UiO : University of Oslo

Vegard Antun

Stability and accuracy in compressive sensing and deep learning

Thesis submitted for the degree of Philosophiae Doctor

Department of Mathematics
Faculty of Mathematics and Natural Sciences

2020

Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo
No. 2354

ISSN 1501-7710

Cover: Hanne Baadsgaard Utigard.
Print production: Reprosentralen, University of Oslo.

Preface

This thesis is submitted in partial fulfilment of the requirements for the degree of Philosophiae Doctor at the University of Oslo. The research presented here has been advised by Senior Lecturer Øyvind Ryan and Professor Anders C. Hansen. The work with this thesis was initiated in the autumn of 2016 and completed in the first quarter of 2020. It is a collection of five papers and an introductory chapter.

Acknowledgements

I want to start by thanking my two advisors Øyvind Ryan and Anders Hansen. Their doors have always been open for discussions, and their advice and support during the work with this thesis have been pivotal.

In this respect, I also need to mention Ben Adcock, who in large, has worked as a third supervisor without formally having that role. During the work with this thesis, I have not only been working with Ben and Anders on research, but I have also been making reproducible code for most of the figures in their upcoming book on compressive imaging. The discussions made during the work with these figures and the book, in general, have been enlightening and set the tone for much of our work together.

I also want to thank my fellow co-authors Nina M. Gottschling, Clarice Poon, Francesco Renna and Laura Thesing for many fruitful discussions and (I would say) successful collaborations. I am also grateful to the Department of Mathematics for providing excellent working facilities. During the work with this thesis, it became apparent that the department needed hardware appropriate for deep learning research. I am thankful to all who saw the need for investing in new computing facilities. These investments have certainly contributed to the results in this thesis, and I know it will contribute to other's work at the department too. In this regard, I need to thank Terje Kvernes for always keeping our deep learning software up-to-date and for many cheerful conversations.

I want to thank all the people on the tenth floor for the lunches we have had. In particular, I want to thank Trygve Bærland, Lorenzo Ciardo, André H. Erhardt, Luca Galimberti, Torkel Haufmann, Alise D. Midtfjord, Adrian Montgomery Ruf and Espen Sande. A special thank also goes to friend and colleague, Martin Helsø, for being a consistent running partner at the gym and for organising 'grøtlunsj' every Friday, together with Nikolai Bjørnestøl Hansen, Håkon A. Kolderup and Bjørn Skauli. I am also very grateful to my friends and colleagues Ulrik Enstad, Fredrik Meyer, Jonas Irgens Kylling, Luca E. Gazdag and Iben C. Simonsen, for helping me to climb new heights. I'm also grateful for everyone else keeping me company during these years. In particular,

Edvard Aksnes, Mari Brathovde, Sissel Fladby, Leif Kenneth L. Hagen, Anders Matheson, Paul A. Maugesten, Lars Andreas Tveiten and Tale B. Ulfsby all need to be mentioned.

As an integrated part of my PhD scholarship, I have had 25 % teaching duty at the Department of Mathematics. During this time, I've been working with Knut Mørken and teaching in his course '*MAT-INF 1100 – Modeling and computations*'. Knut is a highly engaged lecturer with many years of teaching experience. I want to thank Knut for letting me teach his course and for many enriching conversations on teaching.

During the work with this thesis, I have twice, spent six months at the Cambridge Centre for Analysis (CCA) at the University of Cambridge, UK, the first half of 2018 and the first half of 2019. I want to thank my supervisor Anders again, for hosting me, and all the people I have met during these stays. In particular, Matthew Colbrook, Zhen Ning David Liu, Alexander Bastounis and Simen Westbye Moe, who have not already been mentioned.

Lastly, I want to thank my family for support during these years, and Silvia Lavagnini for being patient with me whenever I put work first.

Vegard Antun
Oslo, June 2020

List of papers

Paper I

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'.

Published in Proceedings of the National Academy of Sciences, 2020.

Paper II

N. M. Gottschling, V. Antun, B. Adcock and A. C. Hansen. 'The troublesome kernel: why deep learning for inverse problems is typically unstable'.

Submitted for publication in *SIAM review*.

Paper III

B. Adcock, V. Antun and A. C. Hansen. 'Uniform recovery in infinite-dimensional compressed sensing and applications to structured binary sampling'.

Submitted for publication in *Applied and Computational Harmonic Analysis*.

Paper IV

V. Antun and Ø. Ryan. 'On the unification of schemes for wavelets on the interval'.

Submitted for publication in *Acta Applicandae Mathematicae*.

Paper V

L. Thesing, V. Antun, A. C. Hansen 'What do AI algorithms actually learn? – On false structures in deep learning'.

arXiv: 1906.01478.

Contents

Chapter 1

Introduction

There are currently two paradigm shifts happening in society and scientific computing: (1) Artificial Intelligence (AI) is replacing humans in problem solving, and, (2) AI is replacing the standard algorithms in computational science and engineering. Since reliable numerical calculations are paramount, algorithms for computational science are traditionally based on two pillars: *accuracy* and *stability*. Notably, this is true for image reconstruction, which is a mainstay of computational science, providing fundamental tools in medical, scientific and industrial imaging.

