THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (Toulouse INP)

Discipline ou spécialité :

Informatique et Télécommunication

Présentée et soutenue par :

M. PIERRE-HUGO VIAL

le mardi 29 novembre 2022

Titre :

Reconstruction de phase et de signaux audio avec des fonctions de coût non-quadratiques

---

Ecole doctorale :

Mathématiques, Informatique, Télécommunications de Toulouse (MITT)

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (IRIT)

Directeur(s) de Thèse :

M. CÉDRIC FÉVOTTE

M. THOMAS OBERLIN

Rapporteurs :

M. MATTHIEU KOWALSKI, UNIVERSITE PARIS-SUD

MME CAROLINE CHAUX-MOULIN, CNRS MARSEILLE

Membre(s) du jury :

M. RÉMI GRIBONVAL, INRIA, Président

M. CÉDRIC FÉVOTTE, CNRS TOULOUSE, Membre

MME IRÈNE WALDSPURGER, CNRS PARIS, Membre

M. PAUL MAGRON, INRIA, Membre

M. THOMAS OBERLIN, ISAE-SUPAERO, Membre

PHASE RETRIEVAL AND AUDIO SIGNAL RECONSTRUCTION WITH NON-QUADRATIC COST FUNCTIONS

PIERRE-HUGO VIAL

Pierre-Hugo Vial: *Phase retrieval and audio signal reconstruction with non-quadratic cost functions,* Reconstruction de phase et de signaux audio avec des fonctions de coût non-quadratiques, 2022.

REMERCIEMENTS

Je souhaite profiter de ces quelques paragraphes pour adresser mes remerciements les plus chaleureux aux nombreuses personnes ayant partagé avec moi ces années de thèse. J'espère n'oublier personne, ou a minima me faire pardonner si tel est le cas!

Mes premiers remerciements vont bien évidemment à Cédric, Thomas et Paul. J'ai pu trouver à vos côtés, dès le début de mon stage, un encadrement de grande qualité, des conseils avisés et une habileté à sans cesse stimuler ma curiosité. Plus personnellement, je souhaite également vous remercier pour la richesse de nos discussions et votre attention indéfectible tout au long de cette thèse.

Un grand merci également aux membres de mon jury de soutenance : Caroline Chaux et Matthieu Kowalski, pour avoir rapporté ma thèse, ainsi que Rémi Gribonval et Irène Walspurger pour l'avoir examinée. Je vous exprime toute ma reconnaissance pour vos précieuses remarques et pour avoir accepté d'évaluer mes travaux.

Je souhaite exprimer maintenant mon amitié à tous·tes les membres de l'équipe SC : merci à Marie, Nicolas, Thomas, Cédric, Emmanuel, Henrique, Elsa, Sixin, Paul, Arthur, Cassio, Olivier, Louis, Adrien, Etienne, Baha, Maxime, Camille, Claire, Vinicius, Asma et Florentin. Merci pour votre accueil, votre implication constante dans l'animation de la vie de l'équipe et pour tous ces beaux moments partagés.

Je souhaite maintenant remercier tous·tes mes ami·e·s pour leur présence tout au long de cette thèse. Un grand merci à tous·tes celles et ceux rencontré·e·s aux Siestes Electroniques, en particulier Beliz, Bertrand, Jonathan, Hoel, Kaoutar, Kristell, Laure, Louis, Mathieu, Narjess, Zoé; à mes comparses du Comité des Fêtes Alice, Etienne, Hugo, Juliette, Tom et Yann; également à Camille, Fanny, Johary, Lorène, Lucy, Thien-Anh et Valentin.

Merci à mon frère Tom et mes parents pour leur soutien et leurs encouragements continus, du premier jour au pot de thèse. Enfin, à Jon pour tous les moments passés ensemble et ceux à venir.

ABSTRACT

Audio signal reconstruction consists in recovering sound signals from incomplete or degraded representations. This problem can be cast as an inverse problem. Such problems are frequently tackled with the help of optimization or machine learning strategies. In this thesis, we propose to change the cost function in inverse problems related to audio signal reconstruction. We mainly address the phase retrieval problem, which is common when manipulating audio spectrograms.

A first line of work tackles the optimization of non-quadratic cost functions for phase retrieval. We study this problem in two contexts: audio signal reconstruction from a single spectrogram and source separation. We introduce a novel formulation of the problem with Bregman divergences, as well as algorithms for its resolution.

A second line of work proposes to learn the cost function from a given dataset. This is done under the framework of unfolded neural networks, which are derived from iterative algorithms. We introduce a neural network based on the unfolding of the Alternating Direction Method of Multipliers, that includes learnable activation functions. We expose the relation between the learning of its parameters and the learning of the cost function for phase retrieval.

We conduct numerical experiments for each of the proposed methods to evaluate their performance and their potential with audio signal reconstruction.

RÉSUMÉ

La reconstruction de signaux audio consiste à estimer des signaux sonores à partir de représentations incomplètes ou dégradées. Ce problème peut être formulé comme un problème inverse. Ces derniers sont fréquemment traités à l'aide de stratégies d'optimisation ou d'apprentissage automatique. Dans cette thèse, on propose de modifier la fonction de coût dans les problèmes inverses liés à la reconstruction de signaux audio. On considère principalement le problème de reconstruction de phase, un problème fréquent lors de la manipulation de spectrogrammes audio.

Un premier axe de ces travaux étudie l'optimisation de fonctions de coût non-quadratiques pour la reconstruction de phase. Ce problème est étudié dans deux contextes: la reconstruction de signaux audio à partir d'un spectrogramme et la séparation de sources. Nous proposons une nouvelle formulation du problème à l'aide des divergences de Bregman, ainsi que des algorithmes pour leur résolution.

Un second axe considère l'apprentissage de la fonction de coût à partir d'un jeu de données. On utilise le cadre des réseaux de neurones dépliés, obtenus à partir d'algorithmes itératifs. On propose un réseau de neurones construit via le dépliement de l'algorithme des directions alternées et incluant des fonctions d'activations paramétrées. On explicite la relation entre l'apprentissage de ses paramètres et de la fonction de coût pour la reconstruction de phase.

Enfin, on conduit un travail expérimental pour chaque méthode exposée dans cette thèse afin d'évaluer leur performance et leur potentiel pour la reconstruction de signaux audio.

CONTENTS

Abstract vii
Résumé viii
List of Figures xi
Acronyms xiii
Notations xvi
1 INTRODUCTION 1
1.1 General context and motivation 1
1.2 Outline of the manuscript 2
1.3 Publications 3

I BACKGROUND AND RELATED WORK 5
2 BACKGROUND 7
2.1 Audio signal processing 7
2.1.1 Time-frequency analysis with the short-time Fourier transform 7
2.1.2 Objective evaluation of audio quality 13
2.2 Optimization 15
2.2.1 Wirtinger calculus and gradient methods 15
2.2.2 Proximity operators and proximal methods . . 18
2.2.3 Bregman divergences 21
2.3 Deep learning 23
2.3.1 Neural networks 23
2.3.2 Unfolding iterative algorithms 24
Appendices 29
2.A Proximity operator of the left IS divergence 29
3 RELATED WORK 31
3.1 The phase retrieval problem 31
3.1.1 Problem formulation 31
3.1.2 Algorithms 32
3.2 Phase retrieval in audio 36
3.2.1 Context and applications 36
3.2.2 Specific algorithms 36
3.2.3 Phase retrieval for audio source separation . . . 40
3.3 Phase retrieval with deep learning 41

II PHASE RETRIEVAL WITH BREGMAN DIVERGENCES 43
4 PHASE RETRIEVAL FROM SINGLE SPECTROGRAM 45
4.1 Introduction 45
4.2 Phase retrieval with Bregman divergences 46
4.2.1 Problem setting 46
4.2.2 Accelerated gradient descent 46
4.2.3 ADMM algorithm 48

4.2.4 Implementation details . . . 50
4.3 Numerical experiments . . . 51
4.3.1 Experimental setup . . . 52
4.3.2 Phase retrieval from exact spectrograms . . . 54
4.3.3 Phase retrieval from modified spectrograms . . . 55
4.4 Strategies for the choice of the gradient step size . . . 60
4.4.1 Experimental setup . . . 60
4.4.2 Results . . . 61
4.5 Conclusion . . . 63
Appendices 67
4.A Algorithms derivations for real-valued signals . . . 67
4.B Regularized gradient expression . . . 67
4.C Nonnegativity constraint on $\mathbf{u}$ in ADMM . . . 68
5 PHASE RETRIEVAL FOR AUDIO SOURCE SEPARATION 71
5.1 Introduction . . . 71
5.2 Phase retrieval with Bregman divergences and mixing constraint . . . 72
5.2.1 Problem formulation . . . 72
5.2.2 Projected gradient descent . . . 72
5.2.3 Derivation of the gradient . . . 73
5.2.4 Summary of the algorithm . . . 74
5.3 Numerical experiments . . . 74
5.3.1 Experimental setup . . . 74
5.3.2 Influence of the step size . . . 75
5.3.3 Comparison to other methods . . . 76
5.4 Conclusion . . . 77

III PHASE RETRIEVAL WITH UNFOLDED ALGORITHMS 79
6 LEARNING PROXIMITY OPERATORS FOR PHASE RETRIEVAL 81
6.1 Introduction . . . 81
6.2 Learning proximity operators in unfolded ADMM . . . 82
6.2.1 Proposed general unfolded architecture . . . 82
6.2.2 Proposed parameterization with APL units . . . 83
6.3 Interpretability and characterization . . . 84
6.3.1 Discussion about interpretability . . . 84
6.3.2 Characterization of $\mathsf{F}(\mathbf{y}, \mathbf{r})$ as a proximity operator 85
6.4 Numerical experiments . . . 86
6.4.1 Experimental setup . . . 86
6.4.2 Results . . . 87
6.5 Conclusion . . . 88
7 CONCLUSION AND PERSPECTIVES 91
7.1 Summary . . . 91
7.2 Perspectives and future work . . . 91
A RÉSUMÉ DÉTAILLÉ DE LA THÈSE 93

BIBLIOGRAPHY 99

LIST OF FIGURES

Figure 1 Waveform and time-frequency representations of the 12 first seconds of Bernard Parmegiani's *De Natura Sonorum - Incidences/battements* . . . 9

Figure 2 Bregman divergences plots with fixed first (left) and second (right) arguments. $\beta = 0.5$ with the beta-divergence. 22

Figure 1 Performance of PR from exact spectrograms for the "speech" corpus, measured with the SC (top) and STOI (bottom). Higher values correspond to a better performance. Turquoise, orange and yellow respectively denote gradient descent algorithms, ADMM algorithms and GLA-like algorithms. The boxes indicate the two middle quartiles among the ten excerpts, the middle bar is for the median, the dot for the mean, and the whiskers denote the extremal values. 56

Figure 2 Performance of PR from exact spectrograms for the "music" corpus measured with the SC. 57

Figure 3 STOI for PR from modified speech spectrograms at various input SNRs. 58

Figure 3 STOI for PR from modified speech spectrograms at various input SNRs. 59

Figure 4 Performance of PR from magnitude spectrograms ($d = 1$), measured with the SC (top) and STOI (bottom). The considered cost functions are from left to right : "left" beta-divergence, "left" Kullback-Leibler, Quadratic, "right" Kullback-Leibler, "right" beta-divergence. $\beta = 0.5$ with the beta-divergences. 63

Figure 5 Performance of PR from power spectrograms ($d = 2$), measured with the SC (top) and STOI (bottom). The considered cost functions are from left to right : "left" beta-divergence with $\beta = 0.5$, "left" Kullback-Leibler, Quadratic, "right" Kullback-Leibler, "right" beta-divergence with $\beta = 0.5$. 64

Figure 1 Average SDRi on the validation set obtained with the proposed algorithm at various iSNRs, when $d = 1$ (top) and $d = 2$ (bottom). For better readability, we set the SDRi at 0 when convergence issues occur as visually inspected, or when the SDRi is below 0, as this implies a decreasing performance over iterations, which is not desirable. 76
Figure 2 Average SDRi on the test set obtained with MISI and with the proposed algorithm (in different settings) at various iSNRs. 77
Figure 3 One layer of the proposed unfolded architecture. 82
Figure 4 Training loss (negative STOI) over epochs. Note that `pytorch-stoi` implementation does not exactly replicate the original metric and consequently yields values lower than -1. . . . 88
Figure 5 Performance on the test set. Each box-plot is made up of a central line indicating the median, box edges indicating the 1^{st} and 3^{rd} quartiles, whiskers indicating the extremal values, and circles representing the outliers. 89
Figure 6 Evaluation with STOI over test dataset with iterated model. The solid lines denote the mean STOI and the light colored areas the values between the first and the third quartile. 89
Figure 7 Learned metrics $f_{r,t}(y)$ with $r = 1$. The quadratic cost and Kullback-Leibler divergence $\mathcal{D}_{KL}(y \mid r)$ are also displayed for the sake of comparison. In the "tied" case, f_r in analogous to $\mathcal{D}_\psi(\cdot \mid r)$ involved in the PR optimization problem. For clarity, only 3 of the 15 trained layers $f_{r,t}$ are displayed for the "untied" case. 90

ACRONYMS

ABC/HR ABC test with Hidden Reference

ADMM Alternating Direction Method of Multipliers

APL Adaptive Piecewise Linear

BSSEval Blind Source Separation Evaluation

BB Barzilai–Borwein

BLSTM Bi-directional Long Short Term Memory

BT BackTracking

CNN Convolutional Neural Network

COLA Constant-Overlap-Add

CQT Constant-Q Transform

DEMAND Diverse Environments Multi-channel Acoustic Noise Database

DFT Discrete Fourier Transform

DNN Deep Neural Network

ER Error Reduction

FFT Fast Fourier Transform

FMA Free Music Archive (corpus)

GD Gradient Descent

GLA Griffin–Lim Algorithm

GLADMM Griffin–Lim like phase recovery via ADMM

GSA Gerchberg–Saxton Algorithm

iDFT inverse Discrete Fourier Transform

IS Itakura–Saito (divergence)

ISTA Iterative Shrinkage-Thresholding Algorithm

iSTFT inverse Short-Time Fourier Transform

ITU International Telecomunication Union

KL Kullback–Leibler (divergence)

LASSO Least Absolute Shrinkage and Selection Operator

LISTA Learned Iterative Shrinkage-Thresholding Algorithm

l.s.c. lower semi-continuous

MIR Music Information Retrieval

MISI Multiple Input Spectrogram Inversion

MOS Mean Opinion Score

MP3 Moving Picture experts group-1/2 audio layer III

MRI Magnetic Resonance Imaging

MUSHRA MUltiple Stimuli with Hidden Reference and Anchor

NMF Nonnegative Matrix Factorization

NP Non-deterministic in Polynomial-time

ODG Objective Difference Grade

PEAQ Perceptual Evaluation of Audio Quality

PEASS Perceptual Evaluation of Audio Source Separation

PESQ Perceptual Evaluation of Speech Quality

POLQA Perceptual Objective Listening Quality Assessment

PR Phase Retrieval

PReLU Parametric Rectified Linear Unit

ReLU Rectified Linear Unit

RTISI(-LA) Real-Time Iterative Spectrogram Inversion (with Look-Ahead)

SAR Signal-to-Artifact Ratio

SC Spectral Convergence

SDG Subjective Difference Grade

SDP SemiDefinite Program

SGD Stochastic Gradient Descent

SIR Signal-to-Interference Ratio

(SI-)SDR (Scale-Invariant-) Signal-to-Distortion Ratio

SNR Signal-to-Noise Ratio

SPSI Single Pass Spectrogram Inversion

STFT Short-Time Fourier Transform

STOI Short-Time Objective Intelligibility

TF Time-Frequency

TIMIT Texas Instruments/Massachussetts Institute of Technology (corpus)

UADMM Unfolded Alternating Direction Method of Multipliers

WF Wirtinger Flow

$\mathbf{A}$ (capital, bold font)	matrix, whose (m,n)-th entry is denoted $A(m,n)$ and n-th column (resp. m-th line) is denoted $A(\cdot,n)$ (resp. $A(m,\cdot)$).
$\mathbf{x}$ (lower case, bold font)	column vector, whose ℓ-th entry is denoted $x(\ell)$ or x_ℓ in compact notation.
z (regular)	scalar.
$\lvert\cdot\rvert$, $\angle(\cdot)$, $(\cdot)^*$	magnitude, complex angle, and complex conjugate, respectively.
^{T}, ^{H}	transpose and Hermitian transpose, respectively.
$\mathbb{R}$, $\mathbb{C}$, $\mathbb{N}$	sets of real numbers, complex numbers and natural integers, respectively.
$\mathfrak{R}$, $\mathfrak{I}$	real and imaginary part functions.
$\odot$, $(\cdot)^{\mathrm{d}}$, fraction bar	element-wise matrix or vector multiplication, exponentiation, and division, respectively.
$\langle\cdot\,;\,\cdot\rangle$	inner product.
$\lVert\cdot\rVert_p$	p-norm. With the Euclidean norm ($p=2$), the index is dismissed.
$\mathbf{I}_L$	identity matrix of size L.
$\mathrm{diag}(\mathbf{x})$	diagonal matrix which entries are the components of $\mathbf{x}$.
$\mathcal{P}_{\mathcal{S}}$	projection operator on the set $\mathcal{S}$.
$\chi_{\mathcal{S}}$	indicator function of the set $\mathcal{S}$ such that $\chi_{\mathcal{S}}(x) = \begin{cases} 1 & \text{if } x \in \mathcal{S}, \\ 0 & \text{else.} \end{cases}$

1 INTRODUCTION

1.1 GENERAL CONTEXT AND MOTIVATION

Notation and *recording* aim at making the sensible experience of sound reproducible. With musical notation, sound is visually represented with symbols in order to be re-performed by a musician. Even though a wide number of methods have existed through history and cultures, the Western notation system based on the equal temperament became predominant. It consists in arranging pitches in an equally divided frequency space and in time. Recording stores a signal to allow for its re-generation. Different technologies enabled this practice, initiated with Thomas Edison's Phonograph in the nineteenth century and followed by various supports including wax cylinders, vinyl discs and magnetic tape. Digital recording advented in the end of the twentieth century and consists in the storage of digital representations of sound produced through transduction, sampling and quantization.

Digital sound representations are usually processed using signal processing methods as they can be degraded or incomplete. Such imperfections usually occur through the recording process and include distortion (i. e. , nonlinear transformation of the signal), mixing with an undesired source (e. g. , noise or echoes), reverberation, presence of artifacts, reduction of the frequency bandwidth, or downsampling. They can also be processed with other aims than removing degradations. For example, remastering refines music recordings through equalization and dynamic processors to adapt their character to contemporary standards. After processing, reconstructed signals are usually associated with a superior listening experience: the perceived audio quality is improved while the comprehension is enriched.

As audio signal reconstruction can be interpreted as the recovery of a signal from a set of observations (i. e. , a degraded or incomplete sound representation), it falls within the class of *inverse problems*. Inverse problems inspired a wide range of works in the literature and can be defined as recovering the causal factors $\mathbf{x}^\star$ from a set of observations $\mathbf{r}$. The relation between causes and observations $\mathcal{F}$ is known as the *forward map*, and is such that:

$$\mathbf{r} = \mathcal{F}(\mathbf{x}^\star). \tag{1.1.1}$$

When $\mathcal{F}$ is known or estimated via a *forward model*, inverse problems may be solved via the optimization of a real-valued cost function $\mathcal{D}$,

measuring the error between the estimates and the observations. This writes:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \mathcal{D}(\mathcal{F}(\mathbf{x}), \mathbf{r}). \tag{1.1.2}$$

As the solutions to (1.1.2) may not be existent, unique or stable, the considered problem is said to be *ill-posed*. Common strategies to tackle ill-posed inverse problems include the modification of the cost function, the minimization of an additional *regularization* term or the restriction of the solution set. Such choices are realized with the help of *a priori knowledge* about the solutions. In the scope of audio signal reconstruction, such knowledge may come from listening-specific criteria. The problem should be modified such that the solutions lead to a satisfactory listening experience and that errors in terms of perceived quality and understanding are penalized.

In this thesis, we explore the idea of changing the cost function in inverse problems for audio signal reconstruction. We mainly consider the phase retrieval problem, a non-convex and non-linear inverse problem that arises when processing the most common time-frequency representation: the spectrogram. Phase retrieval occurs within different audio reconstruction tasks, including denoising and source separation, and is notoriously ill-posed as multiple signals can generate the same observations. We take interest in concepts and works from domains ranging from machine learning to audio signal processing to study formulations of the phase retrieval problem with alternative cost functions. Methods to tackle them are proposed and assessed via experimental work, for applications including denoising and audio source separation.

1.2 OUTLINE OF THE MANUSCRIPT

This dissertation is organized as follows. Chapter 2 presents the background of this thesis. Essential tools and concepts from audio signal processing, optimization and deep learning are introduced. Chapter 3 details the problems of interest of this thesis and the state-of-the-art approaches for their resolution. The phase retrieval problem is formulated and studied in the context of audio signals. The chapter ends with the presentation of related work in deep learning. Chapter 4 details the first contribution of this thesis. The phase retrieval problem is extended to alternative cost functions: we present a formulation of the problem with Bregman divergences and we derive two algorithms. Experimental results and a discussion on the choice of the parameters are presented. Chapter 5 extends the work presented in the previous chapter to audio source separation. The phase retrieval problem with Bregman divergences and a mixing constraint is introduced with a proposed extension to the Multiple Input Spectrogram Inversion algorithm. We conduct experimental work for this applica-

tion and present the results. Chapter 6 tackles the problem of learning the cost function for phase retrieval. The ADMM-based algorithm introduced in Chapter 4 is unfolded within a neural network and the proximity operators are replaced with trainable activation functions. Learning their parameters amounts to learning the cost function in the original problem. Experimental work then assesses the efficiency of the method. The final chapter draws concluding remarks and summarizes the results of this thesis. Perspectives for upcoming research are discussed subsequently.

1.3 PUBLICATIONS

Journals

1. Pierre-Hugo Vial, Paul Magron, Thomas Oberlin, and Cédric Févotte. "Phase Retrieval with Bregman Divergences and Application to Audio Signal Recovery." In: *IEEE Journal of Selected Topics in Signal Processing* 15.1 (2021), pp. 51–64

2. Pierre-Hugo Vial, Paul Magron, Thomas Oberlin, and Cédric Févotte. "Learning the Proximity Operator in Unfolded ADMM for Phase Retrieval." In: *IEEE Signal Processing Letters* 29 (2022), pp. 1619–1623

Conferences

1. Pierre-Hugo Vial, Paul Magron, Thomas Oberlin, and Cédric Févotte. "Phase Retrieval with Bregman Divergences: Application to Audio Signal Recovery." In: *Proc. International Traveling Workshop on Interactions between low-complexity data models and Sensing Techniques (iTWIST)*. 2020, pp. 1–2

2. Paul Magron, Pierre-Hugo Vial, Thomas Oberlin, and Cédric Févotte. "Phase Recovery with Bregman Divergences for Audio Source Separation." In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 516–520

Part I

BACKGROUND AND RELATED WORK

2

BACKGROUND

2.1 Audio signal processing 7
2.1.1 Time-frequency analysis with the short-time Fourier transform 7
2.1.2 Objective evaluation of audio quality 13
2.2 Optimization 15
2.2.1 Wirtinger calculus and gradient methods 15
2.2.2 Proximity operators and proximal methods . . 18
2.2.3 Bregman divergences 21
2.3 Deep learning 23
2.3.1 Neural networks 23
2.3.2 Unfolding iterative algorithms 24

In this chapter, we introduce the fundamental tools used in this thesis. Section 2.1 defines essential concepts of audio signal processing with an overview of the short-time Fourier transform and its properties followed by a discussion on audio quality evaluation.
Section 2.2 introduces elements from optimization and machine learning. We first present Wirtinger derivatives and gradient methods. Then, the proximity operator and its properties are introduced. This is followed by the definition of Bregman divergences. A brief introduction to neural networks and unfolded iterative algorithms is presented in Section 2.3.

2.1 AUDIO SIGNAL PROCESSING

2.1.1 *Time-frequency analysis with the short-time Fourier transform*

Representing audio signals

In audio signal processing, it is a common practice to process a sound signal into a representation in order to generate features. According to their properties, different representations can be considered to a given application. In most cases, it is desirable to have an invertible representation in order to reconstruct a sound signal after processing. Figure 1 displays a few usual audio representations.

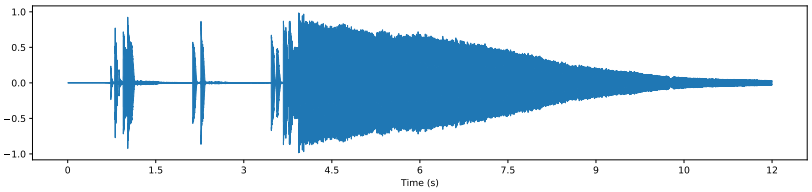
The most simple representation is the *waveform*, which is a sampled and quantized sound signal. The waveform collects amplitude measurements over time that correspond to voltage in the case of electric signals or pressure for an acoustic sound. The measurements can be arranged in an array $\mathbf{X} \in \mathbb{R}^{L \times C}$, where L denotes the time dimension and C is the number of channels. Most of the time, $C = 1$ or 2. In

these cases, signals are respectively refered to as *monophonic* or *stereophonic*. The number of channels can be greater than 2 in the case of *surround sound*, where multiple sound speakers are used to enhance sound spatialization. In this thesis, only mono waveforms will be considered. They will be denoted as time vectors $\mathbf{x} \in \mathbb{R}^L$.

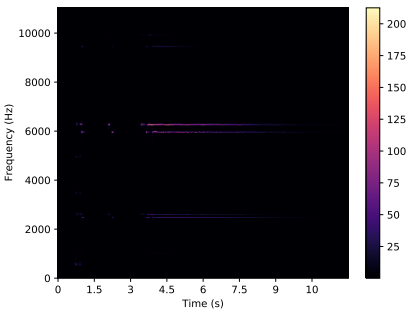
Several usual representations belong to the class of *time-frequency representations* [47]. Such representations display discrete Fourier transform (DFT) based features through time and are most of the time computed with the help of the fast Fourier transform algorithm (FFT). The short-time Fourier transform (STFT) is a commonly used operation in this context. The STFT produces a complex matrix, whose modulus is referred to as *magnitude* and can be interpreted as the time-frequency distribution of the signal energy. The STFT argument is known as *phase* and has a less obvious interpretation. However, it embodies critical information for perception [111, 112] and waveform reconstruction [53, 106]. It will be a prominent point of interest in this thesis. *Magnitude spectrograms* are obtained by considering the magnitude of the short-time Fourier transform (STFT) of a signal. In Figure 1b, the magnitude spectrogram of a 12 seconds music signal is displayed, revealing its harmonic structure. For the sake of illustration, Figure 1c displays the phase of the STFT of the same signal. At first sight, no structure can be observed: the phase spectrogram resembles a noise matrix.

The *power spectrogram* is another common time-frequency representation, computed with the squared magnitude of the STFT. It displays the repartition of power over time and frequency. From a statistical perspective, the power spectrogram is analogous to a variance. As seen in Figure 1d, this representation has a great dynamic range and discriminates largely between low-energy and high-energy time-frequency components of the signal. The *log-spectrogram* (1e) has the opposite property as it reduces the dynamic range. It is then frequently used to visualize low-energy components of the signal. The log-spectrogram is simply computed by considering the logarithm of the magnitude of the STFT.

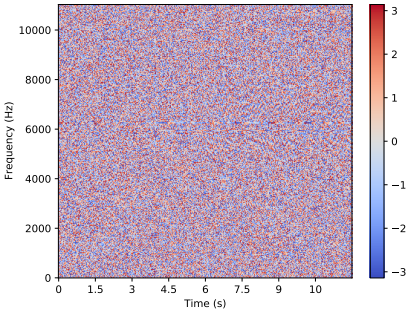
Alternative representations display time-frequency data with a logarithmic frequency axis in order to imitate human hearing, which is considered to be logarithmic with regards to frequency. The *Mel-spectrogram* is obtained via a transform approximating this property, resulting in a quasi-logarithmic spectrogram. Representations with a logarithmic frequency axis can also be computed via non-linear frequency filterbank-based transforms: the input signal is processed via a collection of bandpass filters and the resulting energies are displayed over time. The *Constant-Q transform* (CQT) *spectrogram* (1f) is a common representation belonging to this class. Analogously, the *chromagram* (1g) scales the frequency axis to semitones, the usual pitch unit of Western music systems.



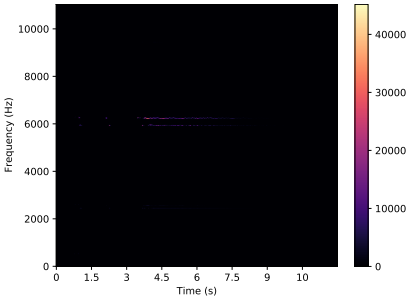
(a) Waveform



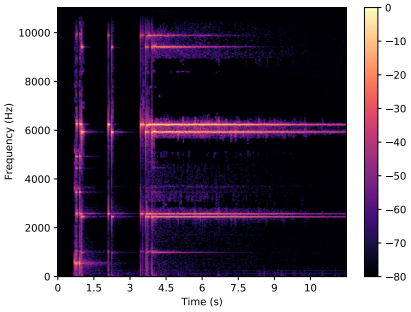
(b) Magnitude spectrogram



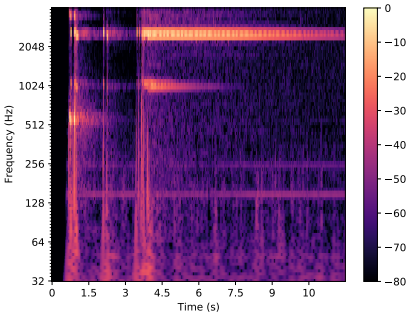
(c) Phase of STFT



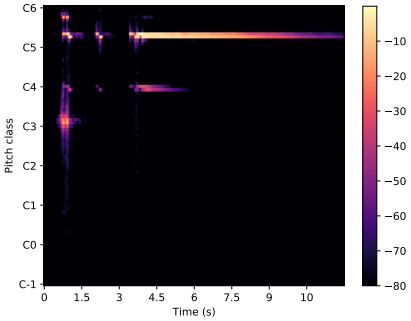
(d) Power spectrogram



(e) Log-spectrogram



(f) CQT spectrogram



(g) Chromagram

Figure 1: Waveform and time-frequency representations of the 12 first seconds of Bernard Parmegiani's *De Natura Sonorum - Incidences/battements*

The short-time Fourier transform

The short-time Fourier transform is commonly used in signal processing to analyze oscillatory and non-stationnary signals. Typical applications include audio, acoustics and seismology. The STFT consists in considering the local spectrum of a signal over a short temporal duration. This is done in practice by extracting short sections of the signal and computing their DFT. Given a discrete signal $\mathbf{x} \in \mathbb{C}^M$, the DFT is defined by:

$$\mathrm{DFT}(\mathbf{x})(m) := \sum_{\ell=0}^{M-1} x(\ell) e^{-i2\pi \frac{m}{M}\ell}, \tag{2.1.1}$$

where $m \in \{0, \dots, M-1\}$ denotes the indexes of the frequency bins. In order to better localize in frequency, the temporal segments are often multiplied by an *analysis window* before the DFT operation. This leads to the so-called *sliding-window* definition of the STFT. Given a discrete signal $\mathbf{x} \in \mathbb{C}^L$ and an analysis window $\mathbf{w} \in \mathbb{R}^M$ such that $M < L$, the STFT is the linear operator $\mathcal{A}_{\mathbf{w}}$ defined by:

$$[\mathcal{A}_{\mathbf{w}}\mathbf{x}](m, n) := (\mathrm{DFT}(\mathbf{x}_n))(m), \tag{2.1.2}$$

where $m \in \{0, \dots, M-1\}$, $n \in \{0, \dots, N-1\}$ respectively denote the indexes of the frequency bins and the time frames. $\mathbf{x}_n$ denotes the n-th windowed temporal frame of the signal $\mathbf{x}$:

$$x_n(\ell) := x(\ell) w(\ell - nH). \tag{2.1.3}$$

$H \in \mathbb{N}^*$ is called the *hop size* and controls the overlap between the successive frames. The *overlap ratio* is defined as $\frac{M-H}{M}$. The STFT writes:

$$[\mathcal{A}_{\mathbf{w}}\mathbf{x}](m, n) := \sum_{\ell=0}^{M-1} x(\ell) w(\ell - nH) e^{-i2\pi \frac{m}{M}\ell}. \tag{2.1.4}$$

The inverse-STFT (iSTFT) can also be constructed with the help of the inverse-DFT (iDFT). Given a complex vector $\mathbf{c} \in \mathbb{C}^M$, the iDFT is defined by:

$$\mathrm{iDFT}(\mathbf{c})(\ell) := \frac{1}{M} \sum_{m=0}^{M-1} c(m) e^{i2\pi \frac{m}{M}\ell}. \tag{2.1.5}$$

For each time index, the iDFT of the STFT frame is computed, resulting in a collection of temporal segments. The signal is then reconstructed through an *overlap-add* procedure. The segments are usually multiplied by a *synthesis window* before being summed up. This procedure is termed *weighted overlap-add* and leads to the following iSTFT definition. Given a complex-valued time-frequency matrix

$\mathbf{C} \in \mathbb{C}^{M \times N}$ and a synthesis window $\mathbf{v} \in \mathbb{R}^M$, the iSTFT is the linear operator $\mathcal{S}_{\mathbf{v}}$ defined by:

$$[\mathcal{S}_{\mathbf{v}}\mathbf{C}](\ell) := \sum_{n=0}^{N-1} [\text{iDFT}(C(\cdot, n))](\ell) v(\ell - nH), \tag{2.1.6}$$

where $\ell \in \{0, \dots, L-1\}$ is the time index. The iSTFT finally writes:

$$[\mathcal{S}_{\mathbf{v}}\mathbf{C}](\ell) := \frac{1}{M} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} C(m, n) v(\ell - nH) e^{i2\pi \frac{m}{M} \ell}. \tag{2.1.7}$$

Overlapp-add decomposition and perfect reconstruction

In order to be able to reconstruct $\mathbf{x}$ from the windowed temporal frames $\{\mathbf{x}_n\}_{n=0}^{N-1}$, the window and the hop size must respect a condition. Let $\mathbf{x}'$ denote the reconstructed signal such that:

$$x'(\ell) = \sum_{n=0}^{N-1} x_n(\ell) \tag{2.1.8}$$

$$= \sum_{n=0}^{N-1} x(\ell) w(\ell - nH) \tag{2.1.9}$$

$$= x(\ell) \sum_{n=0}^{N-1} w(\ell - nH). \tag{2.1.10}$$

In order to satisfy $\mathbf{x} = \mathbf{x}'$, the following *constant-overlap-add* (COLA) constraint must be respected:

$$\forall \ell \in \{0, \dots, L-1\}, \sum_{n=0}^{N-1} w(\ell - nH) = 1. \tag{2.1.11}$$

Windows that respect the COLA property include the Bartlett, the Hann and the Hamming window for half-overlap ($H = \frac{N}{2}$). With appropriate normalization, any COLA window for overlap H is also COLA for $H' = \frac{H}{2}, \frac{H}{3}, \dots, \frac{H}{H}$ if H' is an integer [129].

