by its number of vanishing polynomial moments: it is the largest integer M such that

$$\forall 0 \leq n < M, \int u^n \psi(u) = 0 \tag{2.42}$$

The oldest and simplest example of wavelets is the Haar family, which has only $M = 1$ vanishing moment. In this case, the mother wavelet is piecewise constant:

$$\psi(u) = \begin{cases} 1 & \text{if } 0 \leq u < \frac{1}{2}, \\ -1 & \text{if } \frac{1}{2} \leq u < 1, \\ 0 & \text{otherwise.} \end{cases} \tag{2.43}$$

as well as the associated low-pass function

$$\phi(u) = \mathbb{1}_{[0,1]}(u). \tag{2.44}$$

Together with the low-pass ϕ, the Haar family defines an orthonormal basis of $L^2([0,1])$. There exists more complex orthonormal wavelets, such as those of Battle-Lemarié [Bat87] or Daubechies [Dau88].

Wavelets can be discretized on a regular grid $u \in \mathbb{Z}$. After discretization on the interval $[0, 2^J)$, one obtains a basis with the Haar family made of the atoms $\{\{\psi_{j,n}\}_{0 \leq n \leq 2^{J-j}}\}_{1 \leq j \leq J} \cup \{\phi_J\}$ where

$$\psi_{j,n}(u) = \begin{cases} 2^{-j/2} & 2^j n \leq u < 2^j n + 2^{j-1}, \\ -2^{-j/2} & 2^j n + 2^{j-1} \leq u < 2^j(n+1), \\ 0 & \text{otherwise,} \end{cases} \tag{2.45}$$

$$\phi_J = 2^{-J/2} \mathbb{1}_{[0,2^J)}. \tag{2.46}$$

2.2.1.2 *Long-range Dependent Processes*

Long-range dependent time series have been introduced to model natural processes in fields as diverse as hydrology, telecommunications or finance [AV98; DOT03; Sam07]. We now review their important properties.

A discrete stationary time series $(X(t))_{t \in \mathbb{Z}}$ exhibits a long-range dependent behavior or has a long memory if its autocovariance function γ decreases so slowly that it is not summable:

$$\sum_{u \in \mathbb{Z}} |\gamma(u)| = +\infty \tag{2.47}$$

Typically, $\gamma(u)$ behaves asymptotically as a power-law: there exists $\beta \leq 1$ and $c_1 > 0$ such that

$$|\gamma(u)| \overset{u \to \pm\infty}{\sim} c_1 |u|^{-\beta} \tag{2.48}$$

In the Fourier domain, under mild conditions on γ [DOT03], this translates to a divergence of the spectral density at low frequencies, while remaining overall integrable. In other words, there exists $0 < \nu < 1$ and $c_2 > 0$ such that:

$$|\hat{\gamma}(\omega)| \overset{\omega \to 0}{\sim} c_2 \, |\omega|^{-\nu} \, . \tag{2.49}$$

The fractional Gaussian noise (fGn), defined as the increment of the fractional Brownian motion (fBm), is a well-studied long-memory process that we shall use as a simple mathematical model of long-range dependence [DOT03]. The fGn is a centered stationary Gaussian process $X(t)$ whose normalized autocovariance function $\gamma(u)$ satisfies

$$\gamma(u) = \mathrm{Cov}(X(t), X(t-u)) = \frac{1}{2}\left(|u+1|^{2H} + |u-1|^{2H} - 2\,|u|^{2H} \right). \tag{2.50}$$

The Hurst exponent $0 < H < 1$ governs the memory of the time series:

$$\gamma(u) \overset{u \to \pm\infty}{\sim} H(2H-1)\, |u|^{2(H-1)} \; . \tag{2.51}$$

This proves that $\sum_u |\gamma(u)|$ is summable if and only if $H \leq 1/2$. Therefore, the fGn has a long-memory if $1/2 < H < 1$. We will use this range for synthetic explorations.

Long-range dependent processes have statistical properties which are different than short-memory ones: estimators typically suffer from a larger variance. For instance, the convergence of the standard mean and covariance estimator from a single time-series is qualitatively slower than in the short-memory case [Ber94]. Instead of the rate of convergence in $O(N^{-1/2})$ which are usually obtained for these estimators, one gets $O(N^{-(1-H)})$ for the fGn if $H > 1/2$ [Hos96]. Since linear forecasting methods rely on such estimates, it becomes all the more important to reduce this variance.

2.2.1.3 *Wavelets and long-range dependent processes*

Wavelets are well suited to represent and study long-range dependent processes, as has been noted since long, see *e.g.* [Fla92; DOT03]. This section reviews the main properties of the wavelet coefficients of long-range dependent processes. This will motivate our use of wavelets for forecasting purposes.

Wavelets transform long-range dependent processes into short-range ones along the temporal axis. Indeed, let X be a long-range dependent process and let us filter it with a wavelet ψ_j at a given scale 2^j, resulting in a new process $X \star \psi_j$. The power spectrum of the filtered process $\widehat{\gamma_{X\star\psi_j}}$ can be expressed as

$$\widehat{\gamma}_{X\star\psi_j}(\omega) = \widehat{\gamma_X}(\omega)|\widehat{\psi}_j(\omega)|^2. \tag{2.52}$$

The dilated wavelet ψ_j has the same number M of vanishing moments as the original wavelet ψ. The number of vanishing moments translates into a regularity at low frequencies [Mal08] as

$$\widehat{\psi}_j(\omega) \overset{\omega \to 0}{=} O(\omega^M). \tag{2.53}$$

Plugging the behavior of the original long-range dependent process X at $\omega = 0$ expressed in Equation (2.49), one gets

$$\widehat{\gamma}_{X \star \psi_j}(\omega) \overset{\omega \to 0}{=} O(\omega^{2M-\nu}). \tag{2.54}$$

Since $M \geq 1$ and $\nu < 1$, the filtered process $X \star \psi_j$ has no singularity at the origin. As a consequence, it has a short memory, contrary to the original process.

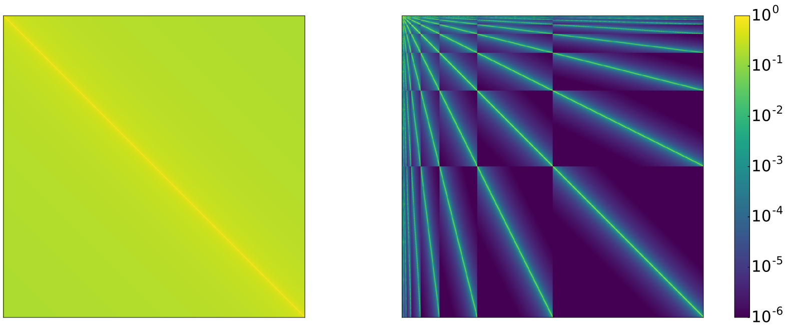

Figure 2.1: Covariance matrix Γ of a long-memory process (fGn, $H = 0.9$) in the Dirac basis (left) and Haar basis (right). The covariance matrix is sparse in the wavelet basis.

Figure 2.1 illustrates this short-memory effect. On the left, the covariance matrix $\Gamma(u, v) = \gamma_X(u - v)$ of the fractional Gaussian noise ($H = 0.9$) is displayed in the canonical Dirac basis. This covariance matrix is dense because of the slow decay of correlations. On the right, the same covariance matrix is displayed in the Haar basis: each diagonal block corresponds to the intra-scale covariance matrix of $\{\langle X \psi_{j,n} \rangle\}_n$ for varying j, while off-diagonal blocks correspond to inter-scale covariance matrices. In all these sub-matrices, off-diagonal coefficients quickly vanish as a consequence of short-memory. Inter-scale interactions tend to be more important than intra-scale interactions.

One could be tempted exploit the resulting sparse structure to compress the covariance operator in a wavelet basis, by *a priori* thresholding small coefficients [BCR91]. Indeed, this would result in little approximation error, while the total number of parameters would shrink. One could then solve the Yule-Walker Equations (2.24) with the approximated operator.

Such an approach would completely ignore the structure of the forecasting problem, which is mainly governed by the interaction of Γ and γ_Δ, defined in Equations (2.12)-(2.13). In what follows, we will therefore follow the approach outlined in Section 2.1, where we directly restrict the subspace of coefficients with respect to this interaction. In particular, this particular structure of the forecasting problem will lead us to keep the family of wavelet coefficients at various scales, but at a fixed position. The resulting subspaces will be called the *foveal* subspaces.

2.2.2 *Foveal cone*

We now analyze the subspaces generated by a wavelet family for forecasting purposes. In light of the forecasting error decomposition performed in Section 2.1, a family of subspaces naturally emerges: the *foveal subspaces*. These subspaces are generated by wavelets closest to the present boundary at all scales. The functions belonging to these subspaces have a resolution which decreases exponentially as one moves away from the present t, like the concentration of cones in the retina, hence their name [Pur+01; Bur88; Mal03].

Our exposition is organized as follows. We first perform a numerical and mathematical analysis of the decomposition of γ_Δ in the Haar basis. A foveal cone emerges, which we formalize with the notion of Haar foveal subspaces. A comparison of the upper bound on the approximation error between Haar foveal subspaces and autoregressive ones shows that the former have a much lower approximation error on the fGN. Eventually, an empirical forecasting experiment shows the favorable behavior of Haar foveal subspaces with respect to autoregressive ones.

2.2.2.1 *Insights of an adapted foveal decomposition*

A well-chosen subspace V for forecasting purposes should yield small approximation and estimation errors. Intuitively, the estimation error is controlled by the dimension of V, while the approximation error is controlled by the decomposition of γ_Δ (2.13) over V, as a result of Equation (2.33). We now look at the decomposition of γ_Δ over subspaces generated by the Haar family.

ASYMPTOTIC ANALYSIS OF γ_Δ Let us first perform a back-of-the-envelope calculation to see how the scalar products $\langle \psi_{j,n}, \gamma_\Delta \rangle$ behave for Haar wavelets $\psi_{j,n}$. We only consider wavelets whose support is included in $[0, +\infty)$.

Let us assume that we are in the asymptotic regime where there exists $c > 0$ and $0 < \nu < 1$ such that

$$\gamma_\Delta(u) = \gamma(\Delta + u) \approx c(\Delta + u)^{-\nu}. \tag{2.55}$$

Further, let us focus in the case where Δ is small with respect to u on the support of $\psi_{j,n}$, so that we perform the crude approximation

$$(\Delta + u)^{-\nu} \approx u^{-\nu}. \tag{2.56}$$

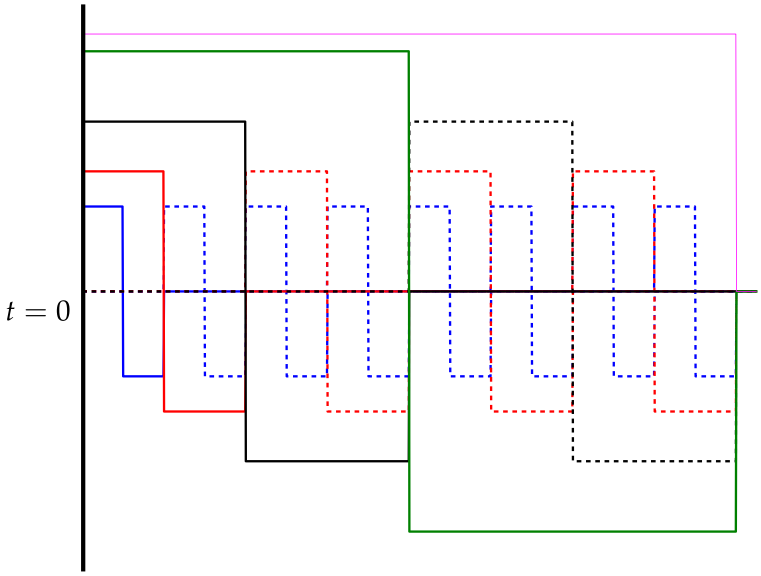

Figure 2.2: Foveal cone of width $K = 1$. Functions from the Haar family belonging to the cone are depicted in full curve, while the dotted wavelets are eliminated. The present is on the left, and one goes into the past on the right.

Under these assumptions, $\gamma_\Delta(u) \approx cu^{-\nu}$. Let us now state a simple asymptotic analysis of the scalar product of $u^{-\nu}$ and wavelets $\psi_{j,n}$, in the continuous-time setting.

Proposition 2.2.1. *Let ψ be a wavelet with M vanishing moments and with a bounded support included in $[0, +\infty)$. Let $0 < \nu < 1$. For all $n \in \mathbb{N}^*$, one has*

$$\int_0^{+\infty} u^{-\nu} \psi_{j,n}(u) du = O\left(a^j n^{-(\nu+M)}\right), \tag{2.57}$$

where $a = 2^{1/2-\nu}$, with a bounding constant independent of j and n.

This proposition is proved in Appendix A. Under the harsh assumptions we have made, the scalar product $\langle \psi_{j,n}, \gamma_\Delta \rangle$ roughly scales as the right-hand-side of Equation (2.57). Importantly, the decreasing rate in n, $\nu + M$ is independent of j. It is larger than the one of the original process since $M \geq 1$, thanks to the shorter memory of wavelet-filtered processes.

As a consequence, the renormalized decomposition coefficients satisfy

$$\frac{\langle \psi_{j,n}, \gamma_\Delta \rangle}{\langle \psi_{j,0}, \gamma_\Delta \rangle} \sim n^{-(\nu+M)}, \tag{2.58}$$

for all j. If these renormalized coefficients are thresholded with respect to any value, one ends up with a family of coefficients $\langle \psi_{j,n}, \gamma_\Delta \rangle$ for all j and for all $0 \leq n < K$, for a certain $K > 0$. We call the resulting set of coefficients the foveal cone of width K. Figure 2.2 depicts the Haar wavelets belonging to the foveal cone of width $K = 1$.

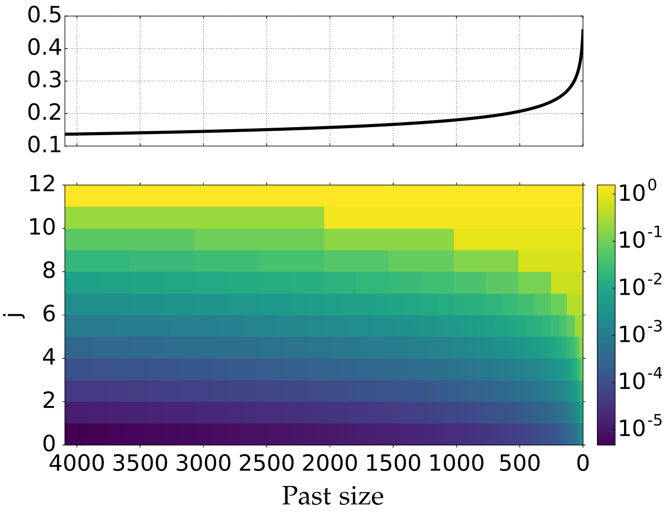

Figure 2.3: Covariance vector between the past and future value $\gamma_\Delta(u)$ (top) and its decomposition in the Haar family (bottom). The horizontal axis corresponds to the distance to the past, with the present on the right. The foveal cone corresponds to the coefficients of large magnitude.

EMPIRICAL DECOMPOSITION OF γ_Δ In the previous paragraph, a rough mathematical analysis of γ_Δ has shown the emergence of a so-called foveal cone $\{j\} \times \{0 \leq n < K\}$. We now look at numerical decompositions of γ_Δ on the discrete Haar family $\{\psi_{j,n}\}_{j,n} \cup \{\phi_J\}$ to see if these subspaces do indeed appear in the discrete empirical setting.

Figure 2.3 shows the decomposition of the vector $\gamma_\Delta(u)$ in the Haar family for the fGN with $H = 0.9$ and $\Delta = 10$. The image displays the decomposition coefficients $\psi_{j,n}^T \gamma_\Delta$ with j along the y-axis and n along the x-axis. The coefficients are displayed as a colored checkerboard, with the width of each box corresponding to the temporal support of the corresponding wavelet.

One can see that the coefficients $\{\psi_{j,n}^T \gamma_\Delta\}_{j,0\leq n<K}$ for each K are the ones concentrating most of the energy of γ_Δ, up to a scaling factor. This is another evidence of the relevance of the foveal cone for forecasting.

2.2.2.2 *Haar foveal subspaces*

In the previous section, we have shown that the energy of γ_Δ in the Haar family is mostly concentrated in the wavelets nearest the temporal boundary. We now provide a formal definition of this set of coefficients in the discrete setting.

HAAR FOVEAL SUBSPACE Let us consider a maximal scale 2^J such that $2^J \leq \tau$, so that the support of the Haar wavelets is lower than the maximal past. The Haar family at scale 2^J consists in the wavelets

$\{\psi_{j,n}\}_{1\leq j\leq J}$ (2.45) along with the low-pass ϕ_J (2.46). The Haar foveal subspace of cone width K and scale parameter J is defined as the set of wavelets closer to the temporal boundary, up to a maximal translation K, with all scales up to 2^J:

$$H_{K,J} = \text{Span}\left(\{\{\psi_{j,n}\}_{0\leq n<\min(K,2^{J-j})}\}_{1\leq j\leq J} \cup \{\phi_J\}\right). \tag{2.59}$$

The technical term $\min(K, 2^{J-j})$ ensures that all functions of $H_{K,J}$ are supported in $[0, 2^J)$. When the parameters are large enough, one recovers the whole space of support 2^J:

$$H_{2^{J-1},J} = \mathbb{R}^{2^J}. \tag{2.60}$$

The family $\{H_{K,J}\}_{K,J}$ is induced with a partial order for the inclusion. Indeed, if $K \leq K'$, $H_{K,J} \subset H_{K',J}$. Moreover, if $J \leq J'$, then $H_{K,J} \subset H_{K,J'}$.

FOVEAL SPACE When $J \to +\infty$, the limiting subspace $H_{K,\infty} \subset \ell^2(\mathbb{N})$ defined by the relation

$$H_{K,\infty} = \overline{\cup_{J=1}^{+\infty} H_{K,J}} \tag{2.61}$$

satisfies the following property:

$$\forall f \in H_{K,\infty},\ f\left(\left\lfloor\frac{\cdot}{2}\right\rfloor\right) \in H_{K,\infty}. \tag{2.62}$$

Equation (2.62) is a discretized version of a property called foveality, which characterizes a subspace V of $L^2(\mathbb{R})$ verifying [Mal03]

$$\forall f \in V,\ f\left(\frac{\cdot}{2}\right) \in V. \tag{2.63}$$

The finite-dimensional subspaces $H_{K,J}$ satisfy a weaker property than Equation (2.62):

$$\forall f \in H_{K,J} \text{ such that } \text{Supp}(f) \subset [0, 2^{J-1}),\ f\left(\left\lfloor\frac{\cdot}{2}\right\rfloor\right) \in H_{K,J}. \tag{2.64}$$

Nonetheless, by analogy, we stick to the adjective foveal for these subspaces.

2.2.2.3 *Comparison of the Haar foveal subspaces and the autoregressive subspaces*

We now compare the Haar foveal subspaces $H_{K,J}$ and the autoregressive subspaces $\mathcal{A}_p$ in terms of approximation power on the fGN, for which the auto-covariance γ is known (2.50).

Figure 2.4 compares the upper bound on the approximation error (2.32) for the Haar foveal subspaces $H_{K,J}$ (2.59) and for the autoregressive spaces $\mathcal{A}_p$ (2.6), for a fGN with $H = 0.9$ and $\Delta = 10$ and for $J = 8$. The horizontal axis corresponds to the dimension of each sub-space, while the vertical axis is the ratio between the upper bound

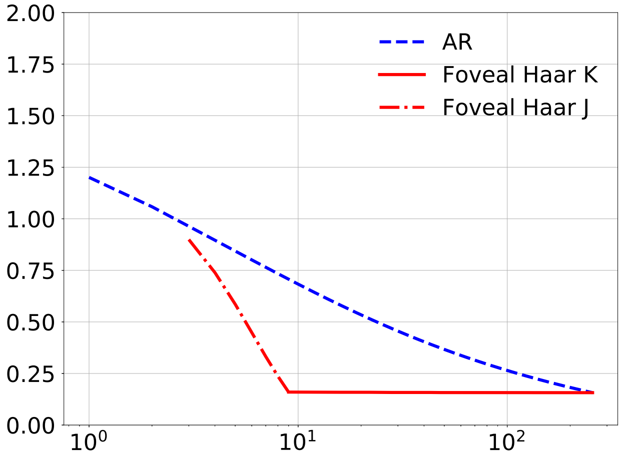

Figure 2.4: Upper bound on the approximation error $C(V)$ (2.32) for the Haar foveal subspaces $H_{K,J}$ (2.59) and for the autoregressive spaces $\mathcal{A}_p$ (2.6). The horizontal axis corresponds to the dimension of the subspace, while the vertical axis corresponds to the ratio between the upper bound $\mathcal{C}(V)$ and the optimal MSE. The dotted curve for foveal Haar corresponds to the family $\{H_{1,j}\}_{1\leq j\leq J}$, while the solid curve corresponds to the family $\{H_{K,J}\}_{1\leq K\leq 2^J}$, both for $J=8$.

and the optimal MSE. A subspace V should achieve a value below 1 with as few parameters as possible. The blue curve corresponds to the autoregressive subspaces, while the red curves to the foveal Haar subspaces. There are two regimes for the Haar subspaces: the dotted curve corresponds to the family with fixed cone width and growing scale $\{H_{1,j}\}_{1\leq j\leq J}$, while the full curve corresponds to a fixed scale and growing cone width $\{H_{K,J}\}_K$.

We observe that for an similar dimension, $\mathcal{C}(H_{K,J}) \leq \mathcal{C}(\mathcal{A}_p)$. As a consequence, the foveal Haar subspaces should be more adapted than the autoregressive subspaces for forecasting purposes. Further, the maximal scale 2^j allows the approximation error to diminish at a faster rate than the past size p of the autoregressive subspaces.

We note in particular that for the subspace $H_{1,J}$, which has the same temporal support as $\mathcal{A}_{2^J}$, the upper bounds on the approximation error are almost equal. However, the dimension of $H_{1,J}$ is $J+1$, whereas $\mathcal{A}_{2^J}$ has dimension 2^J, so we will expect the corresponding estimation error to be much larger in the latter case.

Finally, the cone width K appears to have little effect on the approximation error. We therefore only use $K=1$ in the following.

2.2.3 *Forecasting with the Haar foveal family*

In the previous section, we have introduced the Haar foveal family, which was motivated after an analysis of the decomposition of γ_Δ

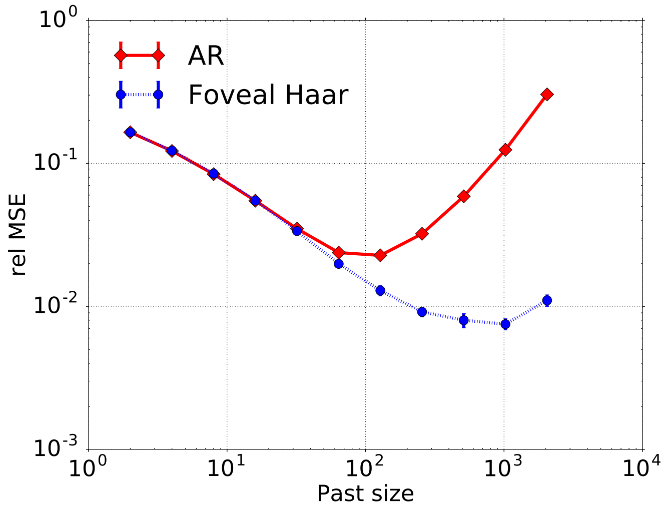

Figure 2.5: Evolution of the relative MSE (2.65) with respect to the temporal support of the autoregressive subspaces $\{\mathcal{A}_p\}_p$ and the Haar foveal subspaces $\{H_{1,j}\}_{1\leq j\leq J}$.

onto the subspaces induced by the Haar family. We now evaluate the effectiveness of the foveal Haar subspaces $\{H_{1,J}\}_J$ compared the baseline autoregressive subspaces $\{A_p\}_p$ on the fGN model. Numerical experiments show that $\{H_{1,J}\}_J$ achieve a smaller relative MSE than $\{A_p\}_p$.

2.2.3.1 *Experimental setup*

Let us introduce the relative MSE with respect to the optimal MSE attained by α^*:

$$\text{relMSE}(\alpha) = \frac{\mathbb{E}|X(t+\Delta) - \tilde{\alpha}_V^T X_t|^2 - \mathbb{E}|X(t+\Delta) - (\alpha^*)^T X_t|^2}{\mathbb{E}|X(t+\Delta) - (\alpha^*)^T X_t|^2} \quad (2.65)$$

The relative MSE measure the sum of the approximation and estimation errors of coefficients α in a subspace V, in units renormalized by the optimal MSE attained at α^*.

We use time-series of length $T = 10^4$, which are realizations of a fGN of Hurst parameter $H = 0.9$. For each subspace V, which is either A_p or $H_{1,J}$ for some p or J, we compute a numerical estimate of the relative MSE with 100 independent realizations. We take $J \in \{1, \dots, 11\}$ and $p \in \{2^j\}_{1\leq J\leq 11}$. We restrict ourselves to a cone width $K = 1$ since a larger cone only brings minor improvements.

2.2.3.2 *Results*

The evolution of the relative MSE with respect to the temporal support of the subspace is represented in Figure 2.5. The foveal Haar subspaces achieve a smaller relative MSE than the autoregressive subspaces.

Indeed, for the foveal Haar subspaces, the minimum is attained for a past size $2^J = 2^{10} = 1024$, corresponding to a space of dimension 11, whereas it is attained for $p = 128$ in the autoregressive case.

These results were expected given the decomposition of the mean-square error (2.28). Indeed, the foveal Haar family $H_{1,J}$ allows to span a similar temporal support as $\mathcal{A}_{2^J}$ with a similar approximation error, whereas the former has a much lower dimension. Therefore, the estimation error of the foveal Haar family is lower than the corresponding autoregressive spaces.

These experiments validate the choice of the foveal Haar family for forecasting purposes on long-range dependent time-series.

2.2.4 *Causality constraint*

In the previous section, we have identified a class of subspaces generated by a family of dilations of wavelets, the foveal subspaces. Numerical experiments with the Haar wavelets have demonstrated that their forecasting performances compare favorably with respect to the autoregressive subspaces on a long-range dependent model, the fGN. We now investigate whether it is possible to use another mother wavelet ψ for building such foveal spaces. To enlighten this choice, we list the constraints that ψ should satisfy.

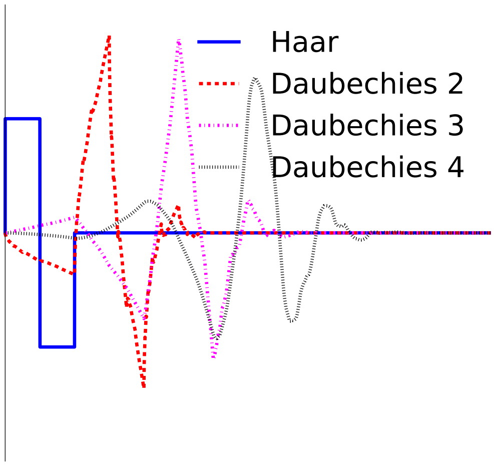

Figure 2.6: Closest wavelets to the present boundary on the left, with the past on the right.

On the one hand, ψ should be causal, i.e. $\psi(t) = 0$ for $t < 0$. This constraint comes from the fact that we should not use future information when performing a forecast. On the other hand, ψ should take a non-zero value in 0. Indeed, index 0 corresponds to the value which is temporally closest to the future, which we attempt to forecast: among all past values, it should be the most correlated with the future.

Both constraints imply that ψ should be discontinuous. As shown in Figure (2.6), this rules out standard Daubechies wavelets. Insofar as a discontinuity is necessary, we are left with nothing but the Haar wavelets. Previous works related to forecasting with wavelets were also led to the same conclusion, see *e.g.* [RSM05].

Haar wavelets obtain good forecasting performances on synthetic time-series, but the foveal subspaces they generate may lack flexibility for real time-series, as the cone width parameter K appears to have little effect. In the remaining of this chapter, we therefore introduce a new class of foveal wavelets, which should be more flexible than the foveal Haar family.

2.3 FOVEAL WAVELETS

In this section, a new representation of the past of a long-range dependent time-series for forecasting purpose is proposed, inspired by previous works studying foveal spaces. The resulting wavelets are foveal by construction, amenable to the causality constraint, and have more flexibility than the Haar wavelets.

2.3.1 *General principle*

We first give a formal construction of the new foveal wavelets. Since this construction relies on dyadic dilations in the spirit of the foveality property of Equation (2.63), we use notations related to a continuous time variable. Discretization issues are addressed in the next sections for each particular family we build.

2.3.1.1 *Strategy*

Starting from a single causal function, which defines a window ϕ^0 at a base scale, its dilations ϕ_j^0 at all scales 2^j define a first foveal subspace. To enlarge this subspace and recover useful information that might have been lost, polynomial terms $u^m\phi^0(u)$ of increasing order are added to the family at all scales in a foveal fashion. These polynomial terms are designed so as to have vanishing polynomial moments, leading to functions ϕ_j^m. Finally, we consider differences between consecutive scales 2^j and 2^{j-1} to increase the number of vanishing polynomial moments, thus defining the foveal wavelets ψ_j^m.

2.3.1.2 *Formal construction*

WINDOW FUNCTION Let $\theta : \mathbb{R}^+ \to \mathbb{R}^+$ be a non-increasing function, with at least an asymptotic exponential decay in $+\infty$. Based on θ, we define a causal normalized window function

$$\phi^0(u) = c_0\left(\theta(u/2) - \theta(u)\right), \tag{2.66}$$

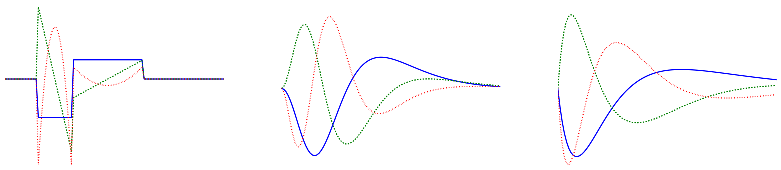

Figure 2.7: Foveal wavelets $\{\psi_j^n\}_{n\in\{0,1,2\}}$ for a fixed scale 2^j. **Left**: Indicator wavelets. **Center**: Gaussian wavelets. **Right**: Exponential wavelets.

where $c_0 > 0$ is a normalizing constant ensuring $\|\phi^0\|_2 = 1$. We then consider dilations of this window for all $j \in \mathbb{Z}$:

$$\phi_j^0(u) = 2^{-j/2}\phi^0(2^{-j}u). \tag{2.67}$$

POLYNOMIALS To enlarge the space of functions spanned by $\{\phi_j^0\}_{j\in\mathbb{Z}}$, we consider products of these functions with polynomials. More precisely, we define

$$\phi^m(u) = \left(\sum_{k=0}^{m} a_k u^k\right)\phi^0(u), \tag{2.68}$$

where the coefficients a_k are specified by enforcing a vanishing moment property:

$$\forall p \in \{0,\dots,m-1\}, \int_0^{+\infty} t^p\phi^m(t)dt = 0. \tag{2.69}$$

Plugging Equation (2.68) into Equation (2.69) leads to a linear system with $m+1$ unknowns and m constraints. The constraints $a_m > 0$ and $\|\phi^m\| = 1$ are further enforced so as to avoid ambiguities. This new function $\{\phi^m\}$ is in turn dilated at all scales j:

$$\phi_j^m(u) = 2^{-j/2}\phi^m(2^{-j}u). \tag{2.70}$$

DIFFERENCES IN SCALE At this point, we note that the order $m = 0$ has non vanishing moment, since ϕ^0 does not flip sign. To increase the number of vanishing moments of the elements of the family, we consider differences between scales, following a method introduced in [Mal03]

$$\psi_j^m = c_{jm}^a\phi_j^m - c_{jm}^b\phi_{j-1}^m, \tag{2.71}$$

where the coefficients c_{jm}^a, c_{jm}^b are chosen such that $c_{jm}^a > 0$, $\|\psi_j^m\| = 1$ and ψ_j^m has $m+1$ vanishing moments. We call the functions ψ_j^m the foveal wavelets.

