Unrolled NESTA: Constructing stable, accurate and efficient neural networks for gradient-sparse imaging problems

by

Maksym Neyra-Nesterenko

B.Sc., Simon Fraser University, 2020

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Mathematics
Faculty of Science

SIMON FRASER UNIVERSITY
Spring 2023

Declaration of Committee

Name:	**Maksym Neyra-Nesterenko**
Degree:	**Master of Science**
Thesis title:	**Unrolled NESTA: Constructing stable, accurate and efficient neural networks for gradient-sparse imaging problems**
Committee:	**Chair:** Nadish de Silva Assistant Professor, Mathematics
	Ben Adcock Supervisor Professor, Mathematics
	Nilima Nigam Committee Member Professor, Mathematics
	Ozgur Yilmaz Examiner Professor, Mathematics University of British Columbia

Abstract

Compressive imaging is vital in computational science, engineering and medicine. Its aim is to perform the challenging task of reconstructing images from highly undersampled physical measurements. Deep learning has shown substantial potential to outperform standard techniques for compressive imaging, with empirical evidence indicating superior accuracy. However, deep learning approaches are fraught with many key issues, including hallucinations, instabilities and unpredictable generalization. This motivates a growing body of research to construct accurate neural networks with stability guarantees. In this thesis, we construct stable, accurate and efficient neural networks designed to tackle Fourier imaging problems under a gradient-sparse image model. The networks are constructed by unrolling a novel optimization algorithm based on NESTA, which reconstructs images from undersampled Fourier measurements via TV minimization. To enable fast image reconstruction, we apply a restart scheme which leads to the number of network layers growing logarithmically in the desired image error. Finally, we validate and explore our findings in a series of numerical experiments. The main impact of our work is the construction of neural networks that achieve the same performance guarantees as state-of-the-art handcrafted methods for gradient-sparse imaging.

Keywords: deep learning; compressed sensing; Fourier imaging; unrolling; restart scheme; adversarial perturbations

Dedication

I dedicate this to my mother Nila and father Arturo. Both your love and support throughout my graduate program has been invaluable and my gratitude cannot be expressed in words.

Acknowledgements

I would like to express my deepest gratitude to Ben Adcock for being an excellent research supervisor. He has provided extensive guidance and informative feedback for my thesis and research work, and has been supportive every step of the way. The research program he has paved for me has been incredibly fulfilling, as it is my aspiration to contribute towards the mathematics of machine learning. I am also grateful for the funding I received during the program from the NSERC CGS-M scholarship and Simon Fraser University.

In addition, I am thankful to Matthew Colbrook and Vegard Antun, two authors of a paper for which this thesis is based on. I had the pleasure of collaborating with Matthew to extend a key part of this thesis into a standalone journal article. Vegard provided helpful advice and feedback for developing the numerical experiments of this thesis. I would also like to extend thanks to Nilima Nigam, for recommending me to apply to the master's program at Simon Fraser University to work with Ben. Finally, I wish to express sincere gratitude towards my family and friends, for their support, love and belief in me.

Table of Contents

List of Figures

Chapter 1

Introduction

The main goal of compressive imaging [5] is to reconstruct images from highly undersampled physical measurements. This is a routine task throughout science and engineering, when it is impractical (or impossible) to directly access the structure or object being imaged. In this setting, a physical device (e.g. a medical scanner) acquires measurements for image reconstruction, but due to time, resource or physical constraints, acquisition is limited to a specific number or range of measurements. With limited measurement data, being able to reconstruct images accurately is crucial to advance scientific and industrial development.

In mathematical terms, compressive imaging is typically framed as an inverse problem, stated as

$$\text{Given noisy measurements } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}, \text{ recover the image } \boldsymbol{x}.$$

The operator $\boldsymbol{A}$ refers to the *forward* (or *measurement*) operator, and describes how measurements arise from the image without noise. The term $\boldsymbol{e}$ models corruption of the measurements, which for example, accounts for sensor noise in data acquisition and modeling errors. We consider a discrete version of the above problem where $\boldsymbol{y}, \boldsymbol{e} \in \mathbb{C}^m$, $\boldsymbol{x} \in \mathbb{C}^N$, and the forward operator $\boldsymbol{A} \in \mathbb{C}^{m \times N}$ is a linear map.

What makes compressive imaging challenging is the ill-posed nature of recovery from underdetermined data, i.e. when $m \ll N$. Even in the absence of noise, there can be many images with measurements (derived from the forward model $\boldsymbol{A}$) that are close to $\boldsymbol{y}$. A reliable way to mitigate this ill-posedness is by exploiting a low-dimensional structure within the class of images being recovered. One standard approach is to leverage the *sparsity* of an image under a suitable transformation. Some common examples include sparsity under wavelets, curvelets, shearlets and gradient operators. From here, an optimization problem is cast by encoding the sparsifying transformation as a regularization term and the measurements as a data fidelity term, and then proceeds to be solved by an optimization algorithm. This approach is part of a broader collection of *model-based* (or handcrafted, regularization) imaging methods. *Compressed sensing* [5, 20, 35] provides a well-established theoretical framework that can show accurate and stable recovery via sparsity model-based

methods. This has led to standardized use and acceptance of imaging implementations based on compressed sensing, especially in medical imaging [5, 66, 81].

Throughout this thesis we examine sparsity under the *discrete gradient operator*, referred to as *gradient-sparsity*. The operator provides a sparse model for piecewise constant images and is known to preserve image discontinuities (edges). Many natural images can be viewed as approximately piecewise constant, and thus be approximately sparse under the gradient operator. Moreover, the operator is directly related to *total variation (TV) minimization*, a tool used widely throughout image processing [25, 27] and compressive imaging [4, 53, 64, 81].

1.1 Motivation

1.1.1 Deep learning in imaging

Deep learning and *deep neural networks* have seen a surge in success and popularity within the past decade, with state-of-the-art performance in a multitude of image processing applications. Such prowess has led to the development and application of deep learning to inverse problems in imaging, including compressive imaging [97]. In such a practice, the role of a deep neural network in a reconstruction procedure is flexible, and can be in any part of the image reconstruction pipeline. Deep learning approaches we distinguish are end-to-end and hybrid-based. For end-to-end, the network $\mathcal{N} : \mathbb{C}^m \to \mathbb{C}^N$ is fully trained, e.g. on pairs of images and their measurements. For hybrid-based, both learning and model-based techniques are combined.

The research of deep learning for imaging is growing and rapidly evolving. We do not attempt to provide a comprehensive review. For surveys with emphasis on medical imaging, e.g. Magnetic Resonance Imaging (MRI), X-ray Computed Tomography (CT) and so on, see [16, 51, 59, 60, 63, 78, 81, 82, 86, 96, 97, 100]. For imaging in a broader context, see [5, 8, 62, 65, 77].

1.1.2 Hallucinations, instability, and unpredictable generalization

Contrary to the promising developments of deep learning for inverse problems, there has been increasing concern that deep learning faces several key issues, which arguably inhibit their use in critical applications. These issues are *hallucinations, instabilities and unpredictable generalization*. Such concerns arise despite claims of performance gains via deep learning over state-of-the-art model-based methods.

A hallucination is an image artifact inserted by the reconstruction procedure that is absent in the ground truth image. The concern lies in the fact that hallucinations can appear realistic or physical despite their falsehood. Instability refers to dramatic changes in the reconstructed image from a small perturbation of the neural network input. The image perturbation can be constructed in an adversarial fashion or sometimes from random noise. Unpredictable generalization refers unpredictable behaviour of the reconstruction procedure when tested on data outside of the training set. In the absence of any performance

guarantees, degradation of image reconstruction quality can occur, even on test data near the training set.

The observation of hallucinations in deep learning for medical imaging has been discussed in several papers [15,17,43,44,69,88]. The worries around hallucinations is best summarized in this quote from the authors of the 2020 fastMRI Challenge results [69], regarding the qualitative radiologist evaluation:

> "Such hallucinatory features are not acceptable and especially problematic if they mimic normal structures that are either not present or actually abnormal. ... our results indicate that hallucination and artifacts remain a real concern, particularly at higher accelerations. This topic is in major need for further development."

Similar sentiments are expressed in the other references listed.

The aforementioned issues in imaging were first discussed in [7,46] and explored further recently with varying focus and research direction. For instance, [32,36] investigate instability and generalization of both deep learning and model-based techniques. They found that both approaches are susceptible to the same issues when exposed to worst-case noise and unseen data, suggesting the problem is not unique to deep learning. By using ideas from [10], [30] provides a rigorous computability theoretic treatment of stability and solving inverse problems via neural networks. Contrary to [32,36], experiments in [30,75] suggest robustness to worst-case noise for their respective model-based solvers. Moreover, [7] show that increasing the number of measurements can degrade learning-based reconstruction quality, a phenomena that does not occur in model-based methods based on compressed sensing. [37] describes a framework to rigorously characterize when instability and hallucinations take place in general inverse problems. As with deep learning for inverse problems, the research into their robustness is undergoing rapid and changing development. Some related and general discussion can be found in the previously mentioned work and also in [5, Chap. 20] and [6,41,47,48,52,57,58,61,68,76,78,87,89,93,94,99]. Nonetheless, we wish to emphasize that not all learning-based methods lead to instability. For example, the constructions in [31,42] lead to provably stable neural networks.

We note that many of these issues are not exclusive to imaging. For instance, hallucinations and instability are closely related to the study of *adversarial examples* in machine learning. For this, we refer to the seminal paper [90], a survey [101] and recent theoretical developments [9]. Issues of generalization performance in machine learning is itself a vast topic.

1.2 Contributions and related work

1.2.1 The central question

Holistically, the reliability of deep learning techniques for imaging remains an ongoing debate. The performance gains from deep learning show great potential to improve existing imaging practices and technologies. However, the drawbacks of hallucinations, instability and unpredictable generalization pose a risk when operating in a critical environment. Considering these key issues, this begs the question: *can we construct deep neural networks for compressive imaging with state-of-the-art performance guarantees?* Put another way, can we construct deep neural networks that perform the same as, or better than, state-of-the-art model-based techniques for imaging? Will they be practical to implement or use, e.g. by having small depth and width? These questions motivate an ongoing concerted research effort to better design deep neural networks for imaging with robustness guarantees, and also motivate the effort and results of this thesis.

A first comprehensive attempt to address these questions were done in [30], of which the approach in [5, Chap. 21] is based on. In it, they present computability theoretic results for neural networks and their instability in inverse problems. One of their main results provides sufficient conditions to compute stable neural networks for inverse problems. To achieve this, they consider a model class of images that are sparse in levels under the orthonormal discrete Haar wavelet transform. Then, using arguments leveraging compressed sensing and convex optimization, their constructed network recovers images up to an error in terms of distance to the model class (accuracy), the measurement noise (stability), and a term decaying exponentially in the number of network layers (efficiency). The benefit of efficiency makes the network practical to use for fast image reconstruction. The precise network construction, termed FIRENETs, is done by unrolling a restarted version of Chambolle and Pock's primal-dual iteration [26, 28] configured to solve an ℓ^1-minimization problem.

1.2.2 Main contributions

To summarize, the central contribution of this thesis is the explicit construction of *stable, accurate and efficient neural networks for Fourier imaging under a gradient-sparsity image model.* In particular, Fourier imaging considers measurements represented in a frequency domain associated with the Fourier transform. To construct the network, we unroll a modified version of NESTA (dubbed *stacked NESTA*), a standard optimization algorithm for ℓ^1-minimization [13, 14]. To address a technicality in the recovery analysis, we state and prove Fourier imaging recovery guarantees using a *Bernoulli model* for sampling. In addition, we completely derive stacked NESTA from first principles using tools from convex optimization. In terms of contributions and discussion, we note that there is considerable overlap with a paper we wrote [75], as it is borne out of the same research program! Adhering to [75], we also refer to our unrolled NESTA networks as *NESTANets*.

We extend the previous work [30] in two ways. First, rather than the Haar wavelet, we consider a model class of images that are gradient-sparse, i.e. sparse under the discrete gradient operator. Despite both yielding sparse representations for piecewise smooth images, this extension is notable since the gradient operator is not invertible. Therefore, a more sophisticated analysis is needed to ensure image recovery, which we adapt from [4,5]. Second, analogous to [75], to construct the network we unroll (stacked) NESTA. We use a restart procedure that modifies the NESTA smoothing parameter to grow the unrolled network depth logarithmically in the desired image error.

1.2.3 Discussion on related topics

The reasons to consider Fourier imaging are twofold. The first pertains to practical impact and relevance, where Fourier measurements for gradient-sparse imaging arise in medical imaging, namely MRI [4,64,81]. With this, we stay close to the topic of deep learning's impact in imaging applications. The second reason is theoretical, where recovery from Fourier measurements via compressed sensing is both well-understood and documented. As part of the technical analysis, it is also convenient to work with both the Fourier matrix and gradient operator.

To study recovery of gradient-sparse images, we consider a reconstruction procedure based on TV minimization, a convex optimization problem (see Section 2.3.4) which specifies ℓ^1-minimization of the image gradient as a regularizer. TV minimization is used throughout compressive imaging [4,20,53,64,81] and was first used for compressed sensing in [20], where they also considered Fourier measurements. In our image recovery analysis, we take advantage of a connection between the TV semi-norm and Haar wavelet coefficients that lead to a Poincaré inequality. This idea was first used in [71,72] to show recovery of gradient-sparse signals. To obtain recovery of both the image and its gradient, we consider a structured sampling scheme that combines uniform random samples with variable-density samples. This was first considered and used in [80]. Finally, many of the tools and arguments we use to prove recovery guarantees are based on [4] and [5, Chap. 17].

As in [30], our network construction involves *unrolling* an optimization algorithm. Unrolling has been an important architectural design for neural networks in inverse problems, yielding some of the best-performing networks. For references on the topic, a good starting point is [67], and also [5, Chap. 21] and [8,60,65,81]. It should be noted that unrolling on its own is not sufficient to guarantee stability of deep learning techniques [37]. Prior to [75], NESTA has not been considered in unrolling schemes.

Restart schemes are an algorithmic framework for accelerating convergence of optimization algorithms. For example, see [3,83,85] and for restarting based on primal-dual iteration, see [29]. In particular, our restart procedure for NESTA extends the one in [85] by considering inexact sparsity and noisy measurements, as opposed to the simpler case of exact sparsity and exact measurements therein.

In comparison to [75], there are novel components exclusive to the presentation here. First, the gradient operator does not form a frame, so our image model class is not in the scope of frame analysis operators in [75]. Second, we provide a compressed sensing analysis of the recovery guarantees in full detail. Third, we use a Bernoulli model for sampling [5, Sec. 11.4.3] that leads to a stacked sampling scheme compatible with an efficient, and practical to unroll, implementation of NESTA. The implementation gives rise to a modified version of NESTA, giving the moniker "stacked NESTA" from the stacking scheme. In compressed sensing, Bernoulli models for sampling schemes appear in previous work, e.g. [20, 80, 84, 91]. To the best of our knowledge, we are the first to consider a Bernoulli model for variable density sampling in Fourier imaging, moreover with inexact sparsity and noisy measurements. Fourth, we provide a complete derivation of stacked NESTA, including a rigorous translation of the algorithm over real-valued data to complex values. All these topics include technical discussion of related work and practical considerations. Lastly, the experiments are updated to give more insight on discussion lacking in [75], such as parameter tuning of δ and the choice of decay factor r.

1.3 Main results

To the state the main result, we need to introduce some notation. For brevity, we defer explicit definitions of some items (e.g. the Fourier matrix) to subsequent chapters. Let $[\![M]\!] = \{1, \dots, M\}$ and for vector $\boldsymbol{u} = (u_i)_{i=1}^M \in \mathbb{C}^M$, denote $\boldsymbol{u}_S$ as the vector with ith entry equal to u_i if $i \in S$ and zero otherwise.

We say a vector is *s-sparse* if at most s of its entries are nonzero. In practice, images are often only *approximately sparse* under a specific transformation. Let us make this notion precise. The *ℓ^1-norm best s-term approximation error of $\boldsymbol{z} \in \mathbb{C}^M$* is the number

$$\sigma_s(\boldsymbol{z})_{\ell^1} = \min\left\{\|\boldsymbol{z} - \boldsymbol{u}_S\|_{\ell^1} : \boldsymbol{u} \in \mathbb{C}^M,\ S \subseteq [\![M]\!],\ |S| \leq s\right\}.$$

Observe that the minimization effectively is taken over the set of s-sparse vectors, and the minima are uniquely determined by taking the s largest components of $\boldsymbol{z}$ in absolute value. Intuitively, $\boldsymbol{z}$ is approximately s-sparse provided that $\sigma_s(\boldsymbol{z})_{\ell^1}$ is sufficiently small.

Now we define relevant vectors and matrices. The d-dimensional images considered are expressed as vectors $\boldsymbol{x} \in \mathbb{C}^{N^d}$. Let $\boldsymbol{F} \in \mathbb{C}^{N^d \times N^d}$ denote the d-dimensional Fourier matrix and $\boldsymbol{V} \in \mathbb{R}^{dN^d \times N^d}$ denote the d-dimensional discrete gradient operator. For their definitions, see Sections 2.3.2 and 2.3.4. To express subsampling rows, let $\{\boldsymbol{e}_i\}_{i=1}^N$ denote the standard basis of $\mathbb{C}^N$, and $\Omega \subseteq [\![N]\!]$ a set of indices. We denote $\boldsymbol{P}_\Omega \in \mathbb{C}^{N \times N}$ as the orthogonal projection onto the linear span of $\{\boldsymbol{e}_i : i \in \Omega\}$. Namely, for $\boldsymbol{x} = (x_i)_{i=1}^N \in \mathbb{C}^N$, the jth entry of $\boldsymbol{P}_\Omega \boldsymbol{x}$ is equal to x_j if $j \in \Omega$, and zero otherwise. We occasionally consider the $|\Omega| \times N$ matrix formed from the nonzero rows of $\boldsymbol{P}_\Omega$. Abusing notation, we continue to write such a matrix as $\boldsymbol{P}_\Omega$.

Now we define the model class of Fourier measurements we seek to recover from gradient-sparse images. Fix $d \geq 1$, $\eta > 0$, $1 \leq s \leq N^d$ and define

$$\mathcal{CS}_{s,d}(\boldsymbol{V}\boldsymbol{x}, \eta) = \frac{\sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1}}{\sqrt{s}} + \left(d + \sqrt{\log(N)}\right)\eta.$$

Given $\chi > 0$, we write

$$\mathbb{I} = \mathbb{I}_{\boldsymbol{V},\chi,\eta} = \left\{(\boldsymbol{x}, \boldsymbol{e}) \in \mathbb{C}^{N^d} \times \mathbb{C}^m : \|\boldsymbol{x}\|_{\ell^2} \leq 1,\ \|\boldsymbol{e}\|_{\ell^2} \leq \eta,\ \mathcal{CS}_{s,d}(\boldsymbol{V}\boldsymbol{x}, \eta) \leq \chi\right\},$$

and define the class of measurements

$$\mathbb{M} = \mathbb{M}_{\boldsymbol{A},\boldsymbol{V},\chi,\eta} = \left\{\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} \in \mathbb{C}^m : (\boldsymbol{x}, \boldsymbol{e}) \in \mathbb{I}_{\boldsymbol{V},\chi,\eta}\right\}. \tag{1.3.1}$$

The interpretation of $\mathbb{M}$ is that it defines noisy measurement vectors $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$ from images $\boldsymbol{x} \in \mathbb{C}^{N^d}$ that are approximately gradient-sparse, i.e. $\sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1}/\sqrt{s} \leq \chi$, and noise vectors $\boldsymbol{e}$ with bounded ℓ^2-norm, i.e. $\|\boldsymbol{e}\|_{\ell^2} \leq \eta \leq \chi$.

Finally, we use the notation $\mathcal{O}(\cdot)$ and $\mathcal{O}_d(\cdot)$ to both refer to standard big-O notation, except the latter indicates that the constant factor depends on d. With this, we can state the main result of the thesis.

Theorem 1.3.1 (Stable, accurate and efficient neural networks for Fourier imaging)**.** *Let $d \geq 1$, $0 < \epsilon < 1$, $2 \leq s, m \leq N^d$, and $\boldsymbol{A} = \frac{1}{\sqrt{m}}\boldsymbol{P}_\Omega\boldsymbol{F} \in \mathbb{C}^{|\Omega| \times N^d}$ be a subsampled d-dimensional Fourier matrix with sampling mask Ω. Suppose $\eta \geq 0$ and $\chi > 0$ and consider the class $\mathbb{M} = \mathbb{M}_{\boldsymbol{A},\boldsymbol{V},\chi,\eta}$. Then for a suitable random sampling scheme defining Ω with $\mathbb{E}(|\Omega|) \asymp m$, the following holds with probability at least $1 - \epsilon$, provided that*

$$m \gtrsim_d s \cdot \mathrm{polylog}(N, s, \epsilon^{-1}).$$

For every $k \geq 1$, one can construct a neural network $\mathcal{N} : \mathbb{C}^{|\Omega|} \to \mathbb{C}^{N^d}$ (NESTANet) such that for all $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} \in \mathbb{M}$, we have

$$\|\boldsymbol{x} - \mathcal{N}(\boldsymbol{y})\|_{\ell^2} \lesssim_d \chi + \mathrm{e}^{-k}, \qquad d \geq 2,$$

and

$$\frac{\|\boldsymbol{x} - \mathcal{N}(\boldsymbol{y})\|_{\ell^2}}{\sqrt{N}} \lesssim \chi + \mathrm{e}^{-k}, \qquad d = 1.$$

In both cases, i.e. $d \geq 1$, the network depth is $\mathcal{O}_d\left(\sqrt{\frac{N^d}{s}} \cdot k\right)$ and the width is $\mathcal{O}\left(dN^d\right)$.

For ease of presentation, both descriptions of the sampling scheme and NESTANet architecture are omitted. Their constructions are provided in Chapters 3 and 5, respectively. The precise technical version of the above theorem is given in Theorems 5.3.1 and 5.3.2, to-

gether with their proofs. Each correspond to separate cases of $d = 1$ and $d \geq 2$, respectively. Note that we also abused notation in Theorem 1.3.1 when defining the subsampled Fourier matrix $\boldsymbol{A}$, since the frequencies indexed by Ω are actually sampled exactly twice. That being said, there are several points to mention about Theorem 1.3.1, which by extension also apply to Theorems 5.3.1 and 5.3.2.

First, the error bound tells us that the image reconstruction is guaranteed to be within an error proportional to χ and a term decaying exponentially in k. Choosing $k = \lceil |\log(\chi)| \rceil$ yields a network that can perform image reconstruction within an error proportional to the desired error χ. This is the efficiency of our network construction, where to guarantee reconstruction within error proportional to χ, the network depth should scale logarithmically in χ. This is precisely analogous to [30, Thm. 4] and [75, Thm. 1]. Moreover, we remark that the network construction in [30, Thm. 3] has a depth proportional to np layers, where n is the restart number and $p \propto \|\boldsymbol{A}\|_{\ell^2}$. This is comparable to the number of layers we use. To see how, if $\boldsymbol{A} \in \mathbb{C}^{m \times N^d}$ has the *restricted isometry property*, a condition frequently used to construct measurements matrices in compressed sensing, then $\|\boldsymbol{A}\|_{\ell^2} \lesssim \sqrt{N^d/s}$ by [5, Rem. 8.8]. The same can be said of the NESTANets in [75].

1.4 Outline

The thesis is organized as follows. In Chapter 2, we provide background definitions and notation needed to navigate the thesis. Chapter 3 is a technical chapter specifying and proving the sufficient conditions needed to reconstruct images from subsampled Fourier measurements. From here, we lead into Chapter 4 which provides a complete derivation of stacked NESTA for Fourier imaging via TV minimization. In addition, we prove associated recovery guarantees for the iterates of NESTA. The resulting recovery properties inform a restart procedure for NESTA, which we show theoretically accelerates image reconstruction. Next, in Chapter 5 we detail the construction of NESTANets (unrolled stacked NESTA) and prove the main result. Chapter 6 showcases the numerical experiments with NESTANets in the setting of a 2-D Fourier imaging task. Much of these experiments are designed to verify or reflect our findings, and offer insight to bridge any potential gap between theory and practice. Finally, we conclude in Chapter 7 by summarizing key aspects of the thesis and offer ideas for future work.

Chapter 2

Background and preliminaries

In this chapter, we provide the necessary terminology, notation, and background needed to navigate most of this thesis. The topics discussed include compressed sensing and its relation to inverse problems, convex optimization, Fourier imaging and TV minimization. The reader interested in any of these topics can refer to the citations provided here and in Chapter 1.

2.1 Compressed sensing for inverse problems

The types of inverse problems we consider are *discrete linear inverse problems*, taking the form

$$\text{Given measurements } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} \in \mathbb{C}^m \text{ and } \|\boldsymbol{e}\|_{\ell^2} \leq \eta, \text{ recover } \boldsymbol{x} \in \mathbb{C}^N. \tag{2.1.1}$$

Here $\boldsymbol{A} \in \mathbb{C}^{m\times N}$ is the measurement (or forward) matrix, $\boldsymbol{e} \in \mathbb{C}^m$ is a noise vector, and $\eta > 0$ is the *noise level*, a parameter bounding the ℓ^2-norm of the noise $\boldsymbol{e}$. We assume $m \ll N$, so one has a highly underdetermined linear system with noise, making the general problem of recovering $\boldsymbol{x}$ ill-posed. The way we go about establishing recovery guarantees for (2.1.1) is leveraging *compressed sensing theory* [5, 35]. For this, we need to introduce several definitions and notation. Some of these were already introduced in Section 1.3, but we include them here for completeness.

Let $[\![M]\!] = \{1, \dots, M\}$ and for vector $\boldsymbol{u} = (u_i)_{i=1}^M \in \mathbb{C}^M$, denote $\boldsymbol{u}_S$ as the vector with ith entry equal to u_i if $i \in S$ and zero otherwise. We say a vector is *s-sparse* if at most s of its entries are nonzero. The notion of approximate sparsity is captured in the following definition.

Definition 2.1.1 (Best s-term approximation error)**.** The *ℓ^1-norm best s-term approximation error of $\boldsymbol{z} \in \mathbb{C}^M$* is the number

$$\sigma_s(\boldsymbol{z})_{\ell^1} = \min\left\{\|\boldsymbol{z} - \boldsymbol{u}_S\|_{\ell^1} : \boldsymbol{u} \in \mathbb{C}^M,\ S \subseteq [\![M]\!],\ |S| \leq s\right\}.$$

$\Diamond$

The minimization is effectively taken over the set of s-sparse vectors, hence $\sigma_s(\boldsymbol{z})_{\ell^1} = \|\boldsymbol{z} - \boldsymbol{z}_S\|_{\ell^1}$ where S are indices of the s largest components of $\boldsymbol{z}$ in absolute value. Also, $\sigma_s(\boldsymbol{z})_{\ell^1} = 0$ whenever $\boldsymbol{z}$ is s-sparse. We consider solutions of (2.1.1) where the vector $\boldsymbol{W}^*\boldsymbol{x} \in \mathbb{C}^M$ is *approximately sparse*, i.e. $\sigma_s(\boldsymbol{W}^*\boldsymbol{x})_{\ell^1}$ is sufficiently small for some $\boldsymbol{W} \in \mathbb{C}^{N\times M}$. The matrix $\boldsymbol{W}$ is dubbed the *analysis* matrix, and we refer to such solutions $\boldsymbol{x}$ as being *approximately analysis-sparse* with respect to analysis matrix $\boldsymbol{W}$. Two prevalent examples of $\boldsymbol{W}$ used in imaging are discrete wavelet transforms and the discrete gradient operator [5].

We now state standard definitions from compressed sensing used to prove recovery guarantees.

Definition 2.1.2 (Robust null space property [5, Defn. 5.14])**.** The matrix $\boldsymbol{A} \in \mathbb{C}^{m\times N}$ satisfies the *robust Null Space Property (rNSP)* with constants $0 < \rho < 1$ and $\gamma > 0$ if

$$\|\boldsymbol{v}_S\|_{\ell^2} \leq \frac{\rho}{\sqrt{s}}\|\boldsymbol{v}_{S^\complement}\|_{\ell^1} + \gamma\|\boldsymbol{A}\boldsymbol{v}\|_{\ell^2},$$

for all $\boldsymbol{v} \in \mathbb{C}^N$ and $S \subseteq [\![N]\!]$ with $|S| \leq s$. $\Diamond$

The intuition behind the rNSP condition is that it ensures the difference of two approximately sparse vectors is not close to the null space of $\boldsymbol{A}$ (see e.g. [5, Chap. 5]). This provides a notion of well-posedness when recovering an approximately sparse signal, informed by the following general inequalities.

Lemma 2.1.3 (rNSP implies ℓ^1 and ℓ^2 distance bounds [5, Lem. 5.15, 5.16])**.** *Suppose that $\boldsymbol{A} \in \mathbb{C}^{m\times N}$ has the rNSP of order s with constants $0 < \rho < 1$ and $\gamma > 0$. Let $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{C}^N$. Then*

$$\|\boldsymbol{z} - \boldsymbol{x}\|_{\ell^1} \leq \frac{1+\rho}{1-\rho}(2\sigma_s(\boldsymbol{x})_{\ell^1} + \|\boldsymbol{z}\|_{\ell^1} - \|\boldsymbol{x}\|_{\ell^1}) + \frac{2\gamma}{1-\rho}\sqrt{s}\|\boldsymbol{A}(\boldsymbol{z}-\boldsymbol{x})\|_{\ell^2},$$

and

$$\|\boldsymbol{z} - \boldsymbol{x}\|_{\ell^2} \leq \frac{(3\rho+1)(\rho+1)}{2(1-\rho)}\left(\frac{2\sigma_s(\boldsymbol{x})_{\ell^1} + \|\boldsymbol{z}\|_{\ell^1} - \|\boldsymbol{x}\|_{\ell^1}}{\sqrt{s}}\right) + \frac{(3\rho+5)\gamma}{2(1-\rho)}\|\boldsymbol{A}(\boldsymbol{z}-\boldsymbol{x})\|_{\ell^2}.$$

In practice, it is difficult to show matrices have the rNSP. A related stronger condition implying the rNSP is often used instead.

Definition 2.1.4 (Restricted isometry property (RIP) [5, Defn. 5.18])**.** Let $1 \leq s \leq N$. The sth *Restricted Isometry Constant (RIC)* δ_s of a matrix $\boldsymbol{A} \in \mathbb{C}^{m\times N}$ is the smallest $\delta \geq 0$ such that

$$(1-\delta)\|\boldsymbol{v}_S\|_{\ell^2}^2 \leq \|\boldsymbol{A}\boldsymbol{v}_S\|_{\ell^2}^2 \leq (1+\delta)\|\boldsymbol{v}_S\|_{\ell^2}^2,$$

for all $\boldsymbol{v} \in \mathbb{C}^N$ and $S \subseteq [\![N]\!]$ with $|S| \leq s$. If $0 < \delta_s < 1$ then $\boldsymbol{A}$ is said to have the *Restricted Isometry Property (RIP) of order s*. $\Diamond$

Lemma 2.1.5 (RIP implies rNSP [5, Lem. 5.20])**.** *Suppose that $\boldsymbol{A} \in \mathbb{C}^{m \times N}$ satisfies the RIP of order $2s$ with constant $\delta_{2s} < \sqrt{2} - 1$. Then $\boldsymbol{A}$ satisfies the rNSP of order s with constants*

$$\rho = \frac{\sqrt{2}\delta_{2s}}{1 - \delta_{2s}}, \qquad \gamma = \frac{\sqrt{1 + \delta_{2s}}}{1 - \delta_{2s}}.$$

We note there are many variations of the RIC condition $\delta_{2s} < \sqrt{2} - 1$ in Lemma 2.1.5, e.g. see the Notes sections of [5, Chap. 5] and [35, Chap. 6]. The references therein sometimes go from the RIP directly to sparse recovery, rather than using the rNSP as an intermediary step.

This concludes the basic tools and terminology needed to study recovery guarantees for inverse problems with analysis sparsity. The next key step is to consider the problem formulation for solving (2.1.1) via computational methods.

2.2 Sparse recovery via convex optimization

To compute a solution to the inverse problem (2.1.1), we formulate and solve the constrained convex optimization problem

$$\min_{\boldsymbol{z} \in \mathbb{C}^N} \|\boldsymbol{W}^* \boldsymbol{z}\|_{\ell^1} \text{ subject to } \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}\|_{\ell^2} \leq \eta. \tag{2.2.1}$$

We refer to (2.2.1) as *Quadratically Constrained Basis Pursuit (QCBP)*. QCBP is a standard problem formulation of sparse recovery, where ℓ^1-norm minimization is known to promote sparse solutions [5, Sec. 5.4] (in this case, sparse in the analysis domain) and the constraint directly encodes our noise assumption $\|\boldsymbol{e}\|_{\ell^2} \leq \eta$.

The QCBP problem cast in (2.2.1) is typically known as the *analysis* problem or formulation. This is distinguished from the *synthesis* formulation, which is defined by

$$\min_{\boldsymbol{c} \in \mathbb{C}^M} \|\boldsymbol{c}\|_{\ell^1} \text{ subject to } \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{W}\boldsymbol{c}\|_{\ell^2} \leq \eta.$$

The analysis problem optimizes over the signal domain whereas the synthesis problem optimizes over the analysis domain. Both formulations are generally not equivalent. Since we model gradient-sparse signals by considering $\boldsymbol{W}^*$ as the discrete gradient operator, we consider the analysis formulation in this thesis. An in-depth look at the differences between analysis and synthesis can be found in [34, 70].

To ensure recovery of $\boldsymbol{x}$, one standard condition is to consider analysis matrices $\boldsymbol{W}$ whose columns form a *frame* of $\mathbb{C}^N$. This holds if there exist constants $\beta \geq \alpha > 0$ such that

$$\alpha \|\boldsymbol{x}\|_{\ell^2}^2 \leq \|\boldsymbol{W}^* \boldsymbol{x}\|_{\ell^2}^2 \leq \beta \|\boldsymbol{x}\|_{\ell^2}^2.$$

This is a central assumption to the theoretical development of NESTANets in [75]. The assumption also appeals to the interest and use of frames throughout compressive imaging and sensing, e.g. wavelet frames [33], curvelets [21–23] and shearlets [39, 40, 55, 56]. Observe that a necessary condition for the columns of $\boldsymbol{W}$ forming a frame is $M \geq N$ and $\boldsymbol{W}$ is full rank. The optimal values of α and β depend on the minimum and maximum singular values of $\boldsymbol{W}$, respectively, via $\alpha = (\sigma_N(\boldsymbol{W}))^2$ and $\beta = (\sigma_1(\boldsymbol{W}))^2 = \|\boldsymbol{W}\|_{\ell^2}^2$. When the columns of $\boldsymbol{W}$ do not form a frame, a more sophisticated analysis is needed to guarantee recovery of $\boldsymbol{x}$. We do this in the context of Fourier inverse problems and TV minimization in Chapter 3, being one of our main contributions in this thesis. TV minimization refers to (2.2.1) when $\boldsymbol{W}^*$ is the discrete gradient operator. In this case, $\boldsymbol{W}$ does not form a frame since it is not a full rank matrix. Fourier inverse problems, the discrete gradient operator, and TV minimization are the topics of the next two sections.

To solve QCBP problems we use an adaptation of *NESTerov's Algorithm (NESTA)*, an accelerated projected gradient method with smoothing [13, 14, 73]. Explicitly defining our version of NESTA, deriving it, and proving error bounds is the primary topic of Section 4.1. As discussed in Section 1.2.3, the use of NESTA for unrolling is unexplored. Many of the developments throughout this thesis can proceed with other optimization algorithms, as has been done several times with Chambolle and Pock's primal-dual iteration [2, 29, 30].

2.3 Fourier imaging

The main problem we tackle throughout this thesis pertains to Fourier imaging. Fourier measurements are a popular model for several imaging modalities, such as Magnetic Resonance Imaging (MRI) [5, 64, 81], Nuclear Magnetic Resonance (NMR) [45, 50], radio interferometry [98] and Helium Atom Scattering (HAS) [49]. Here the inverse problem (2.1.1) is cast with $\boldsymbol{A}$ being a randomly subsampled discrete Fourier transform. The vector $\boldsymbol{x}$ being recovered is a vectorized version of the image. For theoretical results, we present them and their proofs in terms of general d-dimensional images. For this, it is convenient to adopt the notation and definitions from [4, Sec. 2].