When applying the iSTFT after the STFT, the signal $\mathbf{x}'$ is reconstructed from temporal frames multiplied by a synthesis window $\mathbf{v}$:

$$x'(\ell) = \frac{1}{M} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \sum_{\ell'=0}^{M-1} x(\ell') w(\ell' - nH) e^{-i2\pi \frac{m}{M} \ell'} v(\ell - nH) e^{i2\pi \frac{m}{M} \ell} \tag{2.1.12}$$

$$= \frac{1}{M} \sum_{n=0}^{N-1} \sum_{\ell'=0}^{M-1} x(\ell') w(\ell' - nH) v(\ell - nH) \sum_{m=0}^{M-1} e^{-i2\pi \frac{m}{M} (\ell' - \ell)}. \tag{2.1.13}$$

The sum term $\sum_{m=0}^{M-1} e^{-i2\pi \frac{m}{M}(\ell'-\ell)}$ is equal to 0 unless $(\ell'-\ell)$ is an integer multiple of M. In this case,

$$\ell' = \ell + kM \quad \text{and} \quad \sum_{m=0}^{M-1} e^{-i2\pi mk} = M, \tag{2.1.14}$$

with $k \in \mathbb{Z}$. As $\mathbf{w}$ is equal to 0 out of its support, we consider here that $k = 0$. Therefore, the reconstructed signal writes:

$$x'(\ell) = \sum_{n=0}^{N-1} x(\ell) w(\ell - nH) v(\ell - nH) \tag{2.1.15}$$

$$= x(\ell) \sum_{n=0}^{N-1} v(\ell - nH) w(\ell - nH). \tag{2.1.16}$$

In order to recover $\mathbf{x}$ from $\mathbf{x}'$, the following constraint should be satisfied:

$$\forall \ell \in \{0, \dots, L-1\}, \sum_{n=0}^{N-1} v(\ell - nH) w(\ell - nH) = 1. \tag{2.1.17}$$

When this condition is respected, perfect reconstruction such that $\mathbf{x} = \mathcal{S}_{\mathbf{v}} \mathcal{A}_{\mathbf{w}} \mathbf{x}$ can be achieved and $\mathbf{v}$ and $\mathbf{w}$ are said to be *dual*. In practice, the analysis and synthesis windows are often chosen to be equal. In that case, the square-root of any nonnegative COLA window leads to perfect reconstruction. A common choice is the "root-Hann" window (also termed sine window) [78].

STFT and Gabor frames

The STFT can alternatively be written as the output of inner products between $\mathbf{x}$ and Gabor atoms $\gamma_{mn} \in \mathbb{C}^L$, which are functions built via translation and modulation of $\mathbf{w}$ as follows :

$$\gamma_{mn}(\ell) = w(\ell - nH) e^{i2\pi \frac{m}{M} \ell}. \tag{2.1.18}$$

By collecting the Gabor atoms into the columns of an $L \times MN$ matrix $\mathbf{\Gamma_w}$ and ignoring the time-frequency ordering, the STFT of a signal $\mathbf{x}$ can equivalently be obtained by $\mathbf{\Gamma_w^H x}$. Under general conditions [62], the matrix $\mathbf{\Gamma_w}$ defines a frame in the sense that there exists positive constants a and b such that for any $\mathbf{x} \in \mathbb{C}^L$:

$$a\|\mathbf{x}\|^2 \leqslant \|\mathbf{\Gamma_w^H x}\|^2 \leqslant b\|\mathbf{x}\|^2. \tag{2.1.19}$$

Similarly, the synthesis operator $\mathcal{S}_{\mathbf{v}}$ can be expressed as the adjoint of the STFT:

$$\mathcal{S}_{\mathbf{v}} \mathbf{C} = \mathbf{\Gamma_v c}, \tag{2.1.20}$$

where $\mathbf{c} \in \mathbb{C}^{MN}$ is a vectorized version of $\mathbf{C}$. As such, the windows $\mathbf{w}$ and $\mathbf{v}$ are dual if and only if $\mathbf{\Gamma_v \Gamma_w^H x} = \mathbf{x}$. When the same window

can be used for analysis and synthesis with perfect reconstruction (an example being the sine window [129]), then it can be shown that $a = b = 1$ and $\mathbf{\Gamma}_{\mathbf{w}}^{H}$ defines a so-called *Parseval frame*. In the rest of this thesis, the STFT operator will be denoted with the matrix $\mathbf{A}$ (equal to $\mathbf{\Gamma}_{\mathbf{w}}^{H}$). We assume that $\mathbf{w} = \mathbf{v}$ and that the Parseval frame assumption holds (i.e., $\mathbf{A}^{H}\mathbf{A} = \mathbf{I}_{L}$).

2.1.2 *Objective evaluation of audio quality*

In audio processing, evaluating the quality of a signal estimate is of paramount importance to assess reconstruction performance. However, this task is nontrivial as the notion of quality remains imprecise and related to subjective perception. A common criterion is the absence of degradation after applying a chain of processes.

Subjective tests can be conducted to evaluate the perceived quality with the ABC/HR [122] or MUSHRA [123] protocols. They usually output scores such as the Subjective Difference Grade (SDG) or the Mean Opinion Score (MOS), covering a scale from 1 (bad) to 5 (excellent). They are however costly and their reproducibility is sensitive to cognitive biases such as the listener's familiarity with the task, fatigue, or score-equalizing bias [163].

For that matter, objective tests were developed. They usually compute a measure of fit between a signal estimate and a reference, often termed as *ground-truth*. In the following, several objective evaluation scores are introduced. This is followed by a brief presentation of the reference datasets used in this thesis.

Evaluation in the time domain

The signal-to-distortion ratio (SDR) is defined as the ratio of the power of a signal of interest $\mathbf{x}$ over the distortion power. Its expression is:

$$\mathrm{SDR}(\mathbf{x}, \hat{\mathbf{x}}) = 10 \log_{10} \frac{\|\mathbf{x}\|^2}{\|\mathbf{x} - \hat{\mathbf{x}}\|^2}, \tag{2.1.21}$$

where $\hat{\mathbf{x}}$ is the estimate of $\mathbf{x}$. The SDR is expressed in decibels. It is included in the BSSEval toolbox [147], which is widely used to assess performance in audio source separation. BSSEval also encompasses two other metrics: the signal-to-interference ratio (SIR) and the signal-to-artifact ratio (SAR), which respectively measure the rejection of interferences and artifacts in the estimated signal. In [81], the authors propose a *scale-invariant* version of the SDR to evaluate estimation without taking in account scaling effects. Its expression is:

$$\text{SI-SDR}(\mathbf{x}, \hat{\mathbf{x}}) = 10 \log_{10} \frac{\| \frac{\hat{\mathbf{x}}^{H}\mathbf{x}}{\|\mathbf{x}\|^2}\mathbf{x}\|^2}{\| \frac{\hat{\mathbf{x}}^{H}\mathbf{x}}{\|\mathbf{x}\|^2}\mathbf{x} - \hat{\mathbf{x}}\|^2}, \tag{2.1.22}$$

with $\frac{\hat{\mathbf{x}}^H \mathbf{x}}{\|\mathbf{x}\|^2}$ the optimal scaling factor minimizing the quadratic error between the scaled reference and the estimate.

Evaluation in the time-frequency domain

Quality evaluation is also usually performed in the time-frequency domain using ℓ^p norms.

The spectral convergence (SC) [132] is computed from magnitude spectrograms with the ℓ^2 distance:

$$\mathrm{SC}(\mathbf{x}, \hat{\mathbf{x}}) = 10 \log_{10} \frac{\| |\mathbf{A}\mathbf{x}| - |\mathbf{A}\hat{\mathbf{x}}| \|^2}{\| |\mathbf{A}\mathbf{x}| \|^2}. \tag{2.1.23}$$

Another common choice is the ℓ^1 distance with log-spectrograms [2]:

$$\ell^1_{\log}(\mathbf{x}, \hat{\mathbf{x}}) = \| \log |\mathbf{A}\mathbf{x}| - \log |\mathbf{A}\hat{\mathbf{x}}| \|_1. \tag{2.1.24}$$

These two metrics differ in their behavior as cost functions: SC is likely to penalize mostly estimation errors on the large coefficients of the spectrogram while $\ell^1_{\log}$ is more sensitive to errors on the small coefficients.

Perceptually-motivated metrics

Other evaluation metrics aim to model the results obtained with subjective tests. Most of them were introduced as International Telecommunication Union (ITU) recommendations.

The Perceptual Evaluation of Audio Quality score (PEAQ) [24] was introduced in 1999 to model the MOS of perceived quality tests for general sounds. PEAQ includes psychoacoustics models based on filterbanks and time-frequency masks to compute several model variables. The latter are mapped to a single output with a basic neural network. PEMO-Q [70] was proposed later in 2006 for the same purpose. It embodies a simpler model that yet achieves better correlation with the subjective tests.

The Perceptual Evaluation of Speech Quality score (PESQ) [124] estimates the MOS of subjective quality tests for speech signals. This metric consists in computing measures of fit over time-aligned and modified time-frequency representations. It traditionally only considers bandpass filtered versions of the signals (due to its applications in telephony). The Perceptual Objective Listening Quality Assessment (POLQA) [7, 8] score is the successor of PESQ and assesses a larger amount of degradations with more intricate models.

The Short-time Objective Intelligibility score (STOI) [133] models the intelligibility of a speech signal. It is computed through filterbank decomposition and envelope correlation of the reference and estimate signals. STOI outputs a variable that ranges between 0 (unintelligible)

and 1 (excellent) and which has been shown to correlate well with subjective intelligibility measurements of speech.

The Perceptual Evaluation of Audio Source Separation (PEASS) toolbox [39] encompasses perceptual objective scores inspired by the BSSEval toolbox: the proposed criteria account for distortions, interferences, and artifacts in the estimated signal. They were also shown to correlate well with subjective tests in the context of audio source separation.

Datasets

In the experiments of this thesis, we consider two audio datasets. The Texas Instruments/Massachusetts Institute of Technology (TIMIT) [49] corpus is composed of speech signals recorded from 630 American English speakers of different genders and dialects. Each speaker reads 10 sentences selected to be phonetically rich. All the signals in the dataset are single-channel, sampled at 16kHz and 16-bit encoded.

The Free Music Archive (FMA) [34] contains 106,574 Creative Commons-licensed music tracks. They are representative of 16,341 artists and 161 genres. All tracks are provided with metadata and stereo MP3-encoded files. Most of them are sampled at 44.1kHz and encoded at a 320kbit/s bit rate.

2.2 OPTIMIZATION

In this section, we introduce elements and algorithms from optimization. First, we detail the Wirtinger calculus framework, which will be required to derive gradient-based algorithms with cost functions of a complex variable. We present therefore the gradient algorithm and a few variants including acceleration and adapted step sizes. Follows an introduction to the proximity operator and common proximal methods, a family of optimization algorithms frequently used in signal processing [26]. This section ends with an introduction to Bregman divergences, a family of functions that includes measures of fit deriving from a statistical perspective.

2.2.1 *Wirtinger calculus and gradient methods*

Wirtinger calculus

When handling complex-valued data, the use of gradient-based optimization algorithms implies to minimize cost functions of a complex variable. However, as cost functions are real-valued, they are not *complex differentiable*. This means that they do not follow the Cauchy–

Riemann equations. For a function f of a complex variable $z = z_r + iz_i$ the Cauchy–Riemann equations write:

$$\frac{\partial f_r(z)}{\partial z_r} = \frac{\partial f_i(z)}{\partial z_i} \quad \text{and} \quad \frac{\partial f_r(z)}{\partial z_i} = -\frac{\partial f_i(z)}{\partial z_r}, \tag{2.2.1}$$

with $f(z) = f_r(z) + if_i(z)$ and $f_r(z), f_i(z)$ are real-valued. The Wirtinger calculus, also termed $\mathbb{CR}$-calculus, provides a gradient-like operator for those functions. It sees any function of a complex variable as a function of its real and imaginary parts. The Wirtinger derivatives are then defined as:

$$\begin{aligned} \frac{\partial f}{\partial z}(z) &:= \frac{1}{2}\left(\frac{\partial f}{\partial z_r}(z_r, z_i) - i\frac{\partial f}{\partial z_i}(z_r, z_i)\right), \\ \frac{\partial f}{\partial z^*}(z) &:= \frac{1}{2}\left(\frac{\partial f}{\partial z_r}(z_r, z_i) + i\frac{\partial f}{\partial z_i}(z_r, z_i)\right). \end{aligned} \tag{2.2.2}$$

In practice, computing the derivative of f with respect to z (resp. z^*) can be done using usual differentiation by treating z (resp. z^*) as a real variable with z^* (resp. z) treated as a constant [11, 75]:

$$\frac{\partial f}{\partial z} = \left.\frac{\partial f(z, z^*)}{\partial z}\right|_{z^*=\text{const.}}, \tag{2.2.3}$$

$$\frac{\partial f}{\partial z^*} = \left.\frac{\partial f(z, z^*)}{\partial z^*}\right|_{z=\text{const.}}. \tag{2.2.4}$$

Besides, if f is real-valued, the following property is verified:

$$\left(\frac{\partial f}{\partial z}\right)^* = \frac{\partial f}{\partial z^*}. \tag{2.2.5}$$

In a multivariate setting, the gradient of f is then defined as:

$$\nabla f = \left[\frac{\partial f}{\partial z_1}, \ldots, \frac{\partial f}{\partial z_K}\right]^H. \tag{2.2.6}$$

When f is additionally real-valued, the following property holds from (2.2.2) and (2.2.5):

$$\nabla_{\mathbb{R}} f := \left[\frac{\partial f}{\partial z_r(1)}, \ldots, \frac{\partial f}{\partial z_r(K)}\right]^H = 2\Re(\nabla f), \tag{2.2.7}$$

where $\nabla_{\mathbb{R}} f$ denotes the gradient of f with regards to the real part of the variable.

Gradient methods

With the help of the Wirtinger framework, a *gradient descent* algorithm can be formulated to minimize a differentiable function f of a complex variable. The Wirtinger gradient descent algorithm is detailed in Algorithm 1, with μ_t denoting the gradient step size.

Algorithm 1 : (Wirtinger) gradient descent algorithm

Parameters : $(\mu_t) \in \mathbb{R}^{\mathbb{N}}$

Initialize $\mathbf{y}_0$.
while *stopping criteria not met* **do**
  $\mathbf{y}_{t+1} := \mathbf{y}_t - \mu_t \nabla f(\mathbf{y}_t)$
end

Algorithm 2 : Accelerated (Wirtinger) gradient descent algorithm

Parameters : $(\mu_t) \in \mathbb{R}^{\mathbb{N}}$, $\xi \in [0, 1]$

Initialize $\mathbf{y}_0$.
while *stopping criteria not met* **do**
  $\mathbf{q}_{t+1} := \mathbf{y}_t - \mu_t \nabla f(\mathbf{y}_t)$
  $\mathbf{y}_{t+1} := \mathbf{q}_{t+1} + \xi(\mathbf{q}_{t+1} - \mathbf{q}_t)$
end

Like the usual gradient method, the Wirtinger gradient descent converges to a critical point of the function f under conditions on the step size. It can also be accelerated similarly to Polyak's gradient descent with momentum [117]. The Accelerated Wirtinger gradient descent algorithm is displayed in Algorithm 2, where ξ denotes the acceleration parameter.

When ∇f is P-Lipschitz, a common choice for the gradient step size is $\mu_t < \frac{1}{P}$. Alternative strategies for the choice of the step size can be considered. A usual method consists in refining the gradient step size with a *backtracking line search*: at every gradient descent iteration, the gradient step size is repeatedly multiplied by a nonnegative factor smaller than 1 until a stopping criterion is reached. Typically, the following Armijo rule [3] is considered:

$$f(\mathbf{y}_{t+1}) < f(\mathbf{y}_t) - \frac{\mu_t}{2}\|\nabla f(\mathbf{y}_t)\|^2. \tag{2.2.8}$$

This rule can be relaxed such that the gradient step size is updated until the new cost value is smaller than the maximum cost value of the last iterations. This method is referred to as *non-monotonic backtracking line search* [61].

With inspiration from Newton's method, Barzilai and Borwein propose a gradient method with varying step sizes in [4]. Their method accounts for the curvature of the cost function by approximating second-order quantities with finite-difference schemes. The authors introduce two different step sizes:

- the "long" Barzilai-Borwein step:

$$\mu_t^{\mathrm{BB1}} := \frac{\|\mathbf{y}_{t-1} - \mathbf{y}_{t-2}\|^2}{\langle \nabla f(\mathbf{y}_{t-1}) - \nabla f(\mathbf{y}_{t-2})\,;\, \mathbf{y}_{t-1} - \mathbf{y}_{t-2}\rangle}, \tag{2.2.9}$$

- the "short" Barzilai-Borwein step:

$$\mu_t^{BB2} := \frac{\langle \nabla f(\mathbf{y}_{t-1}) - \nabla f(\mathbf{y}_{t-2}) \,;\, \mathbf{y}_{t-1} - \mathbf{y}_{t-2} \rangle}{\|\nabla f(\mathbf{y}_{t-1}) - \nabla f(\mathbf{y}_{t-2})\|^2}. \tag{2.2.10}$$

The Barzilai-Borwein method usually enables to reach the stopping criterion in fewer iterations without computing any Hessian, at the cost of a convergence theoretical guarantee. In the literature, several extensions to this method have been proposed [19, 31].

2.2.2 *Proximity operators and proximal methods*

Definitions

The proximity operator was introduced by Jean-Jacques Moreau [104] in 1962 as a generalization of the projection operator. It is now a fundamental tool in contemporary non-smooth optimization methods. With $\mathcal{H}$ being a Hilbert space, the proximity operator of a lower semi-continuous convex function $f \in \Gamma_0(\mathcal{H})$ is defined as the mapping of an input vector $\mathbf{y} \in \mathcal{H}$ to the unique solution of the following minimization problem:

$$\text{prox}_{\rho^{-1} f}(\mathbf{y}) := \underset{\mathbf{x} \in \mathcal{H}}{\text{argmin}}\, f(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{x} - \mathbf{y}\|^2. \tag{2.2.11}$$

It can alternatively be defined with the help of the subdifferential operator:

$$\text{prox}_{\rho^{-1} f}(\mathbf{y}) := (\text{Id} + \rho^{-1} \partial f)^{-1}(\mathbf{y}), \tag{2.2.12}$$

with $\partial \cdot$ being the subdifferential operator, i. e. , the mapping of a convex function to the set of its subgradients, defined as follows:

$$\partial f(\mathbf{y}) := \{\mathbf{v} \in \mathcal{H} \,|\, \forall \mathbf{x} \in \mathcal{H},\, \langle \mathbf{x} - \mathbf{y} \,;\, \mathbf{v} \rangle + f(\mathbf{y}) \leqslant f(\mathbf{x})\}. \tag{2.2.13}$$

The proximity operator can also be extended to nonconvex functions, resulting in a set-valued operator. In the literature, closed-form expressions of the mapping are known only for a limited number of families of functions (e. g. , indicator functions or ℓ^p norms for some values of p).

Properties and characterization

In the literature, a consequential number of properties for proximity operators and proximal calculus can be found. Only a handful of them will be detailed in the context of this thesis.

One of the fundamental properties of proximity operators is that the fixed points of prox_f are the minimizers of f for every $f \in \Gamma_0(\mathcal{H})$:

$$\forall \mathbf{y} \in \mathcal{H},\, \mathbf{y} = \text{prox}_f(\mathbf{y}) \Leftrightarrow \mathbf{y} \in \text{argmin}\, f. \tag{2.2.14}$$

Moreover, the proximity operator of f is non-expansive [105], i. e. , :

$$\forall \mathbf{y}, \mathbf{y}' \in \mathcal{H}, \|\text{prox}_f(\mathbf{y}) - \text{prox}_f(\mathbf{y}')\| \leqslant \|\mathbf{y} - \mathbf{y}'\|. \tag{2.2.15}$$

The two properties (2.2.14) and (2.2.15) lead to the following theorem, which is essential to optimization with proximity operators:

Theorem 1 ([5]). *Let $f \in \Gamma_0(\mathcal{H})$. If f has a minimizer, the recursion $\mathbf{y}_{t+1} = prox_f(\mathbf{y}_t)$ converges to the minimizer of f as t increases for any $\mathbf{y}_0 \in \mathcal{H}$.*

Under some conditions, a function $g : \mathcal{H} \to \mathcal{H}$ can be characterized as the proximity operator of some convex lower semi-continuous (l.s.c.) function. In [105], Moreau proposed the following characterization theorem:

Theorem 2 ([105]). *It exists $f \in \Gamma_0(\mathcal{H})$ such that $g : \mathcal{H} \to \mathcal{H}$ is the proximity operator of f if and only if the two conditions listed below are met:*

1. *It exists $h \in \Gamma_0(\mathcal{H})$ such that:*

$$\forall \mathbf{y} \in \mathcal{H}, g(\mathbf{y}) \in \partial h(\mathbf{y}). \tag{2.2.16}$$

2. *g is non-expansive, i. e. , :*

$$\forall \mathbf{y}, \mathbf{y}' \in \mathcal{H}, \|g(\mathbf{y}) - g(\mathbf{y}')\| \leqslant \|\mathbf{y} - \mathbf{y}'\|. \tag{2.2.17}$$

When $\mathcal{H} = \mathbb{R}$, the following corollary can be deduced [25]:

Corollary 2.1. *It exists $f \in \Gamma_0(\mathbb{R})$ such that $g : \mathbb{R} \to \mathbb{R}$ is the proximity operator of f if and only if g is non-decreasing and non-expansive.*

In [58], the authors extend the theorem to eventually nonconvex functions $f : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ and demonstrate a relation between f, g and h. In that case, the non-expansiveness condition (2.2.17) can be dismissed and the theorem writes:

Theorem 3 ([58]). *Let $\mathcal{Y}$ be a non-empty subset of $\mathcal{H}$. It exists $f : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ such that $g : \mathcal{Y} \to \mathcal{H}$ is the proximity operator of f if and only if the following condition is met:*

- *It exists $h \in \Gamma_0(\mathcal{H})$ such that:*

$$\forall \mathbf{y} \in \mathcal{Y}, g(\mathbf{y}) \in \partial h(\mathbf{y}). \tag{2.2.18}$$

When (2.2.18) holds, there exist $f : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ and $h \in \Gamma_0(\mathcal{H})$ such that:

$$\forall \mathbf{y} \in \mathcal{Y}, h(\mathbf{y}) = \langle \mathbf{y} ; g(\mathbf{y}) \rangle - \frac{1}{2}\|g(\mathbf{y})\|^2 - f(g(\mathbf{y})). \tag{2.2.19}$$

Corollary 3.1. *Let $\mathcal{Y}$ be a non-empty subset of $\mathbb{R}$. $g : \mathcal{Y} \to \mathbb{R}$ is the proximity operator of some function f if and only if g is non-decreasing.*

Algorithm 3 : Proximal point algorithm

Parameters : $\rho > 0$.

Initialize $\mathbf{y}_0$.
while *stopping criteria not met* **do**
  $\mathbf{y}_{t+1} := \text{prox}_{\rho^{-1} f}(\mathbf{y}_t)$
end

Algorithm 4 : Proximal gradient algorithm

Parameters : $(\mu_t) \in \mathbb{R}^{\mathbb{N}}$

Initialize $\mathbf{y}_0$.
while *stopping criteria not met* **do**
  $\mathbf{y}_{t+1} := \text{prox}_{\mu_t f_2}(\mathbf{y}_t - \mu_t \nabla f_1(\mathbf{y}_t))$
end

Theorem 3 and relation (2.2.19) imply that for any function g that can be characterized as the proximity operator of a function f, the expression of f is connected to h, a "primitive" function of which g is a subgradient. When g is invertible, the expression of f can be retrieved with the change of variable $\mathbf{x} = g(\mathbf{y})$:

$$f(\mathbf{x}) = \langle g^{-1}(\mathbf{x}) \, ; \, \mathbf{x} \rangle - \frac{1}{2}\|\mathbf{x}\|^2 - h(g^{-1}(\mathbf{x})). \tag{2.2.20}$$

Proximal methods

The proximity operator is an essential component of a class of convex optimization algorithms entitled *proximal methods*. We discuss here a few of them.

We consider the following problem of minimizing $f \in \Gamma_0(\mathcal{H})$. With the help of the Theorem 1, the proximal point algorithm displayed in Algorithm 3 is shown to converge to the minimizer of f [5] for any positive ρ.

We now assume that f can be splitted into two convex terms $f_1, f_2 \in \Gamma_0(\mathcal{H})$. The optimization problem now writes:

$$\text{minimize} \quad f_1(\mathbf{y}) + f_2(\mathbf{y}), \tag{2.2.21}$$

If f_1 is differentiable, the *proximal gradient algorithm* [32, 85] can be written as in Algorithm 4, where μ_t denotes the gradient step size.

The proximal gradient algorithm is shown to converge if ∇f_1 is Lipschitz, μ_t is fixed and chosen smaller than the inverse of the Lip-

Algorithm 5 : Alternating direction method of multipliers

Parameters : $\rho > 0$.

Initialize $\mathbf{u}_0$, $\boldsymbol{\lambda}_0$.

while *stopping criteria not met* **do**

  $\mathbf{y}_{t+1} := \text{prox}_{\rho^{-1} f_1}(\mathbf{u}_t - \boldsymbol{\lambda}_t)$

  $\mathbf{u}_{t+1} := \text{prox}_{\rho^{-1} f_2}(\mathbf{y}_{t+1} + \boldsymbol{\lambda}_t)$

  $\boldsymbol{\lambda}_{t+1} := \boldsymbol{\lambda}_t + \mathbf{y}_{t+1} - \mathbf{u}_{t+1}$

end

schitz constant. If f_2 is an indicator function, this algorithm reduces to the *projected gradient algorithm*.

The *Alternating direction method of multipliers* (ADMM) [12] also considers a splitting of the optimizaton problem, with f_1, f_2 eventually both non-differentiable. It consists in minimizing the following *Lagrangian* term $\mathcal{L}$ with regards to each variable alternatively.

$$\mathcal{L}(\mathbf{y}, \mathbf{u}, \boldsymbol{\lambda}) = f_1(\mathbf{y}) + f_2(\mathbf{u}) + \boldsymbol{\lambda}^T(\mathbf{y} - \mathbf{u}) + \frac{\rho}{2}\|\mathbf{y} - \mathbf{u}\|^2, \tag{2.2.22}$$

where $\mathbf{u}$ is an auxiliary variable and $\boldsymbol{\lambda}$ the Lagrange multiplier. The algorithm is summarized in Algorithm 5.

ADMM is shown to converge in terms of objective function and residual (i. e. , $\mathbf{y}$ and $\mathbf{u}$ converge to the same value) for any positive ρ.

2.2.3 *Bregman divergences*

Definition

Bregman divergences are a class of functions measuring the difference between two points. A Bregman divergence [1] $\mathcal{D}_\psi$ is defined from a strictly-convex and continuously-differentiable generating function ψ as follows:

$$\mathcal{D}_\psi(\mathbf{y} \,|\, \mathbf{z}) = \sum_k d_\psi(y_k \,|\, z_k), \tag{2.2.23}$$

where $d_\psi(y_k \,|\, z_k) = \psi(y_k) - \psi(z_k) - \psi'(z_k)(y_k - z_k)$ and ψ' is the derivative of the generating function. Bregman divergences are non-negative, convex with regards to their first argument and generally non-symmetric (i. e. , $\mathcal{D}_\psi(\mathbf{y} \,|\, \mathbf{z}) \neq \mathcal{D}_\psi(\mathbf{z} \,|\, \mathbf{y})$). In this thesis, the divergences are refered to as "left" (respecitvely "right") when they are considered as functions of their first (resp. second) argument with fixed second (resp. first) argument.

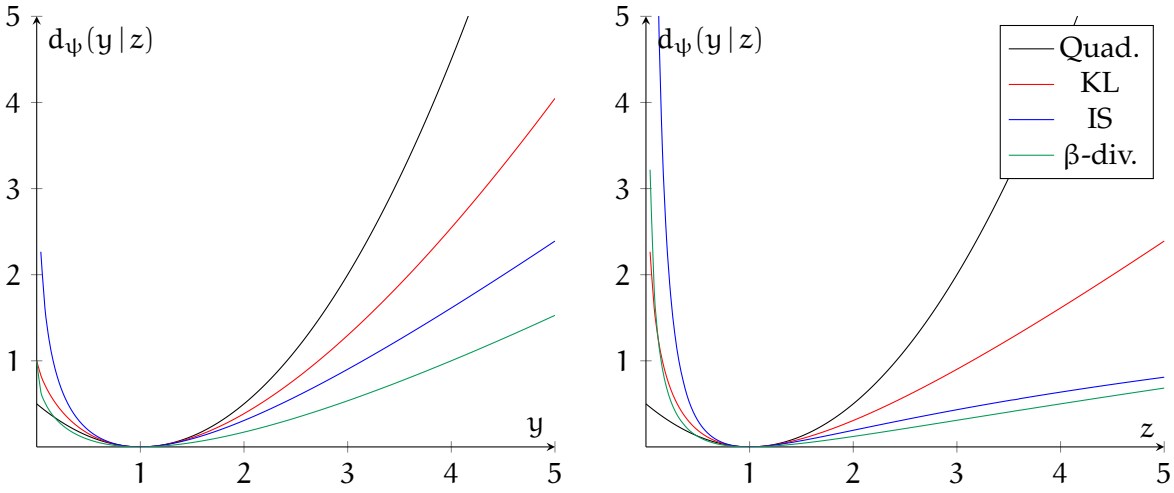
Bregman divergences include many well-known divergences and distances such as beta-divergences [66], which include the Kullback–Leibler and Itakura–Saito divergences as well as the quadratic cost

1 Note that we consider separable divergences in this thesis.

function. Examples of Bregman divergences and their generating functions can be found in Table 1. Figure 2 displays plots of usual Bregman divergences with fixed first and second arguments.

Table 1: Typical Bregman divergences generating functions with their first derivatives.

Divergence	$d_\psi(y \mid z)$	$\psi(y)$	$\psi'(y)$
Quadratic cost	$\frac{1}{2}(y-z)^2$	$\frac{1}{2}y^2$	y
Kullback-Leibler	$y(\log y - \log z) - (y-z)$	$y \log y$	$1 + \log y$
Itakura-Saito	$\frac{y}{z} - \log \frac{y}{z} - 1$	$-\log y$	$-y^{-1}$
beta-divergence ($\beta \in \mathbb{R} \setminus \{0,1\}$)	$\frac{y^\beta}{\beta-1} - \frac{\beta y z^{\beta-1}}{\beta-1} + z^\beta$	$\frac{y^\beta}{\beta(\beta-1)} - \frac{y}{\beta-1} + \frac{1}{\beta}$	$\frac{y^{\beta-1}-1}{\beta-1}$



(a) Bregman divergences with $z = 1$.

(b) Bregman divergences with $y = 1$.

Figure 2: Bregman divergences plots with fixed first (left) and second (right) arguments. $\beta = 0.5$ with the beta-divergence.

Statistical interpretation

Many usual Bregman divergences can be interpreted under the statistical lens as likelihood functions [128]. This means that there exists a probability density function p such that:

$$-\log p(\mathbf{y} \mid \mathbf{z}) = a\mathcal{D}_\psi(\mathbf{y} \mid \mathbf{z}) + b, \tag{2.2.24}$$

where a and b are constants and a is nonnegative.

For example, minimizing the KL divergence between $\mathbf{y}$ and $\mathbf{z}$ assumes that $\mathbf{y}$ follows a Poisson model [150]. Similarly, minimizing the IS divergence implies a multiplicative Gamma noise model while the quadratic cost function implies an additive Gaussian noise model [42].

Proximity operator of usual Bregman divergences

Table 2: Proximity operators of some standard (convex) Bregman divergences. $\mathcal{W}$ is the Lambert W function (i.e., the inverse relation of $z \mapsto ze^z$) applied entry-wise.

Divergence	Expression f	Proximity operator $\mathrm{prox}_f(\mathbf{y})$
Quadratic	$\frac{1}{2\rho}\|\cdot - \mathbf{z}\|^2$	$\frac{\rho\mathbf{y}+\mathbf{z}}{\rho+1}$
left KL	$\rho^{-1}\mathcal{D}_{\mathrm{KL}}(\cdot\,\|\,\mathbf{z})$	$\rho^{-1}\mathcal{W}(\rho\mathbf{z}\odot e^{\rho\mathbf{y}})$
left IS	$\rho^{-1}\mathcal{D}_{\mathrm{IS}}(\cdot\,\|\,\mathbf{z})$	$\frac{1}{2\rho}(-\mathbf{z}^{-1}+\rho\mathbf{y}\pm\sqrt{\Delta'})$ with $\Delta' := 4\rho+(\mathbf{z}^{-1}-\rho\mathbf{y})^2$
right KL	$\rho^{-1}\mathcal{D}_{\mathrm{KL}}(\mathbf{z}\,\|\,\cdot)$	$\frac{1}{2\rho}(\mathbf{y}-1\pm\sqrt{\Delta})$ with $\Delta := 4\rho\mathbf{z}+(1-\mathbf{y})^2$

A closed-form expression of the proximity operator can be obtained for some of the usual Bregman divergences, such as the quadratic cost function and the KL right and left divergences [26, 38]. These are summarized in Table 2.

To the best of our knowledge, the proximity operator of the left IS divergence has not been derived in closed-form in the literature. Therefore, for the sake of completeness, we derive it in Appendix 2.A.

2.3 DEEP LEARNING

2.3.1 *Neural networks*

Definition

Neural networks are a class of machine learning models inspired by the behavior of the biological brain [126]. They are constructed as a composition of operations (*neurons*) computed on data. We consider in the following the feedforward neural network F:

$$\hat{\mathbf{z}} = \mathsf{F}(\mathbf{y}) \text{ and } \mathsf{F} = \mathsf{F}_T \circ \cdots \circ \mathsf{F}_1. \tag{2.3.1}$$

The functions F_t are termed *layers* of the network, while $\mathbf{y}$ and $\hat{\mathbf{z}}$ are respectively refered to as *input* and *output* of the network. Most of the time, every layer F_t is composed of a linear operation (e. g. , a multiplication or a convolution) and an entrywise nonlinear operation, termed *activation function*. We denote by $\boldsymbol{\Theta}$ the collection of all the parameters of the network.

Training a neural network

Neural networks can be considered as universal approximators of any continuous function [30, 57, 68]. They have the ability to model the relationship between inputs and outputs after a training stage. For this purpose, the supervised learning framework considers training as a minimization problem between observed and predicted data:

$$\underset{\Theta}{\text{minimize}} \quad \sum_{(\mathbf{y}_i, \mathbf{z}_i) \in \boldsymbol{\Delta}} J(\mathbf{z}_i, F(\mathbf{y}_i)), \tag{2.3.2}$$

where J is a cost function and $\boldsymbol{\Delta}$ is the training dataset, i. e., a collection of input/output pairs $(\mathbf{y}_i, \mathbf{z}_i)_{i=1,\dots,I}$. The training stage is achieved with the help of an *optimizer*, i. e., an iterative optimization algorithm relying on gradient computation. For each training step, the parameters of the network are updated. A wide range of optimizers exist in the literature and the most popular include Stochastic Gradient Descent (SGD) [125], AdaGrad [37] and Adam [71].

2.3.2 *Unfolding iterative algorithms*

Unfolding (or unrolling) is an attempt to include model knowledge in learning-based approaches. This strategy consists in considering each iteration of a model-based optimization algorithm as a trainable neural layer. This results in a deep neural network with an explainable architecture and a limited number of parameters. Furthermore, empirical work suggests that it is prone to have a good ability to generalize to unseen data or experimental conditions [83].