In this thesis, we demonstrate that the stability pillar is typically absent in current deep learning [47] and AI-based algorithms for image reconstruction, and we present a solution to why this phenomenon occurs for AI-based methods applied both to image reconstruction and to classification in general. This raises two fundamental questions: how reliable are such algorithms when applied in society, and do AI-based algorithms have the unavoidable Achilles heel of instability? We investigate these phenomena, and we introduce a framework designed to demonstrate, investigate and ultimately answer these fundamental questions.

At the forefront of AI developments, we have the field of deep learning, utilising neural networks to solve problems with an accuracy that was not even imaginable ten years ago. A first impression of this new AI revolution appeared in 2012, when the neural network AlexNet [46] almost halved the error rate for image classification compared to standard algorithms at the time. Since then, we have seen neural networks and deep learning reach state-of-the-art performance in historically-challenging problems, such as speech recognition [22, 40], natural language processing [74], game intelligence [67] and autonomous vehicles [11]. Following these breakthroughs, AI techniques are starting to replace standard algorithms in the sciences. Examples include, but are by no means limited to, any type of image reconstruction problems [7, 84], such as Magnetic Resonance Imaging (MRI) and Computed Tomography (CT), solution of ordinary and partial differential equations [20, 27, 61, 79], as well as polynomial [25, 82] and high-dimensional approximations [65].

Despite deep learning's remarkable performance, it is widely acknowledged that neural networks are unstable to certain tiny perturbations of their input, causing the network to produce completely wrong output (often with a highly non-human or unpredictable behaviour). This phenomenon was first observed in image classification [75], where a 'fire truck' can be seen as a 'bird', or two images of a mole, which to a human doctor look identical, are classified with high certainty by the neural network as cancerous and not cancerous, respectively [30]. However, this phenomenon is not restricted to computer vision, but is present in most of the new AI technologies, such as in reinforcement learning [10, 43], speech-to-text translation [16], speech recognition models and voice controllable system [17, 83], natural language processing [32, 49] and image reconstruction [6, 42]. The lack of robustness of modern AI is, therefore, causing concerns in high-stake applications [30] and in society in general [8, 35].

This thesis investigates image reconstruction from the traditional pillars of numerical analysis, namely stability and accuracy. Starting in 2006, with the works by Candes, Romberg & Tao in [13] and Donoho in [26], compressive sensing (CS) created a new standard for image reconstruction in the underdetermined setting. Since then, compressive sensing has spurred renewed interest for inverse problems in imaging, leading to certain accuracy and stability guarantees for methods based on CS. Notably, this has ultimately lead to approval of compressive sensing based techniques in commercial medical imaging [76, 77].

During the last decade, we have also witnessed the development of a myriad of methods based on deep learning for image reconstruction. These methods claim to achieve comparable or even superior performance [84] to compressive sensing based methods. Their potential has therefore been described by *Nature* as ‘transformative’ [71]. However, there is a growing awareness that deep learning techniques have not yet been subjected to the same rigorous standards as other more well-established methods in scientific computing [8].

We investigate compressive sensing and deep learning through the traditional eyes of scientific computing. Some of our findings are summarised below.

(i) By providing numerous examples, we establish that trained neural networks for image reconstruction can be unstable with respect to certain tiny perturbations of their input. As a consequence, the neural networks can potentially add false positives or false negatives in the images. That is, adding false information or removing crucial information in the images, respectively.

(ii) We explain why these instabilities occur, and we show that the instabilities are stable, meaning that the perturbations causing false positives and false negatives do not belong to a set of Lebesgue measure zero.

(iii) We show that, by adding no more than two extra samples to the training set, we can make a neural network trained for image reconstruction arbitrarily unstable.

(iv) We show that deep learning can be more accurate than compressive sensing at the cost of instabilities. Specifically, the more accurate a neural network is, the larger the instabilities will be.

(v) We derive recovery guarantees, ensuring stable and accurate compressive sensing recovery of functions belonging to infinite-dimensional function spaces (specifically for functions in $L^2([0,1])$).

(vi) Wavelet sparsity is a critical component of compressive sensing in imaging. We unify previous deductions of wavelet on the interval preserving polynomial exactness, and we explain how to transfer this to software. These wavelets ensure more accurate compressive sensing reconstructions.

Moreover, we also discuss instabilities in deep learning for classification problems. Our main finding is:

(vii) We present a possible explanation, formulated as conjecture, to why deep learning is so successful in classification problems, and why neural networks based on deep learning are at the same time universally unstable.

In the remainder of this introduction, we discuss what an imaging problem is, how to model it and methods for solving it. We also explain a simple setup for classification problems, and how deep learning tries to learn the right structure. We aim at presenting the background for this thesis in a simplified setting and to motivate many of the results we derive later in the thesis. For this reason, we summarise our findings without going into details. Some of the results are stated here in a simpler and less general form than when they appear later in the thesis.