FOVEAL WAVELETS In the continuous setting, we would consider the foveal family $\{\psi_j^m\}_{j\in\mathbb{Z}}^{0\leq m\leq M} \cup \{\phi_0^m\}^{m\leq M}$. Let V denote the span of this family. It holds that

$$f \in V \Rightarrow f\left(\frac{\cdot}{2}\right) \in V, \tag{2.72}$$

which shows that V is a true foveal space of functions [Mal03]. Note that the idea of considering multiple mother functions is close to the Alpert multi-wavelets [Alp93], except that we only consider dilations of these functions and no translations.

In the discrete setting, we bound the maximal and minimal scales of the wavelet family considered. Let us therefore define the foveal family $F(J, M)$ with minimal scale 1, maximal scale 2^J and maximal polynomial order M as

$$F(J, M) = \{\psi_j^m\}_{0 \leq j \leq J}^{m \leq M} \cup \{\phi_0^m\}^{m \leq M}. \tag{2.73}$$

The associated foveal subspace is the span of $F(J, M)$.

2.3.1.3 *Link with previous foveal wavelets*

Foveal subspaces were mathematically formalized in [Mal03]. In our construction, we use the same basic principle for building a foveal subspace by considering the dilations at all scales of a family of functions. In particular, Equation (2.66) is adapted from [Mal03].

However, there are several differences between the foveal wavelets we introduce and those proposed in [Mal03]. Indeed, we do not use any orthogonalization procedure between scales: ψ_j^m and $\psi_{j'}^m$ are not orthogonal. Further, the procedure we use to generate the functions ϕ_j^m with more polynomial moments is different than the one proposed. Last, at a given scale 2^j, we consider all the generating functions ψ_j^m for $0 \leq m \leq M$, which have an increasing number of vanishing moments, instead of keeping a single one.

2.3.2 *Indicator window*

2.3.2.1 *Definition*

In this section, we consider the simplest window function as the building block of the foveal wavelets, which is the indicator window:

$$\theta^{(I)} = \mathbb{1}_{[0,1]}. \tag{2.74}$$

In this case, one can verify that

$$\phi^0 = \mathbb{1}_{[1,2]}, \tag{2.75}$$

and therefore

$$\phi_j^0 = 2^{-j/2} \mathbb{1}_{[2^{j-1}, 2^j]}. \tag{2.76}$$

We note that ϕ_j^0 and $\phi_{j'}^0$ have distinct supports if $j \neq j'$.

As a consequence of Equation (2.75), the functions ϕ^m, defined by Equation (2.69), consist in an orthonormal family of polynomials whose support lies in the compact interval $[0, 1]$. Therefore, the functions ϕ^m are affine transforms of the Legendre polynomials, which are an orthonormal basis of polynomials on $[-1,1]$ [AS73]. As a consequence, note that in this particular case the wavelets ψ_j^m and $\psi_j^{m'}$ for $m \neq m'$ are orthonormal. Figure 2.7 displays foveal wavelets ψ_j^m for varying m.

2.3.2.2 *Discretization*

The procedure is summarized in Algorithm 1 in Appendix A.2.

case of ϕ_j^n For each integer $j \geq 0$, let us define the dyadic integer interval I_j which supports ϕ_j^m as:

$$I_j = \begin{cases} [0] & \text{if } j = 0, \\ [2^j - 1, 2^j, \dots, 2^{j+1} - 2] & \text{if } j \geq 1. \end{cases} \tag{2.77}$$

Since $|I_j| = 2^j$, in particular it is impossible to build more than 2^j free functions on I_j. As a consequence, ϕ_j^m can only be constructed for $0 \leq m < 2^j$. In cases where m remains smaller than 10, this limitation is only felt for the small scales $j \leq 3$.

For each separate scale $2^j \geq 2$, we orthonormalize the discrete polynomial vectors $\{(u^m)_{u \in I_j}\}_{m<2^j}$ with the Gram-Schmidt algorithm. We thus obtain the functions $\{\phi_j^m\}_j^{m<2^j}$. Note that for the case $j = 0$, we can only build

$$\phi_0^0 = \delta_0, \tag{2.78}$$

where δ_0 stands for the Dirac function located at 0.

case of ψ_j^m To obtain the foveal wavelets ψ_j^m with an additional vanishing moment, we first define

$$\bar{\psi}_j^m = \phi_j^m - c_{jb}^b \phi_{j-1}^m, \tag{2.79}$$

and find c_{jm}^b by enforcing the additional moment property

$$\sum_{u \in I_{j-1} \cup I_j} u^m \bar{\psi}_j^m(u) = 0. \tag{2.80}$$

ψ_j^m is finally found by normalizing $\bar{\psi}_j^m$:

$$\psi_j^m = \|\bar{\psi}_j^m\|_2^{-1} \bar{\psi}_j^m. \tag{2.81}$$

Let us note that if $m \geq 2^{j-1}$, then ϕ_{j-1}^m is not defined. In this case, we propose a proxy for ψ_j^m as follows. In Equation (2.79), we replace ϕ_{j-1}^m by ψ_j^{m-1}, which also has m vanishing moments. By applying this trick recursively on m, one builds a function ψ_j^m which has support in $I_{j-1} \cup I_j$, is polynomial on I_j and on I_{j-1} and has $m + 1$ vanishing moments. As these criteria characterize the wavelets ψ_j^m in the continuous case, this proxy is consistent.

2.3.3 *Gaussian window*

2.3.3.1 *Definition*

We now consider foveal wavelets defined by a Gaussian window, in order to define smoother wavelets which have a better localization in the Fourier domain:

$$\theta^{(G)}(u) = e^{-u^2/2\sigma^2}. \tag{2.82}$$

The bandwidth parameter σ^2 is adapted so that the functions ϕ_j^0 have approximately the same support as the indicator window. We use $\sigma^2 = 0.3$ in numerical applications. Contrary to the indicator wavelets, in this case the functions ϕ_j^m are no longer orthogonal between each other.

2.3.3.2 *Discretization*

Let us now detail the construction of the discrete wavelets ψ_j^m, which is summarized in Algorithm 2 in Appendix A.2.

For $j = 0$, we collapse all polynomial orders in the Dirac delta function δ_0. For each $j \geq 1$, we define a maximal support size τ_j such that the tail of the Gaussian can be neglected:

$$\forall t \geq \tau_j, \theta^{(G)}(2^{-j}u) \leq 10^{-3}. \tag{2.83}$$

On the discrete interval $[0, \tau_j]$, we define the family $\{\phi_j^m\}_{m<2^j}$ by orthonormalizing the family

$$\left\{ \left(u^m(\theta^{(G)}(2^{-(j+1)}u) - \theta^{(G)}(2^{-j}u)) \right)_{u\in[0,\tau_j]} \right\}_{m<2^j} \tag{2.84}$$

with the Gram-Schmidt orthogonalization procedure.

Note that the constraint $m < 2^j$ in Equation (2.84) is not necessary since $2^j \ll \tau_j$, but it is added to allow a fair comparison with the discretized indicator foveal wavelets. The foveal families will thus have the same number of elements for the same values of the hyper-parameters J and M.

Based on the resulting family $\{\{\phi_j^m\}_{m<2^j}\}_{0\leq j\leq J}$, the foveal wavelets $\{\{\psi_j^m\}_{m<2^j}\}_{1\leq j\leq J}$ are built following the same procedure as for the indicator foveal wavelets. We first define, for $m < 2^{j-1}$,

$$\bar{\psi}_j^m = \phi_j^m - c_{jn}^b \phi_{j-1}^m \tag{2.85}$$

where c_{jm}^b is found to ensure that

$$\sum_{u\in[0,\tau_j]} u^m \bar{\psi}_j^m(u) = 0. \tag{2.86}$$

Then, ψ_j^m is a normalized version of $\bar{\psi}_j^m$:

$$\psi_j^m = \|\bar{\psi}_j^m\|_2^{-1} \bar{\psi}_j^m. \tag{2.87}$$

If $m \geq 2^{j-1}$, then we use the same recursive approach as for the indicator wavelets, defining first

$$\bar{\psi}_j^m = \phi_j^m - \beta\psi_j^{m-1} \tag{2.88}$$

and following the same steps. The support size and the number of vanishing moments are conserved with this alternative approach.

2.3.4 *Exponential window*

2.3.4.1 *Definition*

We now introduce the exponential window, which is defined as

$$\theta^{(E)}(u) = e^{-\mu u}, \tag{2.89}$$

where $\mu > 0$ is a constant. This window is smoother than the indicator window, so in the Fourier domain it has a faster decay than its indicator counterpart. Contrary to the Gaussian window, the foveal wavelets generated by this function can be implemented as a rational filter in the discrete case. These wavelets therefore have a lower computational burden than the Gaussian ones.

The constant μ is adjusted so that the functions ϕ_j^0 have approximately the same support for $\theta^{(E)}$ and $\theta^{(I)}$. Numerically, $\mu = 1.4$ is used.

2.3.4.2 *Discretization*

The wavelets ψ_j^m based on the exponential window can be built according to the same procedure as the Gaussian one, with an appropriate truncation of the support. However, they can also be implemented as a rational filter.

Indeed, it is clear that $\theta^{(E)}$ can be implemented as an IIR filter, as it corresponds to an exponential moving average. This remains true for its dilations, provided the temporal constant is dilated accordingly, and for linear combinations of such functions. Similarly, filters of the type $(u^k q^u)_u$ for some integer k and $0 < q < 1$ can be implemented with a rational filter. As the remaining operations for the construction of ψ_j^m only involve dilations and linear recombinations, the wavelets ψ_j^m can be implemented as rational filters.

2.4 forecasting experiments

In this section, we validate the foveal wavelets we have introduced with forecasting experiments on synthetic and real wavelets.

2.4.1 *Synthetic time-series*

We compare the forecasting performances of the proposed foveal wavelets to the foveal Haar wavelets with the fGN synthetic model. This closed-form model allows us to experiment the effect of the parameters J and M of the proposed foveal families ψ_j^n.

2.4.1.1 *Methods*

The experiments use the setup introduced in Section 2.1.2, where only a single observation X_0 is accessible and one wants to find optimal forecasting parameters $\widetilde{\alpha}_V$ in a subspace V. We aim at measuring the out-of-sample mean-square error (2.28) for each subspace V. Decomposition (2.28) is used to approach each term separately.

The optimal value with respect to the whole past is approached numerically by solving the Yule-Walker Equation (2.11) for a large past window. The past window size $\tau = 2 \times 10^4$ used is much larger than the temporal supports of the considered subspaces in order to avoid boundary effects.

The approximation error is computed by solving the Yule-Walker Equation (2.11) into each subspace V. The resulting coefficients α_V^* are then plugged in the closed-form expression of this approximation error.

We do not have any closed-form equation for the estimation error. Therefore, a simple Monte-Carlo simulation is used, where independent realizations of a fGN X_0 of length $T = 10^4$ are generated to compute empirical coefficients $\widetilde{\alpha}_V^*$, which are then used to evaluate the deviation from the optimum. Using 10^3 independent realizations allows to obtain reasonable confidence intervals on the means.

We report relative MSE for coefficients α (relMSE(α)) with respect to the optimal MSE attained at α^*, as defined in Equation (2.65). From Equation (2.28), the relative MSE is the sum of the approximation error and the estimation error, both properly renormalized by the optimal MSE.

The future lag $\Delta = 100$ is used to avoid discretization artifacts. The fractional Gaussian Noise has a Hurst parameter $H = 0.9$. Realizations are simulated using the fast procedure described in [WC94].

2.4.1.2 *Impact of the maximal scale* 2^J

This section investigates the evolution of the forecasting error with the maximal scale 2^J of the foveal family $F(J, 0)$. For a fairer comparison with the foveal Haar family, only $M = 0$ polynomials are used: thus, all foveal representations have the same number of components at a given scale 2^J.

Figure 2.8 reports the relative MSE as well as the approximation error in adimensional units when the maximal scale J varies. When

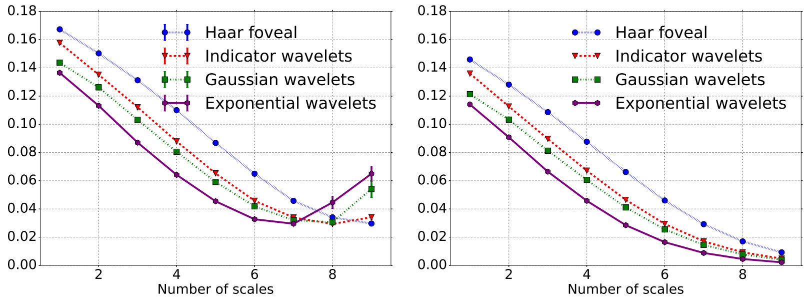

Figure 2.8: MSE for various foveal representations with respect to the number of numbers of scales J for $M = 0$. **Left**: Relative MSE (2.65). **Right**: Approximation error (2.28). All quantities are expressed divided by the optimal MSE so as to have adimensional units.

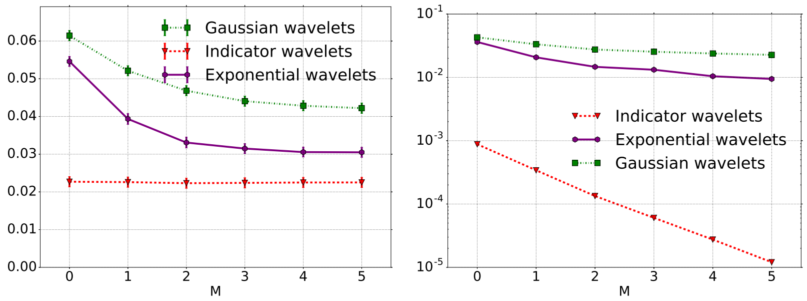

Figure 2.9: MSE for the proposed foveal wavelets at $J = 4$ with respect to the maximal polynomial order M **Left**: Relative MSE (2.65). **Right**: Approximation error (2.28). All quantities are expressed divided by the optimal MSE so as to have adimensional units.

J grows, the approximation error of all foveal wavelets diminishes, with a lower value for foveal wavelets. Indeed, the approximation error mainly diminishes with the temporal support of the representation. When J becomes too large, the relative MSE grows because the estimation error grows. Indeed, this corresponds to the regime where the past size is commensurate with the size of the time-series, so the number of samples used to estimate correlations diminishes. Overall, the minimal value attained over all J tends to be the same for all representation for a simple process such as the fGN.

2.4.1.3 *Impact of the maximal polynomial order M*

We now investigate the impact of the maximal polynomial order M on the forecasting MSE of the fGN. Figure 2.9 displays the evolution of the relative MSE with respect to M for $J = 4$.

In all cases, the approximation error diminishes with respect to M, with a rate which is much faster for the indicator foveal wavelets than the other ones. When it comes to the expected MSE, it tends to diminish with respect to M for the Gaussian and Exponential wavelets, but not for the indicator window. Indeed, for the indicator window the MSE is dominated by the estimation error, which is almost unchanged when varying J, as the support size is not affected by this value.

We conclude that the polynomial order M brings flexibility for the foveal wavelets with a very large support, such as the Gaussian and Exponential ones. The benefit is less obvious for the indicator foveal wavelets.

2.4.2 *Real time-series*

This section investigates the impact of the proposed representations on three real time-series exhibiting a long-range dependence behavior. On this benchmark, the flexibility of the proposed foveal wavelets allows to reduce the mean-square error both with respect to the foveal Haar wavelets and the autoregressive baseline.

2.4.2.1 *Dataset*

The real time-series we use are depicted in Figure 2.10, together with their power spectra $|\widehat{\gamma}_X|^2(\omega)$. The first one is the "Sunspot" time series [SIL49], which is a measure of the number of black spots observed on the Sun each month since 1749. The second one is the "MacKey-Glass" time series [MG77]: it is the solution of a chaotic time-delayed differential equation:

$$\dot{x}(t) = \gamma \frac{x(t-\tau)}{1+x^{10}(t-\tau)} - \beta x(t), \tag{2.90}$$

with delay $\tau = 10$, $\gamma = 2$, $\beta = 1$ and $x(t=0) = 0.2625$. The solution was computed numerically using a fourth-order Runge-Kutta method. Both time series are known in the forecasting literature to exhibit long-range dependent patterns, as can be read from their spectral density function $\hat{\gamma}$. The third time series, called "PM10" is original: it results from hourly measurements of the concentration of atmospheric particulate matter of diameter below 10 μm in Paris, encompassing more than 4 years of data. It features a long memory, witness its power spectrum in Figure 2.10 (bottom right). Prior to preprocessing, it initially displayed a complex seasonal behavior, which was removed to end up with an approximately stationary series.

2.4.2.2 *Methods*

PREPROCESSING We split the time series in a training part and a testing part with respectively 70% and 30% of the data points, the training part coming first chronologically. We then ensured that the training part had zero mean and unit variance and applied the corresponding affine transformation to the testing part. Prior to these steps, in the case of the "PM10" time series, which had very strong

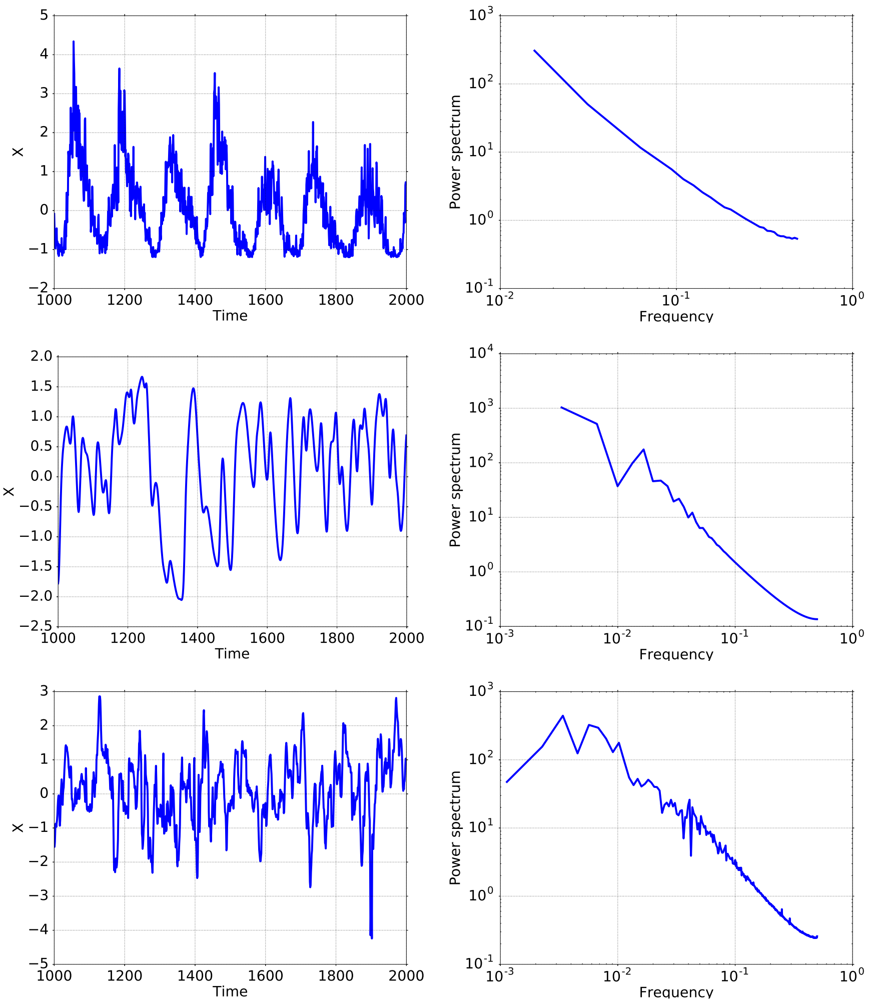

Figure 2.10: Realistic 1D time series as used in the experiments. Left: Subset of the time series; right: Power spectrum $|\hat{\gamma}_X|^2(\omega)$. From top to bottom: “Sunspot”, “MacKey-Glass”, “PM10”.

seasonal patterns and multiplicative increments, we used a logarithmic mapping and filtered out the corresponding frequencies so as to end up with an approximately stationary time series.

forecasting For all representations, we performed a simple least-squares regression (with a bias) to forecast the target at $t + \Delta$. We did not use any regularization, insofar as the foveal representations act as regularization. To measure the accuracy of this forecasting, we used the Normalized Mean Square Error (NMSE), defined on the testing set as follows:

$$\text{NMSE} = \frac{1}{|I_{\text{test}}|} \frac{\sum_{t \in I_{\text{test}}} |x(t+\Delta) - \tilde{x}(t+\Delta)|^2}{\text{Var}\left[(x(t))_{t \in I_{\text{test}}}\right]}, \tag{2.91}$$

where I_{test} is the subset of test indices t, $\tilde{x}(t+\Delta)$ denotes the forecast and $x(t+\Delta)$ the value to predict.

We optimized upon the hyperparameters of each representation, notably the past size, with oracle, *i.e.* according to test set results. This reduces the bias linked to causality by crossed validation in the train set, at the expense of a slight underestimation of the generalization error on the test set. But since the number of parameters for all foveal methods is commensurate, this underestimation remains limited.

The baseline consists in an autoregressive method. The values Δ were set as follows: for "Sunspot", $\Delta = 12$ (corresponding to one-year-ahead forecasting); for "Mackey-Glass", $\Delta = 20$ (in order to avoid discretization artifacts); for "PM10", $\Delta = 24$ (corresponding to one-day-ahead forecasting).

2.4.3 *Results*

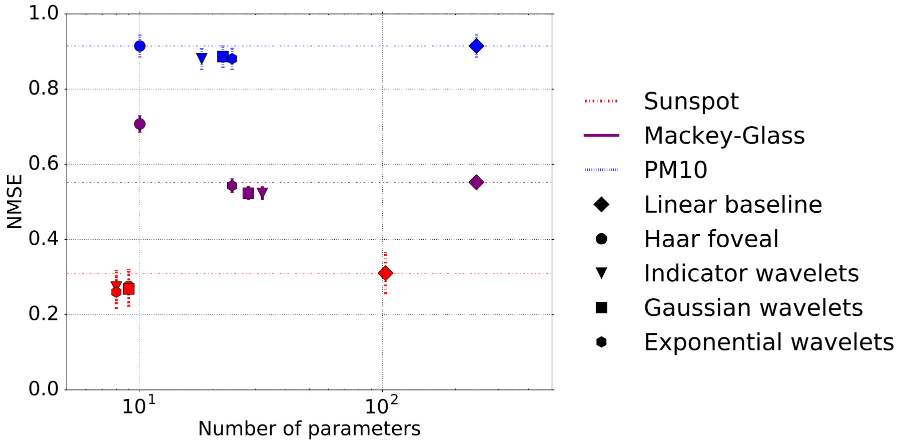

Figure 2.11: Forecasting results for real time series: NMSE (2.91) with respect to the number of parameters, or dimension of the subspace used. Horizontal lines correspond to the autoregressive baseline.

Figure 2.11 shows the forecasting results for all series, with respect to the number of parameters each method used. Observe how, in all cases, the foveal representations use at least one order of magnitude fewer parameters, with at least as good forecasting accuracy as the linear baseline.

More precisely, in the case of "Sunspot", all foveal representations use the same number of parameters, and obtain slightly better results than the linear baseline. In the case of "PM10", our foveal wavelets obtain slightly better results than both the foveal Haar representation and the linear baseline, but the gap is small. In the case of "Mackey-Glass", there is a much greater different between our foveal wavelets and the foveal Haar representation. This is due to Equation (2.90), where the past dependency is exact: to achieve good results, one needs to select a precise time in the past representation. While this is impossible for the foveal Haar representation, which lack such a "vertical" flexibility, it is possible with the proposed wavelets, by increasing the maximal polynomial order. Hence the larger number of parameters of our wavelets in this case.

2.5 CONCLUSION

This chapter introduces a new family of linear estimators of future values of a univariate time series, the foveal family. These estimators consist of a projection of the past of the time series onto a low dimensional subspace. This subspace is built by dilating a small set of functions at a fixed scale.

Three window functions are presented: the indicator window, the Gaussian window and the exponential window. By construction, the exponential window yields convolutions which can be computed recursively. The dyadic dilations match the power-law decay of the correlations with the future values, thus ensuring a small projection error. Crucially, this class of estimators has more expressivity than the foveal Haar family without losing its foveal nature.

Numerical experiments on synthetic and real time-series exhibit a long-range dependent behavior demonstrate that these new estimators allow to reach a lower mean-square error than the baseline autoregressive and Haar wavelet methods.

The foveal wavelets are designed to handle processes with cusp singularities, such as the fGn. Since this kind of singularities are often found in real time series, this bodes well for the wide application of these wavelets, as our numerical experiments have shown. However, for other types of processes, it might not necessarily be adapted. For instance, chirp singular processes, or so-called oscillating singularities [Arn+98], whose spectrum behaves like $\sin(1/\omega)$, do not leave the major part of their energy on the foveal coefficients. Therefore, these kinds of processes would be a major failure case of the proposed foveal wavelets.

3

NON-LINEAR PREDICTION WITH SPARSITY IN NEURAL NETWORKS

In Chapter 2, we have focused on linear prediction, considering the problem of forecasting future values under long-range dependence. Linear prediction methods form very strong baselines in many applications, for which non-linear methods rarely achieve significant gains. This is notably the case in finance, where linear methods are widely used despite large tails behaviors [BP09], which are evidence of a non-Gaussian behavior. The difficulty lies in taking advantage of non-Gaussian priors for forecasting purposes.

In this chapter, we investigate what priors on the stochastic processes allow non-linear prediction methods to bring a significant improvement upon linear predictions. A natural and related question concerns the nature of the forecasting algorithms that can be used for this purpose.

On the one hand, in signal processing, assuming the existence of a sparse decomposition has proven to be very effective to solve numerous tasks in an efficient manner [BDE09]. This hypothesis, which holds for many time-series of interest, including audio signals, allows non-linear methods to improve results with respect to linear algorithms for denoising [Mal08] or inpainting [Adl+12]. One may therefore try to use this assumption for time-series forecasting.

On the other hand, deep neural networks, which are non-linear algorithms, have shown exceptional results at generating time-series exhibiting such a sparse structure, notably audio time-series [Oor+16]. In particular, autoregressive networks [LM11] rely on the estimation of a conditional probability density, which is very close to the forecasting task. However, these networks do not explicitly incorporate the sparsity assumption in their architecture, and the exact priors they use remain poorly understood [Zha+17].

In this chapter, we propose a non-linear algorithm which can be interpreted as a neural network in order to forecast time-series. This algorithm explicitly uses sparse time-frequency decompositions in order to deliver a prediction.

Our demonstration is organized as follows. After a review of the background work in signal processing and deep learning, we show that a simple neural network is able to beat linear predictors on time-series with a sparse time-frequency structure. An analysis of the weights of such a network demonstrates that this sparsity is implicitly exploited. A simple mathematical model is derived to explain it. We then propose a sparse algorithm which exploits the sparsity of the time-series in an explicit fashion. It is extended in a foveal fashion in order to take into account longer temporal dependencies. Eventually, numerical experiments demonstrate that the proposed algorithm outperforms the neural network on synthetic time-series, and reaches close error rates on real time-series, thereby corroborating our modelization.

3.1 RELATED WORK

We begin this chapter by reviewing works related to the non-linear forecasting problem.

Section 3.1.1 tackles sparse priors. Indeed, the existence of a sparse decomposition is an important assumption which regularizes inverse problems such as denoising or inpainting. The resulting solutions are computed with non-linear algorithms, whose quality improves over the ones recovered by linear algorithms.

In Section 3.1.2, feed-forward neural networks are introduced. They implement parametric non-linear algorithms and have been used with success to model audio time-series. In particular, existing connections between these networks and sparse priors are reviewed.

3.1.1 *Sparsity*

3.1.1.1 *Sparse time-frequency decompositions*

DECOMPOSITION IN AN OVER-COMPLETE DICTIONARY Let x be a univariate time-series, possibly complex-valued. We view x as a vector of dimension the length of the time-series. Let $\mathcal{D} = \{\phi_p\}_{p\in\Gamma}$ be a family of unit-norm base signals, indexed by the set Γ. $\mathcal{D}$ is called a dictionary, and the signals ϕ_p atoms of this dictionary. $\mathcal{D}$ should be overcomplete, *i.e.* $|\Gamma|$ is larger than the dimension of the signal x. Let us introduce the analysis operator Φ and its dual synthesis operator Φ^*:

$$\begin{aligned} \Phi : x &\mapsto \{\langle x, \phi_p\rangle\}_{p\in\Gamma}, && (3.1)\\ \Phi^* : z &\mapsto \sum_{p\in\Gamma} z_p \phi_p. && (3.2) \end{aligned}$$

The signal x can be decomposed in the dictionary $\mathcal{D}$ if there exists a set of coefficients $z = \{z_p\}_{p\in\Gamma}$ such that

$$x = \sum_{p\in\Gamma} z_p \phi_p = \Phi^* z. \tag{3.3}$$

Due to the redundancy of the dictionary $\mathcal{D}$, such a decomposition is not unique. The exact decomposition (3.3) may be relaxed to an approximate one, provided the residual $x - \Phi^* z$ has a negligible norm compared to x.

Time-frequency dictionaries are made of functions which are meaningful in the time-frequency plane. Typically, p is a doublet containing the temporal and frequential locations of f_p. Among such dictionaries, local Gabor atoms [Gab46], wavelets [Mal08, chap. 4] and wavelet packets [CM91] are the most popular.

SPARSE DECOMPOSITION The decomposition (3.5) is called sparse if the coefficients $\{z_p\}_p$ are parsimonious, *i.e.* if few of them are non-zero. Such a sparsity is measured by the so-called ℓ^0-"norm", which counts the number of non-zero coordinates:

$$\|z\|_0 = \sum_{p \in \Gamma} \mathbb{1}_{|z_p|>0} \,. \tag{3.4}$$

Such a decomposition is also called a sparse latent code, or simply code, for the signal x. Empirical measurements show that audio signals admit a sparse decomposition in time-frequency dictionaries [Mal08].

Among all decompositions of signal x in an adapted dictionary $\mathcal{D}$, it is often desirable to find the sparsest one. Indeed, this parsimony can be understood as a proxy for simplicity, which should be favored from Ockham's razor. Sparse decomposition coefficients z can be found by solving a constrained optimization problem:

$$\begin{aligned} \min_z \quad & \tfrac{1}{2} \|x - \Phi^* z\|_2^2 \,, \\ \text{s.t.} \quad & \|z\|_0 \leq \eta \,, \end{aligned} \tag{3.5}$$

where $\eta > 0$ is a parameter to tune.

Unfortunately, Problem (3.5) is non-convex and NP-hard [Nat95], which makes it very hard to solve. Numerically, good heuristics exist to find approximate solutions, notably greedy pursuit methods [MZ93; PRK93].