An example: ISTA and LISTA

We consider in the following example the LASSO problem [139]:

$$\underset{\mathbf{y}}{\text{minimize}} \quad \|\mathbf{D}\mathbf{y} - \mathbf{z}\|^2 + \eta\|\mathbf{y}\|_1 \tag{2.3.3}$$

The formulation of LASSO implies knowledge on $\mathbf{y}$: the sparsity of $\mathbf{y}$ is promoted by regularizing (2.3.3) with the ℓ^1 norm.

The Iterative Shrinkage-Thresholding Algorithm (ISTA) is a proximal gradient algorithm applied to LASSO. It alternates a gradient step to descend the least-squares part of (2.3.3) and a proximal step to descend the regularization term. The proximity operator of the ℓ^1 norm is known as *soft-thresholding* operator and it is defined as follows:

$$S_\nu(\mathbf{y}) = \text{sign}(\mathbf{y})(|\mathbf{y}| - \nu)_+, \tag{2.3.4}$$

where $\text{sign}(\cdot)$ returns (entrywise) the sign of its input and $(\cdot)_+$ its positive part.

ISTA is detailed in Algorithm 6, with μ_t denoting the gradient step.

Algorithm 6 : Iterative Shrinkage-Thresholding Algorithm

Parameters : $(\mu_t) \in \mathbb{R}^{\mathbb{N}}$, $\eta \in \mathbb{R}$.

Initialize $\mathbf{y}_0$.

while *stopping criteria not met* **do**

  $\mathbf{y}_{t+1} := \mathsf{S}_{\eta\mu_t}\left(\mathbf{y}_t - \mu_t \mathbf{D}^{\mathsf{H}}(\mathbf{D}\mathbf{y}_t - \mathbf{z})\right)$

end

When considering a finite number of ISTA iterations, one can already note a similarity with deep neural networks: the algorithm alternates affine transforms (the gradient steps) with non-linear entrywise operations (the soft-thresholding steps). The latter can be interpreted as activation functions.

In [56], Gregor and Le Cun propose to unfold ISTA in a deep neural network entitled Learned ISTA (LISTA). They first rewrite the ISTA iteration as:

$$\mathbf{y}_{t+1} = \mathsf{S}_{\nu}\left(\mathbf{W}_{(\mathbf{z})}\mathbf{z} + \mathbf{W}_{(\mathbf{y})}\mathbf{y}_t\right), \tag{2.3.5}$$

with $\mathbf{W}_{(\mathbf{z})} = \mu\mathbf{D}^{\mathsf{H}}$, $\mathbf{W}_{(\mathbf{y})} = \mathrm{Id} - \mu\mathbf{D}^{\mathsf{H}}\mathbf{D}$ and constant gradient step.

The authors unfold the algorithm and choose $\nu, \mathbf{W}_{(\mathbf{y})}$ and $\mathbf{W}_{(\mathbf{z})}$ as learnable parameters, shared among the layers of the network. LISTA compares advantageously to ISTA by producing better solutions with fewer iterations while respecting the sparsity model. Moreover, its architecture is explainable.

Unfolded iterative algorithms for inverse problems

The deep unfolding technique has been applied to a wide range of works following the seminal work [56] and has found application in numerous fields other than audio signal processing. A non-exhaustive list is detailed in the following.

In [10], the authors propose a neural network architecture from a variational formulation of the image restoration problem. They unfold the proximal point algorithm, resulting in a network that outperforms state-of-the-art methods for image deblurring tasks.

Unfolding is considered for an image super-resolution task in [160]. After detailing the degradation model and the formulation of the problem, the authors propose to unfold the half-quadratic splitting algorithm. The proposed network is shown to perform comparably to other standard learning-based methods.

The unfolding of NMF is also proposed in [109], where the authors introduce deep architectures for both supervised and unsupervised learning settings. Their framework outperformed standard approaches in biological data analysis tasks. In [67], the authors propose a deep NMF framework based on unfolding for speech enhancement. By untying the parameters and despite using far fewer, their

approach is competitive with traditional neural networks in their experimental work.

The authors of [88] design a neural network from the iterations of the projected gradient algorithm for multi-spectral image fusion. A CNN replaces the projection operator in this case. In [69], the authors unfold the proximal gradient algorithm for MRI data reconstruction. They add skip connections to the network to simulate the memory of the first iterates in the late layers.

Activation functions and proximity operators

With the LISTA example, the soft-threshoding operator was interpreted as an activation function. Most of the latter can indeed be interpreted as proximity operators. This connection has been investigated in [27], where the authors observe that most of the usual activation functions belong to the same class of functions, that can be characterized as proximity operators. They denote $\mathbb{A}(\mathbb{R})$ the set of non-decreasing, non-expansive functions from $\mathbb{R}$ to $\mathbb{R}$ that take value 0 at 0. With the help of the Corollary 2.1 and (2.2.14), the following theorem can be deduced:

Theorem 4 ([27])**.** *Let* $g : \mathbb{R} \to \mathbb{R}$*. Then,* $g \in \mathbb{A}(\mathbb{R})$ *if and only if there exists* $f \in \Gamma_0(\mathbb{R})$*, which is minimized in* 0*, such that* $g = prox_f$*.*

Following Theorems 3 and 4, most of the usual activation functions can be expressed as proximity operators of functions that can be characterized. Table 3 includes illustrations from the original study [27].

Table 3: Usual activation functions as proximity operators

Name	$\operatorname{prox}_f(y)$	$f(y)$
Identity	y	0
ReLU	$(y)_+$	$\iota_{[0;+\infty[}(y)$
PReLU	$\begin{cases} y & \text{if } y \geqslant 0, \\ \alpha y & \text{else.} \end{cases}$	$\begin{cases} 0 & \text{if } y \geqslant 0, \\ y^2 \frac{1-\alpha}{2\alpha} & \text{else.} \end{cases}$
Sigmoïd	$\frac{1}{1+e^{-y}} - \frac{1}{2}$	$\begin{cases} (y+\frac{1}{2})\log(y+\frac{1}{2}) + (\frac{1}{2}-y)\log(\frac{1}{2}-y) - \frac{1}{2}(y^2+\frac{1}{4}) & \text{if } \lvert y\rvert < \frac{1}{2}, \\ -\frac{1}{4} & \text{if } \lvert y\rvert = \frac{1}{2}, \\ +\infty & \text{else.} \end{cases}$
Arctangent	$\frac{2}{\pi}\arctan$	$\begin{cases} -\frac{2}{\pi}\log(\cos(\frac{\pi y}{2})) - \frac{1}{2}y^2 & \text{if } \lvert y\rvert < 1, \\ +\infty & \text{else.} \end{cases}$
Hyperbolic tangent	$\tanh$	$\begin{cases} \frac{(1+y)\log(1+y)+(1-y)\log(1-y)-y^2}{2} & \text{if } \lvert y\rvert < 1, \\ \log(2) - \frac{1}{2} & \text{if, } \lvert y\rvert = 1, \\ +\infty & \text{else.} \end{cases}$

APPENDICES

2.A PROXIMITY OPERATOR OF THE LEFT IS DIVERGENCE

The closed-form expression of the proximity operator of the left IS divergence is detailed hereafter.

Lemma 1.

$$\forall \mathbf{y} \in \mathbb{R}^K, \quad prox_{\rho^{-1} \mathcal{D}_\psi(\cdot\,|\,\mathbf{z})}(\mathbf{y}) = \frac{1}{2\rho}(-\mathbf{z}^{-1} + \rho\mathbf{y} \pm \sqrt{\Delta'}), \quad (2.A.1)$$

with $\Delta' := 4\rho + \left(\mathbf{z}^{-1} - \rho\mathbf{y}\right)^2$.

Proof. Let us consider ψ such that $\psi(y) = -\log y$. We consider the problem (2.2.11) with $f(\mathbf{y}) = D_\psi(\mathbf{y}\,|\,\mathbf{z})$. Note that such a function is defined only for vectors with nonnegative entries. We can, however, broaden its definition domain to $\mathbb{R}^K$ by assuming that $\mathcal{D}_\psi(\mathbf{y}\,|\,\mathbf{z}) = +\infty$ if $\mathbf{y} \notin \mathbb{R}_+^K$ [38]. We then look for $\mathbf{y}$ such that $\nabla Q(\mathbf{y}) = \mathbf{0}$, where $Q(\mathbf{y}) = D_\psi(\mathbf{y}\,|\,\mathbf{z}) + \frac{\rho}{2}\|\mathbf{y} - \mathbf{p}\|^2$. We have:

$$\nabla Q(\mathbf{y}) = \psi'(\mathbf{y}) - \psi'(\mathbf{z}) + \rho(\mathbf{y} - \mathbf{p}) \quad (2.A.2)$$

$$= \mathbf{z}^{-1} - \mathbf{y}^{-1} + \rho(\mathbf{y} - \mathbf{p}). \quad (2.A.3)$$

Therefore,

$$\nabla Q(\mathbf{y}) = \mathbf{0} \iff \mathbf{y} \odot \mathbf{z}^{-1} - \mathbf{1} + \rho\mathbf{y} \odot (\mathbf{y} - \mathbf{p}) = \mathbf{0} \quad (2.A.4)$$

$$\iff \rho\mathbf{y}^2 + (\mathbf{z}^{-1} - \rho\mathbf{p}) \odot \mathbf{y} - \mathbf{1} = \mathbf{0}. \quad (2.A.5)$$

Finally:

$$\operatorname{prox}_{\rho^{-1} \mathcal{D}_\psi(\cdot\,|\,\mathbf{z})}(\mathbf{y}) = \frac{1}{2\rho}(-\mathbf{z}^{-1} + \rho\mathbf{y} + \sqrt{\Delta'}), \quad (2.A.6)$$

where $\Delta' := 4\rho + (\mathbf{z}^{-1} - \rho\mathbf{y})^2$. □

3

RELATED WORK

3.1 The phase retrieval problem 31
3.1.1 Problem formulation 31
3.1.2 Algorithms 32
3.2 Phase retrieval in audio 36
3.2.1 Context and applications 36
3.2.2 Specific algorithms 36
3.2.3 Phase retrieval for audio source separation . . . 40
3.3 Phase retrieval with deep learning 41

This chapter introduces the problems addressed in this thesis as well as common methods to tackle them. Section 3.1 defines the phase retrieval problem and provides an overview of the usual approaches considered for its resolution. In Section 3.2, the phase retrieval problem is examined from the audio perspective and specific algorithms are presented. The chapter ends in Section 3.3, which describes learning-based methods: deep neural networks in the context of audio signal recovery and unfolded iterative algorithms.

3.1 THE PHASE RETRIEVAL PROBLEM

3.1.1 *Problem formulation*

The phase retrieval (PR) problem consists in reconstructing a signal from phaseless nonnegative measurements. It occurs in a variety of fields including optical imaging [152], astronomy [45], X-ray crystallography [64], and audio signal processing [52, 107], which is the main motivation of this thesis. In this manuscript, the measurements are modeled as follows:

$$\mathbf{r} \approx |\mathbf{A}\mathbf{x}^\star|^d, \tag{3.1.1}$$

where $\mathbf{x}^\star \in \mathbb{C}^L$ is the unknown signal, $\mathbf{A} \in \mathbb{C}^{K\times L}$ is the measurement operator and $\mathbf{r} \in \mathbb{R}_+^K$ collects the phaseless measurements. In practice, the measurements are most of the time either magnitude ($d = 1$) or power ($d = 2$) measurements.

The PR problem is inherently ill-posed as different signals can generate identical measurements. The retrieved signal can thus only be recovered up to a certain level of ambiguity depending on the measurement operator. A trivial ambiguity is the *global phase*: if $\mathbf{x}$ is a solution to PR, $c\mathbf{x}$ is also a solution for all scalar $c \in \mathbb{C}$ such that $|c| = 1$.

Algorithm 7 : Error Reduction algorithm

Initialize $\mathbf{x}_0$.
while *iterate* **do**
  $\mathbf{x}_{t+1} := \mathcal{P}_{\mathcal{S}_0}\left(\mathbf{A}^\dagger\left(\mathbf{r}\odot\frac{\mathbf{A}\mathbf{x}_t}{|\mathbf{A}\mathbf{x}_t|}\right)\right)$
end

3.1.2 *Algorithms*

Phase retrieval may be tackled with various conventional optimization algorithms. An overview of the main methods is detailed in the following. These approaches can be divided into two groups: nonconvex and convex methods.

Nonconvex methods

Phase retrieval is usually expressed as an optimization problem involving a quadratic error function:

$$\min_{\mathbf{x}\in\mathbb{C}^L} \| |\mathbf{A}\mathbf{x}|^d - \mathbf{r}\|^2, \tag{3.1.2}$$

As this formulation is nonconvex, some prior knowledge about the unknown signal and the measurement operator is necessary to yield a meaningful and good quality estimate. Initialization is also crucial in order to converge to better local minima.

In the seminal work [44], the *Error Reduction algorithm* (ER) is proposed to solve PR with Fourier magnitude measurements ($d = 1$ and $\mathbf{A}$ is the DFT). It can be encompassed in the class of alternating projections algorithms. ER alternates two projections: one onto the set of measurements whose magnitude is equal to $\mathbf{r}$ and the other onto $\mathcal{S}_0$, the set of signals that satisfy a specific support constraint. The ER algorithm is displayed in Algorithm 7 with $\mathcal{P}_{\mathcal{S}_0}$ being the projection operator on $\mathcal{S}_0$.

ER generalizes the *Gerchberg-Saxton algorithm* (GSA) [51], an alternating projections algorithm to reconstruct a signal from its modulus and its Fourier magnitude. ER can also be interpreted as a gradient descent algorithm for the quadratic loss (3.1.2) and is shown to converge to a stationnary point [40].

Wirtinger Flow (WF) [16] is another gradient descent algorithm for (3.1.2), with power measurements ($d = 2$). It consists of two steps:

1. an initialization based on a spectral method, that computes the largest eigenvector of the matrix $\mathbf{A}^H\mathrm{diag}(\mathbf{r})\mathbf{A}$;
2. a gradient descent update, calculated via the Wirtinger gradient (see Section 2.2).

Algorithm 8 : Wirtinger Flow algorithm

Parameters : $(\mu_t) \in \mathbb{R}^{\mathbb{N}}$

Initialize $\mathbf{v}_0 \in \mathbb{C}^L$.
for $t = 0$ **to** T **do**
  $\mathbf{v}_{t+1} := \frac{\mathbf{A}^H \mathrm{diag}(\mathbf{r}) \mathbf{A}\mathbf{v}_t}{\|\mathbf{A}^H \mathrm{diag}(\mathbf{r}) \mathbf{A}\mathbf{v}_t\|}$
end
$\mathbf{x}_0 := \mathbf{v}_T$;
while *iterate* **do**
  $\mathbf{x}_{t+1} := \mathbf{x}_t - \mu_t \mathbf{A}^\dagger \left[(\mathbf{A}\mathbf{x}_t) \odot (|\mathbf{A}\mathbf{x}_t|^2 - \mathbf{r}) \right]$
end

In practice, the initialization is computed via the power method, a standard numerical technique to estimate the dominant eigenvector of a matrix [14]. The algorithm is summarized in Algorithm 8 where μ_t denotes the gradient step size.

In [16], the step size is set heuristically and grows exponentially in the first iterations before being fixed at a constant. Variations of this algorithm include the *Truncated Amplitude Flow* [155] and the *Thresholded Wirtinger Flow* algorithms [15].

The *ADMM* algorithm has been used several times to address phase retrieval as well. In [84], PR is expressed as the following constrained problem by introducing auxiliary variables for the magnitude and phase of $\mathbf{Ax}$:

$$\min_{\mathbf{x} \in \mathbb{C}^L, \mathbf{u} \in \mathbb{R}_+^K, \theta \in [0; 2\pi[^K} \|\mathbf{r} - \mathbf{u}\|^2 \text{ s.t. } \mathbf{Ax} = \mathbf{u} \odot e^{i\theta}. \tag{3.1.3}$$

From (3.1.3) one can derive the augmented Lagrangian:

$$\mathcal{L}(\mathbf{x}, \mathbf{u}, \theta, \lambda) = \|\mathbf{r} - \mathbf{u}\|^2 + \Re\left(\lambda^H (\mathbf{Ax} - \mathbf{u} \odot e^{i\theta})\right) + \frac{\rho}{2} \|\mathbf{Ax} - \mathbf{u} \odot e^{i\theta}\|^2, \tag{3.1.4}$$

where λ is the vector of the Lagrange multipliers corresponding to the constraint $\mathbf{Ax} = \mathbf{u} \odot e^{i\theta}$ and ρ is the penalty parameter. By minimizing (3.1.4), the authors derive the ADMM update rules detailed in Algorithm 9.

In [157], Wen et al. address PR as a feasibility problem. Instead of (3.1.2), they consider the following formulation:

$$\text{find} \quad \mathbf{x} \in \mathbb{C}^L \quad \text{s.t.} \quad |\mathbf{Ax}| = \mathbf{r} \text{ and } \mathbf{x} \in \mathcal{S}_0, \tag{3.1.5}$$

where $\mathcal{S}_0$ is the set of signals respecting an additional constraint (in optics, a typical constraint is that the signal is real-valued and non-negative). The ADMM updates are specified in Algorithm 10 with $\mathcal{P}_{\mathcal{S}_0}$ being the projection operator on $\mathcal{S}_0$.

Other optimization algorithms have also been considered for phase retrieval. For instance, *majorization-minimization* is used in [121] with

Algorithm 9 : ADMM ([84])

Parameters : $\rho \in \mathbb{R}$

Initialize $\boldsymbol{\lambda}_0 \in \mathbb{C}^L$, $\mathbf{x}_0 \in \mathbb{C}^K$.
while *iterate* **do**
  $\mathbf{u}_{t+1} = \frac{\rho|\mathbf{A}\mathbf{x}_t + \rho^{-1}\boldsymbol{\lambda}_t| + 2\mathbf{r}}{\rho+2}$
  $\theta_{t+1} = \frac{\mathbf{A}\mathbf{x}_t + \rho^{-1}\boldsymbol{\lambda}_t}{|\mathbf{A}\mathbf{x}_t + \rho^{-1}\boldsymbol{\lambda}_t|}$
  $\mathbf{x}_{t+1} = \mathbf{A}^\dagger \left(\mathbf{u}_{t+1} \odot e^{i\theta_{t+1}} - \frac{\boldsymbol{\lambda}_t}{\rho} \right)$
  $\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \rho \left(\mathbf{A}\mathbf{x}_{t+1} - \mathbf{u}_{t+1} \odot e^{i\theta_{t+1}} \right)$
end

Algorithm 10 : ADMM ([157])

Initialize $\boldsymbol{\lambda}_0$, $\mathbf{u}_0 \in \mathbb{C}^K$.
while *iterate* **do**
  $\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{S}_0}(\mathbf{u}_t - \boldsymbol{\lambda}_t)$
  $\mathbf{u}_{t+1} = \mathbf{A}^\dagger \left(\mathbf{r} \odot \frac{\mathbf{A}(\mathbf{x}_{t+1} + \boldsymbol{\lambda}_t)}{|\mathbf{A}(\mathbf{x}_{t+1} + \boldsymbol{\lambda}_t)|} \right)$
  $\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \rho(\mathbf{x}_{t+1} - \mathbf{u}_{t+1})$
end

four different algorithmic variants. The *Douglas-Rachford Splitting* is also examined in [21, 22] and has been shown to be equivalent to ADMM in the context of phase retrieval [41].

Convex methods

More recently, methods proposing convex relaxations of the PR problem have been presented. *PhaseLift* [17] is the precursor of these techniques. This algorithm results from the following observation concerning power measurements:

$$\forall k = 1, \ldots, K; \quad (|\mathbf{A}\mathbf{x}|^2)_k = \mathrm{Tr}(\mathbf{a}_k \mathbf{a}_k^H \mathbf{x}\mathbf{x}^H), \tag{3.1.6}$$

where $\mathbf{a}_k$ is the k-th row of $\mathbf{A}$ and Tr is the matrix trace operator. With the rank-one matrix $\mathbf{X} = \mathbf{x}\mathbf{x}^H$, the phase retrieval problem now writes:

$$\begin{aligned} \text{find} \quad & \mathbf{X}, \\ \text{s.t.} \quad \forall k; \quad & \mathrm{Tr}(\mathbf{a}_k \mathbf{a}_k^H \mathbf{X}) = r_k, \\ & \mathbf{X} \succeq 0, \\ & \mathrm{rank}(\mathbf{X}) = 1. \end{aligned} \tag{3.1.7}$$

Equation (3.1.7) can equivalently be written as the following rank minimization problem:

$$\begin{aligned} \text{minimize} \quad & \mathrm{rank}(\mathbf{X}), \\ \text{s.t.} \quad \forall k; \quad & \mathrm{Tr}(\mathbf{a}_k \mathbf{a}_k^H \mathbf{X}) = r_k, \\ & \mathbf{X} \succeq 0. \end{aligned} \tag{3.1.8}$$

As the rank minimization problem (3.1.8) is NP-hard, it is relaxed into the following trace minimization problem:

$$\begin{aligned} &\text{minimize} && \operatorname{Tr}(\mathbf{X}), \\ &\text{s.t.} && \forall k; \quad \operatorname{Tr}(\mathbf{a}_k \mathbf{a}_k^H \mathbf{X}) = r_k, \\ & && \mathbf{X} \succeq 0. \end{aligned} \tag{3.1.9}$$

This last problem is a convex semidefinite program (SDP) and can be tackled via a wide range of SDP solvers. When the measurement operator is a collection of random Gaussian vectors, PhaseLift has been shown to recover the unknown signal with high probability [17].

PhaseCut [151] is another semidefinite relaxation of the phase retrieval problem. It starts by separating the amplitude and phase variables in the optimization problem with $d = 1$. PR now writes:

$$\min_{\substack{\mathbf{u} \in \mathbb{C}^K, |\mathbf{u}| = \mathbf{1} \\ \mathbf{x} \in \mathbb{C}^L}} \|\mathbf{A}\mathbf{x} - \mathbf{r} \odot \mathbf{u}\|^2. \tag{3.1.10}$$

As (3.1.10) can be solved explicitly in $\mathbf{x}$ when the phase vector $|\mathbf{u}|$ is fixed, the problem writes:

$$\begin{aligned} &\text{minimize} && \mathbf{u}^H \mathbf{M} \mathbf{u} \\ &\text{s.t.} && |\mathbf{u}| = \mathbf{1}, \end{aligned} \tag{3.1.11}$$

with $\mathbf{M} = \operatorname{diag}(\mathbf{r})(\mathbf{I} - \mathbf{A}\mathbf{A}^H)\operatorname{diag}(\mathbf{r})$. With the rank-one matrix $\mathbf{U} = \mathbf{u}\mathbf{u}^H$, (3.1.11) is equivalent to:

$$\begin{aligned} &\text{minimize} && \operatorname{Tr}(\mathbf{U}\mathbf{M}), \\ &\text{s.t.} && \operatorname{diag}(\mathbf{U}) = \mathbf{1}, \\ & && \mathbf{U} \succeq 0, \\ & && \operatorname{rank}(\mathbf{U}) = 1. \end{aligned} \tag{3.1.12}$$

Similarly than with the Maxcut relaxation [54, 110], (3.1.12) is relaxed in the following SDP:

$$\begin{aligned} &\text{minimize} && \operatorname{Tr}(\mathbf{U}\mathbf{M}), \\ &\text{s.t.} && \operatorname{diag}(\mathbf{U}) = \mathbf{1}, \\ & && \mathbf{U} \succeq 0. \end{aligned} \tag{3.1.13}$$

Problem (3.1.13) then can be solved with the help of a SDP solver.

In the following, methods to tackle PR with audio signals are detailed. Convex methods are rarely employed in this context due to their increased dimensionality and the large dimension of audio signals. Nonconvex methods, on the other hand, are rather common.

3.2 PHASE RETRIEVAL IN AUDIO

3.2.1 *Context and applications*

As many audio signal processing techniques operate on the spectrogram (or other phaseless time-frequency representations), phase retrieval is essential to reconstruct waveforms. Therefore, STFT magnitude or power measurements are usually considered and PR applies in a variety of tasks.

In speech enhancement, noise reduction algorithms are frequently formulated with spectrograms and discard the phase. Several methods have been proposed in order to estimate the phase components of the enhanced spectrogram. They achieve significantly better reconstruction performance than the methods ignoring the phase estimation [52, 73, 107].

PR is also useful with source separation (*cf.* Section 3.2.3), as most algorithms operate on phaseless time-frequency representations. A typical framework begins by estimating the magnitude spectrograms of the different sources of the mixture signal. Then, the signals are reconstructed from the source spectrogram estimates. The phase of the mixture is typically used as a phase estimate and the inverse STFT is computed. Several works observed that using PR approaches to estimate a proper phase results in better separation performance [91, 156, 158].

Audio restoration is another application for PR. In [74], the authors address audio inpainting in the time-frequency domain. They make use of PR algorithms to estimate the phase of the missing TF coefficients, whereas usual approaches only address magnitude inpainting. In [90], declicking (i. e. , removing noise on short time periods) is addressed. The proposed method based on PR outperforms traditional restoration methods.

3.2.2 *Specific algorithms*

There are numerous PR algorithms in the literature that use the STFT operator and audio signals. They can be classified in two categories : iterative nonconvex optimization methods and model-based methods. The first ones are typically the counterparts of the PR algorithms detailed in Section 3.1 and were developped concurrently with them. Convex methods on the other hand, are rarely used in audio applications due to their increased computational cost: these methods do not scale to high dimension. Model-based methods design specific algorithms based on the structure of the STFT and audio signals.

Algorithm 11 : Griffin-Lim algorithm

Initialize $\tilde{\mathbf{x}}_0$.
while *iterate* **do**
  $\tilde{\mathbf{x}}_{t+1} := \mathbf{A}\mathbf{A}^{\mathsf{H}}\left(\mathbf{r} \odot \frac{\tilde{\mathbf{x}}_t}{|\tilde{\mathbf{x}}_t|}\right)$
end
Return $\mathbf{A}^{\mathsf{H}}\tilde{\mathbf{x}}_T$.

Iterative methods: Griffin-Lim algorithm and variants

The *Griffin-Lim algorithm* (GLA) [59] addresses the PR problem with magnitude spectrograms as measurements. This seminal work is the counterpart to the Error Reduction algorithm (*cf.* Section 3.1.2) and it is still widely used in the audio community. GLA alternates projections on $\mathcal{M}$, the set of time-frequency coefficients whose magnitude is equal to the observed measurements, and $\mathcal{C}$, the set of consistent coefficients (i. e. , that correspond to the STFT of time-domain signals). More formally, these sets write:

$$\mathcal{M} = \{\tilde{\mathbf{x}} \in \mathbb{C}^K \,|\, |\tilde{\mathbf{x}}| = \mathbf{r}\}, \tag{3.2.1}$$

$$\mathcal{C} = \{\tilde{\mathbf{x}} \in \mathbb{C}^K \,|\, \tilde{\mathbf{x}} = \mathbf{A}\mathbf{A}^{\dagger}\tilde{\mathbf{x}}\}. \tag{3.2.2}$$

With the assumption of self-duality of the window used in the STFT, we have $\mathbf{A}^{\dagger} = \mathbf{A}^{\mathsf{H}}$ and $\mathbf{A}^{\mathsf{H}}\mathbf{A} = \mathbf{I}_L$. The projections on the two sets then write:

$$\mathcal{P}_{\mathcal{M}}(\tilde{\mathbf{x}}) = \mathbf{r} \odot \frac{\tilde{\mathbf{x}}}{|\tilde{\mathbf{x}}|}, \tag{3.2.3}$$

$$\mathcal{P}_{\mathcal{C}}(\tilde{\mathbf{x}}) = \mathbf{A}\mathbf{A}^{\mathsf{H}}\tilde{\mathbf{x}}. \tag{3.2.4}$$

Although $\mathcal{M}$ is not a subspace and is not convex, $\mathcal{P}_{\mathcal{M}}$ is usually called a projection since it maps an element of $\mathbb{C}^K$ to its unique closest element in $\mathcal{M}$. Alternating these projections after an initialization with random phase results in GLA, which is proved to converge to a critical point of the quadratic loss (3.1.2). GLA is displayed in Algorithm 11.

An accelerated version of GLA, termed *Fast Griffin-Lim algorithm* (FGLA), is proposed in [115] with a momentum strategy with constant acceleration parameter. It is shown experimentally to reach lower local minima of (3.1), yet without theoretical convergence guarantee. FGLA iterations are detailed in Algorithm 12, where ξ is the acceleration parameter.

In [50], the authors propose *Real-Time Iterative Spectrogram Inversion* (RTISI), a real-time variant of GLA. It consists of applying GLA iterations frame by frame while only considering the previous frames.

Algorithm 12 : Fast Griffin-Lim algorithm

Parameters : $\xi \in [0, 1]$

Initialize $\tilde{\mathbf{x}}_0$.
while *iterate* **do**
  $\tilde{\mathbf{u}}_{t+1} := \mathbf{A}\mathbf{A}^H \left(\mathbf{r} \odot \frac{\tilde{\mathbf{x}}_t}{|\tilde{\mathbf{x}}_t|} \right)$
  $\tilde{\mathbf{x}}_{t+1} := \tilde{\mathbf{u}}_{t+1} + \xi(\tilde{\mathbf{u}}_{t+1} - \tilde{\mathbf{u}}_t)$
end
Return $\mathbf{A}^H \tilde{\mathbf{x}}_T$.

Algorithm 13 : Griffin-Lim like phase recovery via ADMM (GLADMM)

Initialize $\tilde{\mathbf{x}}_0$.
$\tilde{\mathbf{u}}_0 = \tilde{\mathbf{x}}_0$
$\boldsymbol{\lambda}_0 = \mathbf{0}$
while *iterate* **do**
  $\tilde{\mathbf{x}}_{t+1} = \mathcal{P}_{\mathcal{M}}(\tilde{\mathbf{u}}_t - \boldsymbol{\lambda}_t)$
  $\tilde{\mathbf{u}}_{t+1} = \mathcal{P}_{\mathcal{C}}(\tilde{\mathbf{x}}_{t+1} + \boldsymbol{\lambda}_t)$
  $\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{u}}_{t+1}$
end
Return $\mathbf{A}^H \tilde{\mathbf{x}}_T$.

In [162], an extension to RTISI called *RTISI with Look-Ahead* (RTISI-LA) is proposed. The authors consider a few of the future frames and the previous frames in RTISI, which significantly improves the reconstruction performance.

A new PR optimization criterion based on consistency is proposed in [76, 78]. The authors also introduce an algorithm to minimize this new criterion, based on local approximations. They notice a connection between their scheme and GLA, which is the alternated minimization of an auxiliary function constructed with the proposed consistency-based loss.

The authors of [99] draw on GLA to propose an ADMM scheme based on the following feasibility problem:

$$\underset{\tilde{\mathbf{x}} \in \mathbb{C}^K}{\text{minimize}} \quad \chi_{\mathcal{M}}(\tilde{\mathbf{x}}) + \chi_{\mathcal{C}}(\tilde{\mathbf{x}}), \tag{3.2.5}$$

with $\chi_{\mathcal{M}}, \chi_{\mathcal{C}}$ respectively the indicator functions of sets $\mathcal{M}$ ad $\mathcal{C}$. Writing ADMM with (3.2.5) results in the algorithm entitled *Griffin-Lim like phase recovery via ADMM* (GLADMM) and detailed in Algorithm 13.

When $\boldsymbol{\lambda}_t = 0$, the GLADMM iteration is identical to a Griffin-Lim algorithm step.

Model-based methods

Various PR methods make use of the properties of the STFT operator and audio signals. Several of them are based on the phase derivatives and magnitude relations introduced in [118]. These relations only hold theoretically when the STFT operator is applied to continuous functions and the analysis window is an infinite-support Gaussian function. With $\mathcal{A}_g$ denoting such an operator, x being a real continuous function, m, n respectively denoting frequency and time, the relations write:

$$\frac{\partial \angle \mathcal{A}_g x}{\partial m}(m, n) = -\gamma \frac{\partial}{\partial n} \log(|\mathcal{A}_g x|(m, n)), \tag{3.2.6}$$

$$\frac{\partial \angle \mathcal{A}_g x}{\partial n}(m, n) = \gamma^{-1} \frac{\partial}{\partial m} \log(|\mathcal{A}_g x|(m, n)) + 2\pi m. \tag{3.2.7}$$

Here, γ denotes the time-frequency support ratio of the Gaussian window g, defined as follows:

$$g(n) = \left(\frac{\gamma}{2}\right)^{-\frac{1}{4}} e^{-\pi \frac{n^2}{4}}. \tag{3.2.8}$$

With (3.2.6) and (3.2.7), the phase gradient can be defined as:

$$\nabla \angle \mathcal{A}_g x(m, n) = \begin{pmatrix} -\gamma \frac{\partial}{\partial n} \log(|\mathcal{A}_g x|(m, n)) \\ \gamma^{-1} \frac{\partial}{\partial m} \log(|\mathcal{A}_g x|(m, n)) + 2\pi m \end{pmatrix}. \tag{3.2.9}$$

In theory, knowing the original phase of a single time-frequency coefficient is enough to reconstruct the phase of the entire spectrogram by integrating the phase gradient (3.2.9) [119]. However, these relations do not stand in practice with discrete signals and finite-support windows. In [119], the authors still make use of them through approximations with a proposed algorithm entitled *Phase Gradient Heap Integration*. Their experimental work results in good reconstruction results, at the expense of theoretical guarantees.

Other methods take interest in signal structure using sinusoidal models: sine waves phase can be reconstructed from the phase of a known coefficient as it grows linearly in time. The *phase unwrapping algorithm* [90] proposes estimating first the instantaneous frequency of the sinusoidal components via quadratic spectrum interpolation. The phase of onsets is assumed to be known or estimated by another algorithm, and the phase is then linearly unwrapped [89]. A similar method is developed by the *Single Pass Spectrogram Inversion algorithm* [6] (SPSI), which detects peaks using quadratic interpolation and accumulates phase linearly.

3.2.3 *Phase retrieval for audio source separation*

Problem formulation

The source separation problem consists in estimating the source signals $\mathbf{x}^{(c)}$ composing a mixture signal $\mathbf{x}$. We consider a linear and instantaneous mixture model:

$$\mathbf{x} = \sum_{c=1}^{C} \mathbf{x}^{(c)}. \tag{3.2.10}$$

Even though more intricate models include gain weights, delays or convolutions (e. g. , for dereverberation applications), these will not be considered in this thesis. As the STFT is linear, the source separation problem with model (3.2.10) can be expressed in the time-frequency domain as well:

$$\text{find } \{\tilde{\mathbf{x}}^{(c)}\}_{c=1}^{C} \quad \text{s.t.} \quad \tilde{\mathbf{x}} = \sum_{c=1}^{C} \tilde{\mathbf{x}}^{(c)}, \tag{3.2.11}$$

with $\tilde{\mathbf{x}}$, $\tilde{\mathbf{x}}^{(c)}$ denoting the STFT of the mixture and the sources, respectively.

(3.2.11) is often solved via time-frequency masking. This technique consists in estimating nonnegative masks $\mathbf{b}^{(c)}$ that are multiplied by the STFT of the mixture to produce complex source estimates $\hat{\tilde{\mathbf{x}}}^{(c)}$:

$$\hat{\tilde{\mathbf{x}}}^{(c)} = \mathbf{b}^{(c)} \odot \tilde{\mathbf{x}}. \tag{3.2.12}$$

A wide range of approaches to mask estimation exist in the literature. Among them, *Wiener filtering* considers the following masks:

$$\mathbf{b}^{(c)} = \frac{\hat{\mathbf{r}}^{(c)}}{\sum_{c'=1}^{C} |\hat{\mathbf{r}}^{(c')}|^2}, \tag{3.2.13}$$

which are optimal in the sense of the mean square error by design and with $\hat{\mathbf{r}}^{(c)}$ denoting a power spectrogram estimate. Methods to estimate $\hat{\mathbf{r}}^{(c)}$ include nonnegative matrix factorization [42], kernel methods [87] and deep learning [153]. The latter can also be used directly to mask estimation without the structure hypothesis (3.2.13).