1.1 Inverse problems in imaging

Daily, we all look at images on our phone, computer, or on the subway on our way to work. A standard camera typically captures images we meet in our everyday life, but images are formed in many other ways. To name a few, we have images acquired in the infrared spectrum, ultrasound images, radar, magnetic resonance imaging and computed tomography. In this thesis, we consider how to go from the measurements produced by the sampling device, to the final image. As we shall see, this is a delicate procedure, where our overall goal is to produce a high-quality image, which accurately represents the underlying object.

Generally, we shall also be interested in using as few measurements as possible. There are many reasons for not sampling more than necessary, but time, power and costs savings are among the key drivers. For medical imaging, saving time can also reduce patient discomfort and reduce exposure to radiation for certain types of sampling modalities such as CT.

In its simplest form, an imaging problem can be described by the equation

$$y = Ax. \tag{1.1}$$

Here $y \in \mathbb{C}^m$ is a vector of acquired measurements and $A \in \mathbb{C}^{m \times N}$ is a simplified model of the sampling device. The vector $x \in \mathbb{C}^N$ is a vectorised version of the unknown image we want to recover. A few examples of measurements y, using different transforms A, can be seen in Figure 1.1.

If $m = N$ and A is non-singular and well conditioned, this problem becomes trivial. This is the case for our everyday cameras, where the identity matrix can model the sampling modality. For more advanced imaging modalities, such as MR or CT, the matrix A is better modelled using a discretised Fourier or

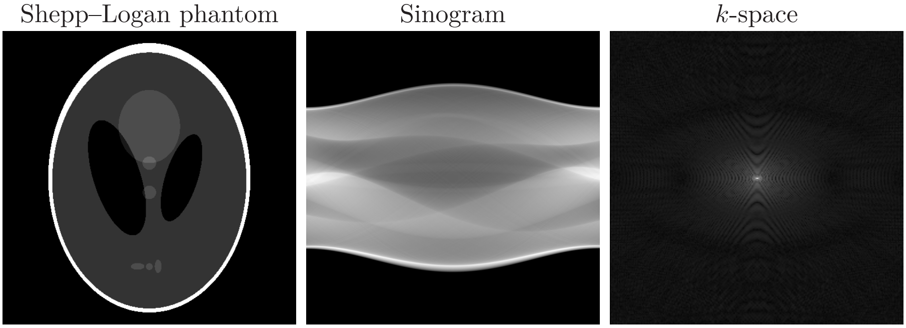


Figure 1.1: (**Sampling domains**). The Shepp-Logan phantom image [66] (left) and its measurements using different sampling modalities. CT is best modelled using the Radon transform. The raw data originating from this transform is called a sinogram and is displayed in the middle image. MR is modelled using Fourier measurements, and in the right image, we see the result of a discrete Fourier transform of the left image. This transformed domain is called 'k-space', and is the data we would get from an MR scanner.

Radon transform. For these modalities, it is often undesirable, and sometimes unfeasible, to do full sampling. We, therefore, have $m < N$ measurements.

When $m < N$, the linear system (1.1) is underdetermined, and there are many x's which agrees with the measurements y. Thus, without more information, we are facing an ill-posed problem. However, images have specific structures, such as edges, lines and large regions with roughly the same colour. This extra information makes the problem of recovering x more tractable.

The linear system (1.1) is a simplified model of the imaging process, with all of its modelling assumptions. While having a simple model might be easier when reconstructing an image, simplified models might lead to undesirable artefacts in the reconstructed image [18, 33, 58]. This is particularly true, for sampling modalities where the acquisition process is better modelled using continuous integral transform, rather than the discrete inner-product between the vectorised image x and the rows of A. For such models, we represent the image as a function f in some infinite-dimensional separable Hilbert space $\mathcal{H}$, and we model the acquisition process as

$$y = \mathscr{A} f. \tag{1.2}$$

Here $y \in \mathbb{C}^m$ still represents the measurements produced by the sampling device, but $\mathscr{A}$ is now a linear operator acting on elements in $\mathcal{H}$. To exactly recover functions in infinite-dimensional function spaces is impossible on a computer, as the coefficients of such functions are infinite sequences. However, by using careful discretizations of (1.2), we can recover an approximation to the M first coefficients of f, in some suitable basis. This is done by rewriting (1.2) as a linear system of the form (1.1). We will discuss the details of this discretisation in Section 1.7, and for now, we simply consider the finite-dimensional model

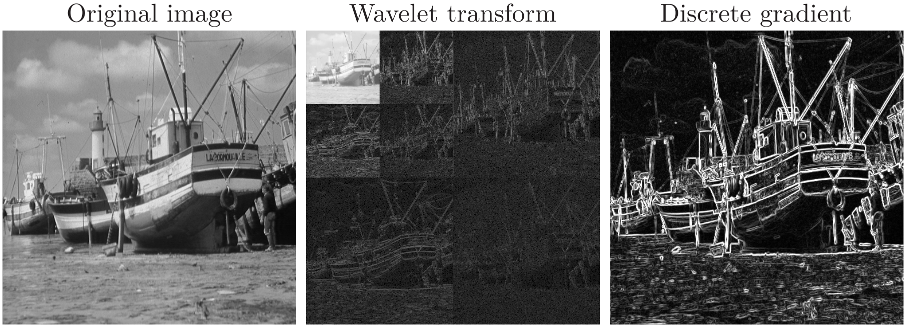


Figure 1.2: (**Images are sparse under certain transforms**). An image $x \in \mathbb{C}^N$ (left) under different sparsifying transforms. In the middle, we see a wavelet transformed image, and to the right, we see a discrete gradient image.