ℓ^1 NORM PROXY One can relax Problem (3.5) into a convex optimization problem by replacing the ℓ^0 norm term by an ℓ^1 norm, yielding the Lasso [Tib96] or Basis Pursuit Denoising [CDS98]. In its unconstrained form, this relaxed convex problem reads

$$\min_z \frac{1}{2} \|x - \Phi^* z\|_2^2 + \frac{\lambda}{|\Gamma|} \|z\|_1 \,, \tag{3.6}$$

where $\lambda > 0$ is a hyperparameter, expressed in units independent of the size of the dictionary $|\Gamma|$.

The convexity of Problem (3.6) allows it to be solved up to machine precision. Further, the recovered solutions may actually be solutions of the original non-convex problem (3.5). [Fuc04; Tro06] prove that provided the existence of a sparse decomposition of the original signal and some technical conditions on the dictionary, parameter λ can be tuned so that solutions to (3.6) recover the correct atoms solution of (3.5).

ALGORITHMS The sparse decomposition problem (3.6) contains a smooth quadratic term and a non-smooth ℓ^1 term, which are both convex. It can be solved by proximal gradient descent [CP11], which is equivalent in this case to the iterative soft-thresholding algorithm (ISTA) [DDDM04].

From any initialization z^0 (typically, $z^0 = 0$), ISTA iterates a guess value z^k which becomes closer and closer to the minimizer of (3.6) with the update rule

$$z^{k+1} = \mathcal{F}_{\lambda\nu}\left[z^k - \tau\Phi(\Phi^* z^k - x)\right] , \tag{3.7}$$

where $\mathcal{F}_\epsilon$ is the soft-thresholding or shrinkage operator, defined as

$$\mathcal{F}_\epsilon(z) = \left\{ \frac{z_p}{|z_p|}(|z_p| - \epsilon)_+ \right\}_p , \tag{3.8}$$

and ν is a step size which should satisfy the bounds

$$0 < \nu < \|\Phi\Phi^*\|^{-1} \tag{3.9}$$

to ensure convergence.

ISTA is very simple to implement, but converges slowly. It can be accelerated by using some momentum, leading to the fast iterative soft-thresholding algorithm (FISTA) [BT09].

FISTA initializes $t^1 = 1$, $v^1 = z^0$, and iterates:

$$\begin{aligned} z^k &= \mathcal{F}_\lambda\left[v^k - \Phi(\Phi^* v^k - x)\right], && (3.10)\\ t^{k+1} &= \frac{1 + \sqrt{1 + 4(t^k)^2}}{2}, && (3.11)\\ v^{k+1} &= z^k + \frac{t^k - 1}{t^{k+1}}(z^k - z^{k-1}). && (3.12) \end{aligned}$$

The sequence of decomposition coefficients z^k converges in $O(k^{-2})$ to the optimum z^*, compared to a convergence in $O(k^{-1})$ for ISTA [BT09].

LEARNING THE DICTIONARY Without any prior knowledge on the nature of the dictionary in which x admits a sparse decomposition, it is temptable to learn this dictionary $\mathcal{D}$. This is possible provided that multiple examples $\{x_i\}_{1 \le i \le N}$ from the same class of images are available. This approach was pioneered by [OF96] on natural images and later applied on time-series [Gro+07]. The resulting sparse dictionary learning problem becomes:

$$\min_{\Phi^*, \{z_i\}} \sum_{i=1}^{N} \left\{ \frac{1}{2}\|x_i - \Phi^* z_i\|_2^2 + \frac{\lambda}{|\Gamma|}\|z_i\|_1 \right\} . \tag{3.13}$$

Problem (3.13) is non-convex and therefore difficult to solve in general. A widely-used heuristic is the K-SVD algorithm [AEB06], which alternates optimization over the codes $\{z_i\}$ and over the dictionary Φ^*. If one wants to use the codes $\{z_i\}$ for another task, for instance classification, the supervision feedback from this task can be used to influence the dictionary learning [MBP12; MG13].

3.1.1.2 *Regularization of inverse problems*

INVERSE PROBLEM Let us assume that we only have a partial measurement of signal x, possibly noisy:

$$y = Mx + \epsilon \, , \tag{3.14}$$

where M is a known linear measurement operator and ϵ a noise variable. Retrieving the actual variable x from the measurement y is called an inverse problem. Typically, M is rank-deficient, which makes the inverse problem ill-posed.

On the one hand, a standard linear regularization for Problem (3.14) is a Tikhonov regularization [TA77], which assumes that the ℓ_2 norm of x is small. Finding x is therefore cast as a minimization problem

$$\min_x \frac{1}{2}\|y - Mx\|_2^2 + \lambda\|x\|_2^2, \tag{3.15}$$

where $\lambda > 0$ ensures the existence of a solution, which admits a closed-form solution

$$\widetilde{x} = (M^*M + 2\lambda Id)^{-1}M^*y. \tag{3.16}$$

This solution is linear with respect to y, but it does not use any sparse prior information.

On the other hand, it is possible to exploit a parsimonious prior to regularize inverse problem (3.14). Indeed, let us assume that x admits a sparse decomposition in dictionary $\mathcal{D}$. Instead of attempting to find x, we now try to find a code z such that $x = \Phi^*z$. With a regularization term to promote sparse z-solutions, the minimization problem (3.15) is rewritten as

$$\min_z \frac{1}{2}\|y - M\Phi^*z\|_2^2 + \frac{\lambda}{|\Gamma|}\|z\|_1 \, . \tag{3.17}$$

The approximation $\widetilde{x}$ of x is then defined as:

$$\widetilde{x} = \Phi^*z. \tag{3.18}$$

In most cases, solutions to Problem (3.17) are non-linear in y, as they are obtained with *e.g.* FISTA.

Under appropriate assumptions on the measurement operator M and the dictionary Φ^*, solving the sparse regularized inverse problem (3.17) allows us to recover a vector very close to the original one, provided it is sparse enough [Dono6].

APPLICATIONS A wide range of signal processing tasks on time-series data can be cast as inverse problems. Under a parsimonious prior, these tasks can benefit from the sparse regularization approach (3.17). The resulting solutions, obtained with a non-linear algorithm, are typically better than the linear solutions obtained with a Tikhonov regularization (3.15).

Denoising is the simplest inverse problem, for which $M = Id$. For audio, [Fev+08] has shown that not only do sparse priors bring good results, but such sparse methods can even be improved by taking into account the time-frequency structure of the decomposition.

In blind source separation, also known as the cocktail party problem, the observation is assumed to be a linear combination of different signals, with mixing weights indicated by the matrix M, which has a horizontal shape. In order to recover each individual recording, [Plu+10] has shown that a sparse prior in time-frequency dictionaries can be exploited.

Closer to this work is the inpaiting problem: the measurement operator M becomes a masking one, as some temporal samples are lost. Again in this case, sparse priors can be exploited to restore the missing measurements, *e.g.* for audio [Adl+12]. Notice that inpainting is an interpolation problem, *i.e.* missing points are always surrounded by known values, whereas the forecasting problem, discussed in the next section, requires to extrapolate values outside of the convex hull of the known values.

3.1.1.3 *Forecasting applications*

We now review and discuss sparse decompositions methods which have been applied to the forecasting problem.

Several authors [Fak15; HFA18] have proposed to use a sparse decomposition method in a spirit close to the k-Nearest Neighbors (KNN) algorithm [HTF09]. Their idea consists in building two dictionaries: a dictionary of past values, which is the collection of all past windows $x_i(\leq t)$ in the training set, and a dictionary of target values, which is the collection of all corresponding future values $x_i(t + \Delta)$ in the training set. Given a new past window $x(\leq t)$, the idea consists in decomposing this past over the dictionary of past values, and using the same code to predict the future value thanks to the dictionary of future values. Contrary to our approach, this forecasting method does not exploit the sparse time-frequency properties of time-series.

In order to forecast electricity demand, a sparse coding method very close to our approach has been proposed by [YMH17]. In this work, a sparse dictionary is learned on the past windows, and the resulting codes are used to predict future values with a ridge regression. The resulting dictionaries contain oscillatory components, meaning that the original time-series have time-frequency properties, even if this property is not explicitly exploited. Algorithmically, our approach is very close to the work done in this paper, except that our dictionary is already known from priors on the signals. However, this paper does not provide any explanation of the ridge regression coefficients, while this chapter proposes an explicit link through the sparse inverse problem framework.

3.1.2 *Neural networks*

We provide a review of neural networks and their ability to model structured time-series, focusing on the emerging links between these networks and sparsity.

3.1.2.1 *Multi-layer perceptron*

Neural networks are parametric functions f_θ, defined in vector spaces $\mathbb{R}^T$. They are used to approximate a given function of interest, by tuning their parameters θ. In our case, these functions are real-valued, so the neural networks we consider are also real-valued.

Multi-layer perceptrons (MLP) [GBC16, Chapter 6] are the simplest neural networks. MLPs are a composition of multiple layers $j \in \{1, \ldots, J\}$, each of them consisting of an affine mapping and a pointwise non-linearity

$$v \mapsto \rho\left(W_j v + b_j\right) \ , \tag{3.19}$$

where W_j is a matrix of same input dimension as v and the bias b_j is a vector of same output dimension as W_j. ρ is a pointwise non-linearity, which is a convenient notation to denote the mapping of the same non-linear function ρ to all coordinates of a vector v:

$$\rho(v) = \left(\rho(v_i)\right)_i \ . \tag{3.20}$$

The most popular non-linearity is the rectified linear unit (ReLU)

$$\rho(u) = \max(u, 0) \ . \tag{3.21}$$

Note that this choice of non-linearity combined with the bias has a thresholding effect: the i^{th} coordinate in the output vector (3.19) returns a non-zero value if and only if the linear combination $(W_j v)_i$ is larger than the threshold $-(b_j)_i$.

In general, an MLP can thus be written as a combination of multiple layers

$$f_\theta(v) = W_J \rho\left(\ldots W_2 \rho\left(W_1 v + b_1\right) + b_2 \ldots\right) + b_J \ , \tag{3.22}$$

where

$$\theta = \{W_j, b_j\}_{1 \leq j \leq J} \tag{3.23}$$

is the set of parameters, which can be stacked as a large vector. The depth of the network denotes the number of layers of the network. Notice that in the last layer, no non-linearity is applied in order to regress values in an unbounded domain, but for other applications such a non-linearity might be present.

An MLP is trained by tuning the parameters or weights θ in order to minimize an objective function. This objective function $\mathcal{L}$ is typically defined as an average empirical loss ℓ between the desired value $f(x_i)$

of the network on examples $\{x_i\}_{1\leq i\leq N}$ and the approached value $f_\theta(x_i)$:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(f_\theta(x_i), f(x_i)) \tag{3.24}$$

$\mathcal{L}$ is a function of θ, which can be optimized by gradient descent methods. The value of the gradient for each weight W_j or b_j is found according to the backpropagation algorithm [Lin76; RHW86], which recursively applies the chain rule for differentiation on the different layers, starting with the last one.

Among all possible MLPs, MLPs with only one hidden layer (1-MLP), or shallow neural networks, are of particular interest. According to the universal approximation theorem [HSW89], they are able to approach functions up to an arbitrary precision, provided the size of the hidden layer is large enough and the weights are properly tuned [Pin15]. MLPs without hidden layers do not enjoy this property [MP69], while MLPs with more hidden layers clearly enjoy it as well. Although theoretical insights [ES16] and empirical results [LBH15] tend to indicate that deep networks have better approximation properties than shallow ones, the few number of layers in 1-MLPs makes it easier to analyze and understand the role of each layer.

3.1.2.2 *Autoregressive networks*

Autoregressive networks are designed to build a parametric model $\widetilde{p}_X$ of the probability density p_X of the process X. This model is constructed by exploiting the decomposition of the probability density into conditionals:

$$p_X(x) = \prod_t p_X(x(t+1)|X(\leq t) = x(\leq t)) \, . \tag{3.25}$$

Building on a stationarity assumption, autoregressive networks focus on approaching each term $p_X(x(t+1)|X(\leq t) = x(\leq t))$ by a single neural network f_θ:

$$f_\theta(x(\leq t), x(t+1)) = \widetilde{p_X}(x(t+1)|X(\leq t) = x(\leq t)) \, . \tag{3.26}$$

Thus, autoregressive networks do ultimately rely on a neural network trained for a forecasting task.

Autoregressive networks have recently achieved outstanding results in modeling time-series exhibiting a sparse time-frequency decomposition, such as speech or music. These networks notably include the WaveNet [Oor+16] and the SampleRNN [Meh+17]. The resulting generation capabilities are now already deployed in industrial applications [Oor+17; She+17].

In state-of-the-art autoregressive networks, the architecture of the predictive network f_θ does not correspond to a multi-layer perceptron. Instead f_θ is chosen either as a deep convolutional neural network [Oor+16] or a recurrent neural network [Meh+17], together with modern tricks such as residual connections [He+16] and multiplicative gates [HS97]. The resulting architecture is very complicated, with many sub-components whose importance is not well understood. As a consequence, it is unclear what operations these networks implement, and how these operations relate to the time-frequency properties of the signals.

3.1.2.3 *Sparsity and neural networks*

We review existing links which have been drawn between sparse decomposition methods and neural networks.

Artifical neural networks with binary activations were historically proposed as a computational model for biological neurons [MP43; Ros58]. Indeed, biological neurons exhibit very sparse activations [AL01; Len03]. Drawing inspiration from these observations, ReLU nonlinearities were introduced in order to promote sparsity of the neuron activations, instead of the then-common hyperbolic tangent nonlinearity [GBB11]. Thus, the role of the ReLU as a sparsifying nonlinearity is well-known.

In terms of signal processing, a striking bond connects the ISTA step (3.7) and a layer of an MLP (3.19). Indeed, provided the original input is concatenated to the intermediate output of the j-th layer, the ISTA step can be rewritten as an MLP layer with a particular choice of matrices and soft-thresholding operator $\mathcal{F}_{\lambda\nu}$, which is very close to a ReLU.

Building on this similarity, the LISTA algorithm [GL10] was proposed to learn an unrolled set of matrices to perform the same task. LISTA achieves the same reconstruction error and sparsity level as ISTA with much fewer iterations. Recent works suggest that this approach actually exploits a particular matrix factorization to speed-up the convergence of the decomposition [MB17].

Inspired by the same observation, a more direct connection between a sparse decomposition model and convolutional neural networks has been recently proposed [PRE17]. Under the proposed ML-CSC model, the signals are generated by a cascade of convolutional sparse coding (CSC) layers. The estimation of the decomposition coefficients in a pursuit scheme is then shown to be equivalent to the forward pass of a convolutional neural network. One should however note that this model is completely unsupervised, whereas most convolutional neural networks are trained and used in a supervised setting.

3.2 time-frequency sparsity can be exploited to forecast time-series

In this section, we show and explain how time-frequency sparsity can be exploited by a multi-layer perceptron with one hidden layer (1-MLP) to beat linear predictors. In 3.2.1, the considered forecasting framework is detailed. In 3.2.2, forecasting results with a 1-MLP are reported and an empirical analysis of its weights is performed, revealing a sparse structure. In 3.2.3, a simple cosine model for sparse time-frequency time-series is introduced. This model provides insights which are coherent with previous empirical results on how to exploit the sparsity for forecasting purposes.

3.2.1 *Forecasting framework*

Let $x = (x(t))_{t \in \mathbb{Z}}$ denote a real-valued time series. We assume that x is sampled from a stationary process X, $x \sim X$, such that each realization x admits a sparse decomposition in time-frequency dictionaries. In particular, this is the case for audio signals.

We consider the problem of forecasting $x(t+\Delta)$ from its past $x(\leq t) = \{x(s)\}_{s \leq t}$. We bound the past $x(\leq t)$ to a size τ, leading to a sliding window denoted with an abuse of notation

$$x(\leq t) := \{x(s)\}_{t-\tau < s \leq t} \in \mathbb{R}^{\tau}. \tag{3.27}$$

We look for a forecasting function $\widetilde{f}$, which may depend on parameters, which minimizes the mean-square forecasting error at a lag Δ for the process X

$$\min_{\widetilde{f}} \mathbb{E} \left| X(t+\Delta) - \widetilde{f}(X(\leq t)) \right|^2 . \tag{3.28}$$

Theoretically, the minimizer of this quantity is the expected value $\mathbb{E}[X(t+\Delta) | X(t-s), 0 \leq s < \tau]$ of the future value conditioned on its past. Therefore, $\widetilde{f}$ should approach this conditional expected value as best as possible from a finite number of samples, among a given set of functions.

As noted in 3.1.2, neural networks have recently achieved very good performance to model conditional probability densities — a task more complex than approaching conditional expected values. Further, 1-MLP are the simplest universal neural networks, which make them easier to analyze and understand. This explains why we use 1-MLPs as a baseline for non-linear prediction:

$$\widetilde{f}(x(\leq t)) = W_2^T \rho(W_1 x(\leq t) + b_1) + b_2, \tag{3.29}$$

with $W_1 \in \mathbb{R}^{H \times \tau}$, $b_1 \in \mathbb{R}^H$, $b_2 \in \mathbb{R}$, $W_2 \in \mathbb{R}^H$ and ρ being the ReLU non-linearity (3.21).

The linear baseline model we consider consists in autoregressive linear predictors, defined as

$$x(\leq t) \mapsto \alpha^T x(\leq t) + \beta, \tag{3.30}$$

where $\alpha \in \mathbb{R}^{\tau}$ and $\beta \in \mathbb{R}$.

3.2.2 *Empirical observation: sparse time-frequency decomposition in 1-MLP*

We show experimentally that a 1-MLP is able to beat linear predictors on the forecasting task (3.28). The analysis of the weights of a trained network reveals that its first layer performs a sparse time-frequency decomposition.

3.2.2.1 *Experimental setup*

Experiments are performed with time series extracted from the VCTK dataset [Yam12], which consists in recordings of native English speakers uttering short sentences. We refer to Section 3.4 for a detailed presentation of this dataset and for all experimental methods. In this part, the prediction is performed for $\Delta = 2.5$ ms.

Figure 3.9 shows that an autoregressive linear predictor (3.30) has a relative root mean-square error (RMSE) (3.68) above 1. On the contrary, a neural network with a single hidden layer implemented with ReLU non-linearities (3.29) produces a relative RMSE below 1. This shows the non-linearities reduce the forecasting error significantly compared to the standard deviation of the target. To understand this improvement, we study in more details the calculations performed by the neural network.

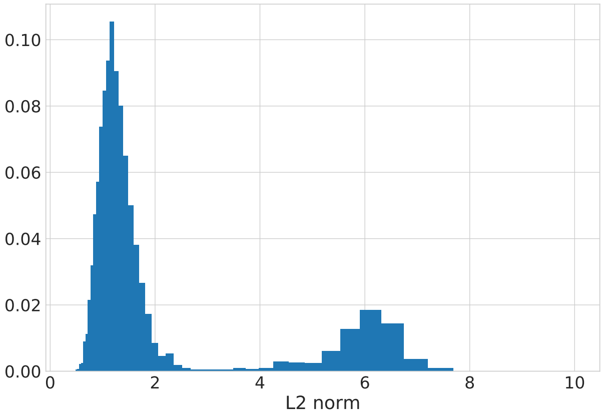

Figure 3.1: Histogram of the ℓ^2 norm of the weights of the first layer of the neural network (3.29). The weights whose norm is about 6 appear to be responsible for the prediction capabilities of the network.

In Figure 3.1, a small subset of input weights in the matrix W_1, corresponding to 6.8% of the total number of inputs weights, has an ℓ^2 norm significantly larger than the rest. Restricting W_1 to these coefficients yields the same error as using the full network, which shows that the predictive power of the network almost exclusively relies on these weights. We thus restrict our analysis to the restriction of W_1 to these $N_S = 278$ hidden units.

3.2.2.2 *Sparse time-frequency behavior of the network*

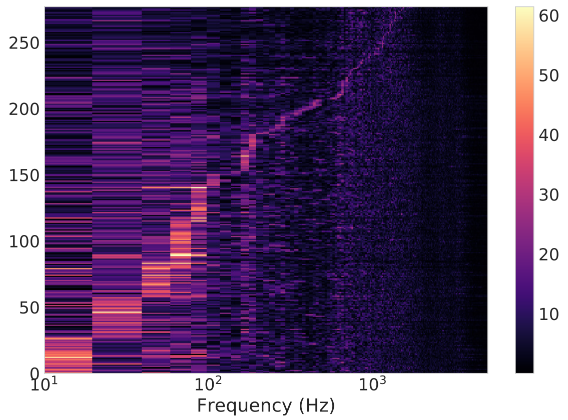

Figure 3.2: Discrete Fourier Transform of the rows of the input neural network matrix W_1. Each Fourier transform appears to select a certain frequency corresponding to brighter coefficients.

Figure 3.2 shows the row-wise discrete Fourier transform of the weights of the matrix W_1. We did a proper permutation of the hidden units to restore the continuity across rows of the maxima of this Fourier transform. The weights are partly organized into blocks which are mostly sensitive to a certain frequency, especially for frequencies below 10^3 Hz.

The output $W_1 x(\leq t) + b_1$ is then filtered by the ReLU non-linearity ρ which produces a highly sparse output. This sparsity is measured by the average proportion of non-zero coefficient at the output:

$$\bar{s} = \frac{1}{N_S} \sum_{k=1}^{N_S} \mathbb{E}_{x(\leq t)} \left[\mathbb{1}_{\rho(W_1 x(\leq t) + b_1)_k > 0} \right] \tag{3.31}$$

Empirically, we measure an average of $\bar{s} \approx 7\%$ non-zero coefficients after the ReLU, which are then filtered by the operator W_2.

Figure 3.3 shows a histogram of the output weights W_2. Their absolute value is concentrated around 0.25, with flipping signs. It indicates that all hidden units are playing a similar role in the prediction, but within different frequency channels.

This analysis shows that the multilayer neural network filters the input signal within different frequency channels with the operator W_1 and outputs a highly sparse array of coefficients which are linearly combined by W_2 with weights of approximately equal magnitude but flipping signs.

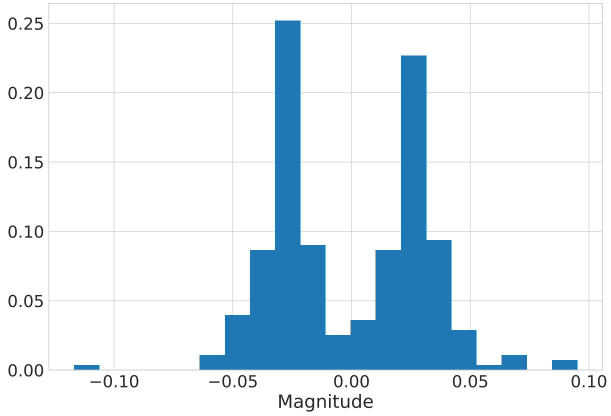

Figure 3.3: Histogram of the individual entries of the output weight $W_2 \in \mathbb{R}^{N_S}$ (3.29).

3.2.3 *Analysis of the empirical results*

We introduce a mathematical framework which incorporates the mechanisms of the neural network studied in 3.2.2. To better understand the difficulty of this forecasting problem, we begin with a simple signal model obtained by summing cosine functions of unknown frequencies. We shall see that linear predictors perform poorly on these signals. This model provides a general strategy on how to exploit the time-frequency sparsity to forecast future values.

3.2.3.1 *Cosine Model*

We introduce a simple cosine signal model, which is sparse in the time-frequency domain. Linear predictors fail on this model.

DEFINITION Let us consider signals which are sums of K cosine functions of unknown frequencies and amplitudes

$$x(u) = \Re \left\{ \sum_{k=1}^{K} a_k e^{i\omega_k u} \right\} , \tag{3.32}$$

with K small and where

$$a_k = r_k e^{i\phi_k} , \tag{3.33}$$

with $r_k > 0$, $\phi_k \in [0, 2\pi)$ and $\omega_k \in [\omega_{\text{min}}, \omega_{\text{max}}] \subset [0, \pi)$. The values of ϕ_k and ω_k are randomly drawn from *i.i.d.* uniform distributions over their supports, while the values of r_k are *i.i.d.* and drawn from an exponential distribution over $(0, +\infty)$. The resulting process from which x is drawn is therefore stationary.

This cosine model is a very crude model of audio time-series, as it only contains pure harmonics. In particular, it does not take into account the temporal variations of the envelopes a_k, and the temporal variations of the frequencies. Therefore, it should only be seen as a localized model of audio time-series.

FAILURE OF LINEAR PREDICTION A linear autoregressive predictor yields a large error on this stationary process.

Indeed, for each fixed set of frequencies $\{\omega_k\}_k$, one can find a set of linear weights giving good prediction results. This amounts to linear predictive coding (LPC) [O'S88] with an all-pole filter, which is a well-known local model of audio time-series [GME11].

For a single frequency ω, one can derive analytical weights as:

$$\cos(\omega(t+\Delta)) = 2\cos(\omega\Delta)\cos(\omega t) - \cos(\omega(t-\Delta)). \tag{3.34}$$

As seen on this toy example, the resulting predictive weights will depend on the given frequency ω. This remains true for a whole set of frequencies $\{\omega_k\}_k$.

For the cosine process we consider, each realization will express a different set of frequencies. Therefore, there is no single set of linear weights which predicts all realizations, and linear autoregressive predictors will yield a large average prediction error.

3.2.3.2 *Exploiting the sparse decomposition*

We investigate the cosine model to provide insights on how to exploit the time-frequency sparsity for forecasting purposes.

Let $\widehat{x}(\omega)$ denote the Fourier transform of a signal $x(u)$,

$$\widehat{x}(\omega) = \sum_u x(u) e^{-i\omega u} . \tag{3.35}$$

Let x^a be the analytical part of x carrying its non-negative frequencies, which is defined in the Fourier domain as

$$\widehat{x^a}(\omega) = 2\widehat{x}(\omega)\mathbb{1}_{\omega\geq 0} . \tag{3.36}$$

The cosine model defined in Equation (3.32) may be rewritten as

$$x(u) = \Re\left(x^a(u)\right) = \Re\left(\int_0^\pi \widehat{x^a}(\omega) e^{i\omega u} d\omega\right) , \tag{3.37}$$

where the analytical part x^a is defined in the Fourier domain as a sum of Dirac delta functions:

$$\widehat{x^a} = \sum_{k=1}^{K} a_k \delta_{\omega_k} . \tag{3.38}$$

Equations (3.37)-(3.38) indicate that x^a has a sparse decomposition over the Fourier family of non-negative frequencies $\{e^{i\omega\cdot}\}_{\omega\in[0,\pi]}$. Moreover, there exists a linear operator mapping the coefficients $(a_k)_k$ of the sparse decomposition (3.38) to the value $x(t+\Delta)$, by taking $u = t+\Delta$ in Equation (3.37):

$$x(t+\Delta) = \Re\left(\int_0^{\pi} \widehat{x^a}(\omega) e^{i\omega(t+\Delta)} d\omega\right) . \qquad (3.39)$$

As a consequence, a natural strategy to obtain a forecast of $x(t+\Delta)$ is to retrieve such a sparse decomposition of x^a. One can then apply a linear operator derived from (3.39), which is independent of frequencies analyzed in x^a.

This strategy is coherent with the empirical analysis of the 1-MLP we have performed: it is likely that the 1-MLP exploits these ideas for forecasting. Our goal is now to derive an algorithm involving minimal learning that fully exploits this sparse time-frequency prior.

3.3 SPARSE FORECASTING ALGORITHM

In the previous section, we have shown that non-linear algorithms can outperform linear predictors on time-series exhibiting a sparse time-frequency decomposition. Based on a simple cosine model, we have outlined a general strategy to exploit this time-frequency decomposition without learning.

In this section, we derive a sparse time-frequency forecasting algorithm based on the aforementioned strategy. The forecasting problem is viewed as an inverse extrapolation problem in Section 3.3.1, which is difficult to solve due to the causality constraint. In the following Section 3.3.2, we show that windowing the input allows us to exploit the sparse time-frequency prior to regularize this inverse problem. Multiple time-frequency dictionaries adapted to the cosine model are then proposed in Section 3.3.3. In order to be amenable to more complex time-series than the cosine model, we introduce a foveal multiscale extension in Section 3.3.4.

3.3.1 *Description of the problem*

3.3.1.1 *Inverse problem formulation*

In order to forecast $x(t+\Delta)$, we only have access to past values $\{x(u)\}_{t-\tau<u\leq t}$ due to the causality constraint. Let $\mathcal{M}_t$ denote the masking operator relative to the forecasting problem:

$$\mathcal{M}_t x = x \times \mathbb{1}_{(t-\tau,t]} . \qquad (3.40)$$

We have access to $\mathcal{M}_t x$, from which we would like to extrapolate $x(t+\Delta)$. This is an inverse problem, which is ill-posed in general. We would like to use a sparse prior in order to regularize it.

Following the cosine model presented in Equation (3.32), a natural prior is the existence of a sparse decomposition of the analytical part

x^a of x. Rewriting Equation (3.37) in abstract terms, there exists a dictionary $\{f_p\}_{p\in\Gamma}$ such that

$$x^a = \sum_{p\in\Gamma} z_p f_p, \tag{3.41}$$

with $\|z\|_0 \ll |\Gamma|$. We focus on the analytical part as it allows us to extract the phases with a simple Lasso formulation in z, using complex numbers. A tentative regularized inverse problem could therefore be formalized as

$$\min_z \left\| (\mathcal{M}_t x)^a - \sum_{p\in\Gamma} z_p \mathcal{M}_t f_p \right\|_2^2 + \lambda \|z\|_1, \tag{3.42}$$

with a given regularization parameter $\lambda > 0$.

3.3.1.2 *Causality issue*

Problem (3.42) is ill-posed because of the causality constraint. Indeed, by linearity of $\mathcal{M}_t$, Equation (3.41) implies

$$\mathcal{M}_t(x^a) = \sum_{p\in\Gamma} z_p \mathcal{M}_t f_p. \tag{3.43}$$

However, the analytical part operator and the masking operator do not commute

$$(\mathcal{M}_t x)^a \neq \mathcal{M}_t(x^a), \tag{3.44}$$

so that a sparse decomposition retrieved by Problem (3.42) would not retrieve the correct coefficients for the decomposition of x^a.

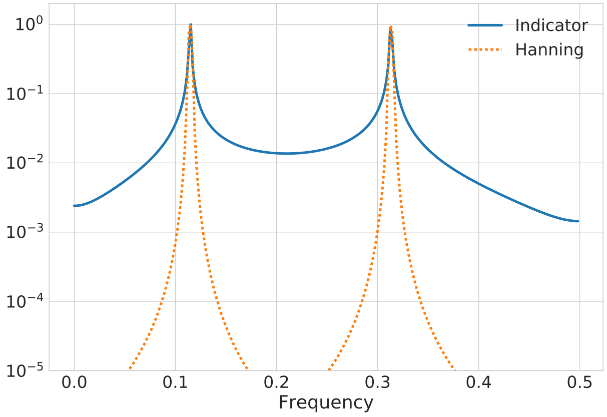

Figure 3.4: Effect of the choice of the window on the stationary cosine model (3.32) with two coefficients in the Fourier domain (positive frequencies only). Windowing the signal yields sparse Fourier coefficients.