However, using nonnegative-valued masks for time-frequency masking implies that the phase of the source STFT estimate is equal to the phase of the mixture STFT. This assumption is known to be incorrect when the sources overlap in the time-frequency domain and to result in low-quality source estimates after the inverse STFT is applied. To address this issue, several methods aim at estimating the phase. *Consistent Wiener filtering* [79, 80] introduces a framework accounting for the consistency of the source estimates in masking. This approach is extended in [93], where the phase is considered to be non-uniform, in contrast to previous Wiener filtering methods. Other methods make use of signal models in this context [20, 91].

Algorithm 14 : Multiple input spectrogram inversion algorithm

Initialize $\mathbf{x}_0^{(1)}, \ldots, \mathbf{x}_0^{(C)}$.
while *iterate* **do**
  for $c = 1$ **to** C **do**
    $\mathbf{y}_{t+1}^{(c)} = \mathbf{A}^\dagger \left(\mathbf{r}^{(c)} \odot \frac{\mathbf{A}\mathbf{x}_t^{(c)}}{|\mathbf{A}\mathbf{x}_t^{(c)}|} \right)$
  end
  for $c = 1$ **to** C **do**
    $\mathbf{x}_{t+1}^{(c)} = \mathbf{y}_{t+1}^{(c)} + \frac{1}{C} \left(\mathbf{x} - \sum_{c'=1}^{C} \mathbf{y}_{t+1}^{(c')} \right)$
  end
end

The multiple input spectrogram inversion algorithm

The *multiple input spectrogram inversion algorithm* [63] (MISI) is an extension to GLA [59] to perform PR with multiple measurements in the context of source separation. With the mixture $\mathbf{x}$ and spectrogram estimates $\mathbf{r}^{(c)}$ of C sources, the problem can be formulated as follows:

$$\min_{\{\mathbf{x}^{(c)} \in \mathbb{R}^L\}_{c=1}^{C}} \sum_{c=1}^{C} \left\| \mathbf{r}^{(c)} - |\mathbf{A}\mathbf{x}^{(c)}| \right\|^2 \text{ s.t. } \sum_{c=1}^{C} \mathbf{x}^{(c)} = \mathbf{x}. \tag{3.2.14}$$

MISI solves (3.2.14) with the iterations detailed in Algorithm 14.

MISI iterations begin with a Griffin-Lim step on each source, followed by a distribution of the mixture error on the different source estimates to enforce the mixture constraint. It was also demonstrated in [96, 154] that this algorithm is related to the majorization-minimization approach and converges.

3.3 PHASE RETRIEVAL WITH DEEP LEARNING

In recent years, many learning-based methods for tackling inverse problems in audio signal processing have been developed. We present hereafter techniques for phase retrieval using deep neural networks (DNN).

In order to tackle audio PR, Arik et al. propose in [2] a novel convolutional neural network architecture, that is trained to synthesize audio from an observed spectrogram. The CNN includes several so-called "heads", which are subblocks of the network working in parallel with the same spectrogram input. The heads include the same transposed convolution layers, with different upsamplig factors. The loss function used is a sum of known losses in audio such as spec-

tral convergence or ℓ^1 distance over log-magnitude spectrograms (c.f. Section 2.1.2). [134] proposes a feed-froward DNN architecture to reconstruct phase from magnitude spectrogram. The authors train the network with cosines losses on phase and group delay, which statistically implies that the phase data follows a von Mises distribution. This work is extended to the modeling of group delay from magnitude spectrograms in [135].

In [98], the authors estimate the instantaneous frequency and the group delay from a magnitude spectrogram via two DNNs. The two networks are trained with a cosine loss. The authors reconstruct the phase with a recurrent unwrapping algorithm. A similar strategy is presented in [136], where two DNNs are also used to estimate the phase derivatives. The networks are trained with a cosine loss with biquadratic regularization. The phase is reconstructed via integration over several paths.

Several deep learning based methods for phase retrieval also take inspiration from unfolding. In [158], the authors unfold the MISI algorithm (see Section 3.2.3) for phase reconstruction in the context of source separation. The unfolded network includes parameterized layers emulating the STFT and the iSTFT, which enables the proposed architecture to learn audio representations from the data. [100] proposes a deep architecture for PR that is inspired by the unfolding of the Griffin-Lim algorithm. Every layer includes a GLA iteration and a denoising sublayer, which is a convolutional neural network trained separately to estimate the residual error.

Part II

PHASE RETRIEVAL WITH BREGMAN DIVERGENCES

4

PHASE RETRIEVAL FROM SINGLE SPECTROGRAM

The contributions of this chapter have been published in [144].

4.1 Introduction 45
4.2 Phase retrieval with Bregman divergences 46
4.2.1 Problem setting 46
4.2.2 Accelerated gradient descent 46
4.2.3 ADMM algorithm 48
4.2.4 Implementation details 50
4.3 Numerical experiments 51
4.3.1 Experimental setup 52
4.3.2 Phase retrieval from exact spectrograms 54
4.3.3 Phase retrieval from modified spectrograms . . 55
4.4 Strategies for the choice of the gradient step size 60
4.4.1 Experimental setup 60
4.4.2 Results 61
4.5 Conclusion 63

4.1 INTRODUCTION

Phase retrieval is commonly formulated as a nonconvex minimization problem involving a quadratic cost function, as follows:

$$\min_{\mathbf{x}\in\mathbb{C}^L} \quad \|\mathbf{r} - |\mathbf{A}\mathbf{x}|^d\|^2. \tag{4.1.1}$$

Problem (4.1.1) may be tackled with conventional optimization algorithms such as gradient descent [16, 33], alternating projections [44, 51], majorization-minimization [121], alternating direction method of multipliers [84, 157], and leveraging the structure of time-frequency measurements [9, 116]. A presentation of several of these algorithms is detailed in Chapter 3, Section 3.1.

Even though a considerable amount of research has been conducted to tackle the PR problem as described in (4.1.1), such an approach suffers from one drawback when it comes to audio. Indeed, it is well established that the quadratic cost is not the best-suited metric for evaluating discrepancies in the time-frequency domain. For instance, it does not properly characterize the perceptually-related properties of audio such as its large dynamic range [55].

As such, we propose to replace hereafter the quadratic cost function in (4.1.1) by alternative divergences which are more appropriate for audio signal processing [55]. We consider Bregman divergences, a

family of cost functions which encompasses the beta-divergence [23, 66] and some of its well-known special cases, the Kullback-Leibler (KL) and Itakura-Saito (IS) divergences. These are acknowledged for their superior performance in nonnegative audio spectral decomposition [42, 95, 128, 149], audio inpainting [77], and music analysis [65, 146]. Besides, these divergences naturally arise from a statistical perspective (*cf.* Section 2.2.3). For instance, minimizing the KL divergence between an observed and an estimated spectrogram assumes that the observations follow a Poisson model. Similarly, minimizing the IS divergence implies a multiplicative Gamma noise model [128]. In order to be as general as possible, we consider any nonnegative power d (we do not restrict to either 1 nor 2) and we account for the fact that these divergences are not symmetric with respect to their input parameters in general, which actually leads to tackling two different problems. To optimize the resulting objective, we derive two algorithms, based on accelerated gradient descent [117] and ADMM [12].

The chapter is organized as follows. Section 4.2 describes the PR problem extended to Bregman divergences and details the two proposed algorithms. Section 4.3 presents the experimental works with audio signal recovery applications. Section 4.4 discusses strategies to choose optimally the step size with the proposed gradient algorithm. Finally, Section 4.5 draws some concluding remarks.

4.2 PHASE RETRIEVAL WITH BREGMAN DIVERGENCES

4.2.1 *Problem setting*

We propose to generalize the problem (4.1.1) by substituting the quadratic cost by a Bregman divergence. As it is not necessarily symmetric with respect to its input arguments, we will tackle the two following formulations of the problem, with $\mathcal{D}_\psi$ denoting the Bregman divergence with generating function ψ:

$$\min_{\mathbf{x}\in\mathbb{C}^L} \overrightarrow{J}(\mathbf{x}) := \mathcal{D}_\psi(\mathbf{r}\,|\,|\mathbf{A}\mathbf{x}|^d), \tag{4.2.1}$$

$$\min_{\mathbf{x}\in\mathbb{C}^L} \overleftarrow{J}(\mathbf{x}) := \mathcal{D}_\psi(|\mathbf{A}\mathbf{x}|^d\,|\,\mathbf{r}). \tag{4.2.2}$$

We will refer to problems (4.2.1) and (4.2.2) as "right PR" and "left PR" respectively.

4.2.2 *Accelerated gradient descent*

Similarly to [16], we first propose a Wirtinger gradient descent algorithm to minimize the objective functions defined in (4.2.1) and (4.2.2).

The gradients of a Bregman divergence with respect to its first and second arguments are given by

$$\nabla_{\mathbf{z}} \mathcal{D}_\psi(\mathbf{y}\,|\,\mathbf{z}) = \psi''(\mathbf{z}) \odot (\mathbf{z} - \mathbf{y}), \tag{4.2.3}$$
$$\nabla_{\mathbf{y}} \mathcal{D}_\psi(\mathbf{y}\,|\,\mathbf{z}) = \psi'(\mathbf{y}) - \psi'(\mathbf{z}). \tag{4.2.4}$$

Using the chain rule [35], we obtain:

$$\nabla \overrightarrow{J}(\mathbf{x}) = (\nabla |\mathbf{A}\mathbf{x}|^d)^H [\psi''(|\mathbf{A}\mathbf{x}|^d) \odot (|\mathbf{A}\mathbf{x}|^d - \mathbf{r})], \tag{4.2.5}$$
$$\nabla \overleftarrow{J}(\mathbf{x}) = (\nabla |\mathbf{A}\mathbf{x}|^d)^H [\psi'(|\mathbf{A}\mathbf{x}|^d) - \psi'(\mathbf{r})], \tag{4.2.6}$$

where the derivative ψ' and second-derivative ψ'' are applied entry-wise and $\nabla |\mathbf{A}\mathbf{x}|^d$ denotes the Jacobian of the multivariate function $\mathbf{x} \to |\mathbf{A}\mathbf{x}|^d$ (the Jacobian being the extension of the gradient for multivariate functions, we may use the same notation ∇).[1] Using differentiation rules for element-wise matrix operations [35], we have:

$$\nabla |\mathbf{A}\mathbf{x}|^d = \frac{d}{2} \operatorname{diag}(|\mathbf{A}\mathbf{x}|^{d-2} \odot (\mathbf{A}\mathbf{x}))\mathbf{A}. \tag{4.2.7}$$

Expressions of ψ, ψ' and ψ'' for some typical Bregman divergences are given in Chapter 2, Table 1.

We rewrite the gradients (4.2.5) and (4.2.6) in the following compact form:

$$\nabla J(\mathbf{x}) = (\nabla |\mathbf{A}\mathbf{x}|^d)^H \, \mathbf{g}_\psi, \tag{4.2.8}$$

where J can be either $\overrightarrow{J}$ or $\overleftarrow{J}$ and

$$\text{for "right" PR, } \mathbf{g}_\psi = \psi''(|\mathbf{A}\mathbf{x}|^d) \odot (|\mathbf{A}\mathbf{x}|^d - \mathbf{r}), \tag{4.2.9}$$
$$\text{for "left" PR, } \mathbf{g}_\psi = \psi'(|\mathbf{A}\mathbf{x}|^d) - \psi'(\mathbf{r}). \tag{4.2.10}$$

As such and together with (4.2.7), we obtain:

$$\nabla J(\mathbf{x}) = \frac{d}{2} \mathbf{A}^H \left[|\mathbf{A}\mathbf{x}|^{d-2} \odot (\mathbf{A}\mathbf{x}) \odot \mathbf{g}_\psi \right]. \tag{4.2.11}$$

Using a constant step-size μ, our generic gradient algorithm writes:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \mu \nabla J(\mathbf{x}_t). \tag{4.2.12}$$

Similarly as in FGLA [115], we furthermore use an acceleration scheme [117] resulting in the following updates:

$$\begin{aligned} \mathbf{q}_{t+1} &= \mathbf{x}_t - \mu \nabla J(\mathbf{x}_t), \\ \mathbf{x}_{t+1} &= \mathbf{q}_{t+1} + \xi(\mathbf{q}_{t+1} - \mathbf{q}_t), \end{aligned} \tag{4.2.13}$$

1 Note that the gradient is not properly defined in some cases when one or more coefficients of $\mathbf{A}\mathbf{x}$ are zero-valued. We present in Appendix 4.5 a detailed and rigorous treatment of this potential issue.

where ξ is the acceleration parameter.
Remark: When considering a quadratic cost (i.e., $\psi(z) = \frac{1}{2}z^2$), problems (4.1.1), (4.2.1) and (4.2.2) become equivalent. In particular, when $d = 1$, both gradients (4.2.5)-(4.2.6) write:

$$\nabla J(\mathbf{x}) = \mathbf{x} - \mathbf{A}^H \left(\mathbf{r} \odot \frac{\mathbf{A}\mathbf{x}}{|\mathbf{A}\mathbf{x}|} \right). \tag{4.2.14}$$

Gradient descent with step size equal to 1 thus yields:

$$\mathbf{x}_{t+1} = \mathbf{A}^H \left(\mathbf{r} \odot \frac{\mathbf{A}\mathbf{x}_t}{|\mathbf{A}\mathbf{x}_t|} \right), \tag{4.2.15}$$

which is nothing but the GLA update given by alternating the projections in Section 3.2. This shows that GLA can be seen as a gradient descent applied to the PR problem (4.1.1).

4.2.3 *ADMM algorithm*

In a similar fashion as in [84], we propose to reformulate PR with Bregman divergences as a constrained problem. We detail hereafter the left PR problem, and a similar derivation can be conducted for its right counterpart. The problem rewrites:

$$\min_{\mathbf{x}\in\mathbb{C}^L, \mathbf{u}\in\mathbb{R}_+^K, \theta\in[0;2\pi[^K} \mathcal{D}_\psi(\mathbf{r}\,|\,\mathbf{u}) \text{ s. t. } (\mathbf{A}\mathbf{x})^d = \mathbf{u} \odot e^{i\theta}, \tag{4.2.16}$$

from which we obtain the augmented Lagrangian:

$$\begin{aligned}\mathcal{L}(\mathbf{x}, \mathbf{u}, \theta, \boldsymbol{\lambda}) =& \mathcal{D}_\psi(\mathbf{r}\,|\,\mathbf{u}) + \Re\left(\boldsymbol{\lambda}^H((\mathbf{A}\mathbf{x})^d - \mathbf{u} \odot e^{i\theta})\right) \\ &+ \frac{\rho}{2}\left\|(\mathbf{A}\mathbf{x})^d - \mathbf{u} \odot e^{i\theta}\right\|^2,\end{aligned} \tag{4.2.17}$$

where ρ is the penalty parameter. The first step of our ADMM algorithm consists in updating the values of $\mathbf{u}$ and θ given $\mathbf{x}_t$ and $\boldsymbol{\lambda}_t$:

$$\{\mathbf{u}_{t+1}, \theta_{t+1}\} = \underset{\mathbf{u}\geqslant 0, \theta}{\operatorname{argmin}}\, \mathcal{L}(\mathbf{x}_t, \mathbf{u}, \theta, \boldsymbol{\lambda}_t). \tag{4.2.18}$$

To that end, we first rewrite $\mathcal{L}$ as:

$$\mathcal{L}(\mathbf{x}, \mathbf{u}, \theta, \boldsymbol{\lambda}) = \mathcal{D}_\psi(\mathbf{r}\,|\,\mathbf{u}) + \frac{\rho}{2}\left\|(\mathbf{A}\mathbf{x})^d + \frac{\boldsymbol{\lambda}}{\rho} - \mathbf{u} \odot e^{i\theta}\right\|^2 - \frac{1}{2\rho}\|\boldsymbol{\lambda}\|^2. \tag{4.2.19}$$

Therefore, problem (4.2.18) can be equivalently formulated as:

$$\{\mathbf{u}_{t+1}, \theta_{t+1}\} = \underset{\mathbf{u}\geqslant 0, \theta}{\operatorname{argmin}}\, \mathcal{D}_\psi(\mathbf{r}\,|\,\mathbf{u}) + \frac{\rho}{2}\|\mathbf{h}_t - \mathbf{u} \odot e^{i\theta}\|^2, \tag{4.2.20}$$

with:

$$\mathbf{h}_t = (\mathbf{A}\mathbf{x}_t)^d + \frac{\boldsymbol{\lambda}_t}{\rho}. \tag{4.2.21}$$

With $\mathbf{u}$ fixed, the second term in (4.2.20) is minimized when the phase of $\mathbf{h}_t$ is equal to θ. Thus, θ is updated as follows:

$$\theta_{t+1} = \angle \mathbf{h}_t. \tag{4.2.22}$$

The problem in $\mathbf{u}$ can then be formulated as:

$$\mathbf{u}_{t+1} = \underset{\mathbf{u}\geqslant 0}{\operatorname{argmin}} \quad \mathcal{D}_\psi(\mathbf{r}\,|\,\mathbf{u}) + \frac{\rho}{2}\||\mathbf{h}_t| - \mathbf{u}\|^2. \tag{4.2.23}$$

As shown in Appendix 4.5, the minimization problem (4.2.23) remains unchanged when the positivity constraint on $\mathbf{u}$ is disregarded. The $\mathbf{u}$ update can therefore be written

$$\mathbf{u}_{t+1} = \operatorname{prox}_{\rho^{-1}\mathcal{D}_\psi(\mathbf{r}\,|\,\cdot)}(|\mathbf{h}_t|). \tag{4.2.24}$$

The expressions of prox_f for some of the divergences considered in our experiments are given in Table 2.

The second step of our ADMM algorithm consists in updating the value of $\mathbf{x}$:

$$\mathbf{x}_{t+1} = \underset{\mathbf{x}\in\mathbb{C}^L}{\operatorname{argmin}}\, \mathcal{L}(\mathbf{x}, \mathbf{u}_{t+1}, \theta_{t+1}, \boldsymbol{\lambda}_t). \tag{4.2.25}$$

Since only the second term on the right-hand side of (4.2.19) depends on $\mathbf{x}$, this problem rewrites:

$$\mathbf{x}_{t+1} = \underset{\mathbf{x}\in\mathbb{C}^L}{\operatorname{argmin}} \left\| (\mathbf{A}\mathbf{x})^d - \mathbf{u}_{t+1} \odot e^{i\theta_{t+1}} + \frac{\boldsymbol{\lambda}_t}{\rho} \right\|^2, \tag{4.2.26}$$

which is a least-squares problem with the following closed-form solution:

$$\mathbf{x}_{t+1} = \mathbf{A}^H \big(\mathbf{u}_{t+1} \odot e^{i\theta_{t+1}} - \frac{\boldsymbol{\lambda}_t}{\rho}\big)^{1/d}. \tag{4.2.27}$$

The final step of our ADMM algorithm consists in updating the Lagrange multipliers $\boldsymbol{\lambda}$, as follows:

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \rho(\mathbf{A}\mathbf{x}_{t+1} - \mathbf{u}_{t+1} \odot e^{i\theta_{t+1}}). \tag{4.2.28}$$

The whole ADMM procedure then consists in iteratively applying the updates given by (4.2.24), (4.2.27) and (4.2.28).

The derivation of the updates for the left PR problem is similar, and the resulting algorithm is unchanged, except for the update of $\mathbf{u}$ in (4.2.24), which becomes:

$$\mathbf{u}_{t+1} = \operatorname{prox}_{\rho^{-1}\mathcal{D}_\psi(\cdot\,|\,\mathbf{r})}(|\mathbf{h}_t|). \tag{4.2.29}$$

Algorithm 15 : Accelerated gradient descent for PR with the Bregman divergence.

Inputs: Measurements $\mathbf{R} \in \mathbb{R}_+^{M \times N}$, initial phase $\boldsymbol{\phi}_0 \in [0, 2\pi[_+^{M \times N}$, step size μ and acceleration parameter ξ.
Initialization:
$\mathbf{X} = \mathbf{R}^{1/d} \odot e^{i\boldsymbol{\phi}_0}$
$\mathbf{x} = \text{iSTFT}(\mathbf{X})$
$\mathbf{q}_{\text{old}} = 0$
while *stopping criteria not reached* **do**
  $\mathbf{X} = \text{STFT}(\mathbf{x})$
  if *PR left* **then**
    $\mathbf{G}_\psi = \psi'(|\mathbf{X}|^d) - \psi'(\mathbf{R})$
  else if *PR right* **then**
    $\mathbf{G}_\psi = \psi''(|\mathbf{X}|^d) \odot (|\mathbf{X}|^d - \mathbf{R})$
  $\mathbf{q} = \mathbf{x} - \mu \frac{d}{2} \text{iSTFT}(\mathbf{X} \odot |\mathbf{X}|^{d-2} \odot \mathbf{G}_\psi)$
  $\mathbf{x} = \mathbf{q} + \xi(\mathbf{q} - \mathbf{q}_{\text{old}})$
  $\mathbf{q}_{\text{old}} = \mathbf{q}$
end
Output: $\mathbf{x}$

4.2.4 *Implementation details*

We have presented gradient descent and ADMM algorithms for phase retrieval in the general case. We now address some specificities of audio signal recovery from a phaseless spectrogram, i.e., when $\mathbf{A}$ is the STFT matrix and $\mathbf{x}$ is real-valued. The STFT matrix $\mathbf{A}$ and its inverse are large structured matrices that allow for fast implementations of matrix-vector products of the forms $\mathbf{Ax}$ and $\mathbf{A}^H\mathbf{y}$ based on the fast Fourier transform [13, 129]. In that setting, one handles time-frequency matrices of size $M \times N$, where M is the number of frequency channels and N the number of time frames, rather than vectors of size $K = MN$. As such, we provide in Algorithms 15 and 16 the pseudo-code for practical implementation of our accelerated gradient and ADMM algorithms, respectively, in the time-frequency audio recovery setting.

For generality, we assumed $\mathbf{x} \in \mathbb{C}^L$ in the previous sections. However, audio signals are real-valued and this deserves some comments. As shown in Appendix 4.5, the iterates $\mathbf{x}_t$ remain real-valued under specific conditions. In a nutshell, a signal is real-valued if and only if its STFT $\mathbf{X} \in \mathbb{C}^{M \times N}$ is frequency-Hermitian, that is:

$$X(m, n) = X(M - m, n)^*. \tag{4.2.30}$$

When $\mathbf{R}$ is the spectrogram of a real-valued signal and when Algorithms 15 and 16 are initialized with a frequency-Hermitian matrix $\mathbf{X}$, all the time-frequency matrices involved in the updates re-

Algorithm 16 : ADMM for PR with the Bregman divergence.

Inputs: Measurements $\mathbf{R} \in \mathbb{R}_+^{M \times N}$, initial phase $\boldsymbol{\Phi}_0 \in [0, 2\pi[_+^{M \times N}$ and augmentation parameter ρ.
Initialization:
$\mathbf{X} = \mathbf{R}^{1/d} \odot e^{i\boldsymbol{\Phi}_0}$
$\mathbf{x} = \text{iSTFT}(\mathbf{X})$
$\boldsymbol{\Lambda} = 0$
while *stopping criteria not reached* **do**
  $\mathbf{X} = \text{STFT}(\mathbf{x})$
  $\mathbf{H} = \mathbf{X}^d + \frac{1}{\rho}\boldsymbol{\Lambda}$
  $\boldsymbol{\Theta} = \angle\mathbf{H}$
  if *PR left* **then**
    $\mathbf{U} = \text{prox}_{\rho^{-1}\mathcal{D}_\psi(\cdot\,|\,\mathbf{r})}(|\mathbf{H}|)$
  else if *PR right* **then**
    $\mathbf{U} = \text{prox}_{\rho^{-1}\mathcal{D}_\psi(\mathbf{r}\,|\,\cdot)}(|\mathbf{H}|)$
  $\mathbf{x} = \text{iSTFT}((\mathbf{U} \odot e^{i\boldsymbol{\Theta}} - \frac{1}{\rho}\boldsymbol{\Lambda})^{1/d})$
  $\boldsymbol{\Lambda} = \boldsymbol{\Lambda} + \rho(\text{STFT}(\mathbf{x}) - \mathbf{U} \odot e^{i\boldsymbol{\Theta}})$
end
Output: $\mathbf{x}$

main frequency-Hermitian (because operations only involve sums and element-wise products with frequency-Hermitian matrices). This in turn ensures that the variable $\mathbf{x}$ remains real-valued. As such, the STFT and inverse STFT (iSTFT) operations in Algorithms 15 and 16 need only return/process the first $\lfloor \frac{M}{2} \rfloor + 1$ frequency channels (usually termed "positive frequencies"), as customary with real-valued signals [130].

More rigorously, we may also re-derive our gradient and ADMM algorithms for $\mathbf{x} \in \mathbb{R}^L$, using real-valued differentiation instead of Wirtinger gradients (and involving the real and imaginary parts of $\mathbf{A}$ in the objective function). This is addressed in Appendix 4.5 which shows that we indeed obtain the same algorithms.

4.3 NUMERICAL EXPERIMENTS

In this section, we conduct experiments on PR tasks. We first assess the potential of the proposed algorithms for recovering signals from exact (i.e., non-modified) spectrograms. Then, we consider a PR task from modified spectrograms, as often encountered in audio applications. In the spirit or reproducible research, the code related to those experiments is available online.[2] We also provide audio examples of reconstructed signals.[3]

2 https://github.com/phvial/PRBregDiv
3 https://magronp.github.io/demos/jstsp21.html

4.3.1 *Experimental setup*

Data

As acoustic material, we use two corpora in our experiments. The first one, referred to as "speech", is composed of 100 utterances taken randomly from the TIMIT database [49]. The second one, referred to as "music", comprises 100 snippets from the Free Music Archive dataset [34]. Signals from the "speech" corpus are 16-bits WAV files and signals from the "music" corpus are MP3 files encoded at 320 kbps. All audio excerpts are single-channel, sampled at 22,050 Hz and cropped to be 2 seconds long. The STFT is computed with a 1024 samples-long (46 ms) self-dual sine window [129] (leading to an effective number of 513 frequency bins) and 50 % overlap. We used the `librosa` Python package [102].

Methods

PR is conducted using the algorithms presented in Section 4.2 under different settings as described next.

PROPOSED GRADIENT DESCENT ALGORITHM We experimented the accelerated gradient algorithm described in Algorithm 15 in the following settings:

- KL ($\beta = 1$) for the "right" and "left" problems with $d \in \{1, 2\}$,
- $\beta = 0.5$ for the "right" and "left" problems and with $d \in \{1, 2\}$,
- IS ($\beta = 0$) for the "right" problem with $d = 2$,
- quadratic cost ($\beta = 2$) with $d \in \{1, 2\}$ (in that case the "right" and "left" problems are equivalent).

The "right" problems with KL, $d = 1$ on the one hand, and IS, $d = 2$ on the other hand, correspond to standard designs in NMF [43, 128]. The value $\beta = 0.5$ with either $d = 1$ or 2 has also been advocated in various papers, e.g., [146].

The algorithms are used with constant step-size μ and acceleration parameter $\xi = 0.99$ (like in [115]). The step-size is empirically set to the largest negative power of 10 enabling convergence for each cost function and value of d in the setting of the experiments reported in Sections 4.3.2 and 4.3.3. A summary of the parameter configurations and choice of cost functions is given in Table 1.

Table 1: Summary of setups considered in the experiments with their parameters (cost function, exponent d, type of algorithm and hyperparameter). Each setup is described by a code that follows this format: *algorithm·cost·direction-d*.

Algorithm	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent
Cost	β-div. ($\beta = 0.5$)	β-div. ($\beta = 0.5$)	Kullback-Leibler	Kullback-Leibler	Quadratic	Itakura-Saito
Direction	right	left	right	left	N/A	right
d	1	1	1	1	1	2
Hyperparameters	$\mu = 10^{-1}$	$\mu = 10^{-6}$	$\mu = 10^{-4}$	$\mu = 10^{-2}$	$\mu = 10^{-1}$	$\mu = 10^{-7}$
Associated code	G·05·R1	G·05·L1	G·KL·R1	G·KL·L1	G·QD·1	G·IS·R2
Algorithm	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent [16]	ADMM
Cost	β-div. ($\beta = 0.5$)	β-div. ($\beta = 0.5$)	Kullback-Leibler	Kullback-Leibler	Quadratic	Itakura-Saito
Direction	right	left	right	left	N/A	left
d	2	2	2	2	2	1
Hyperparameters	$\mu = 10^{-3}$	$\mu = 10^{-6}$	$\mu = 10^{-1}$	$\mu = 10^{-3}$	$\mu = 10^{-5}$	$\rho = 10^{-1}$
Associated code	G·05·R2	G·05·L2	G·KL·R2	G·KL·L2	G·QD·2	A·IS·L1
Algorithm	ADMM	ADMM [84]	Griffin-Lim [59]	Fast Griffin-Lim [115]	GLADMM [99]	Initialisation
Cost	Kullback-Leibler	Quadratic	(Quadratic)	(Quadratic)	(Indicator function)	N/A
Direction	left	N/A	N/A	N/A	N/A	N/A
d	1	1	1	1	1	N/A
Hyperparameter	$\rho = 10^{-1}$	$\rho = 10^{-1}$	N/A	N/A	N/A	N/A
Associated code	A·KL·L1	A·QD·1	GLA	FGLA	GLADMM	INIT

PROPOSED ADMM ALGORITHM Applicability of ADMM is more limited than with gradient descent because it requires the expression of the proximal operators (4.2.24) and (4.2.29). We here consider the quadratic cost and "left" KL and IS problems. We set $d = 1$ and $\rho = 1$, which corresponds to the setting used by Liang et al. [84] for the quadratic cost (which thus falls as a special case of our setting).

OTHER BASELINES AND PARAMETERS The previous algorithms are compared with the following other baselines: GLA, FGLA and GLADMM, presented in Section 3.2 and which use $d = 1$.

All the algorithms (baseline and contributed) are run for 2500 iterations (which ensures that convergence is observed for all algorithms) and initialized with the same uniform random phase (a single realization was used for each excerpt).

Evaluation metrics

The reconstruction quality is evaluated in the time-frequency domain with the standard spectral convergence (SC) metric (*cf.* Chapter 2). Additionally, for the "speech" corpus, we consider the short-term objective intelligibility (STOI) measure [133], which is computed with the `PySTOI` library [113]. This score has been used in several PR-related papers such as [97, 99].

SC is directly related to the PR quadratic cost problem (4.1.1), formulated in the time-frequency domain, while the perceptual STOI is more related to the applicative needs. In both cases, the higher the value, the better the performance.

Let us note that alternative evaluation metrics exist, such as PESQ [124] or PEMO-Q [70], which are tailored for perceptually assessing speech quality. We also computed those, and the obtained results were overall consistent with the STOI measure, up to some minor differences. Besides, it has been shown that these measures are strongly correlated with STOI, notably in PR-related tasks [101].

4.3.2 *Phase retrieval from exact spectrograms*

First, we consider a PR task conducted on exact spectrograms. In this setting, measurements are directly obtained from the ground truth signals $\mathbf{x}^\star$, such that $\mathbf{r} = |\mathbf{A}\mathbf{x}^\star|^d$. These measurements $\mathbf{r}$ are then fed as inputs to the algorithms described in 4.3.1.

The results on the "speech" and "music" corpora are presented in Figures 1 and 2 respectively, from which overall similar conclusions can be drawn.

The best performance in terms of SC are achieved by GLADMM and other ADMM algorithms, which are closely followed by algorithms optimizing the quadratic cost with $d = 1$. Note however that the advantage of quadratic cost-based algorithms against competing

methods is less significant in terms of STOI. As recalled above, SC is directly related to the PR problem with quadratic cost (4.1.1) and consequently favors algorithms that directly tackle this problem.

A performance similar to that of quadratic cost-based algorithms is reached by some of the proposed alternative methods, such as the ADMM algorithms A·IS·L1 and A·KL·L1 and the gradient descent algorithms GD·05·R1, GD·KL·R2 and GD·KL·L2, in terms of SC and STOI (note that for the latter, the best performing methods exhibit a lower variance than the others). This outlines the potential of using alternative divergences rather than the quadratic cost function.

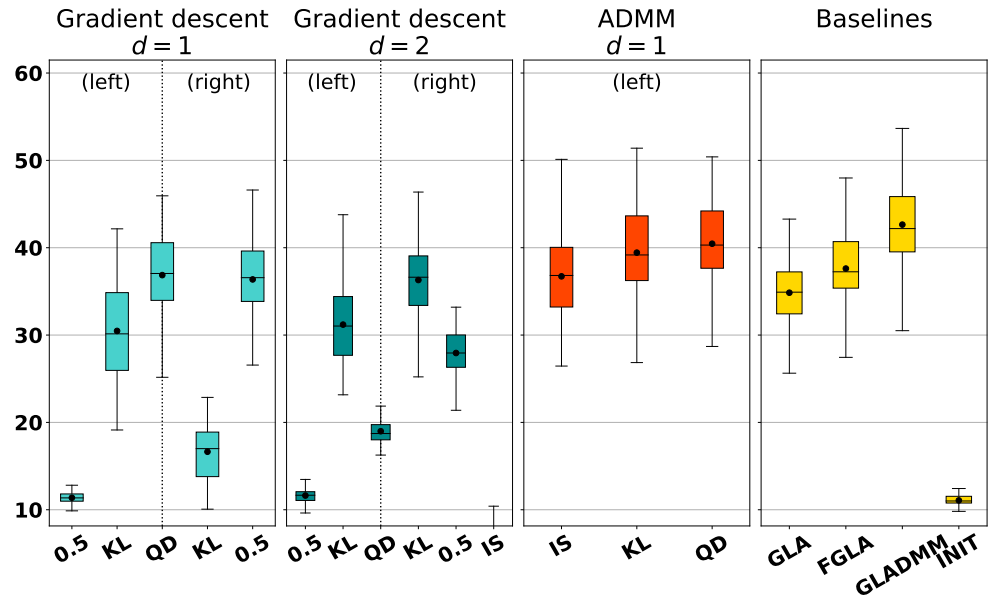
Besides, we observe that the performance of these methods depend on a variety of factors. For instance, the difference between the performance reached by GD·KL·R2 and GD·KL·R1, or between GD·QD·1 and GD·QD·2 (for both metrics and corpora) outlines the impact of d on the reconstruction quality. Likewise, considering a "left" rather than a "right" PR problem may yield very different results (see for instance the two corresponding gradient algorithms with $\beta = 0.5$ and $d = 1$).

Finally, for a given problem, the impact of the optimization strategy (i.e., ADMM vs. gradient descent) depends on the nature of the signals. For the "speech" corpus, ADMM algorithms (for KL and the quadratic cost) perform mildly better than their gradient algorithms respective counterparts. However, for the "music" corpus, A·KL·L1 significantly outperforms GD·KL·L1 in terms of SC.