(1.1).

Given the above setup with $m < N$, approaches for recovering x in imaging usually takes one of two possible forms. Either we use reconstruction techniques based on compressive sensing theory, or we learn the reconstruction map using deep learning. Among the two, compressive sensing has to be regarded as a 'standard algorithm'. It has been around since the mid-2000s and is approved by the FDA for certain types of MRI [76, 77]. Nevertheless, deep learning is an attractive competitor, which might offer higher accuracy. In the next two sections, we review the two approaches separately.

1.2 Compressive sensing

The theory surrounding compressive sensing is based on sparsity. This is motivated by the fact that most images are approximately sparse under certain sparsifying transforms. Two examples, wavelet and discrete gradient, can be seen in Figure 1.2, but other options such as shearlets [50], curvelets [14], and total generalised variation [12] are also used.

Let $W \in \mathbb{C}^{K\times N}$ denote one of these transforms. The idea of compressive sensing is to find a solution $\hat{x}$ of the linear system $Ax = y$ for which Wx is approximately sparse. This is typically done by solving some type of ℓ^1-regularised optimization problem, such as

$$\operatorname*{minimise}_{z\in\mathbb{C}^N} \|Wz\|_{\ell^1} \quad \text{subject to} \quad Az = y,$$

called *basis pursuit*, or an unconstrained version of it, given by

$$\operatorname*{minimise}_{z\in\mathbb{C}^N} \lambda\|Wz\|_{\ell^1} + \frac{1}{2}\|Az - y\|_{\ell^2}^2, \quad \lambda > 0,$$

called (unconstrained) *LASSO*. The above optimization problems are both convex, and we can compute approximations to the minimum using a large number of algorithms. Moreover, the use of the ℓ^1-norm ensures that the solutions $\hat{x}$ is approximately sparse in the transformed domain $W\hat{x}$.

To see why this strategy works, we introduce some notation. For a set $\Omega \subseteq \{1, \dots, N\}$, we let $P_\Omega \colon \mathbb{C}^N \to \mathbb{C}^N$ be the projection onto the coordinates indexed by Ω, i.e., $(P_\Omega x)_i = x_i$ if $i \in \Omega$ and zero otherwise. We sometimes abuse notation slightly and assume that P_Ω is a projection onto $\mathbb{C}^{|\Omega|}$. We also recall that a vector with at most s non-zero coefficients is called s-sparse.

Definition 1.2.1 (Robust Null Space Property)**.** A matrix $A \in \mathbb{C}^{m \times N}$ satisfies the *robust Null Space Property (rNSP)* of order $1 \leq s \leq N$ with constants $0 < \rho < 1$ and $\gamma > 0$, if

$$\|P_\Omega x\|_{\ell^2} \leq \frac{\rho}{\sqrt{s}} \|P_\Omega^\perp x\|_{\ell^1} + \gamma \|Ax\|_{\ell^2},$$

for all $x \in \mathbb{C}^N$ and any $\Omega \subseteq \{1, \dots, N\}$ with $|\Omega| \leq s$.

Next consider a simplified setup, where $W \in \mathbb{C}^{N \times N}$ is unitary, as in the case of certain types of wavelets, for example. It is then possible to show the following result (see e.g., [31, Thm. 4.22]), which ensures accurate and stable recovery, even if x is only approximately s-sparse and the measurements y are contaminated with noise.

Theorem 1.2.2 (rNSP implies accurate and stable recovery)**.** *Suppose that $W \in \mathbb{C}^{N \times N}$ is unitary and $A \in \mathbb{C}^{m \times N}$ is such that AW^{-1} has the rNSP of order s with constants $0 < \rho < 1$ and $\gamma > 0$. Let $x \in \mathbb{C}^N$ and $y = Ax + e \in \mathbb{C}^m$, where $\|e\|_{\ell^2} \leq \eta$ for some $\eta \geq 0$. If*

$$\hat{x} \in \operatorname*{argmin}_z \|Wz\|_{\ell^1} \text{ subject to } \|Az - y\|_{\ell^2} \leq \eta, \tag{1.3}$$

then there are constants C_1, C_2 depending on ρ and γ only, such that

$$\|\hat{x} - x\|_{\ell^2} \leq C_1 \frac{\sigma_s(Wx)_{\ell^1}}{\sqrt{s}} + C_2 \eta,$$

where $\sigma_s(z)_{\ell^1} = \inf\{\|z - v\|_{\ell^1} : v \text{ is } s\text{-sparse}\}$ is the ℓ^1-distance to a s-sparse vector.

In particular, we notice that if x is s-sparse and we do not have noise, then we get exact recovery. Moreover, the further away x is from being s-sparse and the more noise we add, the worse the reconstruction will be. Thus, if x does not satisfy our a priori assumption about being approximately sparse, the minimiser in (1.3) is not the x we are searching for.