2.3.1 Tensors

A d-dimensional complex image $\boldsymbol{X}$ is the d-dimensional tensor

$$\boldsymbol{X} = (X_{i_1,\dots,i_d})_{i_1,\dots,i_d=1}^{N}$$

where the entries are in $\mathbb{C}$. It is mathematically useful to reshape (or vectorize) $\boldsymbol{X}$ into a vector. This can be done using a lexicographical ordering, where given a bijection ς :

$\{1,\dots,N^d\} \to \{1,\dots,N\}^d$, the ordering is then defined by the inverse mapping

$$\varsigma^{-1}(i_1,\dots,i_d) = 1 + \sum_{j=1}^{d} N^{d-j}(i_j - 1), \qquad (i_1,\dots,i_d) \in \{1,\dots,N\}^d.$$

With this, we denote the vectorization of $\boldsymbol{X}$ as $\boldsymbol{x} = \mathrm{vec}(\boldsymbol{X}) \in \mathbb{C}^{N^d}$, where the ith entry of $\boldsymbol{x}$ is $X_{\varsigma(i)}$. The lexicographical ordering we defined above is sometimes referred to as *column-major order*. From this point on, we avoid the use tensors to express images and generally refer to their vectorization.

2.3.2 Discrete Fourier transform

Throughout this thesis, whenever the discrete Fourier transform arises it is assumed that N is a power of two, i.e. $N = 2^R$ for some positive integer R. Now, given the bijection $\varrho : \{1,\dots,N\} \to \{-N/2+1,\dots,N/2\}$ defined by $i \mapsto (-1)^i \lfloor i/2 \rfloor$, we define the *one-dimensional discrete Fourier transform (DFT)* as the matrix $\boldsymbol{F} \in \mathbb{C}^{N\times N}$ with entries

$$(\boldsymbol{F})_{ij} = \exp(-2\pi \mathrm{i} \varrho(i)(j-1)/N), \qquad i,j \in [\![N]\!].$$

The *d-dimensional DFT* $\boldsymbol{F}^{(d)} \in \mathbb{C}^{N^d \times N^d}$ is given by

$$\boldsymbol{F}^{(d)} = \underbrace{\boldsymbol{F} \otimes \cdots \otimes \boldsymbol{F}}_{d \text{ times}},$$

where $\otimes$ is the Kronecker product. Throughout we abuse notation and write $\boldsymbol{F} = \boldsymbol{F}^{(d)}$, where the dimension is inferred from context. Note that we have $\boldsymbol{F}\boldsymbol{F}^* = \boldsymbol{F}^*\boldsymbol{F} = N^d \boldsymbol{I}$.

The range of $\boldsymbol{F}$ naturally corresponds to images represented in frequency space. Using the bijection ϱ, we can associate the d-dimensional Fourier matrix row indices with frequencies using the bijection

$$\begin{aligned} &\varrho^{(d)} : \{1,\dots,N^d\} \to \{-N/2+1,\dots,N/2\}^d, \\ &\varrho^{(d)}(i) = \left(\varrho(\varsigma(i)_1),\dots,\varrho(\varsigma(i)_d)\right), \qquad i \in \{1,\dots,N^d\}. \end{aligned}$$

In similar fashion to [4], when we visualize sampling masks, we shift the zero-frequency component of the discrete Fourier transform to the centre.

2.3.3 Sampling rows of a matrix

In Fourier imaging, we subsample rows of $\boldsymbol{F}$ to form a measurement matrix. Let us introduce notation convenient for this. Let $\{\boldsymbol{e}_i\}_{i=1}^N$ denote the standard basis of $\mathbb{C}^N$, and $\Omega \subseteq [\![N]\!]$ a set of indices. We denote $\boldsymbol{P}_\Omega \in \mathbb{C}^{N\times N}$ as the orthogonal projection onto the linear span of $\{\boldsymbol{e}_i : i \in \Omega\}$. Namely, for $\boldsymbol{x} = (x_i)_{i=1}^N \in \mathbb{C}^N$, the jth entry of $\boldsymbol{P}_\Omega \boldsymbol{x}$ is equal to x_j if $j \in \Omega$,

and zero otherwise. Observe that $\boldsymbol{P}_\Omega$ is a diagonal matrix with ith diagonal entry equal to one if $i \in \Omega$, and zero otherwise. Abusing notation, we sometimes consider the fat matrix $\boldsymbol{P}_\Omega \in \mathbb{C}^{|\Omega| \times N}$ where the rows corresponding to a zero diagonal entry have been removed. Thus, in d dimensions with $\Omega \subseteq [\![N^d]\!]$, our measurement matrices have the form

$$\boldsymbol{A} = c\boldsymbol{P}_\Omega \boldsymbol{F} \in \mathbb{C}^{|\Omega| \times N^d},$$

where c is some normalizing constant.

2.3.4 Gradient operators and TV minimization

The *one-dimensional discrete gradient operator* of $\mathbb{C}^N$ is the circulant matrix defined by

$$\boldsymbol{V} = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ 1 & & & & -1 \end{pmatrix} \in \mathbb{C}^{N \times N}.$$

A signal processing interpretation of this matrix is that it acts as a convolution on a signal $\boldsymbol{x}$ with kernel $(-1, 1)$ and a periodic boundary condition. By virtue of being circulant, the Fourier matrix will diagonalize $\boldsymbol{V}$. Mathematically, this will be convenient in our recovery guarantees analysis.

To generalize $\boldsymbol{V}$ to vectorized d-dimensional tensors, we define the *jth partial derivative operator*, denoted by $\boldsymbol{V}_j \in \mathbb{C}^{N^d \times N^d}$ for $1 \leq j \leq d$, by

$$\boldsymbol{V}_j = \underbrace{\boldsymbol{I} \otimes \cdots \otimes \boldsymbol{I}}_{d-j \text{ times}} \otimes \boldsymbol{V} \otimes \underbrace{\boldsymbol{I} \otimes \cdots \otimes \boldsymbol{I}}_{j-1 \text{ times}},$$

where $\boldsymbol{I}$ is the $N \times N$ identity matrix and $\otimes$ is the Kronecker product. Given a d-dimensional tensor $\boldsymbol{X}$, $\boldsymbol{V}_j$ acts on $\mathrm{vec}(\boldsymbol{X})$ by broadcasting the one-dimensional discrete gradient operator along dimension j of $\boldsymbol{X}$.

The *(anisotropic) d-dimensional discrete gradient operator* of $\mathbb{C}^{N^d}$ is defined as the block matrix

$$\boldsymbol{V}^{(d)} = \begin{pmatrix} \boldsymbol{V}_1 \\ \vdots \\ \boldsymbol{V}_d \end{pmatrix} \in \mathbb{R}^{dN^d \times N^d}.$$

Sometimes we abuse notation and write $\boldsymbol{V} = \boldsymbol{V}^{(d)}$, where the dimension is inferred from context. Now, the *d-dimensional (anisotropic) Total Variation (TV) semi-norm* $\|\cdot\|_{\mathrm{TV}}$:

$\mathbb{C}^{N^d} \to \mathbb{R}_+$ is given by

$$\|\boldsymbol{x}\|_{\mathrm{TV}} = \|\boldsymbol{V}\boldsymbol{x}\|_{\ell^1}, \qquad \forall \boldsymbol{x} \in \mathbb{C}^{N^d}. \tag{2.3.1}$$

This allows us to define the *d-dimensional TV minimization* problem

$$\min_{\boldsymbol{z} \in \mathbb{C}^{N^d}} \|\boldsymbol{z}\|_{\mathrm{TV}} \text{ subject to } \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}\|_{\ell^2} \leq \eta. \tag{2.3.2}$$

Observe that TV minimization is a special case of (2.2.1) where $\boldsymbol{W} = \boldsymbol{V}^\top \in \mathbb{R}^{N^d \times dN^d}$ is the analysis matrix. In addition, $\boldsymbol{V}$ is not full rank since any nonzero constant vector is in the null space of $\boldsymbol{V}$, so the columns of $\boldsymbol{V}^\top$ do not form a frame for $\mathbb{C}^{dN^d}$.

Lastly, for some historical background, we remark that TV minimization is used extensively in image processing [25, 27] and compressive imaging [4, 53, 64, 81]. It should also be noted that there are other ways to define the TV semi-norm apart from (2.3.1). For example, many of our results extend to the *isotropic* TV semi-norm [4, Sec. 2.5], which we omit from our analysis for simplicity. Nonperiodic discrete gradient transforms [54, 71, 72] can also be considered, however our analysis relies on periodicity.

Chapter 3

Bernoulli model for random sampling schemes

We dedicate much of this chapter to proving some key lemmas, which we use to establish image and gradient recovery guarantees in Fourier imaging. The techniques and tools we use are adapted from the work and discussion found in [4, 5]. In our analysis, we consider a Bernoulli model for sampling rows of unitary matrices, which helps give us an efficient unrolling of NESTA. Moreover, we aim to show that for an appropriate sampling scheme, the Fourier matrix and *diagonal-scaled Fourier-Haar matrix* satisfy the RIP with high probability. The former will be sufficient to recover the gradient, and the latter will be sufficient to recover the image. Finally, we end the chapter by discussing the *stacked sampling scheme*. The scheme combines uniform and variable density samples, which is necessary to obtain both image and gradient recovery, but leads to a measurement matrix that cannot be immediately used with NESTA. This motivates Chapter 4, where we derive a novel modification of NESTA, named *stacked NESTA*, that is compatible with the stacking scheme.

3.1 Motivation and terminology

3.1.1 Bernoulli model

Deriving the update formulas for NESTA involves computing projections onto a constraint set. For QCBP (2.2.1), the measurement matrix $\boldsymbol{A}$ needs to have special properties for fast and exact calculation of the projection. One solution is to assume the rows of $\boldsymbol{A}$ are orthonormal up to a constant factor, i.e. $\boldsymbol{A}\boldsymbol{A}^* = c\boldsymbol{I}$ [13], so $\boldsymbol{A}^*\boldsymbol{A}$ is an orthogonal projection matrix. Such an assumption is reasonable when considering measurements obtained using subsampled unitary matrices, which are common in compressed sensing. Two prevalent examples are the Walsh-Hadamard transform and, in our case, the discrete Fourier transform [4]. In terms of mathematical analysis, it is desirable for the rows of the measurement matrix to be independent. This is straightforward to achieve when randomly subsampling rows independently with replacement. However, $\boldsymbol{A}$ could then easily violate the aforementioned

orthogonality condition. We can avoid this issue by using a *Bernoulli model sampling scheme* [5, Sec. 11.4.3] (see also [20, 80, 84, 91]). In such a scheme, each row of a given unitary matrix undergoes an independent Bernoulli trial to determine whether or not it is included in the measurement matrix. By construction, the scheme samples independently without replacement and thus enforces $\boldsymbol{A}\boldsymbol{A}^* = c\boldsymbol{I}$. Let us make this precise.

Definition 3.1.1 (Bernoulli sampling scheme)**.** Let $1 \leq m \leq N$. A *Bernoulli variable density sampling scheme of order* m is a random subset $\Omega \subseteq [\![N]\!]$ where each $j \in [\![N]\!]$ is sampled independently, so that $j \in \Omega$ with probability p_j (and $j \notin \Omega$ with probability $1 - p_j$), and

$$\sum_{i=1}^{N} p_i = m, \qquad 0 < p_j \leq 1, \qquad \forall j \in [\![N]\!].$$

We refer to $\boldsymbol{p} = (p_i)_{i=1}^N$ as a *Bernoulli vector of order* m and indicate such a random set Ω by $\Omega \sim \mathrm{Ber}([\![N]\!], m, \boldsymbol{p})$. In the special case that $p_i = m/N$ for all $i \in [\![N]\!]$, we refer to Ω as a *Bernoulli uniform sampling scheme of order* m and use the notation $\Omega \sim \mathrm{Ber}([\![N]\!], m)$. $\lozenge$

Observe that unlike in random sampling schemes where indices are independently sampled with replacement, the cardinality of $\Omega \sim \mathrm{Ber}([\![N]\!], m, \boldsymbol{p})$ is now a random variable equal to m in expectation. Later in Section 3.2.4 we show that $|\Omega|$ is close to m with high probability.

3.1.2 Jointly isotropic sampling operators

To prove recovery guarantees for Fourier imaging with Bernoulli sampling schemes, we need additional terminology from compressed sensing theory. The first is the notion of jointly isotropic collections, a formalism used to describe sampling operators based on families of random vectors. This specifies a very general way to construct measurement matrices that satisfy the RIP with high probability. Sampling from (isotropic) families of random vectors was first considered in [24], and the extension to collections is due to [1].

Definition 3.1.2 (Joint isotropy condition [5, Defn. 11.4])**.** Let $\mathcal{A}_1, \ldots, \mathcal{A}_m$ be independent families of random vectors on $\mathbb{C}^N$. The collection $\mathcal{C} = \{\mathcal{A}_i\}_{i=1}^m$ is *jointly isotropic* if

$$\frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{\mathcal{A}_i}(\boldsymbol{a}_i \boldsymbol{a}_i^*) = \boldsymbol{I}$$

where $\boldsymbol{a}_i \sim \mathcal{A}_i$ for $i = 1, \ldots, m$. $\lozenge$

The sampling operator corresponding to a collection $\mathcal{C} = \{\mathcal{A}_i\}_{i=1}^m$ samples each family $\mathcal{A}_i$ independently, where the associated measurement matrix is defined by

$$\boldsymbol{A} = \frac{1}{\sqrt{m}} \begin{pmatrix} \boldsymbol{a}_1^* \\ \vdots \\ \boldsymbol{a}_m^* \end{pmatrix} \in \mathbb{C}^{m \times N}, \qquad \boldsymbol{a}_i \sim \mathcal{A}_i, \qquad i = 1, \dots, m. \tag{3.1.1}$$

Thus the joint isotropy condition asserts that $\mathbb{E}(\boldsymbol{A}^*\boldsymbol{A}) = \boldsymbol{I}$, and so $\boldsymbol{A}$ preserves distances and angles in expectation, i.e.

$$\mathbb{E}\left(\|\boldsymbol{A}\boldsymbol{x}\|_{\ell^2}^2\right) = \|\boldsymbol{x}\|_{\ell^2}^2, \qquad \mathbb{E}(\boldsymbol{x}^*\boldsymbol{A}^*\boldsymbol{A}\boldsymbol{z}) = \boldsymbol{x}^*\boldsymbol{z}, \qquad \forall \boldsymbol{x}, \boldsymbol{z} \in \mathbb{C}^N.$$

Note the relation with the RIP, which specifies a matrix approximately preserving distances for sparse vectors with high probability. As we describe below, measurement matrices arising from jointly isotropic collections can be shown to satisfy the RIP with high probability.

Bernoulli sampling as a collection of independent families of random vectors is expressed as follows. We specify the collection by each family having exactly two vectors, a nonzero vector (e.g. the row of a given matrix) and the zero vector. Each family is then assigned a Bernoulli distribution over its vectors with probability parameters defined by the Bernoulli vector from Definition 3.1.1. Let us state this for sampling rows of a unitary matrix with the Bernoulli model, noting the rows must be rescaled to ensure the joint isotropy condition.

Proposition 3.1.3. *Let $\boldsymbol{U} \in \mathbb{C}^{N \times N}$ be a unitary matrix and $\boldsymbol{u}_i = \boldsymbol{U}^*\boldsymbol{e}_i$ denote the ith row of $\overline{\boldsymbol{U}}$ as a column vector. Let $\boldsymbol{p} = (p_i)_{i=1}^N$ be a Bernoulli vector of order m. For each $i \in [\![N]\!]$, let $\mathcal{A}_i$ be the family of two vectors $\sqrt{N/p_i}\boldsymbol{u}_i$ and the zero vector, and for $\boldsymbol{a}_i \sim \mathcal{A}_i$, define*

$$\mathbb{P}\left(\boldsymbol{a}_i = \sqrt{\frac{N}{p_i}}\boldsymbol{u}_i\right) = p_i, \qquad \mathbb{P}(\boldsymbol{a}_i = \boldsymbol{0}) = 1 - p_i.$$

Then the collection $\mathcal{C} = \{\mathcal{A}_i\}_{i=1}^N$ is jointly isotropic.

Proof. Since $\boldsymbol{p}$ has no zero entries, we have

$$\frac{1}{N}\sum_{i=1}^N \mathbb{E}_{\mathcal{A}_i}(\boldsymbol{a}_i\boldsymbol{a}_i^*) = \frac{1}{N}\sum_{i=1}^N p_i \frac{N}{p_i}\boldsymbol{u}_i\boldsymbol{u}_i^* = \sum_{i=1}^N \boldsymbol{u}_i\boldsymbol{u}_i^* = \boldsymbol{U}^*\boldsymbol{U} = \boldsymbol{I}.$$

This verifies that $\mathcal{C}$ is jointly isotropic using Definition 3.1.2. □

In terms of a sampling operator, each row of $\overline{\boldsymbol{U}}$ is sampled at most once, and row i is included with probability p_i. Using (3.1.1), the Bernoulli model sampling operator leads to a measurement matrix of size $N \times N$, with m nonzero rows in expectation. For convenience and without loss of generality, we always ignore the zero rows of the measurement matrix. This gives a matrix of size $|\Omega| \times N$ for $\Omega \sim \mathrm{Ber}([\![N]\!], m, \boldsymbol{p})$ with $\mathbb{E}(|\Omega|) = m$.

3.1.3 RIP matrices from joint isotropy

Note that we allow the possibility of $p_i = 1$, so we can have deterministic samples in the Bernoulli model. If $p_i = 1$, then $\mathcal{A}_i$ is said to be a *singleton family* with $|\mathcal{A}_i| = 1$. More generally, a jointly isotropic collection $\mathcal{C} = \{\mathcal{A}_i\}_{i=1}^m$ has *saturation* $\tilde{m}$ *with* $0 \leq \tilde{m} \leq m$, if exactly $\tilde{m}$ families of $\mathcal{C}$ are singleton families [5, Defn. 11.12]. If $\tilde{m} = 0$, we say $\mathcal{C}$ is *unsaturated*. Saturation affects the measure of *coherence* of a jointly isotropic collection. The concept of coherence plays a role in qualitatively determining the number of measurements needed for the RIP for recovery.

Definition 3.1.4 (Coherence [5, Defn. 11.16, 11.17])**.** Let $\mathcal{A}$ be a family of at least two random vectors. The *coherence of* $\mathcal{A}$, denoted by $\mu(\mathcal{A})$, is the smallest constant such that $\|\boldsymbol{a}\|_{\ell^\infty}^2 \leq \mu(\mathcal{A})$ almost surely for $\boldsymbol{a} \sim \mathcal{A}$. The *coherence of a collection* $\mathcal{C} = \{\mathcal{A}_i\}_{i=1}^m$ is defined as

$$\mu(\mathcal{C}) = \max\left\{\mu(\mathcal{A}_i) \ : \ i = 1, \dots, m, \ |\mathcal{A}_i| \geq 2\right\}.$$

$\Diamond$

By excluding singleton families from the definition of coherence, deterministic samples are omitted from the measure of coherence. More broadly, they do not play a role in the analysis of recovery guarantees.

Remark 3.1.5. For the results throughout this chapter, many of them hold trivially in the fully saturated case, i.e. $m = N^d$. For ease of exposition, our proofs throughout the chapter always assume that $m < N^d$, but note that the results also hold when $m = N^d$. $\Diamond$

Finally, we end the section by connecting the previous concepts by stating a key result we use for recovery guarantees. Namely, it says that given a jointly isotropic collection, under certain conditions, its associated measurement matrix (3.1.1) satisfies the RIP with some (high) probability. Specifically, the condition needed relates the number of families in the collection with its coherence. Its proof can be found in the corresponding citation.

Lemma 3.1.6 (Joint isotropy implies RIP, [5, Cor. 13.15] with $\boldsymbol{G} = \boldsymbol{I}$)**.** *Let* $0 < \epsilon < 1$, $2 \leq s \leq N$, $\mathcal{C} = \{\mathcal{A}_i\}_{i=1}^m$ *be a jointly isotropic collection, and* $\boldsymbol{A}$ *be as in* (3.1.1)*. Suppose that*

$$m \gtrsim \delta^{-2} \cdot \mu(\mathcal{C}) \cdot s \cdot \left(\log(2(\mu(\mathcal{C})s + 1)) \cdot \log^2(s) \cdot \log(N) + \log(\epsilon^{-1})\right), \tag{3.1.2}$$

where $\mu(\mathcal{C})$ *is defined in Definition 3.1.4. Then with probability at least* $1 - \epsilon$, $\boldsymbol{A}$ *has the RIP of order* s *with constant* $\delta_s \leq \delta$.

Note this result is not beneficial for compressive imaging when $\mu(\mathcal{C}) = \mathcal{O}(N)$, since that would mean the number of measurements m needs to be proportional to the image size N. To reap any benefits in the undersampled setting, it is important to construct collections

that are *incoherent*, i.e. where $\mu(\mathcal{C})$ is either independent of N or logarithmically growing in N.

3.2 Compressed sensing theory

In this section, we develop the machinery needed to ensure gradient and image recovery from Fourier measurements via TV minimization (2.3.2). To reconstruct the gradient of the image, it is sufficient for a subsampled Fourier matrix to satisfy the RIP. This is achieved by using a uniform sampling pattern. To reconstruct the image itself, it is sufficient for a subsampled diagonal-scaled Fourier-Haar matrix to satisfy the RIP. This is achieved effectively with a certain nonuniform sampling pattern. The Haar matrix, i.e. the discrete Haar wavelet transform, is introduced to exploit a connection between Haar wavelet coefficients and the TV semi-norm $\|\cdot\|_{\mathrm{TV}}$. This idea first appeared in [71,72] to show accurate and stable recovery via TV minimization. The consideration of using both uniform and nonuniform sampling patterns for Fourier imaging is due to [80].

3.2.1 Recovery guarantees with Fourier measurements

Here we show that a Bernoulli uniform sampling scheme is sufficient to ensure the subsampled Fourier matrix has the RIP. This is leveraged in Section 4.2 to recover the image gradient via TV minimization (2.3.2). Let $\mathcal{C} = \{\mathcal{A}_i\}_{i=1}^{N^d}$ be the jointly isotropic collection in Proposition 3.1.3 corresponding to $\mathrm{Ber}(\llbracket N^d \rrbracket, m)$ with unitary matrix $\boldsymbol{U} = N^{-d/2}\boldsymbol{F}$. In particular, for $\boldsymbol{a}_i \sim \mathcal{A}_i$ we have

$$\mathbb{P}\left(\boldsymbol{a}_i = \frac{N^d}{\sqrt{m}}\boldsymbol{u}_i\right) = mN^{-d}, \qquad \mathbb{P}(\boldsymbol{a}_i = \boldsymbol{0}) = 1 - mN^{-d}.$$

For $m < N^d$ (where $\mathcal{C}$ is not fully saturated) we have

$$\mu(\mathcal{C}) = \max_{i=1,\dots,N^d} \mu(\mathcal{A}_i) = \frac{N^d}{m} \cdot \max_{i=1,\dots,N^d} N^d \|\boldsymbol{u}_i\|_{\ell^\infty}^2 = \frac{N^d}{m} \cdot \max_{i=1,\dots,N^d} \|\boldsymbol{f}_i\|_{\ell^\infty}^2 = \frac{N^d}{m}.$$

Here $\boldsymbol{f}_i$ denotes the ith row of the Fourier matrix $\boldsymbol{F}$. The entries of $\boldsymbol{F}$ lie on the unit circle in $\mathbb{C}$, and from its definition, it follows that $\max_{i=1,\dots,N^d} \|\boldsymbol{f}_i\|_{\ell^\infty} = 1$. Now define the random matrix $\boldsymbol{A}$ corresponding to $\mathcal{C}$ by

$$\boldsymbol{A} = \frac{1}{\sqrt{N^d}} \begin{pmatrix} \boldsymbol{a}_i^* \\ \vdots \\ \boldsymbol{a}_{N^d}^* \end{pmatrix} \in \mathbb{C}^{N^d \times N^d},$$

where $\boldsymbol{a}_i \sim \mathcal{A}_i$ independently for all i. Equivalently, we can write

$$\boldsymbol{A} = \frac{1}{\sqrt{m}} \boldsymbol{P}_\Omega \boldsymbol{F}, \qquad \Omega \sim \mathrm{Ber}([\![N^d]\!], m),$$

where sampling an independent realization of $\boldsymbol{A}$ yields our measurement matrix.

Lemma 3.2.1 (RIP for Bernoulli uniform subsampled Fourier matrix)**.** *Let $\delta > 0$, $0 < \epsilon < 1$, $d \geq 1$, $2 \leq s \leq N^d$, $1 \leq m < N^d$ and $\Omega \sim \mathrm{Ber}([\![N^d]\!], m)$. Suppose $\boldsymbol{A} = m^{-1/2} \boldsymbol{P}_\Omega \boldsymbol{F}$ where $\boldsymbol{F}$ is the d-dimensional Fourier matrix. If*

$$m \gtrsim_d \delta^{-2} \cdot s \cdot \left(\log(Ns) \cdot \log^2(s) \cdot \log(N) + \log(\epsilon^{-1}) \right), \tag{3.2.1}$$

then with probability at least $1 - \epsilon$, $\boldsymbol{A}$ has the RIP of order s with constant $\delta_s \leq \delta$.

Proof. First, the collection $\mathcal{C}$ defining $\boldsymbol{A}$ above is jointly isotropic by Proposition 3.1.3. Then by Lemma 3.1.6 where $\mu(\mathcal{C}) = N^d m^{-1}$, if

$$N^d \gtrsim \delta^{-2} \cdot \frac{N^d}{m} \cdot s \cdot \left(\log\left(2\left(N^d m^{-1} s + 1\right)\right) \cdot \log^2(s) \cdot \log(N^d) + \log(\epsilon^{-1}) \right) \tag{3.2.2}$$

then with probability at least $1 - \epsilon$, $\boldsymbol{A}$ has the RIP of order s with constant $\delta_s \leq \delta$. Multiplying both sides of the inequality by mN^{-d} gives

$$m \gtrsim \delta^{-2} \cdot s \cdot \left(\log\left(2(N^d m^{-1} s + 1)\right) \log^2(s) \cdot \log(N^d) + \log(\epsilon^{-1}) \right).$$

Now, using the bound

$$\log\left(2\left(N^d m^{-1} s + 1\right)\right) \leq \log(4N^d s) \leq 2\log(N^d s),$$

which holds since $2 \leq s \leq N^d$ and $m \geq 1$, and factoring out d from any log terms containing N^d yields the condition (3.2.1). This condition implies (3.2.2), which gives the result. □

3.2.2 Recovery guarantees with Fourier-Haar measurements

Here we adopt the notation and definitions found in [4, Section 3.2]. Consider a Bernoulli variable density sampling scheme with probabilities $\boldsymbol{p} = (p_i)_{i=1}^{N^d}$. Let

$$\varsigma : \{1, \ldots, N^d\} \to \{1, \ldots, N\}^d$$

be the lexicographical ordering described in Section 2.3.1 and

$$\varrho^{(d)} : \{1, \ldots, N^d\} \to \{-N/2 + 1, \ldots, N/2\}^d$$

be the row-to-frequency bijection described in Section 2.3.2. Thus for each i, the probability p_i corresponds to the probability of including frequency $\omega = \varrho^{(d)}(i)$. For convenience, we abuse notation and write $p_\omega := p_{(\varrho^{(d)})^{-1}(\omega)}$.

Now for $\omega \in \mathbb{R}$, denote $\bar{\omega} = \max\{|\omega|, 1\}$. Moreover, if $\omega = (\omega_1, \dots, \omega_d) \in \mathbb{R}^d$, let $\pi : \{1, \dots, d\} \to \{1, \dots, d\}$ be a bijection such that

$$\overline{\omega_{\pi(1)}} \geq \overline{\omega_{\pi(2)}} \geq \cdots \geq \overline{\omega_{\pi(d)}}.$$

In addition, set

$$q_\omega = \begin{cases} \overline{\omega_{\pi(1)}} \cdots \overline{\omega_{\pi(d/2)}} & d \text{ even} \\ \overline{\omega_{\pi(1)}} \cdots \overline{\omega_{\pi((d-1)/2)}} \sqrt{\overline{\omega_{\pi((d+1)/2)}}} & d \text{ odd} \end{cases}.$$

Lastly, let $\Gamma(\boldsymbol{p})$ be the smallest positive constant such that

$$q_\omega^{-2} \leq \frac{\Gamma(\boldsymbol{p}) p_\omega}{m}, \qquad \forall \omega \in \{-N/2+1, \dots, N/2\}^d,\ p_\omega < 1.$$

For the trivial case consisting of $p_\omega = 1$ for all ω, we define $\Gamma(\boldsymbol{p}) = 0$. Observe that $\Gamma(\boldsymbol{p})$ is defined only in terms of frequencies that are not deterministically sampled. Ultimately, the purpose of $\Gamma(\boldsymbol{p})$ will be to bound the coherence of the jointly isotropic collection associated with Bernoulli variable density sampling.

Remark 3.2.2. Similar to the observation in [4, Sec. 3.2], for $m < N^d$ we can show that $\Gamma(\boldsymbol{p}) > 1$ for any Bernoulli vector $\boldsymbol{p}$ of order m with N^d entries. We point out that for our argument to hold, it is enough for N to be even instead of a power of two.

Observe that one always has $\Gamma(\boldsymbol{p}) > 2^d m N^{-d}$ whenever $p_\omega < 1$ for at least one frequency ω. Consequently, if $m \geq (N/2)^d$, then $\Gamma(\boldsymbol{p}) > 1$. Otherwise if $m < (N/2)^d$, then note that there are at most m deterministic samples, so we always have $|\{\omega : p_\omega < 1\}| \geq N^d - m$. Moreover, $q_\omega \leq (N/2)^{d/2}$ for all ω. Therefore

$$\Gamma(\boldsymbol{p}) \geq \sum_{\omega : p_\omega < 1} \frac{\Gamma(\boldsymbol{p}) p_\omega}{m} \geq \sum_{\omega : p_\omega < 1} q_\omega^{-2} \geq \frac{N^d - m}{(N/2)^d} = 2^d - \frac{m}{(N/2)^d}.$$

Since $m < (N/2)^d$, we conclude $\Gamma(\boldsymbol{p}) > 2^d - 1 \geq 1$. ◊

Let $\boldsymbol{U}$ be the unitary matrix given by $\boldsymbol{U} = N^{-d/2} \boldsymbol{F} \boldsymbol{W}$, where $\boldsymbol{W} = \boldsymbol{W}^{(d)}$ is the d-dimensional discrete (orthonormal) Haar wavelet transform as in [4, Sec. SM2.1]. Now let $\mathcal{C} = \{\mathcal{A}_i\}_{i=1}^{N^d}$ be the jointly isotropic collection in Proposition 3.1.3 corresponding to $\mathrm{Ber}(\llbracket N^d \rrbracket, m, \boldsymbol{p})$. We have $\boldsymbol{a}_i \sim \mathcal{A}_i$ where

$$\mathbb{P}\left(\boldsymbol{a}_i = \sqrt{\frac{N^d}{p_i}} \boldsymbol{u}_i\right) = p_i, \qquad \mathbb{P}(\boldsymbol{a}_i = \boldsymbol{0}) = 1 - p_i,$$

with $\boldsymbol{u}_i$ corresponding to the ith row of $\overline{\boldsymbol{U}}$. Considering the nontrivial case $m < N^d$ (where $\mathcal{C}$ is not fully saturated), the coherence of $\mathcal{C}$ is given by

$$\mu(\mathcal{C}) = \max_{i:|\mathcal{A}_i|\geq 2} \mu(\mathcal{A}_i) = N^d \cdot \max_{i:p_i<1} \frac{\|\boldsymbol{u}_i\|_{\ell^\infty}^2}{p_i}.$$

We now claim

$$\mu(\mathcal{C}) \lesssim_d \frac{N^d}{m} \cdot \Gamma(\boldsymbol{p}).$$

To show this, we adapt the proof of [4, Lem. 7.5]. First write $\tilde{\boldsymbol{p}} = \boldsymbol{p}/m$, which has entries $\tilde{p}_i = p_i/m$, $i \in [\![N^d]\!]$. This normalizes $\boldsymbol{p}$ to a new vector $\tilde{\boldsymbol{p}}$ belonging to the standard simplex, which is a property of the vector $\boldsymbol{p}$ in [4, Lem. 7.5]. Thus

$$\mu(\mathcal{C}) = \frac{N^d}{m} \cdot \max_{i:\tilde{p}_i<1/m} \frac{\|\boldsymbol{u}_i\|_{\ell^\infty}^2}{\tilde{p}_i}.$$

Then when deriving [4, Eq. 7.6], we can restrict which frequencies to maximize over in the definition of Θ [4, Sec. SM1.2] and its upper bound, and still have the resulting inequality hold. This gives

$$\max_{i:\tilde{p}_i<1/m} \frac{\|\boldsymbol{u}_i\|_{\ell^\infty}}{\sqrt{\tilde{p}_i}} \lesssim_d \max_{\substack{\omega=(\omega_1,\ldots,\omega_d)\\ \tilde{p}_\omega<1/m}} \max_{j=0,\ldots,R-1} \left\{ \frac{1}{\sqrt{\tilde{p}_\omega}} \prod_{i=1}^d \frac{2^{j/2}}{\max\{\overline{\omega_i}, 2^j\}} \right\},$$

where R is the positive integer exponent for which $N = 2^R$. Following the rest of the proof in the same fashion, until the last equation line, gives

$$\max_{i:\tilde{p}_i<1/m} \frac{\|\boldsymbol{u}_i\|_{\ell^\infty}}{\sqrt{\tilde{p}_i}} \lesssim_d \max_{\substack{\omega=(\omega_1,\ldots,\omega_d)\\ \tilde{p}_\omega<1/m}} \left\{ \frac{1}{q_\omega \sqrt{\tilde{p}_\omega}} \right\} = \max_{\substack{\omega=(\omega_1,\ldots,\omega_d)\\ p_\omega<1}} \left\{ \frac{1}{q_\omega \sqrt{p_\omega/m}} \right\} \leq \sqrt{\Gamma(\boldsymbol{p})}$$

where the last inequality follows from the definition of $\Gamma(\boldsymbol{p})$. We summarize this in a lemma.

Lemma 3.2.3. *Let $\mathcal{C} = \{\mathcal{A}_i\}_{i=1}^{N^d}$ be the jointly isotropic collection in Proposition 3.1.3 arising from $\mathrm{Ber}([\![N^d]\!], m, \boldsymbol{p})$ with $1 \leq m < N^d$ and unitary matrix*

$$\boldsymbol{U} = N^{-d/2} \boldsymbol{F} \boldsymbol{W},$$

where $\boldsymbol{F}$ and $\boldsymbol{W}$ are the d-dimensional Fourier matrix and discrete Haar wavelet transform, respectively. Then

$$\mu(\mathcal{C}) \lesssim_d \frac{N^d}{m} \cdot \Gamma(\boldsymbol{p}).$$

Now we state and prove a key lemma needed for image recovery guarantees from Fourier measurements.

Lemma 3.2.4 (RIP for Bernoulli variable subsampled Fourier-Haar matrix)**.** *Let $\delta > 0$, $0 < \epsilon < 1$, $d \geq 1$, $2 \leq s \leq N^d$, $1 \leq m < N^d$, $\boldsymbol{p} = (p_i)_{i=1}^{N^d}$ and $\Omega \sim \mathrm{Ber}(\llbracket N^d \rrbracket, m, \boldsymbol{p})$. Suppose $\boldsymbol{A} = N^{-d/2}\boldsymbol{P}_\Omega \boldsymbol{D}\boldsymbol{F}\boldsymbol{W}$ where $\boldsymbol{F}$ and $\boldsymbol{W}$ are the d-dimensional Fourier matrix and discrete Haar wavelet transform, respectively, and $\boldsymbol{D}$ is a diagonal matrix with diagonal entries $(\boldsymbol{D})_{ii} = \frac{1}{\sqrt{p_i}}$ for $i = 1, \dots, N^d$. If*

$$m \gtrsim_d \delta^{-2} \cdot \Gamma(\boldsymbol{p}) \cdot s \cdot \left(\log(\Gamma(\boldsymbol{p}) N s) \cdot \log^2(s) \cdot \log(N) + \log(\epsilon^{-1}) \right), \tag{3.2.3}$$

then with probability at least $1 - \epsilon$, $\boldsymbol{A}$ has the RIP of order s with constant $\delta_s \leq \delta$.