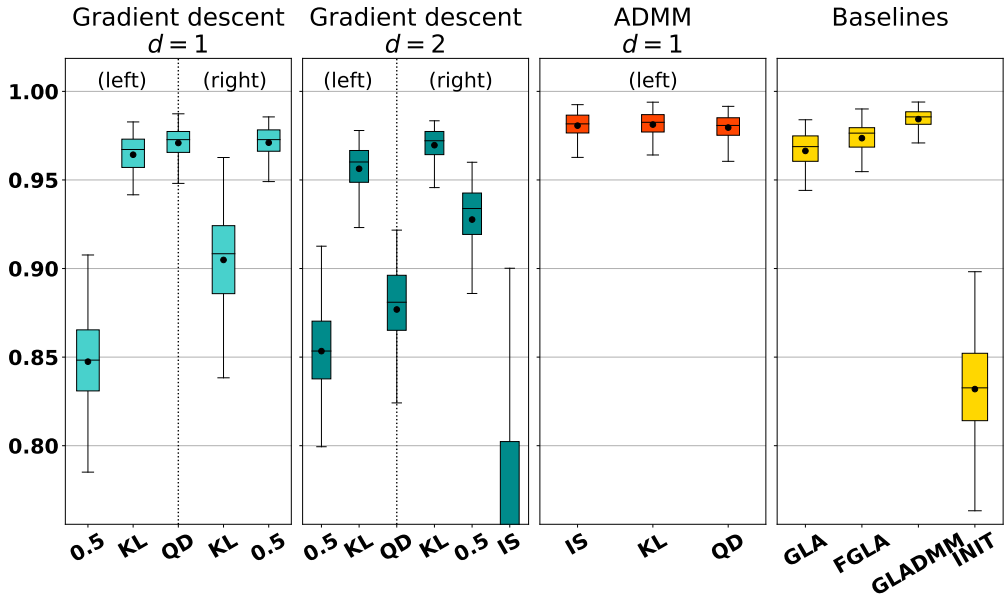
To summarize, when retrieving a signal from an exact spectrogram, GLADMM and quadratic-minimizing algorithms (with $d = 1$) seem to perform best. Some alternative methods yield competitive results, but require to carefully adapt the setup (power d, cost β, "right" or "left" formulation) and optimization strategy (ADMM vs. gradient descent) to the problem, as well as considering the nature of the signals (speech or music). Note than when the data $\mathbf{r}$ is an exact spectrogram (i.e., $\mathbf{r} = |\mathbf{A}\mathbf{x}^\star|^d$), the cost functions (4.2.1) and (4.2.2) share the same minimum value 0 and global solution $\mathbf{x}^\star$ (up to ambiguities) for all ψ. This may explain why the somehow easier-to-optimize quadratic cost performs well in this scenario. However this result is to be contrasted when using modified spectrograms, as shown next.

4.3.3 *Phase retrieval from modified spectrograms*

We now consider a PR task from modified spectrograms. In audio restoration applications such as source separation [148], audio inpainting [1] or time-stretching [36], the spectrogram that results from diverse operations does not necessarily correspond to the magnitude of the STFT of a signal. We propose to simulate this situation by modifying the spectrograms as in [99]. We add synthetic Gaussian white noise in the time domain to each excerpt in the "speech" corpus. For



(a) Spectral convergence



(b) STOI

Figure 1: Performance of PR from exact spectrograms for the "speech" corpus, measured with the SC (top) and STOI (bottom). Higher values correspond to a better performance. Turquoise, orange and yellow respectively denote gradient descent algorithms, ADMM algorithms and GLA-like algorithms. The boxes indicate the two middle quartiles among the ten excerpts, the middle bar is for the median, the dot for the mean, and the whiskers denote the extremal values.

each signal, the noise variance is chosen so that the input signal-to-noise ratio (SNR) takes the following values: 10, 0, −10, and −20 dB. We then apply an oracle Wiener filter [86] to the mixture in the STFT

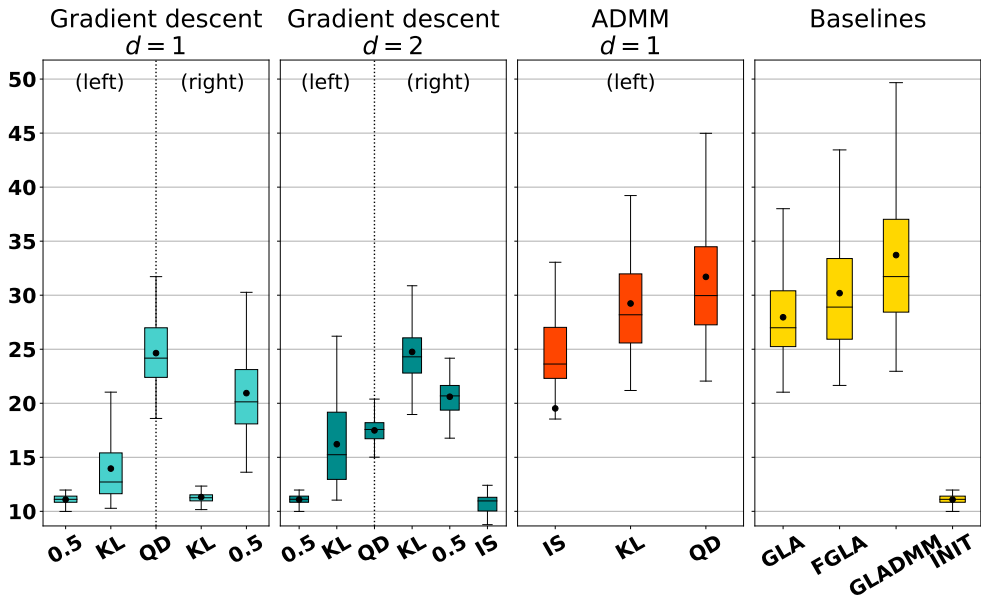


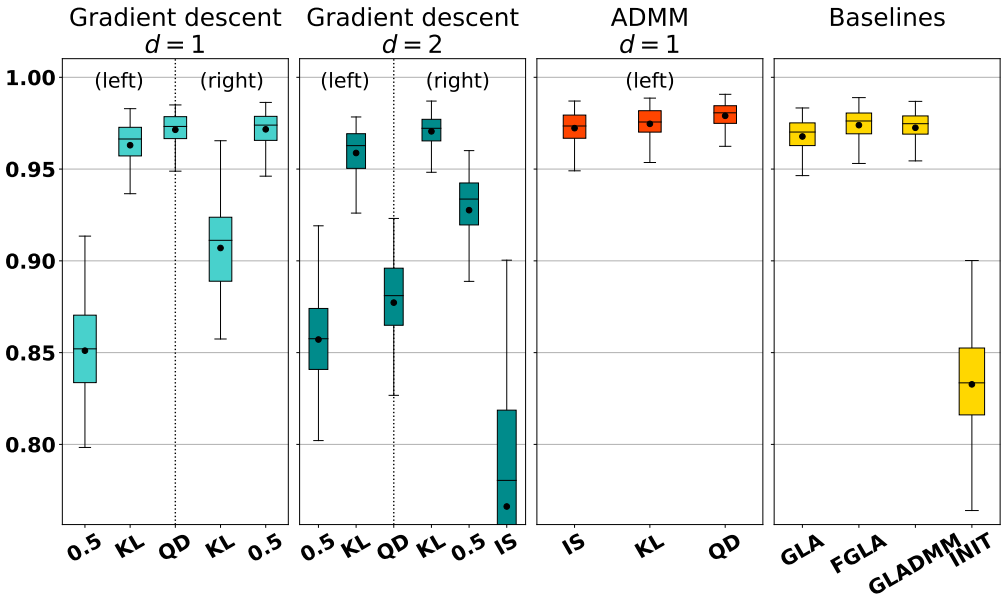
Figure 2: Performance of PR from exact spectrograms for the "music" corpus measured with the SC.

domain: this yields a restored (even though inconsistent) magnitude spectrogram $\mathbf{r}$ which corresponds to realistic applications [99]. To further recover a time-domain signal, we apply the considered PR algorithms to this modified spectrogram.

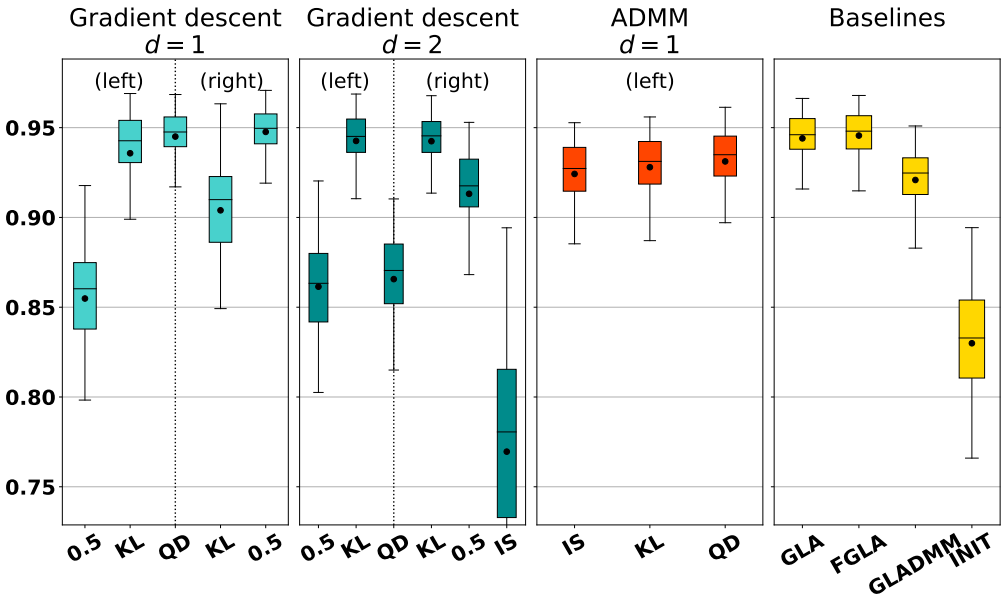
The results in terms of STOI are presented in Figure 3. Note that we do not report the SC, since it is mostly impacted by the spectrogram deformation procedure, not by the subsequent PR task. In that experiment, we observed some convergence problems at low input SNRs for several algorithms and signals: in these few cases, the gradient step size (which we recall was tuned using exact spectrogram data) was reduced by a factor 1/10.

At high input SNR (0 to 10 dB), we observe a similar trend than in the previous experiment: GLADMM and quadratic cost-based algorithms (with $d = 1$) enable better reconstruction in terms of STOI than other categories of algorithms. This confirms that such algorithms are appropriate for addressing the PR problem when the spectrograms are either exact or slightly degraded.

However, we observe a different trend at lower input SNRs, where some algorithms based on alternative cost functions exhibit more robustness to the spectrogram degradation caused by the input noise. For instance, while ADMM algorithms overall perform best at 10 dB input SNR, they are outperformed by alternative algorithms such as GD·KL·L2 at lower input SNRs (from 0 to -20 dB). Similarly and contrary to the case of high input SNRs, GLADMM is outperformed by other GL-based or ADMM algorithms. Interestingly, GD·05·L1 and GD·KL·R1 exhibit the most robust behavior among gradient algorithms with $d = 1$: while they perform worst at 10 dB input SNR, they

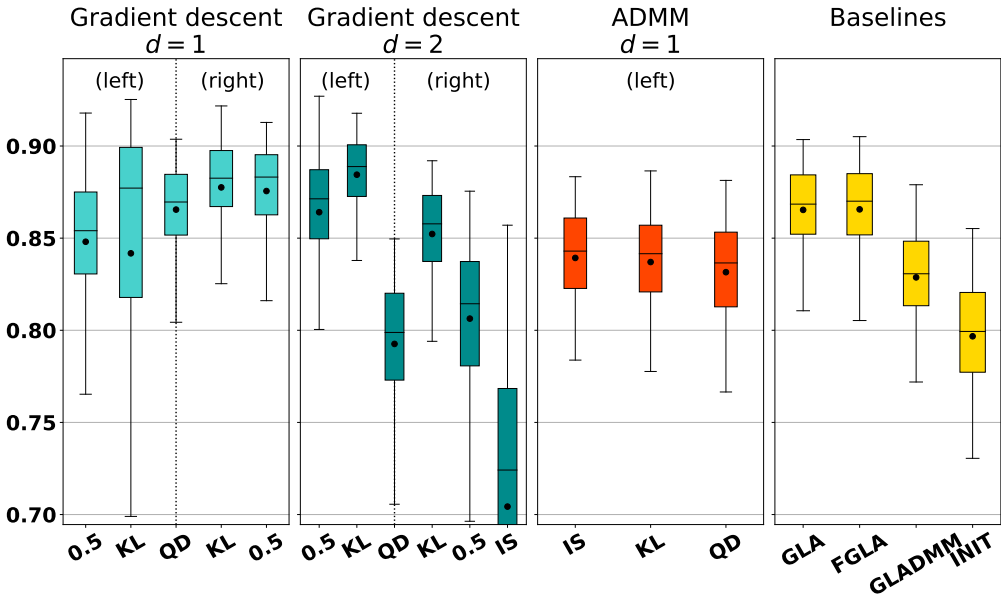


(a) +10 dB

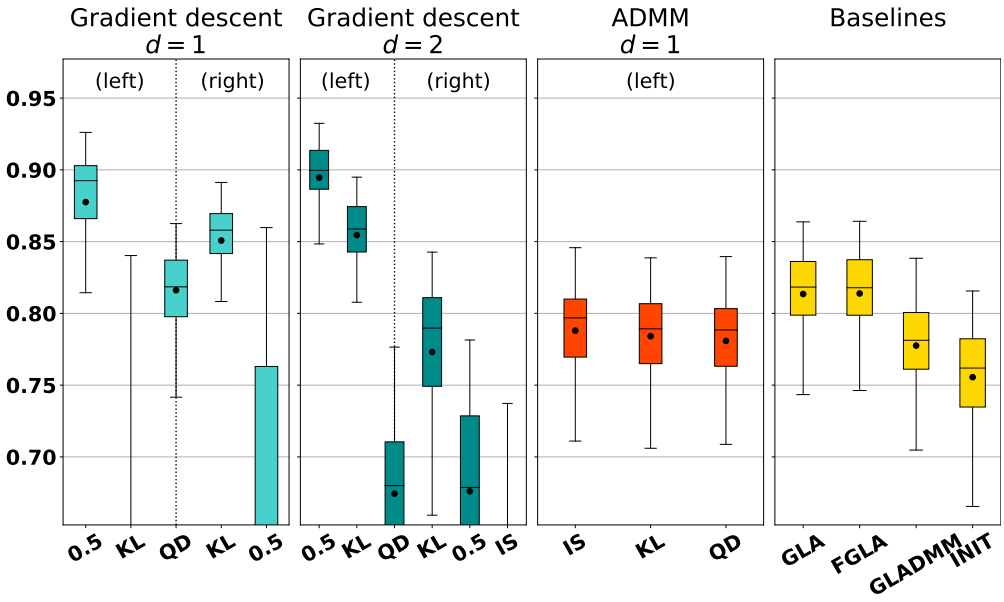


(b) 0 dB

Figure 3: STOI for PR from modified speech spectrograms at various input SNRs.



(c) −10 dB



(d) −20 dB

Figure 3: STOI for PR from modified speech spectrograms at various input SNRs.

actually achieve the best performance at -20 dB input SNR. On the other hand, the performance of several algorithms, such as GD·KL·R2, significantly drops when more noise is added, while they perform relatively well at high input SNRs. Finally, even though the best performance at very low input SNR is achieved by GD·05·L2, GD·KL·L2 might still be a good candidate for the task at hand: indeed, at input SNRs from 10 to -10 dB, it outperforms its counterpart using $\beta = 0.5$, and thus exhibits a more stable performance regarding the level of input noise.

Overall, the usefulness of PR with Bregman divergences is revealed when the spectrograms are highly corrupted (that is, when they are retrieved using a Wiener filter from very noisy observations), as quadratic cost-based algorithms are outperformed by alternative cost-based algorithms in such a scenario. This might be explained by the ability of such divergences to better model and account for the nature of this destructive noise.

4.4 STRATEGIES FOR THE CHOICE OF THE GRADIENT STEP SIZE

In the experimental work detailed in the previous section, the gradient algorithm considered uses a fixed step size chosen empirically. Such a strategy is suboptimal: if the step size is too large, the algorithm diverges; if it is too small, the algorithm is excessively slow to reach a solution. Moreover, a step size that is appropriate for a given divergence may be suboptimal with another: no value is optimal for all. Finally, because PR is nonconvex, adjusting the step size may result in different solutions as different stationary points of the problem may be reached. Therefore, more intricate methods shall be considered to tune the step size of the considered gradient algorithms.

4.4.1 *Experimental setup*

Succinct experimental work is here conducted to study gradient algorithms with varying step size on PR tasks. A single utterance of the TIMIT database is here considered to generate nonlinear observations. 100 initializations of the gradient algorithms are obtained with random phases. The STFT parameters are identical as in the experiments of the previous section. We compare the gradient algorithms detailed in the following with PR tasks conducted on exact spectrograms. Both "left" and "right" PR problems are considered with $d \in \{1, 2\}$ and KL, beta-divergence with $\beta = 0.5$ and quadratic cost functions. All the algorithms are run for 2500 iterations and performance is evaluated in terms of SC and STOI.

We consider the following strategies for chosing the gradient step.

GRADIENT DESCENT WITH FIXED STEP SIZE This method considers a constant step size. This context is similar to the previous experiments, and the step size is chosen according to the values of Table 1. This algorithm is studied without and with acceleration, which are respectively denoted as 'GD" and "GDA".

GRADIENT DESCENT WITH NON-MONOTONIC BACKTRACKING In this method, the considered step size is varying following a non-monotonic backtracking rule as in [60]. At each iteration, the gradient step is adjusted with a scalar factor ν_{BT} until the current cost is smaller than at least one of the last t_w past cost values.

$$\text{While } J(\mathbf{x}_{t+1}) \geqslant \max_{t-t_w<j<t} \left\{J(\mathbf{x}_j)\right\} - \frac{\mu_t}{2}\|\nabla J(\mathbf{x}_t)\|^2: \tag{4.4.1}$$

$$\mu_t \leftarrow \nu_{BT}\mu_t, \tag{4.4.2}$$

$$\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \mu_t \nabla J(\mathbf{x}_t). \tag{4.4.3}$$

Here, the initial step size is chosen equal to 10 times the value in Table 1 and $\nu_{BT} = 0.5$. The number of past cost values regarded with the rule is $t_w = 100$ and a maximal number of backtracking iterations for each gradient iteration is fixed to 15. This algorithm is studied without and with acceleration, which are respectively denoted as "BT" and "BTA".

GRADIENT DESCENT WITH BARZILAI-BORWEIN STEP AND NON-MONOTONIC BACKTRACKING This method also considers a varying step size following a non-monotonic backtracking rule, which is initialized with a Barzilai-Borwein step (*cf.* Chapter 2). The "long" step is here considered:

$$\mu_t = \frac{\|\mathbf{x}_{t-1} - \mathbf{x}_{t-2}\|^2}{\langle \nabla J(\mathbf{x}_{t-1}) - \nabla J(\mathbf{x}_{t-2}) \,;\, \mathbf{x}_{t-1} - \mathbf{x}_{t-2}\rangle}. \tag{4.4.4}$$

As the Barzilai-Borwein method can lead to negative step sizes in locally-nonconvex regions, we adopt a strategy similar to [19]: a large step size is chosen when $\mu_t < 0$. The step size is then refined with the non-monotonic backtracking rule as previously. This algorithm is denoted as "BB+BT". It is not presented here with acceleration, as it empirically shown to lead the algorithm to diverge in early experiments.

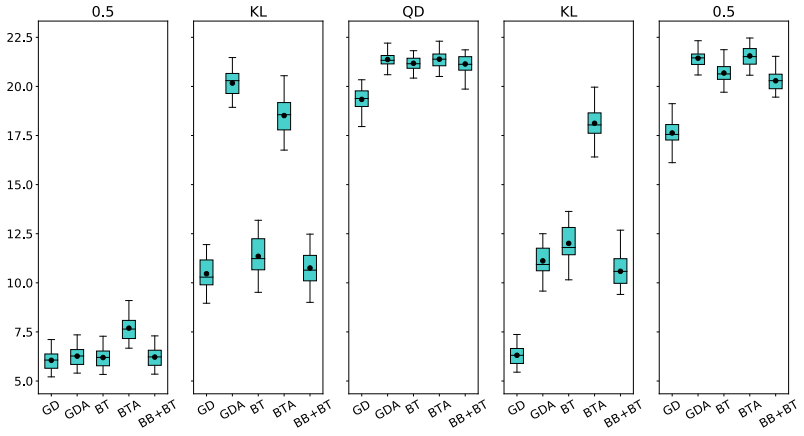
4.4.2 *Results*

First, we present the results of the PR task with magnitude measurements in Figure 4. Generally speaking, the non-accelerated methods are outperformed by their accelerated counterparts with all the considered cost functions and according to both SC and STOI. When compared to the gradient descent algorithm with constant step size,

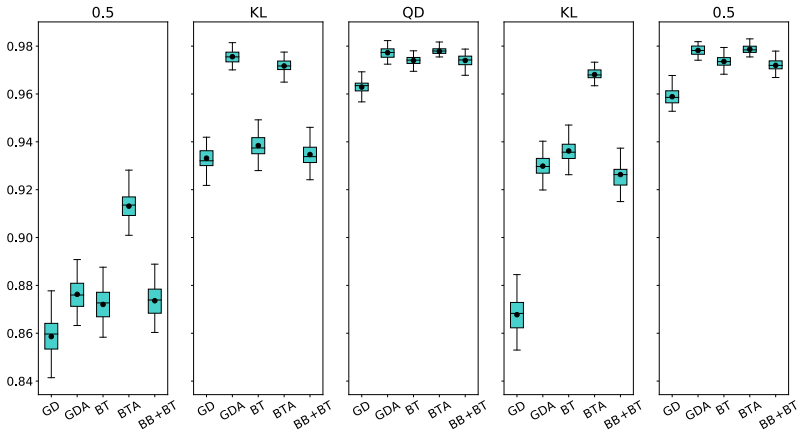
the gradient algorithm with non-monotonic backtracking also shows higher SC and STOI for all cost functions. With acceleration, two different trends can be observed when comparing GDA and BTA. First, where GDA performs poorly ("left" beta-divergence and "right" KL), BTA brings a significant improvement in performance in terms of both SC and STOI. Second, when GDA performs best ("left" KL, quadratic and "right" beta-divergence), we do not observe a significant difference in performance between GDA and BTA. Backtracking with the Barzilai-Borwein step shows little difference in SC and STOI when compared with non-monotonic backtracking or compares poorly. When the step size is tuned more accurately, quadratic cost and "right" beta-divergence lead to the best reconstruction results. However, gradient descent algorithms with divergences that showed poor performance in the experiments of Section 4.3 see their results greatly improved with a proper step size tuning strategy. They might reveal more potential in further research.

The performance of the PR task with power measurements is presented in Figure 5. In a similar fashion than with magnitude spectrograms, the accelerated methods outperform their non-accelerated counterparts in terms of SC and STOI in every setting. BT leads to a significant performance improvement when compared to GD. However, the algorithm appears to diverge with "right" KL. The improvement observed with BT also holds with accelerated methods as BTA significantly outperforms GDA with regards to both metrics. In contrast with PR from magnitude spectrograms, BB+BT shows to be more interesting with power spectrograms. It compares favorably to BT in terms of both SC and STOI and with all settings. Moreover, it converges with "right" KL. With their finest step size tuning method, both "left" and "right" KL, and "right" beta-divergence with $\beta = 0.5$ lead to better reconstruction performance when compared to phase retrieval with quadratic cost function.

In this section, we addressed tuning methods for the step size of the gradient descent algorithm for phase retrieval with Bregman divergences. Experimental work assessed the interest of non-monotonic backtracking strategies in this context, which improved the reconstruction performance with and without acceleration. With power spectrograms, the use of a Barzilai-Borwein step as initialization of the backtracking iterations leads to enhance the performance measured by SC and STOI, but this was however not observed with PR from magnitude spectrograms. Generally speaking, the scenarii that already lead to good results in the previous experiments are only slightly improved with these tuning methods. The divergences that compared favorably to the quadratic cost keep their advantage with proper step size tuning. However, poorly performing settings see their reconstruction performance greatly improved by such methods. Using a non-monotonic backtracking strategy therefore comes with a



(a) Spectral convergence



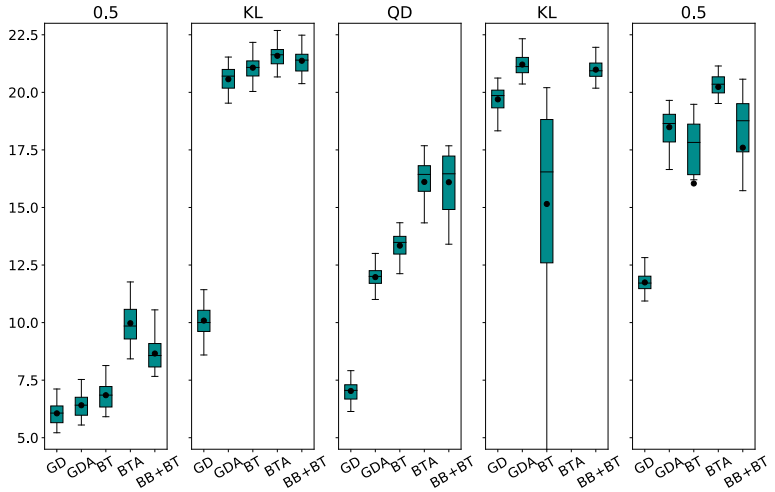
(b) STOI

Figure 4: Performance of PR from magnitude spectrograms ($d = 1$), measured with the SC (top) and STOI (bottom). The considered cost functions are from left to right : "left" beta-divergence, "left" Kullback-Leibler, Quadratic, "right" Kullback-Leibler, "right" beta-divergence. $\beta = 0.5$ with the beta-divergences.

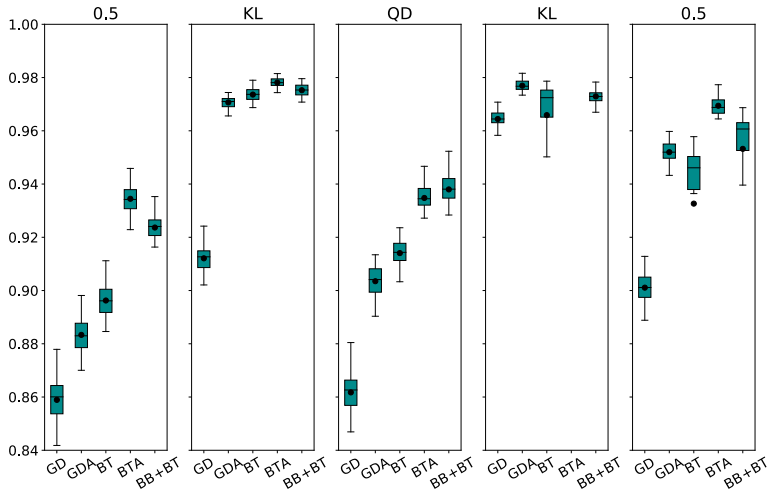
twofold advantage: first, it eases the choice of the step size parameter, which is critical with regards to reconstruction performance. Second, it leads to an improvement of the methods' results and confirms the potential of alternative divergences in phase retrieval.

4.5 CONCLUSION

We have considered the problem of PR when the quadratic cost is replaced by Bregman divergences, a family of discrepancy measures with special cases that are well-suited for audio applications. We derived a gradient algorithm and an ADMM scheme for solving this problem and implemented them in the context of audio signal recovery. We evaluated the performance of these algorithms for PR from exact and modified spectrograms. We experimentally observed that



(a) Spectral convergence



(b) STOI

Figure 5: Performance of PR from power spectrograms (d = 2), measured with the SC (top) and STOI (bottom). The considered cost functions are from left to right : "left" beta-divergence with $\beta = 0.5$, "left" Kullback-Leibler, Quadratic, "right" Kullback-Leibler, "right" beta-divergence with $\beta = 0.5$.

when performing PR from exact or slightly degraded spectrograms, traditional algorithms based on the quadratic cost perform best. However, in the presence of high spectrogram distortion, these are outperformed by algorithms based on alternative cost functions. This highlights the potential of PR with the Bregman divergence for audio signal recovery from spectrograms under very noisy conditions. However it is difficult to recommend a specific alternative divergence at this stage. The choice is dependent on the amount of noise and possibly on the nature of the data itself (e.g., speech vs music). Gradient algorithms are very convenient because they can be applied to any setting, however finding efficient step sizes in every setting was chal-

lenging. In that respect, appropriate methods to choose this parameter such as non-monotonic backtracking has proved to be helpful and improved the reconstruction performance. Our ADMM algorithms appeared more stable with respect to the level of noise and to the nature of the data but their applicability is more limited as they require the proximal operator to be known for each setting.

In future work, we intend to further improve the proposed gradient descent algorithms, notably by leveraging more refined initialization schemes, and to explore other optimization strategies such as majorization-minimization. It would be also useful to conduct subjective listening tests to fully assess the potential of using Bregman divergences for a phase retrieval task. In the next chapter, we tackle PR with non-quadratic measures of fit in frameworks where some additional phase information is available: speech enhancement and source separation applications.

APPENDICES

4.A ALGORITHMS DERIVATIONS FOR REAL-VALUED SIGNALS

We here discuss the adaptation of our proposed gradient and ADMM algorithms to the specific case when the input signal is real-valued $\mathbf{x} \in \mathbb{R}^L$.

In this setting, the gradient algorithm can be easily deduced from its complex-valued counterpart. Indeed, since $\mathbf{x}$ is real-valued, the gradient of J simply reduces to $\nabla_{\mathbb{R}} J(\mathbf{x})$, as defined in Section 2.2. According to the property (2.2.7), this gradient is given by:

$$\nabla_{\mathbb{R}} J(\mathbf{x}) = 2\mathfrak{R}(\nabla J(\mathbf{x})). \tag{4.A.1}$$

where $\nabla J(\mathbf{x})$ is computed using the Wirtinger derivatives. Consequently, the gradient update rule is similar to the complex-valued case, up to a constant factor of 2 and with the difference that we only need to retain the real part after applying $\mathbf{A}^H$ (in practice, the inverse STFT).

Regarding the ADMM algorithm, we need to address the following sub-problem, in lieu of (4.2.26):

$$\min_{\mathbf{x}\in\mathbb{R}^L} \|(\mathbf{A}\mathbf{x})^d - \mathbf{b}\|_2^2. \tag{4.A.2}$$

where we note $\mathbf{b} = \mathbf{u}_{t+1} \odot e^{i\theta_{t+1}} - \frac{\lambda_t}{\rho}$. Since we only use ADMM algorithms with $d = 1$ in our experiments, we focus hereafter on this setting. By using again (2.2.7), we compute the gradient of the cost in (4.A.2) and set it at 0:

$$2\mathfrak{R}(\mathbf{A}^H\mathbf{A}\mathbf{x} - \mathbf{A}^H\mathbf{b}) = 0. \tag{4.A.3}$$

This yields the following solution:

$$\mathbf{x} = (\mathfrak{R}(\mathbf{A}^H\mathbf{A}))^{-1}\mathfrak{R}(\mathbf{A}^H\mathbf{b}). \tag{4.A.4}$$

When using the STFT with a self-dual window we have $\mathbf{A}^H\mathbf{A} = \mathbf{I}_L$ and the update becomes

$$\mathbf{x} = \mathfrak{R}(\mathbf{A}^H\mathbf{b}). \tag{4.A.5}$$

It is the same update as in the complex-valued case (4.2.27) up to retaining the real part after applying the inverse STFT $\mathbf{A}^H$.

4.B REGULARIZED GRADIENT EXPRESSION

For some Bregman divergences and/or exponents d, the gradient of the cost functions (4.2.1) and (4.2.2) is not defined when one or more

coefficients of $\mathbf{Ax}$ are zero-valued, which leads to division by zero and other potential numerical or conceptual issues. This is the case, for instance, when $d \leqslant 1$, when computing $|\mathbf{Ax}|^{d-2}$ with $d < 2$, or when computing $\psi'(|\mathbf{Ax}|^d)$ for a beta-divergence such that $\beta \leqslant 1$. Therefore, we propose a rigorous treatment of this issue by considering regularized cost functions. More specifically, we consider the following alternative cost for the PR right problem (a similar technique is used for treating its left counterpart):

$$J_\varepsilon(\mathbf{x}) := \mathcal{D}_\psi\left((\mathbf{r}^{\frac{2}{d}} + \varepsilon)^{\frac{d}{2}} \,|\, (|\mathbf{Ax}|^2 + \varepsilon)^{\frac{d}{2}}\right), \tag{4.B.1}$$

with $\varepsilon \ll 1$, such that J_ε is now always defined and differentiable at 0. This yields the corresponding regularized gradient expression:

$$\nabla J_\varepsilon(\mathbf{x}) = \frac{d}{2}\mathbf{A}^H\left[(|\mathbf{Ax}|^2 + \varepsilon)^{\frac{d}{2}-1} \odot \mathbf{Ax} \odot \mathbf{g}_{\psi,\varepsilon}\right], \tag{4.B.2}$$

with

$$\mathbf{g}_{\psi,\varepsilon} = \psi''((|\mathbf{Ax}|^2 + \varepsilon)^{\frac{d}{2}}) \odot ((|\mathbf{Ax}|^2 + \varepsilon)^{\frac{d}{2}} - (\mathbf{r}^{\frac{2}{d}} + \varepsilon)^{\frac{d}{2}}). \tag{4.B.3}$$

For the PR left problem, a similar expression is obtained:

$$\mathbf{g}_{\psi,\varepsilon} = \psi'((|\mathbf{Ax}|^2 + \varepsilon)^{\frac{d}{2}}) - \psi'((\mathbf{r}^{\frac{2}{d}} + \varepsilon)^{\frac{d}{2}}). \tag{4.B.4}$$

We used this variant in our experiments, and implemented it in practice with $\varepsilon = 10^{-8}$.

4.C NONNEGATIVITY CONSTRAINT ON $\mathbf{u}$ IN ADMM

Here we prove that the nonnegativity constraint on $\mathbf{u}$ in problem (4.2.24) can be ignored. Let us first rewrite this problem into scalar form, as this problem is separable entrywise:

$$\underset{u_k \geqslant 0}{\operatorname{argmin}}\, d_\psi(r_k \,|\, u_k) + \frac{\rho}{2}\||h_k| - u_k\|^2. \tag{4.C.1}$$

We will remove the index k in what follows for clarity. We aim to prove that:

$$\text{If } u < 0, \quad d_\psi(r \,|\, 0) + \frac{\rho}{2}|h|^2 \;\leqslant\; d_\psi(r \,|\, u) + \frac{\rho}{2}||h| - u|^2, \tag{4.C.2}$$

If this inequality holds, then the minimizer of the function defined in (4.C.1) necessarily belongs to $\mathbb{R}_+$. Consequently, the nonnegativity constraint can be dismissed. Equation (4.C.2) rewrites:

$$\begin{aligned}\psi(r) - \psi(0) - \psi'(0)r + \frac{\rho}{2}|h|^2 \leqslant \psi(r) - \psi(u) \\ - \psi'(u)(r-u) + \frac{\rho}{2}||h| - u|^2,\end{aligned} \tag{4.C.3}$$

which is equivalent to:

$$\psi(0) - \psi(u) + r\psi'(0) - \psi'(u)(0-u) \\ - r\psi'(u) + \frac{\rho}{2}[-2u|h| + u^2] \geqslant 0, \quad (4.C.4)$$

which finally rewrites:

$$\underbrace{d_\psi(0\,|\,u)}_{\text{term 1}} + \underbrace{r(\psi'(0) - \psi'(u))}_{\text{term 2}} + \underbrace{\frac{\rho}{2}[-2u|h| + u^2]}_{\text{term 3}} \geqslant 0. \quad (4.C.5)$$

The latter inequality holds for the following reasons:

- Term 1 is nonnegative by nonnegativity of Bregman divergences.
- Term 2 is nonnegative by convexity of ψ and nonnegativity of r: ψ is convex, therefore ψ' is monotonically non-decreasing. As $u < 0$, $\psi'(u) \leqslant \psi'(0)$ and $r(\psi'(0) - \psi'(u)) \geqslant 0$.
- Term 3 is nonnegative because u is negative.