1.2.1 Compressive sensing for image reconstruction

In imaging, the matrix A models some physical sampling device, which often has certain restrictions on how it can acquire its samples. As an example, consider MRI, where the matrix A can be modelled as

$$A = P_\Omega F, \tag{1.4}$$

where $F \in \mathbb{C}^{N\times N}$ a normalised discrete Fourier transform (DFT) matrix. Here we can choose the projection set $\Omega \subseteq \{1,\ldots,N\}$ as we want, but the matrix F represents the underlying physics of the MR scan, and is not something we can change arbitrarily. Other examples include Fluorescence microscopy [72] and lensless imaging [85], where the matrix F above is replaced with a binary matrix, such as the unitary Hadamard matrix $H \in \mathbb{R}^{N\times N}$. Another example is CT modelled using a Radon transform and we have some freedom to choose the angles $\theta \in [0,\pi)$ of the rays we want to measure.

Likewise, we have some freedom in how to choose the sparsifying transform W, but also this transform must be chosen so that images are sparse in the transformed domain.

Going back to the setup of Theorem 1.2.2, we assume $W \in \mathbb{C}^{N\times N}$ is unitary, and recall that the stability and accuracy guarantees provided by the theorem hinge on the matrix AW^{-1} having the rNSP of order s. For any unitary matrix W, it possible to construct a (random) matrix A by drawing its entries from certain probability distributions in such a way that AW^{-1} satisfies the rNSP of order s with high probability, see, e.g., [31, Thm. 9.1]. The problem, however, is that most such matrices A can not be represented by physical devices in imaging.

In these cases, where our imaging problem is constrained by the physics of a sampling device, a more refined theory is necessary to provide recovery guarantees. There exist many works in this direction, see, e.g., [4, 9, 48]. In Paper III we build on these models, and provide uniform recovery guarantees for an infinite-dimensional model in compressive sensing. To do so, we define properties similar to the rNSP and derive theorems with similar recovery guarantees as given in Theorem 1.2.2. The concrete setup and a summary of our results are described in Section 1.7 and Section 1.8, respectively.

1.3 Deep learning

The term *deep learning* is used for almost any type of problem solving method using neural networks. In this short introduction we focus on the core application of deep learning, namely supervised learning problems of the following form. Given pairs of data

$$\left\{(y^{(1)},x^{(1)}),\ldots,(y^{(n)},x^{(n)})\right\} = \mathcal{T} \subset \mathbb{R}^m \times \mathbb{R}^N, \tag{1.5}$$

and a semi-metric d on $\mathbb{R}^N$, we want to find a function $\Psi\colon \mathbb{R}^m \to \mathbb{R}^N$ for which $d(\Psi(y^{(i)}),x^{(i)}) \leq \eta$, for $i \in S \subseteq \{1,\ldots,n\}$ and some $\eta \geq 0$. The set S is often not decided beforehand, but it is determined while searching for Ψ. We emphasize that $S \subseteq \{1,\ldots,n\}$ might be a proper subset, and note that herein lies some of the delicacy of data driven methods.

Indeed, to find a function Ψ which interpolates all the data in $\mathcal{T}$ is straightforward and can be achieved with a large number of different functions, the simplest being

$$\Psi(y) = \begin{cases} x^{(i)} & \text{if } y = y^{(i)} \ i \in \{1, \dots, n\} \\ 0 & \text{otherwise} \end{cases}.$$

The delicacy of learning is to find a function which agrees with all data in $\mathcal{T}$ using only a subset of the data. This is done as follows. We partition the set $\mathcal{T}$ into two subset,

$$\mathcal{T}_{\text{train}} \cup \mathcal{T}_{\text{test}} = \mathcal{T}, \quad \text{where} \quad \mathcal{T}_{\text{train}} \cap \mathcal{T}_{\text{test}} = \emptyset$$

and find a function which agrees with all or most of the data in $\mathcal{T}_{\text{train}}$ without looking at $\mathcal{T}_{\text{test}}$. Then, we check if Ψ also agrees with the data in $\mathcal{T}_{\text{test}}$. If it agrees with most of the data in $\mathcal{T}_{\text{test}}$, then we have found our desired Ψ, and we stop. Otherwise, we repeat the procedure.

Obviously, it is hard to repeat this procedure until $d(\Psi(y^{(i)}), x^{(i)}) \leq \eta$, for all $i \in \{1, \dots, n\}$, but in the end, we settle for a function which approximate most or all of data with an error which is no more than $\eta \geq 0$.

The above procedure for learning a function Ψ applies to any supervised learning problem and is not restricted to deep learning. What is characteristic for deep learning, is to search for functions Ψ which are neural networks. In fact, one of the reasons why neural networks have become popular is that a neural network fitting the data in $\mathcal{T}_{\text{train}}$, remarkably often also fit the data in $\mathcal{T}_{\text{test}}$, for widely different data $\mathcal{T}$ and choices of semi-metrics d.

Below we introduce the standard definition of a neural network, and we discuss how this definition does not always agree with what is being used in practice.