Proof. Given the jointly isotropic collection $\mathcal{C} = \{\mathcal{A}_i\}_{i=1}^{N^d}$ from Lemma 3.2.3 and $\boldsymbol{a}_i \sim \mathcal{A}_i$ for each i, the associated measurement matrix

$$\boldsymbol{A} = \frac{1}{\sqrt{N^d}} \begin{pmatrix} \boldsymbol{a}_1^* \\ \vdots \\ \boldsymbol{a}_{N^d}^* \end{pmatrix} \in \mathbb{C}^{N^d \times N^d}$$

is precisely $\boldsymbol{A} = N^{-d/2}\boldsymbol{P}_\Omega \boldsymbol{D}\boldsymbol{F}\boldsymbol{W}$, where $\Omega \sim \mathrm{Ber}(\llbracket N^d \rrbracket, m, \boldsymbol{p})$ and $\boldsymbol{D}$ is a diagonal matrix with diagonal entries $(\boldsymbol{D})_{ii} = p_i^{-1/2}$ for $i = 1, \dots, N^d$. Then for $0 < \epsilon < 1$, $2 \leq s \leq N^d$, by Lemma 3.1.6, if

$$N^d \gtrsim \delta^{-2} \cdot \mu(\mathcal{C}) \cdot s \cdot \left(\log(2(\mu(\mathcal{C})s + 1)) \cdot \log^2(s) \cdot \log(N^d) + \log(\epsilon^{-1}) \right) \tag{3.2.4}$$

then with probability at least $1 - \epsilon$, $\boldsymbol{A}$ has the RIP of order s with constant $\delta_s \leq \delta$. Now, by Lemma 3.2.3 we have

$$\mu(\mathcal{C}) \lesssim_d \frac{N^d}{m} \cdot \Gamma(\boldsymbol{p}),$$

and in combination with $\Gamma(\boldsymbol{p}) > 1$ (Remark 3.2.2) and $N, s \geq 2$ we have

$$\log(2(\mu(\mathcal{C})s + 1)) \lesssim_d \log(\Gamma(\boldsymbol{p}) N s).$$

This is shown as follows. Using Lemma 3.2.3, we have

$$\mu(\mathcal{C}) \leq \phi(d) \cdot \frac{N^d}{m} \cdot \Gamma(\boldsymbol{p})$$

for some (positive) function ϕ. Also, $N, s \geq 2$ and $\Gamma(\boldsymbol{p}) > 1$, so $\log(\Gamma(\boldsymbol{p}) N s) \geq 1$. Therefore

$$\begin{aligned} \log(2(\mu(\mathcal{C})s + 1)) &\leq \log(2(\phi(d) N^d m^{-1} \Gamma(\boldsymbol{p}) s + 1)) \\ &\leq \log\left(2(\phi(d) + 1) N^d \Gamma(\boldsymbol{p}) s\right) \\ &\leq \log(2(\phi(d) + 1)) + d \log(\Gamma(\boldsymbol{p}) N s) \\ &\leq (\log(2(\phi(d) + 1)) + d) \log(\Gamma(\boldsymbol{p}) N s) \end{aligned}$$

giving $\log(2(\mu(\mathcal{C})+1)) \lesssim_d \log(\Gamma(\boldsymbol{p})Ns)$. Manipulating (3.2.4), by using the previous bounds and factoring out d from $\log(N^d)$, yields the condition

$$N^d \gtrsim_d \delta^{-2} \cdot \frac{N^d}{m} \cdot \Gamma(\boldsymbol{p}) \cdot s \cdot \left(\log(\Gamma(\boldsymbol{p})Ns) \cdot \log^2(s) \cdot \log(N) + \log(\epsilon^{-1})\right).$$

Multiplying both sides by mN^{-d} gives the condition (3.2.3). This condition implies (3.2.4), giving the result. □

Now we state and prove a crucial result we use to show image recovery, a Poincaré-like inequality, that arises from a connection between the TV semi-norm and Haar wavelet coefficients. This connection was first used in [71, 72] to show recovery of gradient-sparse signals. It has since become a standard tool to show gradient-sparse recovery guarantees, see e.g. [4, 54, 80] and [5, Chap. 17].

Lemma 3.2.5 (Poincaré inequality)**.** *Consider the setup of Lemma 3.2.4 with $\delta = 1/3$ and $\boldsymbol{A} = m^{-1/2}\boldsymbol{P}_\Omega \boldsymbol{F}$. For $d = 1$, given the measurement condition*

$$m \gtrsim \Gamma(\boldsymbol{p}) \cdot s \cdot \left(\log(\Gamma(\boldsymbol{p})Ns) \cdot \log^2(s) \cdot \log(N) + \log(2\epsilon^{-1})\right) \tag{3.2.5}$$

then with probability at least $1 - \epsilon/2$ the following holds

$$\|\boldsymbol{x}\|_{\ell^2} \lesssim \sqrt{\Gamma(\boldsymbol{p})}\|\boldsymbol{A}\boldsymbol{x}\|_{\ell^2} + \frac{\sqrt{N}\|\boldsymbol{x}\|_{\mathrm{TV}}}{s}, \qquad \forall \boldsymbol{x} \in \mathbb{C}^N.$$

For $d \geq 2$, given the measurement condition

$$m \gtrsim_d \Gamma(\boldsymbol{p}) \cdot s \cdot \log^2(N) \cdot \left(\log(\Gamma(\boldsymbol{p})N\log^2(N)s) \cdot \log^2(s\log^2(N)) \cdot \log(N) + \log(2\epsilon^{-1})\right) \tag{3.2.6}$$

then with probability at least $1 - \epsilon/2$ the following holds

$$\|\boldsymbol{x}\|_{\ell^2} \lesssim_d \sqrt{\Gamma(\boldsymbol{p})}\|\boldsymbol{A}\boldsymbol{x}\|_{\ell^2} + \frac{\|\boldsymbol{x}\|_{\mathrm{TV}}}{\sqrt{s}}, \qquad \forall \boldsymbol{x} \in \mathbb{C}^{N^d}.$$

Proof. Write $\boldsymbol{B} = N^{-d/2}\boldsymbol{P}_\Omega \boldsymbol{D}\boldsymbol{F}$. By Lemma 3.2.4 with $\delta = 1/3$ and respective measurement conditions (3.2.5) and (3.2.6), with probability at least $1 - \epsilon/2$, the matrix $\boldsymbol{B}\boldsymbol{W}$ has either, for $d = 1$, the RIP of order s with constant $\delta_s \leq 1/3$, or for $d \geq 2$, the RIP of order

$s' = \lceil s \log^2(N) \rceil$ with constant $\delta_{s'} \leq 1/3$. By [4, Lem. 7.4][1], in the case of $d = 1$ we get

$$\|\boldsymbol{x}\|_{\ell^2} \lesssim \|\boldsymbol{B}\boldsymbol{x}\|_{\ell^2} + \frac{\sqrt{N}\|\boldsymbol{x}\|_{\mathrm{TV}}}{s} \qquad \forall \boldsymbol{x} \in \mathbb{C}^N,$$

and for the case $d \geq 2$ we get

$$\|\boldsymbol{x}\|_{\ell^2} \lesssim_d \|\boldsymbol{B}\boldsymbol{x}\|_{\ell^2} + \frac{\|\boldsymbol{x}\|_{\mathrm{TV}}}{\sqrt{s}}, \qquad \forall \boldsymbol{x} \in \mathbb{C}^{N^d}.$$

It now remains to show that $\|\boldsymbol{B}\boldsymbol{x}\|_{\ell^2} \leq \sqrt{\Gamma(\boldsymbol{p})}\|\boldsymbol{A}\boldsymbol{x}\|_{\ell^2}$. Observe that we can express

$$\boldsymbol{B} = N^{-d/2}\boldsymbol{P}_\Omega \boldsymbol{D}\boldsymbol{F} = (mN^{-d})^{1/2}\boldsymbol{P}_\Omega \boldsymbol{D}\boldsymbol{P}_\Omega^\top \boldsymbol{A}$$

and we have $\|\boldsymbol{P}_\Omega\|_{\ell^2} = 1$, so

$$\|\boldsymbol{B}x\|_{\ell^2} \leq \sqrt{\frac{m}{N^d}}\|\boldsymbol{D}\|_{\ell^2}\|\boldsymbol{A}\boldsymbol{x}\|_{\ell^2} \leq \sqrt{\frac{m}{N^d}} \cdot \frac{1}{\min_\omega\{\sqrt{p_\omega}\}}\|\boldsymbol{A}\boldsymbol{x}\|_{\ell^2} < \sqrt{\Gamma(\boldsymbol{p})}\|\boldsymbol{A}\boldsymbol{x}\|_{\ell^2}.$$

The final inequality follows from there being at least one ω for which $p_\omega < 1$ and

$$\sqrt{\frac{m}{N^d}} \cdot \frac{1}{\sqrt{p_\omega}} \leq \sqrt{\Gamma(\boldsymbol{p})} \cdot \frac{q_\omega}{\sqrt{N^d}} \leq \sqrt{\Gamma(\boldsymbol{p})},$$

for all ω satisfying $p_\omega < 1$. □

Note that the choice of $\delta = 1/3$ is arbitrary, where any $\delta < \sqrt{2} - 1$ for Lemma 2.1.5 is sufficient for later developments.

3.2.3 Near-optimal variable sampling strategy

Here we derive a theoretically near-optimal sampling strategy for Bernoulli variable density sampling. By near-optimal, we mean that it nearly minimizes the parameter $\Gamma(\boldsymbol{p})$ arising in the measurement conditions (3.2.3), (3.2.5) and (3.2.6). In particular, our choice of $\boldsymbol{p}$ will lead to measurement conditions of the form $m \gtrsim_d s \cdot \mathrm{polylog}(N, s, \epsilon^{-1})$.

First, we show for an arbitrary Bernoulli vector $\boldsymbol{p}$ that $\Gamma(\boldsymbol{p}) \gtrsim \log(Nm^{-1/d})$. Second, we show there is a specific choice $\hat{\boldsymbol{p}}$ for which $\Gamma(\hat{\boldsymbol{p}}) \lesssim_d \log(N)$. Note the near-optimality of $\hat{\boldsymbol{p}}$, where the upper bound on $\Gamma(\hat{\boldsymbol{p}})$ is almost the same order as the global lower bound. This is inspired from the discussion and results in [4, Sec. 4].

[1] The referenced lemma assumes $\boldsymbol{B}\boldsymbol{W}$ has the RIP of order $5k$ with constant $\delta_{5k} < \delta = 1/3$. To clarify, the relevance of $\delta = 1/3$ in the proof is only when they reference [4, Lem. SM1.7], which only requires $\delta \leq 1/2$. This is compatible with our use of $\delta = 1/3$. Note that these values of δ are arbitrarily chosen to ensure the RIP implies the rNSP (e.g. see Lemma 2.1.5).

Proposition 3.2.6. *Let $1 \leq m < N^d$. Given a Bernoulli variable density sampling scheme* $\mathrm{Ber}(\llbracket N^d \rrbracket, m, \boldsymbol{p})$*, we have* $\Gamma(\boldsymbol{p}) \gtrsim \log(Nm^{-1/d})$.

Proof. Observe that in general $q_\omega \leq \overline{\omega_{\pi(1)}}^{d/2}$. Then we have

$$\Gamma(\boldsymbol{p}) \geq \sum_{\omega : p_\omega < 1} \frac{\Gamma(\boldsymbol{p}) p_\omega}{m} \geq \sum_{\omega : p_\omega < 1} q_\omega^{-2} \geq \sum_{\omega : p_\omega < 1} (\overline{\omega_{\pi(1)}})^{-d}.$$

Now we want a lower bound of the right-hand side independent of the sampling strategy. We do this by only considering frequencies ω outside the box $\|\omega\|_{\ell^\infty} \leq \lceil m^{1/d}/2 \rceil$. The purpose of this is to ignore at least m of the frequencies ω which maximize q_ω^{-2}, which serves as a general lower bound regardless of $\boldsymbol{p}$. For this argument to work, we further assume that $m \leq (N-2)^d$.

The number of frequencies inside the box is $(2\lceil m^{1/d}/2 \rceil + 1)^d > m$, so it contains at least m frequencies. The assumption $m \leq (N-2)^d$ ensures the set of frequencies outside the box is nonempty. Now, for $k = N/2, N/2-1, \ldots, \lceil m^{1/d}/2 \rceil + 1$, there are k^{d-1} frequencies ω where $\omega_1 = k$ and $0 < \omega_j \leq k$ for $2 \leq j \leq d$. Note again that the assumption $m \leq (N-2)^d$ ensures there are a nonzero number of k values. This yields

$$\sum_{\omega : p_\omega < 1} (\overline{\omega_{\pi(1)}})^{-d} \geq \sum_{k = \lceil m^{1/d}/2 \rceil + 1}^{N/2} k^{d-1} k^{-d} = \sum_{k = \lceil m^{1/d}/2 \rceil + 1}^{N/2} \frac{1}{k} \gtrsim \log\left(\frac{N}{m^{1/d}}\right),$$

giving the result. In the other case when $(N-2)^d < m < N^d$, the result holds from Remark 3.2.2 and by observing that $\log(Nm^{-1/d}) \searrow 0$ when $m \nearrow N^d$. □

Now we choose a Bernoulli vector $\boldsymbol{p}$ that minimizes $\Gamma(\boldsymbol{p})$. To do so, we take inspiration from the optimal sampling distribution in [4, Lem. 4.1]. The key idea is to ensure the bound in Proposition 3.2.6 is as tight as possible. Intuitively from the proof, we can achieve this by sampling frequencies satisfying $\|\omega\|_{\ell^\infty} \lesssim m^{1/d}$ with high probability, and sampling frequencies outside this region with low probability. This can be done more precisely as follows. We define $\hat{\boldsymbol{p}}$ by

$$\hat{p}_\omega = \min\left\{ m C q_\omega^{-2}, 1 \right\}, \qquad \omega \in \{-N/2+1, \ldots, N/2\}^d,$$

where $C > 0$ is a unique constant determined by the constraint $\sum_\omega \hat{p}_\omega = m$. To verify the existence and uniqueness of C, define the function

$$\phi(t) = \left(\sum_\omega \min\left\{ m t q_\omega^{-2}, 1 \right\} \right) - m, \qquad t \in [0, \infty).$$

Then $\sum_\omega \hat{p}_\omega = m$ if and only if C is a root of ϕ. Note that ϕ is continuous at every $t \in [0, \infty)$. Writing

$$T := \frac{1}{m} \cdot \Big(\max_\omega q_\omega\Big)^2 = \frac{1}{m}\left(\frac{N}{2}\right)^d,$$

then by continuity of ϕ, and observing that $\phi(0) = -m < 0$ and $\phi(T) = N^d - m > 0$, the intermediate value theorem gives that there exists $C \in (0, T)$ such that $\phi(C) = 0$. Uniqueness of C follows from the fact that ϕ is strictly increasing on $(0, T)$. We conclude that $\hat{\boldsymbol{p}}$ is well-defined.

Next, note that

$$\Gamma(\hat{\boldsymbol{p}}) = \max_{\omega:\hat{p}_\omega<1}\left\{\frac{mq_\omega^{-2}}{\hat{p}_\omega}\right\} = \max_{\omega:\hat{p}_\omega<1}\left\{\frac{1}{C}\right\} = \frac{1}{C}.$$

Now to bound $\Gamma(\hat{\boldsymbol{p}})$ above, we have

$$m = \sum_\omega \hat{p}_\omega = \sum_\omega \min\{mCq_\omega^{-2}, 1\} \le \sum_\omega mCq_\omega^{-2} = mC\sum_\omega q_\omega^{-2} \lesssim_d mC\log(N),$$

where we used the inequality $\sum_\omega q_\omega^{-2} \lesssim_d \log(N)$ in the last step (see the proof of Lemma 4.1 in [4, Sec. SM3], noting their definition of q_ω is the same as ours). Dividing out by m and C yields

$$\Gamma(\hat{\boldsymbol{p}}) = \frac{1}{C} \lesssim_d \log(N),$$

which is what we wanted to show. This establishes near-optimality of $\hat{\boldsymbol{p}}$ with

$$\log\left(\frac{N}{m^{1/d}}\right) \lesssim \Gamma(\hat{\boldsymbol{p}}) \lesssim_d \log(N).$$

An example of $\hat{\boldsymbol{p}}$ for $d = 2$ can be seen in Fig. 3.1 (left) together with a random draw corresponding to $\hat{\boldsymbol{p}}$ (right). Note that this resembles the theoretically optimal sampling mask shown in [4, Sec. 4.1] for $d = 2$. However, the approach considered therein involves random sampling with replacement.

Finally, we close this discussion with a couple of comments. First, unlike in [4, Lem. 4.1], our general bound $\Gamma(\boldsymbol{p}) \gtrsim \log(Nm^{-1/d})$ depends on m. This is an artifact of deterministic samples arising in the Bernoulli model. Second, we emphasize that our sampling strategy $\hat{\boldsymbol{p}}$ is near-optimal in a theoretical sense. In practice, recovery performance can be improved by ad hoc or heuristic approaches (see [5, Chap. 4] for instance).

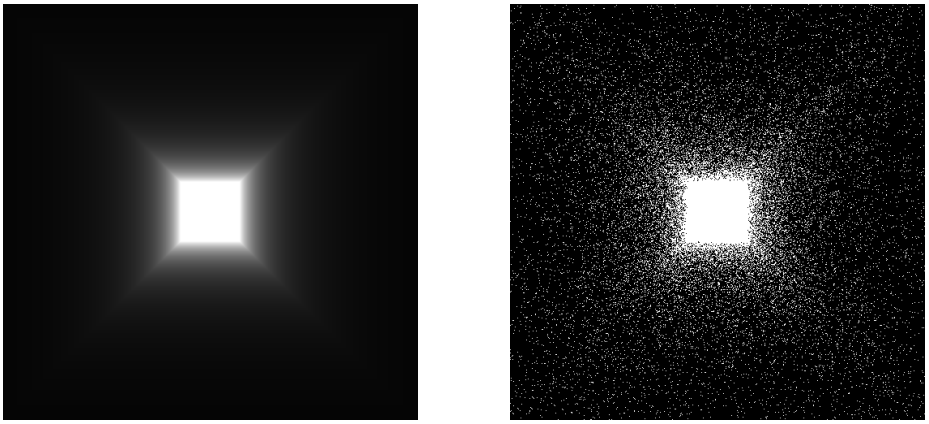


Figure 3.1: Near-optimal variable sampling Bernoulli vector (left) and mask (right) for $d = 2$ with $N = 512$, 10% sampling rate and centred zero-frequency. For the Bernoulli vector, dark and light pixels correspond to near zero or near one probability, respectively.

3.2.4 Bounds for expected number of measurements

Recall that we opted to use the Bernoulli model to enforce sampling each frequency at most once, given a Bernoulli vector of probabilities. This contrasts with other forms of sampling done in compressed sensing (e.g. each frequency is sampled independently from a density over the frequencies). The use of the Bernoulli model is necessary for the algorithm we use to perform image reconstruction, but has the downside that the number of measurements itself becomes a random variable. Here we show that given a random draw of frequencies $\Omega \sim \text{Ber}(\llbracket N \rrbracket, m, \boldsymbol{p})$, then $|\Omega|$ is near m with high probability. This is straightforward to show using *concentration inequalities* (e.g. see [95, Chap. 2]). For example, we can show $|\Omega|$ deviates from m with exponentially decaying probability.

Proposition 3.2.7. *If* $\Omega \sim \text{Ber}(\llbracket N \rrbracket, m, \boldsymbol{p})$*, then* $\mathbb{P}\left(||\Omega| - m| \geq t\right) \leq \mathrm{e}^{-2t^2/N}$.

Proof. For $j \in \llbracket N \rrbracket$, let X_j denote the random variable where $X_j = 1$ if $j \in \Omega$ (with probability p_j) and $X_j = 0$ otherwise. Then $\{X_i\}$ are all independent and the result follows from Hoeffding's inequality [95, Thm. 2.2.6]. □

Note one can achieve tighter bounds by using other concentration inequalities (e.g. Chernoff bounds), but sacrifice being practical to work with and interpret. The use of Hoeffding's inequality is both simple and sufficient for what we claimed.

3.3 Stacking scheme with NESTA

Recall that NESTA is the ℓ^1-minimization algorithm we use for image reconstruction via TV minimization (2.3.2). Here we present the notion of a *stacking scheme* with Bernoulli sampling. The stacking scheme is primarily inspired to derive a practical implementation of NESTANets, i.e. unrolling of NESTA, and the theoretical requirements for recovery in

Fourier imaging. For the former reason, this goes hand-in-hand with using the Bernoulli model. Regarding the latter, to carry out the recovery analysis in Section 4.2, both uniform and variable subsampled measurements are used. They are generated independently, and then "stacked" together, possibly duplicating existing measurements to form a sampling mask compatible with NESTA.

The *stacking* scheme is constructed as follows. We are given two sampling masks, uniform and variable, that is $\Omega_1 \sim \mathrm{Ber}(\llbracket N^d \rrbracket, m/2)$ and $\Omega_2 \sim \mathrm{Ber}(\llbracket N^d \rrbracket, m/2, \boldsymbol{p})$ for some Bernoulli vector $\boldsymbol{p}$. This defines two sets of measurements

$$\tilde{\boldsymbol{y}}_i = \boldsymbol{B}_i \boldsymbol{x} + \boldsymbol{e}_i \in \mathbb{C}^{|\Omega_i|}, \qquad \boldsymbol{B}_i = \frac{1}{\sqrt{m}} \boldsymbol{P}_{\Omega_i} \boldsymbol{F} \in \mathbb{C}^{|\Omega_i| \times N^d}, \qquad i = 1, 2,$$

where $\boldsymbol{x} \in \mathbb{C}^{N^d}$ is the image to recover. By virtue of the Bernoulli model, each frequency is sampled at most twice.

Now the goal is to extend Ω_1 and Ω_2 to a common superset Ω so that the measurement matrix $\boldsymbol{B} = m^{-1/2} \boldsymbol{P}_\Omega \boldsymbol{F} \in \mathbb{C}^{|\Omega| \times N^d}$ is the same for both sets of measurements. A suitable choice would be $\Omega = \Omega_1 \cup \Omega_2$. Consequently, we must also insert additional entries to $\tilde{\boldsymbol{y}}_1$ and $\tilde{\boldsymbol{y}}_2$ to be compatible with $\boldsymbol{B}$ in both size and correspondence of entries with sampled frequencies.

To do this precisely with $\Omega = \Omega_1 \cup \Omega_2$, extend $\tilde{\boldsymbol{y}}_1$ by taking the measurements indexed by $\Omega_2 \backslash \Omega_1$ in $\tilde{\boldsymbol{y}}_2$ and canonically insert them into $\tilde{\boldsymbol{y}}_1$. This defines the extended measurements $\boldsymbol{y}_1$. Symmetrically, extend $\tilde{\boldsymbol{y}}_2$ by taking measurements indexed by $\Omega_1 \backslash \Omega_2$ in $\tilde{\boldsymbol{y}}_1$ and insert them into $\tilde{\boldsymbol{y}}_2$. This yields the extended measurements $\boldsymbol{y}_2$. Therefore, we can express the extended measurements and measurement matrix using the *stacked form*

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}, \qquad \boldsymbol{A} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{B} \end{bmatrix} \in \mathbb{C}^{2|\Omega| \times N^d}, \ \boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} \in \mathbb{C}^{2|\Omega|}.$$

The vector $\boldsymbol{y}$ is referred to as the *stacked measurements*. Note that we do not have $\boldsymbol{A}\boldsymbol{A}^* = c\boldsymbol{I}$ for some constant $c > 0$, as required for NESTA. Instead we have

$$\boldsymbol{A}\boldsymbol{A}^* = \frac{N^d}{m} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{I} \end{bmatrix},$$

which motivates deriving a modified version of NESTA in Chapter 4 that uses stacked measurements. Moreover, up to a row permutation, the stacked matrix $\boldsymbol{A}$ can be expressed as

$$\boldsymbol{A} = \frac{1}{\sqrt{m}} \begin{bmatrix} \boldsymbol{P}_{\Omega_1} \\ \boldsymbol{P}_{\Omega_2} \\ \boldsymbol{P}_{\Omega_3} \end{bmatrix} \boldsymbol{F},$$

where Ω_1, Ω_2 are as before and $\Omega_3 = \Omega_1 \triangle \Omega_2$ is the *symmetric difference of* Ω_1 *and* Ω_2. Specifically, we have $\Omega_1 \triangle \Omega_2 := (\Omega_1 \cup \Omega_2) \backslash (\Omega_1 \cap \Omega_2)$. This serves to separate the uniform and variable density measurements to prove gradient and image recovery in Section 4.2.

Finally, we quickly comment on the size of Ω in expectation. Since the samples are independent, we have

$$\mathbb{E}(|\Omega_1 \cap \Omega_2|) = \sum_{i=1}^{N^d} \mathbb{P}(i \in \Omega_1) \cdot \mathbb{P}(i \in \Omega_2) = \sum_{i=1}^{N^d} \frac{m}{2N^d} \cdot p_i = \frac{m^2}{4N^d}.$$

Thus $\mathbb{E}(|\Omega|) = m - \frac{m^2}{4N^d}$ by the inclusion-exclusion principle. In particular, the expectation grows linearly in m, i.e. $\frac{3}{4}m \leq \mathbb{E}(|\Omega|) \leq m$. Succinctly, we express this as $\mathbb{E}(|\Omega|) \asymp m$. Lastly, the same arguments from Proposition 3.2.7 can be applied to show $|\Omega|$ deviates from its expected value with exponentially decaying probability.

Chapter 4

Solving Fourier imaging problems with TV minimization

Here we present the technical details of the algorithm we use to solve Fourier imaging problems via TV minimization. First, we introduce and derive *stacked NESTA*, a first-order optimization algorithm based on NESTA [13]. In particular, stacked NESTA involves computing orthogonal projections onto the QCBP constraint set, which is computationally exact and efficient for stacked matrices introduced in Section 3.3. Second, we combine NESTA error bounds and the results in Chapter 3 to prove accurate and stable recovery via stacked NESTA for Fourier imaging. This is based on the proof techniques found in [5, Chap. 8]. Furthermore, we describe a restart procedure that theoretically guarantees an exponential decay in reconstruction error, thus accelerating recovery.

4.1 Solutions by gradient-based optimization via smoothing

4.1.1 The NESTA algorithm and error bound

NESTA [13,14] is an algorithm for ℓ^1-minimization, which is derived from *Nesterov's method* with smoothing [73]. Nesterov's method is a general accelerated projected gradient-descent algorithm, with update steps expressed as constrained optimization problems. The steps have closed-form expressions if exact formulas are known for the orthogonal projection onto the constraint set. NESTA is obtained by applying Nesterov's method with smoothing to QCBP (2.2.1), which assumes that $\boldsymbol{A}\boldsymbol{A}^* = c\boldsymbol{I}$ for some $c > 0$ [13] to efficiently and exactly compute projections. The resulting algorithm is given in Algorithm 1.

A discussion of computing orthogonal projections with general $\boldsymbol{A}$ for QCBP is found in [14, Sec. 3.7], which provides a procedure when the singular value decomposition (SVD) of $\boldsymbol{A}$ is known. We avoid this approach since, first, computing the SVD of $\boldsymbol{A}$ for imaging is often intractable, and second, one has to solve a nonlinear equation in each update step, making unrolling an impractical endeavour. This motivates modifying NESTA to compute

Algorithm 1: NESTA for QCBP.

Input : Vectors $\boldsymbol{y} \in \mathbb{C}^m$, matrix $\boldsymbol{A} \in \mathbb{C}^{m \times N}$ satisfying $\boldsymbol{A}\boldsymbol{A}^* = c\boldsymbol{I}$ for $c > 0$, parameters $\eta > 0$, $\mu > 0$, real sequences $\{\alpha_n\}_{j=0}^{\infty}$, $\{\tau_n\}_{n=0}^{\infty}$, number of iterations $t > 0$, and $\boldsymbol{z}_0 \in \mathbb{C}^N$.

Output: The vector $\boldsymbol{x}_{t-1}$, which estimates a minimizer of (2.2.1) with $\boldsymbol{A}$ and $\boldsymbol{y}$ satisfying (4.1.5).

for $n = 0, 1, \ldots, t-1$ **do**
  Compute $\boldsymbol{x}_n$:
  $\boldsymbol{q} \leftarrow \boldsymbol{z}_n - (\|\boldsymbol{W}\|_{\ell^2}^2/\mu)^{-1}\boldsymbol{W}(\mathcal{T}_\mu(\boldsymbol{W}^*\boldsymbol{z}_n))$
  $\lambda \leftarrow \max\{0,\ \eta^{-1}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{q}\|_{\ell^2} - 1\}$
  $\boldsymbol{x}_n \leftarrow \frac{\lambda}{c}\boldsymbol{B}^*(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}) + \boldsymbol{q}$.
  Compute $\boldsymbol{v}_n$:
  $\boldsymbol{q} \leftarrow \boldsymbol{z}_0 - (\|\boldsymbol{W}\|_{\ell^2}^2/\mu)^{-1}\sum_{i=0}^{n}\alpha_i\boldsymbol{W}(\mathcal{T}_\mu(\boldsymbol{W}^*\boldsymbol{z}_i))$
  $\lambda \leftarrow \max\{0,\ \eta^{-1}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{q}\|_{\ell^2} - 1\}$
  $\boldsymbol{v}_n \leftarrow \frac{\lambda}{c}\boldsymbol{B}^*(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}) + \boldsymbol{q}$.
  Compute $\boldsymbol{z}_{n+1} \leftarrow \tau_n\boldsymbol{v}_n + (1 - \tau_n)\boldsymbol{x}_n$.
end

exact projections with stacked measurement matrices from Section 3.3, enabling a practical unrolling of stacked NESTA.

Since Nesterov's method [73] requires computing derivatives of the objective function, we cannot directly apply it to QCBP as the ℓ^1-norm is not differentiable everywhere. To circumvent this, we apply *smoothing* [12, 73] (see [11, Sec. 10.8] for recent discussion), which defines a new optimization problem amenable to Nesterov's method. This is done by using a smooth approximation of the nonsmooth objective function. The solution of the smoothed problem will approximate a solution to QCBP, with a small tradeoff in requiring more iterations to compute a solution when using a better smooth approximation. Given $\mu > 0$, we consider the *smoothed* QCBP problem

$$\min_{\boldsymbol{z} \in \mathbb{C}^N} \|\boldsymbol{W}^*\boldsymbol{z}\|_{\ell^1,\mu} \text{ subject to } \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{y}\|_{\ell^2} \leq \eta, \tag{4.1.1}$$

where

$$\|\boldsymbol{z}\|_{\ell^1,\mu} = \sum_{i=1}^{M} H_\mu(z_i), \qquad \boldsymbol{z} \in (z_i)_{i=1}^M \in \mathbb{C}^M, \tag{4.1.2}$$

is the smoothed ℓ^1-norm. The function $H_\mu : \mathbb{C} \to \mathbb{R}_+$ is the *Huber function*, defined by

$$H_\mu(z) = \begin{cases} \frac{1}{2\mu}|z|^2 & |z| \leq \mu \\ |z| - \frac{\mu}{2} & |z| > \mu \end{cases}, \qquad z \in \mathbb{C}. \tag{4.1.3}$$

Algorithm 2: Stacked NESTA for QCBP.

Input : Vectors $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{C}^{\frac{m}{2}}$, matrix $\boldsymbol{B} \in \mathbb{C}^{\frac{m}{2} \times N}$ satisfying $\boldsymbol{B}\boldsymbol{B}^* = c\boldsymbol{I}$ for $c > 0$, parameters $\eta > 0$, $\mu > 0$, real sequences $\{\alpha_n\}_{j=0}^{\infty}$, $\{\tau_n\}_{n=0}^{\infty}$, number of iterations $t > 0$, and $\boldsymbol{z}_0 \in \mathbb{C}^N$.

Output: The vector $\boldsymbol{x}_{t-1}$, which estimates a minimizer of (2.2.1) with $\boldsymbol{A}$ and $\boldsymbol{y}$ satisfying (4.1.5).

for $n = 0, 1, \ldots, t-1$ **do**
  Compute $\boldsymbol{x}_n$:
  $\boldsymbol{q} \leftarrow \boldsymbol{z}_n - (\|\boldsymbol{W}\|_{\ell^2}^2/\mu)^{-1}\boldsymbol{W}(\mathcal{T}_\mu(\boldsymbol{W}^*\boldsymbol{z}_n))$
  $\lambda \leftarrow \max\left\{0,\ \frac{1}{2}\left(1 - \frac{\sqrt{2\eta^2 - \|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_{\ell^2}^2}}{\|\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}\|_{\ell^2}}\right)\right\}$
  $\boldsymbol{x}_n \leftarrow \frac{\lambda}{c}\boldsymbol{B}^*(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}) + \boldsymbol{q}$.
  Compute $\boldsymbol{v}_n$:
  $\boldsymbol{q} \leftarrow \boldsymbol{z}_0 - (\|\boldsymbol{W}\|_{\ell^2}^2/\mu)^{-1}\sum_{i=0}^{n}\alpha_i\boldsymbol{W}(\mathcal{T}_\mu(\boldsymbol{W}^*\boldsymbol{z}_i))$
  $\lambda \leftarrow \max\left\{0,\ \frac{1}{2}\left(1 - \frac{\sqrt{2\eta^2 - \|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_{\ell^2}^2}}{\|\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}\|_{\ell^2}}\right)\right\}$
  $\boldsymbol{v}_n \leftarrow \frac{\lambda}{c}\boldsymbol{B}^*(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}) + \boldsymbol{q}$.
  Compute $\boldsymbol{z}_{n+1} \leftarrow \tau_n\boldsymbol{v}_n + (1 - \tau_n)\boldsymbol{x}_n$.
end

Here the parameter μ is known as the *smoothing parameter*, and controls the approximation of the ℓ^1-norm in the sense that if $\mu \searrow 0$, then $\|\cdot\|_{\ell^1,\mu} \to \|\cdot\|_{\ell^1}$ uniformly. We refer to the vector function $\mathcal{T}_\mu : \mathbb{C}^M \to \mathbb{C}^M$ formally as the *gradient of* $\|\cdot\|_{\ell^1,\mu}$, which is defined elementwise by

$$(\mathcal{T}_\mu(\boldsymbol{z}))_i = \begin{cases} \frac{z_i}{\mu}, & |z_i| \le \mu \\ \frac{z_i}{|z_i|}, & |z_i| > \mu \end{cases}, \qquad i = 1, \ldots, M. \tag{4.1.4}$$

One may verify that $\mathcal{T}_\mu$ is indeed the gradient of the smoothed ℓ^1-norm when restricting ourselves to $\mathbb{R}^M$. We motivate the choice of formulas for $\|\cdot\|_{\ell^1,\mu}$ and $\mathcal{T}_\mu$ in Section 4.1.4. Recall that the stacking scheme from Section 3.3 involves a matrix $\boldsymbol{A} \in \mathbb{C}^{m \times N}$ and column vector $\boldsymbol{y} \in \mathbb{C}^m$ with even m, satisfying the stacking form

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{B} \end{bmatrix}, \qquad \boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix}, \qquad \boldsymbol{B}\boldsymbol{B}^* = c\boldsymbol{I}, \qquad \boldsymbol{B} \in \mathbb{C}^{\frac{m}{2} \times N},\ \boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{C}^{\frac{m}{2}},\ c > 0. \tag{4.1.5}$$

With this, Nesterov's method for (4.1.1), assuming $\boldsymbol{A}$ and $\boldsymbol{y}$ are expressed in stacking form, is given by Algorithm 2. We dub this algorithm as *stacked NESTA*.

We now state an objective error bound for the stacked NESTA iterates. This makes precise how solving the smoothed problem approximates a solution to the nonsmooth problem, at least in terms of the objective error.

Algorithm 3: Nesterov's method

Input : A K-smooth function f and set $Q \subseteq \mathbb{R}^N$ as in (4.1.6), prox-function p_p with strong convexity constant σ_p and unique minimizer $\boldsymbol{z}_0 \in Q$, sequences $\{\alpha_n\}_{n=0}^{\infty}$, $\{\tau_n\}_{n=0}^{\infty}$, and number of iterations $t > 0$.

Output: The vector $\boldsymbol{x}_{t-1}$, which estimates a minimizer of (4.1.6).

for $n = 0, 1, \ldots, t-1$ **do**

$\quad \boldsymbol{x}_n \leftarrow \underset{\boldsymbol{x} \in Q}{\operatorname{argmin}} \frac{K}{2}\|\boldsymbol{x} - \boldsymbol{z}_n\|_{\ell^2}^2 + \langle \nabla f(\boldsymbol{z}_n), \boldsymbol{x} - \boldsymbol{z}_n \rangle$

$\quad \boldsymbol{v}_n \leftarrow \underset{\boldsymbol{x} \in Q}{\operatorname{argmin}} \frac{K}{\sigma_p} p_p(\boldsymbol{x}) + \sum_{i=0}^{n} \alpha_i \langle \nabla f(\boldsymbol{z}_i), \boldsymbol{x} - \boldsymbol{z}_i \rangle$

$\quad \boldsymbol{z}_{n+1} \leftarrow \tau_n \boldsymbol{v}_n + (1 - \tau_n)\boldsymbol{x}_n$

end

Lemma 4.1.1. *Suppose $\boldsymbol{A}$ and $\boldsymbol{y}$ satisfies (4.1.5). Let $\boldsymbol{x}_n$ be the result of the nth iteration of Algorithm 2 with initial vector $\boldsymbol{z}_0 \in \mathbb{C}^N$ satisfying $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}_0\|_{\ell^2} \leq \eta$, and parameters $\alpha_i = \frac{i+1}{2}$ and $\tau_i = \frac{2}{i+3}$. Then*

$$\|\boldsymbol{W}^*\boldsymbol{x}_n\|_{\ell^1} - \|\boldsymbol{W}^*\boldsymbol{x}\|_{\ell^1} \leq \frac{2\|\boldsymbol{W}\|_{\ell^2}^2}{\mu(n+1)^2}\|\boldsymbol{x} - \boldsymbol{z}_0\|_{\ell^2}^2 + \frac{M\mu}{2}, \qquad \forall \boldsymbol{x} : \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_{\ell^2} \leq \eta.$$

For the proof, see Section 4.1.5. Observe that the second term $\frac{M\mu}{2}$ of the error bound is the error from approximation by smoothing. This term can be shrunk by taking μ to be small, but as a tradeoff, one needs to run more iterations to sufficiently reduce the first term in the error bound.