Therefore, (4.C.2) holds, which demonstrates that the nonnegativity constraint in (4.2.24) can be dismissed. Finally, using a similar proof, we can show that the same holds for the "left" PR problem.

5

PHASE RETRIEVAL FOR AUDIO SOURCE SEPARATION

The contributions of this chapter have been published in [94].

5.1 Introduction 71
5.2 Phase retrieval with Bregman divergences and mixing constraint 72
5.2.1 Problem formulation 72
5.2.2 Projected gradient descent 72
5.2.3 Derivation of the gradient 73
5.2.4 Summary of the algorithm 74
5.3 Numerical experiments 74
5.3.1 Experimental setup 74
5.3.2 Influence of the step size 75
5.3.3 Comparison to other methods 76
5.4 Conclusion 77

5.1 INTRODUCTION

Audio source separation [29] consists in extracting the underlying *sources* that add up to form an observable audio *mixture*. As presented in Chapter 3, Section 3.2.3, state-of-the-art approaches for source separation estimate a nonnegative mask that is applied to a time-frequency (TF) representation of the audio mixture, such as the short-time Fourier transform (STFT) [153]. Applying a nonnegative mask to the mixture's STFT results in assigning its phase to each isolated source. Even though this practice is common and yields satisfactory results, it is well established [92] that when sources overlap in the TF domain, using the mixture's phase induces residual interference and artifacts in the estimates.

In this chapter, we consider phase recovery in audio source separation as an optimization problem involving alternative divergences which are more appropriate for audio processing. In Chapter 4, we addressed phase recovery with the Bregman divergences in a single-source setting. Here, we propose to extend this approach to a single-channel and multiple-sources framework, where the mixture's information can be exploited. To optimize the resulting objective, we derive a projected gradient algorithm [26]. We experimentally assess the potential of our approach for a speech enhancement task. Our results show that this method outperforms the multiple input spectrogram inversion (MISI) algorithm [63] for several Bregman divergences.

The rest of this chapter is structured as follows. In Section 5.2 we consider a new formulation of the problem with Bregman divergences and derive the proposed algorithm. Section 5.3 presents the experimental results. Finally, Section 5.4 draws some concluding remarks.

5.2 PHASE RETRIEVAL WITH BREGMAN DIVERGENCES AND MIXING CONSTRAINT

5.2.1 *Problem formulation*

Given an observed mixture $\mathbf{x} \in \mathbb{R}^L$ of C sources $\mathbf{x}^{(c)} \in \mathbb{R}^L$, whose target nonnegative TF measurements are $\mathbf{r}^{(c)}$, PR with multiple sources can be formulated as [96]:

$$\min_{\{\mathbf{x}^{(c)} \in \mathbb{R}^L\}_{c=1}^C} \sum_{c=1}^C \left\| \mathbf{r}^{(c)} - |\mathbf{A}\mathbf{x}^{(c)}|^d \right\|^2 \text{ s.t. } \sum_{c=1}^C \mathbf{x}^{(c)} = \mathbf{x}. \tag{5.2.1}$$

We propose to extend our previous approach described in Chapter 4 to a single-channel source separation framework. Indeed, as described in Chapter 3, Section 3.2.3, it is necessary to include the mixture information in the optimization problem so that the estimates sum up to the mixture. We replace the cost function in (5.2.1) with a Bregman divergence, which yields the following optimization problem:

$$\min_{\{\mathbf{x}^{(c)} \in \mathbb{R}^L\}_{c=1}^C} \sum_{c=1}^C J^{(c)}\left(\mathbf{x}^{(c)}\right) \text{ s.t. } \sum_{c=1}^C \mathbf{x}^{(c)} = \mathbf{x}, \tag{5.2.2}$$

where $J^{(c)}\left(\mathbf{x}^{(c)}\right) = \mathcal{D}_\psi\left(\mathbf{r}^{(c)} \,||\, |\mathbf{A}\mathbf{x}^{(c)}|^d\right)$ for the "right" problem and $J^{(c)}\left(\mathbf{x}^{(c)}\right) = \mathcal{D}_\psi\left(|\mathbf{A}\mathbf{x}^{(c)}|^d \,|\, \mathbf{r}^{(c)}\right)$ for its "left" counterpart.

5.2.2 *Projected gradient descent*

Similarly as in Chapter 4, we propose a gradient descent algorithm to minimize the objective defined in (5.2.2). The set of signals whose sum is equal to the observed mixture $\mathbf{x}$, appearing in the constraint of (5.2.2), is convex. As such, we may use the projected gradient algorithm [26] which boils down to alternating the two following updates:

$$\forall c,\ \mathbf{y}_{t+1}^{(c)} = \mathbf{x}_t^{(c)} - \mu \nabla J^{(c)}(\mathbf{x}_t^{(c)}) \tag{5.2.3}$$

$$\forall c,\ \mathbf{x}_{t+1}^{(c)} = \mathbf{y}_{t+1}^{(c)} + \frac{1}{C}\left(\mathbf{x} - \sum_{i=1}^C \mathbf{y}_i^{(t)}\right) \tag{5.2.4}$$

where $\nabla J^{(c)}$ denotes the gradient of $J^{(c)}$ with respect to $\mathbf{x}^{(c)}$ and $\mu > 0$ is the gradient step size. In a nutshell, (5.2.3) performs a gradient descent, and (5.2.4) projects the auxiliary variables $\mathbf{y}^{(c)}$ onto the set of estimates whose sum is equal to the mixture.

5.2.3 *Derivation of the gradient*

We derive hereafter the gradient of $J^{(c)}$. Using the chain rule [35], we have:

$$\nabla J^{(c)}(\mathbf{x}^{(c)}) = (\nabla|\mathbf{A}\mathbf{x}^{(c)}|^d)^T \mathbf{g}^{(c)}, \tag{5.2.5}$$

where $\nabla|\mathbf{A}\mathbf{x}^{(c)}|^d$ denotes the Jacobian of the multivariate function $\mathbf{x}^{(c)} \to |\mathbf{A}\mathbf{x}^{(c)}|^d$ (the Jacobian being the extension of the gradient for multivariate functions, we may use the same notation ∇), and:

$$\begin{aligned} \text{for the “right" problem, } \mathbf{g}^{(c)} &= \psi''(|\mathbf{A}\mathbf{x}^{(c)}|^d) \odot (|\mathbf{A}\mathbf{x}^{(c)}|^d - \mathbf{r}^{(c)}) \\ \text{for the “left" problem, } \mathbf{g}^{(c)} &= \psi'(|\mathbf{A}\mathbf{x}^{(c)}|^d) - \psi'(\mathbf{r}^{(c)}) \end{aligned}$$

where ψ' and ψ'' are applied entrywise. Now, let us note $\mathbf{A}_r$ and $\mathbf{A}_i$ the real and imaginary parts of $\mathbf{A}$, respectively. Using differentiation rules for element-wise matrix operations [35] and calculations similar as in Chapter 4, we have:

$$\begin{aligned} \nabla|\mathbf{A}\mathbf{x}^{(c)}|^d &= \nabla\left((\mathbf{A}_r\mathbf{x}^{(c)})^2 + (\mathbf{A}_i\mathbf{x}^{(c)})^2\right)^{\frac{d}{2}} \\ &= d \times \operatorname{diag}(|\mathbf{A}\mathbf{x}^{(c)}|^{d-2})\left(\operatorname{diag}(\mathbf{A}_r\mathbf{x}^{(c)})\mathbf{A}_r + \operatorname{diag}(\mathbf{A}_i\mathbf{x}^{(c)})\mathbf{A}_i\right). \end{aligned} \tag{5.2.6}$$

We now inject (5.2.6) in (5.2.5) and develop, which yields:

$$\begin{aligned} \nabla J^{(c)}(\mathbf{x}^{(c)}) = \mathbf{A}_r^T &\left(d \times \operatorname{diag}(\mathbf{A}_r\mathbf{x}^{(c)})\operatorname{diag}(|\mathbf{A}\mathbf{x}^{(c)}|^{d-2})\mathbf{g}^{(c)}\right) \\ + \mathbf{A}_i^T &\left(d \times \operatorname{diag}(\mathbf{A}_i\mathbf{x}^{(c)})\operatorname{diag}(|\mathbf{A}\mathbf{x}^{(c)}|^{d-2})\mathbf{g}^{(c)}\right). \end{aligned} \tag{5.2.7}$$

We remark that $\forall \mathbf{u}, \mathbf{v} \in \mathbb{C}^K$, $\operatorname{diag}(\mathbf{u})\mathbf{v} = \mathbf{u} \odot \mathbf{v}$, so we further simplify this expression:

$$\begin{aligned} \nabla J^{(c)}(\mathbf{x}^{(c)}) = \mathbf{A}_r^T &\left(d \times (\mathbf{A}_r\mathbf{x}^{(c)}) \odot |\mathbf{A}\mathbf{x}^{(c)}|^{d-2} \odot \mathbf{g}^{(c)}\right) \\ + \mathbf{A}_i^T &\left(d \times (\mathbf{A}_i\mathbf{x}^{(c)}) \odot |\mathbf{A}\mathbf{x}^{(c)}|^{d-2} \odot \mathbf{g}^{(c)}\right). \end{aligned} \tag{5.2.8}$$

Finally, we remark that $\forall \mathbf{u} \in \mathbb{C}^K$, $\Re(\mathbf{A}^H\mathbf{u}) = \mathbf{A}_r^T\Re(\mathbf{u}) + \mathbf{A}_i^T\Im(\mathbf{u})$, thus we can rewrite the gradient (5.2.8) as:

$$\nabla J^{(c)}(\mathbf{x}^{(c)}) = d \times \Re\left(\mathbf{A}^H((\mathbf{A}\mathbf{x}^{(c)}) \odot |\mathbf{A}\mathbf{x}^{(c)}|^{d-2} \odot \mathbf{g}^{(c)})\right). \tag{5.2.9}$$

Remark: When considering the quadratic cost (for which the “right" and “left" problems are equivalent) with $d = 1$ and step size $\mu = 1$, the gradient update becomes equivalent to the MISI update. This outlines that our method generalizes MISI, as the latter can be seen as a particular case of the projected gradient descent algorithm.

Algorithm 17 : Phase recovery with Bregman divergences for audio source separation: gradient descent.

Inputs: Measurements $\mathbf{R}^{(c)} \in \mathbb{R}_+^{M\times N}$, mixture $\mathbf{x} \in \mathbb{R}^L$, step size $\mu > 0$, Bregman divergence function ψ.
Initialization:
$\forall c,\ \mathbf{x}^{(c)} = \mathrm{iSTFT}(\left(\mathbf{R}^{(c)}\right)^{1/d} \odot \frac{\mathrm{STFT}(\mathbf{x})}{|\mathrm{STFT}(\mathbf{x})|})$
while *stopping criteria not reached* **do**
  $\forall c,\ \mathbf{x}^{(c)} = \mathrm{STFT}(\mathbf{x}^{(c)})$
  if *"right"* **then**
    $\mathbf{G}^{(c)} = \psi''(|\mathbf{x}^{(c)}|^d) \odot (|\mathbf{x}^{(c)}|^d - \mathbf{R}^{(c)})$
  else if *"left"* **then**
    $\mathbf{G}^{(c)} = \psi'(|\mathbf{x}^{(c)}|^d) - \psi'(\mathbf{R}^{(c)})$
  $\forall c,\ \mathbf{y}^{(c)} = \mathbf{x}^{(c)} - \mu d \times \mathrm{iSTFT}(\mathbf{x}^{(c)} \odot |\mathbf{x}^{(c)}|^{d-2} \odot \mathbf{G}^{(c)})$
  $\forall c,\ \mathbf{x}^{(c)} = \mathbf{y}^{(c)} + (\mathbf{x} - \sum_{i=1}^{C} \mathbf{y}_i)/C$
end
Output: $\{\mathbf{x}^{(c)}\}_{c=1}^{C}$

5.2.4 *Summary of the algorithm*

The proposed algorithm consists of alternating the updates (5.2.3) and (5.2.4). A natural choice for obtaining initial source estimates consists in assigning the mixture's phase to each source's STFT, which is known as *amplitude masking* and is commonly employed to initialize MISI [63, 156, 158]:

$$\forall c,\ \mathbf{x}_0^{(c)} = \mathbf{A}^\dagger \left(\left(\mathbf{r}^{(c)}\right)^{1/d} \odot \frac{\mathbf{A}\mathbf{x}}{|\mathbf{A}\mathbf{x}|} \right). \tag{5.2.10}$$

We provide in Algorithm 17 the pseudo-code for practical implementation of our method.

5.3 NUMERICAL EXPERIMENTS

In this section, we assess the potential of Algorithm 17 for a speech enhancement task, that is, with $C = 2$ and where $\mathbf{x}_1$ and $\mathbf{x}_2$ correspond to the clean speech and noise, respectively. Note however that this framework is applicable to alternative separation scenarios, such as musical instruments [18] or multiple-speakers [153] separation. The code related to these experiments is available online.[1]

5.3.1 *Experimental setup*

DATA. As acoustic material, we build a set of mixtures of clean speech and noise. The clean speech is obtained from the VoiceBank

1 https://github.com/magronp/bregmisi

test set [141], from which we randomly select 100 utterances. The noise signals are obtained from the DEMAND dataset [137], from which we select noises from three real-world environments: a living room, a bus, and a public square. For each clean speech signal, we randomly select a noise excerpt cropped at the same length than that of the speech signal. We then mix the two signals at various input signal-to-noise ratios (iSNRs) (10, 0, and -10 dB). All audio excerpts are single-channel and sampled at 16,000 Hz. The STFT is computed with a 1024 samples-long (64 ms) Hann window, no zero-padding, and 75% overlap. The dataset is split into two subsets of 50 mixtures: a *validation* set, on which the step size is tuned (see Section 5.3.2); and a *test* set, on which the proposed algorithm is compared to MISI.

SPECTROGRAM ESTIMATION. In realistic scenarios, the nonnegative measurements $\mathbf{r}^{(c)}$ are estimates of the magnitude or power spectrograms of the sources. To obtain such estimates, we use Open-Unmix [131], an open implementation of a three-layer BLSTM neural network, originally tailored for music source separation applications. This network has been adapted to a speech enhancement task. It was trained on our dataset, except using different speakers and noise environments, as described in [142]. We use the trained model available at [140]. This network is fed with the noisy mixtures and outputs an estimate for the clean speech and noise spectrograms, which serve as inputs to the phase retrieval methods.

COMPARED METHODS. We test the proposed projected gradient descent method described in Algorithm 17 in a variety of settings. We consider magnitude and power measurements ($d = 1$ or 2), "right" and "left" problems, and various values of β for the divergence ($\beta = 0$ to 2 with a step of 0.25). The step size is tuned on the validation set. As comparison baseline, we consider the MISI algorithm (which corresponds to our algorithm with $\beta = 2$, $d = 1$ and $\mu = 1$). Following traditional practice with MISI [156, 158], all algorithms are run with 5 iterations.
In order to evaluate the speech enhancement quality, we compute the signal-to-distortion ratio (SDR) between the true clean speech $\mathbf{x}_1^\star$ and its estimate $\mathbf{x}_1$. For more clarity, we will present the SDR improvement (SDRi) of a method (whether MISI or Algorithm 17) over initialization.

5.3.2 *Influence of the step size*

First, we study the impact of the step size on the performance of the proposed algorithm using the validation set. The mean SDRi on this subset is presented in Figure 1 in the "right" setting, but similar conclusions can be drawn in the "left" setting. For $d = 1$, we

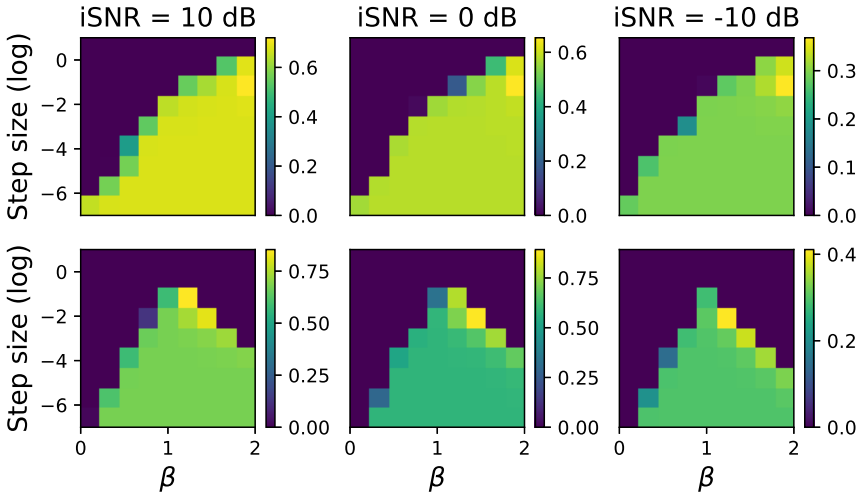


Figure 1: Average SDRi on the validation set obtained with the proposed algorithm at various iSNRs, when $d = 1$ (top) and $d = 2$ (bottom). For better readability, we set the SDRi at 0 when convergence issues occur as visually inspected, or when the SDRi is below 0, as this implies a decreasing performance over iterations, which is not desirable.

remark that the range of possible step sizes becomes more limited as β decreases towards 0 (which corresponds to the IS divergence). Conversely, when $d = 2$, we observe that divergences corresponding to β close to 1 (i.e., the KL divergence) allow for more flexibility when it comes to choosing an appropriate step size.
For each setting, we pick the value of the step size that maximizes the SDR on this subset and use it in the following experiment.

5.3.3 *Comparison to other methods*

The separation results on the test set are presented in Figure 2. We observe that at high (10 dB) or moderate (0 dB) iSNRs, the proposed algorithm overall outperforms MISI when $d = 2$ and for $\beta \geqslant 1$. We notably remark a performance peak at around $\beta = 1.25$ depending on the iSNR. This observation is consistent with the findings of Chapter 4, where the gradient algorithm using the KL divergence (i.e., $\beta = 1$) in a similar scenario ($d = 2$ and "left" formulation) exhibited good performance.

At low iSNR (−10 dB), the proposed method outperforms the MISI baseline when $d = 2$ and for the "left" problem formulation. This behavior is somewhat reminiscent of Chapter 4: when the spectrograms are severely degraded (i.e., at low iSNR), the algorithm based on the quadratic cost (here, MISI) is outperformed by algorithms based on more suitable alternative cost functions. Besides, it is also outperformed by a gradient algorithm based on the same quadratic cost

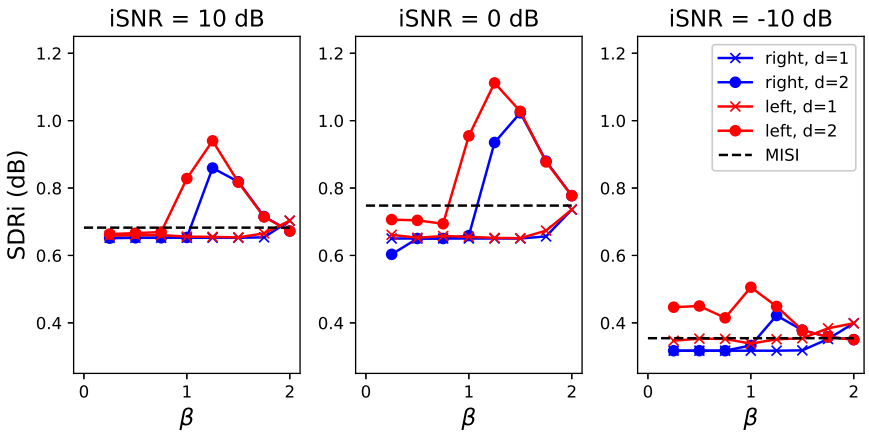


Figure 2: Average SDRi on the test set obtained with MISI and with the proposed algorithm (in different settings) at various iSNRs.

when using a fine-tuned step size. This highlights the potential interest of phase recovery with Bregman divergences in such a scenario.

Finally, note that the performance of the proposed method strongly depends on the speaker and the kind of noise used in the experiments For instance, for public square and bus noises, the proposed method consistently outperforms MISI at 10 dB iSNR while both methods perform similarly at -10 dB iSNR. However, for living room noises, a different trend is observed: in particular, the improvement of the proposed algorithm over MISI becomes more significant at -10 dB iSNR. As a result, further investigations are needed to identify the optimal β for a given class of signals, which should reduce this sensitivity and improve the above results.

5.4 CONCLUSION

In this chapter, we have addressed the problem of phase recovery with Bregman divergences in the context of audio source separation. We derived a projected gradient algorithm for optimizing the resulting cost. We experimentally observed that when the spectrograms are highly degraded, some of these Bregman divergences induce better speech enhancement performance than the quadratic cost, upon which the widely-used MISI algorithm builds.

In future work, we will explore other optimization schemes for addressing this problem, such as majorization-minimization or the ADMM algorithm introduced in Chapter 4. We will also leverage these algorithms in a deep unfolding paradigm, which combines the qualities of model-based and learning-based methods. This approach is studied in the next chapter with PR.

Part III

PHASE RETRIEVAL WITH UNFOLDED ALGORITHMS

6

LEARNING PROXIMITY OPERATORS FOR PHASE RETRIEVAL

The contributions of this chapter have been published in [145].

6.1 Introduction 81
6.2 Learning proximity operators in unfolded ADMM . . . 82
6.2.1 Proposed general unfolded architecture 82
6.2.2 Proposed parameterization with APL units . . . 83
6.3 Interpretability and characterization 84
6.3.1 Discussion about interpretability 84
6.3.2 Characterization of $\mathsf{F}(\mathbf{y}, \mathbf{r})$ as a proximity operator 85
6.4 Numerical experiments 86
6.4.1 Experimental setup 86
6.4.2 Results 87
6.5 Conclusion 88

6.1 INTRODUCTION

The phase retrieval problem is traditionnally formulated as an optimization problem with a quadratic cost. In Chapter 4, PR has been addressed by replacing the quadratic cost function with Bregman divergences. As seen in Chapter 4, prescribing a cost function that is optimal for all signal processing problems and classes of audio signals remains however challenging.

On the other hand, recent PR approaches have leveraged deep neural networks (DNNs) [2, 134, 136, 138]. Despite their successful performance in a large number of tasks, the enthusiasm for DNNs can be tempered by a general lack of explainability due to their black box structure, and by their limited ability to generalize to unseen data or experimental conditions. Deep unfolding (or unrolling) [56, 67] is a promising attempt to alleviate these limitations with model-based architectures derived from iterative algorithms.

In this chapter, we propose to unfold the ADMM algorithm for PR proposed in Chapter 4. Our method builds upon observing that the choice of the discrepancy measure only affects the computation of a proximity operator in the ADMM updates. Therefore, we can recast the problem of metric learning as a problem of proximity operator learning in the unfolded ADMM. To that end, we replace this proximity operator with a trainable activation function. We show that the proposed parameterization of the network is connected to the metric involved in the original optimization problem, which yields an

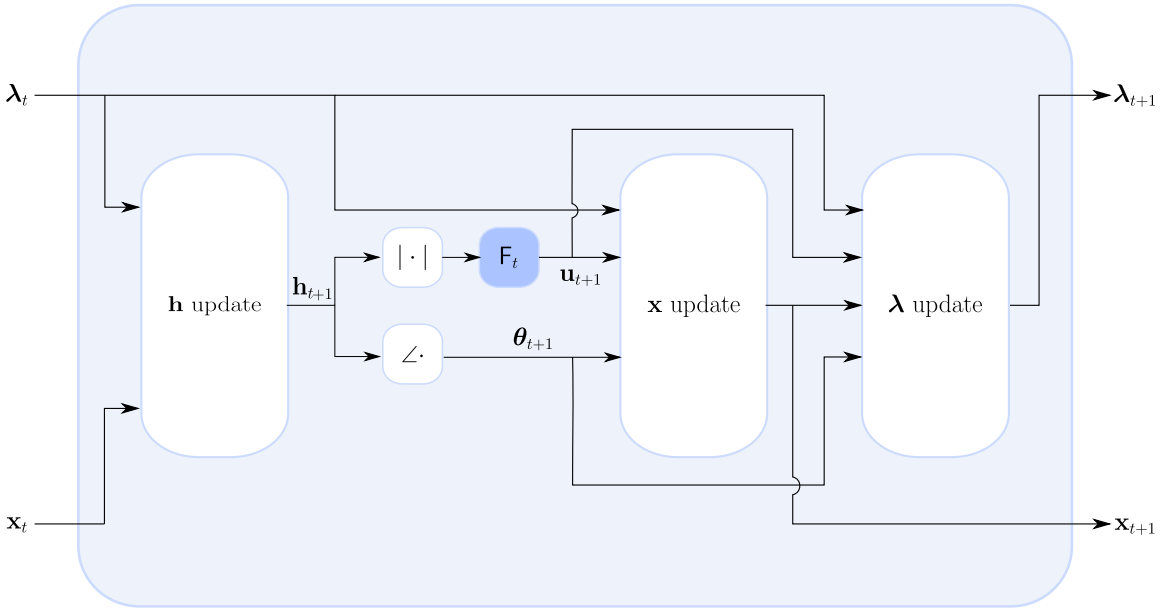


Figure 3: One layer of the proposed unfolded architecture.

interpretable architecture. Experiments performed on speech signals demonstrate the efficiency of our method, which outperforms a baseline ADMM [84] with a number of iterations equal to the number of layers in the unfolded ADMM.

This chapter is structured as follows. Section 6.2 presents the proposed method. A discussion about interpretability and the characterization of proximity operators follows in Section 6.3. Our method is tested experimentally in Section 6.4 and Section 6.5 draws some concluding remarks.

6.2 LEARNING PROXIMITY OPERATORS IN UNFOLDED ADMM

6.2.1 *Proposed general unfolded architecture*

The ADMM updates detailed in Section 4.2.3 consist in successive linear and nonlinear computations. As such, this algorithm can be viewed as a neural network U via unfolding the iterations:

$$(\mathbf{x}_T, \boldsymbol{\lambda}_T) = \mathsf{U}(\mathbf{x}_0, \boldsymbol{\lambda}_0) = \mathsf{U}_1 \circ \cdots \circ \mathsf{U}_T(\mathbf{x}_0, \boldsymbol{\lambda}_0), \tag{6.2.1}$$

where U_t denotes the t-th layer of the network, mimicking the t-th iteration of the ADMM algorithm, as illustrated in Fig. 3. The layer U_t can be decomposed into two linear parts denoted by $\mathsf{L}_t^{(1)}$ and $\mathsf{L}_t^{(2)}$, and a nonlinear part NL_t as follows:

$$\mathsf{L}_t^{(1)} : (\mathbf{x}_{t-1}, \boldsymbol{\lambda}_{t-1}) \mapsto \mathbf{h}_t \tag{6.2.2}$$

$$\mathsf{NL}_t : \mathbf{h}_t \mapsto (\mathbf{u}_t, \boldsymbol{\theta}_t) = (\mathsf{F}_t(|\mathbf{h}_t|, \mathbf{r}), \angle \mathbf{h}_t) \tag{6.2.3}$$

$$\mathsf{L}_t^{(2)} : (\mathbf{x}_{t-1}, \boldsymbol{\lambda}_{t-1}, \mathbf{u}_t, \boldsymbol{\theta}_t) \mapsto (\mathbf{x}_t, \boldsymbol{\lambda}_t), \tag{6.2.4}$$

with $\mathbf{h}_t, \mathbf{x}_t, \boldsymbol{\lambda}_t$ respectively defined as in (4.2.21), (4.2.27) and (4.2.28). F_t denotes a parameterized sublayer modeling the proximity operator of equation (4.2.24). Since the choice of the discrepancy measure $\mathcal{D}_\psi$ only affects the proximity operator (4.2.24) in the updates, we can recast the problem of metric learning as the problem of proximity operator learning. We propose to leverage a trainable activation function in order to model this layer and learn the proximity operator.

6.2.2 *Proposed parameterization with APL units*

To build the non-linear sublayers F_t that model $\mathrm{prox}_{\rho^{-1}\mathcal{D}_{\psi(\cdot\,|\,\mathbf{r})}}$, we first reformulate this operator as follows. Let $\mathbf{v} \in \mathbb{R}^K$ and $f(\mathbf{z}) = \sum_{k=1}^K [\psi(z_k) + v_k z_k]$. We have [28]:

$$\mathrm{prox}_{\rho^{-1} f}(\mathbf{y}) = \mathrm{prox}_{\rho^{-1}\tilde{\psi}}\left(\mathbf{y} - \rho^{-1}\mathbf{v}\right), \tag{6.2.5}$$

where $\tilde{\psi}(\mathbf{z}) = \sum_k \psi(z_k)$. Setting $\mathbf{v} = -\psi'(\mathbf{r})$ in (6.2.5), with ψ' applied entrywise, it is straightforward to see that:

$$\mathrm{prox}_{\rho^{-1}\mathcal{D}_\psi(\cdot\,|\,\mathbf{r})}(\mathbf{y}) = \mathrm{prox}_{\rho^{-1}\tilde{\psi}}(\mathbf{y} + \rho^{-1}\psi'(\mathbf{r})). \tag{6.2.6}$$

This formulation of the proximity operator is more convenient than (4.2.24) since the measurements $\mathbf{r}$ no longer appear in the input function of the proximity operator, but instead in the argument of the latter (with $\mathbf{y}$). This leads to a more natural parameterization for unrolling.

Let us first derive the proximity operator (6.2.6) in a simple scenario, namely the quadratic cost ($\tilde{\psi} = \frac{1}{2}\|\cdot\|^2$, $\psi'(\mathbf{r}) = \mathbf{r}$). In this case we have [84]:

$$\mathrm{prox}_{\rho^{-1}\frac{1}{2}\|\cdot-\mathbf{r}\|^2}(\mathbf{y}) = \frac{\mathbf{y} + \rho^{-1}\mathbf{r}}{1 + \rho^{-1}}. \tag{6.2.7}$$

As a result, a first simple approach for proximity operator learning would consist in treating ρ as a learnable parameter. However, our early experiments have shown poor performance with this approach, which is due to the very low expressive power of such a model (only one scalar value). More generally, one can consider a beta-divergence with shape parameter β, for which $\psi'(\mathbf{r}) = \frac{\mathbf{r}^{\beta-1}}{\beta-1}$ [66]. However, the proximity operator of $\tilde{\psi}$ is not available for every beta-divergence.

To alleviate this issue, we model this unavailable proximity operator using Adaptive Piecewise Linear (APL) activations [48]. They are defined by:

$$\mathsf{APL}(\mathbf{y}) := \max(\mathbf{y}, 0) + \sum_{c=1}^{C} w_c \max(-\mathbf{y} + b_c, 0), \tag{6.2.8}$$

where w_c and b_c are learnable parameters controlling the slopes and biases of the linear segments, and the max is applied entry-wise.

Then, we propose the following parameterization of the nonlinear layer F_t:

$$\mathsf{F}_t(\mathbf{y}, \mathbf{r}) = \mathsf{APL}_t\left(\gamma_t^{(1)}\mathbf{y} + \gamma_t^{(2)}\frac{\mathbf{r}^{\beta_t - 1}}{\beta_t - 1}\right), \tag{6.2.9}$$

with learnable parameters $w_{c,t}, b_{c,t}, \gamma_t^{(1)}, \gamma_t^{(2)}$, and β_t. Even though *ad hoc*, this parameterization is motivated by the following considerations:

- APL can represent any continuous piecewise linear function over a subset of real numbers. As such, it generalizes the proximity operator obtained in the quadratic case [84].
- The term in the form of $\frac{\mathbf{r}^{\beta-1}}{\beta-1}$ in (6.2.9) is reminiscent of ψ' for beta-divergences, as mentioned above.
- Introducing learnable weights $\gamma_t^{(1)}$ and $\gamma_t^{(2)}$ allows to increase the model capacity, as it was shown beneficial in our preliminary experiments.

Note that when $w_c = 0$, $\gamma_t^{(1)} = \frac{1}{1+\rho^{-1}}$, $\gamma_t^{(2)} = \frac{\rho^{-1}}{1+\rho^{-1}}$ and $\beta_t = 2$, F_t is equal to the proximity operator for the quadratic cost (6.2.7). Overall, our parameterization (6.2.9) is an interesting trade-off between tractability, interpretability, and expressiveness.

Two variants of the proposed architecture will be considered in our experiments. In the "untied" variant, each layer uses different parameters, and the global set of parameters is $\left\{\{w_{c,t}, b_{c,t}\}_{c=1}^{C}, \gamma_t^{(1)}, \gamma_t^{(2)}, \beta_t\right\}_{t=1}^{T}$, while in the "tied" variant, the parameters are shared among layers, i.e., constant with t.

In the end, after learning these parameters (see Section 6.4), the proposed method, termed unfolded ADMM (UADMM), estimates a signal $\mathbf{x}_T$ via $(\mathbf{x}_T, \boldsymbol{\lambda}_T) = \mathsf{U}(\mathbf{x}_0, \boldsymbol{\lambda}_0)$, where $\mathbf{x}_0$ is some initial estimate.

6.3 INTERPRETABILITY AND CHARACTERIZATION

6.3.1 *Discussion about interpretability*

Under mild assumptions detailed in the following, we can prove that there exists a closed-form function $f_{\mathbf{r},t} : \mathbb{R}^K \to \mathbb{R} \cup \{+\infty\}$ such that $\mathsf{F}_t(\mathbf{y}, \mathbf{r}) = \mathrm{prox}_{f_{\mathbf{r},t}}(\mathbf{y})$. In the "tied" variant, where $f_{\mathbf{r},t} = f_{\mathbf{r}}$, reconstructing $f_{\mathbf{r}}$ from the learned parameters is analogous to identifying the metric $\mathcal{D}_\psi(\cdot|\mathbf{r})$ involved in the PR optimization problem. With the relaxation proposed in the "untied" case, this interpretation is more limited as $f_{\mathbf{r},t}$ is different in each layer of the network.

Note that when replacing (6.2.6) with (6.2.9), we have disentangled the proximity operator of $\tilde{\psi}$ and the derivative ψ', in addition to introducing weights $\gamma_t^{(1)}$ and $\gamma_t^{(2)}$. As a result, the function $f_{\mathbf{r}}$ is no longer

guaranteed to be a Bregman divergence, strictly speaking. Nevertheless, we can still interpret it as a measure of discrepancy between $\mathbf{y}$ and $\mathbf{r}$.

6.3.2 *Characterization of* $\mathsf{F}(\mathbf{y}, \mathbf{r})$ *as a proximity operator*

We address the problem of identifying a function $f_{\mathbf{r}} : \mathbb{R}^K \to \mathbb{R} \cup \{+\infty\}$ such that $\mathsf{F}(\mathbf{y}, \mathbf{r}) = \text{prox}_{f_{\mathbf{r}}}(\mathbf{y})$, with F defined in (6.2.9). Note that we ignore here the layer index t for simplicity.