Definition 1.3.1. An L-layer neural network is a function $\Psi : \mathbb{R}^m \to \mathbb{R}^N$ of the form

$$\Psi(y) = V_L(\rho(V_{L-1}(\rho(\dots \rho(V_1(y)) \dots)))), \quad y \in \mathbb{R}^m,$$

where each $V_j : \mathbb{R}^{n_{j-1}} \to \mathbb{R}^{n_j}$ is an affine map, namely

$$V_j(y) = W_j y + b_j, \qquad W_j \in \mathbb{R}^{n_j \times n_{j-1}}, \quad b_j \in \mathbb{R}^{n_j}, \tag{1.6}$$

and $\rho : \mathbb{R} \to \mathbb{R}$ is a non-linear function acting component-wise on a vector, and $n_0 = m$, $n_L = N$. We let $\mathcal{NN}_{[\mathbf{N},L,\rho]}$, with $\mathbf{N} = [m = n_0, n_1, \dots, n_L = N]$, denote the set of all such L-layer neural networks.

Here the W_j's and b_j's are referred to as the *weights* of the network, and ρ is called an *activation function*. Typical choices for ρ are the *Rectified Linear Unit (ReLU)*, defined by $\rho(x) = \max\{0, x\}$, or the *sigmoid*, defined by $\rho(x) = 1/(1 + \mathrm{e}^{-x})$. The *architecture* of a neural network refers to fixed choices of ρ, L and $\mathbf{N}$. If all of the W_j's are convolutional operators, the network is called a convolutional neural network (CNN).

While the above definition is mostly used by mathematicians, we note that functions not satisfying the conditions of Definition 1.3.1 are called neural networks.[1] In particular, most deep learning partitioner's would not feel restricted to use the same activation function between all the affine maps. Moreover, it is also common to use *max-pooling* layers, mapping a vector $v = (v_k)_{k=1}^K$ as

$$\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_K \end{bmatrix} \mapsto \begin{bmatrix} \max\{v_1, v_2\} \\ \max\{v_3, v_4\} \\ \vdots \\ \max\{v_{K-1}, v_K\} \end{bmatrix}.$$

Many modern architectures also use so-called skip connections,[2] where the input to one layer is sent directly forward to a layer further down in the network. As an example, let $h_1 \colon \mathbb{R}^m \to \mathbb{R}^m$ and $h_2 \colon \mathbb{R}^m \to \mathbb{R}^N$ be neural networks according to Definition 1.3.1, then

$$h(x) = h_2(x + h_1(x))$$

is a neural network with a skip connection. This type of function compositions can be done multiple times, so that one obtains both internal and global (that is $h(x) = x + h_1(x)$) skip connections.

The motivation for deviating from Definition 1.3.1 is often to work with functions which more easily fit the data in both $\mathcal{T}_{\text{train}}$ and $\mathcal{T}_{\text{test}}$.

1.3.1 Deep learning for image reconstruction

Before we start, we note that many imaging problems are formulated using complex numbers, while the presentation above, focus on real-valued neural networks. A standard way of associating a vector $y \in \mathbb{C}^m$ with a vector $y' \in \mathbb{R}^{2m}$ is to let y' consist of the real and imaginary parts of y. We may then apply a real-valued network $\Psi : \mathbb{R}^{2m} \to \mathbb{R}^{2N}$. Similarly, $x' = \Psi(y')$ is associated with a complex image $x \in \mathbb{C}^N$. We assume that the complex case is treated in this way, and henceforth we simply write $\Psi : \mathbb{C}^m \to \mathbb{C}^N$ for a network taking complex inputs and outputs.

We recall from Section 1.1 that an imaging problem is of the form

$$\text{given measurements} \qquad y = Ax, \qquad \text{recover } x, \tag{1.7}$$

where $A \in \mathbb{C}^{m \times N}$ with $m < N$ is a model of the sampling modality. This means that there are many x's which map to the same measurements y. To make the problem tractable, we introduce

$$\begin{aligned} &\text{a domain } \mathcal{M}_1 \subset \mathbb{C}^N \text{ where } (\mathcal{M}_1, d_1) \text{ is a metric space, and} \\ &\text{the range } \mathcal{M}_2 = A(\mathcal{M}_1) \subset \mathbb{C}^m, \text{ where } (\mathcal{M}_2, d_2) \text{ is a metric space.} \end{aligned}$$

The idea is that the domain $\mathcal{M}_1$ is the 'set of natural images', which is substantially smaller than $\mathbb{C}^N$. This could make it possible to learn a mapping

[1] This mix of terminology is not new, and have been noted by Pinkus [59] already in 1999.

[2] The term 'skip connection' is far from unique for this type of connections, and the terms residual [36], short-cut [19, 69] or jump connections [81] are all used.