The non-commutativity of the two operators is a consequence of the discontinuity of the indicator window $\mathbb{1}_{(t-\tau,t]}$, which has a slow

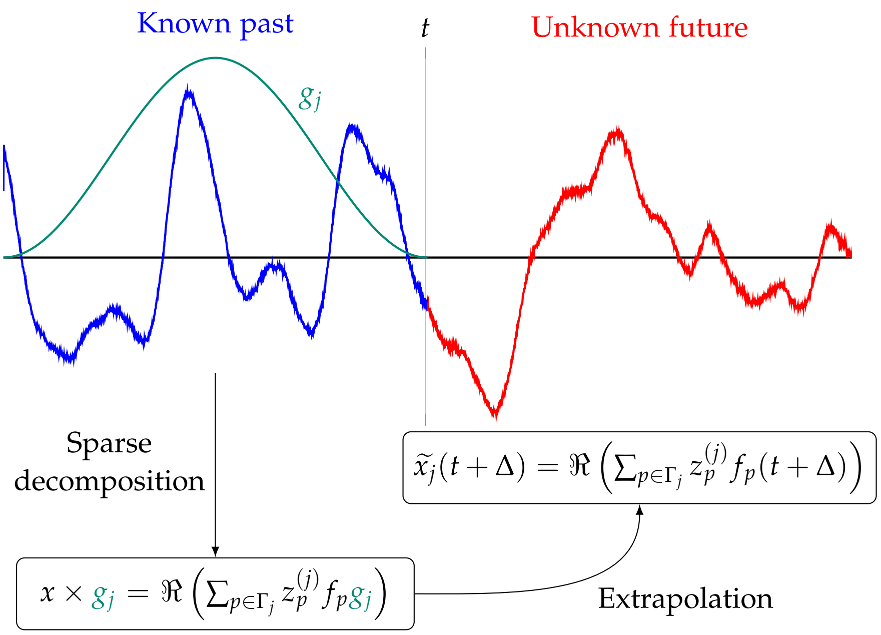
Known past
t
Unknown future
g_j
Sparse decomposition
$\tilde{x}_j(t+\Delta) = \Re\left(\sum_{p\in\Gamma_j} z_p^{(j)} f_p(t+\Delta)\right)$
$x \times g_j = \Re\left(\sum_{p\in\Gamma_j} z_p^{(j)} f_p g_j\right)$
Extrapolation

Figure 3.5: Sparse forecasting algorithm. The windowed past is decomposed onto a dictionary, out of which the future value is extrapolated.

decay in the Fourier domain. Indeed, by convolution, $x \times \mathbb{1}_{(t-\tau,t]}$ is not sparse in the Fourier domain. In particular, there are significant overlaps between negative and positive frequencies, as well as between the different frequencies ω_k, as displayed in Figure 3.4. Hence the non-commutativity of the operators.

Therefore, we cannot exploit the analytical part of $\mathcal{M}_t x$ in order to get information on the analytical part of x^a, from which we could make the prediction. Further, the overlaps within the analytical part degrades the estimation of the frequencies ω_k and coefficients a_k. One needs to mitigate this causality issue in order to exploit the sparsity of the signal x for forecasting.

3.3.2 *Causal sparse decomposition*

We propose an approach to mitigate the causality constraint in order to exploit the sparse time-frequency prior of the cosine model. Figure 3.5 summarizes the proposed approach. The past signal is first artificially windowed in a causal fashion. A sparse decomposition of this windowed signal is obtained in a windowed time-frequency dictionary. The future value is obtained by removing the window from the dictionary's atoms and evaluating the value of their linear combination at the future point $t+\Delta$.

SPARSIFYING CAUSAL WINDOW Let us assume that $\tau = 2^j$ for some positive integer j. In order to build a sparse time-frequency representation of the past signal $x \times \mathbb{1}_{(t-\tau,t]}$, we multiply it by a causal and regular window g_j with support in $(t-\tau, t]$. We thus define a new windowing operator

$$\mathcal{G}_{t,j}x(u) = x(u)g_j(u) =: y_j(u) \; . \tag{3.45}$$

Notice that $\mathcal{G}_{t,j}$ and the previous masking operator $\mathcal{M}_t$ commute, so that we can effectively access $\mathcal{G}_{t,j}x$ from empirical data.

We assume that g_j is smooth and symmetric with respect to $t - 2^{j-1}$ in order to ensure that its Fourier transform has a fast decay. In the following, g_j is a Hanning window:

$$g_j(u) = \sin^2\left(\frac{\pi(u-t)}{\tau - 1}\right) \; . \tag{3.46}$$

Figure 3.4 compares the Fourier transform of the signal windowed with the indicator function $\mathcal{M}_t x$ and of the windowed signal $\mathcal{G}_{t,j}x$ computed with a Hanning function. Notice that $\mathcal{G}_{t,j}x$ is much sparser in Fourier than $\mathcal{M}_t x$. In particular, as proved in Section 3.3.3.1, there is no overlap between the positive and negative frequencies, provided that the frequencies are larger than some threshold. Therefore, it holds that

$$\left(\mathcal{G}_{t,j}x\right)^a \approx \mathcal{G}_{t,j}(x^a) \; . \tag{3.47}$$

Using the new masking operator $\mathcal{G}_{t,j}$ to circumvent the causality problem, it becomes possible to regularize the inverse problem with the sparse prior.

WINDOWED DICTIONARY We assume that x^a admits a sparse representation in a family of complex-valued functions $\{f_p(u)\}_{p\in\Gamma_j}$. Γ_j is the set of indices for this family, which depends on the parameter j.

We associate to the family $\{f_p(u)\}_{p\in\Gamma_j}$ a complex-valued dictionary $\mathcal{D}_j$ supported over past values and windowed by g_j:

$$\mathcal{D}_j \;=\; \{\phi_{j,p}\}_{p\in\Gamma_j} \; , \tag{3.48}$$

$$\phi_{j,p}(u) \;=\; f_p(u)g_j(u) \; . \tag{3.49}$$

We obtain a time-frequency dictionary by choosing f_p to be a complex exponential whose frequency varies with p. However, we shall later see that better results are obtained by incorporating polynomial components.

SPARSE CAUSAL DECOMPOSITION We compute a sparse decomposition of y_j onto the dictionary $\mathcal{D}_j$:

$$y_j^a(u) = \sum_{p\in\Gamma_j} z_p^{(j)} \phi_{j,p}(u), \tag{3.50}$$

for $u \in (t - 2^j, t]$. The complex decomposition coefficients $z_p^{(j)}$ should satisfy $\|z^{(j)}\|_0 \ll |\Gamma_j|$. They are found by solving a basis pursuit problem:

$$\min_z \left\| y_j^a - \sum_{p \in \Gamma_j} z_p^{(j)} \phi_{j,p} \right\|_2^2 + \lambda \|z\|_1, \tag{3.51}$$

where $\lambda > 0$ is a hyperparameter.

Since $y_j^a \approx \mathcal{G}_{t,j}(x^a)$ and $\phi_{j,p} = \mathcal{G}_{t,j} f_p$, under appropriate sparsity conditions, the coefficients $z_p^{(j)}$ obtained by solving Problem (3.51) are the ones corresponding to the actual decomposition of x^a [Mal08].

FORECAST Since $\phi_{j,p}(u) = f_p(u)\, g_j(u)$ we eventually derive from the sparse expansion (3.50) a sparse expansion of $x(u)$ for $u \geq t$ which is given by

$$\widetilde{x}_j(u) = \Re \left(\sum_{p \in \Gamma_j} z_p^{(j)} f_p(u) \right). \tag{3.52}$$

In particular, for the target time $u = t + \Delta$ we get the estimator

$$\widetilde{x}_j(t + \Delta) = \Re \left(\sum_{p \in \Gamma_j} z_p^{(j)} f_p(t + \Delta) \right). \tag{3.53}$$

3.3.3 *Choice of dictionary*

We now show that the dictionary $\mathcal{D}_j$ can be defined with functions $f_p(u)$ which are oversampled Fourier exponentials, or oversampled Fourier exponentials with polynomial terms.

3.3.3.1 *Windowed Fourier dictionary*

Let us first study in detail the effect of windowing on the sum of cosine model. Inserting (3.32) in (3.50) with expression proves that for all $\omega \in [-\pi, \pi]$:

$$\widehat{y}_j(\omega) = \sum_{k=1}^{K} \frac{\rho_k}{2} \left(e^{i\phi_k} \hat{g}_j(\omega - \omega_k) + e^{-i\phi_k} \hat{g}_j(\omega + \omega_k) \right). \tag{3.54}$$

Extracting the analytical part as in Equation (3.37) means restricting the Fourier support to positive frequencies. This is effective only if the negative and positive frequencies have a negligible interaction. Since $\omega_k \geq \omega_{\min}$, it is sufficient to have

$$|\hat{g}_j(0)| \gg |\hat{g}_j(2\omega_{\min})|. \tag{3.55}$$

When (3.55) holds, then for all $\omega \geq 0$,

$$\widehat{y}_j(\omega) \approx \frac{1}{2} \sum_{k=1}^{K} a_k \hat{g}_j(\omega - \omega_k), \tag{3.56}$$

which translates in the time domain as

$$y_j(u) = x(u)\, g_j(u) \approx \Re\left(\sum_{k=1}^{K} a_k g_j(u) e^{i\omega_k u}\right). \tag{3.57}$$

This approximate analytical decomposition, proving Equation (3.47), motivates the choice of $f_p(u) = e^{i\xi_p u}$, for appropriate frequencies ξ_p.

Let us emphasize that the analyticity property is essential to obtain good forecasting results. Indeed, the forecasting problem boils down to a phase estimation which requires the use of analytical functions for precision.

3.3.3.2 *Oversampled windowed Fourier dictionary*

Following (3.57) we may choose

$$f_p(u) = c_p e^{i\xi_p u}, \tag{3.58}$$

where c_p is a renormalization factor ensuring $\|\phi_{j,p}\|_2 = 1$. To obtain a complete set of time-frequency atoms $c_p e^{i\xi_p u} g_j(u)$, the positive frequencies $\{\xi_p\}_{p\in\Gamma}$ must sample $[0,\pi]$ at intervals smaller than 2^{-j}, with an oversampling factor $P \geq 2$:

$$\xi_p = \frac{2\pi p}{P 2^j} \text{ for } 0 \leq p \leq P2^{j-1}\,. \tag{3.59}$$

The inner products $\langle y_j^a, \phi_p\rangle$ are efficiently computed with a Fast Fourier Transform (FFT) algorithm.

3.3.3.3 *Polynomial windowed Fourier dictionary*

To compute the sparse expansion coefficients a_k in (3.56) with a good precision, we typically need to use a large large oversampling factor $P \geq 16$ which increases computations. We now show that the dictionary size can be reduced with polynomial terms.

Let ξ_k denote the frequency on a discrete Fourier grid (possibly oversampled as well) closest to a frequency ω_k. If $\epsilon_k = \omega_k - \xi_k$ then for all $u \in (t-2^j, t]$, inserting the expansion

$$e^{i\epsilon u} = 1 + \epsilon u + \cdots + \frac{\epsilon^m u^m}{m!} + o(\epsilon^m) \tag{3.60}$$

in Equation (3.57), we derive that

$$y_j(u) \approx \Re\left(\sum_{k=1}^{K}\sum_{n=0}^{m} a_k \frac{\epsilon_k^n}{n!} u^n g_j(u) e^{i\xi_k u}\right). \tag{3.61}$$

Therefore, y_j^a admits a sparse decomposition in the dictionary

$$\left\{\phi_{j,n,\ell}(u) := c_{n,\ell} u^n g_j(u) e^{i\frac{2\pi\ell}{P2^j}u}\right\}_{0\leq n\leq m, 0\leq \ell\leq P2^{j-1}}, \tag{3.62}$$

where $c_{n,l}$ is a normalization constant ensuring $\|\phi_{j,n\ell}\|_2 = 1$. For this dictionary, we shall write $p = (n, \ell)$ and

$$f_p(u) = c_{n,\ell} u^n e^{i\frac{2\pi\ell}{P2^j}u}. \tag{3.63}$$

The inner products $\langle y_j^a, \phi_{j,p}\rangle$ can also be computed with an FFT thanks to the following equation:

$$\langle y_j^a, \phi_{j,p}\rangle = \langle y_j^a, c_p u^n g_j e^{i\frac{2\pi\ell}{2^j}u}\rangle = \langle c_p u^n g_j y_j^a, e^{i\frac{2\pi\ell}{2^j}u}\rangle. \tag{3.64}$$

Figure 3.6 (left) shows two atoms with the same frequency, but varying polynomial order n.

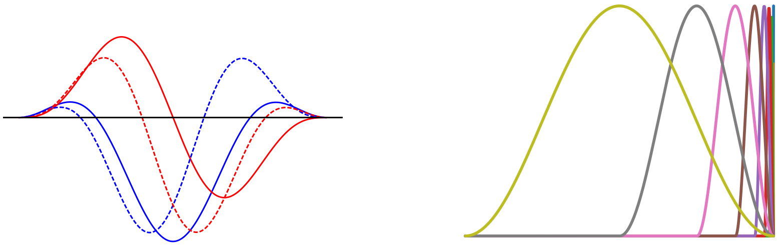

Figure 3.6: **Left**: Gabor atoms $\phi_{n,j}(u)$ in (3.62) for polynomial order $n = 0$ (full curve) and $n = 1$ (dotted curve). The blue curves are the real parts, and the red curves the imaginary parts. **Right**: Windows g_j at multiple dyadic scales 2^j. Small scales carry the information closer to the boundary and thus retrieve parts of the signal which have been lost by larger scales.

3.3.4 *Foveal multiscale extension*

The sparse forecasting algorithm we have previously proposed assumes a cosine model, which is very simple. Moreover, a single window g_j of size 2^j nearly eliminates signal information over the temporal interval $(t - 2^{j-1}, t]$, as shown in Figure 3.6 (right). We therefore propose to combine windows at multiple scales $1 \leq j \leq J$ in order to recover the missing information and to be amenable to more complex processes than the cosine model.

As we tackle a forecasting task, following Chapter 2, it is natural to consider a foveal multiscale extension, where we take into account multiple windows g_j near the temporal border t. In the following, we propose two types of such multiscale extensions, a weak one combining predictions and a strong one combining dictionaries, shown in Figure 3.7.

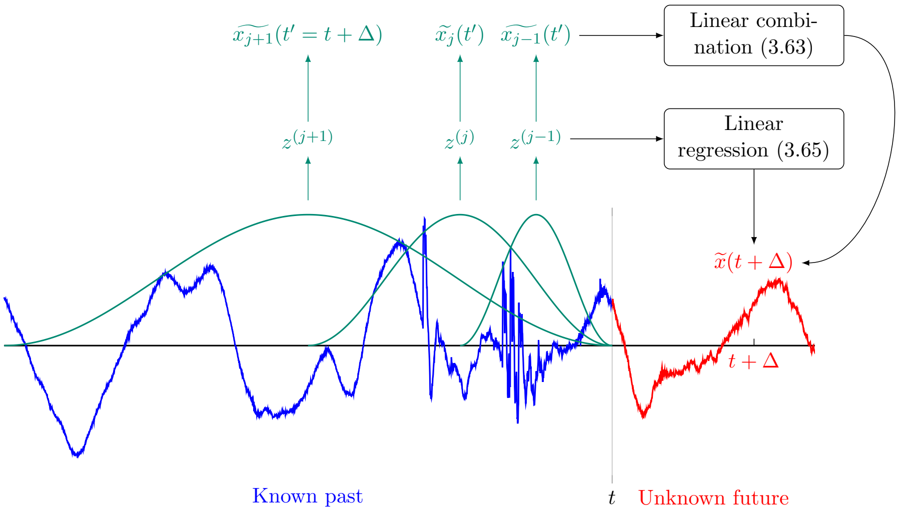

Figure 3.7: Foveal multiscale extension of the sparse forecasting algorithm. The weak extension (3.65) linearly combines the predictions $\widetilde{x}_j(t+\Delta)$ at each scale, while the strong extension (3.67) makes a linear prediction based on the concatenated codes $z^{(j)}$.

3.3.4.1 *Weak combination*

We propose a weak combination of the predictions performed at each scale $\widetilde{x}_j(t+\Delta)$, using a weighted linear combination:

$$\widetilde{x}(t+\Delta) = \sum_{j=1}^{J} \alpha_j \widetilde{x}_j(t+\Delta),$$

with weights $\alpha_j \in \mathbb{R}$. When $j=1$ we set $\tilde{x}_1(t+\Delta) = x(t)$, which keeps track of the last value. Inserting (3.53) gives

$$\tilde{x}(t+\Delta) = \sum_{j=1}^{J} \alpha_j \Re \underbrace{\left(\sum_{p \in \Gamma_j} z_p^{(j)} f_p(t+\Delta) \right)}_{\widetilde{x}_j(t+\Delta)}. \tag{3.65}$$

The weights α_j are learnt using training time-series in order to minimize the forecasting mean-square error (MSE), which is a convex problem in α_j. This recombination of the fixed predictors defines a data-dependent predictor which can be learned by a small amount of data because it only depends upon J variables.

3.3.4.2 *Strong combination*

We now propose a stronger combination of the different scales. This combination merges the decomposition vectors $z^{(j)}$ in a single vector and directly learns the linear operator L mapping this vector to the output prediction.

By observing that Equation (3.65) can be interpreted as a real-valued linear operator

$$\{z^{(j)}\}_{1\leq j\leq J} \mapsto \widetilde{x}(t+\Delta), \tag{3.66}$$

whose coefficients are specified by α_j and $f_p(t+\Delta)$, we can generalize this operator as

$$L : \{z^{(j)}\}_{1\leq j\leq J} \mapsto \sum_{j=1}^{J} \Re\left(\sum_{p\in\Gamma_j} w_{p,j} z_p^{(j)}\right) = \widetilde{x}(t+\Delta), \tag{3.67}$$

for arbitrary complex coefficients $w_{p,j}$.

The coefficients $\{\{w_{p,j}\}_{p\in\Gamma_j}\}_{1\leq j\leq J}$ can be learnt by minimizing the MSE of the prediction. This is a convex problem which can be solved efficiently. It allows more flexibility than the weak combination of predictions. Nevertheless, compared with (3.65) this approach requires to estimate much more parameters. For an oversampling P and a polynomial order m the matrix L has $\mathcal{O}(2^J Pm)$ parameters instead of J in (3.65).

3.4 NUMERICAL BENCHMARK

In this section, we compare the performance of the proposed algorithm to the baseline non-linear predictor (multi-layer perceptron). We show that it is able to outperfom it on synthetic time series, and to match its performance on real time series.

3.4.1 *Synthetic time-series*

We first consider the prediction problem for the cosine sum model (3.32). The Gabor dictionary $\mathcal{D}_j$ is computed at a single large scale $2^j = 512$. In this case, increasing the window size 2^j improves results because the signal is stationary. Using smaller scale windows does not reduce the error. We shall compare the impact of the maximum polynomial order m and the oversampling P. The sparse forecasting is compared with a linear regression and a one hidden layer neural network.

3.4.1.1 *Experimental setup*

The random process x is defined according to (3.32) with $K = 2$ frequencies, and $a_k = \rho_k e^{i\phi_k}$ where ϕ_k is a random variable uniformly distributed in $[0, 2\pi]$ and ρ_k has an exponential distribution of mean 1. The two frequencies ω_1 and ω_2 have a uniform distribution over $[1/10, 9\pi/10]$. with rejection sampling when $|\omega_1 - \omega_2| \geq 1$. The future lag Δ is set to 1. The lasso hyperparameter λ is fixed to 100 for all dictionaries. The oversampling parameter P is chosen within the set $\{1, 2, 4, 8\}$.

We use the FISTA algorithm [BT09] exposed in Section 3.1.1.1 to perform the minimization problem (3.51). The step size ν is equal to 0.9 of the upper bound on the step size ensuring convergence, defined in Equation (3.9). The numerical implementation is performed in python and relies on the Tensorflow library [Aba+16], implemented on Graphical Processing Units (GPUs) to speed up computations. In all numerical experiments, unless mentioned otherwise, we use 10^3 iterations which provides a convergence of the loss up to $\pm 10^{-5}$.

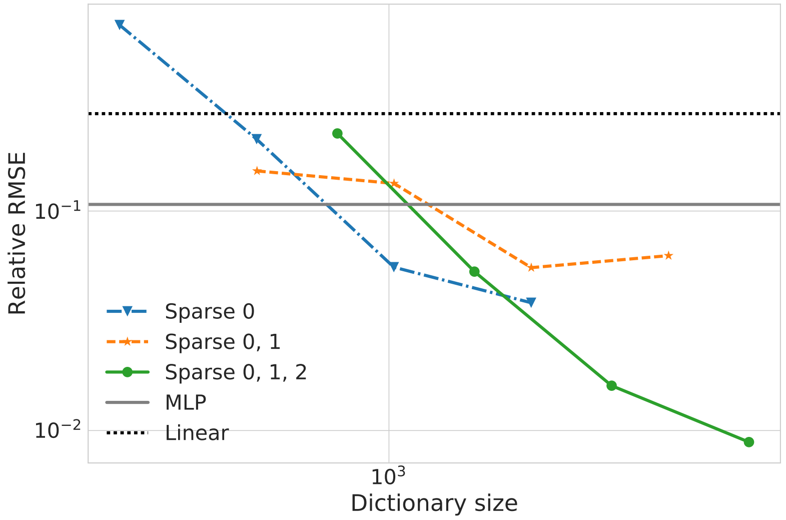

Figure 3.8: Relative RMSE results on the cosine model (3.32) (lower is better). The top line gives the error of the linear estimator. Below is the error of the neural network. Each sparse $0, \ldots, m$ curve corresponds to a dictionary computed with polynomials up to an order m. The horizontal axis specifies the size of the dictionary, which also depends upon the oversampling factor P.

For fair comparison, baseline methods are also computed on an input of size $2^9 = 512$. The neural network has one hidden layer of 4096 units, with ReLU activations and linear readouts. This neural network is trained on 10^6 independent instances of the problem with a stochastic gradient descent using the Adam optimizer of [KB14], with standard hyperparameters for 100 epochs, with batches of size 1024. The linear predictor is trained in a batch mode on 10^4 independent examples with a ridge regularization parameter of value 10^{-5}.

The forecasting error is computed with a relative Root Mean-Square Error (RMSE) in order to provide dimensionless results:

$$\text{Relative RMSE} = \frac{\sqrt{\mathbb{E}\left|x(t+\Delta) - \tilde{x}(t+\Delta)\right|^2}}{\sqrt{\operatorname{Var}[x(t+\Delta)]}}, \tag{3.68}$$

where all expectations should be understood in an empirical sense.

3.4.1.2 *Results*

The forecasting results are displayed in Figure 3.8. The worst results are obtained by the linear estimator. For a fixed maximum polynomial order m, results are improved by increasing the oversampling factor P and hence the dictionary size. Increasing the polynomial order m also improves results provided P is large enough. For a sufficiently large dictionary, the sparse windowed forecasting outperforms both the linear and neural network estimators in this case.

3.4.2 *Real time-series*

We now test these forecasting algorithms over speech signals which are known to be sparse in a time-frequency dictionary. We show that the sparse windowed forecasting (3.67) yields an error which is smaller than with linear estimators and which is of the same order as the error of a one hidden layer neural network, although it uses much fewer parameters.

3.4.2.1 *Experimental setup*

The CSTR voice cloning toolkit (VCTK) [Yam12] is a corpus of speech data, with standardized sentences uttered by 109 native English speakers with various accents. Biophysical mechanisms producing speech generate signals with a sparse time-frequency structure [GME11]. Figure 1.2 illustrates this sparsity on an example of the VCTK dataset.

In these experiments, we only used recordings of the "p260" speaker in the data set. Among them, we chose 25 files randomly, split as 20/5 in train and test sets. For each recording, we normalized the time series to have values in $[-1, 1]$, and cropped the beginning and the ending of the signal with a threshold on the absolute value of 0.1. For a past of 512 samples and a future lag $\Delta = 200$, this left us with approximately 3×10^5 examples in the train set, and approximately 6×10^4 examples in the test set. We study forecasting results over this subset of the VCTK dataset, at multiple future lags Δ. The multiscale sparse windowed forecasting is computed for scale parameters $1 \leq j \leq 8$ using past samples over 51.2 ms. The Gabor dictionary is computed with oversampling factor $P = 4$ and a maximum polynomial degree $m = 1$. The sparse regularization parameter λ is set to 1 at all scales. Results are compared with a one hidden layer network with 4096 hidden units with ReLU activations. Training uses batches of size 128, and 20 epochs over the training dataset, to achieve convergence.

Results are measured with the relative RMSE. The future lags are sampled logarithmically from $\Delta = 10^{-1}$ ms (smallest possible lag) to 20 ms, which is chosen as the smallest value for which all methods returned an RMSE larger than the standard deviation of the target.

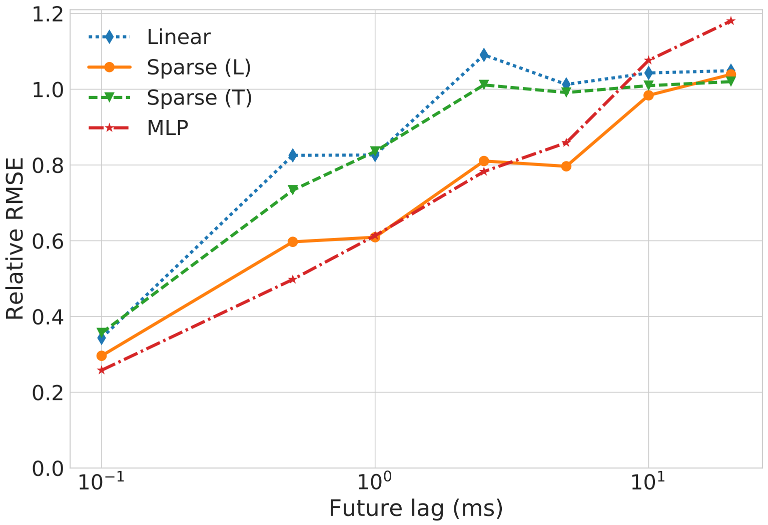

Figure 3.9: Relative RMSE on VCTK as a function of the future lags Δ. Sparse (T) stands for weakly parametrized multiscale forecasting (3.65), and sparse (L) for strongly parametrized multiscale forecasting (3.67). Sparse methods outperform the linear predictor, and have an error which is nearly the same as the neural network.

3.4.2.2 *Results*

The results of all methods computed on the test set for the forecasting task at multiple future lags Δ are displayed in Figure 3.9. In more details, the forecasting RMSE of a linear estimator, a neural network and multiscale sparse forecastings computed with a small number of parameters in (3.65) or a large number of parameters in (3.67) are given. In all cases, the non-linear estimators achieve a lower error than the optimized linear estimator. Multiscale forecasting with large number of parameters yield an error of the same order as the neural network.

3.5 CONCLUSION

In this chapter, we have investigated what priors can be used to improve upon linear predictors. Numerical experiments show that sparse time-frequency priors are good candidates for this task, allowing a 1-MLP to beat linear predictors. Building on a simple cosine model, a sparse time-frequency forecasting estimator has been proposed. It takes advantage of this sparsity by solving an inverse problem calculated with multiscale windows. Numerical experiments on synthetic and real signals show that numerical errors are close for the 1-MLP and the sparse multiscale windowed predictions.

4 TIME-SERIES GENERATION WITH SCATTERING INVERSE NETWORKS

In this dissertation, the second modeling task we consider is univariate time-series generation. This consists in estimating the unknown probability density function p_X of the stochastic process X from a finite number of samples $\{x_i\}_{1\leq i\leq N}$. We focus on audio time-series, such as speech or music recordings. Such time-series exhibit sparse linear decompositions in time-frequency dictionaries, such as wavelets. This sparsity leads to fat tails for the decomposition coefficients, which is evidence of a non-Gaussian distribution of the underlying process. Due to the presence of a hierarchical structure, these time-series also have long-range dependencies. Therefore, as explained in Section 1.1.1, without any additional assumption, the probability density estimation problem suffers from the curse of dimensionality.

Thus, in this context, we investigate the following questions: which priors can be used on the time-series to estimate the probability density in a tractable fashion? What algorithms can be used to perform this estimation?

Deep neural networks have recently achieved outstanding generation results for such processes. For audio generation, state-of-the-art results are obtained with probabilistic autoregressive networks, such as WaveNet [Oor+16] or SampleRNN [Meh+17]. These networks assume that the time-series has Markov dependencies of size τ, such that the modeled probability density reads

$$\widetilde{p_X}(\{x(t)\}_t) = \prod_{t=1}^{T} \widetilde{p_X}(x(t)|x(t-\tau),\dots,x(t-1)), \qquad (4.1)$$

where T is the total length of the time-series. Each conditional density $\widetilde{p}_X(x(t)|x(t-\tau),\dots,x(t-1))$ is then approximated by a neural network $f_\theta(x(t-\tau),\dots,x(t))$ under the assumption of stationarity. The past size τ is typically large, $\tau \geq 10^3$, so that this approach should suffer from the curse of dimensionality. The outstanding generation results lead to speculate the existence of implicit priors in the architecture of the network. However, due to the complexity of these networks, it is difficult to relate these priors to the actual processes.

In the case of image generation, remarkable results have been obtained in modeling complex, non-Gaussian distributions with implicit models [RMC16; Tol+18]. These models rely on a non-linear generator G and a latent variable Z having a fixed, white Gaussian distribution p_Z. New signals $\tilde{X}$ are sampled as $G(Z)$, so that the probability density p_X is modeled as $\widetilde{p}_X = G_* p_Z$. The generator G is typically a deep convolutional neural network, as is the case in Generative Adversarial Networks (GANs) [Goo+14] and Variational Auto-Encoders (VAEs) [KW14]. The white noise structure of Z shows that G disentangles the variabilities of the input signal along its different dimensions [Ben09]. This simplification of the variabilities of the signals also appears with signal transformations: a simple linear interpolation in the latent space $G(\theta z_1 + (1-\theta) z_2)$ corresponds to a meaningful warping in the signal space from $G(z_1)$ to $G(z_2)$ [RMC16].

The mathematical objective optimized by GANs and VAEs with a deterministic encoder requires to measure the equality of probability distributions, in the signal space for GANs or in the latent space for VAEs. However, the curse of dimensionality makes these measurements intractable in high dimension [Aro+17]. In order to circumvent this limitation, [AM18b] recently introduced an approach where the encoder is not learned, but replaced by a fixed scattering transform chosen using priors on the input signal. Learning the generator G is framed as an inverse problem, which is tractable under sparsity assumptions [Mal08].

In this chapter, we bridge the gap between latent generative networks and sparse time-frequency time-series, with the following contributions. Building on the framework proposed by [AM18b], we introduce an unsupervised architecture to generate time-series of arbitrary length from a white noise input, as $\tilde{X} = G(Z)$. The architecture of this Scattering Inverse Network G is especially crafted to take into account causality constraints, both during training and generation. Further, we introduce an adapted loss to train this model: it combines two terms, one stemming from an inverse problem in an adapted metric, the other a moment-matching term to control the generation from white noise. Part of the material contained in this chapter has been published in [AM18a].

This chapter is organized as follows. In Section 4.1, we cover the background material related to this chapter, notably linear models, GANs, VAEs, and audio probabilistic approaches such as WaveNet. In Section 4.2, a general overview of our non-linear filtering technique, relying on a scattering autoencoder model, is provided. In Section 4.3, we define the fixed encoder with a scattering transform using our priors on the signals. In Section 4.4, we describe the generator G used to produce signals, focusing on its architecture and training loss. In Section 4.5, numerical experiments performed to validate our approach are reported.