Characterization with APL

We first address the case of strictly increasing APL functions as defined in (6.2.8), with negative weights and at least one nonnegative bias b_c. Let us consider the following convex, lower semi-continuous function $\overline{\mathsf{APL}}$ such that $\forall z \in \mathbb{R}$:

$$\overline{\mathsf{APL}}(z) = \frac{z^2}{2}\chi_{[0;+\infty]}(z) + \sum_{c=1}^{C} w_c \left(\frac{-z^2}{2} + b_c z\right) \chi_{]-\infty;b_c]}(z). \quad (6.3.1)$$

Since for any $z \in \mathbb{R}$, $\mathsf{APL}(z)$ is a subgradient of $\overline{\mathsf{APL}}(z)$, and denoting $\widetilde{\mathsf{APL}}(\mathbf{z}) = \sum_{k=1}^{K} \overline{\mathsf{APL}}(z_k)$, it is straightforward to show that the Theorem 3 stands for $g(\mathbf{z}) = \mathsf{APL}(\mathbf{z})$ and $\tilde{g}(\mathbf{z}) = \widetilde{\mathsf{APL}}(\mathbf{z})$. Besides, since APL is invertible we can use the relation (2.2.19) to identify σ:

$$\sigma(\mathbf{y}) = \langle \mathsf{APL}^{-1}(\mathbf{y}) \,;\, \mathbf{y} \rangle - \frac{1}{2}\|\mathbf{y}\|^2 - \widetilde{\mathsf{APL}}(\mathsf{APL}^{-1}(\mathbf{y})), \quad (6.3.2)$$

with:

$$\mathsf{APL}^{-1}(\mathbf{y}) = \frac{\mathbf{y} - \sum_{c=1}^{C} w_c b_c \chi_{]-\infty, \mathsf{APL}(b_c)]}(\mathbf{y})}{\chi_{[\mathsf{APL}(0), +\infty[}(\mathbf{y}) - \sum_{c=1}^{C} w_c \chi_{]-\infty, \mathsf{APL}(b_c)]}(\mathbf{y})}. \quad (6.3.3)$$

Characterization with F

Finally, let us retrieve $f_{\mathbf{r}} : \mathbb{R}^K \to \mathbb{R} \cup \{+\infty\}$ such that $\mathsf{F}(\mathbf{y}, \mathbf{r}) = \text{prox}_{f_{\mathbf{r}}}(\mathbf{y})$. Drawing on the previous section and using the definition of F from (6.2.9), we have:

$$\mathsf{F}(\mathbf{y}, \mathbf{r}) = \text{prox}_{\sigma}\left(\gamma^{(1)}\mathbf{y} + \gamma^{(2)}\frac{\mathbf{r}^{\beta-1}}{\beta-1}\right). \quad (6.3.4)$$

To fully identify $f_{\mathbf{r}}$, we first need to reformulate (6.3.4) so that the argument of the right hand side term simply becomes $\mathbf{y}$. To that end, we leverage a property from [28], which consists in first rewriting (6.3.4) as follows:

$$\mathsf{F}(\mathbf{y}, \mathbf{r}) = \text{prox}_{\sigma}\left(\frac{\mathbf{y} - \mathbf{q}}{2\alpha + 1}\right), \quad (6.3.5)$$

with $\alpha = \dfrac{1-\gamma^{(1)}}{2\gamma^{(1)}}$ and $\mathbf{q} = -\dfrac{\gamma^{(2)}}{\gamma^{(1)}}\dfrac{\mathbf{r}^{\beta-1}}{\beta-1}$. The property from [28] then states that:

$$\operatorname{prox}_{\varphi+\alpha\|\cdot\|^2+\langle\mathbf{q};\cdot\rangle}(\mathbf{y}) = \operatorname{prox}_{\varphi/(2\alpha+1)}\left(\frac{\mathbf{y}-\mathbf{q}}{2\alpha+1}\right). \tag{6.3.6}$$

Let $\varphi = (2\alpha+1)\sigma$. Combining (6.3.5) and (6.3.6) yields:

$$\mathsf{F}(\mathbf{y},\mathbf{r}) = \operatorname{prox}_{(2\alpha+1)\sigma+\alpha\|\cdot\|^2+\langle\mathbf{q};\cdot\rangle}(\mathbf{y}). \tag{6.3.7}$$

As a result, from (6.3.7) we can identify $\mathsf{f}_{\mathbf{r}}$ such that its proximity operator is F. If we further exploit the definition of σ from (6.3.2), we finally have:

$$\mathsf{f}_{\mathbf{r}}(\mathbf{y}) = \frac{1}{\gamma^{(1)}}\left\langle \mathsf{APL}^{-1}(\mathbf{y}) - \gamma^{(2)}\frac{\mathbf{r}^{\beta-1}}{\beta-1}\,;\,\mathbf{y}\right\rangle - \frac{1}{2}\|\mathbf{y}\|^2 - \frac{1}{\gamma^{(1)}}\widetilde{\mathsf{APL}}(\mathsf{APL}^{-1}(\mathbf{y})). \tag{6.3.8}$$

Therefore, using (6.3.8) one can recover the function associated with the learned proximity operator, and consequently identify the metric involved in the formulation of the PR problem.

6.4 NUMERICAL EXPERIMENTS

In this section, we assess the potential of UADMM for PR of speech signals. Our code is implemented using the PyTorch framework [120] and is available online for reproducibility.[1]

6.4.1 *Experimental setup*

Data

We build a set of speech signals from the TIMIT dataset [49]. The dataset is split into training, validation, and test subsets containing 1000, 10, and 50 utterances, respectively (note that we did not observe a significant performance improvement when using a larger training set). The signals are mono, sampled at 16 kHz and cropped to 2 seconds. The STFT is computed with a 1024 samples-long (46 ms) self-dual sine window (*cf.* Chapter 4) and 50% overlap. STFT magnitudes ($\mathrm{d} = 1$) are considered as nonnegative observations $\mathbf{r}$.

Training

The network is trained with the ADAM algorithm [71] using a learning rate of 10^{-4}. We use a structure with $\mathrm{T} = 15$ layers and $\mathrm{C} = 3$, as these values have shown to be a good trade-off between performance

1 https://github.com/phvial/LearningProxPR

and number of parameters in preliminary experiments. Batches of 10 signals with a maximum of 200 epochs are used for training. Training is stopped when the cost function on the validation subset starts increasing. Given that we consider speech signals, we train the network by minimizing the negative STOI between the estimated and ground truth signals. Indeed, this strategy was shown to be efficient for speech enhancement applications [108, 161]. The negative STOI metric used for training the network is implemented in PyTorch via the `pytorch-stoi` library [114].

Methods

As baselines, we consider GLA [59] (run for 1500 iterations), and ADMM using a quadratic cost and $\rho = 10^{-3}$, since this setup has exhibited good performance in our previous study under similar conditions (*cf.* Chapter 4). ADMM is run for a variable number of iterations, with a maximum of 1500 (performance does not further improve beyond). For fairness, the linear parts of UADMM use the same value for ρ, and it is initialized such that F_t replicates the quadratic proximity operator . All methods use the same initial signal estimate $\mathbf{x}_0$ computed using the ground truth magnitudes $\mathbf{r}$ and a random uniform phase, and $\boldsymbol{\lambda}_0 = 0$.

Evaluation

Reconstruction performance is assessed with the STOI metric computed on the test set with the `pystoi` library [113].

6.4.2 *Results*

First, we display the training loss over epochs in Fig. 4. Both UADMM variants outperform the baseline ADMM with 15 iterations on the training set. UADMM-untied reaches a lower cost value than its tied counterpart, which was expected since this variant contains more learnable parameters. They reach a performance comparable to that of ADMM using 150 and 75 iterations, respectively. The results on the test set presented in Fig. 5 confirm that the proposed UADMM approach significantly outperforms the classical ADMM using the same number of iterations, as well as the GLA baseline. A fully-converged ADMM algorithm (using 1500 iterations) exhibits a higher STOI than our 15 layers-based approach. Nonetheless, a more fair comparison would involve that both approaches use the same total number of iterations/layers.

To that end, we consider an *ad hoc* extension of our method, where we duplicate the 15-layer trained UADMM network in order to increase the total amount of layers without additional training. The results presented in Fig. 6 show that this method consistently and

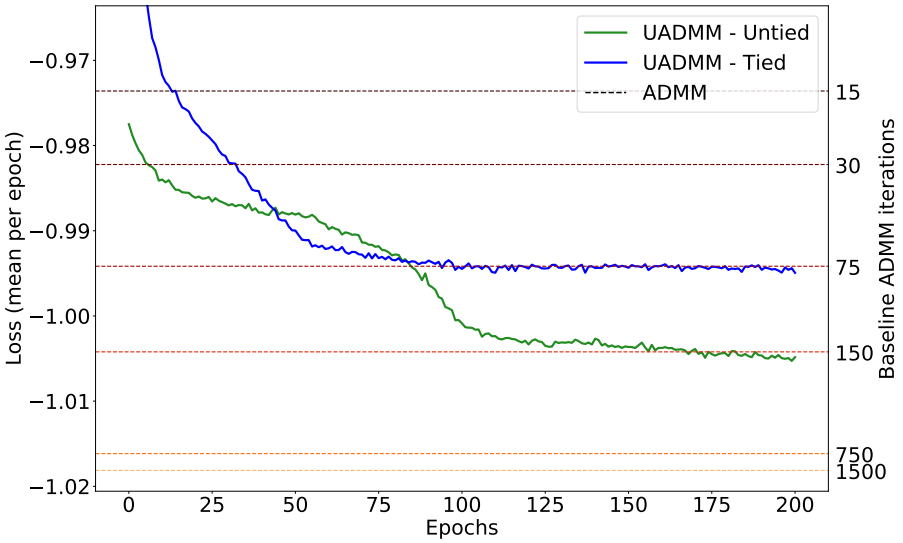


Figure 4: Training loss (negative STOI) over epochs. Note that `pytorch-stoi` implementation does not exactly replicate the original metric and consequently yields values lower than −1.

significantly outperforms ADMM for any number of iterations. In particular, the performance of the fully-converged ADMM (after 1500 iterations) is reached at only 30 "iterations" for UADMM-untied (i.e., twice the number of trained layers), which exhibits the computational advantage of the proposed approach.

Finally, let us point out that UADMM-tied with T layers is equivalent to applying T iterations of a standard ADMM algorithm using a learned metric $f_{\mathbf{r}}$ (note however that it differs from the ADMM baseline used in these experiments, which uses a quadratic cost). Following the derivations in Section 6.3, we compute these metrics ($f_{\mathbf{r}}$ in the tied case and $f_{\mathbf{r},t}$ in the untied case) from the trained activation functions, and display them in Fig. 7. These resemble beta-divergences with $\beta \in [1.5, 2.5]$. This is consistent with previous results from the literature [46], where this range of values has shown good performance for audio spectral decomposition.

6.5 CONCLUSION

In this chapter, we have addressed the problem of metric learning for phase retrieval by unfolding the ADMM algorithm proposed in Chapter 4 into a neural network. We proposed to replace the proximity operator involved in this algorithm with learnable activation functions, since this operator conveys the information about the discrepancy measure used in formulating the PR problem. Experiments conducted on speech signals show that this approach outperforms the ADMM algorithm while keeping a light and interpretable structure. In future work, we intend to further study the parameterization

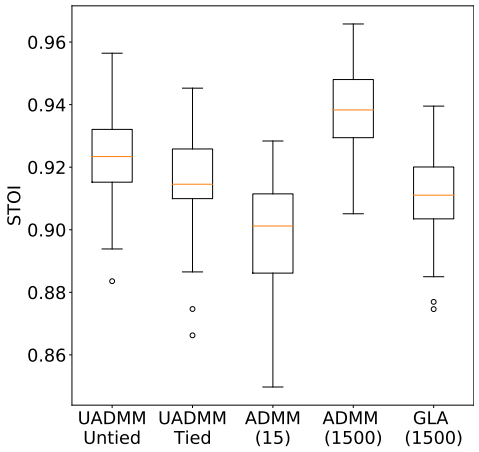


Figure 5: Performance on the test set. Each box-plot is made up of a central line indicating the median, box edges indicating the 1st and 3rd quartiles, whiskers indicating the extremal values, and circles representing the outliers.

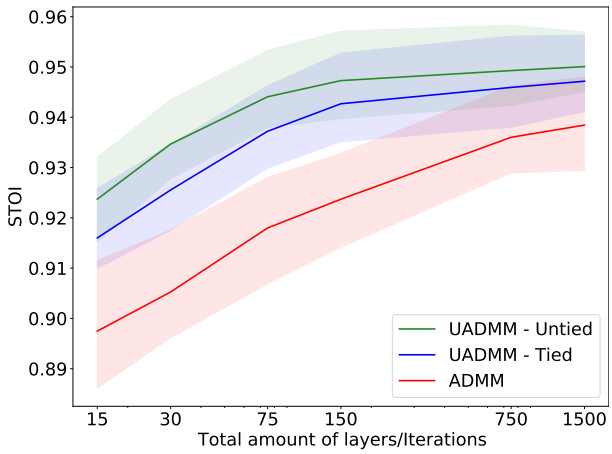


Figure 6: Evaluation with STOI over test dataset with iterated model. The solid lines denote the mean STOI and the light colored areas the values between the first and the third quartile.

of the unfolded network. For example, it would be useful to learn the linear operators that correspond to intermediate representations of the estimate. We also intend to extend this framework to other inverse problems in audio, such as declipping or dereverberation.

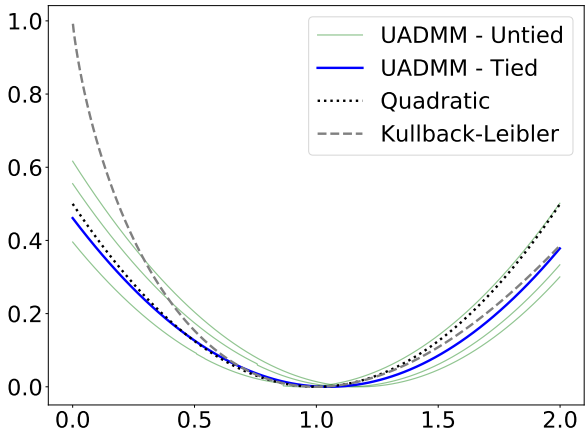


Figure 7: Learned metrics $f_{r,t}(y)$ with $r = 1$. The quadratic cost and Kullback-Leibler divergence $\mathcal{D}_{KL}(y\,|\,r)$ are also displayed for the sake of comparison. In the "tied" case, f_r in analogous to $\mathcal{D}_\psi(\cdot\,|\,r)$ involved in the PR optimization problem. For clarity, only 3 of the 15 trained layers $f_{r,t}$ are displayed for the "untied" case.

7 CONCLUSION AND PERSPECTIVES

7.1 SUMMARY

In this thesis, we studied inverse problems related to the reconstruction of audio signals. We especially focused on phase retrieval, a problem that typically occurs with applications manipulating the spectrogram. It is also notoriously challenging due to its ill-posedness, non-convexity and non-linearity.

This work resulted in several contributions. First, we proposed a novel formulation of the PR problem in which the quadratic cost function is substituted by Bregman divergences, a family of functions that includes divergences considered to be well-suited for audio applications. Two algorithms based on accelerated gradient descent and ADMM were developed and implemented to solve the problem. Experimental work evaluated the methods' performance and underlined the potential of PR with Bregman divergences for audio signal reconstruction from highly-degraded spectrograms.

Second, we extended the previous approach to phase recovery in audio source separation. We took inspiration from the MISI algorithm and considered a novel formulation of the problem that includes an error term based on Bregman divergences and a mixture constraint. We proposed a projected gradient algorithm and assessed experimentally the potential of our method for a speech enhancement task. The proposed method was shown to outperform MISI in this application and with severely degraded spectrograms.

Finally, we proposed to unfold the previously described ADMM algorithm into a deep neural network. We replaced the proximity operators of the Bregman divergences with learnable activation functions and trained the resulting network to perform a PR task with speech signals. We showed that the activation functions trained are proximity operators and characterized their corresponding function. The experimental results revealed that our method outperforms ADMM.

7.2 PERSPECTIVES AND FUTURE WORK

We discuss in the following future research directions and potential extensions of the work presented in this thesis.

First, different optimization algorithms might be used to tackle the phase retrieval problem formulated with Bregman divergences. For example, the majorization-minimization framework might be investigated. The initialization of these algorithms should also be further

explored to get first estimates closer to a local minimum. Number of works examine noniterative methods for phase retrieval [6, 90], which might be combined with the proposed work. Furthermore, regularization should be explored to take into consideration the spectrogram's properties even further (for example, group sparsity in the time-frequency domain [127]). Finally, non-separable divergences could be considered to account for time-frequency correlation.

Besides, further research on the parameterization of the unfolded networks might be undertaken. Attempts to learn the linear operations of the gradient and ADMM algorithms, for example, might be conducted. This would result in learning intermediate representations of the estimated signal. Weight initialization strategies should also be considered, as this procedure was realized empirically in this work but shown to be critical in the deep learning literature [82, 103]. Furthermore, to better account for the properties of the input spectrogram, we could make the network weights dependent from their coordinates in the time-frequency plane or compute them from the input.

Finally, the approaches discussed in this thesis may be extended to various inverse problems involving audio signals. Problems such as audio inpainting, audio declipping and dereverberation, for example, are often solved with proximal methods [72, 159]. The unfolding of these algorithms, as well as the substitution of the proximity operators with trainable activation functions might improve their performances and, for example, learn adequate regularizations.

A

RÉSUMÉ DÉTAILLÉ DE LA THÈSE

INTRODUCTION

La reconstruction de signaux audio consiste à estimer des signaux sonores à partir de représentations incomplètes ou dégradées. Elle permet à l'auditeur une expérience d'écoute améliorée : la qualité sonore perçue sera meilleure et l'information présente plus intelligible.

La reconstruction de signaux audio peut également être considérée comme un problème inverse. C'est-à-dire qu'elle peut être formulée comme l'estimation de paramètres inconnus à partir d'observations et de la connaissance du problème direct. Les problèmes inverses sont fréquemment traités via la minimisation d'une fonction de coût à valeurs réelles, mesurant l'erreur entre les estimations et les observations. Les problèmes dont les solutions ne sont pas existantes, uniques ou stables sont dits mal-posés. Pour résoudre ces derniers, on utilise une connaissance a priori sur les solutions dans diverses stratégies. On pourra alors régulariser le problème, c'est-à-dire modifier la fonction de coût ou encore réduire l'ensemble des solutions.

Dans cette thèse, on propose de modifier la fonction de coût dans les problèmes inverses inhérents à la reconstruction de signaux audio. On considère principalement le problème de reconstruction de phase, un problème fréquent lors de la manipulation de la représentation temps-fréquence la plus courante : le spectrogramme.

CONTEXTE

Le chapitre 2 introduit différents outils issus des domaines d'intérêt de cette thèse. On étudie tout d'abord la représentation des signaux sonores. Les catégories de représentation les plus fréquentes incluent la forme d'onde et les représentations temps-fréquences. La première est une collection de valeurs d'amplitude échantillonées et quantifiées tandis que les secondes présentent des caractéristiques fréquentielles issues de la transformée de Fourier discrète (TFD) dans le temps. Ces dernières sont donc souvent calculées à l'aide de la transformée de Fourier à court-terme (TFCT). La TFCT considère le spectre local d'un signal sur de courtes durées. En pratique, elle est calculée via l'extraction de courts segments temporels du signal suivie du calcul de la TFD. On revient au signal temporel via la TFD inverse et une opération d'addition avec superposition.

À partir de telles représentations, diverses techniques nous permettent d'évaluer objectivement la qualité d'un signal audio. Celles-ci comprennent des calculs de distances dans les domaines temporels et temps-fréquences, ainsi que des scores construits autour de critères perceptuels.

On présente ensuite plusieurs outils issus de l'optimisation. Tout d'abord, le calcul de Wirtinger fournit un cadre nous permettant de dériver les fonctions d'une variable complexe non-différentiables au sens complexe. On peut ainsi calculer un gradient de ces fonctions puis mettre en œuvre un algorithme de descente. Différentes variantes de ce dernier existent via l'accélération ou le calcul de pas de gradient de taille variable. Ensuite, les opérateurs proximaux, des généralisations de l'opérateur de projection, sont utiles dans diverses méthodes d'optimisation récentes. Plusieurs des propriétés de tels opérateurs ont été étudiées dans la littérature et sont présentées dans cette thèse. En particulier, on s'intéressera à la caractérisation de fonctions comme opérateurs proximaux et à leur fonction associée.

Enfin, les divergences de Bregman sont introduites. Cette classe de fonctions inclut différentes divergences bien connues telles que les divergences de Kullback–Leibler et d'Itakura–Saito, les bêta-divergences ou encore la fonction de coût quadratique. Elles peuvent également être interprétées sous le prisme statistique comme des fonctions de log-vraisemblance. Enfin, on s'intéresse aux opérateurs proximaux de plusieurs cas particuliers.

Pour clore ce chapitre, on présente brièvement les réseaux de neurones artificiels, une classe d'algorithmes issus de l'apprentissage automatique et inspirés par le fonctionnement du cerveau. Les réseaux de neurones sont définis par une succession d'opérations linéaires et non-linéaires séparables. L'apprentissage des paramètres des opérations est alors réalisé sur un jeu de données connu au préalable via la minimisation d'une fonction de coût entre les données observées et prédites. On présente ensuite le dépliement d'algorithmes itératifs en réseaux de neurones. Cette stratégie consiste à considérer un nombre fini d'itérations d'un algorithme d'optimisation comme un réseau de neurones dont on apprendra certains paramètres. Enfin, on étudiera la relation entre opérateurs proximaux et fonctions d'activation.

Le chapitre 3 introduit le problème de reconstruction de phase. Tout d'abord, on présente la définition générale du problème, qui consiste à estimer un signal à partir du module d'observations linéaires. Ce dernier peut éventuellement être élevé au carré pour des observations de puissance. Le problème de reconstruction de phase est mal-posé : ses solutions ne peuvent être estimées qu'à plusieurs ambiguïtés près, parmi elles un changement de phase globale.

On présente ensuite différentes méthodes pour la reconstruction de phase. Celles-ci peuvent être classées en deux catégories : les méthodes convexes et non-convexes. Les méthodes non-convexes consis-

tent en la minimisation d'un coût quadratique via différents algorithmes, tels que les projections alternées, la descente de gradient ou encore l'algorithme des directions alternées. Les méthodes convexes transforment le problème en différentes relaxations convexes du problème de reconstruction de phase, traitées via des solveurs issus de l'optimisation semi-définie positive.

Le problème de reconstruction de phase est ensuite étudié dans le champ du traitement du signal audio. Dans ce contexte, les observations sont souvent des spectrogrammes, c'est à dire des modules de TFCT. La reconstruction de phase s'applique alors à différents problèmes tels que le débruitage ou la séparation de sources. Différentes méthodes pour la reconstruction de phase spécifiques à l'audio ont pu être proposées dans la littérature, que l'on peut classer selon deux catégories. La première catégorie comprend les méthodes itératives, qui peuvent être considérés comme les équivalents pour l'audio des méthodes non-convexes vues précédemment. Parmi elles, on étudie l'exemple le plus célèbre : l'algorithme de Griffin–Lim, un algorithme de projections alternées. Cette approche historique peut être interprétée comme un algorithme de descente de gradient avec une fonction de coût quadratique et a généré de multiples variantes, par exemple accélérées ou étendues au temps réel. La seconde catégorie de méthodes considère les caractéristiques spécifiques des signaux audio et de la TFCT. Parmi elles, certaines exploitent par exemple les relations entre les dérivées de la phase et le module de la TFCT. Enfin, on présente le problème de reconstruction de phase pour la séparation de sources sonores.

La dernière section du chapitre présente quelques méthodes issues du champ de l'apprentissage automatique. On s'intéresse tout d'abord aux méthodes d'apprentissage profond considérant le problème de reconstruction de phase en audio. Dans ces approches, la fonction de coût considérée pour l'apprentissage des réseaux de neurones utilise des caractéristiques spécifiques aux signaux audio. Enfin, on présente plusieurs méthodes de dépliement pour les problèmes inverses. Ces dernières introduisent une dimension d'apprentissage dans des algorithmes préalablement construits autour de modèles et donc interprétables.

RECONSTRUCTION DE PHASE AVEC DES DIVERGENCES DE BREGMAN

Le chapitre 4 présente la première contribution de cette thèse. On remplace la fonction de coût quadratique par une divergence de Bregman lorsque le problème de reconstruction de phase est formulé comme un problème de minimisation. En effet, l'optimisation d'un coût quadratique n'est pas nécessairement appropriée pour des signaux audio dans le domaine temps-fréquence : un certain nombre

de travaux de la littérature scientifique établissent l'intérêt d'utiliser des divergences alternatives, telles que les divergences d'Itakura–Saito ou Kullback–Leibler, pour différentes applications telles que la séparation de sources ou la reconstruction de données audio manquantes. Les divergences de Bregman comprennent comme cas particuliers les divergences mentionnées précédemment et s'interprètent d'une perspective statistique. Celles-ci étant non-symmétriques, on propose deux formulations différentes du problème de reconstruction de phase.

Deux algorithmes sont proposés pour traiter les problèmes introduits. Tout d'abord, on considère un algorithme de descente de gradient. La fonction de coût proposée n'étant pas différentiable au sens complexe, on utilise le calcul de Wirtinger pour obtenir l'expression du gradient. Celle-ci dépend de la fonction génératrice de la divergence de Bregman considérée. L'algorithme proposé généralise l'algorithme de Griffin–Lim, qui correspond au cas du coût quadratique et des observations d'amplitude. Ensuite, nous détaillons un algorithme des directions alternées. Celui-ci correspond à la minimisation alternée du Lagrangien augmenté de la fonction de coût proposée. Les itérations de l'algorithme comprennent le calcul d'un opérateur proximal de la divergence de Bregman, dont l'expression en forme close n'est pas toujours explicite.

Une démarche expérimentale évalue les performances de la méthode proposée avec différentes divergences pour deux tâches de reconstruction de phase. Tout d'abord, nous considérons la reconstruction à partir de spectrogrammes exacts. Dans ce contexte, les algorithmes de directions alternées offrent des performances généralement supérieures aux algorithmes de gradient. Pour ces dernières, on obtient avec certaines divergences alternatives des performances similaires à celles des méthodes construites autour de coûts quadratiques. Dans la seconde tâche, nous considérons la reconstruction de phase à partir de spectrogrammes modifiés, dont la non-consistence est simulée par l'ajout de bruit puis le filtrage de Wiener. Dans ce contexte, lorsque les spectrogrammes considérés sont sévèrement dégradés, les algorithmes de gradient considérant des coûts non-quadratiques mènent aux meilleures performances de reconstruction. Enfin, nous considérons l'étude d'algorithmes de gradient avec un pas variable. L'utilisation du pas de Barzilai-Borwein et d'une stratégie de recherche linéaire non-monotone permettent l'amélioration des performances de reconstruction.

RECONSTRUCTION DE PHASE AVEC DES DIVERGENCES DE BREGMAN POUR LA SÉPARATION DE SOURCES AUDIO

Le chapitre 5 étend les résultats du chapitre 4 au problème de séparation de sources audio. Pour traiter ce dernier, une méthodolo-

gie fréquente consiste à estimer les spectrogrammes des différentes sources et d'utiliser la phase du mélange. Dans ce contexte, les techniques de reconstruction de phase permettent l'obtention d'estimations de meilleure qualité. Nous proposons ici une formulation du problème de reconstruction de phase pour la séparation de sources avec des divergences de Bregman. Celle-ci inclut une contrainte de mélange et est traitée à l'aide de l'algorithme du gradient projeté. Comme précédemment, le gradient de la fonction de coût considéré sera calculé via le formalisme du calcul de Wirtinger. L'algorithme proposé généralise l'algorithme d'inversion de spectrogrammes à multiples entrées, qui correspond au cas quadratique.

Une étude expérimentale évalue la méthode proposée pour une tâche de séparation à deux sources. On choisit une opération de débruitage, qui sera pratiquée sur des signaux mélant parole et bruits issus d'environnements acoustiques réels. L'estimation des spectrogrammes des sources est réalisée via Open-Unmix, un réseau de neurones pré-entraîné sur cette tâche. La reconstruction de phase est assurée par notre méthode. L'étude expérimentale confirme le potentiel des méthodes de reconstruction construites à l'aide de divergences non-quadratiques lorsque les spectrogrammes considérés sont sévèrement dégradés. En effet, on pourra observer que la reconstruction de phase avec des bêta-divergences délivre les meilleures performances pour $\beta = 1.25$, lorsque que le cas quadratique correspond à $\beta = 2$.

APPRENTISSAGE D'OPÉRATEURS PROXIMAUX POUR LA RECONSTRUCTION DE PHASE

Le chapitre 6 propose le dépliement de l'algorithme des directions alternées introduit dans le chapitre 4. Cette approche a pour objectif de lever plusieurs difficultés. Tout d'abord, elle permet l'obtention d'une architecture neuronale interprétable en considérant chaque itération d'un algorithme comme une couche neuronale paramétrable. Ensuite, elle permet l'apprentissage des opérateurs proximaux dans ce contexte, dont une expression en forme close n'est pas disponible pour les divergences de Bregman.

L'architecture du réseau proposé considère comme couches les itérations de l'algorithme des descentes alternées pour la reconstruction de phase avec des divergences de Bregman. Chaque opérateur proximal est remplacé par une fonction d'activation paramétrée, dont les poids seront estimés lors d'une phase d'apprentissage. On démontre que les fonctions d'activation proposées sont bien des opérateurs proximaux et on caractérise leur fonction associée. Deux modalités d'apprentissage sont alors étudiées : la première considère des poids «liés », c'est à dire partagés par toutes les couches, et la seconde des poids «déliés », soit libres de prendre des valeurs différentes dans les différentes couches. Dans l'approche aux poids liés, la fonction

métrique associée à l'opérateur proximal appris correspond à la divergence minimisée dans le problème de reconstruction de phase initial. Cette interprétation ne tient pas pour l'approche aux poids déliés, bien que les fonctions associées de chaque couche puissent être caractérisées.

On procède à une étude expérimentale pour évaluer la performance de notre méthode. Le réseau de neurones proposé est soumis à une phase d'apprentissage avec un corpus composé de signaux de parole et de leurs spectrogrammes. Pour un nombre égal d'itérations et de couches, la méthode proposée se montre plus performante que l'algorithme à directions alternées initial. On pourra également remarquer que la méthode aux poids déliés, qui comprend un nombre supérieur de paramètres, se montre plus performante que la méthode aux poids liés pour une tâche de reconstruction de phase. Les deux sont cependant dépassées par la méthode de référence lorsque celle-ci arrive à convergence, pour un grand nombre d'itérations. On comparera alors deux nouvelles architectures, contruites par la duplication des réseaux précédents pré-entraînés, à l'algorithme des descentes alternées. Celles-ci se montreront alors plus efficaces dans la reconstruction et permettront l'emploi d'un grand nombre de couches sans surcoût lié à l'apprentissage d'un grand nombre de paramètres. Enfin, on pourra étudier les fonctions associées aux opérateurs proximaux appris. Ces dernières comportent des ressemblances avec les bêta-divergences étudiées précédemment.

CONCLUSION ET PERSPECTIVES

Pour conclure, le chaptitre 7 dresse un résumé des résultats de cette thèse, suivi d'une discussion portant sur de possibles perspectives de recherche. Parmi celles-ci, on pourra s'intéresser à des algorithmes d'optimisation différents pour la recontruction de phase avec les divergences de Bregman. En particulier, l'étude de l'algorithme de majorisation-minimisation, de stratégies d'initialisation pour l'optimisation non-convexe et de régularisation du problème nous semblent pertinentes. De plus, le travail autour du dépliement des algorithmes itératifs pour la reconstruction de phase nous semble pouvoir être approfondi. On pourra par exemple étudier l'initialisation et le choix des paramètres du réseau. Par exemple, l'apprentissage des opérations linéaires correspondrait à l'apprentissage de représentations des itérées. Enfin, on pourra considérer les méthodes étudiées dans cette thèse pour différents problèmes inverses avec des signaux audio. Par exemple, des problèmes tels que la reconstruction de données manquantes (*inpainting*) ou saturées (*declipping*) et la déréverbération sont souvent traités via des méthodes proximales qui pourraient bénéficier de stratégies de dépliement et d'apprentissage des opérateurs proximaux.

BIBLIOGRAPHY

[1] 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 (2011), pp. 922–932 (cit. on p. 55).

[2] Sercan Ö Arık, Heewoo Jun, and Gregory Diamos. "Fast Spectrogram Inversion Using Multi-head Convolutional Neural Networks." In: *IEEE Signal Processing Letters* 26.1 (2018), pp. 94–98 (cit. on pp. 14, 41, 81).

[3] Larry Armijo. "Minimization of Functions Having Lipschitz Continuous First Partial Derivatives." In: *Pacific Journal of Mathematics* 16.1 (1966), pp. 1–3 (cit. on p. 17).

[4] Jonathan Barzilai and Jonathan M Borwein. "Two-point Step Size Gradient Methods." In: *IMA journal of numerical analysis* 8.1 (1988), pp. 141–148 (cit. on p. 17).

[5] Heinz H. Bauschke, Patrick L. Combettes, et al. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Vol. 408. Springer, 2011 (cit. on pp. 19, 20).

[6] Gerald T Beauregard, Mithila Harish, and Lonce Wyse. "Single Pass Spectrogram Inversion." In: *Proc. IEEE International Conference on Digital Signal Processing (DSP)*. IEEE. 2015, pp. 427–431 (cit. on pp. 39, 92).

[7] John G Beerends, Christian Schmidmer, Jens Berger, Matthias Obermann, Raphael Ullmann, Joachim Pomy, and Michael Keyhl. "Perceptual Objective Listening Quality Assessment (POLQA), the Third Generation ITU-T Standard for End-to-end Speech Quality Measurement Part I—temporal Alignment." In: *Journal of the Audio Engineering Society* 61.6 (2013), pp. 366–384 (cit. on p. 14).

[8] John G Beerends, Christian Schmidmer, Jens Berger, Matthias Obermann, Raphael Ullmann, Joachim Pomy, and Michael Keyhl. "Perceptual Objective Listening Quality Assessment (POLQA), the Third Generation ITU-T Standard for End-to-end Speech Quality Measurement Part II—perceptual Model." In: *Journal of the Audio Engineering Society* 61.6 (2013), pp. 385–402 (cit. on p. 14).

[9] Tamir Bendory, Yonina C Eldar, and Nicolas Boumal. "Non-convex Phase Retrieval from STFT Measurements." In: *IEEE Transactions on Information Theory* 64.1 (2018), pp. 467–484 (cit. on p. 45).

[10] Carla Bertocchi, Émilie Chouzenoux, Marie-Caroline Corbineau, Jean-Christophe Pesquet, and Marco Prato. "Deep Unfolding of a Proximal Interior Point Method for Image Restoration." In: *Inverse Problems* 36.3 (2020), p. 034005 (cit. on p. 25).

[11] Pantelis Bouboulis. "Wirtinger's Calculus in General Hilbert Spaces." In: *CoRR* abs/1005.5170 (2010). URL: http://arxiv.org/abs/1005.5170 (cit. on p. 16).

[12] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers." In: *Foundations and Trends in Machine learning* 3.1 (2011), pp. 1–122 (cit. on pp. 21, 46).

[13] E. Oran Brigham and R. E. Morrow. "The Fast Fourier Transform." In: *IEEE Spectrum* 4.12 (1967), pp. 63–70 (cit. on p. 50).