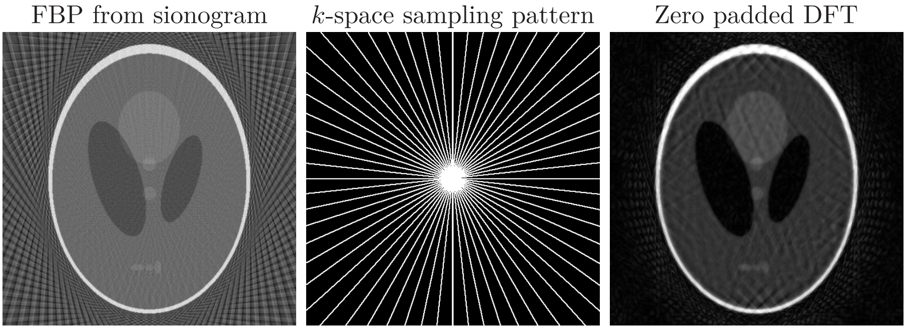


Figure 1.3: (**Linear reconstruction maps**). Linear reconstruction maps, using a Filtered back projection (left) on sinogram data, and a zero-padded DFT matrix $A^* = F^* P_\Omega$ on k-space data (right). In the middle we see the k-space sampling pattern Ω. Here a white dot indicate that we sample this pixel in k-space and a black dot indicate no sampling.

$\Psi\colon \mathcal{M}_2 \to \mathcal{M}_1$. In deep learning, the sets $\mathcal{M}_1$ and $\mathcal{M}_2$ are not known, but are learned from the data $\mathcal{T}$. Thus, we assume that $\mathcal{T} \subset \mathcal{M}_2 \times \mathcal{M}_1$ and try to learn the mapping $\Psi\colon \mathcal{M}_2 \to \mathcal{M}_1$ using the dataset $\mathcal{T}$.

Often, trying to learn the mapping $y \mapsto x$ directly is difficult. This is because the data y typically have a completely different structure than the image x, after being transformed by A. To undo the linear map A, a common approach is to use a known linear operator $H\colon \mathbb{C}^m \to \mathbb{C}^N$ (depending on A) to map the measurements y back into the image domain $\mathbb{C}^N$. For Fourier sampling, i.e., for $A = P_\Omega F$ where $F \in C^{N \times N}$ is a normalised DFT matrix, a common choice for H is A^*, whereas for Radon sampling a Filtered back projection (FBP) (see, e.g., [56, Ch. 5]) matrix is often used . Whenever the resulting image is complex-valued, we always display the modulus of the complex numbers in our figures. A few examples of data y, looking nothing like the underlying image x, can be seen in Figure 1.1. In Figure 1.3, we show the cheap and often erroneous reconstructions provided by H for Fourier and Radon sampling.

The simplest approaches for learning the mapping $\Psi\colon \mathcal{M}_2 \to \mathcal{M}_1$, rewrite the problem into a denoising problem. That is, we let the neural network Ψ have the form

$$\Psi(y) = Hy - \phi(Hy), \tag{1.8}$$

where $\phi\colon \mathbb{C}^N \to \mathbb{C}^N$ is a trainable convolutional neural network which learns to recognise the noisy artefacts in the images Hy. While the structure of Ψ in (1.8) is uncomplicated and more advanced setups exist, such as [34, 41, 51, 64, 70], the main idea of building the known operator H into the network architecture and use convolutional layers to improve image quality is used by most networks [53].

1.3.2 Deep learning for classification

Consider a classification problem where each sample $y^{(i)} \in \mathbb{R}^m$, $i \in \{1, \dots, n\}$ from $\mathcal{T}$ associate to exactly one out of N different labels. If the $y^{(i)}$'s are images, the labels might be {'cat', 'firetruck', . . . , 'football'}, but we label them as $\{1, \dots, N\}$. The neural network $\Psi \colon \mathbb{R}^m \to \mathbb{R}^N$ is a function whose output is a vector in $\mathbb{R}^N$. To measure the networks accuracy in terms of correctly classified samples, it is nessesary to convert this vector to an integer in the range $1, \dots, N$. One way to do this is as follows. Let $\gamma \colon \mathbb{R}^N \to \{1, \dots, N\}$ be the function

$$\gamma(v) = \text{smallest } i \text{ such that } v_i \geq v_j \text{ for all } j \in \{1, \dots, N\}, \tag{1.9}$$

for a vector $v \in \mathbb{R}^N$. For $y^{(i)} \in \mathbb{R}^m$, $i \in \{1, \dots, n\}$, let the corresponding vector $x^{(i)} \in \mathbb{R}^N$ be *one-hot* encoded. That is

$$(x^{(i)})_k = \begin{cases} 1 & \text{if } y^{(i)} \text{ belong to class } k \\ 0 & \text{otherwise} \end{cases}.$$

Using the semi-metric $d(v, w) = d^{\text{disc}}(\gamma(v), \gamma(w))$, where

$$d^{\text{disc}}(a, b) = \begin{cases} 1 & \text{if } a \neq b \\ 0 & \text{otherwise} \end{cases}$$

is the discrete metric, we can measure the neural networks accuracy, in terms of correctly classified samples, on the set $\mathcal{T}$ by

$$\frac{1}{|\mathcal{T}|} \sum_{(y,x) \in \mathcal{T}} d(\Psi(y), x).$$

As for image reconstruction, it is common in classification to deviate from Definition 1.3.1 when constructing the architecture of the neural network. For high dimensional problems, such as image classification, the use of max-pooling layers is widespread to reduce the size effectively. Moreover, to use internal skip connections between layers is also common [36].