4.1 BACKGROUND

In this section, we provide a detailed review of the different methods and priors which enable to generate univariate stationary time-series. Under a Gaussian distribution assumption and a Markov dependence hypothesis, the statistics of the time-series can be reproduced by linear models, which lead to simple generative models. Latent generative networks such as GANs and VAEs generate non-Gaussian signals from noise using a convolutional neural network, but the relationship between the noise vector and the corresponding signal is unclear and their training presents theoretical and numerical issues. Autoregressive probabilistic networks provide state-of-the-art generation results for audio time series, but are much more difficult to interpret mathematically than GANs and VAEs and also suffer from the curse of dimensionality. We finally review sampling strategies constrained by statistics and discuss their relevance to our problem.

4.1.1 *Linear models*

Linear models yield a minimal mean-square error under a Gaussian prior for the process X. The process X is Gaussian if and only if for any number of coordinates k, for any temporal indices $t_1, \dots, t_k$ and for any weights $w_1, \dots, w_k \in \mathbb{R}$, the random variable $\sum_{i=1}^{k} w_i X(t_i)$ has a Gaussian distribution. Under this assumption, the probability density p_X is characterized by its first and second-order moments: the mean $\mu = \mathbb{E}[X(t)]$ and the autocovariance $\gamma_X(t) = \mathrm{Cov}(X(0), X(t))$.

A standard simplifying prior for p_X is the assumption of short-term dependence: there exists a time $\tau > 0$ such that $\gamma_X(t) = O(e^{-|t|/\tau})$. Under this hypothesis, Herglotz' theorem [BD91] states that the autocovariance γ_X is characterized by its Fourier transform $\widehat{\gamma}_X$, also called power spectrum:

$$\widehat{\gamma_X}(\omega) = \sum_t \gamma_X(t) e^{-i\omega t} \ .$$

ARMA models are linear parametric model which approach the power spectrum with a rational function $\widetilde{\gamma}_X$ defined in Fourier as:

$$\widehat{\widetilde{\gamma}_X}(\omega) = \frac{\sum_{m=0}^{M} b_m e^{i\omega m}}{\sum_{l=0}^{L} a_l e^{i\omega l}} \ . \tag{4.2}$$

This is equivalent to assuming that there exists a Gaussian white process Z, of zero mean and with uncorrelated temporal samples, such that the approximated process $\widetilde{X}$ satisfies

$$\sum_{l=0}^{L} a_l \widetilde{X}(t-l) = \sum_{m=0}^{M} b_m Z(t-m) \ . \tag{4.3}$$

Provided that the polynomial $\sum_{l=0}^{L} a_l u^l$ has no zero on the complex unit circle $|u| = 1$ [BD91, Chapter 3], one can invert this relationship to view $\widetilde{X}$ as the white noise Z convolved by a filter h:

$$\widetilde{X}(t) = \sum_{m=-\infty}^{+\infty} h_m Z(t-m) = h \star Z(t) \ , \tag{4.4}$$

where the coefficients h_m are defined on an open set containing the complex unit disk by

$$\sum_{m=-\infty}^{+\infty} h_m u^m = \frac{\sum_{m=0}^{M} b_m u^m}{\sum_{l=0}^{L} a_l u^l}. \tag{4.5}$$

In the end, linear models yield a generative model defined as the convolution of a Gaussian white noise Z by a filter h (4.4). The resulting signal is therefore Gaussian, and is poorly suited to generate natural signals such as speech or music, whose statistics are non-Gaussian.

4.1.2 *Latent generative models*

We review in detail latent generative models relying on deep neural networks, which are depicted schematically in Figure 1.3. We cover in particular Auto-Encoders and Generative Adversarial Networks (GANs). These models map a Gaussian white noise variable Z defined on a latent space $\mathcal{Z}$ to the signal space $\mathcal{X}$ with a generator G, which consists in a neural network. Therefore, the model $\widetilde{p}_X$ for the probability distribution of X is implicitly defined as $\widetilde{p}_X = G_* p_Z$. In both cases, these networks have shown a great success at modeling complex image distributions. However, the reasons supporting these successes remain unclear, since both approaches are subject to the curse of dimensionality.

4.1.2.1 *Auto-Encoders*

In what follows, we review the approach of [Tol+18], which uses deterministic auto-encoders instead of probabilistic ones. This approach is conceptually simpler than the original Variational Auto-Encoders [KW14], while still leading to remarkable modeling results.

Auto-encoder models rely on two networks: an encoder $E : \mathcal{X} \to \mathcal{Z}$ and a decoder $G : \mathcal{Z} \to \mathcal{X}$. These networks are trained with respect to a loss based on two terms, balancing reconstruction and latent space modeling.

The first loss term is a reconstruction loss measuring how close $G(E(x))$ and x are, for all training inputs x, with respect to a criterion $c(x, G(E(x)))$. Such a criterion can simply be chosen as the mean-square error $\|x - G(E(x))\|_2^2$.

The second loss term controls the distance between the distribution of $\widetilde{Z} = E(X)$ and a prior density distribution p_Z, such as a white centered Gaussian distribution, with an adequate divergence d_Z. Such a divergence can be implemented *e.g.* with a Maximum Mean Discrepancy Distance [Gre+07].

Maximum Mean Discrepancy measures a distance between two probability measures by computing the maximal difference of expectations under the two measures, among a given set of functions.

Provided this function set can be spanned by a Reproducible Kernel Hilbert Space, this distance can be computed empirically thanks to the "kernel trick" [Gre+07].

The resulting loss reads:

$$\min_{G,E} \mathbb{E}[c(X, G(E(X))] + \eta d_Z(p_{\widetilde{Z}}, p_Z) , \tag{4.6}$$

where $\eta > 0$ is a trade-off parameter, and the expectations and distances are computed from samples.

Auto-Encoders have lead to outstanding generation results [Tol+18]. Combined with an additional supervision in the latent space, they allow to transform signals with a simple modification of coordinates in the latent space [Lam+17; Eng+17]. For time-series, multiple works have focused on modeling of spectrograms with 2D auto-encoders to generate and transform audio [BB16; HZG17]. However, the resulting spectrograms need to be inverted with the Griffin-Lim algorithm [GL84], which leads to undesired artifacts.

Auto-Encoders face the curse of dimensionality with the second term in loss (4.6), which is impossible to estimate accurately without additional assumptions. Therefore, hidden priors must lie in the architecture used by these networks.

4.1.2.2 *Generative Adversarial Networks*

DEFINITION Generative Adversarial Networks (GANs) [Goo+14] rely on two networks, a generator $G : \mathcal{Z} \to \mathcal{X}$ and a discriminator $D : \mathcal{X} \to (0, 1)$. These networks are trained jointly from a corpus of samples $\{x_i\}_{1 \le i \le N}$.

These networks are both deep neural networks, but have opposite, or adversarial, objectives. The goal of the generator is to map a Gaussian white noise variable $Z \in \mathcal{Z}$ to new realistic samples $\widetilde{X} = G(Z)$. The goal of the discriminator is to classify actual images belonging to the training set from newly generated images. These competing objectives result in the following loss:

$$\min_G \max_D \mathbb{E}\left[\log D(X)\right] + \mathbb{E}\left[\log\left(1 - D(G(Z))\right)\right] . \tag{4.7}$$

MATHEMATICAL ANALYSIS For a fixed generator G, the maximum over all possible functions $D : \mathcal{X} \to (0, 1)$ is attained for the Bayes classifier

$$D^*(x) = \frac{p_X(x)}{p_X(x) + p_{\widetilde{X}}(x)} . \tag{4.8}$$

For this choice of optimal discriminator D^*, it can be shown [Goo+14] that the generator G minimizes the Jensen-Shannon divergence [Lin91] between the distribution p_X and the distribution $p_{\widetilde{X}}$, which motivates this training formulation.

As shown by [Aro+17], this appealing property is in fact fooled for two reasons. First, because of the finite size of the discriminative network, it is impossible to approach the optimal discriminator (4.8) in a large dimensional setting, unless additional assumptions are made. Second, the estimation of the Jensen-Shannon divergence is subject to the curse of dimensionality, unless dimensionality reduction assumptions are made.

EMPIRICAL SUCCESS Despite these theoretical issues, GANs relying on deep convolutional networks [RMC16] have led to outstanding generation results for images. In particular, these networks appear to factorize the variabilities of images: linear interpolations in the latent space lead to meaningful interpolations in the signal space. In the case of audio, a recent work [DMP18] shows that GANs are also able to generate raw audio with an impressive quality.

Because of the theoretical limitations reviewed above, there must be hidden priors made by GANs, notably the deep convolutional architecture, which allow to obtain such results.

4.1.3 *Autoregressive probabilistic networks*

4.1.3.1 *General approach*

Autoregressive probabilistic networks aim at modeling the probability density p_X of the process X in an explicit fashion, with a model $\widetilde{p}_X$ fitted on data.

For any probability density function p_X, the following conditional decomposition holds:

$$p_X(x) = \prod_{t=1}^{T} p_X(x(t)|x(< t)) \ . \tag{4.9}$$

Autoregressive probabilistic networks simplify this decomposition with a Markov dependency assumption, which allows the replacement of $x(< t)$ by $x(t-\tau), \dots, x(t-1)$, and a stationarity assumption, which makes it possible to model $p_X(x(t)|x(t-k), 1 \leq k \leq \tau)$ with the same function f_θ for all t:

$$\widetilde{p_X}(x) = \prod_{t=1}^{T} \underbrace{\widetilde{p_X}\,(x(t)|x(t-k), 1 \leq k \leq \tau)}_{f_\theta(x(t-\tau),\dots,x(t))} \ . \tag{4.10}$$

The parametrized f_θ is a neural network with parameters θ.

The autoregressive formulation (4.10) allows for a tractable evaluation of $\widetilde{p}_X$ on samples x. This is a noticeable difference from energy-based unsupervised models, which require to compute a partition function containing an exponential number of terms [DBC11]. Further, such a model permits to sample new signals exactly and easily, by sampling each conditional $\widetilde{p_X}(x(t)|x(t-k), 1 \leq k \leq \tau)$ sequentially for $t = 1, 2, \dots$

The model $\widetilde{p_X}$ is fitted by maximizing its log-likelihood by a variant of stochastic gradient descent, Adam [KW14], or, equivalently, by minimizing its negative log-likelihood, on training data $\{x_i\}_{1 \leq i \leq N}$

$$\min_{\theta} \; -\frac{1}{N}\sum_{i=1}^{N} \log \widetilde{p_X}(x_i). \tag{4.11}$$

Plugging (4.10) into (4.11) yields the minimization problem

$$\min_{\theta} \; -\frac{1}{N}\sum_{i=1}^{N}\sum_{t} \log f_\theta(x_i(t-\tau),\ldots,x_i(t)) \, . \tag{4.12}$$

4.1.3.2 *WaveNet*

WaveNet [Oor+16] is the prominent model for time-series following this autoregressive approach. It uses a deep causal convolutional neural network for f_θ.

The convolutional structure of f_θ allows a parallel evaluation of (4.12) both across time t and samples i. This parallelization can be used to speed-up training on GPUs, thereby allowing to process large datasets.

In order to attain a large receptive field τ without having too many layers, it uses so-called dilated convolutions [YK16]. Dilated convolutions insert zeros between the coefficients of the kernel of the convolution, so that the receptive field is broader than the size of the kernel. By doubling the size of the inserted zeros at each layer, one obtains a receptive field of τ with $\log_2(\tau)$ layers.

This network has achieved outstanding modeling results on audio time-series, such as speech and piano music, by directly working on the raw waveforms. When conditioned on higher-level features, such as words or phonemes, it reaches state-of-the-art text-to-speech quality [Ar1+17; She+17].

4.1.3.3 *ParallelWavenet*

WaveNet [Oor+16] allows for a fast training thanks to its convolutional structure. However, generation is sequential along time, which may be too slow for real-time audio synthesis.

The Parallel WaveNet [Oor+17] architecture was proposed to make both training and generation parallel across time. Its generator architecture maps white noise variables Z to probability densities $G(Z)$, so that new samples are obtained as $X(t) \sim G(Z)(t)$. The generator G is a deep convolutional neural network close to the original WaveNet. Thanks to its convolutional structure, synthesis can be performed in parallel across time, reaching a thousand-fold speed-up with respect to sequential generation.

The Parallel WaveNet is trained with a pre-trained WaveNet. This pre-trained WaveNet evaluates the log-likelihood probability densities produced by the Parallel WaveNet after sampling, and sends a feedback to the Parallel WaveNet so as to maximize this log-likelihood. Such an approach is very close to the adversarial training of GANs.

4.1.3.4 *Mathematical Analysis*

Both WaveNet and Parallel WaveNet rely on the evaluation of the log-likelihood of the model on samples. This is equivalent to measuring the Kullback-Leibler divergence between the model and the empirical measure of the process. However, without prior assumptions on the underlying process, this empirical estimate is a very poor measurement of the divergence between the model and the actual probability density p_X [Aro+17]. Therefore, the network must use hidden assumptions made by its architecture.

The architecture of the WaveNet is very difficult to understand. It relies on many modern deep learning tricks, such as residual connections [He+16] and multiplicative gates [HS97], whose importance is difficult to assess. Interpreting this network in light of signal processing knowledge is a very complex task.

4.1.4 *Sampling constrained by statistics*

We now review a sampling method standard for textures which can be used to perform signal transformations, and discuss the differences with our approach.

Given an original signal x_0 sampled from the process X, let $\Phi(x_0)$ be a vector representation of x_0. This technique generates new signals close to x by solving a minimization problem in x

$$\min_x \|\Phi(x) - \Phi(x_0)\| . \tag{4.13}$$

By using different random seeds for x and solving the minimization problem up to a threshold ϵ, one gets new different samples, which might differ from the original x_0. Notice the algorithmic difference with deep implicit generative models, where no minimization problem is solved at generation time.

This approach is especially useful in modeling audio or visual textures [Jul62]. Textures consist in a repetition of patterns, for instance bricks on a wall or cicadas singing. These signals are modelled as ergodic processes [BD91]. Provided the representation $\Phi(x)$ involves a spatial or temporal mean, by ergodicity it will converge to the mean of the process up to a renormalization constant, *i.e.* $\Phi(x) \approx_c \mathbb{E}\Phi(X)$. As a consequence, assuming a maximum entropy distribution on the space of admissible solutions to (4.13), the newly generated signals are endowed with a Gibbs probability distribution constrained by these statistics [ZWM98].

In neuroscience, these methods are used to probe neural perceptual systems [CRS05; MS11]. The goal is to find neurally plausible statistics

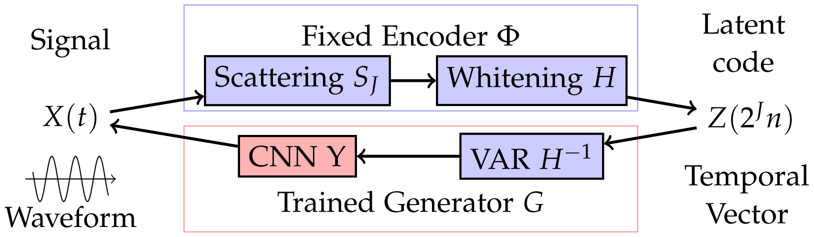

Figure 4.1: General approach to generate new signals from a white noise input. The encoder Φ is chosen using priors on the signals, while the generator G is trained to invert the encoder.

which lead to perceptually indistinguishible textures [DF81; PS00]. More generally, this approach can be used to probe a given representation Φ which will be later used *e.g.* for classification purposes [BM13a; GEB15; Los17].

While this approach builds a model of the probability density of the whole process in the case of textures, this is not true for more complex, non-ergodic signals. As a consequence, we stress that such a sampling is conceptually different from generative modeling, whose goal is to model the whole probability distribution of the signals.

Statistics matching can be used to transform signals, with a method which is more complicated than mere linear interpolations. Assuming that the representation Φ is splitted between a content representation and a style representation, it is possible to transfer the style of a signal to another signal while keeping the content. For instance, this allows the generation of contemporary building pictures as if painted by impressionnists [GEB16]. For speech, this approach enables changing the voice of a speaker without altering its content [Cho+18]. In the case of music signals, early results [Gri+17] stress the difficulty of defining and separating the style from content of complex musical pieces; the more convincing results are obtained when one of the signals is a simple texture.

4.2 GENERAL APPROACH

In this section, we give a general overview of our approach to solving the unsupervised generation problem, sketched in Figure 4.1. It consists in an encoder $\Phi : \mathcal{X} \to \mathcal{Z}$ and a decoder $G : \mathcal{Z} \to \mathcal{X}$, which are mappings between the signal space $\mathcal{X}$ and the latent space $\mathcal{Z}$. The encoder Φ is fixed, and defined so as to map the input signals X to an approximately Gaussian white noise $\widetilde{Z}$. The decoder or generator G is learnt to invert Φ on training examples. The generator G should then be able to map new noise vectors Z to realistic signals $\widetilde{X} = G(Z)$.

4.2.1 *Encoder predefined with priors: whitened scattering transform*

Following [AM18b], Φ is designed by hand. Indeed, optimizing Φ to ensure that the distribution of $\tilde{Z} = \Phi(X)$ is close to the distribution of Z is intractable, since the very objective of this optimization suffers from the curse of dimensionality. Φ should ideally map X to a Gaussian white noise variable, while preserving invertibility in a stable fashion. Using prior information on the input signals, notably the existence of a sparse time-frequency decomposition and the decorrelation of the coefficients of such a decomposition at long time intervals, it is possible to hand-craft this encoder.

The main tool we employ to produce Gaussian variable is a local averaging, computed with a low-pass filter along time. Under assumptions of decorrelations of first- and second-order moments, and provided the averaging window is large enough, the resulting variable will asymptotically converge to a Gaussian one [Das08]. However, averaging can also lose a lot of information on the original signal. To alleviate this effect, we follow the scattering approach, which consists in employing multiple channels recursively splitting the information contained in different frequency bands [Mal12; BM13b; AM14; ALM15].

Provided the aforementioned assumptions on the signals hold, the resulting representation, known as the scattering transform $S_J(X)$, will be invertible in a tractable fashion. Indeed, it exploits an underlying sparse representation, known to stabilize deconvolution problems [Mal08].

$S_J(X)$ may be close to a Gaussian variable, but it does not have a white covariance matrix: it has non-negligible channel-wise and temporal correlations. In order to map it to a Gaussian white noise, we use an autoregressive filter H trained on the dataset, and output the corresponding innovations $H(S_J(X))$. The latter defines the encoder $\Phi(X) = H(S_J(X))$.

4.2.2 *Generator: Scattering Inverse Network*

The operator G consists in a de-whitening operator H^{-1}, which retrieves approximate scattering vectors, followed by a causal convolutional neural network Y which maps scattering vectors to signals X. The operator G is learned so as to invert the scattering transform, with an adequate metric.

The inverse problem that G is trained to solve is tractable. Indeed, the scattering transform performs an averaging with respect to the temporal axis. Insofar as the underlying representation of the scattering is sparse, this averaging can be inverted. Further, the scalogram representation without averaging can also be inverted [Wal15]. Note that the training of G does not rely on probability distribution estimates, so that the training avoids the curse of dimensionality.

G is trained to invert the mapping Φ on training examples. However, it does not hold that G is the inverse of Φ. Indeed, G has a finite expressivity because of the finite number of neurons in each layer. As a consequence, G is a regularized inverse of Φ. The exact nature of this regularization remains an open problem, which is linked to the general inductive bias of deep neural networks [Zha+17; AM18b].

4.3 WHITENED SCATTERING Φ: INFORMATIVE GAUSSIAN ENCODER

In the previous section, we have introduced the general approach to generate time-series. In this section, following [AM18b], we show how priors on the signals X allow to specify an embedding $\widetilde{Z} = \Phi(X)$ which is an approximately Gaussian white noise, yet contains sufficient information on X to reconstruct an approximation of X from $\Phi(X)$. Among such priors are a sparse decomposition of the signals in time-frequency dictionaries, the perceptual stability to small deformations, and the decorrelation of signals at large scales.

These assumptions lead us to use a whitened scattering transform $\Phi = H \circ S_J(X)$, where H is a whitening operator and S_J the scattering transform [ALM15]. Indeed, by construction, the scattering transform is stable to small deformations of the input signal. The sparsity of the input signals stabilize the inversion of $S_J(X)$. Its low-pass averaging yields approximately Gaussian variables under decorrelation assumptions thanks to the central limit theorem. H whitens the Gaussian variable $S_J(X)$ thanks to a vector autoregressive (VAR) filter, outputting the innovations of the process. H can be inverted analytically.

4.3.1 *Low-pass averaging yields Gaussianization*

Gaussian variables naturally arise asymptotically as a result of the central limit theorem. In its original statement, the central limit theorem requires independent random variables. However, under decorrelation assumptions of first- and second-order moments on the stationary sequence $X(t)$, $\sum_{t=-2^J}^{2^J} X(t)$ converges asymptotically to a Gaussian variable, up to a renormalization term [Das08]. As a consequence, we consider that for J "large enough", this variable is "approximately" Gaussian. In order to reach a faster decay in the Fourier domain, local sum $\sum_{t=-2^J}^{2^J} X(t)$ is approached as a convolution $X \star \phi_J(0)$ with a smooth averaging filter ϕ_J whose temporal support scales as 2^J

$$\phi_J(u) = 2^{-J}\phi(2^{-J}u). \tag{4.14}$$

We will therefore consider the whole sequence $(X \star \phi_J(t))_t$, which then becomes approximately Gaussian when J grows large. Since ϕ_J is a smooth low-pass filter, it is possible to subsample $X \star \phi_J$ without loss of information, leading to the quantity

$$(X \star \phi_J(2^J n))_{n \in \mathbb{Z}}. \tag{4.15}$$

However, for the signals in which we are interested, the low-pass filtering removes all information from X: most of the energy of the signals lies in a frequency band higher than the cut-off frequency of the filter ϕ_J.

As a consequence, one should not simply average X, but a whole representation $U(X)$ containing more information on the original X, where $U(X)(t)$ is a vector with K_J coordinates. Provided this representation $U(X)$ also decorrelates for large temporal intervals, the representation

$$S_J(X)(2^J n) = U(X) \star \phi_J(2^J n) \tag{4.16}$$

will approximately converge to a Gaussian variable. In the next sections, we explain how to build $U(X)$ so that $S_J(X)$ is sufficiently informative about X and is stable to time-frequency deformations.

4.3.2 *Scalogram: Wavelet Modulus Transform*

4.3.2.1 *Wavelet Transform*

The low-pass averaging $X \star \phi_J$ loses too much information on the original signal X. The missing information is located in high frequencies, which can be accessed with band-pass filters. Such filters could stem from a windowed Fourier transform. However, the high frequencies of a windowed Fourier transform are instable with respect to small temporal warpings [AM14]. In order to achieve stability with respect to temporal deformations, it is necessary to group frequencies among dyadic packs, leading to wavelets [Mal12; AM14].

Wavelets $\{\psi_\lambda\}_{\lambda \in \Lambda}$ are obtained from the dilatations of a single mother wavelet ψ, possibly complex-valued:

$$\psi_\lambda(t) = 2^{-\lambda} \psi(2^{-\lambda} t). \tag{4.17}$$

Contrary to Chapter 2, λ is not limited to integer values: typically, one uses $\lambda = q/Q$ where q is an integer, and $Q \geq 1$ is the number of intermediate scales per octave. This allows us to finely segregate partials at high frequencies. We detail later in this subsection adequate choices for ψ used in this dissertation, namely Morlet and Gammatone wavelets.

The set Λ is chosen so that the wavelet transform $\mathbf{W}$, defined as

$$\mathbf{W} : x \mapsto (x \star \phi_J, \{x \star \psi_\lambda\}_{\lambda \in \Lambda}), \tag{4.18}$$

is approximately isometric: in other words, there exists $\epsilon > 0$ such that for all signals x with limited bandwidth,

$$(1-\epsilon)\|x\|^2 \leq \|x \star \phi_J\|^2 + \sum_{\lambda \in \Lambda} \|x \star \psi_\lambda\|^2 \leq \|x\|^2. \tag{4.19}$$

4.3.2.2 *Wavelet Modulus*

In order to obtain approximately Gaussian features, we would like to average the wavelet coefficients $X \star \psi_\lambda$ with the low-pass filter ϕ_J. Unfortunately, since ψ_λ is well located in high frequencies, the resulting signal $X \star \psi_\lambda \star \phi_J$ contains almost zero energy. It is necessary to consider a non-linear transform of $X \star \psi_\lambda$ before performing the averaging. In [Bru13], a pointwise complex modulus is proved to be an adequate non-linearity to achieve stability with respect to small time warpings. As a consequence, we use the modulus wavelet features $|X \star \psi_\lambda|$ for all λ, which are then averaged. We thus denote the scalogram or log-spectrogram $U_1(X)$, defined as

$$U_1(X)(t, \lambda) = |X \star \psi_\lambda|(t). \tag{4.20}$$

In order for the averages $|X \star \psi_\lambda| \star \phi_J$ to be informative, $|X \star \psi_\lambda|$ should be a low-pass signal. A sufficient condition for $|X \star \psi_\lambda|$ to be a low-pass signal is that in the Fourier domain, the wavelets ψ_λ are concentrated around a single frequency [Oya17]. In particular, they need to be approximately analytic, that is

$$\forall \omega < 0, \hat{\psi}_\lambda(\omega) \approx 0. \tag{4.21}$$

Even though $|X \star \psi_\lambda|$ is designed to have significant low-frequency energy, the low-pass averaging with ϕ_J might neglect significant portions of its spectrum. It is therefore tempting to apply recursively another wavelet transform on top of each $|X \star \psi_\lambda|$, followed by a modulus. This would give rise to the temporal wavelet transform, which was introduced by [BM13b; AM14]. In this dissertation, we use a stronger version of scattering, the time-frequency scattering proposed by [ALM15], where the second wavelet transform is applied both across time and scale index λ. Section 4.3.3 is devoted to its exposition.

We conclude this section with an overview of the wavelets used in this chapter.

4.3.2.3 *Symmetric Morlet Wavelets*

Morlet wavelets ψ are very close to Gabor filters [Gab46], up to a low-pass term which ensures an exact zero-mean, making them proper wavelets. Figure 4.2 (left) shows a Morlet wavelet. The Morlet mother wavelet ψ, centered at frequency ξ and with bandwidth σ, is defined as:

$$\psi(t) = \alpha g_\sigma(t)(e^{i\xi t} - \beta)\ , \tag{4.22}$$

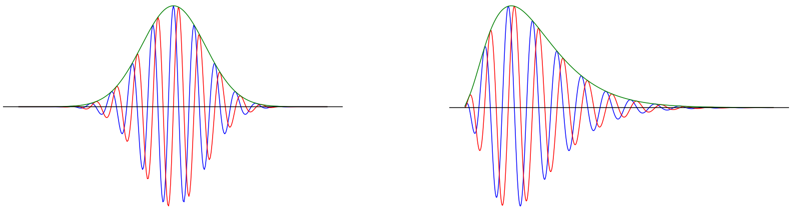

Figure 4.2: Morlet wavelets (left) and Gammatone wavelets (right). The blue and red curves respectively correspond to the real and imaginary parts of the wavelet, while the green curve denotes the envelope.

where α is a normalization parameter ensuring $\|\psi\|_1 = 1$, β is a parameter adapted so that ψ has zero mean, and g_σ is a Gaussian envelope:

$$g_\sigma(t) = e^{-\frac{t}{2\sigma^2}} \ , \tag{4.23}$$

The corresponding low-pass filter ϕ is defined as

$$\phi(t) = \frac{g_{\sigma_\phi}(t)}{\|g_{\sigma_\phi}\|_1}. \tag{4.24}$$

where σ_ϕ is a bandwidth parameter. This parameter is tuned so as to ensure the energy stability of the whole wavelet transform, by filling exactly the frequency gap left by the band-pass filters.

4.3.2.4 *Causal Gammatone Wavelets*

Pseudo-analytic gammatone wavelets were designed in [VAS14] in order to provide a mathematical frame to the gammatone filters which are common in the neurophysiology literature. These filters model the low-level filters used in the brain to process audio. In the continuous-time, the Gammatone mother wavelet is defined as

$$\psi(t) = \alpha \frac{d}{dt}\left(t^p e^{-t/\sigma} e^{i\xi t}\right) 1_{t \geq 0}, \tag{4.25}$$

where the integer $p \geq 1$ controls the smoothness of ψ, σ controls the bandwidth of ψ, ξ is its central frequency, and α is a normalizing parameter. Figure 4.2 (right) displays one Gammatone wavelet.

Thanks to the exponential damping, Gammatone wavelets can be implemented with recursive infinite impulse response (IIR) and finite impulse response (FIR) filters. After discretization, the Fourier transform of the filters reads:

$$\hat{\psi_\lambda}(\omega) = \frac{(1 - e^{-\sigma_\lambda})^N}{(1 - e^{-\sigma_\lambda} e^{-i(\omega - \xi_\lambda)})^N} (1 - e^{-i\omega}), \tag{4.26}$$

where $\sigma_\lambda = 2^{-\lambda}\sigma_\psi$ and $\xi_\lambda = 2^{-\lambda}\xi$.

The corresponding low-pass filter is defined as

$$\hat{\phi}_J(\omega) = \frac{(1-e^{-\sigma_J})^N}{(1-e^{-\sigma_J}e^{-i\omega})^N}, \tag{4.27}$$

with $\sigma_J = 2^{-J}\sigma_\phi$.

The parameters σ_ψ, ξ_ψ and σ_ϕ are chosen in order to preserve energy. One can follow calculations similar to those derived in [Los17], after an adaptation to the discrete setting.

4.3.3 *Time-Frequency Scattering*

We review the time-frequency scattering transform [ALM15]. It makes it possible to recover the information lost by the averaging of the scalogram $|X \star \psi^1_\lambda| \star \phi_J$, while ensuring the stability of the resulting representation to joint time-frequency warpings. As shown in Figure 4.3, it consists in a time-frequency filtering of the scalogram $|X \star \psi^1_\lambda|(t)$ in the variables (t, λ) with time-frequency wavelets.

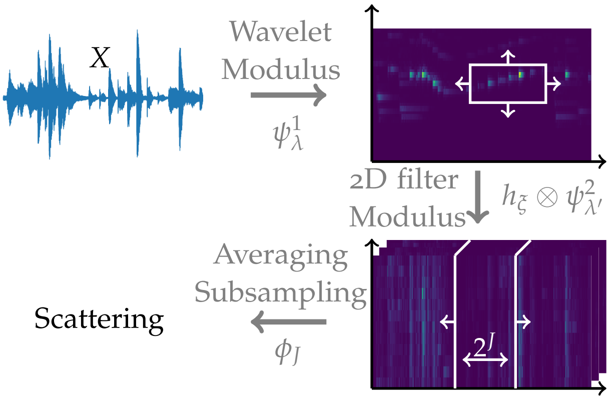

Figure 4.3: Time-frequency scattering transform $S_J(X)$. The scalogram is obtained with a first wavelet transform ψ^1_λ followed by a point-wise modulus. A joint time-frequency filtering of this log-spectrogram with the filters $h_\xi \otimes \psi^2_{\lambda'}$ regularizes the time-frequency deformations of the signal. The low-pass convolution with ϕ_J Gaussianizes the resulting tensor.