The remainder of this section is dedicated to deriving smoothed QCBP, Algorithm 2, and proving Lemma 4.1.1.

4.1.2 A primer on Nesterov's method

Before proceeding with deriving stacked NESTA, we quickly introduce Nesterov's method. To do this, we need a definition used commonly when discussing gradient-based optimization algorithms.

Definition 4.1.2 (K-smoothness [11, Defn. 5.1])**.** A function $f : D \to \mathbb{R}$ with effective domain $D \subseteq \mathbb{R}^N$ is said to be *K-smooth over a set $Q \subseteq D$* if it is differentiable over Q and satisfies

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_{\ell^2} \leq K\|\boldsymbol{x} - \boldsymbol{y}\|_{\ell^2} \text{ for all } \boldsymbol{x}, \boldsymbol{y} \in Q.$$

$\Diamond$

NESTA is a specific implementation of *Nesterov's method*, an accelerated projected gradient-based optimization algorithm. In particular, it tackles general constrained convex

optimization problems of the form

$$\min_{\boldsymbol{x}\in Q} f(\boldsymbol{x}), \tag{4.1.6}$$

where $Q \subseteq \mathbb{R}^N$ is closed and convex, and $f : D \to \mathbb{R}$ is a closed, convex and K-smooth function with effective domain $D \subseteq \mathbb{R}^N$ containing Q. Nesterov's method is presented in Algorithm 3. The algorithm makes use of a *prox-function* $p_p : D \to \mathbb{R}$, a strongly convex function which specifies an initial point with its unique minimizer. A standard choice that we use is $p_p(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}_0\|_{\ell^2}$ for given $\boldsymbol{z}_0 \in Q$, where $\sigma_p = 1$ here.

Nesterov's method is an "accelerated" algorithm in the sense that, specific choices of sequences $\{\alpha_n\}$ and $\{\tau_n\}$ lead to the objective error $f(\boldsymbol{x}_n) - \hat{f} = \mathcal{O}(n^{-2})$. This improves upon the typical sublinear convergence rates of many first-order methods [19, 74], which guarantee an objective error of order $\mathcal{O}(n^{-1})$ or $\mathcal{O}(n^{-1/2})$ in the nth iterate.

4.1.3 NESTA derivation: real-valued data

In this section, the data considered only involve real numbers. Let us consider the problem

$$\min_{\boldsymbol{z}\in\mathbb{R}^N} \phi(\boldsymbol{z}) \text{ subject to } \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}\|_{\ell^2} \leq \eta, \tag{4.1.7}$$

where $\phi : \mathbb{R}^N \to \mathbb{R}$ is a K-smooth convex function, $\boldsymbol{A} \in \mathbb{R}^{m\times N}$, $\boldsymbol{y} \in \mathbb{R}^m$, and $\eta > 0$. This is an extension of QCBP where the objective has been replaced with a general differentiable convex function ϕ. The purpose of considering (4.1.7) is to show we can obtain closed-form projections onto the constraint set when the matrix $\boldsymbol{A}$ adheres to the stacking scheme. This is necessary to compute the update steps of Nesterov's method exactly. Later in this section we address when ϕ is nonsmooth, which is relevant to when we consider applying Nesterov's method to solve QCBP.

Using Algorithm 3, the objective function $f = \phi$ is being minimized over the set $Q = \{\boldsymbol{z} : \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}\|_{\ell^2} \leq \eta\} \subseteq \mathbb{R}^N$. To derive explicit formulas for $\boldsymbol{x}_n$ and $\boldsymbol{v}_n$ (lines 2 and 3 of Algorithm 3, respectively), we require some additional assumptions. First, we choose the (primal) prox-function $p_p(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}_0\|_{\ell^2}^2$ for fixed $\boldsymbol{z}_0 \in Q$, noting that the strong convexity constant of p_p is $\sigma_p = 1$. Second, taking m to be even, we assume $\boldsymbol{A}$ and $\boldsymbol{y}$ take on the stacked form

$$\boldsymbol{A} = \begin{bmatrix}\boldsymbol{B}\\ \boldsymbol{B}\end{bmatrix}, \qquad \boldsymbol{y} = \begin{bmatrix}\boldsymbol{y}_1\\ \boldsymbol{y}_2\end{bmatrix}, \qquad \boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{R}^{\frac{m}{2}},\ \boldsymbol{B} \in \mathbb{R}^{\frac{m}{2}\times N},\ \boldsymbol{B}\boldsymbol{B}^\top = c\boldsymbol{I} \text{ with } c > 0, \tag{4.1.8}$$

which is (4.1.5) restricted to real numbers. For the prescribed prox-function, the update rules for $\boldsymbol{x}_n$ and $\boldsymbol{v}_n$ become

$$\begin{aligned}\boldsymbol{x}_n &= \underset{\boldsymbol{z}:\|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{y}\|_{\ell^2}\leq\eta}{\operatorname{argmin}} \frac{K}{2}\|\boldsymbol{z}-\boldsymbol{z}_n\|_{\ell^2}^2 + \langle\nabla\phi(\boldsymbol{z}_n), \boldsymbol{z}-\boldsymbol{z}_n\rangle,\\ \boldsymbol{v}_n &= \underset{\boldsymbol{z}:\|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{y}\|_{\ell^2}\leq\eta}{\operatorname{argmin}} \frac{K}{2}\|\boldsymbol{z}-\boldsymbol{z}_0\|_{\ell^2}^2 + \sum_{j=0}^{n}\alpha_j\langle\nabla\phi(\boldsymbol{z}_j), \boldsymbol{z}-\boldsymbol{z}_j\rangle.\end{aligned}$$

Both update steps involve finding a minimizer to a quadratic function subject to a quadratic constraint. As is done in [5, Section 7.6.3], we can equivalently express the $\boldsymbol{v}_n$ update as

$$\boldsymbol{v}_n = \underset{\boldsymbol{z}:\|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{y}\|_{\ell^2}\leq\eta}{\operatorname{argmin}} \frac{K}{2}\|\boldsymbol{z}-\boldsymbol{z}_0\|_{\ell^2}^2 + \left\langle\sum_{j=0}^{n}\alpha_j\nabla\phi(\boldsymbol{z}_j), \boldsymbol{z}-\boldsymbol{z}_0\right\rangle, \tag{4.1.9}$$

so both the $\boldsymbol{x}_n$ and $\boldsymbol{v}_n$ update steps can be expressed as

$$\begin{aligned}\underset{\boldsymbol{z}:\|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{y}\|_{\ell^2}\leq\eta}{\operatorname{argmin}} \frac{K}{2}\|\boldsymbol{z}-\boldsymbol{v}\|_{\ell^2}^2 + \langle\boldsymbol{u}, \boldsymbol{z}-\boldsymbol{v}\rangle &= \underset{\boldsymbol{z}:\|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{y}\|_{\ell^2}\leq\eta}{\operatorname{argmin}} \left\|\sqrt{\frac{K}{2}}(\boldsymbol{z}-\boldsymbol{v}) + \frac{\boldsymbol{u}}{\sqrt{2K}}\right\|_{\ell^2}^2\\ &= \underset{\boldsymbol{z}:\|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{y}\|_{\ell^2}\leq\eta}{\operatorname{argmin}} \left\|\boldsymbol{z}-\left(\boldsymbol{v}-\frac{\boldsymbol{u}}{K}\right)\right\|_{\ell^2}^2,\end{aligned} \tag{4.1.10}$$

for some fixed $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^N$. Specifically, (4.1.10) is the orthogonal projection of $\boldsymbol{v}-\boldsymbol{u}/K$ onto the constraint set. We consider a convenient reformulation of (4.1.10), where we aim to compute a closed form expression for

$$\underset{\boldsymbol{\xi}:\|\boldsymbol{A}\boldsymbol{\xi}-\boldsymbol{b}\|_{\ell^2}\leq\eta}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{\xi}\|_{\ell^2}^2. \tag{4.1.11}$$

This equation is obtained from (4.1.10) using the change of variables $\boldsymbol{\xi} = \boldsymbol{z}-\boldsymbol{w}$ and $\boldsymbol{b} = \boldsymbol{y}-\boldsymbol{A}\boldsymbol{w}$, where $\boldsymbol{w} = \boldsymbol{v}-\boldsymbol{u}/K$. Under special circumstances, we can explicitly find (4.1.11) by algebraically solving equations defined by the Karush-Kuhn-Tucker (KKT) conditions. To do this, one usually imposes $\boldsymbol{A}$ to have certain structure, such as the orthonormal row condition $\boldsymbol{A}\boldsymbol{A}^\top = c\boldsymbol{I}$ for $c > 0$. As we will see, (4.1.11) can be computed explicitly with the stacking assumption (4.1.8).

Let us proceed to find (4.1.11) with $\boldsymbol{A}$ satisfying (4.1.8). For additional notation, we express the vector $\boldsymbol{b} \in \mathbb{R}^m$ as a stacked vector by

$$\boldsymbol{b} = \begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\end{bmatrix}, \qquad \boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathbb{R}^{\frac{m}{2}}.$$

First we have

$$\boldsymbol{A}^\top \boldsymbol{A} = 2\boldsymbol{B}^\top \boldsymbol{B} \in \mathbb{R}^{N\times N}, \qquad \boldsymbol{A}\boldsymbol{A}^\top = c\begin{bmatrix} \boldsymbol{I} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{I} \end{bmatrix} \in \mathbb{R}^{m\times m}, \tag{4.1.12}$$

where $\boldsymbol{I}$ here is the $\frac{m}{2} \times \frac{m}{2}$ identity matrix. Next, observe that (4.1.11) defines a convex optimization problem. Writing $F(\boldsymbol{\xi}) = \frac{1}{2}\|\boldsymbol{\xi}\|_{\ell^2}^2$ and $G(\boldsymbol{\xi}) = \frac{1}{2}\left(\|\boldsymbol{A}\boldsymbol{\xi} - \boldsymbol{b}\|_{\ell^2}^2 - \eta^2\right)$, then the KKT conditions say for some $\lambda \in \mathbb{R}$ and $\boldsymbol{\xi} \in \mathbb{R}^N$,

$$\nabla F(\boldsymbol{\xi}) + \lambda \nabla G(\boldsymbol{\xi}) = \boldsymbol{0}, \tag{4.1.13}$$

$$G(\boldsymbol{\xi}) \leq 0, \tag{4.1.14}$$

$$\lambda G(\boldsymbol{\xi}) = 0, \tag{4.1.15}$$

$$\lambda \geq 0, \tag{4.1.16}$$

if and only if $\boldsymbol{\xi}$ solves (4.1.11) and λ solves the respective dual problem. The point $\boldsymbol{\xi}$ from a pair $(\boldsymbol{\xi}, \lambda)$ solving the KKT system will be unique, since the objective in (4.1.11) is strongly convex. Using $\nabla F(\boldsymbol{\xi}) = \boldsymbol{\xi}$ and $\nabla G(\boldsymbol{\xi}) = \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{\xi} - \boldsymbol{b})$, the first-order condition (4.1.13) is

$$\nabla F(\boldsymbol{\xi}) + \lambda G(\boldsymbol{\xi}) = \boldsymbol{\xi} + \lambda \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{\xi} - \lambda \boldsymbol{A}^\top \boldsymbol{b} = \left(I + \lambda \boldsymbol{A}^\top \boldsymbol{A}\right)\boldsymbol{\xi} - \lambda \boldsymbol{A}^\top \boldsymbol{b} = \boldsymbol{0}.$$

Multiplying the above by

$$\left(\boldsymbol{I} + \lambda \boldsymbol{A}^\top \boldsymbol{A}\right)^{-1} = \left(\boldsymbol{I} - \frac{\lambda}{1+2c\lambda}\boldsymbol{A}^\top \boldsymbol{A}\right)$$

and rearranging gives

$$\begin{aligned}
\boldsymbol{\xi} &= \left(\boldsymbol{I} - \frac{\lambda}{1+2c\lambda}\boldsymbol{A}^\top \boldsymbol{A}\right)(\lambda \boldsymbol{A}^\top \boldsymbol{b}) \\
&= \lambda \boldsymbol{A}^\top \boldsymbol{b} - \frac{\lambda^2}{1+2c\lambda}\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{A}^\top \boldsymbol{b} \\
&= \lambda \boldsymbol{A}^\top \boldsymbol{b} - \frac{c\lambda^2}{1+2c\lambda}\boldsymbol{A}^\top \begin{bmatrix} \boldsymbol{I} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{I} \end{bmatrix} \boldsymbol{b} \\
&= \lambda \begin{bmatrix} \boldsymbol{B}^\top & \boldsymbol{B}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix} - \frac{c\lambda^2}{1+2c\lambda} \begin{bmatrix} \boldsymbol{B}^\top & \boldsymbol{B}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{b}_1 + \boldsymbol{b}_2 \\ \boldsymbol{b}_1 + \boldsymbol{b}_2 \end{bmatrix} \\
&= \left(\lambda - \frac{2c\lambda^2}{1+2c\lambda}\right)\boldsymbol{B}^\top(\boldsymbol{b}_1 + \boldsymbol{b}_2) \\
&= \frac{\lambda}{1+2c\lambda}\boldsymbol{B}^\top(\boldsymbol{b}_1 + \boldsymbol{b}_2),
\end{aligned}$$

where we used (4.1.12) to simplify. Thus

$$\boldsymbol{\xi} = \frac{\lambda}{1+2c\lambda}\boldsymbol{B}^\top(\boldsymbol{b}_1+\boldsymbol{b}_2) \tag{4.1.17}$$

is the unique vector equal to (4.1.11), expressed in terms of the KKT multiplier λ. To find a closed form expression for λ, first note that if $\lambda = 0$, then $\boldsymbol{\xi} = \boldsymbol{0}$. Otherwise if $\lambda > 0$, then from the slackness condition (4.1.15) we have $\|\boldsymbol{A}\boldsymbol{\xi} - \boldsymbol{b}\|_{\ell^2} = \eta$. Multiplying (4.1.17) by $\boldsymbol{A}$ and subtracting by $\boldsymbol{b}$ yields

$$\boldsymbol{A}\boldsymbol{\xi} - \boldsymbol{b} = \frac{\lambda}{1+2c\lambda}\boldsymbol{A}\boldsymbol{B}^\top(\boldsymbol{b}_1+\boldsymbol{b}_2) - \boldsymbol{b} = \frac{c\lambda}{1+2c\lambda}\begin{bmatrix}\boldsymbol{b}_1+\boldsymbol{b}_2\\ \boldsymbol{b}_1+\boldsymbol{b}_2\end{bmatrix} - \boldsymbol{b}$$

For brevity, we write

$$\tilde{\boldsymbol{b}} = \begin{bmatrix}\boldsymbol{b}_1+\boldsymbol{b}_2\\ \boldsymbol{b}_1+\boldsymbol{b}_2\end{bmatrix}.$$

Making another change of variables, we denote $\rho = \frac{c\lambda}{1+2c\lambda}$. Taking the ℓ^2-norm and squaring the previous calculation

$$\eta^2 = \|A\boldsymbol{\xi} - \boldsymbol{b}\|_{\ell^2}^2 = \|\rho\tilde{\boldsymbol{b}} - \boldsymbol{b}\|_{\ell^2}^2 = \rho^2\|\tilde{\boldsymbol{b}}\|_{\ell^2}^2 - 2\rho\langle\tilde{\boldsymbol{b}}, \boldsymbol{b}\rangle + \|\boldsymbol{b}\|_{\ell^2}^2,$$

gives a quadratic equation in ρ. Assuming $\tilde{\boldsymbol{b}} \neq \boldsymbol{0}$, we obtain the roots

$$\rho = \frac{\langle\tilde{\boldsymbol{b}}, \boldsymbol{b}\rangle \pm \sqrt{\langle\tilde{\boldsymbol{b}}, \boldsymbol{b}\rangle^2 - \|\tilde{\boldsymbol{b}}\|_{\ell^2}^2\left(\|\boldsymbol{b}\|_{\ell^2}^2 - \eta^2\right)}}{\|\tilde{\boldsymbol{b}}\|_{\ell^2}^2}.$$

Further simplifying the formula for ρ by using $\|\tilde{\boldsymbol{b}}\|_{\ell^2}^2 = 2\langle\tilde{\boldsymbol{b}}, \boldsymbol{b}\rangle$ gives

$$\rho = \frac{1}{2}\left(1 \pm \sqrt{1 - 4\cdot\frac{\|\boldsymbol{b}\|_{\ell^2}^2 - \eta^2}{\|\tilde{\boldsymbol{b}}\|_{\ell^2}^2}}\right).$$

To identify which root to assign to ρ, since $\lambda \geq 0$ from (4.1.16) and

$$\rho = \frac{c\lambda}{1+2c\lambda} \quad \Longleftrightarrow \quad \lambda = \frac{\rho}{c(1-2\rho)},$$

it is necessary that $0 \leq \rho < \frac{1}{2}$. Thus, we take the negative-signed root. Therefore, provided $\tilde{\boldsymbol{b}} \neq \boldsymbol{0}$,

$$\lambda = \frac{\rho}{c(1-2\rho)}, \qquad \rho = \max\left\{0, \frac{1}{2}\left(1 - \sqrt{1 - 4\cdot\frac{\|\boldsymbol{b}\|_{\ell^2}^2 - \eta^2}{\|\tilde{\boldsymbol{b}}\|_{\ell^2}^2}}\right)\right\} \tag{4.1.18}$$

and thus we obtained a closed form expression for λ. This gives us the desired $\boldsymbol{\xi}$ for (4.1.11).

We now put everything together to obtain a closed form expression for (4.1.10). This will give us the exact formulas for $\boldsymbol{x}_n$ and $\boldsymbol{v}_n$ in Algorithm 3. Write $\boldsymbol{y}$ as the stacked vector

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix}, \qquad \boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{R}^{\frac{m}{2}},$$

and recall $\boldsymbol{A}$ in its stacked form (4.1.8) in terms of $\boldsymbol{B}$. For simplicity, we omit mention of the measurement matrix $\boldsymbol{A}$ and vector $\boldsymbol{y}$ when defining NESTA for the stacking scheme, instead referring to their block components $\boldsymbol{B}$ and $\boldsymbol{y}_1$, $\boldsymbol{y}_2$, respectively.

Using $\boldsymbol{b} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}$ and (4.1.12), one can show the identities

$$\|\tilde{\boldsymbol{b}}\|_{\ell^2}^2 = 2\|\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{w}\|_{\ell^2}^2, \qquad \frac{1}{2}\|\tilde{\boldsymbol{b}}\|_{\ell^2}^2 - 2\|\boldsymbol{b}\|_{\ell^2}^2 = -\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_{\ell^2}^2,$$

giving

$$1 - 4 \cdot \frac{\|\boldsymbol{b}\|_{\ell^2}^2 - \eta^2}{\|\tilde{\boldsymbol{b}}\|_{\ell^2}^2} = \frac{2\eta^2 - \|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_{\ell^2}^2}{\|\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{w}\|_{\ell^2}^2},$$

hence (4.1.18) can be written as

$$\lambda = \frac{\rho}{c(1-2\rho)}, \qquad \rho = \max\left\{0, \frac{1}{2}\left(1 - \frac{\sqrt{2\eta^2 - \|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_{\ell^2}^2}}{\|\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{w}\|_{\ell^2}}\right)\right\}.$$

Remark 4.1.3. There are two points of interest pertaining to this form of ρ. First, observe that because ρ is real-valued, the inequality $\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_{\ell^2} \leq \sqrt{2}\eta$ must hold. This gives a necessary condition for (4.1.10) to have a solution, and is otherwise infeasible. Therefore η must be correctly specified when given $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ a priori. The same inequality can be used as a lower bound for η to ensure feasibility.

Second, if the denominator $\|\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{w}\|_{\ell^2} = 0$, we assign $\rho = 0$. To see why this is reasonable, observe that if $\|\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{w}\|_{\ell^2} = 0$, then we have $\boldsymbol{B}\boldsymbol{w} = \frac{1}{2}(\boldsymbol{y}_1 + \boldsymbol{y}_2)$. Thus for any feasible point $\boldsymbol{x}$ one has

$$\begin{aligned} \|\boldsymbol{A}\boldsymbol{w} - \boldsymbol{y}\|_{\ell^2} &= \sqrt{\|\boldsymbol{B}\boldsymbol{w} - \boldsymbol{y}_1\|_{\ell^2}^2 + \|\boldsymbol{B}\boldsymbol{w} - \boldsymbol{y}_2\|_{\ell^2}^2} = \frac{\sqrt{2}}{2}\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_{\ell^2} \\ &= \frac{1}{2}\left\| \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} - \begin{bmatrix} \boldsymbol{y}_2 \\ \boldsymbol{y}_1 \end{bmatrix} \right\|_{\ell^2} \leq \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_{\ell^2} \leq \eta, \end{aligned}$$

where we used the triangle inequality in the second-last step. This shows $\boldsymbol{w}$ is a feasible point of (4.1.7). Therefore, in view of the update step general form (4.1.10), we must have $\boldsymbol{z} = \boldsymbol{w} = \boldsymbol{v} - \boldsymbol{u}/K$. To be consistent with the KKT conditions (4.1.13) to (4.1.16) for

Algorithm 4: Nesterov's method for (4.1.19).

Input : Vectors $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{R}^m$, matrix $\boldsymbol{B} \in \mathbb{R}^{m\times N}$ satisfying $\boldsymbol{B}\boldsymbol{B}^\top = c\boldsymbol{I}$ for $c > 0$, parameter $\eta > 0$, K-smooth function ϕ, real sequences $\{\alpha_n\}_{j=0}^{\infty}$, $\{\tau_n\}_{n=0}^{\infty}$, number of iterations $t > 0$, and $\boldsymbol{z}_0 \in \mathbb{R}^N$.

Output: The vector $\boldsymbol{x}_{t-1}$, which estimates a minimizer of (4.1.19), or equivalently (4.1.7) if $\boldsymbol{A}$ and $\boldsymbol{y}$ satisfy (4.1.8).

for $n = 0, 1, \ldots, t-1$ **do**
  Compute $\boldsymbol{x}_n$:
  $\boldsymbol{q} \leftarrow \boldsymbol{z}_n - K^{-1}\nabla\phi(\boldsymbol{z}_n)$
  $\lambda \leftarrow \max\left\{0,\ \frac{1}{2}\left(1 - \frac{\sqrt{2\eta^2 - \|\boldsymbol{y}_1-\boldsymbol{y}_2\|_{\ell^2}^2}}{\|\boldsymbol{y}_1+\boldsymbol{y}_2-2\boldsymbol{B}\boldsymbol{q}\|_{\ell^2}}\right)\right\}$
  $\boldsymbol{x}_n \leftarrow \frac{\lambda}{c}\boldsymbol{B}^\top(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}) + \boldsymbol{q}$.
  Compute $\boldsymbol{v}_n$:
  $\boldsymbol{q} \leftarrow \boldsymbol{z}_0 - K^{-1}\sum_{i=0}^{n}\alpha_i\nabla\phi(\boldsymbol{z}_i)$
  $\lambda \leftarrow \max\left\{0,\ \frac{1}{2}\left(1 - \frac{\sqrt{2\eta^2 - \|\boldsymbol{y}_1-\boldsymbol{y}_2\|_{\ell^2}^2}}{\|\boldsymbol{y}_1+\boldsymbol{y}_2-2\boldsymbol{B}\boldsymbol{q}\|_{\ell^2}}\right)\right\}$
  $\boldsymbol{v}_n \leftarrow \frac{\lambda}{c}\boldsymbol{B}^\top(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}) + \boldsymbol{q}$.
  Compute $\boldsymbol{z}_{n+1} \leftarrow \tau_n\boldsymbol{v}_n + (1-\tau_n)\boldsymbol{x}_n$.
end

(4.1.11), this means that the dual feasibility constraint is active, i.e. $\lambda = 0$, and in turn, $\rho = 0$. ◊

Finally, using $\boldsymbol{\xi} = \boldsymbol{z} - \boldsymbol{w}$, $\boldsymbol{b} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}$, and $\lambda = \frac{\rho}{c(1-2\rho)}$ one obtains the vector $\boldsymbol{z}$ for (4.1.10), given by

$$\boldsymbol{z} = \frac{\rho}{c}\boldsymbol{B}^\top(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{w}) + \boldsymbol{w}.$$

Therefore we have an exact formula for $\boldsymbol{x}_n$ if we set $\boldsymbol{w} = \boldsymbol{z}_n - K^{-1}\nabla\phi(\boldsymbol{z}_n)$, and otherwise for $\boldsymbol{v}_n$ if we set $\boldsymbol{w} = \boldsymbol{z}_0 - K^{-1}\sum_{j=0}^{n}\alpha_j\nabla\phi(\boldsymbol{z}_j)$. This gives us Algorithm 4 to solve the optimization problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^N}\phi(\boldsymbol{x}) \text{ subject to } \sqrt{\|\boldsymbol{y}_1 - \boldsymbol{B}\boldsymbol{x}\|_{\ell^2}^2 + \|\boldsymbol{y}_2 - \boldsymbol{B}\boldsymbol{x}\|_{\ell^2}^2} \le \eta, \tag{4.1.19}$$

which is equivalent to (4.1.7) provided $\boldsymbol{A}$ satisfies the stacked form (4.1.8). Note that to maintain consistent notation with Algorithm 2 we replace $\boldsymbol{w}$ and ρ from our derivation with $\boldsymbol{q}$ and λ, respectively, when defining Algorithm 4.

The objective error bound for the iterates in Algorithm 4 can be obtained from [73, Thm. 2]. We state a modified version of this theorem as a proposition, which will be used to prove Lemma 4.1.1.

Proposition 4.1.4. *Let $\boldsymbol{x}_n$ be the result of the nth iteration of Algorithm 4 with initial vector $\boldsymbol{z}_0$ a feasible point of (4.1.19), ϕ also convex, and parameters $\alpha_i = \frac{i+1}{2}$ and $\tau_i = \frac{2}{i+3}$.*

Then

$$\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{x}) \leq \frac{2K}{(n+1)^2}\|\boldsymbol{x} - \boldsymbol{z}_0\|_{\ell^2}^2, \qquad \forall \boldsymbol{x} : \sqrt{\|\boldsymbol{y}_1 - \boldsymbol{B}\boldsymbol{x}\|_{\ell^2}^2 + \|\boldsymbol{y}_2 - \boldsymbol{B}\boldsymbol{x}\|_{\ell^2}^2} \leq \eta.$$

Proof. Observe that Algorithm 4 is an instance of Nesterov's method (Algorithm 3) with, besides the already mentioned assumptions, prox-function $p_p(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}_0\|_{\ell^2}^2$ with strong convexity constant $\sigma_p = 1$. After slightly modifying the proof of [73, Thm. 2] (see the footnote in [75, Pg. 10]), we get the result. □

4.1.4 NESTA derivation: complex-valued data

Algorithm 2 presented in Section 4.1.1 is, in essence, an extension of Algorithm 4 to complex-valued data and with a specific choice of objective function. In this section we verify that this extension is the right one, by deriving Algorithm 2 and the smoothed QCBP problem (4.1.1). Recall the goal is to approximate a solution of the QCBP problem

$$\min_{\boldsymbol{z} \in \mathbb{C}^N} \|\boldsymbol{W}^* \boldsymbol{z}\|_{\ell^1} \text{ subject to } \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}\|_{\ell^2} \leq \eta, \tag{4.1.20}$$

with $\boldsymbol{y} \in \mathbb{C}^m$, $\boldsymbol{A} \in \mathbb{C}^{m \times N}$, $\boldsymbol{W} \in \mathbb{C}^{N \times M}$ and $\eta > 0$. We also assume m is even and that $\boldsymbol{A}$ takes on the stacking form specified in (4.1.5). To derive Algorithm 2, the key idea is to carefully represent (4.1.20) as a problem over real numbers for which we can apply Algorithm 4, i.e. Nesterov's method. This is done by using the canonical isomorphism between $\mathbb{C}^n$ and $\mathbb{R}^{2n}$. Proceeding, for $\boldsymbol{w} = (w_i)_{i=1}^{2M} \in \mathbb{R}^{2M}$, we write

$$\|\boldsymbol{w}\|_{\ell^1,\mathbb{R}^2} := \sum_{i=1}^{M} \sqrt{w_i^2 + w_{M+i}^2},$$

which defines a norm on $\mathbb{R}^{2M}$. This follows from observing that $\|\cdot\|_{\ell^1,\mathbb{R}^2}$ is equivalent to the ℓ^1-norm for $\mathbb{C}^M$, in the sense that if $\boldsymbol{w} = (\boldsymbol{a}, \boldsymbol{b})$ where $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^M$, then $\|\boldsymbol{w}\|_{\ell^1,\mathbb{R}^2} = \|\boldsymbol{a} + \mathrm{i}\boldsymbol{b}\|_{\ell^1}$. Then (4.1.20) is equivalent to the real problem

$$\begin{aligned} &\min_{\boldsymbol{u},\boldsymbol{v} \in \mathbb{R}^N} \left\| \begin{bmatrix} \mathrm{Re}\,(\boldsymbol{W})^\top & \mathrm{Im}\,(\boldsymbol{W})^\top \\ -\mathrm{Im}\,(\boldsymbol{W})^\top & \mathrm{Re}\,(\boldsymbol{W})^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix} \right\|_{\ell^1,\mathbb{R}^2} \\ &\text{subject to } \left\| \begin{bmatrix} \mathrm{Re}\,(\boldsymbol{A}) & -\mathrm{Im}\,(\boldsymbol{A}) \\ \mathrm{Im}\,(\boldsymbol{A}) & \mathrm{Re}\,(\boldsymbol{A}) \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix} - \begin{bmatrix} \mathrm{Re}\,(\boldsymbol{y}) \\ \mathrm{Im}\,(\boldsymbol{y}) \end{bmatrix} \right\|_{\ell^2} \leq \eta. \end{aligned} \tag{4.1.21}$$

Equivalence here is in the sense that $\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}} \in \mathbb{R}^N$ solve the real problem (4.1.21) if and only if $\hat{\boldsymbol{z}} = \hat{\boldsymbol{u}} + \mathrm{i}\hat{\boldsymbol{v}}$ solves (4.1.20).

To simplify notation, for all $n_1, n_2 \in \mathbb{N}$ we denote the *real equivalent* of complex vectors $\boldsymbol{z} \in \mathbb{C}^{n_1}$ and matrices $\boldsymbol{Z} \in \mathbb{C}^{n_1 \times n_2}$ by

$$\boldsymbol{z}' = \begin{bmatrix} \mathrm{Re}\,(\boldsymbol{z}) \\ \mathrm{Im}\,(\boldsymbol{z}) \end{bmatrix} \in \mathbb{R}^{2n_1}, \qquad \boldsymbol{Z}' = \begin{bmatrix} \mathrm{Re}\,(\boldsymbol{Z}) & -\mathrm{Im}\,(\boldsymbol{Z}) \\ \mathrm{Im}\,(\boldsymbol{Z}) & \mathrm{Re}\,(\boldsymbol{Z}) \end{bmatrix} \in \mathbb{R}^{2n_1 \times 2n_2}.$$

In this way, operations like matrix-vector or matrix-matrix multiplication, adjoint, and ℓ^2-norm over complex numbers, have analogous operations over real numbers. These are stated in the following proposition with proof omitted.

Proposition 4.1.5. *Let $n_1, n_2, n_3 \in \mathbb{N}$. Then for all $\boldsymbol{P} \in \mathbb{C}^{n_1 \times n_2}$, $\boldsymbol{Q}, \boldsymbol{R} \in \mathbb{C}^{n_2 \times n_3}$, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^{n_2}$, the following properties hold*

1. *(Multiplication) $(\boldsymbol{P}\boldsymbol{x})' = \boldsymbol{P}'\boldsymbol{x}'$ and $(\boldsymbol{P}\boldsymbol{Q})' = \boldsymbol{P}'\boldsymbol{Q}'$*
2. *(Matrix adjoint) $(\boldsymbol{P}^*)' = (\boldsymbol{P}')^\top$*
3. *(ℓ^2-norm) $\|\boldsymbol{x}\|_{\ell^2} = \|\boldsymbol{x}'\|_{\ell^2}$ and $\|\boldsymbol{P}\|_{\ell^2} = \|\boldsymbol{P}'\|_{\ell^2}$*
4. *(Addition) $(\boldsymbol{x} + \boldsymbol{y})' = \boldsymbol{x}' + \boldsymbol{y}'$ and $(\boldsymbol{Q} + \boldsymbol{R})' = \boldsymbol{Q}' + \boldsymbol{R}'$*

Now the real problem (4.1.21) can be written compactly with the new notation as

$$\min_{\boldsymbol{z} \in \mathbb{R}^{2N}} \|(\boldsymbol{W}')^\top \boldsymbol{z}\|_{\ell^1, \mathbb{R}^2} \text{ subject to } \|\boldsymbol{A}'\boldsymbol{z} - \boldsymbol{y}'\|_{\ell^2} \leq \eta. \tag{4.1.22}$$

Before we can apply Algorithm 4 to the problem above, we need to address two things. First is to show that we can modify $\boldsymbol{A}'$ and $\boldsymbol{y}'$ to satisfy the stacked form (4.1.8) while preserving the constraint set. Second, since $\|\cdot\|_{\ell^1, \mathbb{R}^2}$ is not differentiable in all of $\mathbb{R}^{2M}$, we use a suitable smooth approximation instead.

For the first part, note that if $\boldsymbol{A} \in \mathbb{C}^{m \times N}$ satisfies (4.1.5), then $\boldsymbol{A}' \in \mathbb{R}^{2m \times 2N}$ generally does not satisfy (4.1.8). This can be corrected by using the permutation matrix

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \in \mathbb{R}^{2m \times 2m},$$

where $\boldsymbol{0}$ and $\boldsymbol{I}$ here are the $\frac{m}{2} \times \frac{m}{2}$ zero and identity matrix, respectively, so that

$$\boldsymbol{P}\boldsymbol{A}' = \begin{bmatrix} \boldsymbol{B}' \\ \boldsymbol{B}' \end{bmatrix} \in \mathbb{R}^{2m \times N}, \qquad \boldsymbol{P}\boldsymbol{y}' = \begin{bmatrix} \boldsymbol{y}_1' \\ \boldsymbol{y}_2' \end{bmatrix} \in \mathbb{R}^{2m}.$$

Since the ℓ^2-norm is invariant under unitary transformations, the constraint in (4.1.22) is equivalent to $\|\boldsymbol{P}\boldsymbol{A}'\boldsymbol{z} - \boldsymbol{P}\boldsymbol{y}'\|_{\ell^2} \leq \eta$. To address the second part, we use the Moreau envelope (also known as Moreau-Yosida regularization) to obtain a smooth approximation of $\|\cdot\|_{\ell^1, \mathbb{R}^2}$.