[14] Richard L Burden, J Douglas Faires, and Annette M Burden. *Numerical Analysis*. Cengage learning, 2015 (cit. on p. 33).

[15] T Tony Cai, Xiaodong Li, and Zongming Ma. "Optimal Rates of Convergence for Noisy Sparse Phase Retrieval via Thresholded Wirtinger Flow." In: *The Annals of Statistics* 44.5 (2016), pp. 2221–2251 (cit. on p. 33).

[16] Emmanuel J Candès, Xiaodong Li, and Mahdi Soltanolkotabi. "Phase Retrieval via Wirtinger Flow: Theory and Algorithms." In: *IEEE Transactions on Information Theory* 61.4 (2014), pp. 1985–2007 (cit. on pp. 32, 33, 45, 46, 53).

[17] Emmanuel J Candès, Thomas Strohmer, and Vladislav Voroninski. "Phaselift: Exact and Stable Signal Recovery from Magnitude Measurements Ia Convex Programming." In: *Communications on Pure and Applied Mathematics* 66.8 (2013), pp. 1241–1274 (cit. on pp. 34, 35).

[18] Estefania Cano, Derry FitzGerald, Antoine Liutkus, Mark D. Plumbley, and Fabian-Robert Stöter. "Musical Source Separation: An Introduction." In: *IEEE Signal Processing Magazine* 36.1 (2019), pp. 31–40 (cit. on p. 74).

[19] Camille Castera, Jérôme Bolte, Cédric Févotte, and Edouard Pauwels. "Second-order Step-size Tuning of SGD for Non-convex Optimization." In: *Neural Processing Letters* 54.3 (2022), pp. 1727–1752 (cit. on pp. 18, 61).

[20] Carlos Eduardo Cancino Chacón and Pejman Mowlaee. "Least Squares Phase Estimation of Mixed Signals." In: *Fifteenth Annual Conference of the International Speech Communication Association*. 2014 (cit. on p. 40).

[21] Pengwen Chen and Albert Fannjiang. "Coded Aperture Ptychography: Uniqueness and Reconstruction." In: *Inverse Problems* 34.2 (2018), p. 025003 (cit. on p. 34).

[22] Pengwen Chen and Albert Fannjiang. "Fourier Phase Retrieval with a Single Mask by Douglas–Rachford Algorithms." In: *Applied and computational harmonic analysis* 44.3 (2018), pp. 665–699 (cit. on p. 34).

[23] Andrzej Cichocki and Shun ichi Amari. "Families of Alpha- Beta- and Gamma- Divergences: Flexible and Robust Measures of Similarities." In: *Entropy* 12.6 (2010), pp. 1532–1568 (cit. on p. 46).

[24] Catherine Colomes, Christian Schmidmer, Thilo Thiede, and William C Treurniet. "Perceptual Quality Assessment for Digital Audio: PEAQ-the New ITU Standard for Objective Measurement of the Perceived Audio Quality." In: *Proc. Audio Engineering Society Conference*. Audio Engineering Society. 1999, pp. 3–29 (cit. on p. 14).

[25] Patrick L Combettes and Jean-Christophe Pesquet. "Proximal Thresholding Algorithm for Minimization Over Orthonormal Bases." In: *SIAM Journal on Optimization* 18.4 (2008), pp. 1351–1376 (cit. on p. 19).

[26] Patrick L Combettes and Jean-Christophe Pesquet. "Proximal Splitting Methods in Signal Processing." In: *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011, pp. 185–212 (cit. on pp. 15, 23, 71, 72).

[27] Patrick L Combettes and Jean-Christophe Pesquet. "Deep Neural Network Structures Solving Variational Inequalities." In: *Set-Valued and Variational Analysis* (2020), pp. 1–28 (cit. on p. 26).

[28] Patrick L. Combettes and Valérie R. Wajs. "Signal Recovery by Proximal Forward-backward Splitting." In: *Multiscale Modeling & Simulation* 4.4 (2005), pp. 1168–1200 (cit. on pp. 83, 85, 86).

[29] Pierre Comon and Christian Jutten. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic press, 2010 (cit. on p. 71).

[30] George Cybenko. "Approximation by Superpositions of a Sigmoidal Function." In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314 (cit. on p. 24).

[31] Yuhong Dai, Jinyun Yuan, and Ya-Xiang Yuan. "Modified Two-point Stepsize Gradient Methods for Unconstrained Optimization." In: *Computational Optimization and Applications* 22.1 (2002), pp. 103–109 (cit. on p. 18).

[32] 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 (cit. on p. 20).

[33] Rémi Decorsière, Peter L Søndergaard, Ewen N MacDonald, and Torsten Dau. "Inversion of Auditory Spectrograms, Traditional Spectrograms, and Other Envelope Representations." In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.1 (2014), pp. 46–56 (cit. on p. 45).

[34] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. "FMA: A Dataset for Music Analysis." In: *arXiv preprint arXiv:1612.01840* (2016) (cit. on pp. 15, 52).

[35] Matrix Differential, Calculus with Applications, to, Simple, Hadamard, and Kronecker Products. "Pii: 0022-2496(85)90006-9." In: (4144) (cit. on pp. 47, 73).

[36] Jonathan Driedger and Meinard Müller. "A Review of Time-scale Modification of Music Signals." In: *Applied Science* 6.57 (2016) (cit. on p. 55).

[37] John Duchi, Elad Hazan, and Yoram Singer. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization." In: *Journal of Machine Learning Research (JMLR)* 12.7 (2011) (cit. on p. 24).

[38] Mireille El Gheche, Giovanni Chierchia, and Jean-Christophe Pesquet. "Proximity Operators of Discrete Information Divergences." In: *IEEE Transactions on Information Theory* 64.2 (2017), pp. 1092–1104 (cit. on pp. 23, 29).

[39] Valentin Emiya, Emmanuel Vincent, Niklas Harlander, and Volker Hohmann. "The PEASS Toolkit-perceptual Evaluation Methods for Audio Source Separation." In: *Proc. International Conference on Latent Variable Analysis and Signal Separation*. 2010 (cit. on p. 15).

[40] Albert Fannjiang and Thomas Strohmer. "The Numerics of Phase Retrieval." In: *Acta Numerica* 29 (2020), pp. 125–228 (cit. on p. 32).

[41] Albert Fannjiang and Zheqing Zhang. "Fixed Point Analysis of Douglas–rachford Splitting for Ptychography and Phase Retrieval." In: *SIAM Journal on Imaging Sciences* 13.2 (2020), pp. 609–650 (cit. on p. 34).

[42] Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu. "Non-negative Matrix Factorization with the Itakura-Saito Divergence: With Application to Music Analysis." In: *Neural Computation* 21.3 (2009), pp. 793–830 (cit. on pp. 22, 40, 46).

[43] Cédric Févotte, Emmanuel Vincent, and Alexei Ozerov. "Single-channel Audio Source Separation with NMF: Divergences, Constraints and Algorithms." In: *Audio Source Separation*. Ed. by S. Makino. Springer, 2018 (cit. on p. 52).

[44] James R Fienup. "Phase Retrieval Algorithms: A Comparison." In: *Applied Optics* 21.15 (1982), pp. 2758–2769 (cit. on pp. 32, 45).

[45] James R. Fienup and J. C. Dainty. "Phase Retrieval and Image Reconstruction for Astronomy." In: *Image recovery: theory and application* 231 (1987), pp. 231–275 (cit. on p. 31).

[46] Derry FitzGerald, Matt Cranitch, and Eugene Coyle. "On the Use of the Beta Divergence for Musical Source Separation." In: *Proc. IET Irish Signals and Systems Conference (ISSC)*. 2008, pp. 1–6 (cit. on p. 88).

[47] Patrick Flandrin. *Time-frequency/Time-scale Analysis*. Academic press, 1998 (cit. on p. 8).

[48] Peter Sadowski Pierre Baldi Forest Agostinelli Matthew Hoffman. "Learning Activation Functions to Improve Deep Neural Networks." In: *Proc. International Conference on Learning Representations (ICLR) workshop*. 2015, pp. 1–9 (cit. on p. 83).

[49] John S. Garofolo, Lori. F. Lamel, W. M. Fisher, Jonathan G. Fiscus, David S. Pallett, and Nancy L. Dahlgren. "TIMIT Acoustic-phonetic Continuous Speech Corpus." In: *Linguistic Data Consortium* (1993) (cit. on pp. 15, 52, 86).

[50] Lonce Wyse Gerald T. Beauregard Xinglei Zhu. "An Efficient Algorithm for Real-time Spectrogram Inversion." In: *Proc. International Conference on Digital Audio Effects (DAFx)*. 2005, pp. 116–118 (cit. on p. 37).

[51] Ralph W Gerchberg and W. O. Saxton. "A Practical Algorithm for the Determination of Plane from Image and Diffraction Pictures." In: *Optik* 35.2 (1972), pp. 237–246 (cit. on pp. 32, 45).

[52] Timo Gerkmann, Martin Krawczyk-Becker, and Jonathan Le Roux. "Phase Processing for Single-channel Speech Enhancement: History and Recent Advances." In: *IEEE Signal Processing Magazine* 32.2 (2015), pp. 55–66 (cit. on pp. 31, 36).

[53] Timo Gerkmann, Martin Krawczyk, and Robert Rehr. "Phase Estimation in Speech Enhancement—Unimportant, Important, or Impossible?" In: *Proc. IEEE Convention of Electrical and Electronics Engineers in Israel*. IEEE. 2012, pp. 1–5 (cit. on p. 8).

[54] Michel X Goemans and David P Williamson. "Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming." In: *Journal of the ACM (JACM)* 42.6 (1995), pp. 1115–1145 (cit. on p. 35).

[55] Robert Gray, Andrés Buzo, Augustine Gray, and Yasuo Matsuyama. "Distortion Measures for Speech Processing." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.4 (1980), pp. 367–376 (cit. on p. 45).

[56] Karol Gregor and Yann LeCun. "Learning Fast Approximations of Sparse Coding." In: *Proc. International Conference on Machine Learning (ICML)*. 2010, pp. 399–406 (cit. on pp. 25, 81).

[57] Rémi Gribonval, Gitta Kutyniok, Morten Nielsen, and Felix Voigtlaender. "Approximation Spaces of Deep Neural Networks." In: *Constructive Approximation* 55.1 (2022), pp. 259–367 (cit. on p. 24).

[58] Rémi Gribonval and Mila Nikolova. "A Characterization of Proximity Operators." In: *Journal of Mathematical Imaging and Vision* 62.6 (2020), pp. 773–789 (cit. on p. 19).

[59] 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 (cit. on pp. 37, 41, 53, 87).

[60] L. Grippo, F. Lampariello, and S. Lucidi. "A Nonmonotone Line Search Technique for Newton's Method." In: *SIAM Journal on Numerical Analysis* 23.4 (1986), pp. 707–716 (cit. on p. 61).

[61] Luigi Grippo, Francesco Lampariello, and Stephano Lucidi. "A Nonmonotone Line Search Technique for Newton's Method." In: *SIAM Journal on Numerical Analysis* 23.4 (1986), pp. 707–716 (cit. on p. 17).

[62] Karlheinz Gröchenig. *Foundations of Time-frequency Analysis*. Springer Science & Business Media, 2001 (cit. on p. 12).

[63] David Gunawan and Deep Sen. "Iterative Phase Estimation for the Synthesis of Separated Sources from Single-channel Mixtures." In: *IEEE Signal Processing Letters* 17.5 (2010), pp. 421–424 (cit. on pp. 41, 71, 74).

[64] Robert W. Harrison. "Phase Problem in Crystallography." In: *Journal of the Optical Society of America A* 10.5 (1993), pp. 1046–1055 (cit. on p. 31).

[65] Romain Hennequin, Roland Badeau, and Bertrand David. "NMF with Time–frequency Activations to Model Nonstationary Audio Events." In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.4 (2011), pp. 744–753 (cit. on p. 46).

[66] Romain Hennequin, Bertrand David, and Roland Badeau. "Beta-divergence As a Subclass of Bregman Divergence." In: *IEEE Signal Processing Letters* 18.2 (2011), pp. 83–86 (cit. on pp. 21, 46, 83).

[67] John R. Hershey, Jonathan Le Roux, and Felix Weninger. "Deep Unfolding: Model-based Inspiration of Novel Deep Architectures." In: *CoRR* abs/1409.2574 (2014). URL: http://arxiv.org/abs/1409.2574 (cit. on pp. 25, 81).

[68] Kurt Hornik. "Approximation Capabilities of Multilayer Feedforward Networks." In: *Neural networks* 4.2 (1991), pp. 251–257 (cit. on p. 24).

[69] Seyed Amir Hossein Hosseini, Burhaneddin Yaman, Steen Moeller, Mingyi Hong, and Mehmet Akçakaya. "Dense Recurrent Neural Networks for Accelerated MRI: History-cognizant Unrolling of Optimization Algorithms." In: *IEEE Journal of Selected Topics in Signal Processing* 14.6 (2020), pp. 1280–1291 (cit. on p. 26).

[70] Rainer Huber and Birger Kollmeier. "PEMO-Q—A New Method for Objective Audio Quality Assessment Using a Model of Auditory Perception." In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.6 (2006), pp. 1902–1911 (cit. on pp. 14, 54).

[71] Diederik P Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization." In: *arXiv preprint arXiv:1412.6980* (2014) (cit. on pp. 24, 86).

[72] Ina Kodrasi, Ante Jukić, and Simon Doclo. "Robust Sparsity-promoting Acoustic Multi-channel Equalization for Speech Dereverberation." In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2016, pp. 166–170 (cit. on p. 92).

[73] Martin Krawczyk and Timo Gerkmann. "STFT Phase Improvement for Single Channel Speech Enhancement." In: *Proc. International Workshop on Acoustic Signal Enhancement (IWAENC)*. 2012, pp. 1–4 (cit. on p. 36).

[74] A Marina Krémé, Valentin Emiya, and Caroline Chaux. "Phase Reconstruction for Time-frequency Inpainting." In: *Proc. International Conference on Latent Variable Analysis and Signal Separation*. Springer. 2018, pp. 417–426 (cit. on p. 36).

[75] Ken Kreutz-Delgado. *The Complex Gradient Operator and the CR-calculus*. 2005 (cit. on p. 16).

[76] Jonathan Le Roux, Hirokazu Kameoka, Nobutaka Ono, and Shigeki Sagayama. "Fast Signal Reconstruction from Magnitude STFT Spectrogram Based on Spectrogram Consistency." In: *Proc. International Conference on Digital Audio Effects (DAFx)*. Vol. 10. 2010, pp. 397–403 (cit. on p. 38).

[77] Jonathan Le Roux, Hirokazu Kameoka, Nobutaka Ono, Alain de Cheveigné, and Shigeki Sagayama. "Computational Auditory Induction As a Missing-data Model-fitting Problem with Bregman Divergence." In: *Speech Communication* 53.5 (2011), pp. 658–676 (cit. on p. 46).

[78] Jonathan Le Roux, Nobutaka Ono, and Shigeki Sagayama. "Explicit Consistency Constraints for STFT Spectrograms and Their Application to Phase Reconstruction." In: *Proc. SAPA INTERSPEECH*. 2008, pp. 23–28 (cit. on pp. 12, 38).

[79] Jonathan Le Roux and Emmanuel Vincent. "Consistent Wiener Filtering for Audio Source Separation." In: *IEEE signal processing letters* 20.3 (2012), pp. 217–220 (cit. on p. 40).

[80] Jonathan Le Roux, Emmanuel Vincent, Yuu Mizuno, Hirokazu Kameoka, Nobutaka Ono, and Shigeki Sagayama. "Consistent Wiener Filtering: Generalized Time-frequency Masking Respecting Spectrogram Consistency." In: *International Conference on Latent Variable Analysis and Signal Separation*. Springer. 2010, pp. 89–96 (cit. on p. 40).

[81] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey. "SDR–Half-baked or Well Done?" In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 626–630 (cit. on p. 13).

[82] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. "Efficient Backprop." In: *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48 (cit. on p. 92).

[83] Yuelong Li, Mohammad Tofighi, Junyi Geng, Vishal Monga, and Yonina C Eldar. "Efficient and Interpretable Deep Blind Image Deblurring via Algorithm Unrolling." In: *IEEE Transactions on Computational Imaging* 6 (2020), pp. 666–681 (cit. on p. 24).

[84] Junli Liang, Petre Stoica, Yang Jing, and Jian Li. "Phase Retrieval via the Alternating Direction Method of Multipliers." In: *IEEE Signal Processing Letters* 25.1 (2018), pp. 5–9 (cit. on pp. 33, 34, 45, 48, 53, 54, 82–84).

[85] Pierre-Louis Lions and Bertrand Mercier. "Splitting Algorithms for the Sum of Two Nonlinear Operators." In: *SIAM Journal on Numerical Analysis* 16.6 (1979), pp. 964–979 (cit. on p. 20).

[86] Antoine Liutkus and Roland Badeau. "Generalized Wiener Filtering with Fractional Power Spectrograms." In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 266–270 (cit. on p. 56).

[87] Antoine Liutkus, Derry Fitzgerald, Zafar Rafii, Bryan Pardo, and Laurent Daudet. "Kernel Additive Models for Source Separation." In: *IEEE Transactions on Signal Processing* 62.16 (2014), pp. 4298–4310 (cit. on p. 40).

[88] Suhas Lohit, Dehong Liu, Hassan Mansour, and Petros T Boufounos. "Unrolled Projected Gradient Descent for Multispectral Image Fusion." In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 7725–7729 (cit. on p. 26).

[89] Paul Magron, Roland Badeau, and Bertrand David. "Phase Reconstruction of Spectrograms Based on a Model of Repeated Audio Events." In: *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE. 2015, pp. 1–5 (cit. on p. 39).

[90] Paul Magron, Roland Badeau, and Bertrand David. "Phase Reconstruction of Spectrograms with Linear Unwrapping: Application to Audio Signal Restoration." In: *Proc. European Signal Processing Conference (EUSIPCO)*. IEEE. 2015, pp. 1–5 (cit. on pp. 36, 39, 92).

[91] Paul Magron, Roland Badeau, and Bertrand David. "Model-based STFT Phase Recovery for Audio Source Separation." In: *IEEE/ACM Transactions on Audio, Speech and Language Processing* 26.6 (2018), pp. 1095–1105 (cit. on pp. 36, 40).

[92] Paul Magron, Konstantinos Drossos, Stylianos Ioannis Mimilakis, and Tuomas Virtanen. "Reducing Interference with Phase Recovery in DNN-based Monaural Singing Voice Separation." In: *Proc. Interspeech*. 2018, pp. 332–336 (cit. on p. 71).

[93] Paul Magron, Jonathan Le Roux, and Tuomas Virtanen. "Consistent anisotropic wiener filtering for audio source separation." In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2017, pp. 269–273 (cit. on p. 40).

[94] Paul Magron, Pierre-Hugo Vial, Thomas Oberlin, and Cédric Févotte. "Phase Recovery with Bregman Divergences for Audio Source Separation." In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 516–520 (cit. on pp. 3, 71).

[95] Paul Magron and Tuomas Virtanen. "Towards Complex Nonnegative Matrix Factorization with the Beta-divergence." In: *Proc. International Workshop on Acoustic Signal Enhancement (IWAENC)*. 2018, pp. 156–160 (cit. on p. 46).

[96] Paul Magron and Tuomas Virtanen. "Online Spectrogram Inversion for Low-latency Audio Source Separation." In: *IEEE Signal Processing Letters* 27 (2020), pp. 306–310 (cit. on pp. 41, 72).

[97] Yoshiki Masuyama, Kohei Yatabe, Yuma Koizumi, Yasuhiro Oikawa, and Noboru Harada. "Deep Griffin–Lim Iteration." In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 61–65 (cit. on p. 54).

[98] Yoshiki Masuyama, Kohei Yatabe, Yuma Koizumi, Yasuhiro Oikawa, and Noboru Harada. "Phase Reconstruction Based on Recurrent Phase Unwrapping with Deep Neural Networks." In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 826–830 (cit. on p. 42).

[99] Yoshiki Masuyama, Kohei Yatabe, and Yasuhiro Oikawa. "Griffin–Lim like Phase Recovery via Alternating Direction Method of Multipliers." In: *IEEE Signal Processing Letters* 26.1 (2019), pp. 184–188 (cit. on pp. 38, 53–55, 57).

[100] Yoshiki Masuyama, Kohei Yatabe, and Yasuhiro Oikawa. "Low-rankness of Complex-valued Spectrogram and Its Application to Phase-aware Audio Processing." In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 855–859 (cit. on p. 42).

[101] Florian Mayer, Donald S. Williamson, Pejman Mowlaee, and DeLiang Wang. "Impact of Phase Estimation on Single-channel Speech Separation Based on Time-frequency Masking." In: *The Journal of the Acoustical Society of America* 141.6 (2017), pp. 4668–4679 (cit. on p. 54).

[102] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. "Librosa: Audio and Music Signal Analysis in Python." In: *Proc. Python in Science Conference*. Vol. 8. 2015, pp. 18–25 (cit. on p. 52).

[103] Dmytro Mishkin and Jiri Matas. "All You Need is a Good Init." In: *arXiv preprint arXiv:1511.06422* (2015) (cit. on p. 92).

[104] Jean-Jacques Moreau. "Fonctions convexes duales et points proximaux dans un espace Hilbertien." In: *Comptes rendus hebdomadaires des séances de l'Académie des sciences* 255 (1962), pp. 2897–2899 (cit. on p. 18).

[105] Jean-Jacques Moreau. "Proximité et dualité dans un espace Hilbertien." In: *Bulletin de la Société mathématique de France* 93 (1965), pp. 273–299 (cit. on p. 19).

[106] Pejman Mowlaee and Rainer Martin. "On Phase Importance in Parameter Estimation for Single-channel Source Separation." In: *Proc. International Workshop on Acoustic Signal Enhancement (IWAENC)*. VDE. 2012, pp. 1–4 (cit. on p. 8).

[107] Pejman Mowlaee, Rahim Saeidi, and Yannis Stylianou. "Advances in Phase-aware Signal Processing in Speech Communication." In: *Speech Communication* 81 (2016), pp. 1–29 (cit. on pp. 31, 36).

[108] Gaurav Naithani, Joonas Nikunen, Lars Bramslow, and Tuomas Virtanen. "Deep Neural Network Based Speech Separation Optimizing an Objective Estimator of Intelligibility for Low Latency Applications." In: *Proc. International Workshop on Acoustic Signal Enhancement (IWAENC)*. 2018, pp. 386–390 (cit. on p. 87).

[109] Rami Nasser, Yonina C Eldar, and Roded Sharan. "Deep Unfolding for Non-negative Matrix Factorization with Application to Mutational Signature Analysis." In: *Journal of Computational Biology* 29.1 (2022), pp. 45–55 (cit. on p. 25).

[110] Yurii E Nesterov. "Semidefinite Relaxation and Nonconvex Quadratic Optimization." In: *Optimization methods and software* 9.1-3 (1998), pp. 141–160 (cit. on p. 35).

[111] Kuldip K Paliwal and Leigh D Alsteris. "On the Usefulness of STFT Phase Spectrum in Human Listening Tests." In: *Speech Communication* 45.2 (2005), pp. 153–170 (cit. on p. 8).

[112] Kuldip K Paliwal and Leigh Alsteris. "Usefulness of Phase Spectrum in Human Speech Perception." In: *Proc. European Conference on Speech Communication and Technology*. Citeseer. 2003, pp. 1–4 (cit. on p. 8).

[113] Manuel Pariente. *PySTOI*. `https://github.com/mpariente/pystoi`. 2018 (cit. on pp. 54, 87).

[114] Manuel Pariente. *PyTorch implementation of STOI*. `https://github.com/mpariente/pytorch_stoi`. [Online; accessed 13-February-2022]. 2021 (cit. on p. 87).

[115] Nathanaël Perraudin, Peter Balazs, and Peter L Søndergaard. "A Fast Griffin–Lim Algorithm." In: *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE. 2013, pp. 1–4 (cit. on pp. 37, 47, 52, 53).

[116] Götz E. Pfander and Palina Salanevich. "Robust Phase Retrieval Algorithm for Time-frequency Structured Measurements." In: *SIAM Journal on Imaging Sciences* 12.2 (2019), pp. 736–761 (cit. on p. 45).

[117] Boris T Polyak. "Some Methods of Speeding up the Convergence of Iteration Methods." In: *Ussr computational mathematics and mathematical physics* 4.5 (1964), pp. 1–17 (cit. on pp. 17, 46, 47).

[118] Michael R Portnoff. "Magnitude-phase Relationships for Short-time Fourier Transforms Based on Gaussian Analysis Windows." In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 4. IEEE. 1979, pp. 186–189 (cit. on p. 39).

[119] Zdeněk Pruša, Peter Balazs, and Peter Lempel Søndergaard. "A Noniterative Method for Reconstruction of Phase from STFT Magnitude." In: *IEEE-ACM Transactions on Audio, Speech, and Language Processing* 25.5 (2017), pp. 1154–1164 (cit. on p. 39).

[120] *PyTorch*. https://pytorch.org/ (cit. on p. 86).

[121] Tianyu Qiu, Prabhu Babu, and Daniel P. Palomar. "PRIME: Phase Retrieval via Majorization-minimization." In: *IEEE Transactions on Signal Processing* 64.19 (2016), pp. 5174–5186 (cit. on pp. 33, 45).

[122] ITU-R Rec. *BS. 1116-3 Methods for the Subjective Assessment of Small Impairments in Audio Systems*. 2015 (cit. on p. 13).

[123] ITU-R Rec. *BS. 1534-3 Method for the Subjective Assessment of Intermediate Quality Level of Coding Systems*. 2015 (cit. on p. 13).

[124] Antony W Rix, Michael P Hollier, John G Beerends, and Andries P Hekstra. "PESQ-the New ITU Standard for End-to-end Speech Quality Assessment." In: *Proc. Audio Engineering Society Convention*. Audio Engineering Society. 2000, pp. 1–18 (cit. on pp. 14, 54).

[125] Herbert Robbins and Sutton Monro. "A Stochastic Approximation Method." In: *The Annals of Mathematical Statistics* (1951), pp. 400–407 (cit. on p. 24).

[126] Frank Rosenblatt. *Principles of Neurodynamics. Perceptrons and the Theory of Brain Mechanisms*. Tech. rep. Cornell Aeronautical Lab Inc Buffalo NY, 1961 (cit. on p. 23).

[127] Kai Siedenburg, Matthieu Kowalski, and Monika Dörfler. "Audio declipping with social sparsity." In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2014, pp. 1577–1581 (cit. on p. 92).

[128] Paris Smaragdis, Cédric Févotte, Gautham J Mysore, Nasser Mohammadiha, and Matthew Hoffman. "Static and Dynamic Source Separation Using Nonnegative Factorizations: A Unified View." In: *IEEE Signal Processing Magazine* 31.3 (2014), pp. 66–75 (cit. on pp. 22, 46, 52).

[129] Julius O. Smith III. *Spectral Audio Signal Processing*. W3K publishing, 2011 (cit. on pp. 11, 13, 50, 52).

[130] Peter Lempel Søndergaard, Bruno Torrésani, and Peter Balazs. "The Linear Time Frequency Analysis Toolbox." In: *International Journal of Wavelets, Multiresolution and Information Processing* 10.4 (2011) (cit. on p. 51).

[131] Fabian-Robert Stöter, Stefan Uhlich, Antoine Liutkus, and Yuki Mitsufuji. "Open-Unmix - a Reference Implementation for Music Source Separation." In: *Journal of Open Source Software* (2019) (cit. on p. 75).

[132] Nicolas Sturmel, Laurent Daudet, et al. "Signal reconstruction from STFT magnitude: A state of the art." In: *International conference on digital audio effects (DAFx)*. 2011, pp. 375–386 (cit. on p. 14).

[133] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. "An Algorithm for Intelligibility Prediction of Time–frequency Weighted Noisy Speech." In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.7 (2011), pp. 2125–2136 (cit. on pp. 14, 54).

[134] Shinnosuke Takamichi, Yuki Saito, Norihiro Takamune, Daichi Kitamura, and Hiroshi Saruwatari. "Phase Reconstruction from Amplitude Spectrograms Based on von Mises Distribution Deep Neural Network." In: *Proc. International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE. 2018, pp. 286–290 (cit. on pp. 42, 81).

[135] Shinnosuke Takamichi, Yuki Saito, Norihiro Takamune, Daichi Kitamura, and Hiroshi Saruwatari. "Phase Reconstruction from Amplitude Spectrograms Based on Directional-statistics Deep Neural Networks." In: *Signal Processing* 169 (2020), p. 107368. ISSN: 0165-1684 (cit. on p. 42).

[136] Lars Thieling, Daniel Wilhelm, and Peter Jax. "Recurrent Phase Reconstruction Using Estimated Phase Derivatives from Deep Neural Networks." In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021, pp. 7088–7092 (cit. on pp. 42, 81).

[137] Joachim Thiemann, Nobutaka Ito, and Emmanuel Vincent. *DEMAND: a collection of multi-channel recordings of acoustic noise in diverse environments*. https://doi.org/10.5281/zenodo.1227121. 2013 (cit. on p. 75).

[138] Nguyen Binh Thien, Yukoh Wakabayashi, Kenta Iwai, and Takanobu Nishiura. "Two-stage Phase Reconstruction Using DNN and von Mises Distribution-based Maximum Likelihood." In: *Proc. Annual Summit and Conference of the Asia-Pacific Signal and Information Processing Association (APSIPA ASC)*. 2021, pp. 995–999 (cit. on p. 81).

[139] Robert Tibshirani. "Regression Shrinkage and Selection via the LASSO." In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288 (cit. on p. 24).

[140] Stefan Uhlich and Yuki Mitsufuji. *Open-unmix for Speech Enhancement (UMX SE)*. https://doi.org/10.5281/zenodo.3786908. 2020 (cit. on p. 75).

[141] Cassia Valentini-Botinhao. *Noisy Speech Database for Training Speech Enhancement Algorithms and TTS Models*. https://doi.org/10.7488/ds/2117. 2017 (cit. on p. 75).

[142] Cassia Valentini-Botinhao, Xin Wang, Shinji Takaki, and Junichi Yamagishi. "Speech Enhancement for a Noise-robust Text-to-speech Synthesis System Using Deep Recurrent Neural Networks." In: *Proc. Interspeech*. 2016, pp. 352–356 (cit. on p. 75).

[143] Pierre-Hugo Vial, Paul Magron, Thomas Oberlin, and Cédric Févotte. "Phase Retrieval with Bregman Divergences: Application to Audio Signal Recovery." In: *Proc. International Traveling Workshop on Interactions between low-complexity data models and Sensing Techniques (iTWIST)*. 2020, pp. 1–2 (cit. on p. 3).

[144] Pierre-Hugo Vial, Paul Magron, Thomas Oberlin, and Cédric Févotte. "Phase Retrieval with Bregman Divergences and Application to Audio Signal Recovery." In: *IEEE Journal of Selected Topics in Signal Processing* 15.1 (2021), pp. 51–64 (cit. on pp. 3, 45).

[145] Pierre-Hugo Vial, Paul Magron, Thomas Oberlin, and Cédric Févotte. "Learning the Proximity Operator in Unfolded ADMM for Phase Retrieval." In: *IEEE Signal Processing Letters* 29 (2022), pp. 1619–1623 (cit. on pp. 3, 81).

[146] Emmanuel Vincent, Nancy Bertin, and Roland Badeau. "Adaptive Harmonic Spectral Decomposition for Multiple Pitch Estimation." In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.3 (2010), pp. 528–537 (cit. on pp. 46, 52).

[147] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. "Performance Measurement in Blind Audio Source Separation." In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.4 (2006), pp. 1462–1469 (cit. on p. 13).

[148] Emmanuel Vincent, Tuomas Virtanen, and Sharon Gannot. *Audio Source Separation and Speech Enhancement*. Wiley, 2018 (cit. on p. 55).

[149] Tuomas Virtanen. "Monaural Sound Source Separation by Nonnegative Matrix Factorization with Temporal Continuity and Sparseness Criteria." In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.3 (2007), pp. 1066–1074 (cit. on p. 46).

[150] Tuomas Virtanen, A Taylan Cemgil, and Simon Godsill. "Bayesian Extensions to Non-negative Matrix Factorisation for Audio Signal Modelling." In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2008, pp. 1825–1828 (cit. on p. 22).

[151] Irène Waldspurger, Alexandre d'Aspremont, and Stéphane Mallat. "Phase Recovery, Maxcut and Complex Semidefinite Programming." In: *Mathematical Programming* 149.1 (2015), pp. 47–81 (cit. on p. 35).

[152] Adriaan Walther. "The Question of Phase Retrieval in Optics." In: *Optica Acta: International Journal of Optics* 10.1 (1963), pp. 41–49 (cit. on p. 31).

[153] DeLiang Wang and Jitong Chen. "Supervised Speech Separation Based on Deep Learning: An Overview." In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26.10 (2018), pp. 1702–1726 (cit. on pp. 40, 71, 74).

[154] Dongxiao Wang, Hirokazu Kameoka, and Koichi Shinoda. "A Modified Algorithm for Multiple Input Spectrogram Inversion." In: *Proc. Interspeech 2019* (2019), pp. 4569–4573 (cit. on p. 41).

[155] Gang Wang, Georgios B Giannakis, and Yonina C Eldar. "Solving Systems of Random Quadratic Equations via Truncated Amplitude Flow." In: *IEEE Transactions on Information Theory* 64.2 (2017), pp. 773–794 (cit. on p. 33).

[156] Zhong-Qiu Wang, Jonathan Le Roux, DeLiang Wang, and John R. Hershey. "End-to-end Speech Separation with Unfolded Iterative Phase Reconstruction." In: *Proc. Interspeech*. 2018, pp. 2708–2712 (cit. on pp. 36, 74, 75).

[157] Zaiwen Wen, Chao Yang, Xin Liu, and Stefano Marchesini. "Alternating Direction Methods for Classical and Ptychographic Phase Retrieval." In: *Inverse Problems* 28.11 (2012), p. 115010 (cit. on pp. 33, 34, 45).

[158] Gordon Wichern and Jonathan Le Roux. "Phase Reconstruction with Learned Time-frequency Representations for Single-channel Speech Separation." In: *Proc. International Workshop on Acoustic Signal Enhancement (IWAENC)*. 2018, pp. 396–400 (cit. on pp. 36, 42, 74, 75).

[159] Pavel Záviška, Pavel Rajmic, Alexey Ozerov, and Lucas Rencker. "A Survey and an Extensive Evaluation of Popular Audio Declipping Methods." In: *IEEE Journal of Selected Topics in Signal Processing* 15.1 (2020), pp. 5–24 (cit. on p. 92).

[160] Kai Zhang, Luc Van Gool, and Radu Timofte. "Deep Unfolding Network for Image Super-resolution." In: *Proc. IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 3217–3226 (cit. on p. 25).

[161] Yan Zhao, Buye Xu, Ritwik Giri, and Tao Zhang. "Perceptually Guided Speech Enhancement Using Deep Neural Networks." In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, pp. 5074–5078 (cit. on p. 87).

[162] Xinglei Zhu, Gerald T Beauregard, and Lonce Wyse. "Real-time Iterative Spectrum Inversion with Look-ahead." In: *Proc. IEEE International Conference on Multimedia and Expo*. IEEE. 2006, pp. 229–232 (cit. on p. 38).

[163] Slawomir Zielinski, Philip Hardisty, Christopher Hummersone, and Francis Rumsey. "Potential Biases in MUSHRA Listening Tests." In: *Proc. Audio Engineering Society Convention*. Audio Engineering Society. 2007, pp. 1–10 (cit. on p. 13).