4.3.3.1 *Time-Frequency Deformations*

Time-frequency deformations correspond to local deformations of the scalogram $U_1(t, \lambda)$, seen as an image in the variables (t, λ). Among these deformations are frequential transpositions, which correspond to displacements along the axis λ:

$$\mathcal{T}_b U_1(t, \lambda) \approx U_1(t, \lambda - b). \tag{4.28}$$

Here, we used an approximate equality because in general, such an operation is mathematically ill-defined due to constraints on the wavelet transform [Mal08, Chapter 4]. We refer to [Los17] for a detailed treatment of transpositions.

Applying only temporal convolutions on top of the scalogram $U_1(t, \lambda)$ would result in a representation which is unstable to frequential transpositions, that is, a small displacement $|b| \ll 1$ would result in a large displacement in the representation [Los17]. In order to ensure stability along such frequential displacements, one could think of performing convolutions along the frequential axis only. Such an approach would lose too much information on the original signal, as it would be unable to differentiate separable deformations from joint ones.

Joint time-frequency deformations cannot be written as a plain temporal or frequential warping. A simple example consists in a frequency-dependent time-shift $b(\lambda)$:

$$\mathcal{T}_{b(\lambda)} U_1(t, \lambda) = U_1(t - b(\lambda), \lambda). \tag{4.29}$$

Performing separate convolutions along time or frequency on $\mathcal{T}_{b(\lambda)}$ would ignore the relative displacements of the different frequencies, thus losing crucial information. It is thus necessary to employ a joint time-frequency filtering of the scalogram in order to obtain an informative and stable encoder.

4.3.3.2 *Time-Frequency Scattering*

We present the time-frequency scattering transform [ALM15]. It is displayed in Figure 4.3. This transform applies joint time-frequency filters to the scalogram $U_1(t, \lambda) = |X \star \psi^1_\lambda(t)|$ in order to obtain an informative representation of the signal which is stable to time-frequency deformations.

The time-frequency filters are built as a separable product $h_\xi \otimes \psi^2_{\lambda'}$ of frequential filters h_ξ and temporal filters $\psi^2_{\lambda'}$. The frequential filters h_ξ are localized Fourier atoms of discrete frequency ξ with a Hann window whose size P matches one octave, $P = Q$. The convolution is computed in half-overlaps over the frequency axis. The temporal filters $\psi^2_{\lambda'}$ are similar to the wavelets used to compute U_1, but with only $Q_2 = 1$ wavelet per octave hence the use of a subscript 2 to distinguish them from the filters ψ^1_λ used in the first scalogram U_1. After performing the convolution, a modulus non-linearity is also applied in order to remove the local phase, thereby regularizing the representation. We thus define:

$$U_2^{\text{joint}}(t, \xi, \lambda') = \left| |X \star_t \psi_\lambda| \star_{t,\lambda} (h_\xi \odot \psi^2_{\lambda'}) \right|. \tag{4.30}$$

As shown empirically in [ALM15], only the paths with $\lambda' \geq \lambda$ need to be considered because the other ones contain negligible energy.

The scalogram $U_1(t, \lambda)$ is also filtered along frequency as

$$U_1^{\text{joint}}(t, \xi) = |U_1(t, \cdot) \star_\lambda h_\xi|. \tag{4.31}$$

in order for U_1 to also be stable to such time-frequency deformations. Note that a true analog to Equation (4.30) would have required to use the joint filter $h_\xi \odot \phi_J$, but as a low-pass filter will still be applied to U_1^{joint} afterwards, this intermediate low-pass filter is redundant.

The final representation $U(X)$ for the joint time-frequency scattering is thus:

$$U(X)(t) = \left[X(t), U_1^{\text{joint}}(t,\xi), U_2^{\text{joint}}(t,\xi,\lambda')\right]_{\xi,\lambda'}. \tag{4.32}$$

$U(X)$ is a vector evolving along time with K_J coordinates or channels indexed by (ξ, λ'). One can compute that

$$K_J = 1 + Q(2J+3) + Q(J-2)^2. \tag{4.33}$$

$U(X)$ is finally averaged along time with ϕ_J and downsampled by a factor 2^J as

$$\begin{aligned} S_J(X)(2^J n) &= U(X) \star_t \phi_J(2^J n), \\ &= \Big[X \star \phi_J(2^J n), ||X \star_t \psi_\lambda| \star_\lambda h_\xi| \star_t \phi_J(2^J n), \\ &\quad ||X \star_t \psi_\lambda| \star_{t,\lambda} (h_\xi \odot \psi^2_{\lambda'})| \star_t \phi_J(2^J n)\Big]_{\lambda,\xi}. \end{aligned} \tag{4.34}$$

As shown in [ALM15], there is no need to apply a third wavelet transform on top of the time-frequency scattering, insofar as it contains only negligible energy. Further, the sparse structure of the channels contains enough information so as to allow to reconstruct an approximate inverse of X from $S_J(X)$.

4.3.4 *Whitening operator H*

The scattering operator maps the time-series X to a random variable $S_J(X)$ which is approximately Gaussian. $S_J(X)$ has a non-white correlation structure, both along the temporal axis and the channel axis. The whitening operator H removes this covariance structure using an autoregressive filter trained for prediction. It outputs the innovations of the process, which follow an approximately white noise distribution.

The whitening operator H is defined by a vector autoregressive (VAR) model of $S_J(X)$ [BD91, Chapter 11]. It is calculated by regressing $S_J(X)(2^J(n+1))$ over M_J past values $S_J(X)(2^J(n-m))$ for lags $0 \leq m < M_J$ as

$$\widetilde{S}_J(X)(2^J(n+1)) = \sum_{m=0}^{M_J-1} W_{m,J} S_J(X)(2^J(n-m)) + b_J, \tag{4.35}$$

where $W_{m,J}$ are regression matrices of size $K_J \times K_J$ and b_J is a vector of dimension K_J. The resulting regression error is defined by a zero-mean white noise vector $\widetilde{Z} = H(S_J(X))$, which satisfies:

$$\begin{aligned}
\widetilde{Z}(2^J(n+1)) &= \Sigma^{-1/2}\Big[S_J(X)(2^J(n+1)) - \widetilde{S}_J(X)(2^J(n+1))\Big] \\
&= \Sigma^{-1/2}\Big[S_J(X)(2^J(n+1)) - b_J \\
&\qquad - \sum_{m=0}^{M_J-1} W_{m,J} S_J(X)(2^J(n-m))\Big], \qquad (4.36)
\end{aligned}$$

where Σ is the covariance matrix of the centered regression errors.

4.4 GENERATOR G

In the previous section, a causal embedding $\tilde{Z} = \Phi(X)$ has been introduced. In this section, we define a neural network G to invert this embedding. The architecture of G is crafted so as to take into account the causality structure of the embedding $\tilde{Z}$ with respect to the original time-series X, and long-range dependencies within X. The generator G then maps a Gaussian white noise Z to approximate signals $\tilde{X} = G(Z)$.

The network G is built with two components, Υ and H^{-1}, as $G = \Upsilon \circ H^{-1}$. On the one hand, H^{-1} is the inverse of the whitening operator H. H^{-1} maps a white noise Z to an approximation X_J of the scattering $S_J(X)$. The autoregressive nature of H^{-1} allows us to create long-range dependencies in X_J with few parameters. On the other hand, the convolutional network Υ contains J hidden layers computed with causal *à trous* convolution filters and pointwise non-linearities. Υ is trained to invert $S_J(X)$ and outputs $\tilde{X} = \Upsilon(X_J)$, an approximate reconstruction of the original signal X.

Υ and H^{-1} are trained separately. H is trained by minimizing a prediction error on $S_J(X)$. Υ is trained to invert the scattering transform $S_J(X)$ on training examples. We introduce a perceptual metric to carry out this inversion in order to reconstruct high frequencies. Further, in order to control the behavior of the network G inputted with white noise, we introduce a loss term controlling the scattering moments of the generation.

4.4.1 *Network definition*

Figure 4.4 illustrates the architecture. Linear predictions are computed at the largest scale 2^J by the autoregressive layer H^{-1}. Each prediction is propagated across scales by a network Υ which inverts the scattering transform. One prediction outputs a block of 2^J time samples.

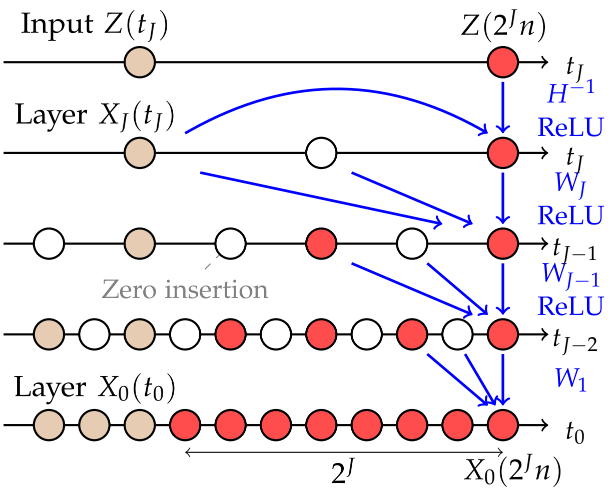

Figure 4.4: A Scattering Inverse Network is a linear recurrent network followed by a causal deep convolutional network with J layers. It takes as input a vector of Gaussian white noise $Z(2^J n)$ (top right, red), and computes the corresponding scattering vector $X_J(2^J n)$ by applying H^{-1}, and a ReLU to ensure non-negativity. Intermdiate layers $X_j(t_j)$ are then computed with causal convolutions denoted by blue arrows and zero insertions (white points). The single vector $Z(2^J n)$ outputs 2^J values for $X_0(t_0)$, marked with red points.

4.4.1.1 *Top autoregressive layer*

The first layer consists in the linear autoregressive filter H^{-1} applied on the noise Z. The resulting vector sequence $H^{-1}Z$ should represent scattering vectors, which are non-negative by construction. In order to enforce this non-negativity, the linear operator H^{-1} is followed by the pointwise ReLU non-linearity $\rho(u) = \max(u, 0)$ to compute X_J as

$$X_J = \rho\left(H^{-1} Z\right) . \tag{4.37}$$

The operator H^{-1} is readily defined from Equation (4.36) as

$$\begin{aligned}\{H^{-1}Z\}(2^J(n+1)) = \;& \textstyle\sum_{m=0}^{M_J-1} W_{m,J}\{H^{-1}Z\}(2^J(n-m)) + b_J \\ & +\Sigma^{1/2} Z(2^J(n+1)) \; .\end{aligned} \tag{4.38}$$

Each vector $X_J(2^J n)$ has the same dimension K_J as $S_J(X)(2^J n)$. The matrices $W_{m,J}$ and the vector b_J are the same as the ones corresponding to the whitening operator H.

4.4.1.2 *Convolutional layers* Y

The convolutional network Y constitutes the rest of the network G: it maps the layer X_J, which represents the scattering vectors obtained from the noise Z, to the temporal values X_0. The layers of Y are convolutional layers X_j for $0 \leq j \leq J$, where X_j is mapped to X_{j-1}

with an upsampling followed a convolution and a ReLU pointwise non-linearity. Let K_j denote the number of channels of each layer X_j, with $K_0 = 1$.

At each layer j, we first double the size of X_j with a zero insertion leading to $\check{X}_j$

$$\check{X}_j(2^j n) = X_j(2^j n) \text{ and } \check{X}_j(2^j n + 2^{j-1}) = 0 \ . \tag{4.39}$$

X_{j-1} is then calculated from $\check{X}_j$ as

$$X_{j-1} = \rho\left(W_j \check{X}_j\right) \ . \tag{4.40}$$

Following standard convolutional network architectures, the operator W_j is an affine operator along channels and a convolutional operator along time. More precisely, the parameters of W_j are matrices $W_{j,m}$ of size $K_{j-1} \times K_j$ for $0 \leq m < M_j$ and a bias vector b_j such that

$$\{W_j \check{X}_j\}(2^J n + k 2^{j-1}) = \sum_{m=0}^{M_j - 1} W_{j,m} \check{X}_j(2^J n + 2^j + (k-m) 2^{j-1}) + b_j. \tag{4.41}$$

Numerically, as is standard for deep architectures [GBC16], W_j is implemented as the composition of a linear operator and a batch normalization to ensure training stability. The last operator W_1 is not followed by a ReLU in order to be able to output a signal with negative values.

4.4.2 *Relative time shifts ensure causality*

A subtle aspect of the construction of the Scattering Inverse Network is how to handle the relative temporal shifts between each layer. In Figure 4.4, the network is shown to perform block-wise causal computations: each new vector $Z(2^J n)$ yields 2^J temporal points $X_0(t)$ for $2^J(n-1) < t_0 \leq 2^J n$. Actually, this is only the case when considering relative time-shifts $t_j = t - 2^j$ at each layer. We now explain the rationale behind these time-shifts.

We assume that the scattering of the encoder relies on the Gammatone wavelets (4.25). As a consequence, the low-pass filter ϕ_J as well as the wavelets are causal, so that the scattering transform S_J is causal. It results that $S_J(x)(2^J n)$ only depends on past values $\{x(t)\}_{t \leq 2^J n}$. Since the filters used are smooth in time, if $2^J(n-1) < t \leq 2^J n$ then the values of $x(t)$ have a relatively small impact on values of $S_J(x)(2^J k)$ for $k \leq n$. Recovering these $x(t)$ from $S_J(x)(2^J k)$ for $k \leq n$ would thus be unstable.

In order to remove the instability, the future value $S_J(x)(2^J(n+1))$ is used as well as previous values of $S_J(x)$ to retrieve the values $x(t)$ for which $2^J(n-1) + 1 < t \leq 2^J n + 1$. This means that a future value is used, and that this information is exploited to obtain an interpolation at finer scales $2^j < 2^J$. At training time, this future value is obtained from the groundtruth. At generation time, this future value is obtained with a linear prediction X_J at the largest scale 2^J, where $S_J(X)$ is approximately Gaussian; this prediction is then distributed over previous values at finer scales.

The interpolation is carried out by making a 2^j-time-shift at each layer x_j. From Equation (4.41), $X_{j-1}(2^J n + 2^{j-1})$ is obtained by accessing $X_j(2^J n + 2^j)$. As a consequence, the new value $X_J(2^J(n+1))$ yield 2^j values $X_j(t)$ at layer j for all j and for $2^J(n-1) + 2^j < t \leq 2^J n + 2^j$.

Introducing the relative time-shift

$$t_j = t - 2^j \; , \tag{4.42}$$

one can straighten the path followed by the algorithm, yielding Figure 4.4. The same graph with absolute times t would yield a tree slanted to the right, due to the causality of the encoder.

From an algorithmic point of view these time-shifts of internal network variables can be ignored, and computations exactly occur as in Figure 4.4. In SampleRNN [Meh+17] or WaveNet [Oor+16] networks, this time-shift does not appear because the internal variables are not related to the input value as our proposed encoder.

4.4.3 *Network training*

The parameters of the network $G = \Upsilon \circ H^{-1}$ are trained separately with respect to each operator H and Υ. Let $(x_i)_{1 \leq i \leq N}$ denote N independent time-series of length T used for training.

4.4.3.1 *Training H*

The whitening operator H is defined by Equation (4.36). Its parameters $W_{m,J}$, b_J and Σ are found by minimizing a prediction error on the training dataset :

$$\min_{W_{m,J}, b_J} \sum_{i=1}^{N} \sum_{n=M_J+1}^{\lfloor T2^{-J} \rfloor - 1} \left\| S_J x_i(2^J(n+1)) - \sum_{m=0}^{M_J - 1} W_{m,J} S_J x_i(2^J(n-m)) - b_J \right\|^2 . \tag{4.43}$$

This minimization can be carried out by solving the corresponding vectorial Yule-Walker equations, similar to the ones introduced in Chapter 2, or by performing a stochastic gradient descent. Once optimal parameters $W_{m,J}$ and b_J have been found, the matrix Σ is defined as the covariance matrix of the resulting prediction errors.

4.4.3.2 *Training* Υ

The parameters of the network Υ are optimized by inverting the scattering transform $S_J(X)$. The loss $\mathcal{L}$ is composed of two terms: a first term $\mathcal{L}_{\text{inv}}$ measuring the accuracy of the inversion, and a second discriminative term $\mathcal{L}_{\text{MM}}$ measuring the distance between the scattering moments of the synthesized signals and the scattering moments of the original signals:

$$\min_{Y} \mathcal{L} = \mathcal{L}_{\text{inv}} + \eta \mathcal{L}_{\text{MM}}, \tag{4.44}$$

where $\eta \geq 0$ is a trade-off parameter.

PERCEPTUAL LOSS The inverse problem loss $\mathcal{L}_{\text{inv}}$ computes the reconstruction error on each training example x_i from its scattering $S_J(x_i)$. The reconstruction error is calculated over scattering coefficients computed at a scale $2^K < 2^J$:

$$\mathcal{L}_{\text{inv}} = \frac{1}{N} \sum_{i=1}^{N} \| S_K(x_i) - S_K Y(S_J(x_i)) \|_1 \tag{4.45}$$

The ℓ^1 norm favors sparse responses, while the scattering S_K allows us to generate signals which may be locally deformed, but are perceptually similar to the original ones.

MOMENT-MATCHING LOSS The loss $\mathcal{L}_{\text{inv}}$ does not control the quality of the generated samples $G(z)$ when z is sampled as a Gaussian white noise and transformed into a pseudo-scattering as $H^{-1}z$. Similarly to GANs which have a discriminator, this is controlled by introducing another loss term $\mathcal{L}_{\text{MM}}$, which controls the distance between the generated distribution and the distribution of the original signals.

This moment-matching term controls the distance between scattering coefficients of generated signals averaged over time t and sub-batch index i, $\overline{S_K G(z_i)(t)}$, and scattering coefficients of the training signals averaged over time t and training examples i, $\overline{S_K x_i(t)}$:

$$\mathcal{L}_{\text{MM}} = \left\| \overline{S_K x_i(t)} - \overline{S_K G(z_i)(t)} \right\|^2 . \tag{4.46}$$

The codes $\{z_i\}$ correspond to a batch of random vectors which is renewed at each iteration of the gradient descent algorithm.

The loss $\mathcal{L}_{\text{MM}}$ is similar to the Maximum Mean Discrepancy regularization introduced in [Gre+07], which was already proposed for generative models [LSZ15; Li+17]. The moment matching term can be interpreted as a distance with a scattering transform kernel. However, in this case it can be directly implemented as a difference of empirical moments over the distributions.

4.5 EXPERIMENTAL VALIDATION

In this section, we numerically validate the modeling capacities of the proposed approach on music and speech signals. As explained in Section 1.2, these signals are non-Gaussian, exhibit very long dependencies and have a sparse decomposition in time-frequency dictionaries; as a consequence, they are an excellent testbed for our method.

We first describe in 4.5.1 the protocol used for the experiments. In 4.5.2, we investigate the impact of the loss term $\mathcal{L}_{\text{inv}}$ on the reconstruction quality. In 4.5.3, we evaluate the effect of the moment-matching term. In 4.5.4, we check the impact of the input scattering $\Phi(X)$ on the reconstruction and the generation. In 4.5.5, we investigate the ability of the network to perform meaningful interpolations in the signal space via linear interpolations in the latent space.

4.5.1 *Protocol*

We describe the experimental protocol used to perform our experimental validation.

4.5.1.1 *Metrics*

We evaluate the performance of the inverse scattering network G over three types of input data. Training errors compare $\Upsilon(S_J(x))$ with x for signals x which are in the training set. Testing errors are evaluated for signals x which are not in the training set. In both cases, the reconstruction error is measured in term of the scattering loss $\mathcal{L}_{\text{inv}}$ (4.45).

Generation properties are evaluated from realizations of the Gaussian white noise Z by computing synthesized signals $\tilde{X} = G(Z)$. We report the moment-matching loss $\mathcal{L}_{\text{MM}}$ (4.46). Insofar as our synthesis is not based on estimations of conditional probability distributions, we cannot report any log-likelihood measures to compare to Autoregressive Networks.

4.5.1.2 *Datasets*

We use three different datasets: TIMIT [FDGM86], NSynth [Eng+17] and Beethoven [Meh+17], which are standard for audio generation [Chu+15; HZG17; Meh+17].

For all datasets, the amplitudes of all recordings is normalized so as to fit in $[-1, 1]$, without adding any bias which would artificially create low frequencies. The sampling rate is reduced to 4096 Hz with a Kaiser filter [KS80], which corresponds to a low phone quality [GME11], so as to reduce the computational complexity. It is very likely that the quality of the synthesis could be greatly improved by increasing this sampling rate.

TIMIT TIMIT [FDGM86] contains 6300 sentences lasting each a few seconds, elicited by 630 different speakers. A pre-defined data split between the training and testing sets is provided by the authors.

NSYNTH NSynth [Eng+17] is a large dataset consisting of about $300,000$ annotated musical notes from multiple instruments. All recordings have a standardized length of 4 s, with the onset of the note at the beginning of the note, and a sample rate of 16000 Hz. We restrict ourselves to two classes of instruments: acoustic keyboards and acoustic flutes, totalling about 40 different instruments, with pitches on the MIDI scale ranging from 20 to 110. In the original dataset, there is no intersection between the instruments of the training set and the testing set. In this paper, we use an alternative split, based on the MIDI velocity's attributes of the original training set, which measures the force with which each note is played. for each considered instrument, a random velocity is picked to define the test set. We limit ourselves to the first 2 s of the recordings, insofar as it is the region which concentrates most of the energy across the whole dataset.

BEETHOVEN The Beethoven [Meh+17] dataset consists in 8 s extracts of Beethoven's piano sonata, with a sampling rate of 16000 Hz. In this case, we use the train-test split provided by the authors, leading to about 15×10^3 training samples after preprocessing and 1024 test samples. This dataset is closer to an actual musical composition than NSynth, insofar as it has a greater temporal density of musical events.

4.5.1.3 *Hyperparameters*

The scattering embedding introduced in Section 4.3 is computed with wavelets ψ^1 having $Q = 12$ intermediate frequencies per octave. In order to alleviate the large variations in the amplitudes of the scattering coordinates, scattering coefficients are renormalized coordinate-wise so that, for each coordinate, the mean over the whole training dataset is equal to 1.

The neural network loss (4.45) is computed with a scattering transform S_K calculated with a automatically differentiable implementation, adapted from the 2D version [OBZ17]. An independent release is currently undergoing in order to merge this 1D scattering transform with the 2D version. We use Morlet wavelets for S_K because causality is not required for this metric. The scale factor is chosen to be $2^K = 2^5 = 32$. Only first order coefficients are used, so that S_K is a time-averaged scalogram.

The parameters of the scattering inverse network are set as follows. The number of channels K_j of the convolutional network vectors X_j are chosen to grow arithmetically between $K_1 = 10$ and K_J as set in (4.33). The vector autoregressive filter H is of order $M_J = 4$, and for all $j < J$ the convolution filters have a time support of size $M_j = 7$. Between each convolution and the ReLU non-linearity, we use a batch-normalization [IS15], as is standard for generative networks [RMC16].

For each data-set and corresponding experiment, unless specified otherwise, we train a separate network. In all cases, the scattering inverse networks are trained for 1200 epochs using the Adam optimizer [KB14] with an initial learning rate of 10^{-3}.

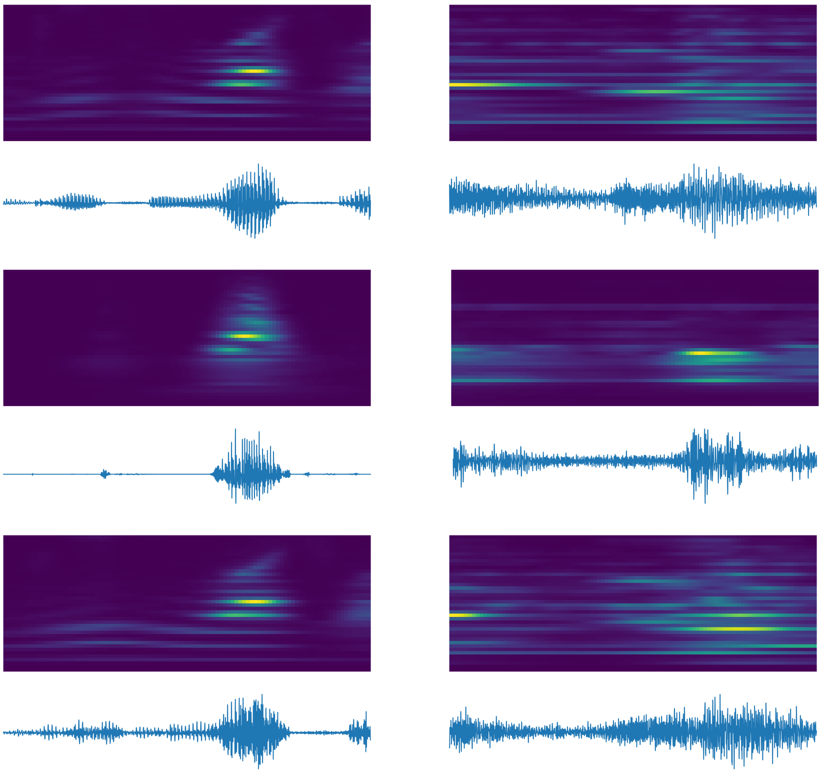

Figure 4.5: Impact of the metric on the reconstructions performed by Υ. Qualitative reconstruction examples on testing parts of all datasets. Left column: TIMIT example. Right column: Beethoven example. Top line: groundtruth excerpt. Middle line: reconstruction by a network trained with MSE. Bottom line: reconstruction by a network trained with perceptual metric. Note how both the waveforms and the spectral contents are much closer with the perceptual metric.

4.5.2 *Choice of the reconstruction metric*

We investigate the impact of the metric used in the reconstruction loss $\mathcal{L}_{\text{inv}}$ to measure the quality of the reconstruction of x from $S_J(x)$ by Υ. We compare the proposed perceptual metric (4.45)

$$\|S_K x - S_K \Upsilon(S_J(x))\|_1 \,, \tag{4.47}$$

to the baseline mean-square error (MSE)

$$\|x - \Upsilon(S_J(x))\|_2^2 \,. \tag{4.48}$$

We train two networks Υ with each of these metrics.

4.5.2.1 *Reconstruction results*

Figure 4.5 shows qualitative reconstruction examples on the training and testing sets of all datasets. We report in Table 4.1 quantitative

Training metric	TIMIT		Beethoven	
	Train	Test	Train	Test
MSE (4.48)	0.65	0.84	0.64	0.77
Perceptual (4.47)	0.22	0.50	0.14	0.34

Table 4.1: Reconstruction losses on both datasets. Reported numbers correspond to the perceptual metric, which is relative: a value close to 1 has 100% error, while a value close to 0 has 0% error. In both cases, the SIN Y was trained with the same hyperparameters, except for the reconstruction metric. Direcly training with the perceptual metric brings a clear quantitative improvement.

results. Notice that a network trained with MSE overfits the training set, and does not generalize the testing set. Early stopping could be thought of as a possible regularization, but then it would lead to bad quality reconstructions both on the training and testing sets. Directly training the network with the perceptual metric clearly improves results both qualitatively and quantitatively.

4.5.2.2 *Analysis*

We analyze two factors which contribute to the improvement of the results with the scattering metric.

FINITE NETWORK SIZE The Scattering Inverse Network (SIN) G is trained to retrieve the signal x from the encoder $\Phi(x)$. However, this network does not have access to the structure of Φ (*e.g.* through a computational graph), but only to samples $\{\Phi(x_i)\}_{1 \leq i \leq N}$, which are used in the empirical loss. Moreover, the network only has a finite size, meaning a finite set of weights to tune and therefore a finite "capacity" of patterns of operations to store [Aro+17]. In order to avoid wasting the finite capacity on patterns which are non-discriminative, the training of the network should incentivize the perceptual content of the reconstructed samples.

The MSE metric is not only sensitive to the local frequencies of the signals, but also to the absolute phase of each partial. Because of the wavelet modulus, the scattering transform supporting the encoder $\Phi(x)$ loses the absolute phase of these partials. As a consequence, when trained with the MSE loss, the network G needs to store the patterns in the input allowing to retrieve the local phase for each input. This raw memorization is useful at training time, but does not allow to generalize. The MSE metric is therefore prone to overfitting.

Using a scattering metric allows the model to focus on the perceptual content of the original and reconstructed signals, and to compare them accurately. Since absolute phases are also lost with this metric, the

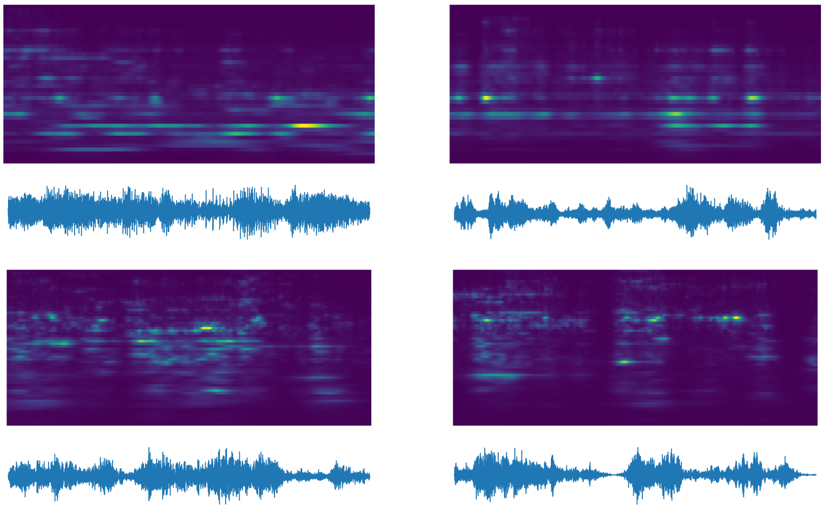

Figure 4.6: Generation examples from networks trained with (top) and without (bottom) a moment-matching term on the Beethoven dataset.

Method	$\mathcal{L}_{\text{inv}}$	$\mathcal{L}_{\text{inv}} + \mathcal{L}_{\text{MM}}$	Testing set
$\mathcal{L}_{\text{MM}}$	38.7	0.176	0.334

Table 4.2: Moment matching loss term on the Beethoven dataset. Note how adding the moment-matching term allows us to create samples whose distribution is as close to the training set as to the test set.

network does not waste its finite capacity to remember them, and does generalize.

RECOVERY OF HIGH FREQUENCIES The scattering metric enjoys the benefits of the scattering transform, including its invariance to small time-frequency deformations. This invariance is useful to retrieve high-frequency components. Under the MSE metric, solutions to the deconvolution problem typically lack high-frequency components as a small error in the position yields a large error in the reconstruction. The invariance to small translations of the scattering metric allows us to retrieve high-frequency components of the signal, both on the training and testing sets of the signal.