Definition 4.1.6 (Moreau envelope [11, Defn. 6.52])**.** Let $n \in \mathbb{N}$. Given a proper closed convex function $f : \mathbb{R}^n \to \mathbb{R}$ and $\mu > 0$, the *Moreau envelope* of f is the function $\mathcal{M}_f^\mu : \mathbb{R}^n \to \mathbb{R}$ defined by

$$\mathcal{M}_f^\mu(\boldsymbol{w}) = \min_{\boldsymbol{v} \in \mathbb{R}^n} \left\{ f(\boldsymbol{v}) + \frac{1}{2\mu} \|\boldsymbol{w} - \boldsymbol{v}\|_{\ell^2}^2 \right\}, \qquad \boldsymbol{w} \in \mathbb{R}^n.$$

$\Diamond$

The parameter μ here is, as before, called the *smoothing parameter*. The Moreau envelope is well-defined since the corresponding minimization problem always has a unique solution. In addition, if f is proper and convex, then $\mathcal{M}_f^\mu$ is real-valued and convex [11, Thm. 6.55]. From the perspective of optimization theory, an extensive view of the Moreau envelope and its properties are found in [11, Sec. 6.7]. From there we use a handful of results relevant to our setup. Combining [11, Ex. 6.54] and [11, Thm. 6.58], the Moreau envelope of $f_1 = \|\cdot\|_{\ell^1,\mathbb{R}^2}$ is precisely

$$\mathcal{M}_{f_1}^\mu(\boldsymbol{w}) = \sum_{i=1}^{M} \mathcal{M}_{\|\cdot\|_{\ell^2}}^\mu(w_i, w_{M+i}), \qquad \boldsymbol{w} = (w_i)_{i=1}^{2M} \in \mathbb{R}^{2M},$$

where for any fixed $k \in \mathbb{N}$,

$$\mathcal{M}_{\|\cdot\|_{\ell^2}}^\mu(\boldsymbol{\xi}) = \begin{cases} \frac{1}{2\mu}\|\boldsymbol{\xi}\|_{\ell^2}^2, & \|\boldsymbol{\xi}\|_{\ell^2} \le \mu \\ \|\boldsymbol{\xi}\|_{\ell^2} - \frac{\mu}{2}, & \|\boldsymbol{\xi}\|_{\ell^2} > \mu \end{cases}, \qquad \boldsymbol{\xi} \in \mathbb{R}^k,$$

is known as the *Huber function*. Its gradient is

$$\nabla \mathcal{M}_{\|\cdot\|_{\ell^2}}^\mu(\boldsymbol{\xi}) = \begin{cases} \frac{\boldsymbol{\xi}}{\mu}, & \|\boldsymbol{\xi}\|_{\ell^2} \le \mu \\ \frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|_{\ell^2}}, & \|\boldsymbol{\xi}\|_{\ell^2} > \mu \end{cases}, \qquad \boldsymbol{\xi} \in \mathbb{R}^k,$$

and therefore the gradient of $\mathcal{M}_{f_1}^\mu$ is given elementwise by

$$\left(\nabla \mathcal{M}_{f_1}^\mu(\boldsymbol{w})\right)_i = \begin{cases} \frac{w_j}{\mu}, & j = i \text{ or } M+i,\ \sqrt{w_i^2 + w_{M+i}^2} \le \mu \\ \frac{w_j}{\sqrt{w_i^2 + w_{M+i}^2}}, & j = i \text{ or } M+i,\ \sqrt{w_i^2 + w_{M+i}^2} > \mu \end{cases}$$

for $i = 1, \ldots, 2M$. We can express $\mathcal{M}_{f_1}^\mu$ as a function over $\mathbb{C}^M$ by writing

$$\|\boldsymbol{z}\|_{\ell^1,\mu} = \sum_{i=1}^{M} H_\mu(z_i), \qquad \boldsymbol{z} \in (z_i)_{i=1}^{M} \in \mathbb{C}^M,$$

where

$$H_\mu(z) = \begin{cases} \frac{1}{2\mu}|z|^2 & |z| \le \mu \\ |z| - \frac{\mu}{2} & |z| > \mu \end{cases}, \qquad z \in \mathbb{C}.$$

This is precisely what we wrote in (4.1.2) and (4.1.3). In turn, representing $\nabla \mathcal{M}_{f_1}^{\mu}$ as a function over $\mathbb{C}^M$, which we denote by $\mathcal{T}_\mu$, gives

$$(\mathcal{T}_\mu(\boldsymbol{z}))_i = \begin{cases} \frac{z_i}{\mu}, & |z_i| \leq \mu \\ \frac{z_i}{|z_i|}, & |z_i| > \mu \end{cases}, \qquad i = 1, \ldots, M,$$

which is exactly (4.1.4). Therefore, the smoothed version of (4.1.22) is precisely

$$\min_{\boldsymbol{z} \in \mathbb{R}^{2N}} \mathcal{M}_{f_1}^{\mu}((\boldsymbol{W}')^\top \boldsymbol{z}) \text{ subject to } \|\boldsymbol{A}'\boldsymbol{z} - \boldsymbol{y}'\|_{\ell^2} \leq \eta, \tag{4.1.23}$$

Now, since the Moreau envelope is $\frac{1}{\mu}$-smooth [11, Thm. 6.60], the objective function of (4.1.23) is $\|\boldsymbol{W}'\|_{\ell^2}^2/\mu$-smooth. Moreover, f_1 is convex and defined everywhere, thus $\mathcal{M}_{f_1}^{\mu}$ is real-valued and convex [11, Thm. 6.55]. Now we can apply Algorithm 4 to (4.1.23) with data $\boldsymbol{B} := \boldsymbol{B}'$, $\boldsymbol{y}_1 := \boldsymbol{y}_1'$, $\boldsymbol{y}_2 := \boldsymbol{y}_2'$, and $\phi := \mathcal{M}_{f_1}^{\mu}((\boldsymbol{W}')^\top \cdot)$. Using Proposition 4.1.5, we can express (4.1.23) and the update steps in Algorithm 4 with their complex equivalents. This yields (4.1.1) and Algorithm 2, respectively, which is what we wanted to show.

Remark 4.1.7 (Nesterov smoothing and the Moreau envelope)**.** Moreau-Yosida regularization is a standard technique for smoothing in optimization, with the Moreau envelope directly related to proximal maps. A smoothed function obtained by Nesterov smoothing [73], under some additional assumptions, can be viewed as the Fenchel dual of the Moreau envelope [12, Sec. 4.3]. We briefly show that for QCBP and right choice of dual prox-function, the Moreau envelope and Nesterov smoothing coincide.

Simplifying some of the assumptions in Nesterov's paper [73], suppose $\boldsymbol{W} \in \mathbb{R}^{N \times M}$ and $Q_p \subseteq \mathbb{R}^N$, $Q_d \subseteq \mathbb{R}^M$ are closed, convex and bounded sets. It is assumed that the objective function f, possibly nonsmooth, can be expressed as

$$f(\boldsymbol{x}) = \max_{\boldsymbol{u} \in Q_d} \left\{ \langle \boldsymbol{W}^\top \boldsymbol{x}, \boldsymbol{u} \rangle \right\}, \qquad \forall \boldsymbol{x} \in Q_p. \tag{4.1.24}$$

The idea now is to select a dual prox-function $p_d : Q_d \to \mathbb{R}$, assumed to be strongly convex on Q_d with some strong convexity constant $\sigma_d > 0$. Then for $\mu > 0$, the smooth approximation of f is defined as

$$f_\mu(\boldsymbol{x}) = \max_{\boldsymbol{u} \in Q_d} \left\{ \langle \boldsymbol{W}^\top \boldsymbol{x}, \boldsymbol{u} \rangle - \mu p_d(\boldsymbol{u}) \right\}, \qquad \boldsymbol{x} \in Q_p.$$

In the case of the QCBP objective, it is a standard result that the ℓ^1-norm is the dual norm of the ℓ^∞-norm, hence

$$f(\boldsymbol{x}) = \|\boldsymbol{W}^\top \boldsymbol{x}\|_{\ell^1} = \max_{\boldsymbol{u} \in Q_d} \left\{ \langle \boldsymbol{W}^\top \boldsymbol{x}, \boldsymbol{u} \rangle \right\}, \qquad Q_d = \{\boldsymbol{u} \in \mathbb{R}^M : \|\boldsymbol{u}\|_{\ell^\infty} \leq 1\},$$

and so f has the desired representation (4.1.24). Using tools from convex optimization, observe that f is equal to the support function on Q_d [11, Sec. 2.4]. Thus the convex conjugate of f is given by $f^* = \delta_{Q_d}$ [11, Ex. 4.9], the indicator function on Q_d. Therefore the Moreau envelope of f^* is given by

$$\mathcal{M}_{f^*}^{\mu}(\boldsymbol{W}^\top \boldsymbol{x}) = \min_{\boldsymbol{u} \in Q_d} \frac{1}{2\mu} \|\boldsymbol{W}^\top \boldsymbol{x} - \boldsymbol{u}\|_{\ell^2}^2$$

The Moreau decomposition theorem [11, Thm. 6.67] says

$$\mathcal{M}_{f}^{\mu}(\boldsymbol{W}^\top \boldsymbol{x}) + \mathcal{M}_{f^*}^{1/\mu}(\boldsymbol{W}^\top \boldsymbol{x}/\mu) = \frac{1}{2\mu} \|\boldsymbol{W}^\top \boldsymbol{x}\|_{\ell^2}^2.$$

Thus

$$\mathcal{M}_{f}^{\mu}(\boldsymbol{W}^\top \boldsymbol{x}) = \frac{1}{2\mu} \|\boldsymbol{W}^\top \boldsymbol{x}\|_{\ell^2}^2 - \min_{\boldsymbol{u} \in Q_d} \frac{\mu}{2} \|\boldsymbol{W}^\top \boldsymbol{x}/\mu - \boldsymbol{u}\|_{\ell^2}^2 = \max_{\boldsymbol{u} \in Q_d} \langle \boldsymbol{W}^\top \boldsymbol{x}, \boldsymbol{u} \rangle - \frac{\mu}{2} \|\boldsymbol{u}\|_{\ell^2}^2.$$

This shows that if we take $p_d(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{u}\|_{\ell^2}^2$, then $f_\mu = \mathcal{M}_f^\mu(\boldsymbol{W}^\top \cdot)$. ◊

4.1.5 Proof of error bounds for smoothing

Proof of Lemma 4.1.1. For $f_1 = \|\cdot\|_{\ell^1,\mathbb{R}^2}$, first apply Proposition 4.1.4 with objective function $\phi = \mathcal{M}_{f_1}^{\mu}((\boldsymbol{W}')^\top \cdot)$ so that $K = \|\boldsymbol{W}\|_{\ell^2}^2/\mu$. Note that the Moreau envelope is always a $\frac{1}{\mu}$-smooth function [11, Thm. 6.60]. Then the error bound expressed in its complex equivalent is

$$\begin{aligned} &\|\boldsymbol{W}^* \boldsymbol{x}_n\|_{\ell^1,\mu} - \|\boldsymbol{W}^* \boldsymbol{x}\|_{\ell^1,\mu} \leq \frac{2\|\boldsymbol{W}\|_{\ell^2}^2}{\mu(n+1)^2} \|\boldsymbol{x} - \boldsymbol{z}_0\|_{\ell^2}^2, \\ &\forall \boldsymbol{x} : \sqrt{\|\boldsymbol{y}_1 - \boldsymbol{B}\boldsymbol{x}\|_{\ell^2}^2 + \|\boldsymbol{y}_2 - \boldsymbol{B}\boldsymbol{x}\|_{\ell^2}^2} \leq \eta. \end{aligned} \tag{4.1.25}$$

Now, the Huber function satisfies

$$H_\mu(z) \leq H_\mu(z) \leq H_\mu(z) + \frac{\mu}{2}, \qquad \forall z \in \mathbb{C},$$

and thus for $\|\boldsymbol{z}\|_{\ell^1,\mu} = \sum_{i=1}^M H_\mu(z_i)$ we get

$$\|\boldsymbol{z}\|_{\ell^1,\mu} \leq \|\boldsymbol{z}\|_{\ell^1} \leq \|\boldsymbol{z}\|_{\ell^1,\mu} + \frac{M\mu}{2}, \qquad \forall \boldsymbol{z} \in \mathbb{C}^N. \tag{4.1.26}$$

Combining this observation and the previous one gives

$$\begin{aligned} \|\boldsymbol{W}^* \boldsymbol{x}_n\|_{\ell^1} &\leq \|\boldsymbol{W}^* \boldsymbol{x}_n\|_{\ell^1,\mu} + \frac{M\mu}{2} \\ &\leq \|\boldsymbol{W}^* \boldsymbol{x}\|_{\ell^1,\mu} + \frac{2\|\boldsymbol{W}\|_{\ell^2}^2}{\mu(n+1)^2} \|\boldsymbol{x} - \boldsymbol{z}_0\|_{\ell^2}^2 + \frac{M\mu}{2} \end{aligned}$$

$$\leq \|\boldsymbol{W}^*\boldsymbol{x}\|_{\ell^1} + \frac{2\|\boldsymbol{W}\|_{\ell^2}^2}{\mu(n+1)^2}\|\boldsymbol{x}-\boldsymbol{z}_0\|_{\ell^2}^2 + \frac{M\mu}{2},$$

which is what we wanted to show. □

Remark 4.1.8. Here we consider an alternative proof of Lemma 4.1.1. We wish to highlight that the Huber function inequalities arise from the Moreau envelope of globally Lipschitz functions, and so these techniques can be used for nonsmooth problems that are more general than QCBP. First we consider [11, Defn. 10.43], which is stated as follows.

Let $a, b > 0$. A convex function $f : \mathbb{R}^N \to \mathbb{R}$ is called (a, b)-*smoothable* if for any $\mu > 0$ there exists a convex differentiable function $f_\mu : \mathbb{R}^N \to \mathbb{R}$ such that

1. $f_\mu(\boldsymbol{x}) \leq f(\boldsymbol{x}) \leq f_\mu(\boldsymbol{x}) + b\mu$ for all $x \in \mathbb{R}^N$
2. f_μ is $\frac{a}{\mu}$-smooth

The function f_μ is referred to as a $\frac{1}{\mu}$-*smooth approximation* of f with parameters (a, b), and μ is referred to as the *smoothing parameter*.

The idea now is to use the fact that Lipschitz continuous functions have a $\frac{1}{\mu}$-smooth approximation by their Moreau envelope with parameter μ. This yields a proof of Lemma 4.1.1 summarized below.

Write $f_1 = \|\cdot\|_{\ell^1,\mathbb{R}^2}$. For $n \in \mathbb{N}$, observe that for any $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{C}^n$, we have

$$|f_1(\boldsymbol{a}') - f_1(\boldsymbol{b}')| = |\|\boldsymbol{a}\|_{\ell^1} - \|\boldsymbol{b}\|_{\ell^1}| \leq \|\boldsymbol{a}-\boldsymbol{b}\|_{\ell^1} \leq \sqrt{n}\|\boldsymbol{a}-\boldsymbol{b}\|_{\ell^2} = \sqrt{n}\|\boldsymbol{a}'-\boldsymbol{b}'\|_{\ell^2}.$$

Thus f_1 is Lipschitz with constant $\sqrt{n}$. Combining [11, Thm. 10.51] and [11, Thm. 10.46(b)], for all $\mu > 0$ we have that

$$(f_1)_\mu = \mathcal{M}_{f_1}^{\mu}((\boldsymbol{W}')^\top \cdot)$$

is a $\frac{1}{\mu}$-smooth approximation of f_1 with parameters $(\|\boldsymbol{W}'\|_{\ell^2}^2, M/2)$. Now to get the result, apply Proposition 4.1.4 with $\phi = (f_1)_\mu$ and use the definition of (a, b)-smoothable to conclude

$$(f_1)_\mu(\boldsymbol{x}_n) - (f_1)_\mu(\boldsymbol{x}) \leq \frac{2\|\boldsymbol{W}'\|_{\ell^2}^2}{\mu(n+1)^2}\|\boldsymbol{x}-\boldsymbol{z}_0\|_{\ell^2}^2 + \frac{M\mu}{2},$$
$$\forall \boldsymbol{x} : \sqrt{\|\boldsymbol{y}_1' - \boldsymbol{B}'\boldsymbol{x}\|_{\ell^2}^2 + \|\boldsymbol{y}_2' - \boldsymbol{B}'\boldsymbol{x}\|_{\ell^2}^2} \leq \eta.$$

Finally, express the error bound and constraint as their complex equivalents. ◊

4.2 Recovery guarantees for TV-Fourier inverse problems

4.2.1 Image and gradient recovery via NESTA

Here we state and prove error bounds for reconstructing images and their gradient from Fourier measurements under gradient sparsity. We do this in relation to the reconstruction procedure of using stacked NESTA (Algorithm 2) for TV minimization (2.3.2).

Theorem 4.2.1 (Accuracy and stability of NESTA reconstruction, $d = 1$)**.** *Let $d = 1$, $0 < \epsilon < 1$, $2 \leq s, m \leq N$, and*

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{B} \end{bmatrix}, \qquad \boldsymbol{B} = \frac{1}{\sqrt{m}} \boldsymbol{P}_\Omega \boldsymbol{F} \in \mathbb{C}^{|\Omega| \times N},$$

be a subsampled Fourier matrix where $\Omega = \Omega_1 \cup \Omega_2$ with $\Omega_1 \sim \mathrm{Ber}([\![N]\!], m/2)$ and $\Omega_2 \sim \mathrm{Ber}([\![N]\!], m/2, \boldsymbol{p})$. Now if

$$m \gtrsim \Gamma(\boldsymbol{p}) \cdot s \cdot \left(\log(\Gamma(\boldsymbol{p}) N s) \cdot \log^2(s) \cdot \log(N) + \log(2\epsilon^{-1}) \right) \tag{4.2.1}$$

then the following holds with probability at least $1 - \epsilon$. For all $\boldsymbol{x} \in \mathbb{C}^N$ and $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} \in \mathbb{C}^{2|\Omega|}$ with $\|\boldsymbol{e}\|_{\ell^2} \leq \eta$ for some $\eta > 0$, if $\boldsymbol{x}_n \in \mathbb{C}^N$ is the nth iterate of Algorithm 2, with input feasible point $\boldsymbol{z}_0$, sequences $\{\alpha_j\}$, $\{\tau_j\}$ from Lemma 4.1.1, matrix $\boldsymbol{B}$ and vectors $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2)$, then

$$\|\boldsymbol{V}\boldsymbol{x}_n - \boldsymbol{V}\boldsymbol{x}\|_{\ell^2} \lesssim \frac{\sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1}}{\sqrt{s}} + \eta + \frac{N\mu}{2\sqrt{s}} + \frac{1}{\mu(n+1)^2\sqrt{s}} \|\boldsymbol{x} - \boldsymbol{z}_0\|_{\ell^2}^2, \tag{4.2.2}$$

$$\frac{\|\boldsymbol{x}_n - \boldsymbol{x}\|_{\ell^2}}{\sqrt{N}} \lesssim \frac{\sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1}}{\sqrt{s}} + \left(\sqrt{\Gamma(\boldsymbol{p})} + 1 \right) \eta + \frac{N\mu}{2\sqrt{s}} + \frac{1}{\mu(n+1)^2\sqrt{s}} \|\boldsymbol{x} - \boldsymbol{z}_0\|_{\ell^2}^2. \tag{4.2.3}$$

Note that the constants in $\lesssim$ do not depend on n.

Theorem 4.2.2 (Accuracy and stability of NESTA reconstruction, $d \geq 2$)**.** *Let $d \geq 2$, $0 < \epsilon < 1$, $2 \leq s, m \leq N^d$, and*

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{B} \end{bmatrix}, \qquad \boldsymbol{B} = \frac{1}{\sqrt{m}} \boldsymbol{P}_\Omega \boldsymbol{F} \in \mathbb{C}^{|\Omega| \times N^d},$$

be a subsampled Fourier matrix where $\Omega = \Omega_1 \cup \Omega_2$ with $\Omega_1 \sim \mathrm{Ber}([\![N^d]\!], m/2)$ and $\Omega_2 \sim \mathrm{Ber}([\![N^d]\!], m/2, \boldsymbol{p})$. Now if

$$m \gtrsim_d \Gamma(\boldsymbol{p}) \cdot s \cdot \log^2(N) \cdot \left(\log(\Gamma(\boldsymbol{p}) N \log^2(N) s) \cdot \log^2(s \log^2(N)) \cdot \log(N) + \log(2\epsilon^{-1}) \right) \tag{4.2.4}$$

then the following holds with probability at least $1 - \epsilon$. For all $\boldsymbol{x} \in \mathbb{C}^{N^d}$ and $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} \in \mathbb{C}^{2|\Omega|}$ with $\|\boldsymbol{e}\|_{\ell^2} \leq \eta$ for some $\eta > 0$, if $\boldsymbol{x}_n \in \mathbb{C}^{N^d}$ is the nth iterate of Algorithm 2,

with input feasible point $\boldsymbol{z}_0$*, sequences* $\{\alpha_j\}$*,* $\{\tau_j\}$ *from Lemma 4.1.1, matrix* $\boldsymbol{B}$ *and vectors* $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2)$*, then*

$$\|\boldsymbol{V}\boldsymbol{x}_n - \boldsymbol{V}\boldsymbol{x}\|_{\ell^2} \lesssim \frac{\sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1}}{\sqrt{s}} + d\eta + \frac{dN^d\mu}{2\sqrt{s}} + \frac{d}{\mu(n+1)^2\sqrt{s}}\|\boldsymbol{x} - \boldsymbol{z}_0\|_{\ell^2}^2, \tag{4.2.5}$$

$$\|\boldsymbol{x}_n - \boldsymbol{x}\|_{\ell^2} \lesssim_d \frac{\sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1}}{\sqrt{s}} + \left(\sqrt{\Gamma(\boldsymbol{p})} + d\right)\eta + \frac{dN^d\mu}{2\sqrt{s}} + \frac{d}{\mu(n+1)^2\sqrt{s}}\|\boldsymbol{x} - \boldsymbol{z}_0\|_{\ell^2}^2. \tag{4.2.6}$$

Note that the constants in $\lesssim$ *and* $\lesssim_d$ *do not depend on* n*.*

The proof structure and arguments for both Theorems 4.2.1 and 4.2.2 directly combine and adapt the proofs presented in [4, Sec. 7] and [5, Chap. 8.3]. For completeness and considering modifications are needed, we present the proofs in full detail.

Proof of Theorems 4.2.1 and 4.2.2. The frequencies of indices $\Omega = \Omega_1 \cup \Omega_2$ are sampled exactly twice by $\boldsymbol{A}$, so writing $\Omega_3 = \Omega_1 \triangle \Omega_2$, we can express $\boldsymbol{A}$ as

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \boldsymbol{A}_3 \end{bmatrix} \quad \boldsymbol{A}_i = \frac{1}{\sqrt{m}} P_{\Omega_i}\boldsymbol{F}, \quad i = 1, 2, 3.$$

Note that it is valid to represent $\boldsymbol{A}$ in this way since the ℓ^2-norm constraint of TV minimization (2.3.2) is invariant under unitary transformations (and thus, row permutations common to both $\boldsymbol{A}$ and $\boldsymbol{y}$). This reasoning allows us to interchange between other row permutations of both $\boldsymbol{A}$ and $\boldsymbol{y}$, including those that lead to the stacking form (4.1.5). This allows one to apply Algorithm 2 to solve TV minimization.

Gradient recovery. Here Ω_1 corresponds to Bernoulli uniform sampling, so applying Lemma 3.2.1 with $\delta = 1/3$, $\boldsymbol{A}_1$ satisfies the RIP of order $2s$ with constant $\delta_{2s} \leq 1/3$ with probability at least $1 - \epsilon/2$, provided

$$m \gtrsim_d s \cdot \left(\log(Ns) \cdot \log^2(s) \cdot \log(N) + \log(2\epsilon^{-1})\right).$$

This condition holds in both cases of $d = 1$ and $d \geq 2$, given our assumed measurement conditions (4.2.1) and (4.2.4). Note the choice of $\delta = 1/3$ is arbitrary, as it is enough for $\delta < \sqrt{2} - 1$. By Lemma 2.1.5, $\boldsymbol{A}_1$ has the rNSP of order s with constants $0 < \rho < 1$, $\gamma > 0$.

Define $\boldsymbol{A}_1^{(d)}$ to be $\boldsymbol{B}$ in [4, Lem. SM1.4] with diagonal blocks $\boldsymbol{A} := \boldsymbol{A}_1$. Thus, $\boldsymbol{A}_1^{(d)}$ has the rNSP of order s with constants $\rho' = \rho$ and $\gamma' = \sqrt{d}\gamma$. Then the rNSP ℓ^2-norm bound of Lemma 2.1.3 holds with matrix $\boldsymbol{A}_1^{(d)}$, giving

$$\|\boldsymbol{V}\boldsymbol{x}_n - \boldsymbol{V}\boldsymbol{x}\|_{\ell^2} \leq C_1\left(\frac{2\sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1} + \|\boldsymbol{V}\boldsymbol{x}_n\|_{\ell^1} - \|\boldsymbol{V}\boldsymbol{x}\|_{\ell^1}}{\sqrt{s}}\right) + C_2\sqrt{d}\|\boldsymbol{A}_1^{(d)}(\boldsymbol{V}\boldsymbol{x}_n - \boldsymbol{V}\boldsymbol{x})\|_{\ell^2}$$

$$\lesssim \frac{\sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1}}{\sqrt{s}} + \frac{\|\boldsymbol{V}\boldsymbol{x}_n\|_{\ell^1} - \|\boldsymbol{V}\boldsymbol{x}\|_{\ell^1}}{\sqrt{s}} + \sqrt{d}\|\boldsymbol{A}_1^{(d)}(\boldsymbol{V}\boldsymbol{x}_n - \boldsymbol{V}\boldsymbol{x})\|_{\ell^2}$$

where $C_1 = \frac{(3\rho+1)(\rho+1)}{2(1-\rho)}$ and $C_2 = \frac{(3\rho+5)\gamma}{2(1-\rho)}$. To bound the $\boldsymbol{A}_1^{(d)}$ term, we use the commuting property of the Fourier matrix and circulant matrices. Specifically, we use the identity $\boldsymbol{F}\boldsymbol{V}_i = \boldsymbol{D}_i\boldsymbol{F}$ [4, Lem. 7.1] where $\boldsymbol{D}_i \in \mathbb{C}^{N^d \times N^d}$ is a diagonal matrix with $\|\boldsymbol{D}_i\|_{\ell^2} \leq 2$. Thus

$$\begin{aligned}
\|\boldsymbol{A}_1^{(d)}\boldsymbol{V}(\boldsymbol{x}_n - \boldsymbol{x})\|_{\ell^2} &= \sqrt{\sum_{i=1}^{d} \|\boldsymbol{A}_1\boldsymbol{V}_i(\boldsymbol{x}_n - \boldsymbol{x})\|_{\ell^2}^2} \\
&= \sqrt{\sum_{i=1}^{d} \left\| \frac{1}{\sqrt{m}}\boldsymbol{P}_{\Omega_1}\boldsymbol{F}\boldsymbol{V}_i(\boldsymbol{x}_n - \boldsymbol{x}) \right\|_{\ell^2}^2} \\
&= 2\sqrt{d}\left\| \frac{1}{\sqrt{m}}\boldsymbol{P}_{\Omega_1}\boldsymbol{F}(\boldsymbol{x}_n - \boldsymbol{x}) \right\|_{\ell^2} \\
&= 2\sqrt{d}\|\boldsymbol{A}_1(\boldsymbol{x}_n - \boldsymbol{x})\|_{\ell^2} \\
&\leq 2\sqrt{d}\|\boldsymbol{A}(\boldsymbol{x}_n - \boldsymbol{x})\|_{\ell^2} \\
&\leq 4\sqrt{d}\eta.
\end{aligned}$$

In the second last step, we used that both $\boldsymbol{x}_n$ and $\boldsymbol{x}$ are feasible for TV minimization (2.3.2). Next, to bound the term $\|\boldsymbol{V}\boldsymbol{x}_n\|_{\ell^1} - \|\boldsymbol{V}\boldsymbol{x}\|_{\ell^1}$, we use the feasibility of $\boldsymbol{x}$ and apply Lemma 4.1.1. Note from the commuting property, one has $\|\boldsymbol{V}_i\|_{\ell^2} \leq 2$, hence

$$\|\boldsymbol{V}\|_{\ell^2} \leq \sqrt{\sum_{i=1}^{d} \|\boldsymbol{V}_i\|_{\ell^2}^2} \leq \sqrt{4d} = 2\sqrt{d}.$$

Combining everything gives

$$\|\boldsymbol{V}\boldsymbol{x}_n - \boldsymbol{V}\boldsymbol{x}\|_{\ell^2} \lesssim \frac{\sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1}}{\sqrt{s}} + 4d\eta + \frac{dN^d\mu}{2\sqrt{s}} + \frac{8d}{\mu(n+1)^2\sqrt{s}}\|\boldsymbol{x} - \boldsymbol{z}_0\|_{\ell^2}^2$$

and in turn (4.2.2) and (4.2.5).

Image recovery. Here we only consider $d \geq 2$, since the proof of the case $d = 1$ is nearly identical. The conditions of Lemma 3.2.5 are met, so we have with probability at least $1 - \epsilon/2$ that

$$\|\boldsymbol{x}_n - \boldsymbol{x}\|_{\ell^2} \lesssim_d \sqrt{\Gamma(\boldsymbol{p})}\|\boldsymbol{A}_2(\boldsymbol{x}_n - \boldsymbol{x})\|_{\ell^2} + \frac{\|\boldsymbol{x}_n - \boldsymbol{x}\|_{\mathrm{TV}}}{\sqrt{s}}. \tag{4.2.7}$$

As established earlier, with probability at least $1 - \epsilon/2$, $\boldsymbol{A}_1^{(d)}$ satisfies the rNSP of order s with constants $\rho' = \rho$ and $\gamma' = \sqrt{d}\gamma$. By the union bound, both observations simultaneously hold with probability with at least $1 - \epsilon$.

To bound the first term, we apply the triangle inequality to obtain

$$\|\boldsymbol{A}_2(\boldsymbol{x}_n - \boldsymbol{x})\|_{\ell^2} \leq \|\boldsymbol{A}(\boldsymbol{x}_n - \boldsymbol{x})\|_{\ell^2} \leq 2\eta.$$

For the second term, noting that $\|\boldsymbol{x}_n - \boldsymbol{x}\|_{\mathrm{TV}} = \|\boldsymbol{V}\boldsymbol{x}_n - \boldsymbol{V}\boldsymbol{x}\|_{\ell^1}$ and using the rNSP ℓ^1-norm bound of Lemma 2.1.3 with the matrix $\boldsymbol{A}_1^{(d)}$, one obtains

$$\begin{aligned}\|\boldsymbol{V}\boldsymbol{x}_n - \boldsymbol{V}\boldsymbol{x}\|_{\ell^1} &\leq C_3(2\sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1} + \|\boldsymbol{V}\boldsymbol{x}_n\|_{\ell^1} - \|\boldsymbol{V}\boldsymbol{x}\|_{\ell^1}) + C_4\sqrt{d}\sqrt{s}\|\boldsymbol{A}_1^{(d)}(\boldsymbol{V}\boldsymbol{x}_n - \boldsymbol{V}\boldsymbol{x})\|_{\ell^2} \\ &\lesssim \sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1} + \|\boldsymbol{V}\boldsymbol{x}_n\|_{\ell^1} - \|\boldsymbol{V}\boldsymbol{x}\|_{\ell^1} + \sqrt{d}\sqrt{s}\|\boldsymbol{A}_1^{(d)}(\boldsymbol{V}\boldsymbol{x}_n - \boldsymbol{V}\boldsymbol{x})\|_{\ell^2}.\end{aligned}$$

where $C_3 = \frac{1+\rho}{1-\rho}$ and $C_4 = \frac{2\gamma}{1-\rho}$. Now apply the same bounds to $\|\boldsymbol{A}_1^{(d)}\boldsymbol{V}(\boldsymbol{x}_n - \boldsymbol{x})\|_{\ell^2}$ and $\|\boldsymbol{V}\boldsymbol{x}_n\|_{\ell^1} - \|\boldsymbol{V}\boldsymbol{x}\|_{\ell^1}$ as in the gradient recovery case, yielding

$$\|\boldsymbol{x}_n - \boldsymbol{x}\|_{\mathrm{TV}} = \|\boldsymbol{V}\boldsymbol{x}_n - \boldsymbol{V}\boldsymbol{x}\|_{\ell^1} \lesssim \sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1} + d\sqrt{s}\eta + \frac{dN^d\mu}{2} + \frac{d}{\mu(n+1)^2}\|\boldsymbol{x} - \boldsymbol{z}_0\|_{\ell^2}^2.$$

Using this to bound (4.2.7) gives (4.2.6). □

4.3 Restart scheme to accelerate reconstruction

To motivate acceleration of reconstruction, consider the image reconstruction error bound (4.2.6) from Theorem 4.2.2. For fixed n, minimizing this quantity with respect to μ yields

$$\mu = \frac{\sqrt{2}\|\boldsymbol{x} - \boldsymbol{z}_0\|_{\ell^2}}{N^{d/2}(n+1)}.$$

Substituting this value of μ into the bound (4.2.6) gives

$$\|\boldsymbol{x}_n - \boldsymbol{x}\|_{\ell^2} \lesssim_d \frac{\sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1}}{\sqrt{s}} + \left(\sqrt{\Gamma(\boldsymbol{p})} + d\right)\eta + \frac{\sqrt{2}dN^{d/2}}{(n+1)\sqrt{s}}\|\boldsymbol{x} - \boldsymbol{z}_0\|_{\ell^2},$$

thus the reconstruction error decays with order $\mathcal{O}(n^{-1})$ down to the model class distance $\sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1}$ and noise level η. This sublinear decay in error also occurs in practice, e.g. see Section 6.2. This motivates the need to accelerate the reconstruction, which would result in NESTANets using substantially fewer layers.

A 'restart scheme' refers to an algorithmic framework that repeatedly restarts an optimization algorithm, possibly altering initial parameters. Under quite general conditions on the optimization problem [3], a restart scheme can accelerate the convergence rate of a first-order method. This is particularly useful for nonsmooth problems (e.g. TV minimization), where first-order methods only produce sublinear rates in theory and practice [19,74]. As shown in [3,30,75], many compressed sensing problems are amenable to acceleration by restarts and observe a key performance gain from sublinear to linear (i.e. exponential) decay.

Algorithm 5: Restarted stacked NESTA for QCBP.

Input : Initial point $\boldsymbol{x}_0^\star$, sequences $\{\mu_k\}_{k=1}^{t+1}$, $\{n_k\}_{k=1}^{t+1}$, and number of restarts t.
Output: The vector $\boldsymbol{x}_{t+1}^\star$, which estimates a minimizer of (2.2.1).

```
for k = 1, ..., t + 1 do
    Set x*_k as the output of stacked NESTA (Algorithm 2) with initial point x*_{k−1},
     smoothing parameter μ_k and number of iterations n_k.
end
```

To motivate restarting NESTA, one observes the recursive nature of the image error formulas in Theorems 4.2.1 and 4.2.2 in terms of the error in the initial guess. The restart procedure starts by running NESTA with initial guess $\boldsymbol{z}_0$ and smoothing parameter μ_1 for a fixed number of iterations, say n_1. Then we feed the output as an initial guess to a new instance of NESTA with smoothing parameter μ_2 and n_2 iterations. This repeated reinitialization, or restarting, is performed for finitely many steps. The restart scheme is summarized in Algorithm 5.

We can apply the same approach found in [75] to choose the smoothing parameters $\{\mu_k\}$ and inner iterations $\{n_k\}$ that guarantee convergence acceleration, by using Theorems 4.2.1 and 4.2.2. For this, we introduce and use the notation

$$\mathcal{CS}_{s,d}(\boldsymbol{z}, \boldsymbol{p}, \eta) = \frac{\sigma_s(\boldsymbol{z})_{\ell^1}}{\sqrt{s}} + \left(\sqrt{\Gamma(\boldsymbol{p})} + d\right)\eta,$$

and refer to this quantity as the *compressed sensing error*. We now state the accuracy and stability results for Fourier imaging via restarted NESTA.