4.5.3 *Impact of the moment matching loss*

The effect of the moment matching loss on the generated samples is difficult to assess qualitatively. Figure 4.6 shows time-series generated from different white noise realizations z_i with a network G trained with and without the moment matching term on the Beethoven

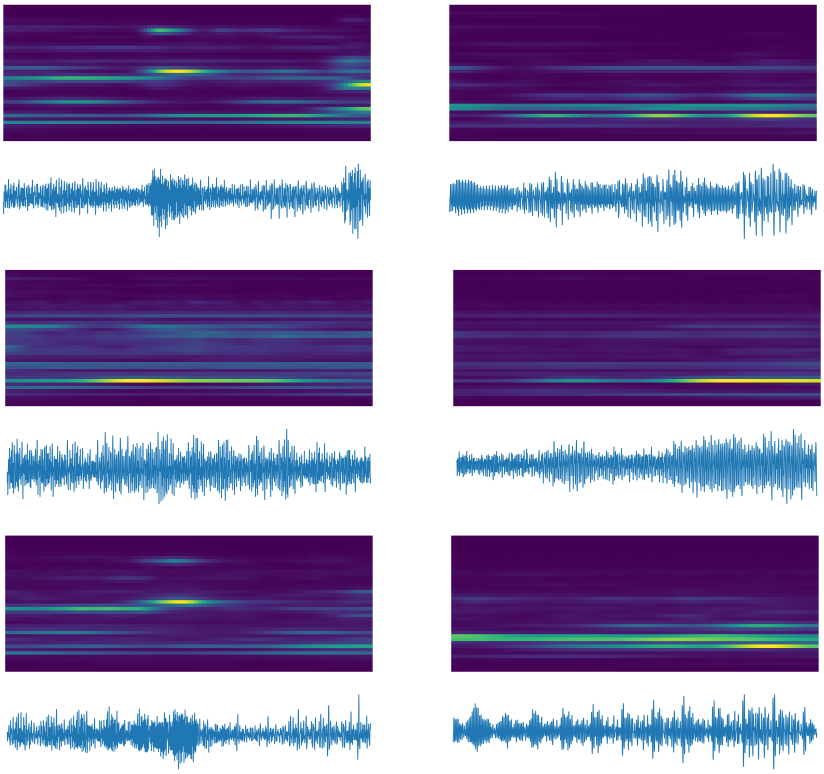

Figure 4.7: Comparison of the reconstruction with and without the moment matching term $\mathcal{L}_{\text{MM}}$ on the testing part of the Beethoven dataset. Each column corresponds to a different example. The top line is the original signal, the middle line is the reconstruction from a network trained without the moment-matching term and the bottom line the reconstruction from a network trained with the moment matching term. Notice the clear improvement in quality when adding the moment matching term.

Data split / Method	Perceptual	Perceptual + Moment Matching
Train	0.16	0.23
Test	0.37	0.31
Relative gap test/train (dB)	3.53	1.21

Table 4.3: Reconstruction error results on the Beethoven dataset. Adding the moment-matching term during training improves the reconstruction results and the generalization.

dataset. One observes a slight qualitative difference between the log-spectrograms, as those obtained with a network trained with this term appear to be slightly more structured. Quantitatively, there is a large gap between both distributions, as seen in Table 4.2. In terms of scattering moments, the samples generated from the MM-SIN are almost indistinguishable from the training samples, witness relative moments which even closer to the training dataset than the testing set. Thus, the distributions are much closer in this weak metric. In order to measure the variability of the generated samples, we measure the spread σ of the distribution of the time-averaged scattering coefficients $S_K(X)$ of the samples. This spread corresponds to the average Euclidean distance between the time-averaged scattering of the waveforms and the average scattering coefficients of this distribution. In the case of the training distribution, $\sigma = 6.69$ is obtained, whereas $\sigma = 3.51$ is obtained for the distribution generated from white noise. This shows that the generated samples exhibit a non-negligible variability, even though it is lower than the one expressed in the training set.

We now investigate whether the moment matching term improves the reconstruction. We hypothesize that the loss term $\mathcal{L}_{\text{MM}}$ should regularize the network and provide an additional supervision. As such, the generalization of the network to unknown data points should be improved. Figure 4.7 displays several reconstruction examples from networks trained with and without moment matching. Qualitatively, we observe that this additional loss term allows the network to retrieve several frequential contents which are lost otherwise. Quantitatively, Table 4.3 shows that the moment matching term slightly hurts results on the training set, but improves results on the testing set and also reduces the generalization gap between the training set and the testing set.

4.5.4 *Input representation $\Phi(X)$*

In this section, we confirm the choice of the input representation $\Phi(X)$, which was solely based on prior information on signals X. We first perform an ablation experiment to validate the usage of second-order scattering coefficients. We then show that the scale parameter 2^J is a trade-off parameter between reconstruction and generation, which hints at the underlying Gaussianization occurring.

4.5.4.1 *Second order coefficients*

We consider the ablation of the second-order terms in the representation $S_J(X)$. Because of the temporal averaging with ϕ_J, this ablation is expected to yield reconstructions of lower quality. This is demonstrated quantitatively in Table 4.4 and qualitatively in Figure 4.8.

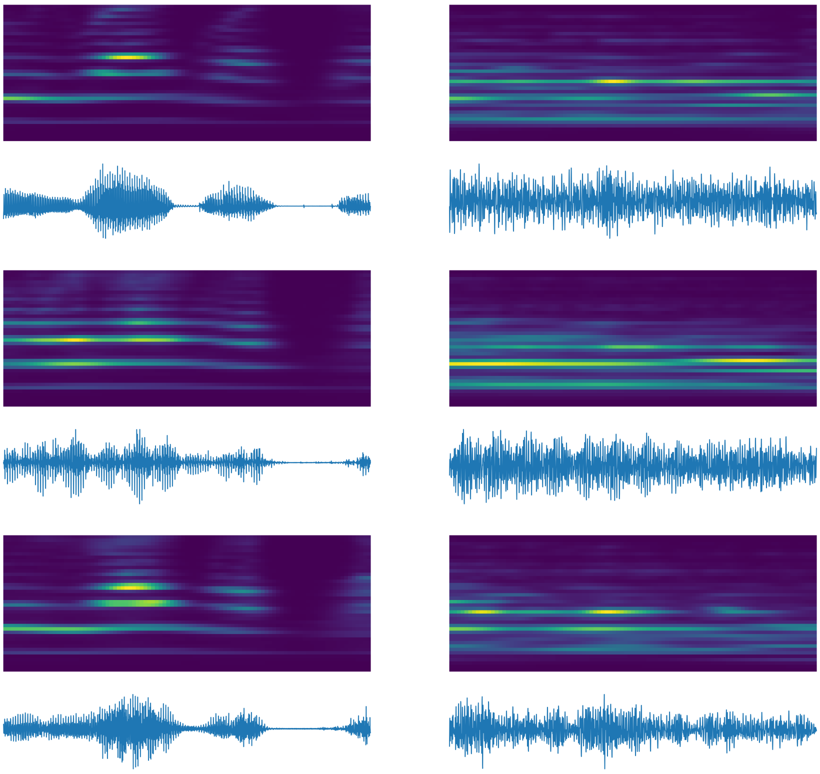

Figure 4.8: Reconstruction from an ablated scattering (no second order terms) in the middle line and from a full scattering in the bottom line, compared to the original signal (top line). Left: TIMIT example. Right: Beethoven example. Notice the improvement in quality, notably for the Beethoven dataset: the second-order terms allow to recover the temporal dynamics within the dominating frequency.

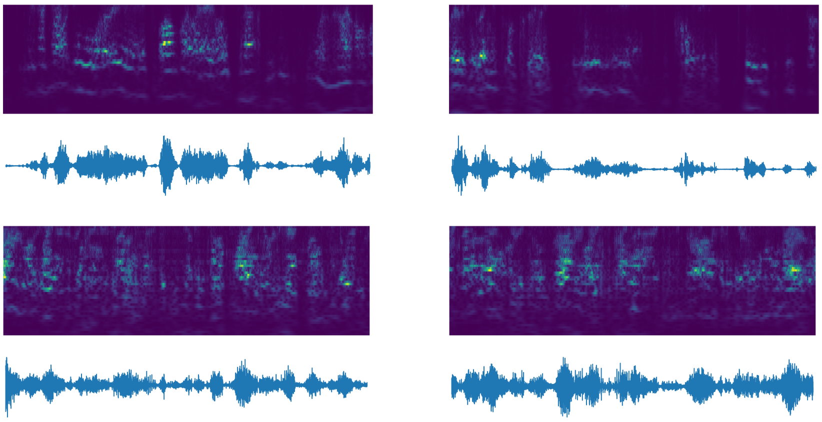

Figure 4.9: Generation from Gaussian white noise $G(Z)$ on the TIMIT dataset. Top examples: computed with $J = 10$. Bottom examples: with $J = 6$. As the scale 2^J increases, $S_J(X)$ becomes more Gaussian and the model is more realistic. The duration of each time series is 3.3 seconds.

4.5.4.2 *Averaging scale* 2^J

We show that the scale 2^J is a trade-off between Gaussianization (leading to good generation) and invertibility (leading to a good recovery). Figure 4.9 displays generation examples for $J = 10$ and $J = 6$ on the TIMIT data-set. As the scale 2^J increases, $S_J(X)$ becomes more Gaussian and the model is more realistic.

4.5.5 *Interpolation examples*

We now study the ability of the algorithm to transform the pitch of musical signals with arithmetic operations in the latent space. We use the NSynth dataset, whose careful construction allows us to perform modifications with fixed factors of variability. In the test set, we pick two samples belonging to the same instrument, but with a pitch separated by 5 MIDI scales. We compute their embeddings Z_1 and Z_2, their mean embedding $(Z_1 + Z_2)/2$, and reconstruct the corresponding signals with the generator: $G(Z_1)$, $G(Z_2)$, and $G((Z_1 + Z_2)/2)$.

The results are displayed in Figure 4.10. The interpolation in the latent space does not result in a linear interpolation in the signal space, which would double the number of harmonics. It yields one fundamental frequency in each case. Furthermore, this fundamental frequency is indeed interpolation by this simple arithmetic. Observe that this is also the case of the partials, as can be seen in particular in the bottom example. However, this interpolation suffers from some artifacts: for instance, in the middle example, the partials at highest frequencies are cluttered and the resulting signal misses harmonicity. Yet, these results showcase the ability to transform signals via simple linear interpolations in the latent space with a simple unsupervised learning procedure and a predefined embedding.

Scattering / Dataset	TIMIT		Beethoven	
	Train	Test	Train	Test
1st order (ablated)	0.37	0.64	0.28	0.39
1st and 2nd order (full)	0.22	0.50	0.14	0.34

Table 4.4: Reconstruction losses on both datasets, measured via the perceptual metric (lower is better). In both cases, the network was trained with the perceptual metric. The second-order terms (full scattering) bring a clear quantitative improvement.

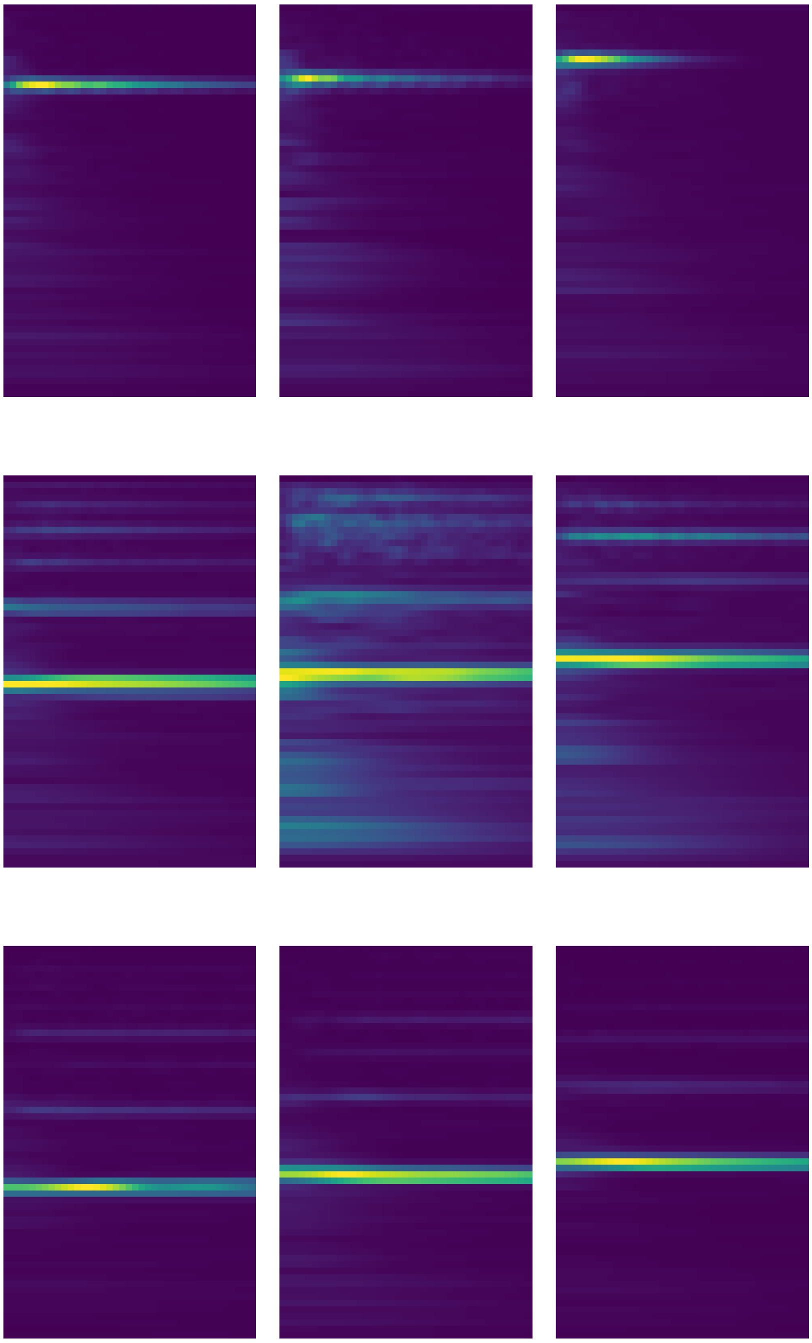

Figure 4.10: Pitch interpolation. Left column: $G(Z_1)$. Middle column: $G((Z_1 + Z_2)/2)$. Right column: $G(Z_2)$. Z_1 and Z_2 are the embeddings of samples from the test set. The generator interpolates the fundamental frequency with a simple arithmetic. The frequential displacement from left to right corresponds to 5 MIDI scales.

4.6 CONCLUSION

This chapter investigates which assumptions can be used to derive estimators of probability densities in a tractable fashion, and which algorithms can be used to build the corresponding models.

Inspired by state-of-the-art deep learning algorithms and by recent works in image generation, this chapter introduces a scattering autoencoder architecture to build such an estimator. The encoder is defined as a whitened scattering transform thanks to assumptions on the signals, notably the existence of sparse time-frequency decompositions and decorrelation over long time intervals, in order to Gaussianize the input signal with local averages. The generator is a causal convolutional neural network which maps back the resulting codes to raw waveforms. The resulting system synthesizes new realistic signals and performs the transformation of low-level attributes, such as pitch, by simple linear interpolation in the latent space.

Although this work proposes an estimator which is trained in a tractable fashion, it still relies on the implicit regularization of convolutional neural networks to perform the generation. The nature of the regularization performed by this network to compute the inverse is still an open research question. It would be worth investigating the nature of the operations performed by the generative network in order to gain more understanding of the implicit assumptions it postulates.

5 CONCLUSION

We now summarize the main findings of this dissertation on time-series modeling, discuss their limitations and propose future research directions.

5.1 SUMMARY OF CONTRIBUTIONS

5.1.1 *Linear forecasting under long-range dependencies*

The first problem considered is the linear forecasting of a Gaussian long-range dependent process $(X(t))_t$, which consists in regressing $X(t+\Delta)$ based on the knowledge of the past $X(\leq t)$. Because of the Gaussian prior, it is sufficient to consider linear estimators of the future value. The power-tail behavior of the autocovariance operator γ_X implies that $X(\leq t)$ needs to be large to achieve a low mean-square error, but this increases the number of parameters to estimate. In this framework, Chapter 2 investigates how to represent the past $X(\leq t)$ for forecasting purposes.

State-of-the-art representations for long-range dependent processes rely on wavelets [DOT03]. Indeed, dilation properties make them amenable to the power-tail behavior of such processes [Fla92]. An analysis of the structure of the Yule-Walker equation, which governs linear forecasting, for a closed-form model of long-range dependence, leads us to consider a particular subspace induced by wavelet: the foveal cone. This foveal cone consists in the wavelets closer to the present t, at all scales. However, the causality constraint implies that only Haar wavelets can really be useful for forecasting purposes, which leaves little flexibility.

Our main contribution consists in introducing a new class of foveal wavelets, which is simpler than the existing ones [Mal03]. These new wavelets rely on the choice of a window function, which is multiplied by polynomials and dilated at multiple scales. They are therefore foveal by design, while the space of functions they span can be increasingly refined thanks to the polynomials. Three classes of such wavelets are proposed: indicator, Gaussian and exponential foveal wavelets. Numerical experiments on synthetic and real time-series exhibiting a long-range behavior demonstrate that they result in a lower mean-square error than the foveal Haar family.

5.1.2 *Non-linear forecasting of sparse time-frequency processes*

In many instances of time-series forecasting problems, the algorithms yielding the lowest error are linear estimators, despite being in a non-Gaussian regime. Chapter 3 therefore investigates which priors could be leveraged for non-linear algorithms to outperform linear ones.

In signal processing, the prior knowledge of the existence of a sparse decomposition has led to improvements over linear algorithms for many tasks, for instance inpainting [Mal08]. Recently, non-linear deep neural networks algorithms have achieved exceptional results at modeling sparse time-frequency time-series, such as audio [Oor+16]. Therefore, sparse time-frequency time-series are good candidates for our forecasting problem.

Our main contribution in this chapter consists in formalizing how to exploit a sparse time-frequency decomposition for forecasting purposes. Building on the analysis of a trained multi-layer perceptron (MLP) and a simple cosine model, a non-linear algorithm which performs forecasting in a sparse inverse problem framework is proposed. This algorithm is extended in a foveal multiscale fashion in order to accommodate more complex signals. Forecasting experiments on artificial and real data show that this algorithm performs almost as well as a trained MLP for a forecasting task, thereby validating our approach.

5.1.3 *Time-series Generation*

The last problem we tackle is time-series generation, focusing on those exhibiting a sparse time-frequency behavior, such as audio signals. These time-series follow a non-Gaussian distribution and exhibit long-range dependencies, which makes the probability density estimation problem very challenging. In this context, we investigate which assumptions on the signals can be leveraged to build models of such time-series, and which algorithms can be used to learn these models.

Recent developments of deep learning algorithms [Oor+16; RMC16] have shown that these networks are able to generate realistic natural signals, such as images or music time-series, with a spectacular quality. However, the understanding of these performances remains at best elusive [Aro+17; ARZ18]. Indeed, the explicit assumptions are not sufficient to explain the quality of the generated samples. It is likely that these networks exploit implicit assumptions on the signals they model, but it is difficult to uncover these priors due to their sheer complexity.

Building on the framework proposed in [AM18b] in the case of images, we introduce an autoencoder approach to model time-series. The encoder part is not learned and defined thanks to assumptions on the signals as a whitened time-frequency scattering transform. Under the stationarity hypothesis and assuming the signals exhibit sparse time-frequency decompositions which decorrelate at long times, this encoder maps the signals to approximate Gaussian random variables because of the central limit theorem.

The decoder consists in a convolutional neural network which takes into account the particular temporal structure of the encoder, and closely resembles state-of-the-art architectures. The decoder training is cast as the inverse problem of retrieving the samples from their encodings, thereby avoiding any reference to probability distributions. In order to make this inversion tractable, we have proposed to use a scattering metric, which exploits at best the capacity of the neural network to reconstruct the perceptual content. We have also proposed a moment-matching loss to better control the behavior of the network at generation time.

Numerical experiments on speech and music signals demonstrate the ability of the proposed architecture to reproduce and generate realistic samples. In particular, we show that the use of a pre-defined encoder regularizing small time-frequency deformations allows us to perform transformations of low-level attributes of music, such as pitch, through a linear interpolation in the latent space.

5.2 PERSPECTIVES

We now discuss possible improvements for the work exposed in this thesis and propose new directions of research.

5.2.1 *Spatiotemporal forecasting*

In this dissertation, we have only considered univariate time-series forecasting, both in a linear and non-linear fashion. Our work has thus focused on the temporal axis: in Chapter 2, we have considered long-range dependent time-series, while in Chapter 3 we have considered sparse time-frequency processes. The resulting modeling has allowed us to derive tractable estimators in both cases.

However, many time-series of interest are generated in groups, leading to multivariate time-series $(\mathbf{x}(t))_t$ where $\mathbf{x}(t) \in \mathbb{R}^d$ is a vector. It is often useful to consider the joint trajectory of this vector, leading to a multivariate forecasting problem. This is notably the case in fields as diverse as finance, medicine, fluid dynamics or video analysis.

Just like univariate forecasting, multivariate forecasting suffers from the curse of dimensionality, which is amplified by the dimension d of the vectors. Indeed, for a similar past size τ, one needs to consider probability distributions defined in dimension $d\tau$ instead of τ. It thus becomes paramount to reduce the dimension of the problem, which requires to make assumptions on the data.

It would be possible to devise extensions of the algorithms proposed in Chapter 2 and 3. Under the assumption that each univariate time-series follows the same priors as the ones assumed for the corresponding algorithm, one can simply use a different representation of the past for each time-series, and use the resulting vector for forecasting the next vector. The dimension is thus reduced across the temporal dimension, but the dimension of the channels is left unchanged. Preliminary experiments in the linear forecasting case have shown little effect on the forecasting accuracy for financial stocks.

In fact, the fundamental richness of this problem lies in the interaction between the different dimensions of the time-series. When these coordinates correspond to an underlying structure, recent works have shown that it is possible to exploit this prior to improve forecasting. For instance, in meteorology, the different coordinates of the time-series correspond to an underlying spatial grid, which allows convolutional neural networks to exploit this structure [Xin+15]. Similarly, when each coordinate corresponds to the positions of a point cloud sampled from a low-dimensional manifold, as is the case for three-dimensional shapes, smart representations stemming from harmonic analysis exploiting this geometry allow to improve forecasting [Kos+18].

Eventually, an accurate forecasting model for multivariate time-series will involve a dimension reduction jointly performed across the temporal and coordinates axis, and not simply factorized across each of the axis. It is likely that the tools proposed in this dissertation for temporal forecasting will remain useful in such a joint setting.

5.2.2 *Beyond MSE prediction*

In this dissertation, we have mainly considered deterministic forecasting in the mean-square error sense. This amounts to estimating the conditional expectation $\mathbb{E}[X(t+\Delta)|X(\leq t)]$ instead of the whole probability density $p_X(x(t+\Delta)|x(\leq t))$.

The MSE criterion is relevant for Gaussian stationary time-series, as is the case in Chapter 2, since the variance of the variable $X(t+\Delta)$ can be extrapolated from the past, so that the probability density is entirely known. However, for non-Gaussian time-series such as sparse time-frequency processes considered in Chapter 3, this criterion leads to a quantity which characterizes very little of the probability density. In order to allow for a better modeling, one would need more than conditional means.

We therefore face the question of characterizing the conditional density $p_X(x(t+\Delta)|x(\leq t))$. Despite the fact that $x(t+\Delta)$ is one-dimensional, the past $x(\leq t)$ is still large so that this problem suffers from the curse of dimensionality. The main question is therefore to understand which priors allow to derive tractable estimators for this quantity.

Deep autoregressive networks such as WaveNet [Oor+16] estimate a full probability density for $\Delta = 1$ with a complex estimator. This estimator relies on an intricate architecture which involves convolutions, multiplicative gates and residual connections. The interpretation of this network is therefore very challenging.

As we have shown both for forecasting with a MSE criterion and for generation purposes, using a sparse time-frequency decomposition allows us to simplify such estimation problems. It is therefore probable that the WaveNet exploits this sparse time-frequency to perform this task. Yet, to the best of our knowledge, no empirical analysis of these networks has been performed to ground this fact. Moreover, it is unclear whether these networks exploit other properties of audio or speech signals. For instance, are larger coherent structures used as well? We believe that performing such an analysis would result in an increased knowledge both with respect to the signals and to deep learning algorithms.

5.2.3 *Invertible linearized dynamics*

A pregnant idea in the study of both univariate and multivariate time-series is the linearization of dynamics: let us assume that the original time-series $(x(t))$ follows a non-linear dynamics:

$$x(t+1) = f(x(\leq t), \zeta(t)), \tag{5.1}$$

where $\zeta(t)$ is a noise variable, possibly non-Gaussian. Ideally, one would like to find an invertible and causal mapping Φx such that

$$\Phi x(t+1) = A\Phi x(t) + Z(t), \tag{5.2}$$

where A is a linear operator and Z a Gaussian noise. In this case, the time-series x could be predicted through the mapping Φ.

In Chapter 4, we have exploited the existence of a sparse time-frequency decomposition as well its decorrelation at long time intervals to map the original signal to a vector-valued time-series with an approximate Gaussian distribution, using the scattering transform. This approximate Gaussian distribution has allowed us to whiten the scattering vectors with a linear autoregressive model. In other words, the dynamics of the original process have been linearized by the scattering transform.

However, there are several obstacles to the use of the scattering operator as a forecasting operator. First, we have seen that causality constraints prevents the recovery of $(x(u))_{u \leq t}$ from $(S_J x(u))_{u \leq t}$: typically, one can only recover $x(u)$ until $x(t - 2^J)$. Second, and more importantly, the scattering transform removes the phase of the wavelet transforms. This means that the original signal x can only be recovered up to a global phase. Since forecasting is very sensitive to the phase, the scattering transform seems to be ill-suited for this purpose.

A promising direction of research would therefore consist in developing an analogue of the scattering transform, but invertible in a pointwise fashion. This new operator would not only provide a simple and elegant forecasting algorithm, but could also be used to study and characterize dynamical systems whose behavior remains poorly understood, such as turbulent fluid dynamics.

A APPENDIX

A.1 PROOF OF PROPOSITION 2.2.1

Let us introduce a convenient notation for Taylor expansions of non-integer power functions. For any $\beta \in \mathbb{R} - \mathbb{N}$ and any integer $n \in \mathbb{N}$, define

$$\binom{\beta}{n} = \begin{cases} 1 & \text{if } n = 0, \\ \frac{1}{n!} \prod_{k=0}^{n-1} \beta - k & \text{otherwise.} \end{cases} \tag{A.1}$$

This notation makes it possible to easily write the derivative of power functions u^β for β non integer, as:

$$\frac{d^k}{du^k}\left(u^\beta\right) = k!\binom{\beta}{k} u^{\beta-k}. \tag{A.2}$$

Let us now state and prove a technical lemma, which will be useful for the main proof.

Lemma A.1.1. *Let I be a compact interval of non-negative real numbers, $0 < \nu < 1$ and M be a positive integer. For any integer $n \geq 1$ and real $u \in I$, the following decomposition holds:*

$$(u+n)^{-\nu} = n^{-\nu} + \sum_{k=1}^{M-1} \binom{-\nu}{k} u^k + R_{M-1}(u), \tag{A.3}$$

where $R_{M-1}(u)$ is bounded as

$$|R_{M-1}(u)| \leq n^{-(\nu+M)} u^M. \tag{A.4}$$

Proof of Lemma A.1.1. For any $n \geq 1$, let us introduce the function

$$\begin{array}{rccc} f_n : & [0,+\infty) & \to & \mathbb{R}^+ \\ & u & \mapsto & (u+n)^{-\nu}. \end{array} \tag{A.5}$$

f_n is infinitely differentiable, and in particular one gets:

$$f_n^{(M)}(u) = M!\binom{-\nu}{M}(u+n)^{-(\nu+M)}. \tag{A.6}$$

The positive quantity $u \mapsto (u+n)^{-(\nu+M)}$ decreases with u as $\nu > 0$, so that

$$\forall u \geq 0, |u+n|^{-(\nu+M)} \leq n^{-(\nu+M)}. \tag{A.7}$$

Using the bound

$$\left| M!\binom{-\nu}{M} \right| = \prod_{k=0}^{M-1} |-\nu-k| \leq \prod_{k=0}^{M-1}(1+k) = M!, \tag{A.8}$$

where we have used $|-\nu-k| = \nu+k \leq k+1$, we thus obtain the bound:

$$\forall u \geq 0, \forall n \geq 1, |f_n^{(M)}(u)| \leq M! n^{-(\nu+M)} := C_n. \tag{A.9}$$

Using this upper bound C_n on $f_n^{(M)}$, we now apply Taylor-Lagrange's theorem to f_n at order $M-1$ on I: for all $u \in I$,

$$\left| \underbrace{f_n(u) - f_n(0) - \sum_{k=1}^{M-1} \frac{f_n^{(k)}(0)}{k!} u^k}_{R_{M-1}(u)} \right| \leq C_n \frac{u^M}{M!}. \tag{A.10}$$

Since

$$\frac{f_n^{(k)}(0)}{k!} = \binom{-\nu}{k} n^{-(\nu+k)}, \tag{A.11}$$

and $f_n(0) = n^{-\nu}$, we recognize that

$$R_{M-1}(u) = (u+n)^{-\nu} - n^{-\nu} - \sum_{k=1}^{M-1} \binom{-\nu}{k} u^k. \tag{A.12}$$

Since $C_n = M! n^{-(\nu+M)}$, we get that for all $u \in I$,

$$|R_{M-1}(u)| \leq n^{-(\nu+M)} u^M. \tag{A.13}$$

Hence the result. □

We can now proceed to the main proof.

Proof of Proposition 2.2.1.
Let us write

$$\langle u^{-\nu}, \psi_{j,n} \rangle = \int_0^{+\infty} u^{-\nu} \psi_{j,n}(u) du. \tag{A.14}$$

Thanks to the change of variable $z = 2^{-j}u$, for $n \in \mathbb{N}$ it holds that

$$\langle u^{-nu}, \psi_{j,n} \rangle = 2^{j\left(\frac{1}{2}-\nu\right)} \langle z^{-\nu}, \psi_{0,n} \rangle. \tag{A.15}$$

Thus, the decomposition of the function $u \mapsto u^{-\nu}$ has no characteristic scale, and all scales 2^j carry information.

The change of variable $u = z - n$ gives

$$\langle z^{-\nu}, \psi_{0,n} \rangle = \int_{-n}^{+\infty} (u+n)^{-\nu} \psi(u) du. \tag{A.16}$$

Let I denote the support of ψ, which is compact and included in $[0, +\infty)$ by assumption. We can restrict the right-hand-side integral to this interval:

$$\langle z^{-\nu}, \psi_{0,n} \rangle = \int_I (u+n)^{-\nu} \psi(u) du. \tag{A.17}$$

Using Lemma A.1.1, we get that for all $u \in I$,

$$(n+u)^{-\nu} = n^{-\nu} + \sum_{k=1}^{M-1} \binom{-\nu}{k} u^k + R_{M-1}(u), \tag{A.18}$$

and the residual R_{M-1} is bounded as

$$|R_{M-1}(u)| \leq n^{-(\nu+M)} u^M. \tag{A.19}$$

Using the property of vanishing moments of ψ for $k < M$, we obtain

$$\langle u^{-\nu}, \psi_{0,n} \rangle = \int_0^{+\infty} R_{M-1}(u)\psi(u)du. \tag{A.20}$$

Bounding the right-hand-side integral by its absolute value, we thus get

$$|\langle u^{-\nu}, \psi_{0,n} \rangle| \leq n^{-(\nu+M)} \underbrace{\int_I u^M |\psi(u)| du}_{\text{Constant independent of } n}, \tag{A.21}$$

where the last integral is finite because I is compact.