Theorem 4.3.1 (Performance of restarted NESTA reconstruction, $d = 1$)**.** *Consider $d = 1$ and $\boldsymbol{A}$, Ω, $\boldsymbol{p}$, ϵ, s, m, N, η from Theorem 4.2.1 and suppose the measurement condition (4.2.1) holds. Then the following holds with probability at least $1 - \epsilon$. There is a constant $C > 0$ such that for all $\boldsymbol{x} \in \mathbb{C}^N$ and $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} \in \mathbb{C}^{2|\Omega|}$ with $\|\boldsymbol{e}\|_{\ell^2} \leq \eta$ for some $\eta > 0$, and all $0 < r < 1$, one has the following. Define*

$$\zeta = C \cdot \mathcal{CS}_{s,1}(\boldsymbol{V}\boldsymbol{x}, \boldsymbol{p}, \eta), \qquad \mu_k = \frac{r\sqrt{s}}{CN}\varepsilon_{k-1}, \qquad n_k = \left\lceil \frac{\sqrt{2}CN}{r\sqrt{s}} \right\rceil - 1,$$

where ε_k is a sequence defined recursively by

$$\varepsilon_0 = \frac{\|\boldsymbol{x}\|_{\ell^2}}{\sqrt{N}}, \qquad \varepsilon_{k+1} = r\varepsilon_k + \zeta, \qquad k \geq 0.$$

Let $\mathcal{P}_Q : \mathbb{C}^N \to \mathbb{C}^N$ denote the orthogonal projection map of

$$Q = \{\boldsymbol{z} \in \mathbb{C}^N : \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{y}\|_{\ell^2} \leq \eta\}.$$

Then applying Algorithm 5 with these values of μ_k and n_k, $\boldsymbol{x}_0^\star = \mathcal{P}_Q(\mathbf{0})$, gives iterates $\{\boldsymbol{x}_k^\star\}$ satisfying

$$\frac{\|\boldsymbol{x}_k^\star - \boldsymbol{x}\|_{\ell^2}}{\sqrt{N}} \leq \varepsilon_k, \qquad \varepsilon_{k+1} = r^{k+1}\frac{\|\boldsymbol{x}\|_{\ell^2}}{\sqrt{N}} + \frac{1-r^{k+1}}{1-r}\zeta, \qquad k \geq 0.$$

Theorem 4.3.2 (Performance of restarted NESTA reconstruction, $d \geq 2$)**.** *Consider $d \geq 2$ and $\boldsymbol{A}$, Ω, $\boldsymbol{p}$, ϵ, s, m, N^d, η from Theorem 4.2.2 and suppose the measurement condition (4.2.4) holds. Then the following holds with probability at least $1-\epsilon$. There is a constant C depending on d such that for all $\boldsymbol{x} \in \mathbb{C}^{N^d}$ and $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} \in \mathbb{C}^{2|\Omega|}$ with $\|\boldsymbol{e}\|_{\ell^2} \leq \eta$ for some $\eta > 0$, and all $0 < r < 1$, one has the following. Define*

$$\zeta = C \cdot \mathcal{CS}_{s,d}(\boldsymbol{V}\boldsymbol{x}, \boldsymbol{p}, \eta), \qquad \mu_k = \frac{r\sqrt{s}}{CdN^d}\varepsilon_{k-1}, \qquad n_k = \left\lceil \frac{\sqrt{2}CdN^{d/2}}{r\sqrt{s}} \right\rceil - 1,$$

where ε_k is a sequence defined recursively by

$$\varepsilon_0 = \|\boldsymbol{x}\|_{\ell^2}, \qquad \varepsilon_{k+1} = r\varepsilon_k + \zeta, \qquad k \geq 0.$$

Let $\mathcal{P}_Q : \mathbb{C}^{N^d} \to \mathbb{C}^{N^d}$ denote the orthogonal projection map of

$$Q = \{\boldsymbol{z} \in \mathbb{C}^{N^d} : \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{y}\|_{\ell^2} \leq \eta\}.$$

Then applying Algorithm 5 with these values of μ_k and n_k, $\boldsymbol{x}_0^\star = \mathcal{P}_Q(\mathbf{0})$, gives iterates $\{\boldsymbol{x}_k^\star\}$ satisfying

$$\|\boldsymbol{x}_k^\star - \boldsymbol{x}\|_{\ell^2} \leq \varepsilon_k, \qquad \varepsilon_{k+1} = r^{k+1}\|\boldsymbol{x}\|_{\ell^2} + \frac{1-r^{k+1}}{1-r}\zeta, \qquad k \geq 0.$$

Proof of Theorems 4.3.1 and 4.3.2. Here we only prove the case when $d \geq 2$, since the $d = 1$ case is very similar. The formula for ε_k follows from the recurrence relation. Moreover, the conditions of Theorem 4.2.2 are satisfied, so the bound (4.2.6) holds with probability $1-\epsilon$. Let C denote the constant of $\lesssim_d$ in (4.2.6).

Now we proceed by induction on k to prove the restart scheme error bound. For $k = 0$, note that $\boldsymbol{x} \in Q$ and Q is closed and convex. By the non-expansiveness property of $\mathcal{P}_Q$ (for instance, see [11, Thm. 6.41]) we have

$$\|\boldsymbol{x}_0^\star - \boldsymbol{x}\|_{\ell^2} = \|\mathcal{P}_Q(\mathbf{0}) - \mathcal{P}_Q(\boldsymbol{x})\|_{\ell^2} \leq \|\boldsymbol{x}\|_{\ell^2} = \varepsilon_0,$$

which establishes the base case.

Suppose the result now holds for $k \geq 0$. Then by (4.2.6) we have

$$\|\boldsymbol{x}_{k+1}^\star - \boldsymbol{x}\|_{\ell^2} \leq \zeta + \frac{CdN^d\mu_{k+1}}{2\sqrt{s}} + \frac{Cd}{\mu_{k+1}(n_{k+1}+1)^2\sqrt{s}}\varepsilon_k^2. \tag{4.3.1}$$

By definition of μ_{k+1} and n_{k+1} we have

$$\frac{CdN^d\mu_{k+1}}{2\sqrt{s}} = \frac{r}{2}\varepsilon_k,$$

and

$$\frac{Cd}{\mu_{k+1}(n_{k+1}+1)^2\sqrt{s}}\varepsilon_k^2 = \frac{C^2d^2N^d}{(n_{k+1}+1)^2rs}\varepsilon_k \leq \frac{r}{2}\varepsilon_k.$$

Hence (4.3.1) gives

$$\|\boldsymbol{x}_{k+1}^\star - \boldsymbol{x}\|_{\ell^2} \leq \zeta + r\varepsilon_k = \varepsilon_{k+1},$$

completing the proof. □

Remark 4.3.3. The orthogonal projection map $\mathcal{P}_Q$ is obtained from the stacked NESTA derivation in Section 4.1.3. Explicitly we have

$$\mathcal{P}_Q(\boldsymbol{u}) = \frac{\lambda}{c}\boldsymbol{B}^*(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{u}) + \boldsymbol{u}, \quad \lambda = \max\left\{0, \frac{1}{2}\left(1 - \frac{\sqrt{2\eta^2 - \|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_{\ell^2}^2}}{\|\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{u}\|_{\ell^2}}\right)\right\}.$$

We note that the orthogonal projection of the *zero vector* as the initial point $\boldsymbol{x}_0^\star$ is done for convenience. Other starting points can be used with minor adjustment to the previous results. ◊

To summarize, Theorems 4.3.1 and 4.3.2 describe the parameters $\{\mu_k\}$ and $\{n_k\}$ we sought for. They ensure that the restart scheme (Algorithm 5) produces iterates $\{\boldsymbol{x}_k^\star\}$ for which the image reconstruction error $\|\boldsymbol{x}_k^\star - \boldsymbol{x}\|_{\ell^2}$ decays exponentially in k down to a finite tolerance proportional to ζ. The quantity ζ is itself proportional to the compressed sensing error $\mathcal{CS}_{s,d}(\boldsymbol{V}\boldsymbol{x}, \boldsymbol{p}, \eta)$, thus yielding the desired recovery guarantees. Lastly, there are also a few things to note about the parameter choices. First, the inner iterations n_k does not depend on k, so it is constant. Second, the smoothing parameters μ_k are proportional to the predefined error bound ε_{k-1}, hence they are adjusted to decay exponentially in each restart k. Third, both n_k and μ_k depend on constants that are generally unknown, such as the sparsity s and the constant C, which itself depends on rNSP constants of $\boldsymbol{A}$.

Chapter 5

Neural networks via unrolling optimization algorithms

For this chapter, we detail the construction of NESTANets and prove the main result of this thesis, Theorem 1.3.1. In other words, we provide a stable, accurate and efficient neural network construction for Fourier imaging with a gradient-sparsity model, by unrolling restarted stacked NESTA. The unrolling construction is a minor modification of the one found in [75, Sec. 4.2], and thus there is significant overlap. We nonetheless provide all the proofs here for completeness.

5.1 Class of neural networks

For unrolling, we consider and restate verbatim the neural network architectures described in [5, Sec. 21.3.2]. These are complex-valued feedforward neural networks $\mathcal{N} : \mathbb{C}^m \to \mathbb{C}^N$ of the form

$$\mathcal{N}(\boldsymbol{y}) = \boldsymbol{A}^{(L)} \circ \sigma^{(L-1)} \circ \boldsymbol{A}^{(L-1)} \circ \cdots \circ \sigma^{(1)} \circ \boldsymbol{A}^{(1)}(\boldsymbol{y}),$$

where $L \geq 2$, and for each $l = 1, \ldots, L$, $\boldsymbol{A}^{(l)} : \mathbb{C}^{n_{l-1}} \to \mathbb{C}^{n_l}$ is an affine map of the form

$$\boldsymbol{A}^{(l)}(x) = \boldsymbol{W}^{(l)} x + \boldsymbol{b}^{(l)}(\boldsymbol{y}), \qquad \boldsymbol{W}^{(l)} \in \mathbb{C}^{n_l \times n_{l-1}},$$

whose biases $\boldsymbol{b}^{(l)}(\boldsymbol{y})$ are themselves an affine map of the input $\boldsymbol{y}$, i.e.

$$\boldsymbol{b}^{(l)}(\boldsymbol{y}) = \boldsymbol{R}^{(l)} \boldsymbol{y} + \boldsymbol{c}^{(l)}, \qquad \boldsymbol{R}^{(l)} \in \mathbb{C}^{n_l \times m},\ \boldsymbol{c}^{(l)} \in \mathbb{C}^{n_l}.$$

Here $n_0 = m$ and $n_L = N$. The activation functions $\sigma^{(l)} : \mathbb{C}^{n_l} \to \mathbb{C}^{n_l}$ have one of the two following forms:

1. There is an index set $I^{(l)} \subseteq \{1, \ldots, n_l\}$ such that $\sigma^{(l)}$ acts componentwise on those components of the input vector with indices in $I^{(l)}$ while leaving the rest unchanged.

2. There is a nonlinear function $\rho^{(l)} : \mathbb{C} \to \mathbb{C}$ such that, if the input vector $\boldsymbol{x}$ to the layer takes the form $\boldsymbol{x} = (x_1, \boldsymbol{u}, \boldsymbol{v})$, where x_1 is a scalar and $\boldsymbol{u}$, $\boldsymbol{v}$ are (possibly) vectors, then $\sigma^{(l)}(\boldsymbol{x}) = (0, \rho^{(l)}(x_1)\boldsymbol{u}, \boldsymbol{v})$.

We denote the class of networks of this form as $\mathcal{N}^* = \mathcal{N}^*_{\boldsymbol{n},L,q}$, where

$$\boldsymbol{n} = (n_0, n_1, \ldots, n_{L-1}, n_L), \qquad n_0 = m,\ n_L = N,$$

with L denoting the number of layers and q the number of different nonlinear activation functions used.

While the class $\mathcal{N}^*$ and specification of activation functions is broader than what is often seen with common feed-forward neural network architectures, it is both useful and standard when unrolling optimization algorithms. For more information, see [5, Sec. 21.3.2] and [30].

5.2 Unrolled NESTA construction

We adopt the same unrolling approach of NESTANets from [75, Sec. 4.2]. First we construct networks for the update steps of stacked NESTA (Algorithm 2), then combine them to compute one iteration of stacked NESTA. Finally, this is used to unroll n iterations of stacked NESTA as a network, and in turn, unroll restarted NESTA. Given that stacked NESTA differs from NESTA presented in [75], the proofs require slight modification. We present them here for completeness.

Lemma 5.2.1. *Let $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathbb{C}^N$, $\boldsymbol{W} \in \mathbb{C}^{N\times M}$, $\alpha \in \mathbb{C}$, and $\mathcal{T}_\mu : \mathbb{C}^M \to \mathbb{C}^M$ be the Huber function gradient (4.1.4). Then the map $\mathbb{C}^{2N} \to \mathbb{C}^N$ defined by*

$$\begin{pmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{pmatrix} \mapsto \boldsymbol{z}_1 - \alpha \boldsymbol{W}\mathcal{T}_\mu(\boldsymbol{W}^*\boldsymbol{z}_2)$$

can be expressed as a neural network $\mathcal{N} \in \mathcal{N}^*_{\boldsymbol{n},2,1}$ with $\boldsymbol{n} = (2N, N+M, N)$ and with all biases equal to zero, i.e. independent of the input.

Proof. Write the map as the following sequence of maps

$$\begin{pmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{pmatrix} \overset{\text{(a)}}{\longmapsto} \begin{pmatrix} \boldsymbol{z}_1 \\ \boldsymbol{W}^*\boldsymbol{z}_2 \end{pmatrix} \overset{\text{(b)}}{\longmapsto} \begin{pmatrix} \boldsymbol{z}_1 \\ \mathcal{T}_\mu(\boldsymbol{W}^*\boldsymbol{z}_2) \end{pmatrix} \overset{\text{(c)}}{\longmapsto} \boldsymbol{z}_1 - \alpha \boldsymbol{W}\mathcal{T}_\mu(\boldsymbol{W}^*\boldsymbol{z}_2).$$

Here (a) and (c) are linear maps, noting that $\alpha \in \mathbb{C}$ and $\boldsymbol{W} \in \mathbb{C}^{N\times M}$ are fixed. The map (b) applies the gradient of the Huber function with fixed parameter μ, componentwise to the M entries of $\boldsymbol{W}^*\boldsymbol{z}_2$. Such a map corresponds to a nonlinear activation function of type (i). This gives the result. □

Lemma 5.2.2. *Fix $\eta > 0$, $\boldsymbol{B} \in \mathbb{C}^{\frac{m}{2} \times N}$ and $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{C}^{\frac{m}{2}}$. The map $\mathbb{C}^N \to \mathbb{C}$ defined by*

$$\boldsymbol{q} \mapsto \max\left\{0, \frac{1}{2}\left(1 - \frac{\sqrt{2\eta^2 - \|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_{\ell^2}^2}}{\|\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}\|_{\ell^2}}\right)\right\}$$

can be expressed as a neural network $\mathcal{N} \in \mathcal{N}^*_{\boldsymbol{n},4,3}$ with $\boldsymbol{n} = (N, m, 2, 1, 1)$ and biases depending affinely on $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, but otherwise independent of the input.

Proof. For brevity, let σ_1 denote the squaring activation function $x \mapsto |x|^2$ and σ_2 denote the nonlinear activation function $x \mapsto \max\left\{0, \frac{1}{2}(1 - x^{-1/2})\right\}$. Then we can express the map in question as the following sequence

$$\begin{aligned}\boldsymbol{q} &\overset{\text{(a)}}{\longmapsto} \begin{pmatrix} \boldsymbol{y}_1 - \boldsymbol{y}_2 \\ \boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q} \end{pmatrix} \overset{\text{(b)}}{\longmapsto} \begin{pmatrix} \sigma_1(\boldsymbol{y}_1 - \boldsymbol{y}_2) \\ \sigma_1(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}) \end{pmatrix} \overset{\text{(c)}}{\longmapsto} \begin{pmatrix} 2\eta^2 - \mathbf{1}^\top \sigma_1(\boldsymbol{y}_1 - \boldsymbol{y}_2) \\ \mathbf{1}^\top \sigma_1(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}) \end{pmatrix} \\ &\overset{\text{(d)}}{\longmapsto} \begin{pmatrix} 0 \\ \frac{\mathbf{1}^\top \sigma_1(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q})}{2\eta^2 - \mathbf{1}^\top \sigma_1(\boldsymbol{y}_1 - \boldsymbol{y}_2)} \end{pmatrix} \overset{\text{(e)}}{\longmapsto} \frac{\mathbf{1}^\top \sigma_1(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q})}{2\eta^2 - \mathbf{1}^\top \sigma_1(\boldsymbol{y}_1 - \boldsymbol{y}_2)} \overset{\text{(f)}}{\longmapsto} \sigma_2\left(\frac{\mathbf{1}^\top \sigma_1(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q})}{2\eta^2 - \mathbf{1}^\top \sigma_1(\boldsymbol{y}_1 - \boldsymbol{y}_2)}\right)\end{aligned}$$

Here $\mathbf{1}$ denotes a vector of ones, where for the above calculation they have $m/2$ entries. Moreover, we have the identity $\mathbf{1}^\top \sigma_1(\boldsymbol{x}) = \|\boldsymbol{x}\|_{\ell^2}^2$. Now, (a) and (c) are affine maps and (e) is a linear map. The maps (b) and (f) apply the nonlinear activation functions σ_1 and σ_2, respectively. Both (b) and (f) are of type (i). The map (d) applies the nonlinear activation function $x \mapsto x^{-1}$, which corresponds to an activation function of type (ii). Finally, to ensure the above sequence of maps corresponds to a network in $\mathcal{N}^*_{\boldsymbol{n},4,3}$, the last map must be affine. We achieve this by appending the identity map to the end of the sequence. Combining these facts gives the desired neural network. □

Lemma 5.2.3. *Let $\lambda, c \in \mathbb{C}$, $\boldsymbol{q} \in \mathbb{C}^N$, $\boldsymbol{B} \in \mathbb{C}^{\frac{m}{2} \times N}$, and $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{C}^{\frac{m}{2}}$. Then the map $\mathbb{C}^{N+1} \to \mathbb{C}^N$ described by*

$$\begin{pmatrix} \lambda \\ \boldsymbol{q} \end{pmatrix} \mapsto \frac{\lambda}{c}\boldsymbol{B}^*(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}) + \boldsymbol{q}$$

can be expressed as a neural network $\mathcal{N} \in \mathcal{N}^*_{\boldsymbol{n},2,1}$ with $\boldsymbol{n} = (N+1, 2N+1, N)$ and biases depending affinely on $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, but otherwise independent of the input.

Proof. Considering the sequence

$$\begin{aligned}\begin{pmatrix} \lambda \\ \boldsymbol{q} \end{pmatrix} &\overset{\text{(a)}}{\longmapsto} \begin{pmatrix} \lambda \\ \frac{1}{c}\boldsymbol{B}^*(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}) \\ \boldsymbol{q} \end{pmatrix} \overset{\text{(b)}}{\longmapsto} \begin{pmatrix} 0 \\ \frac{\lambda}{c}\boldsymbol{B}^*(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}) \\ \boldsymbol{q} \end{pmatrix} \\ &\overset{\text{(c)}}{\longmapsto} \frac{\lambda}{c}\boldsymbol{B}^*(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}) + \boldsymbol{q}\end{aligned}$$

the result follows from observing that the map (a) is affine, (c) is linear, and (b) uses a nonlinear activation function of type (ii) corresponding to the identity map. □

Using the aforementioned lemmas, we now construct a network that computes one iterative step of Algorithm 2. Recall that the nth iteration performs the update $\boldsymbol{z}_n \to \boldsymbol{z}_{n+1}$. To simplify writing this as a neural network, we keep track of the value of $\boldsymbol{q}$ used to compute the intermediate vector $\boldsymbol{v}_n$, as this value depends on not just $\boldsymbol{z}_n$, but also $\boldsymbol{z}_0, \dots, \boldsymbol{z}_{n-1}$. Therefore, the map we want to derive a network for is

$$\begin{pmatrix} \boldsymbol{q}_v^{(n-1)} \\ \boldsymbol{z}_n \end{pmatrix} \mapsto \begin{pmatrix} \boldsymbol{q}_v^{(n)} \\ \boldsymbol{z}_{n+1} \end{pmatrix}. \tag{5.2.1}$$

The vector $\boldsymbol{q}_v^{(k)}$ refers to the value of $\boldsymbol{q}$ used to calculate $\boldsymbol{v}_k$ – see line 7 of Algorithm 2. We analogously define the vectors and scalars $\boldsymbol{q}_x^{(k)}$, $\lambda_v^{(k)}$ and $\lambda_x^{(k)}$ to be $\boldsymbol{q}$ and λ used for $\boldsymbol{x}_k$ and $\boldsymbol{v}_k$, which is inferred from the notation. For convenience, when $n = 0$ we set $\boldsymbol{q}_v^{(-1)} = \boldsymbol{z}_0$, where $\boldsymbol{z}_0$ is the initial vector of Algorithm 2.

Lemma 5.2.4. *The update step* (5.2.1) *can be performed by a neural network* $\mathcal{N} \in \mathcal{N}^*_{\boldsymbol{n},7,5}$ *where*

$$\boldsymbol{n} = (2N, 2N + M, 2N + 3m/2, 2N + 3, 2N + 2, 3N + 2, 3N + 1, 2N)$$

and the biases depend affinely on $\boldsymbol{y}_1$ *and* $\boldsymbol{y}_2$ *only. Moreover, the nonlinear activations are independent of* n.

Proof. First we write (5.2.1) as the sequence of maps

$$\begin{pmatrix} \boldsymbol{q}_v^{(n-1)} \\ \boldsymbol{z}_n \end{pmatrix} \xmapsto{T_1} \begin{pmatrix} \boldsymbol{q}_v^{(n)} \\ \boldsymbol{q}_x^{(n)} \end{pmatrix} \xmapsto{T_2} \begin{pmatrix} \lambda_v^{(n)} \\ \lambda_x^{(n)} \\ \boldsymbol{q}_v^{(n)} \\ \boldsymbol{q}_x^{(n)} \end{pmatrix} \xmapsto{T_3} \begin{pmatrix} \boldsymbol{q}_v^{(n)} \\ \boldsymbol{z}_{n+1} \end{pmatrix}.$$

For T_1, we know that the corresponding Algorithm 2 updates are

$$\boldsymbol{q}_v^{(n)} = \boldsymbol{q}_v^{(n-1)} - \frac{\mu}{\|\boldsymbol{W}\|_{\ell^2}^2} \alpha_n \boldsymbol{W} \mathcal{T}_\mu(\boldsymbol{W}^* \boldsymbol{z}_n), \tag{5.2.2}$$

$$\boldsymbol{q}_x^{(n)} = \boldsymbol{z}_n - \frac{\mu}{\|\boldsymbol{W}\|_{\ell^2}^2} \boldsymbol{W} \mathcal{T}_\mu(\boldsymbol{W}^* \boldsymbol{z}_n), \tag{5.2.3}$$

for all $n \geq 0$. By Lemma 5.2.1, (5.2.2) and (5.2.3) can be expressed using neural networks $\mathcal{N}_{\boldsymbol{v}_n}^{(1)}, \mathcal{N}_{\boldsymbol{x}_n}^{(1)} \in \mathcal{N}^*_{(2N,N+M,N),2,1}$, where $\mathcal{N}_{\boldsymbol{v}_n}^{(1)}$ uses $\alpha := \mu\alpha_n / \|\boldsymbol{W}\|_{\ell^2}^2$ and $\mathcal{N}_{\boldsymbol{x}_n}^{(1)}$ uses $\alpha := \mu / \|\boldsymbol{W}\|_{\ell^2}^2$. Thus

$$\boldsymbol{q}_v^{(n)} = \mathcal{N}_{\boldsymbol{v}_n}^{(1)}(\boldsymbol{q}_v^{(n-1)}, \boldsymbol{z}_n), \qquad \boldsymbol{q}_x^{(n)} = \mathcal{N}_{\boldsymbol{x}_n}^{(1)}(\boldsymbol{z}_n, \boldsymbol{z}_n).$$

These networks can be run in parallel and be embedded into a larger network that computes T_1. We do this by stacking the layers of $\mathcal{N}_{\boldsymbol{v}_n}^{(1)}$ and $\mathcal{N}_{\boldsymbol{x}_n}^{(1)}$ on top of each other and merging redundant copies of vectors and their network connections. Any missing connections simply correspond to zero weights. Note that a permutation of elements in the layers would simply be a linear mapping applied to the affine map for that layer before the nonlinear activation. This procedure yields the map sequence (with affine and nonlinear activations combined into one map)

$$\begin{pmatrix} \boldsymbol{q}_v^{(n-1)} \\ \boldsymbol{z}_n \end{pmatrix} \mapsto \begin{pmatrix} \boldsymbol{q}_v^{(n-1)} \\ \boldsymbol{z}_n \\ \mathcal{T}_\mu(\boldsymbol{W}^* \boldsymbol{z}_n) \end{pmatrix} \mapsto \begin{pmatrix} \boldsymbol{q}_v^{(n)} \\ \boldsymbol{q}_x^{(n)} \end{pmatrix}.$$

Note that the only nonlinear activations used here are of type (i). The above map defines a network $\mathcal{N}^{(1)} \in \mathcal{N}^*_{(2N,2N+M,2N),2,1}$ that computes T_1.

Regarding map T_2, by Lemma 5.2.2 and the definition of Algorithm 2, $\lambda_v^{(n)}$ and $\lambda_x^{(n)}$ each can be expressed as the output of a network $\mathcal{N}_\lambda^{(2)} \in \mathcal{N}^*_{(N,m,2,1,1),4,3}$, where $\boldsymbol{y}_1$, $\boldsymbol{y}_2$ in the lemma corresponds to $\boldsymbol{y}_1$, $\boldsymbol{y}_2$ here. Thus

$$\lambda_v^{(n)} = \mathcal{N}_\lambda^{(2)}(\boldsymbol{q}_v^{(n)}), \qquad \lambda_x^{(n)} = \mathcal{N}_\lambda^{(2)}(\boldsymbol{q}_x^{(n)}). \tag{5.2.4}$$

Adopting the same strategy as for map T_1, we construct a network computing both $\lambda_v^{(n)}$ and $\lambda_x^{(n)}$ in parallel. This gives the network layer sequence (with affine map and nonlinear activation combined per mapping)

$$\begin{pmatrix} \boldsymbol{q}_v^{(n)} \\ \boldsymbol{q}_x^{(n)} \end{pmatrix} \mapsto \begin{pmatrix} \sigma_1(\boldsymbol{y}_1 - \boldsymbol{y}_2) \\ \sigma_1(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}_v^{(n)}) \\ \sigma_1(\boldsymbol{y}_1 + \boldsymbol{y}_2 - 2\boldsymbol{B}\boldsymbol{q}_x^{(n)}) \\ \boldsymbol{q}_v^{(n)} \\ \boldsymbol{q}_x^{(n)} \end{pmatrix} \mapsto \begin{pmatrix} 0 \\ \frac{\mathbf{1}^\top \sigma_1(\boldsymbol{y}_1+\boldsymbol{y}_2-2\boldsymbol{B}\boldsymbol{q}_v^{(n)})}{2\eta^2 - \mathbf{1}^\top \sigma_1(\boldsymbol{y}_1-\boldsymbol{y}_2)} \\ \frac{\mathbf{1}^\top \sigma_1(\boldsymbol{y}_1+\boldsymbol{y}_2-2\boldsymbol{B}\boldsymbol{q}_x^{(n)})}{2\eta^2 - \mathbf{1}^\top \sigma_1(\boldsymbol{y}_1-\boldsymbol{y}_2)} \\ \boldsymbol{q}_v^{(n)} \\ \boldsymbol{q}_x^{(n)} \end{pmatrix} \mapsto \begin{pmatrix} \lambda_v^{(n)} \\ \lambda_x^{(n)} \\ \boldsymbol{q}_v^{(n)} \\ \boldsymbol{q}_x^{(n)} \end{pmatrix}.$$

Note that σ_1 is the squaring activation function from Lemma 5.2.2. This map sequence embeds two copies of $\mathcal{N}_\lambda^{(2)}$ and four identity maps. Note that we have implicitly included an identity map as the final affine map of the network. Moreover, the single type (ii) activation function in $\mathcal{N}_\lambda^{(2)}$ is applied to bias terms, and is thus independent of the input. In both evaluations of (5.2.4), the scalar input to the type (ii) activation are the same. By construction, T_2 is computed by a network $\mathcal{N}^{(2)} \in \mathcal{N}^*_{\boldsymbol{a},4,3}$ where

$$\boldsymbol{a} = (2N, 2N + 3m/2, 2N + 3, 2N + 2, 2N + 2)\,,$$

with biases as affine maps of $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$.

Lastly, we construct a network that computes T_3. Consider T_3 as the sequence of maps

$$\begin{pmatrix} \lambda_v^{(n)} \\ \lambda_x^{(n)} \\ \boldsymbol{q}_v^{(n)} \\ \boldsymbol{q}_x^{(n)} \end{pmatrix} \overset{(a)}{\longmapsto} \begin{pmatrix} 0 \\ \frac{\lambda_x^{(n)}}{c}\boldsymbol{B}^*(\boldsymbol{y}_1+\boldsymbol{y}_2-2\boldsymbol{B}\boldsymbol{q}_x^{(n)}) \\ \boldsymbol{q}_x^{(n)} \\ \lambda_v^{(n)} \\ \boldsymbol{q}_v^{(n)} \end{pmatrix} \overset{(b)}{\longmapsto} \begin{pmatrix} 0 \\ \frac{\lambda_v^{(n)}}{c}\boldsymbol{B}^*(\boldsymbol{y}_1+\boldsymbol{y}_2-2\boldsymbol{B}\boldsymbol{q}_v^{(n)}) \\ \boldsymbol{x}_n \\ \boldsymbol{q}_v^{(n)} \end{pmatrix} \overset{(c)}{\longmapsto} \begin{pmatrix} \boldsymbol{q}_v^{(n)} \\ \boldsymbol{z}_{n+1} \end{pmatrix}.$$

Using Lemma 5.2.3, we deduce the following. The map (a) first applies an affine map, then uses a nonlinear activation of type (ii) corresponding to the identity map using the scalar $\lambda_x^{(n)}$. Similar to (a), map (b) first performs an affine mapping, then uses the same nonlinear activation as in (a), but instead using the scalar $\lambda_v^{(n)}$. Lastly, the final map (c) is linear, noting that $\boldsymbol{z}_{n+1} = \tau_n \boldsymbol{v}_n + (1-\tau_n)\boldsymbol{x}_n$. Observe that the bias terms of the affine mappings are affine in $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. This gives us the desired network $\mathcal{N}^{(3)}$ corresponding to T_3, belonging to the class $\mathcal{N}^{(3)} \in \mathcal{N}^*_{\boldsymbol{b},3,1}$ with

$$\boldsymbol{b} = (2N+2, 3N+2, 3N+1, 2N).$$

Composing the networks to form $\mathcal{N} = \mathcal{N}^{(3)} \circ \mathcal{N}^{(2)} \circ \mathcal{N}^{(1)} \in \mathcal{N}^*_{\boldsymbol{n},7,5}$, noting that in the composition, we merge each set of consecutively composed affine maps into a single affine map, gives a network that computes one iteration of Algorithm 2, i.e. (5.2.1), at any step n. Moreover, $\mathcal{N}$ has the property that all of its bias terms are affine maps of $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, and the nonlinear activations do not depend on n. This completes the proof. □

Theorem 5.2.5 (Unrolled NESTA)**.** *For $n \geq 0$, let $\boldsymbol{x}_n$ be the nth iterate produced by NESTA (Algorithm 2) with input $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2) \in \mathbb{C}^m$, so that $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{C}^{\frac{m}{2}}$, and initial point $\boldsymbol{z}_0$. Then there exists a neural network $\mathcal{N} \in \mathcal{N}^*_{\boldsymbol{n},6(n+1)+1,5}$ with activation functions independent of n, and*

$$\boldsymbol{n} = (m, \underbrace{2N+M, 2N+3m/2, 2N+3, 2N+2, 3N+2, 3N+1}_{n+1 \text{ times}}, N),$$

such that

$$\boldsymbol{x}_n = \mathcal{N}(\boldsymbol{y}).$$

Proof. Let $\phi : \mathbb{C}^m \to \mathbb{C}^{2N}$ be the affine map defined by $\boldsymbol{y} \mapsto (\boldsymbol{z}_0, \boldsymbol{z}_0)$, and $\mathcal{N}_0, \mathcal{N}_1, \ldots, \mathcal{N}_{n-1}$ be copies of the network in Lemma 5.2.4. Since we plan to compose $\mathcal{N}_0$ with ϕ, note that this defines $\boldsymbol{q}_v^{(-1)} = \boldsymbol{z}_0$. Moreover, define $\mathcal{N}_n$ as a modification of the network in Lemma 5.2.4 by changing the linear map of its last layer to output $\boldsymbol{x}_n$ instead of $(\boldsymbol{q}_v^{(n)}, \boldsymbol{z}_{n+1})$. This is done by omitting $\boldsymbol{q}_v^{(n)}$ and setting τ_n to be zero, since $\boldsymbol{z}_{n+1} = \tau_n \boldsymbol{v}_n + (1-\tau_n)\boldsymbol{x}_n$. Then the

composition

$$\mathcal{N} = \mathcal{N}_n \circ \mathcal{N}_{n-1} \circ \cdots \circ \mathcal{N}_0 \circ \phi,$$

where each set of consecutively composed affine maps are merged into a single affine map, gives the desired network. □

Theorem 5.2.6 (Unrolled restarted NESTA)**.** *Let $\boldsymbol{x}_{t+1}^\star$ be the output of restarted NESTA (Algorithm 5) with $t \geq 0$ restarts and $n_k = n \geq 0$ for all $k = 1, \ldots, t+1$ corresponding to a fixed number of NESTA iterations for each restart. Additionally, let $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2) \in \mathbb{C}^m$ be the input, with $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{C}^{\frac{m}{2}}$, and $\boldsymbol{x}_0^\star$ the initial point. Then there exists a neural network $\mathcal{N} \in \mathcal{N}^*_{\boldsymbol{n}, 6(t+1)(n+1)+1, 5}$ with activation functions independent of n and t, and*

$$\boldsymbol{n} = (m, \underbrace{2N + M, 2N + 3m/2, 2N + 3, 2N + 2, 3N + 2, 3N + 1}_{(t+1)(n+1) \text{ times}}, N)$$

such that

$$\boldsymbol{x}_{t+1}^\star = \mathcal{N}(\boldsymbol{y}).$$

Proof. From the proof of Theorem 5.2.5, consider $\phi, \mathcal{N}_0, \ldots, \mathcal{N}_{n-1}, \mathcal{N}_n$. Define $\widetilde{\mathcal{N}}_n$ as a modification of $\mathcal{N}_n$ that outputs $(\boldsymbol{x}_n, \boldsymbol{x}_n)$ instead of $\boldsymbol{x}_n$. Then defining the compositions

$$\begin{aligned} \mathcal{N}_k^\star &= \widetilde{\mathcal{N}}_n \circ \mathcal{N}_{n-1} \circ \cdots \circ \mathcal{N}_0, \qquad k = 1, \ldots, t, \\ \mathcal{N}_{t+1}^\star &= \mathcal{N}_n \circ \mathcal{N}_{n-1} \circ \cdots \circ \mathcal{N}_0, \end{aligned}$$

the desired network is

$$\mathcal{N} = \mathcal{N}_{t+1}^\star \circ \mathcal{N}_t^\star \circ \cdots \circ \mathcal{N}_1^\star \circ \phi,$$

with each set of consecutively composed affine maps merged into a single affine map. □

5.3 Stable, accurate, and efficient neural network for TV-Fourier problems

Reiterating the setup of Chapter 1, we define the model class of Fourier measurements we seek to recover from gradient-sparse images. Fix $d \geq 1$, $\eta > 0$, $1 \leq s \leq N^d$ and define the *compressed sensing error*

$$\mathcal{CS}_{s,d}(\boldsymbol{V}\boldsymbol{x}, \eta) = \frac{\sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1}}{\sqrt{s}} + \left(d + \sqrt{\log(N)}\right)\eta.$$

Given $\chi > 0$ and $\boldsymbol{A} \in \mathbb{C}^{m \times N^d}$, we write

$$\mathbb{I}_{\boldsymbol{V},\chi,\eta} = \left\{(\boldsymbol{x}, \boldsymbol{e}) \in \mathbb{C}^{N^d} \times \mathbb{C}^m : \|\boldsymbol{x}\|_{\ell^2} \leq 1,\ \|\boldsymbol{e}\|_{\ell^2} \leq \eta,\ \mathcal{CS}_{s,d}(\boldsymbol{V}\boldsymbol{x}, \eta) \leq \chi\right\},$$

and define the class of measurement vectors

$$\mathbb{M}_{\boldsymbol{A},\boldsymbol{V},\chi,\eta} = \{\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} \in \mathbb{C}^m : (\boldsymbol{x}, \boldsymbol{e}) \in \mathbb{I}_{\boldsymbol{V},\chi,\eta}\}. \tag{5.3.1}$$

We simply write $\mathbb{I} = \mathbb{I}_{\boldsymbol{V},\chi,\eta}$ and $\mathbb{M} = \mathbb{M}_{\boldsymbol{A},\boldsymbol{V},\chi,\eta}$ when the parameters can be inferred from context. To remind the reader, the interpretation of $\mathbb{M}$ is that it defines noisy measurement vectors $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$ from images $\boldsymbol{x} \in \mathbb{C}^{N^d}$ that are approximately gradient-sparse, i.e. $\sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1}/\sqrt{s} \leq \chi$, and noise vectors $\boldsymbol{e}$ with bounded ℓ^2-norm, i.e. $\|\boldsymbol{e}\|_{\ell^2} \leq \eta \leq \chi$. In essence, $\mathbb{M}$ formalizes the Fourier inverse problems that can be solved well (in the sense of χ) via TV minimization.

The following two theorems are the main results of this thesis. In particular, they imply the previously stated Theorem 1.3.1 from Chapter 1.

Theorem 5.3.1 (Stable, accurate and efficient neural networks for Fourier imaging, $d = 1$)**.** *Let $d = 1$, $0 < \epsilon < 1$, $2 \leq s, m \leq N$, and $\hat{\boldsymbol{p}}$ be the Bernoulli vector from Section 3.2.3. Define the subsampled Fourier matrix*

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{B} \end{bmatrix}, \qquad \boldsymbol{B} = \frac{1}{\sqrt{m}} \boldsymbol{P}_\Omega \boldsymbol{F} \in \mathbb{C}^{|\Omega| \times N},$$

where $\Omega = \Omega_1 \cup \Omega_2$ satisfies $\Omega_1 \sim \mathrm{Ber}([\![N]\!], m/2)$ and $\Omega_2 \sim \mathrm{Ber}([\![N]\!], m/2, \hat{\boldsymbol{p}})$, with $\mathbb{E}(|\Omega|) = m\left(1 - \frac{m}{4N}\right) \asymp m$ from Section 3.3. In addition, let $\eta \geq 0$, $\chi > 0$ and consider the class $\mathbb{M} = \mathbb{M}_{\boldsymbol{A},\boldsymbol{V},\chi,\eta}$ defined in (5.3.1). Then the following holds with probability at least $1 - \epsilon$ provided that

$$m \gtrsim \log(N) \cdot s \cdot \left(\log(N \log(N) s) \cdot \log^2(s) \cdot \log(N) + \log(2\epsilon^{-1})\right).$$

For every $0 < r < 1$ and $k \geq 1$ one can construct a neural network $\mathcal{N} : \mathbb{C}^{2|\Omega|} \to \mathbb{C}^N$ such that

$$\frac{\|\boldsymbol{x} - \mathcal{N}(\boldsymbol{y})\|_{\ell^2}}{\sqrt{N}} \leq c_1 \cdot \chi + r^k, \qquad \forall \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} \in \mathbb{M},$$

where c_1 only depends on r. The network depth is bounded by $c_2 \cdot \sqrt{\frac{N}{s}} \cdot k$, where c_2 only depends on r, and the network width is bounded by $4N$.

Theorem 5.3.2 (Stable, accurate and efficient neural networks for Fourier imaging, $d \geq 2$)**.** *Let $d \geq 2$, $0 < \epsilon < 1$, $2 \leq s, m \leq N^d$, and $\hat{\boldsymbol{p}}$ be the Bernoulli vector from Section 3.2.3. Define the subsampled Fourier matrix*

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{B} \end{bmatrix}, \qquad \boldsymbol{B} = \frac{1}{\sqrt{m}} \boldsymbol{P}_\Omega \boldsymbol{F} \in \mathbb{C}^{|\Omega| \times N^d},$$

where $\Omega = \Omega_1 \cup \Omega_2$ *satisfies* $\Omega_1 \sim \mathrm{Ber}([\![N^d]\!], m/2)$ *and* $\Omega_2 \sim \mathrm{Ber}([\![N^d]\!], m/2, \hat{\boldsymbol{p}})$*, with* $\mathbb{E}(|\Omega|) = m\left(1 - \frac{m}{4N^d}\right) \asymp m$ *from Section 3.3. In addition, let* $\eta \geq 0$*,* $\chi > 0$ *and consider the class* $\mathbb{M} = \mathbb{M}_{\boldsymbol{A},\boldsymbol{V},\chi,\eta}$ *defined in (5.3.1). Then the following holds with probability at least* $1 - \epsilon$ *provided that*

$$m \gtrsim_d \log^3(N) \cdot s \cdot \left(\log(N\log^3(N)s) \cdot \log^2(s\log^2(N)) \cdot \log(N) + \log(2\epsilon^{-1})\right). \tag{5.3.2}$$

For every $0 < r < 1$ *and* $k \geq 1$ *one can construct a neural network* $\mathcal{N} : \mathbb{C}^{2|\Omega|} \to \mathbb{C}^{N^d}$ *such that*

$$\|\boldsymbol{x} - \mathcal{N}(\boldsymbol{y})\|_{\ell^2} \leq c_1 \cdot \chi + r^k, \qquad \forall \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} \in \mathbb{M},$$

where c_1 *only depends on* r *and* d*. The network depth is bounded by* $c_2 \cdot \sqrt{\frac{N^d}{s}} \cdot k$*, where* c_2 *only depends on* r *and* d*, and the network width is bounded by* $(3+d)N^d$*.*

Proofs of Theorems 5.3.1 and 5.3.2. We only prove the result for $d \geq 2$ since the $d = 1$ case is very similar. What follows is a slight modification of the proof of [75, Thm. 1].

Fix $k \geq 1$ and $0 < r < 1$. Let $\boldsymbol{p} = \hat{\boldsymbol{p}}$ be the near-optimal Bernoulli vector from Section 3.2.3 and C the constant stated from Theorem 4.3.2. Consider Algorithm 5 defined with $K = k - 1$ restarts and parameters specified in Theorem 4.3.2. By Theorem 5.2.6, there exists a neural network $\mathcal{N} \in \mathcal{N}^*_{\boldsymbol{n},6k(n+1)+1,5}$ with

$$\begin{aligned} n &= \left\lceil \frac{\sqrt{2}CdN^{d/2}}{r\sqrt{s}} \right\rceil - 1, \\ \boldsymbol{n} &= (2|\Omega|, \underbrace{(2+d)N^d, 2N^d + 3m/2, 2N^d + 3, 2N^d + 2, 3N^d + 2, 3N^d + 1}_{k(n+1) \text{ times}}, N^d), \end{aligned}$$

satisfying: for any input $\boldsymbol{y}$ to restarted NESTA, the final iterate $\boldsymbol{x}_k^\star$ is computed by $\mathcal{N}$, i.e. $\mathcal{N}(\boldsymbol{y}) = \boldsymbol{x}_k^\star$. Now we specify the input $\boldsymbol{y} \in \mathbb{M}_{\boldsymbol{A},\boldsymbol{V},\chi,\eta}$ so that $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$ for some $(\boldsymbol{x}, \boldsymbol{e}) \in \mathbb{I}_{\boldsymbol{V},\chi,\eta}$. Then using $\mathcal{N}(\boldsymbol{y}) = \boldsymbol{x}_k^\star$ and Theorem 4.3.2, we have

$$\|\boldsymbol{x} - \mathcal{N}(\boldsymbol{y})\|_{\ell^2} \leq r^k \|\boldsymbol{x}\|_{\ell^2} + \frac{1 - r^k}{1 - r}\zeta.$$

By definition of the class $\mathbb{M}$ and ζ, we have $\|\boldsymbol{x}\|_{\ell^2} \leq 1$ and $\zeta \leq C \cdot \mathcal{CS}(\boldsymbol{V}\boldsymbol{x}, \hat{\boldsymbol{p}}, \eta) \leq C\chi$. Using this we further bound the error between $\boldsymbol{x}$ and $\mathcal{N}(\boldsymbol{y})$ to get

$$\|\boldsymbol{x} - \mathcal{N}(\boldsymbol{y})\|_{\ell^2} \leq r^k + \frac{1 - r^k}{1 - r}C\chi \leq r^k + \frac{C\chi}{1 - r}.$$

Thus c_1 and c_2 as in the statement of the main result (Theorem 5.3.2) are identified by

$$c_1 = \frac{C}{1 - r}, \qquad c_2 = \frac{6\sqrt{2}Cd}{r} + 7.$$

Reading off the layer sizes in $\boldsymbol{n}$, the width of the network is bounded above by $(3+d)N^d$. Finally, $\boldsymbol{y}$ was arbitrary. Using $\Gamma(\hat{\boldsymbol{p}}) \lesssim_d \log(N)$, the measurement condition (5.3.2) is sufficient for Theorem 4.3.2 to hold with probability at least $1-\epsilon$, so the main result holds with probability at least $1-\epsilon$. This completes the proof. □

Let us end this chapter by going over what we have stated and proven. In essence, these results tell us what we sought for: we can construct efficient neural networks with stable and accurate recovery of gradient-sparse images from Fourier measurements. In particular, such networks match the state-of-the-art performance of model-based methods based on compressed sensing, with high probability.

The error bounds show that the image reconstruction is guaranteed to be within an error proportional to χ and a term decaying exponentially in k. Choosing $k = \lceil |\log(\chi)/\log(1/r)| \rceil$ yields a network that can perform image reconstruction within an error proportional to the desired error χ. This is a measure of efficiency for our network construction, where to guarantee reconstruction within error proportional to χ, the network depth should scale logarithmically in χ. Repeating once more for emphasis, the parameter χ describes the model class of measurements, with recovery error up to distance from the model class of gradient-sparse images (i.e. accuracy in the sense of $\sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1}/\sqrt{s} \leq \chi$) and the noise level (i.e. stability in the sense that $\|\boldsymbol{e}\|_{\ell^2} \leq \eta \leq \chi$).

As stated before, this is precisely analogous to [30, Thm. 4] and [75, Thm. 1]. In addition, the network construction in [30, Thm. 3] has a depth proportional to np layers, where n is the restart number and $p \propto \|\boldsymbol{A}\|_{\ell^2}$. If $\boldsymbol{A} \in \mathbb{C}^{m \times N^d}$ has the RIP, which is the condition we consider when building our random measurement matrices, then $\|\boldsymbol{A}\|_{\ell^2} \lesssim \sqrt{N^d/s}$ by [5, Rem. 8.8]. This is comparable to the number of layers we have. The same can be said of the network construction in [75].

Finally, in relation to Theorem 1.3.1, there are a few points to make. First, the results of Theorems 5.3.1 and 5.3.2 provide an explicit description for the sampling scheme used. Second, the results hold for a general decay factor $0 < r < 1$, rather than only $r = \mathrm{e}^{-1}$ in Theorem 1.3.1. The latter is motivated by the discussion in Section 6.3 regarding the choice of r for numerical experiments. Third, from the discussion in Section 3.3, we have $\mathbb{E}(|\Omega|) \lesssim m$ and $\mathbb{E}(|\Omega|) \gtrsim m$, yielding the statement $\mathbb{E}(|\Omega|) \asymp m$ in Theorem 1.3.1. Despite $|\Omega|$ being a random variable, deviating from its expected value occurs with exponentially decaying probability.

Chapter 6

Numerical experiments

In this chapter, we showcase several numerical experiments that aim to reaffirm theoretical results and otherwise explore gaps between theory and practice. Our presentation is in tandem with [75, Sec. 6], and has considerable overlap with the experiments presented and discussion found therein. For the first experiment, we demonstrate that the exponential decay in reconstruction error occurs as in the error bound of Theorem 5.3.2. Second, we compare the performance of NESTANets with and without restarts (Algorithms 2 and 5, respectively). Third, we discuss and provide insight on hyperparameter tuning of stacked NESTA, accompanied by two experiments for empirical justification. Lastly, the fifth experiment demonstrates stability of NESTANets by computing a worst-case perturbation of the measurements.

6.1 Setup

Our main example for the experiments is Fourier imaging in two dimensions. Here the ground truth image $\boldsymbol{x} \in \mathbb{C}^{N^2}$ and the measurement matrix

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{B} \end{bmatrix}, \qquad \boldsymbol{B} = \frac{1}{\sqrt{m}} \boldsymbol{P}_\Omega \boldsymbol{F},$$

is a subsampled Fourier matrix with $\boldsymbol{F} \in \mathbb{C}^{N^2 \times N^2}$. The sampling scheme $\Omega = \Omega_1 \cup \Omega_2$ conforms to the stacking scheme described in Theorem 5.3.2 (see also Section 3.3). That is, Ω_1 and Ω_2 are Bernoulli uniform and near-optimal variable sampling patterns (Section 3.2.3), respectively. By design, we can use NESTANets, i.e. restarted stacked NESTA, to solve the TV minimization problem

$$\min_{\boldsymbol{z} \in \mathbb{C}^{N^2}} \|\boldsymbol{z}\|_{\mathrm{TV}} \equiv \|\boldsymbol{V}\boldsymbol{z}\|_{\ell^1} \text{ subject to } \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{y}\|_{\ell^2} \leq \eta,$$

where $\boldsymbol{y}$ are the noisy measurements corresponding to Ω and $\boldsymbol{V} \in \mathbb{R}^{2N^2 \times N^2}$ is the 2-D discrete gradient operator (Section 2.3.4). In particular, $\boldsymbol{B}$ satisfies $\boldsymbol{B}\boldsymbol{B}^* = \frac{N^2}{m}\boldsymbol{I}$, since

$\boldsymbol{F}\boldsymbol{F}^* = N^2\boldsymbol{I}$ and $\boldsymbol{P}_\Omega$ is a row selector matrix. This is the required condition of $\boldsymbol{B}$ to use Algorithm 2, and in turn, Algorithm 5.

We implement numerical experiments for NESTANets (unrolled NESTA and restarted NESTA from Algorithms 2 and 5), with a stacking scheme for Fourier imaging via TV minimization. The implementation is a fork of the experiments in [75], and is written in Python using PyTorch [79]. PyTorch is an open-source machine learning Python package that offers a wide scope of tools to implement neural networks and manipulate arrays. PyTorch also supports GPU acceleration and automatic differentiation, the former of which enables scaling up the problem size and computing a fast Fourier transform, and the latter is crucial to running stability experiments for NESTANets. The code and its respective documentation can be found in

https://github.com/mneyrane/MSc-thesis-NESTAnets.

A key aspect to designing the experiments is choosing hyperparameters, for which there are several, and reducing the number of them if possible. The first hyperparameter pertains to the problem definition itself, namely the *sampling rate* m/N^d (i.e. the target number of measurements). Note that m defines the expected number of measurements produced by the uniform and variable density sampling patterns. The second set of hyperparameters define the solver. For restarted NESTA, these are the number of restarts t, inner iteration sequence $\{n_k\}$, smoothing parameter sequence $\{\mu_k\}$, and noise level η. Motivated by Theorems 4.2.2 and 4.3.2, $\{n_k\}$ and $\{\mu_k\}$ are defined in terms of the decay factor r, the special constant $\zeta \geq 0$ which we refer to as the *(target) error level*, and a constant $\delta > 0$. Referring to Theorem 4.3.2, we define $\delta = \frac{\sqrt{s}}{CdN^d}$ and the error level ζ corresponds to ζ in the theorem. Thus, δ determines the number of inner iterations $n_k = n = \left\lceil \frac{\sqrt{2}}{r\delta N^{d/2}} \right\rceil - 1$ and the smoothing parameters $\mu_k = r\delta\varepsilon_{k-1}$ where ε_{k-1} is defined in terms of r and ζ as in Theorem 4.3.2. Note that δ accounts for both the generally unknown constants s and C, the latter depending on the rNSP constants ρ and γ of the measurement matrix. In this way, it is unnecessary to treat s and C as two separate hyperparameters. Unless stated otherwise, we fix $r = \mathrm{e}^{-1}$ and $\zeta = 10^{-9}$, with further explanation of these choices provided in Section 6.3. The remaining hyperparameters are defined per experiment.

We use two test images which are shown in Fig. 6.1. The GLPU phantom [38] is suitable to test Fourier imaging techniques since it is a realistic image with a known analytic expression for its Fourier transform. In addition, it is piecewise constant and thus exactly sparse under the discrete gradient transform. The brain MR image was provided by the authors of [5]. The GLPU phantom is used all throughout except in the stability experiment, where we use the brain MR image.

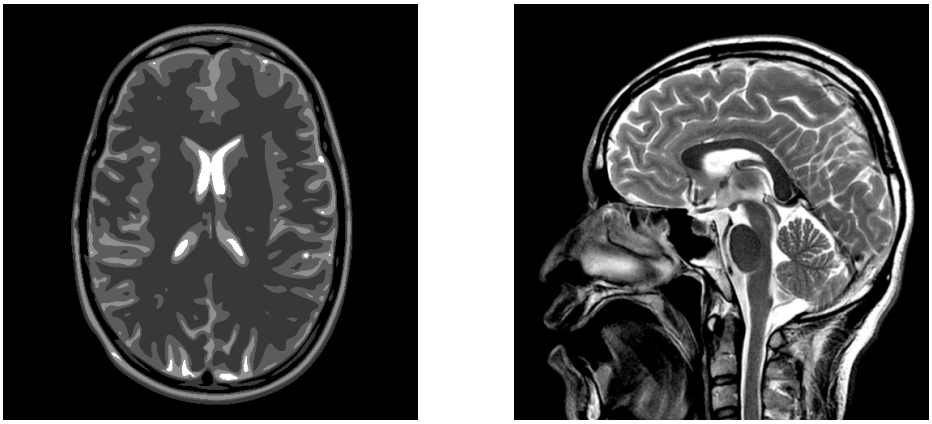


Figure 6.1: GLPU phantom (left) and brain MR image (right).

6.2 Restarted NESTA performance

6.2.1 Exponential decay of reconstruction error

Here we fix a 12.5% sampling rate, $t = 19$ restarts, $\delta = 2 \cdot 10^{-4}$, and $\eta = 10^{-i}$ for fixed $i = 1, 2, \ldots, 6$. Each restart iteration runs 38 inner iterations. A reminder that the measurements are computed as described in Section 3.3 yielding stacked measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$. Here $\boldsymbol{e}$ is drawn uniformly random from the surface of a ball of radius η, i.e. $\{\boldsymbol{e} : \|\boldsymbol{e}\|_{\ell^2} = \eta\}$. The restart iterate error $\|\boldsymbol{x}_k^\star - \boldsymbol{x}\|_{\ell^2}$ for each k is plotted in Fig. 6.2 (left). For each noise level η, the error decays exponentially to a limiting error proportional to η. This is consistent with error bounds established in Theorems 4.3.2 and 5.3.2. Since the GLPU phantom is piecewise constant, it is exactly sparse under the discrete gradient transform, so $\sigma_s(\boldsymbol{V}\boldsymbol{x})_{\ell^1} = 0$ for some suitable s (which is implicitly defined in δ). As we expect, the final reconstruction error is from the uncertainty in the measurements due to noise, corresponding to the noise level η.

6.2.2 Comparing NESTA with and without restarts

To compare NESTA with and without restarts, we evaluate the reconstruction error of the inner iterates from restarted NESTA and arrange the results by total number of iterations. We reuse the same parameters from the previous experiment, except $\eta = 10^{-7}$, $t = 199$. In the no-restart case, we fix smoothing parameters $\mu = 10^{-i}$, $i = 3, 4, 5, 6, 7$. Here we run NESTA for at least 5000 total iterations for both with and without restarts. Fig. 6.2 (right) plots the reconstruction error $\|\boldsymbol{x}_t - \boldsymbol{x}\|_{\ell^2}$ for each total iteration t. As observed in the previous experiment, the restart scheme reconstruction error exponentially decays to a limiting error level proportional to η. Without restarts, the convergence rate and limiting error level is sensitive to the smoothing parameter μ. Larger μ means the solver converges quickly albeit to a larger error level, whereas smaller μ leads to a lower error level at the cost of needing many more iterations. This is consistent with what we observe in the theory,

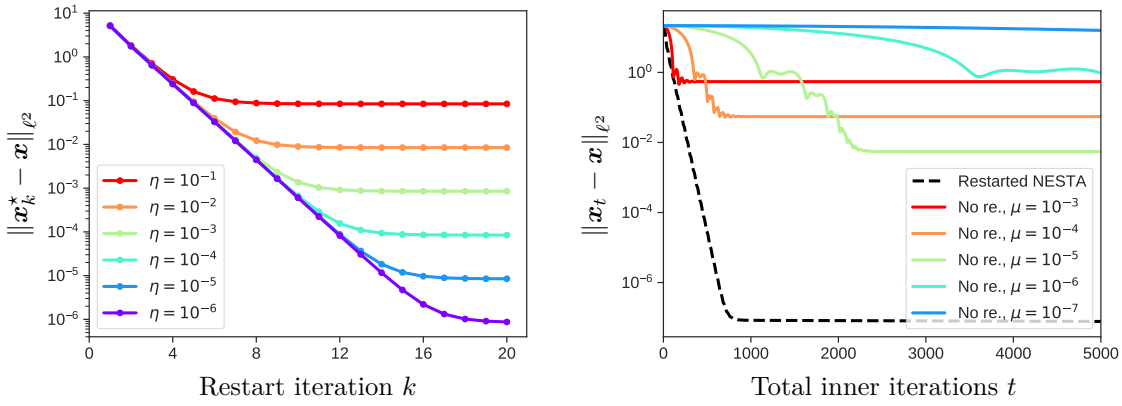


Figure 6.2: The left plot shows performance of restarted NESTA with different noise levels η, displaying exponential decay in the image error $\|\boldsymbol{x}_k^\star - \boldsymbol{x}\|_{\ell^2}$. The right plot compares NESTA with and without restarts for varying smoothing parameters μ.

namely the error bound in Theorem 4.2.2 for NESTA without restarts. As anticipated, the restart scheme excels in performance for problems in a low-noise regime.

Observe that the no-restart cases intuitively inform a restarting procedure, whereby we rerun NESTA reducing the smoothing parameter μ after some sufficient decrease in error. This decrease occurs rapidly when the initial guess is sufficiently close and a suitable new μ is chosen. The restart scheme effectively automates selecting the number of inner iterations and smoothing parameters via Theorem 4.3.2, using the values ζ, δ and r. The parameter ζ (and also η) control the extent of the limiting error. Here r controls the rate of convergence with a tradeoff in number of inner iterations. The number δ correctly scales the inner iterations and smoothing parameters, and is described in detail in the next section.

6.3 Hyperparameter selection

For this section, we offer extended discussion on how to select the restarted NESTA parameters t, r, ζ, η, and δ. These include general guidelines and observations that stay close to the theoretical results presented throughout this thesis. For ζ and δ, we show two respective experiments informing our selections.

The hyperparameter choices are broadly informed by Theorems 4.3.1 and 4.3.2. This presents a gap between theory and practice, whereby some of these parameter values are unknown a priori. For example, δ depends on rNSP/RIP constants which are not practical to compute [92][1] and ζ is unknown in the absence of the true signal being recovered.

[1]The authors prove that the decision problem of whether a matrix satisfies the NSP (Null Space Property) or RIP, with specific constants, are both NP-hard. We expect an analogous NP-hardness result to hold with the rNSP by adapting the arguments described in the paper.

To choose the number of restarts t, one can select minimal t so that the first error term of Theorem 4.3.2 is at most precision $\alpha > 0$, i.e. $r^{t+1}\|\boldsymbol{x}\|_{\ell^2} \leq \alpha$. In practice, only a domain-specific upper bound to $\|\boldsymbol{x}\|_{\ell^2}$ can be used when choosing α. Otherwise, choosing t turns into a matter of trial and error.

The decay factor r controls the rate of convergence and the limiting error bound, but is also inversely proportional to the number of inner iterations n_k. Effectively, to obtain faster convergence per restart and a lower error, we need to compute more inner iterations. To strike a balance between the two, an optimal choice of r can be derived when we fix the number of restarts and minimize the total number of inner iterations. Suppose $\alpha = r^{k+1}\|\boldsymbol{x}\|_{\ell^2}$, which corresponds to the first error term in Theorem 4.3.2 for some fixed $k \geq 0$. Then using $n = \left\lceil \frac{\sqrt{2}CdN^{d/2}}{r\sqrt{s}} \right\rceil - 1 \lesssim \frac{1}{r}$, the total number of iterations is bounded by

$$(k+1)(n+1) \lesssim \frac{1}{r\log(1/r)}.$$

Minimizing the upper bound with respect to $r > 0$ yields $r = \mathrm{e}^{-1}$. Observe that this optimal choice of r is independent of N, d, C and so on. This provides a sensible default value of r while avoiding arbitrary selection as resorted to in [75], wherein the same reasoning can be applied. With this, we chose $r = \mathrm{e}^{-1}$ for all experiments presented in this chapter.

Regarding the choice of ζ, experimentally we found it to be sufficient to choose $\zeta < \eta$. More generally, ζ can arbitrarily small but nonzero, so for instance one can choose machine epsilon. To showcase this, we run an experiment that uses the same parameters from the exponential decay experiment (Section 6.2.1), except the sampling rate is 25%, $t = 49$, $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, and $\eta, \zeta = 10^{-i}$ for $i = 0, 1, \ldots, 8$. A higher sampling rate was chosen to guarantee a high precision reconstruction for the lowest noise levels. In Fig. 6.3, we plot the reconstruction error of restarted NESTA's final iterate for different starting values of η and ζ. The contours are approximately of the form $\max\{\eta, \zeta/10\}$, which suggests that restarted NESTA cannot produce a reconstruction much better than the assumed noise level η or error level ζ. This further suggests that one can choose ζ to be less than the true error value without sacrificing accuracy. However, the theory alone (Theorems 4.3.1 and 4.3.2) does not conclude this, which instead assumes ζ to be an upper bound for the true error level. This assumption is leveraged to show exponential decay in the restart scheme's reconstruction error. Regardless, the experiment highlights that in practice we do not need to treat ζ as an upper bound. As already mentioned, ζ can be as small as possible, provided the computed smoothing parameters $\{\mu_k\}$ do not become zero (i.e. fall below machine precision).

For the choice of δ, there are a few comments to make. By definition, δ is proportional to μ_k and inversely proportional to n_k, suggesting a tradeoff between quality of reconstruction and number of inner iterations. In practice, it is difficult to determine an *optimal* choice of δ. The optimal choice is one where we avoid both being short of the best reconstruction and expending unnecessary calculations of the iterates. This tradeoff is showcased in the plots

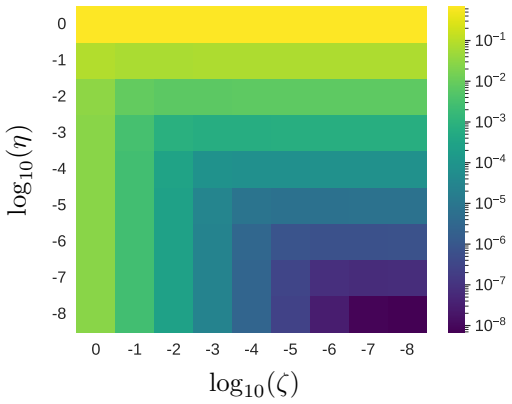


Figure 6.3: Contours of the error $\|\hat{\boldsymbol{x}}_{\eta,\zeta} - \boldsymbol{x}\|_{\ell^2}$, where $\hat{\boldsymbol{x}}_{\eta,\zeta}$ is the final iterate of restarted NESTA with given parameters η and ζ.

of Fig. 6.4. For this, the corresponding experiment matches the setup of Section 6.2.1, with 12.5% (left) and 25% (right) sampling rates, and share $t = 49$, $\eta = 10^{-5}$, and $\delta = 10^{-i}$ for $i = 3.40, 3.44, \ldots, 3.76$ (left) and $i = 3.20, 3.24, \ldots, 3.56$ (right). Focusing on the left plot, observe that for $\delta \leq 10^{-3.68}$, the restart scheme attains the lowest possible error level after about 15 restart iterations. Until then, there is a critical value $10^{-3.68} < \delta < 10^{-3.64}$ (i.e. $33 \leq n \leq 36$) where the reconstruction performance drastically deteriorates and the limiting error level becomes $\mathcal{O}(1)$ for larger values of δ. The same phenomenon is demonstrated numerically in [2, 3]. An analogous phenomenon holds for the right plot, where the critical δ differs and is instead $10^{-3.32} < \delta < 10^{-3.28}$ (i.e. $n = 15$). More broadly, the true value of δ is expected to depend on the sparsity s and the sampling mask, as this would yield different rNSP constants. To clearly see how δ relates to n and $\sqrt{s}/C$, see Fig. 6.5.

Lastly, considering a priori information to select δ is absent in a practical setting, one must resort to tuning δ. See Chapter 7 for further discussion.

6.4 Worst-case perturbations

Computing worst-case (or *adversarial*) perturbations is a standard empirical technique to verify the stability of a deep neural network. The concept originates in image classification [90], and has since become important in many other machine learning applications [101]. Computing adversarial perturbations in compressive imaging [7, 32, 36, 46] is a recent, yet important, development to study robustness of deep learning for inverse problems. For more information, see the references in Chapter 1.

To empirically verify stability, given stacked measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ we compute a worst-case perturbation [7] $\boldsymbol{e}$ of the measurements $\boldsymbol{y}$ that maximize the difference between $\mathcal{N}(\boldsymbol{y})$

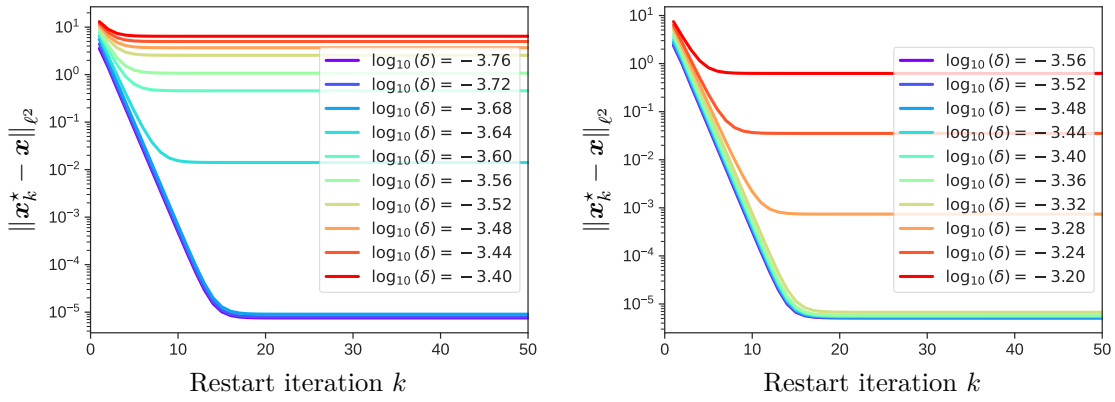


Figure 6.4: Performance of restarted NESTA with varying values of parameter δ. The corresponding problem sampling rates are 12.5% (left) and 25% (right).

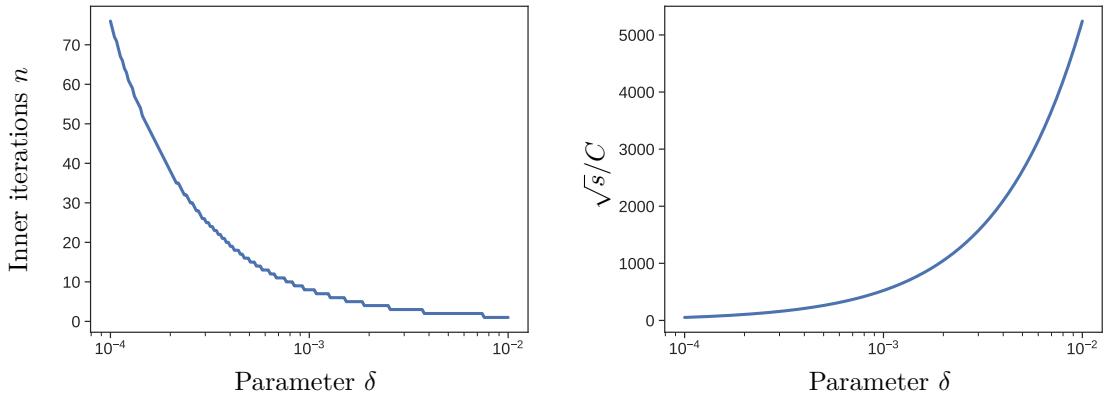


Figure 6.5: Relation of δ with inner iterations n (left) and ratio of unknown constants $\sqrt{s}/C$, assuming $d = 2$ and $N = 512$.

and $\mathcal{N}(\boldsymbol{y} + \boldsymbol{e})$, where $\mathcal{N} : \mathbb{C}^m \to \mathbb{C}^{N^2}$ is a NESTANet. More precisely, we try to solve

$$\max_{\boldsymbol{e}=(\boldsymbol{e}_1,\boldsymbol{e}_2)\in\mathbb{C}^m} \|\mathcal{N}(\boldsymbol{y}) - \mathcal{N}(\boldsymbol{y}+\boldsymbol{e})\|_{\ell^2}^2 \quad \text{subject to } \|\boldsymbol{e}\|_{\ell^2} \leq \tilde{\eta},\ \boldsymbol{e}_1 = \boldsymbol{e}_2, \tag{6.4.1}$$

analogous to the setup in [36, Sec. 3.4]. Note that $\boldsymbol{e}$ here is expressed as a stacked vector (see (4.1.5)), where abusing notation, $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ refer to its first and second block.

Here $\tilde{\eta}$ is a parameter controlling the maximum size permitted for the perturbation. To solve (6.4.1), we use projected gradient ascent for a fixed number of iterations, noting that the projection onto the feasible set of (6.4.1) is straightforward to compute. Over all the gradient ascent iterates we select the one that produced the largest objective value of (6.4.1). Note that due to a technical artifact of the stacking scheme described in Remark 4.1.3, an implicit constraint on $\boldsymbol{e}$ must be satisfied otherwise the solver update steps are undefined.

Specifically, we need $\|\boldsymbol{e}_1 - \boldsymbol{e}_2\|_{\ell^2} \leq \sqrt{2}\eta$ where η is the stacked NESTA noise level parameter. For simplicity, we instead enforce $\boldsymbol{e}_1 = \boldsymbol{e}_2$, noting that in practice one usually does not generate measurements for the same frequency more than once.

Observe that exactly solving (6.4.1) means determining the local $\tilde{\eta}$-Lipschitz constant of the network $\mathcal{N}$, and therefore if small, assert stability of $\mathcal{N}$ at the given $\boldsymbol{y}$. Unfortunately, due to the high-dimensional nonconvex nature of (6.4.1) we can at best approximate local optima. Also, the local optima one finds is now sensitive to the choice of gradient ascent's step size and initialization. For these reasons, numerical results indicative of stability fall short of being a conclusive verification of stability.

To best address the nonlinearity of (6.4.1), our experiment consists of running many independent trials of projected gradient ascent. We use an adaptive step size via PyTorch's `ReduceLROnPlateau` with an initial step size of 1 and initialize $\boldsymbol{e}$ uniformly random over the ℓ^2-ball of radius $\tilde{\eta}/\sqrt{m}$ centred at zero. The perturbation selected is one that maximizes the objective over all iterates in all trials belonging to a specific value of $\tilde{\eta}$.

Since automatic differentiation is used to compute the gradient of the objective in (6.4.1), the optimization procedure is computationally expensive for large numbers of restarts or inner iterations defining the NESTANet. This limits performing this experiment to smaller unrolled networks. Hence we use a much lower number of unrolled iterations here in comparison to the previous experiments.

Detailing the experiment, we use a 25% sampling rate, $K = 7$ restarts, $\delta = 5 \cdot 10^{-4}$, $\eta = 10^{-2}$ and $\tilde{\eta} = 10^i \eta$ with $i = 0, 1, 2, 3$. The worst-case perturbations $\boldsymbol{e}$ are computed for measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ where $\boldsymbol{x}$ is the brain MR image in Fig. 6.1. We perform 1000 trials of gradient ascent for each value of $\tilde{\eta}$, each consisting of 500 ascent iterations. For each value of $\tilde{\eta}$, we plot the worst-case perturbation maximizing (6.4.1) and the reconstruction difference of the perturbed measurements in Fig. 6.6. The perturbations themselves are represented in the image domain by applying the (right) Moore-Penrose pseudoinverse $\boldsymbol{B}^\dagger = \frac{m}{N^2}\boldsymbol{B}^*$ to the perturbation $\boldsymbol{e}_1 = \boldsymbol{e}_2$. The use of colour plots are to help visualize the perturbation and reconstruction differences, since for lower noise levels the visual artifacts are hard (or impossible) to see in a grayscale image.

What we observe is analogous to [75], where for $\tilde{\eta} = \eta$ we achieve the stability we expect from the theory, e.g. Theorem 4.2.2. In fact, even for larger perturbations beyond the assumed noise level ($\tilde{\eta} > \eta$), the algorithm remains stable. It may be possible to also theoretically justify this observation in our setting, e.g. by modifying [18]. The experiment suggests that NESTANets are also stable to perturbations exceeding the prescribed noise level. Interestingly, the worst-case perturbations computed for each $\tilde{\eta}$ tend to insert minor visual artifacts near the discontinues of the image, i.e. the boundaries of piecewise smooth components. A closeup of this is shown in Fig. 6.7 for $\tilde{\eta} = 10^3 \cdot \eta$, where visual artifacts appear along edges in the reconstruction of the perturbed measurements.