Equations (A.15) and (A.21) prove Equation (2.57). □

A.2 NUMERICAL ALGORITHMS CONSTRUCTING THE FOVEAL WAVELETS

Algorithm 1 Indicator foveal wavelets ψ_j^m generation

```
Input: Scale $J$, Maximal polynomial order $M$
1) Construction of the functions $\phi_j^m$:
for $j = 0, \dots, J$ do
    Orthonormalize $\{(t^m)_{t \in D_j}\}_{m<2^j}$ via the Gram-Schmidt algorithm, resulting in $\{\phi_j^m\}_{m<2^j}$
end for
2) Construction of the foveal wavelets $\psi_j^m$:
Initialization: $\psi_0^0 \to \phi_0^0$
for $j = 1, \dots, J$ do
    for $m = 0, \dots, M$ do
        if $n < 2^{j-1}$ then
            $c_{jm}^b \leftarrow \left(\sum_{t \in D_j} \phi_j^m(t) t^n\right) \left(\sum_{t \in D_{j-1}} \phi_{j-1}^m(t) t^m\right)^{-1}$
            $\bar{\psi}_j^m \leftarrow \phi_j^m - c_{jm}^b \phi_{j-1}^m$
        else
            $c_{jm}^b \leftarrow \left(\sum_{t \in D_j} \phi_j^m(t) t^m\right) \left(\sum_{t \in D_{j-1} \cup D_j} \psi_{j-1}^m(t) t^m\right)^{-1}$
            $\bar{\psi}_j^m \leftarrow \phi_j^n - c_{jm}^b \psi_{j-1}^m$
        end if
        $\psi_j^m \to \bar{\psi}_j^m / \|\bar{\psi}_j^m\|_2$
    end for
end for
return $\psi_j^n$ and $\phi_0^0$
```

Algorithm 2 Gaussian foveal wavelets ψ_j^m generation

```
Input: Scale $J$, Maximal polynomial order $M$
1) Construction of the functions $\phi_j^m$:
for $j = 0, \dots, J$ do
    if $j = 0$ then
        $\phi_0^0 \leftarrow \delta_0$
    else
        Orthonormalize $\left\{ \left( t^m (\theta^{(G)}(2^{-(j+1)}t) - \theta^{(G)}(2^{-j}t)) \right)_{t \in [0, S_j]} \right\}_{m < 2^j}$
via the Gram-Schmidt algorithm
    end if
end for
2) Construction of the foveal wavelets $\psi_j^m$:
Initialization: $\psi_0^0 \rightarrow \phi_0^0$
for $j = 1, \dots, J$ do
    for $m = 0, \dots, M$ do
        if $m < 2^{j-1}$ then
            $c_{jm}^b \leftarrow \left( \sum_{t \in [0, S_j]} \phi_j^m(t) t^m \right) \left( \sum_{t \in [0, S_{j-1}]} \phi_{j-1}^m(t) t^m \right)^{-1}$
            $\bar{\psi}_j^m \leftarrow \phi_j^m - c_{jm}^b \phi_{j-1}^m$
        else
            $c_{jm}^b \leftarrow \left( \sum_{t \in [0, S_j]} \phi_j^m(t) t^m \right) \left( \sum_{t \in [0, S_j]} \psi_{j-1}^m(t) t^m \right)^{-1}$
            $\bar{\psi}_j^m \leftarrow \phi_j^n - c_{jm}^b \psi_{j-1}^m$
        end if
        $\psi_j^m \rightarrow \bar{\psi}_j^m / \|\bar{\psi}_j^m\|_2$
    end for
end for
return $\psi_j^m$ and $\phi^0$
```

BIBLIOGRAPHY

[Aba+16] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. "Tensorflow: a system for large-scale machine learning." In: *OSDI*. Vol. 16. 2016, pp. 265–283.

[AH+14] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. "Convolutional Neural Networks for Speech Recognition." In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22.10 (2014), pp. 1533–1545.

[AS73] Milton Abramovitz and Irene A. Stegun, eds. *Handbook of mathematical functions : with formulas, graphs and mathematical tables*. 9th. Vol. 1. Dover Publications, New York, 1973.

[AV98] Patrice Abry and Darryl Veitch. "Wavelet analysis of long-range-dependent traffic." In: *IEEE Transactions on Information Theory* 44.1 (1998), pp. 2–15.

[AVF98] Patrice Abry, Darryl Veitch, and Patrick Flandrin. "Long-range Dependence: Revisiting Aggregation with Wavelets." In: *Journal of Time Series Analysis* 19.3 (1998), pp. 253–266.

[Adl+12] Amir Adler, Valentin Emiya, Maria G Jafari, Michael Elad, Rémi Gribonval, and Mark D Plumbley. "Audio inpainting." In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.3 (2012), pp. 922–932.

[AEB06] M. Aharon, M. Elad, and A. Bruckstein. "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation." In: *IEEE Transactions on Signal Processing* 54.11 (2006), pp. 4311–4322.

[Alp93] Bradley K. Alpert. "A Class of Bases in L^2 for the Sparse Representation of Integral Operators." In: *SIAM Journal on Mathematical Analysis* 24.1 (1993), pp. 246–262.

[ALM15] J. Anden, V. Lostanlen, and S. Mallat. "Joint Time-Frequency Scattering for Audio Classification." In: *Proc. of IEEE MLSP*. 2015.

[AM14] J. Andén and S. Mallat. "Deep scattering spectrum." In: *IEEE Transactions on Signal Processing* 62.16 (2014), pp. 4114–4128.

[AM18a] Mathieu Andreux and Stéphane Mallat. "Music Generation and Transformation with Moment Matching-Scattering Inverse Networks." In: *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, Paris, France*. 2018.

[AM18b] Tomàs Angles and Stéphane Mallat. "Generative networks as inverse problems with scattering transforms." In: *International Conference on Learning Representations*. 2018.

[App04] David Applebaum. "Lévy processes - from probability theory to finance and quantum groups." In: *Notices of the American Mathematical Society* 51.11 (2004), pp. 1336–1347.

[Arı+17] Sercan Ö. Arık et al. "Deep Voice: Real-time Neural Text-to-Speech." In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. 2017, pp. 195–204.

[ACB17] Martín Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein GAN." In: *CoRR* abs/1701.07875 (2017).

[Arn+98] A. Arneodo, E. Bacry, S. Jaffard, and J. F. Muzy. "Singularity spectrum of multifractal functions involving oscillating singularities." In: *Journal of Fourier Analysis and Applications* 4.2 (1998), pp. 159–174.

[ARZ18] Sanjeev Arora, Andrej Risteski, and Yi Zhang. "Do GANs learn the distribution? Some Theory and Empirics." In: *International Conference on Learning Representations*. 2018.

[Aro+17] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. "Generalization and Equilibrium in Generative Adversarial Nets (GANs)." In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, 2017, pp. 224–232.

[AL01] David Attwell and Simon B Laughlin. "An energy budget for signaling in the grey matter of the brain." In: *Journal of Cerebral Blood Flow & Metabolism* 21.10 (2001), pp. 1133–1145.

[BMM15] Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. "Hawkes Processes in Finance." In: *Market Microstructure and Liquidity* 01.01 (2015), p. 1550005.

[Bat87] Guy Battle. "A block spin construction of ondelettes. Part I: Lemarié functions." In: 110 (1987).

[BT09] Amir Beck and Marc Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems." In: *SIAM journal on imaging sciences* 2.1 (2009), pp. 183–202.

[Ben09] Yoshua Bengio. "Learning Deep Architectures for AI." In: *Found. Trends Mach. Learn.* 2.1 (2009), pp. 1–127.

[Ber94] Jan Beran. *Statistics for long-memory processes*. Vol. 61. Chapman & Hall/CRC Monographs on Statistics and Applied Probability. CRC press, 1994.

[BCR91] G. Beylkin, R. Coifman, and V. Rokhlin. "Fast wavelet transforms and numerical algorithms I." In: *Communications on Pure and Applied Mathematics* 44.2 (1991), pp. 141–183.

[BB16] M. Blaauw and J. Bonada. "Modeling and Transforming Speech Using Variational Autoencoders." In: *Interspeech*. 2016, pp. 1770–1774.

[BP09] Jean-Philippe Bouchaud and Marc Potters. *Theory of Financial Risk and Derivative Pricing: From Statistical Physics to Risk Management*. 2nd ed. Cambridge University Press, 2009.

[BD91] Peter J Brockwell and Richard A Davis. *Time Series: Theory and Methods*. Berlin, Heidelberg: Springer-Verlag, 1991.

[BDE09] Alfred M Bruckstein, David L Donoho, and Michael Elad. "From sparse solutions of systems of equations to sparse modeling of signals and images." In: *SIAM review* 51.1 (2009), pp. 34–81.

[Bru13] Joan Bruna. "Scattering Representations for Recognition. (Representations en Scattering pour la Reconaissance)." PhD thesis. École Polytechnique, Palaiseau, France, 2013.

[BM13a] Joan Bruna and Stéphane Mallat. "Audio Texture Synthesis with Scattering Moments." In: *CoRR* abs/1311.0407 (2013).

[BM13b] Joan Bruna and Stéphane Mallat. "Invariant Scattering Convolution Networks." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1872–1886.

[Bru+15] Joan Bruna, Stéphane Mallat, Emmanuel Bacry, and Jean-François Muzy. "Intermittent process analysis with scattering moments." In: *Annals of Statistics* 43.1 (2015), p. 323.

[Bur88] Peter J. Burt. "Smart sensing within a pyramid vision machine." In: *Proceedings of the IEEE* 76.8 (1988), pp. 1006–1015.

[Cel11] Carmine-Emanuele Cella. "Towards a symbolic approach to sound analysis." PhD thesis. Università di Bologna, 2011.

[CDS98] S. Chen, D. Donoho, and M. Saunders. "Atomic Decomposition by Basis Pursuit." In: *SIAM Journal on Scientific Computing* 20.1 (1998), pp. 33–61.

[CRS05] Taishih Chi, Powen Ru, and Shihab A Shamma. "Multiresolution spectrotemporal analysis of complex sounds." In: *The Journal of the Acoustical Society of America* 118.2 (2005), pp. 887–906.

[Cho+18] J. Chorowski, R.J. Weiss, R. A. Saurous, and S. Bengio. "On Using Backpropagation for Speech Texture Generation and Voice Conversion." In: *International Conference on Audio and Speech Processing (ICASSP)*. 2018.

[Chu+15] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. "A recurrent latent variable model for sequential data." In: *Advances in Neural Information Processing Systems*. 2015, pp. 2980–2988.

[CM91] Ronald Coifman and Yves Meyer. "Remarques sur l'analyse de Fourier à fenêtre." In: I (1991), pp. 259–261.

[CP11] Patrick L. Combettes and Jean-Christophe Pesquet. "Proximal Splitting Methods in Signal Processing." In: *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Ed. by Heinz H. Bauschke, Regina S. Burachik, Patrick L. Combettes, Veit Elser, D. Russell Luke, and Henry Wolkowicz. New York, NY: Springer New York, 2011, pp. 185–212.

[Das08] A. DasGupta. *Asymptotic Theory of Statistics and Probability*. Springer, 2008.

[Dau88] Ingrid Daubechies. "Orthonormal bases of compactly supported wavelets." In: *Communications on Pure and Applied Mathematics* 41.7 (1988), pp. 909–996.

[DDDM04] Ingrid Daubechies, Michel Defrise, and Christine De Mol. "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint." In: *Communications on Pure and Applied Mathematics* 57.11 (2004), pp. 1413–1457.

[DBC11] Guillaume Desjardins, Yoshua Bengio, and Aaron C Courville. "On Tracking The Partition Function." In: *Advances in Neural Information Processing Systems*. 2011, pp. 2501–2509.

[DF81] Persi Diaconis and David Freedman. “On the statistics of vision: The Julesz conjecture.” In: *Journal of Mathematical Psychology* 24.2 (1981), pp. 112 –138.

[DMP18] Chris Donahue, Julian McAuley, and Miller Puckette. “Synthesizing Audio with Generative Adversarial Networks.” In: *arXiv preprint arXiv:1802.04208* (2018).

[Don06] David Donoho. “For most large underdetermined systems of equations, the minimal l1-norm near-solution approximates the sparsest near-solution.” In: 59 (2006), pp. 907–934.

[DOT03] Paul Doukhan, George Oppenheim, and Murad S. Taqqu. *Theory and Applications of Long-Range Dependence | Paul Doukhan | Springer*. Birkhäuser Boston, 2003.

[ES16] Ronen Eldan and Ohad Shamir. “The power of depth for feedforward neural networks.” In: *Conference on Learning Theory*. 2016, pp. 907–940.

[Eng+17] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan. “Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders.” In: *Proceedings of the 34th International Conference on Machine Learning*. 2017, pp. 1068–1077.

[Fak15] Mohamed Waleed Fakhr. “Online nonstationary time series prediction using sparse coding with dictionary update.” In: *Information and Communication Technology Research (ICTRC), 2015 International Conference on*. IEEE. 2015, pp. 112–115.

[Fev+08] C. Fevotte, B. Torresani, L. Daudet, and S. J. Godsill. “Sparse Linear Regression With Structured Priors and Application to Denoising of Musical Audio.” In: *IEEE Transactions on Audio, Speech, and Language Processing* 16.1 (2008), pp. 174–185.

[FDGM86] W.M. Fisher, G.R. Doddington, and K.M. Goudie-Marshall. “The DARPA speech recognition research database: specifications and status.” In: *Proc. DARPA Workshop on Speech Recognition*. 1986, pp. 93–99.

[Fla92] P. Flandrin. “Wavelet analysis and synthesis of fractional Brownian motion.” In: *IEEE Transactions on Information Theory* 38.2 (1992), pp. 910–917.

[FT] Robert Fox and Murad S. Taqqu. “Central Limit Theorems for Quadratic Forms in Random Variables Having Long-Range Dependence.” In: *Probability Theory and Related Fields* 74 (), pp. 213–240.

[Fuc04] J. J. Fuchs. “On sparse representations in arbitrary redundant bases.” In: *IEEE Transactions on Information Theory* 50.6 (2004), pp. 1341–1344.

[Gab46] Dennis Gabor. *Theory of communication*. Institution of Electrical Engineering, 1946.

[GEB15] L. Gatys, A. S. Ecker, and M. Bethge. “Texture synthesis using convolutional neural networks.” In: *Advances in Neural Information Processing Systems*. 2015, pp. 262–270.

[GEB16] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. “Image Style Transfer Using Convolutional Neural Networks.” In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2414–2423.

[GBB11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. “Deep Sparse Rectifier Neural Networks.” In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, 2011, pp. 315–323.

[GME11] Ben Gold, Nelson Morgan, and Dan Ellis. *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. 2nd. New York, NY, USA: Wiley-Interscience, 2011.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[Goo+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative Adversarial Nets.” In: 2014, pp. 2672–2680.

[Gra+17] Timothy Graves, Robert Gramacy, Nicholas Watkins, and Christian Franzke. “A Brief History of Long Memory: Hurst, Mandelbrot and the Road to ARFIMA, 1951–1980.” In: *Entropy* 19.9 (2017).

[GL10] Karol Gregor and Yann LeCun. “Learning Fast Approximations of Sparse Coding.” In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. Omnipress, 2010, pp. 399–406.

[Gre+07] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. “A kernel method for the two-sample-problem.” In: *Advances in Neural Information Processing Systems*. 2007, pp. 513–520.

[GL84] Daniel Griffin and Jae Lim. "Signal estimation from modified short-time Fourier transform." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 32.2 (1984), pp. 236–243.

[Gri+17] E. Grinstein, N. Duong, A. Ozerov, and P. Pérez. "Audio style transfer." In: *HAL preprint hal-01626389* (2017).

[Gro+07] Roger Grosse, Rajat Raina, Helen Kwong, and Andrew Ng. "Shift-invariant Sparse Coding for Audio Classification." In: *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*. UAI'07. 2007, pp. 149–158.

[Gut03] Alan H Guth. "Time since the beginning." In: *arXiv preprint astro-ph/0301199* (2003).

[Här+04] Wolfgang Härdle, Axel Werwatz, Marlene Müller, and Stefan Sperlich. *Nonparametric and Semiparametric Models*. Berlin, Heidelberg: Springer-Verlag, 2004.

[HTF09] Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2009.

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 2016, pp. 770–778.

[HFA18] Ahmed Helmi, Mohamed W. Fakhr, and Amir F. Atiya. "Multi-step ahead time series forecasting via sparse coding and dictionary based techniques." In: *Applied Soft Computing* 69 (2018), pp. 464 –474.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory." In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators." In: *Neural networks* 2.5 (1989), pp. 359–366.

[Hos96] Jonathan R. M. Hosking. "Asymptotic distributions of the sample mean, autocovariances, and autocorrelations of long-memory time series." In: *Journal of Econometrics* 73.1 (1996), pp. 261–284.

[HZG17] Wei-Ning Hsu, Yu Zhang, and James Glass. "Learning Latent Representations for Speech Generation and Transformation." In: *Proc. Interspeech 2017*. 2017, pp. 1273–1277.

[Hur51] Harold E. Hurst. "Long-Term Storage Capacity of Reservoirs." In: *Transactions of the American Society of Civil Engineers* 116.1 (1951), pp. 770–799.

[IS15] S. Ioffe and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." In: *International Conference on Machine Learning*. 2015, pp. 448–456.

[JPH07] G R Jafari, P Pedram, and L Hedayatifar. "Long-range correlation and multifractality in Bach's Inventions pitches." In: *Journal of Statistical Mechanics: Theory and Experiment* 2007.04 (2007).

[Jul62] Bela Julesz. "Visual pattern discrimination." In: *IRE transactions on Information Theory* 8.2 (1962), pp. 84–92.

[KS80] James F. Kaiser and Ronald W. Schafer. "On the use of the I lt;inf gt;0 lt;/inf gt;-sinh window for spectrum analysis." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.1 (1980), pp. 105–107.

[Kal+16] Nal Kalchbrenner, Aaron van den Öord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. "Video pixel networks." In: *arXiv preprint arXiv:1610.00527* (2016).

[KB14] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization." In: *arXiv preprint arXiv:1412.6980* (2014).

[KW14] D. P. Kingma and M. Welling. "Auto-encoding variational bayes." In: *International Conference on Learning Representations*. 2014.

[Koe+13] Stefan Koelsch, Martin Rohrmeier, Renzo Torrecuso, and Sebastian Jentschke. "Processing of hierarchical syntactic structure in music." In: *Proceedings of the National Academy of Sciences* (2013).

[Kos+18] Ilya Kostrikov, Zhongshi Jiang, Daniele Panozzo, Denis Zorin, and Joan Bruna. "Surface Networks." In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. 2018.

[KT06] Matthieu Kowalski and Bruno Torrésani. "A family of random waveform models for audio coding." In: *ICASSP 2006*. Toulouse, France, 2006, p. 11024.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: *Advances in Neural Information Processing Systems*. 2012, pp. 1097–1105.

[KS12] Reimer Kühn and Peter Sollich. "Spectra of empirical auto-covariance matrices." In: *EPL (Europhysics Letters)* 99.2 (2012), p. 20008.

[Lam+17] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc-Aurelio Ranzato. "Fader Networks:Manipulating Images by Sliding Attributes." In: *Advances in Neural Information Processing Systems*. 2017, pp. 5967–5976.

[LM11] Hugo Larochelle and Iain Murray. "The neural autoregressive distribution estimator." In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 2011, pp. 29–37.

[LB98] Yann LeCun and Yoshua Bengio. "The Handbook of Brain Theory and Neural Networks." In: MIT Press, 1998. Chap. Convolutional Networks for Images, Speech, and Time Series, pp. 255–258.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." In: *nature* 521.7553 (2015), p. 436.

[Len03] Peter Lennie. "The cost of cortical computation." In: *Current biology* 13.6 (2003), pp. 493–497.

[LJ83] Fred Lerdahl and Ray Jackendoff. "An overview of hierarchical structure in music." In: *Music Perception: An Interdisciplinary Journal* 1.2 (1983), pp. 229–252.

[Li+17] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Poczos. "MMD GAN: Towards Deeper Understanding of Moment Matching Network." In: *Advances in Neural Information Processing Systems*. 2017, pp. 2203–2213.

[LSZ15] Yujia Li, Kevin Swersky, and Rich Zemel. "Generative Moment Matching Networks." In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. PMLR, 2015, pp. 1718–1727.

[Lin91] J. Lin. "Divergence measures based on the Shannon entropy." In: *IEEE Transactions on Information Theory* 37.1 (1991), pp. 145–151.

[Lin76] Seppo Linnainmaa. "Taylor expansion of the accumulated rounding error." In: *BIT Numerical Mathematics* 16.2 (1976), pp. 146–160.

[Los17] Vincent Lostanlen. "Convolutional operators in the time-frequency domain." PhD thesis. PSL Research University, 2017.

[MG77] MC Mackey and L Glass. "Oscillation and chaos in physiological control systems." In: *Science* 197.4300 (1977), p. 287.

[MBP12] Julien Mairal, Francis Bach, and Jean Ponce. "Task-driven dictionary learning." In: *IEEE transactions on pattern analysis and machine intelligence* 34.4 (2012), pp. 791–804.

[MZ93] S. G. Mallat and Zhifeng Zhang. "Matching pursuits with time-frequency dictionaries." In: *IEEE Transactions on Signal Processing* 41.12 (1993), pp. 3397–3415.

[Mal12] S. Mallat. "Group invariant scattering." In: *Communications on Pure and Applied Mathematics* 65.10 (2012), pp. 1331–1398.

[Mal08] Stephane Mallat. *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.

[Mal03] Stéphane Mallat. "Foveal detection and approximation for singularities." In: *Applied and Computational Harmonic Analysis* 14.2 (2003), pp. 133–180.

[MP43] Warren S. McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[MS11] Josh H McDermott and Eero P Simoncelli. "Sound texture perception via statistics of the auditory periphery: evidence from sound synthesis." In: *Neuron* 71.5 (2011), pp. 926–940.

[Meh+17] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio. "SampleRNN: An unconditional end-to-end neural audio generation model." In: *International Conference on Learning Representations*. 2017.

[MG13] Nishant Mehta and Alexander Gray. "Sparsity-based generalization bounds for predictive sparse coding." In: *International Conference on Machine Learning*. 2013, pp. 36–44.

[MP69] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA, USA: MIT Press, 1969.

[MT05] Stéphane Molla and Bruno Torrésani. "A hybrid scheme for encoding audio signal using hidden Markov models of waveforms." In: *Applied and Computational Harmonic Analysis* 18.2 (2005), pp. 137–166.

[MB17] Thomas Moreau and Joan Bruna. "Understanding Trainable Sparse Coding Via Matrix Factorization." In: *International Conference on Learning Representations*. 2017.

[Nat95] B. K. Natarajan. “Sparse Approximate Solutions to Linear Systems.” In: *SIAM J. Comput.* 24.2 (1995), pp. 227–234.

[O’S88] D. O’Shaughnessy. “Linear predictive coding.” In: *IEEE Potentials* 7.1 (1988), pp. 29–32.

[OF96] Bruno A. Olshausen and David Field. “Emergence of simple-cell receptive field properties by learning a sparse code for natural images.” In: 381 (1996), pp. 607–9.

[OKK16] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. “Pixel Recurrent Neural Networks.” In: *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 2016, pp. 1747–1756.

[Oor+16] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. “Wavenet: A generative model for raw audio.” In: *arXiv preprint arXiv:1609.03499* (2016).

[Oor+17] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, et al. “Parallel WaveNet: Fast High-Fidelity Speech Synthesis.” In: *arXiv preprint arXiv:1711.10433* (2017).

[OBZ17] E. Oyallon, E. Belilovsky, and S. Zagoruyko. “Scaling the Scattering Transform: Deep Hybrid Networks.” In: *Proc. of ICCV*. 2017.

[Oya17] Edouard Oyallon. “Analyzing and Introducing Structures in Deep Convolutional Neural Networks.” PhD thesis. Ecole normale supérieure - PSL Research University, 2017.

[PRE17] Vardan Papyan, Yaniv Romano, and Michael Elad. “Convolutional Neural Networks Analyzed via Convolutional Sparse Coding.” In: *Journal of Machine Learning Research* 18.83 (2017), pp. 1–52.

[PRK93] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. “Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition.” In: *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*. Vol. 1. 1993, pp. 40–44.

[PMK10] Jouni Paulus, Meinard Müller, and Anssi Klapuri. “State of the Art Report: Audio-Based Music Structure Analysis.” In: *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*. 2010, pp. 625–636.

[Per+18] Antoine Perquin, Gwénolé Lecorvé, Damien Lolive, and Laurent Amsaleg. "Phone-Level Embeddings for Unit Selection Speech Synthesis." In: *6th International Conference on Statistical Language and Speech Processing (SLSP), 2018*. 6th International Conference on Statistical Language and Speech Processing (SLSP), 2018. 2018.

[Pin15] Allan Pinkus. *Ridge Functions*. Cambridge Tracts in Mathematics. Cambridge University Press, 2015.

[Plu+10] Mark D. Plumbley, Thomas Blumensath, Laurent Daudet, Rémi Gribonval, and Mike E. Davies. "Sparse Representations in Audio and Music: from Coding to Source Separation." In: *Proceedings of the IEEE.* 98.6 (2010), pp. 995–1005.

[PS00] Javier Portilla and Eero P. Simoncelli. "A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients." In: *International Journal of Computer Vision* 40.1 (2000), pp. 49–70.

[Pow06] James L. Powell. "Notes on Nonparametric Density Estimation." 2006.

[Pur+01] Dale Purves, George J Augustine, David Fitzpatrick, Lawrence C Katz, Anthony-Samuel Lamantia, James O McNamara, and S Mark Williams. *Neuroscience*. 2nd. Sinauer Associates, 2001.

[RS78] Lawrence R Rabiner and Ronald W Schafer. *Digital processing of speech signals*. Prentice Hall, 1978.

[RMC16] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks." In: *International Conference on Learning Representations*. 2016.

[RSM05] O. Renaud, J.-L. Starck, and F. Murtagh. "Wavelet-based combined signal filtering and prediction." In: *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 35.6 (2005), pp. 1241–1251.

[Ron+16] Srikanth Ronanki, Oliver Watts, Simon King, and Gustav Eje Henter. "Median-Based Generation of Synthetic Speech Durations using a Non-Parametric Approach." In: *Proc. IEEE Workshop on Spoken Language Technology (SLT)*. 2016.

[Ros58] Frank Rosenblatt. "The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain." In: *Psychological Review* (1958), pp. 65–386.

[RHW86] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors." In: *Nature* 323.6088 (1986), p. 533.

[SIL49] SILSO World Data Center. "The International Sunspot Number." In: *International Sunspot Number Monthly Bulletin and online catalogue* (1749-2015).

[Sam07] Gennady Samorodnitsky. "Long Range Dependence." In: *Foundations and Trends in Stochastic Systems* 1.3 (2007), pp. 163–257.

[She+17] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, et al. "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions." In: *arXiv preprint arXiv:1712.05884* (2017).

[Tib96] Robert Tibshirani. "Regression shrinkage and selection via the lasso." In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1996), pp. 267–288.

[TA77] Andrey N. Tikhonov and Vasiliy Y. Arsenin. *Solutions of ill-posed problems*. Translated from the Russian, Preface by translation editor Fritz John, Scripta Series in Mathematics. Washington, D.C.: John Wiley & Sons, New York: V. H. Winston & Sons, 1977.

[Tol+18] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schölkopf. "Wasserstein Auto-Encoders." In: *International Conference on Learning Representations*. 2018.

[Tro06] J. A. Tropp. "Just relax: convex programming methods for identifying sparse signals in noise." In: *IEEE Transactions on Information Theory* 52.3 (2006), pp. 1030–1051.

[VAS14] A. Venkitaraman, A. Adiga, and C. S. Seelamantula. "Auditory-motivated Gammatone wavelet transform." In: *Signal Processing* 94 (2014), pp. 608–619.

[Vin07] Emmanuel Vincent. "Complex nonconvex lp norm minimization for underdetermined source separation." In: *7th Int. Conf. on Independent Component Analysis and Blind Source Separation (ICA)*. London, United Kingdom, 2007, pp. 430–437.

[Wal15] Irène Waldspurger. "Wavelet transform modulus : phase retrieval and scattering." PhD thesis. École Normale Supérieure, Paris, France, 2015.

[WC94] Andrew T. A. Wood and Grace Chan. "Simulation of Stationary Gaussian Processes in [0,1]d." In: *Journal of Computational and Graphical Statistics* 3.4 (1994), pp. 409–432.

[Xin+15] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. “Convolutional LSTM network: A machine learning approach for precipitation nowcasting.” In: *Advances in Neural Information Processing Systems*. 2015, pp. 802–810.

[Yam12] Junichi Yamagishi. *English multi-speaker corpus for CSTR voice cloning toolkit*. 2012. URL: http://homepages.inf.ed.ac.uk/jyamagis/page3/page58/page58.html.

[YMH17] Chun-Nam Yu, Piotr Mirowski, and Tin Kam Ho. “A Sparse Coding Approach to Household Electricity Demand Forecasting in Smart Grids.” In: *IEEE Transactions on Smart Grid* 8.2 (2017), pp. 738–748.

[YK16] Fisher Yu and Vladlen Koltun. “Multi-Scale Context Aggregation by Dilated Convolutions.” In: *International Conference on Learning Representations*. 2016.

[Zha+17] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. “Understanding deep learning requires rethinking generalization.” In: *International Conference on Learning Representations*. 2017.

[ZWM98] Song Chun Zhu, Yingnian Wu, and David Mumford. “Filters, Random Fields and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling.” In: *International Journal of Computer Vision* 27.2 (1998), pp. 107–126.

Résumé

Cette thèse s'intéresse à la modélisation non-supervisée de séries temporelles univariées. Nous abordons tout d'abord le problème de prédiction linéaire des valeurs futures de séries temporelles gaussiennes sous hypothèse de longues dépendances, qui nécessitent de tenir compte d'un large passé. Nous introduisons une famille d'ondelettes fovéales et causales qui projettent les valeurs passées sur un sous-espace adapté au problème, réduisant ainsi la variance des estimateurs associés. Dans un deuxième temps, nous cherchons sous quelles conditions les prédicteurs non-linéaires sont plus performants que les méthodes linéaires. Les séries temporelles admettant une représentation parcimonieuse en temps-fréquence, comme celles issues de l'audio, réunissent ces conditions, et nous proposons un algorithme de prédiction utilisant une telle représentation. Le dernier problème que nous étudions est la synthèse de signaux audios. Nous proposons une nouvelle méthode de génération reposant sur un réseau de neurones convolutionnel profond, avec une architecture encodeur-décodeur, qui permet de synthétiser de nouveaux signaux réalistes. Contrairement à l'état de l'art, nous exploitons explicitement les propriétés temps-fréquence des sons pour définir un encodeur avec la transformée en scattering, tandis que le décodeur est entraîné pour résoudre un problème inverse dans une métrique adaptée.

Mots Clés

Séries temporelles, Scattering, Réseaux de neurones profonds, Audio, Prédiction, Synthèse

Abstract

This dissertation studies unsupervised time-series modeling. We first focus on the problem of linearly predicting future values of a time-series under the assumption of long-range dependencies, which requires to take into account a past of large duration. We introduce a family of causal and foveal wavelets which project past values on a subspace adapted to the problem, thereby reducing the variance of the associated estimators. We then investigate under which conditions non-linear predictors exhibit better performances than linear ones. Time-series which admit a sparse time-frequency representation, such as audio ones, satisfy these requirements, and we propose a prediction algorithm using such a representation. The last problem we tackle is audio time-series synthesis. We propose a new generation method relying on a deep convolutional neural network, with an encoder-decoder architecture, which allows to synthesize new realistic signals. Contrary to state-of-the-art methods, we explicitly use time-frequency properties of sounds to define an encoder with the scattering transform, while the decoder is trained to solve an inverse problem in an adapted metric.

Keywords

Time-Series, Scattering, Deep learning, Audio, Prediction, Synthesis