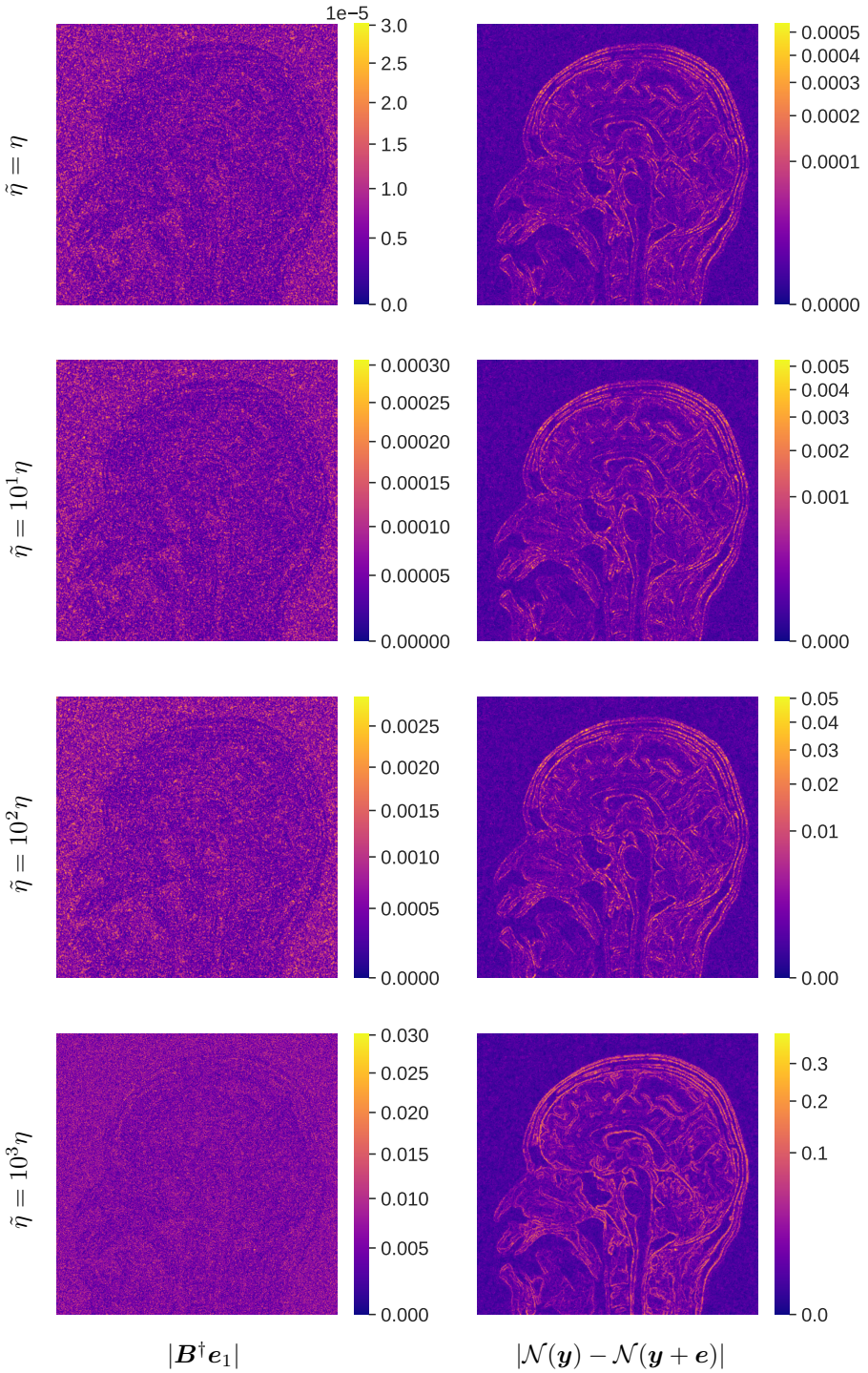
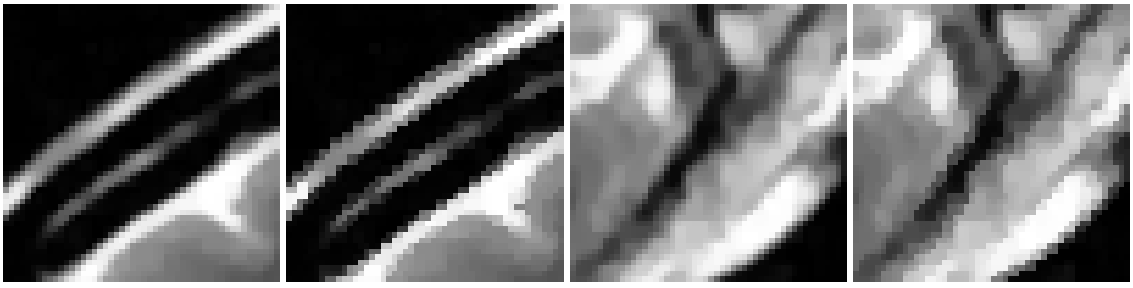


Figure 6.6: Colour plots of estimated worst-case perturbations $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2)$ in the image domain (left column) and the reconstruction differences (right column). The absolute value is applied elementwise. The constraint parameter for $\boldsymbol{e}$ is varied by row of plots, with $\tilde{\eta} = 10^i \eta$ with $\eta = 0.01$ and $i = 0, 1, 2, 3$. For ease of visualization, the plots in the left and right column use a power-law colourmap rescaling of $4/5$ and $2/5$, respectively.



$|\mathcal{N}(\boldsymbol{y})|$ $|\mathcal{N}(\boldsymbol{y}+\boldsymbol{e})|$ $|\mathcal{N}(\boldsymbol{y})|$ $|\mathcal{N}(\boldsymbol{y}+\boldsymbol{e})|$

Figure 6.7: Crops of $\mathcal{N}(\boldsymbol{y})$ and $\mathcal{N}(\boldsymbol{y}+\boldsymbol{e})$ for the computed worst-case perturbation $\boldsymbol{e}$ with $\tilde{\eta} = 10^3\eta$. The images are grayscale renders of clipped elementwise absolute values of the reconstructions.

Chapter 7

Conclusions and future work

To summarize, we presented the existence of stable, accurate and efficient neural networks, termed NESTANets, for Fourier imaging with a gradient-sparse model. We demonstrated theoretically that such networks can be constructed via a novel unrolling of the (stacked) NESTA optimization algorithm, and verified their properties empirically with several numerical experiments. There are several avenues of future research that may be of interest.

The proof techniques take inspiration from [30] and extend the results of [75]. Our main result shows that one can *construct* a neural network that matches the optimal accuracy and stability tradeoff from compressed sensing. We made no attempt to investigate whether such a network can be learned or if a network with better performance can be computed. This motivates two possible research directions. The first is to find nontrivial sufficient or necessary conditions for a network to *learn* the state-of-the-art performance of model-based methods. From the discussion in [37], we anticipate that such a learning procedure will be hybrid-based. On that note, the second direction would be to study recovery guarantees for hybrid-based techniques, ideally performing as well as state-of-the-art model-based methods.

To our knowledge, this work and [75] are the first instances of NESTA, and more generally Nesterov's method with smoothing, used as part of an unrolling scheme. Smoothing via the Moreau envelope is a standard tool in optimization that can be used for more general nonsmooth problems other than QCBP. It would be interesting to see smoothing used and adapted for other unrolling schemes.

The Bernoulli model for sampling has not been widely used and is not standard in compressed sensing. This may be due to the fact that the number of measurements is a random variable. Despite this, as reflected in our work, the Bernoulli model serves as a technical and practical benefit to compute the orthogonal projection of NESTA. Computing projections is not specific to NESTA, since any optimization algorithm that enforces feasibility will need to compute some kind of projection. It is worth noting that since we have computed the orthogonal projection for the stacked QCBP constraint, it can now be used in other projected optimization methods.

In terms of applications, it would be interesting to apply the Bernoulli model and stacking scheme to say, a more standard Fourier imaging application such as parallel MRI, where redundancy in sampled frequencies occur. It would also be interesting to see if there are other imaging modalities, apart from those in Fourier imaging, for which the stacking scheme can be applied.

As shown in this thesis, the network depth can be significantly reduced by using a restart scheme in the unrolling procedure. The efficiency of the restart scheme is guaranteed from the image reconstruction error analysis. However, a notable downside is that the recovery performance is sensitive to tuning the parameter δ. Its optimal value depends on constants unknown in practice, which vary with the sampling pattern and image structure (sparsity). The same issue also arises in [29, 30, 75]. We took this as motivation to investigate ways to avoid parameter tuning in restarts, while also maintaining efficiency. This led to [3], where we describe a restarting framework that performs a scheduled grid search on unknown parameters (expressible in terms of δ) while preserving the exponential decay in image reconstruction error. The framework also has the desirable property of omitting the error level ζ as a parameter. Moreover, our restarting framework applies to any first-order method used for any convex optimization problem. This is more general than restarting NESTA for QCBP. A future line of work would be to produce an unrolling of this generalized restart scheme.

Finally, some unexplored territory would be to extend these results to a variety of other settings. For instance, other gradient-sparse-like models (e.g. total generalized variation), measurement models (e.g. Walsh-Hadamard sampling, unknown noise levels), sampling patterns or other classes of analysis operators.

Bibliography

[1] B. Adcock, C. Boyer, and S. Brugiapaglia. On oracle-type local recovery guarantees in compressed sensing. *Inf. Inference*, 10(1):1–49, Mar. 2021.

[2] B. Adcock, S. Brugiapaglia, N. Dexter, and S. Moraga. On efficient algorithms for computing near-best polynomial approximations to high-dimensional, Hilbert-valued functions from limited samples, Mar. 2022. arXiv:2203.13908 [cs, math].

[3] B. Adcock, M. J. Colbrook, and M. Neyra-Nesterenko. Restarts subject to approximate sharpness: a parameter-free and optimal scheme for first-order methods, Jan. 2023. arXiv:2301.02268 [cs, math].

[4] B. Adcock, N. Dexter, and Q. Xu. Improved recovery guarantees and sampling strategies for TV minimization in compressive imaging. *SIAM J. Imaging Sci.*, 14(3):1149–1183, Jan. 2021.

[5] B. Adcock and A. C. Hansen. *Compressive Imaging: Structure, Sampling, Learning.* Cambridge University Press, 1st edition, Sept. 2021.

[6] R. Alaifari, G. S. Alberti, and T. Gauksson. Localized adversarial artifacts for compressed sensing MRI, June 2022. arXiv:2206.05289 [cs, eess].

[7] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen. On instabilities of deep learning in image reconstruction and the potential costs of AI. *Proc. Natl. Acad. Sci. USA*, 117(48):30088–30095, Dec. 2020.

[8] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb. Solving inverse problems using data-driven models. *Acta Numer.*, 28:1–174, May 2019.

[9] A. Bastounis, A. C. Hansen, and V. Vlačić. The mathematics of adversarial attacks in AI – why deep learning is unstable despite the existence of stable neural networks, Sept. 2021. arXiv:2109.06098 [cs, math, stat].

[10] A. Bastounis, A. C. Hansen, and V. Vlačić. The extended Smale's 9th problem – on computational barriers and paradoxes in estimation, regularisation, computer-assisted proofs and learning, Aug. 2022. arXiv:2110.15734 [math].

[11] A. Beck. *First-Order Methods in Optimization.* Society for Industrial and Applied Mathematics, Philadelphia, PA, Oct. 2017.

[12] A. Beck and M. Teboulle. Smoothing and first order methods: a unified framework. *SIAM J. Optim.*, 22(2):557–580, Jan. 2012.

[13] S. Becker, J. Bobin, and E. J. Candès. NESTA: a fast and accurate first-order method for sparse recovery. *SIAM J. Imaging Sci.*, 4(1):1–39, Jan. 2011.

[14] S. R. Becker. *Practical Compressed Sensing: modern data acquisition and signal processing*. PhD, California Institute of Technology, June 2011.

[15] C. Belthangady and L. A. Royer. Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction. *Nat. Methods*, 16(12):1215–1225, Dec. 2019.

[16] H. Ben Yedder, B. Cardoen, and G. Hamarneh. Deep learning for biomedical image reconstruction: a survey. *Artif. Intell. Rev.*, 54(1):215–251, Jan. 2021.

[17] S. Bhadra, V. A. Kelkar, F. J. Brooks, and M. A. Anastasio. On hallucinations in tomographic image reconstruction. *IEEE Trans. Med. Imag.*, 40(11):3249–3260, Nov. 2021.

[18] S. Brugiapaglia and B. Adcock. Robustness to unknown error in sparse regularization. *IEEE Trans. Inform. Theory*, 64(10):6638–6661, Oct. 2018.

[19] S. Bubeck and others. Convex optimization: algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3-4):231–357, 2015. Publisher: Now Publishers, Inc.

[20] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52(2):489–509, Feb. 2006.

[21] E. J. Candès and D. L. Donoho. Curvelets—a surprisingly effective nonadaptive representation for objects with edges. In C. Rabut, A. Cohen, and L. L. Schumaker, editors, *Curves and Surfaces*, pages 105–120. Vanderbilt University Press, Nashville, TN, 2000.

[22] E. J. Candès and D. L. Donoho. Recovering edges in ill-posed inverse problems: optimality of curvelet frames. *Ann. Statist.*, 30(3):784–842, 2002.

[23] E. J. Candès and D. L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. *Comm. Pure Appl. Math.*, 57(2):219–266, 2004.

[24] E. J. Candès and Y. Plan. A probabilistic and RIPless theory of compressed sensing. *IEEE Trans. Inform. Theory*, 57(11):7235–7254, Nov. 2011.

[25] A. Chambolle, M. Novaga, D. Cremers, T. Pock, and V. Caselles. An Introduction to total variation for image analysis. In M. Fornasier, editor, *Theoretical Foundations and Numerical Methods for Sparse Recovery*, volume 9 of *Radon Series in Computational and Applied Mathematics*, pages 263–340. de Gruyter, Berlin, 2010.

[26] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision*, 40(1):120–145, May 2011.

[27] A. Chambolle and T. Pock. An introduction to continuous optimization for imaging. *Acta Numer.*, 25:161–319, May 2016.

[28] A. Chambolle and T. Pock. On the ergodic convergence rates of a first-order primal–dual algorithm. *Math. Program.*, 159(1-2):253–287, Sept. 2016.

[29] M. J. Colbrook. WARPd: a linearly convergent first-order primal-dual algorithm for inverse problems with approximate sharpness conditions. *SIAM J. Imaging Sci.*, 15(3):1539–1575, Sept. 2022.

[30] M. J. Colbrook, V. Antun, and A. C. Hansen. The difficulty of computing stable and accurate neural networks: on the barriers of deep learning and Smale's 18th problem. *Proc. Natl. Acad. Sci. USA*, 119(12):e2107151119, Mar. 2022.

[31] P. L. Combettes and J.-C. Pesquet. Lipschitz certificates for layered network structures driven by averaged activation operators. *SIAM J. Math. Data Sci.*, 2(2):529–557, Jan. 2020.

[32] M. Z. Darestani, A. S. Chaudhari, and R. Heckel. Measuring robustness in deep learning based compressive sensing. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2433–2444. PMLR, July 2021.

[33] I. Daubechies, B. Han, A. Ron, and Z. Shen. Framelets: MRA-based constructions of wavelet frames. *Appl. Comput. Harmon. Anal.*, 14(1):1–46, Jan. 2003.

[34] M. Elad, P. Milanfar, and R. Rubinstein. Analysis versus synthesis in signal priors. *Inverse Problems*, 23(3):947–968, June 2007.

[35] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Applied and Numerical Harmonic Analysis. Birkhäuser, New York, 2013. OCLC: ocn860992971.

[36] M. Genzel, J. Macdonald, and M. Marz. Solving inverse problems with deep neural networks – robustness included? *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(1):1119–1134, Jan. 2023.

[37] N. M. Gottschling, V. Antun, A. C. Hansen, and B. Adcock. The troublesome kernel – on hallucinations, no free lunches and the accuracy-stability trade-off in inverse problems, Jan. 2023. arXiv:2001.01258 [cs].

[38] M. Guerquin-Kern, L. Lejeune, K. P. Pruessmann, and M. Unser. Realistic analytical phantoms for parallel magnetic resonance imaging. *IEEE Trans. Med. Imag.*, 31(3):626–636, Mar. 2012.

[39] K. Guo, G. Kutyniok, and D. Labate. Sparse multidimensional representations using anisotropic dilation and shear operators. In G. Chen and M.-J. Lai, editors, *Wavelets and Splines: Athens 2005*, pages 189–201. Nashboro Press, Brentwood, TN, 2006.

[40] K. Guo and D. Labate. Optimally sparse multidimensional representation using shearlets. *SIAM J. Math. Anal.*, 39(1):298–318, 2007.

[41] N. P. Hardy, P. Mac Aonghusa, P. M. Neary, and R. A. Cahill. Intraprocedural artificial intelligence for colorectal cancer detection and characterisation in endoscopy and laparoscopy. *Surg. Innov.*, 28(6):768–775, Dec. 2021.

[42] M. Hasannasab, J. Hertrich, S. Neumayer, G. Plonka, S. Setzer, and G. Steidl. Parseval proximal neural networks. *J. Fourier Anal. Appl.*, 26(4):59, Aug. 2020.

[43] E. H. Herskovits. Artificial intelligence in molecular imaging. *Ann. Transl. Med.*, 9(9):824–824, May 2021.

[44] D. P. Hoffman, I. Slavitt, and C. A. Fitzpatrick. The promise and peril of deep learning in microscopy. *Nat. Methods*, 18(2):131–132, Feb. 2021.

[45] D. J. Holland, M. J. Bostock, L. F. Gladden, and D. Nietlispach. Fast multidimensional NMR spectroscopy using compressed sensing. *Angew. Chem. Int. Ed.*, 50(29):6548–6551, July 2011.

[46] Y. Huang, T. Würfl, K. Breininger, L. Liu, G. Lauritsch, and A. Maier. Some investigations on robustness of deep learning in limited angle tomography. In A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, volume 11070, pages 145–153. Springer International Publishing, Cham, 2018. Series Title: Lecture Notes in Computer Science.

[47] A. Jalal, L. Liu, A. G. Dimakis, and C. Caramanis. Robust compressed sensing using generative models. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 713–727. Curran Associates, Inc., 2020.

[48] P. M. Johnson, G. Jeong, K. Hammernik, J. Schlemper, C. Qin, J. Duan, D. Rueckert, J. Lee, N. Pezzotti, E. De Weerdt, S. Yousefi, M. S. Elmahdy, J. H. F. Van Gemert, C. Schülke, M. Doneva, T. Nielsen, S. Kastryulin, B. P. F. Lelieveldt, M. J. P. Van Osch, M. Staring, E. Z. Chen, P. Wang, X. Chen, T. Chen, V. M. Patel, S. Sun, H. Shin, Y. Jun, T. Eo, S. Kim, T. Kim, D. Hwang, P. Putzky, D. Karkalousos, J. Teuwen, N. Miriakov, B. Bakker, M. Caan, M. Welling, M. J. Muckley, and F. Knoll. Evaluation of the robustness of learned MR image reconstruction to systematic deviations between training and test data for the models from the fastMRI challenge. In N. Haq, P. Johnson, A. Maier, T. Würfl, and J. Yoo, editors, *Machine Learning for Medical Image Reconstruction*, volume 12964, pages 25–34. Springer International Publishing, Cham, 2021. Series Title: Lecture Notes in Computer Science.

[49] A. Jones, A. Tamtögl, I. Calvo-Almazán, and A. Hansen. Continuous compressed sensing for surface dynamical processes with helium atom scattering. *Sci. Rep.*, 6(1):27776, June 2016.

[50] K. Kazimierczuk and V. Y. Orekhov. Accelerated NMR spectroscopy by using compressed sensing. *Angew. Chem. Int. Ed.*, 50(24):5556–5559, June 2011.

[51] F. Knoll, K. Hammernik, C. Zhang, S. Moeller, T. Pock, D. K. Sodickson, and M. Akcakaya. Deep-learning methods for parallel magnetic resonance imaging reconstruction: a survey of the current approaches, trends, and issues. *IEEE Signal Process. Mag.*, 37(1):128–140, Jan. 2020.

[52] F. Knoll, T. Murrell, A. Sriram, N. Yakubova, J. Zbontar, M. Rabbat, A. Defazio, M. J. Muckley, D. K. Sodickson, C. L. Zitnick, and M. P. Recht. Advancing machine

learning for MR image reconstruction with an open competition: Overview of the 2019 fastMRI challenge. *Magn. Reson. Med.*, 84(6):3054–3070, Dec. 2020.

[53] F. Krahmer, C. Kruschel, and M. Sandbichler. Total variation minimization in compressed sensing. In H. Boche, G. Caire, R. Calderbank, M. März, G. Kutyniok, and R. Mathar, editors, *Compressed Sensing and its Applications*, pages 333–358. Springer International Publishing, Cham, 2017. Series Title: Applied and Numerical Harmonic Analysis.

[54] F. Krahmer and R. Ward. Stable and robust sampling strategies for compressive imaging. *IEEE Trans. Image Process.*, 23(2):612–622, Feb. 2014.

[55] G. Kutyniok and W.-Q. Lim. Compactly supported shearlets are optimally sparse. *J. Approx. Theory*, 163(11):1564–1589, Nov. 2011.

[56] D. Labate, W.-Q. Lim, G. Kutyniok, and G. Weiss. Sparse multidimensional representation using shearlets. In M. Papadakis, A. F. Laine, and M. A. Unser, editors, *Wavelets XI*, volume 5914, pages 254–262, Bellingham, WA, 2005. SPIE. Backup Publisher: International Society for Optics and Photonics.

[57] R. F. Laine, I. Arganda-Carreras, R. Henriques, and G. Jacquemet. Avoiding a replication crisis in deep-learning-based bioimage analysis. *Nat. Methods*, 18(10):1136–1144, Oct. 2021.

[58] D. B. Larson, H. Harvey, D. L. Rubin, N. Irani, J. R. Tse, and C. P. Langlotz. Regulatory frameworks for development and evaluation of artificial intelligence-based diagnostic imaging algorithms: summary and recommendations. *J. Am. Coll. Radiol.*, 18(3):413–424, Mar. 2021.

[59] J. Leuschner, M. Schmidt, P. S. Ganguly, V. Andriiashen, S. B. Coban, A. Denker, D. Bauer, A. Hadjifaradji, K. J. Batenburg, P. Maass, and M. van Eijnatten. Quantitative comparison of deep learning-based image reconstruction methods for low-dose and sparse-angle CT applications. *J. Imaging*, 7(3):44, Mar. 2021.

[60] D. Liang, J. Cheng, Z. Ke, and L. Ying. Deep magnetic resonance image reconstruction: inverse problems meet neural networks. *IEEE Signal Process. Mag.*, 37(1):141–151, Jan. 2020.

[61] X. Liu, B. Glocker, M. M. McCradden, M. Ghassemi, A. K. Denniston, and L. Oakden-Rayner. The medical algorithmic audit. *The Lancet Digital Health*, 4(5):e384–e397, May 2022.

[62] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos. Using deep neural networks for inverse problems in imaging: beyond analytical methods. *IEEE Signal Process. Mag.*, 35(1):20–36, Jan. 2018.

[63] A. S. Lundervold and A. Lundervold. An overview of deep learning in medical imaging focusing on MRI. *Z. Med. Phys.*, 29(2):102–127, May 2019.

[64] M. Lustig, D. Donoho, and J. M. Pauly. Sparse MRI: the application of compressed sensing for rapid MR imaging. *Magn. Reson. Med.*, 58(6):1182–1195, Dec. 2007.

[65] M. T. McCann, K. H. Jin, and M. Unser. Convolutional neural networks for inverse problems in imaging: a review. *IEEE Signal Process. Mag.*, 34(6):85–95, Nov. 2017.

[66] M. T. McCann and M. Unser. Biomedical image reconstruction: from the foundations to deep neural networks. *Found. Trends Signal Process.*, 13(3):283–357, 2019.

[67] V. Monga, Y. Li, and Y. C. Eldar. Algorithm unrolling: interpretable, efficient deep learning for signal and image processing. *IEEE Signal Process. Mag.*, 38(2):18–44, Mar. 2021.

[68] J. N. Morshuis, S. Gatidis, M. Hein, and C. F. Baumgartner. Adversarial robustness of MR image reconstruction under realistic perturbations. In N. Haq, P. Johnson, A. Maier, C. Qin, T. Würfl, and J. Yoo, editors, *Machine Learning for Medical Image Reconstruction*, volume 13587, pages 24–33. Springer International Publishing, Cham, 2022. Series Title: Lecture Notes in Computer Science.

[69] M. J. Muckley, B. Riemenschneider, A. Radmanesh, S. Kim, G. Jeong, J. Ko, Y. Jun, H. Shin, D. Hwang, M. Mostapha, S. Arberet, D. Nickel, Z. Ramzi, P. Ciuciu, J.-L. Starck, J. Teuwen, D. Karkalousos, C. Zhang, A. Sriram, Z. Huang, N. Yakubova, Y. W. Lui, and F. Knoll. Results of the 2020 fastMRI challenge for machine learning MR image reconstruction. *IEEE Trans. Med. Imag.*, 40(9):2306–2317, Sept. 2021.

[70] S. Nam, M. Davies, M. Elad, and R. Gribonval. The cosparse analysis model and algorithms. *Appl. Comput. Harmon. Anal.*, 34(1):30–56, Jan. 2013.

[71] D. Needell and R. Ward. Near-optimal compressed sensing guarantees for total variation minimization. *IEEE Trans. Image Process.*, 22(10):3941–3949, Oct. 2013.

[72] D. Needell and R. Ward. Stable image reconstruction using total variation minimization. *SIAM J. Imaging Sci.*, 6(2):1035–1058, Jan. 2013.

[73] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, May 2005.

[74] Y. Nesterov. *Lectures on Convex Optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer International Publishing, Cham, 2018.

[75] M. Neyra-Nesterenko and B. Adcock. NESTANets: stable, accurate and efficient neural networks for analysis-sparse inverse problems. *Sampl. Theory Signal Process. Data Anal.*, 21(1):4, June 2023.

[76] S. Noordman. Current Issues in Deep Learning for Undersampled Image Reconstruction. Master's thesis, Utrecht University, Dec. 2021.

[77] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett. Deep learning techniques for inverse problems in imaging. *IEEE J. Sel. Areas Inf. Theory*, 1(1):39–56, May 2020.

[78] A. Pal and Y. Rathi. A review and experimental evaluation of deep learning methods for MRI reconstruction. *Mach. Learn. Biomed. Imag.*, 1(March 2022 issue), 2022.

[79] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: an imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[80] C. Poon. On the role of total variation in compressed sensing. *SIAM J. Imaging Sci.*, 8(1):682–720, Jan. 2015.

[81] S. Ravishankar, J. C. Ye, and J. A. Fessler. Image reconstruction: from sparsity to data-adaptive methods and machine learning. *Proc. IEEE*, 108(1):86–109, Jan. 2020.

[82] A. J. Reader, G. Corda, A. Mehranian, C. d. Costa-Luis, S. Ellis, and J. A. Schnabel. Deep learning for PET image reconstruction. *IEEE Trans. Radiat. Plasma Med. Sci.*, 5(1):1–25, Jan. 2021.

[83] J. Renegar and B. Grimmer. A simple nearly optimal restart scheme for speeding up first-order methods. *Found. Comput. Math.*, 22(1):211–256, Feb. 2022.

[84] J. Romberg. Compressive sensing by random convolution. *SIAM J. Imaging Sci.*, 2(4):1098–1128, Jan. 2009.

[85] V. Roulet and A. d'Aspremont. Sharpness, restart, and acceleration. *SIAM J. Optim.*, 30(1):262–289, Jan. 2020.

[86] C. M. Sandino, J. Y. Cheng, F. Chen, M. Mardani, J. M. Pauly, and S. S. Vasanawala. Compressed sensing: from research to clinical practice with deep neural networks: shortening scan times for magnetic resonance imaging. *IEEE Signal Process. Mag.*, 37(1):117–127, Jan. 2020.

[87] E. Shimron, J. I. Tamir, K. Wang, and M. Lustig. Implicit data crimes: machine learning bias arising from misuse of public data. *Proc. Natl. Acad. Sci. USA*, 119(13):e2117203119, Mar. 2022.

[88] E. Y. Sidky, I. Lorente, J. G. Brankov, and X. Pan. Do CNNs solve the CT inverse problem? *IEEE Trans. Biomed. Eng.*, 68(6):1799–1810, June 2021.

[89] V. Stumpo, J. M. Kernbach, C. H. B. van Niftrik, M. Sebök, J. Fierstra, L. Regli, C. Serra, and V. E. Staartjes. Machine learning algorithms in neuroimaging: an overview. In V. E. Staartjes, L. Regli, and C. Serra, editors, *Machine Learning in Clinical Neuroscience*, volume 134, pages 125–138. Springer International Publishing, Cham, 2022. Series Title: Acta Neurochirurgica Supplement.

[90] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations*, 2014. arXiv:1312.6199 [cs].

[91] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht. Compressed sensing off the grid. *IEEE Trans. Inform. Theory*, 59(11):7465–7490, Nov. 2013.

[92] A. M. Tillmann and M. E. Pfetsch. The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing. *IEEE Trans. Inform. Theory*, 60(2):1248–1259, Feb. 2014.

[93] M. Torres-Velazquez, W.-J. Chen, X. Li, and A. B. McMillan. Application and construction of deep learning networks in medical imaging. *IEEE Trans. Radiat. Plasma Med. Sci.*, 5(2):137–159, Mar. 2021.

[94] M. Tölle, M.-H. Laves, and A. Schlaefer. A mean-field variational inference approach to deep image prior for inverse problems in medical imaging. In M. Heinrich, Q. Dou, M. de Bruijne, J. Lellmann, A. Schläfer, and F. Ernst, editors, *Proceedings of the Fourth Conference on Medical Imaging with Deep Learning*, volume 143 of *Proceedings of Machine Learning Research*, pages 745–760. PMLR, July 2021.

[95] R. Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Number 47 in Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge ; New York, NY, 2018.

[96] G. Wang, J. C. Ye, and B. De Man. Deep learning for tomographic image reconstruction. *Nat. Mach. Intell.*, 2(12):737–748, Dec. 2020.

[97] G. Wang, J. C. Ye, K. Mueller, and J. A. Fessler. Image reconstruction is a new frontier of machine learning. *IEEE Trans. Med. Imag.*, 37(6):1289–1296, June 2018.

[98] Y. Wiaux, L. Jacques, G. Puy, A. M. M. Scaife, and P. Vandergheynst. Compressed sensing imaging techniques for radio interferometry. *Mon. Not. R. Astron. Soc.*, 395(3):1733–1742, May 2009.

[99] C. Zhang, J. Jia, B. Yaman, S. Moeller, S. Liu, M. Hong, and M. Akcakaya. Instabilities in conventional multi-coil MRI reconstruction with small adversarial perturbations. In *2021 55th Asilomar Conference on Signals, Systems, and Computers*, pages 895–899, Pacific Grove, CA, USA, Oct. 2021. IEEE.

[100] H.-M. Zhang and B. Dong. A review on deep learning in medical image reconstruction. *J. Oper. Res. Soc. China*, 8(2):311–340, June 2020.

[101] J. Zhang and C. Li. Adversarial examples: opportunities and challenges. *IEEE Trans. Neural Netw. Learn. Syst.*, pages 1–16, 2019.

Appendix A

Notation and abbreviations

General notation

$\mathbb{R}$, $\mathbb{C}$	real and complex numbers
$\mathbb{R}^n$, $\mathbb{C}^n$	n-dimensional real and complex vector spaces
$[\![M]\!]$	set of positive integers $\{1, \dots, M\}$
$\boldsymbol{A}, \boldsymbol{B}, \dots$	matrices (denoted by boldface uppercase characters)
$\boldsymbol{a}, \boldsymbol{b}, \dots$	vectors (denoted by boldface lowercase characters)
$\boldsymbol{A}^*, \boldsymbol{a}^*$	adjoint of a matrix or vector
$\boldsymbol{A}^\top, \boldsymbol{a}^\top$	transpose of a matrix or vector
$\overline{\boldsymbol{A}}, \bar{\boldsymbol{a}}$	complex conjugate of a matrix or vector
$a, A, \alpha, \dots$	numbers or sets (denoted by regular characters)
$\mathbb{E}$	expected value
$\mathbb{P}$	probability measure
ℓ^p	p-norm
$\|\cdot\|_{\ell^p}$	vector p-norm of $\mathbb{R}^n$ or $\mathbb{C}^n$
$\langle \cdot, \cdot \rangle$	2-norm inner product of $\mathbb{R}^n$ or $\mathbb{C}^n$
$\lvert \cdot \rvert$	absolute value of a number, or cardinality of a set
$\otimes$	Kronecker product
$\odot$	elementwise multiplication
$\lesssim$, $\lesssim_d$	less than or equal to, by a constant factor (depending on d)
$\gtrsim$, $\gtrsim_d$	greater than or equal to, by a constant factor (depending on d)
$\asymp$	when $\lesssim$ and $\gtrsim$ both hold
$\propto$	proportional to

Compressed sensing and inverse problems

m	number of measurements
s	sparsity, i.e. number of nonzero entries
N	image/signal dimension

η	measurement noise level
$\boldsymbol{A}$	measurement matrix
$\boldsymbol{W}$	analysis matrix
$\boldsymbol{x}$	image/signal vector
$\boldsymbol{y}$	measurement vector
$\boldsymbol{e}$	noise vector
S	finite subset of positive integers
$S^{\complement}$	set complement of S
$\boldsymbol{z}_S$	vector formed by the entries of $\boldsymbol{z}$ indexed by S
$\sigma_s(\boldsymbol{z})_{\ell^1}$	best s-term approximation error
ρ, γ	robust Null Space Property (rNSP) constants
δ_s	sth Restricted Isometry Constant (RIC)
δ	Restricted Isometry Property (RIP) parameter
$\mathcal{A}, \mathcal{A}_i$	family of random vectors
$\mathcal{C}$	collection of families of random vectors
$\mu(\mathcal{A})$, $\mu(\mathcal{C})$	coherence of a family or collection, respectively

Fourier and gradient-sparse imaging

d	image dimension
N^d	image size
$\|\cdot\|_{\mathrm{TV}}$	anisotropic total variation (TV) semi-norm
$\boldsymbol{V}$, $\boldsymbol{V}^{(d)}$	anisotropic discrete gradient operator
$\boldsymbol{V}_i$	discrete ith partial derivative operator
$\boldsymbol{F}$, $\boldsymbol{F}^{(d)}$	Fourier matrix (i.e. discrete Fourier transform)
$\boldsymbol{W}$, $\boldsymbol{W}^{(d)}$	discrete orthonormal Haar wavelet transform
$\boldsymbol{U}$	unitary matrix
$\boldsymbol{I}$	identity matrix
Ω	set of positive integer indices
$\boldsymbol{P}_\Omega$	row selection matrix based on indices Ω
$\triangle$	symmetric different set operator
$\boldsymbol{e}_i$	ith standard basis vector
ω	Fourier domain frequency indices
ς	lexicographical ordering of d-dimensional indices
ϱ, $\varrho^{(d)}$	arrangement of frequencies for (d-dimensional) Fourier matrix
$\boldsymbol{p}$	Bernoulli vector
$\hat{\boldsymbol{p}}$	near-optimal Bernoulli vector
$\mathrm{Ber}(\llbracket N \rrbracket, m)$	Bernoulli uniform sampling scheme of order m
$\mathrm{Ber}(\llbracket N \rrbracket, m, \boldsymbol{p})$	Bernoulli variable density sampling scheme of order m
p_ω, p_i	probability of sampling frequency ω (with index i)
q_ω	weights for variable density sampling
$\Gamma(\boldsymbol{p})$	constant factor for Fourier-Haar coherence bound

Convex optimization

f, F, g, G, ϕ	functions
∇f	gradient of f
K	Lipschitz constant of gradient, i.e. K-smoothness
μ	smoothing parameter
f_μ	$\frac{1}{\mu}$ smooth-approximation of function f with parameter μ
$\mathcal{M}_f^\mu$	Moreau envelope of f with parameter μ
$\|\cdot\|_{\ell^1,\mu}$	smoothed ℓ^1-norm with parameter μ
$\mathcal{T}_\mu$	gradient of smoothed ℓ^1-norm with parameter μ
H_μ	Huber function with parameter μ
ρ, λ	KKT multipliers
α_n, τ_n	parameter sequences in NESTA and Nesterov's method
σ_p, σ_d	strong convexity constants for Nesterov's method
p_p	primal prox-function for Nesterov's method
p_d	dual prox-function for Nesterov's method
Q	constraint set for Nesterov's method
$\boldsymbol{z}'$, $\boldsymbol{Z}'$	real equivalent of a complex vector and matrix

Neural networks

$\mathcal{N}^*$, $\mathcal{N}^*_{\boldsymbol{n},L,q}$	class of neural networks
$\mathcal{N}$	neural network
L	number of layers
$\boldsymbol{n}$	vector of neural network layer sizes
q	number of nonlinear activation functions

Imaging recovery guarantees

χ	bounding parameter for compressed sensing error
ζ	error level parameter
μ_k, n_k	restart scheme smoothing and inner iteration parameters
$\boldsymbol{x}_k^\star$	kth restart iterate
δ	smoothing-iteration tradeoff parameter
r	restart scheme decay factor
$\mathcal{CS}_{s,d}(\boldsymbol{z}, \boldsymbol{p}, \eta)$	compressed sensing error of $\boldsymbol{z}$
$\mathbb{I}$, $\mathbb{I}_{\boldsymbol{V},\chi,\eta}$	gradient-sparse image model class
$\mathbb{M}$, $\mathbb{M}_{\boldsymbol{A},\boldsymbol{V},\chi,\eta}$	measurement model class of gradient-sparse images
$\text{polylog}(\dots)$	polynomial of logarithmic terms
$\mathcal{O}(\cdot), \mathcal{O}_d(\cdot)$	big O-notation (with constant factor depending on d)

Abbreviatons

KKT	Karush-Kuhn-Tucker
MRI	Magnetic Resonance Imaging
NESTA	NESTerov's Algorithm for QCBP
QCBP	Quadratically-Constrained Basis Pursuit
RIC	Restricted Isometry Constant
RIP	Restricted Isometry Property
rNSP	robust Null Space Property
TV	Total Variation