1.3.3 Instabilities in deep learning for classification

Despite deep learnings ability to produce state-of-the-art accuracy for a wide range of problems, it is widely recognised that deep learning and neural networks are often highly unstable with respect to tiny perturbations of their input [5, 37]. The most prominent examples of instabilities are found within image classification [28, 29, 45, 54, 55, 57, 68, 73, 75], but the instability phenomenon for neural networks have been demonstrated in other areas as well [1, 10, 16, 32, 43, 49, 80].

For the classification problem described above with N different labels, a worst-case perturbation of the input can take many forms. To make the setup sufficiently general, we start by introducing some notation from [29]. Let $\mathcal{R}$

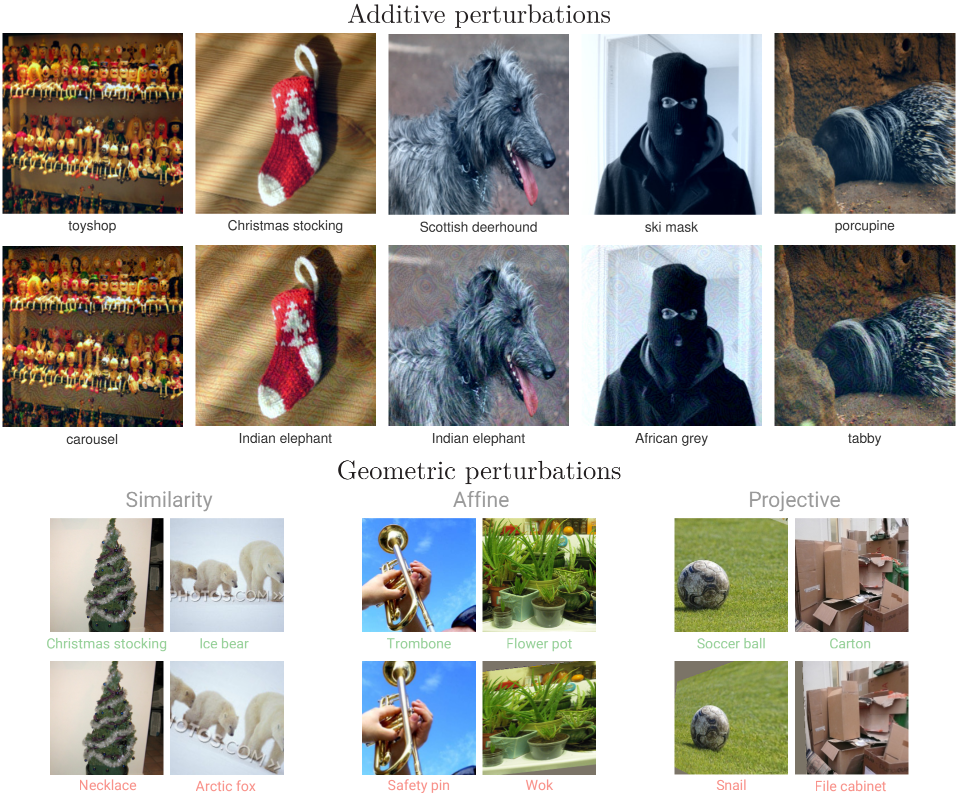


Figure 1.4: (**Tiny perturbations of the input leads to misclassification**). In the top two rows we add a fixed perturbation $r \in \mathbb{R}^m$, to all the images. The figures are from [54]. In the last two rows we consider different types of geometric transformations to the input images. Figures are from [45].

denote a set with a magnitude function $\|\cdot\|_{\mathcal{R}} \colon \mathcal{R} \to \mathbb{R}_+$ and let $T_r \colon \mathbb{R}^m \to \mathbb{R}^m$ be a family of mappings defined for each $r \in \mathcal{R}$. The mapping T_r is what we call a perturbation mapping and it can have different forms, depending on what type of perturbations we consider.

As an example, suppose we are working with image classification. Then the set $\mathcal{R}$ could be $[0, 2\pi)$ and the perturbations T_r could be rotation of the image by an angle $r \in \mathcal{R}$. Another more general example is to consider $\mathcal{R} \subset \mathbb{R}^m$, and let $T_r(y) = y + r$ denote additive perturbation.

Thus, given the triple $\{\mathcal{R}, \|\cdot\|_{\mathcal{R}}, T_r\}$ and a neural network $\Psi \colon \mathbb{R}^m \to \mathbb{R}^N$, we define the worst case perturbation for a given sample $y \in \mathbb{R}^m$ as

$$r^*(y) \in \operatorname{argmin}_{r \in \mathcal{R}} \|r\|_{\mathcal{R}} \quad \text{subject to} \quad \gamma(\Psi(T_r(y))) \neq \gamma(\Psi(y)), \tag{1.10}$$

where γ is the function from (1.9). The optimization problem (1.10) defines the perturbation $r^*(y)$ we are searching for, and can be used as a starting point

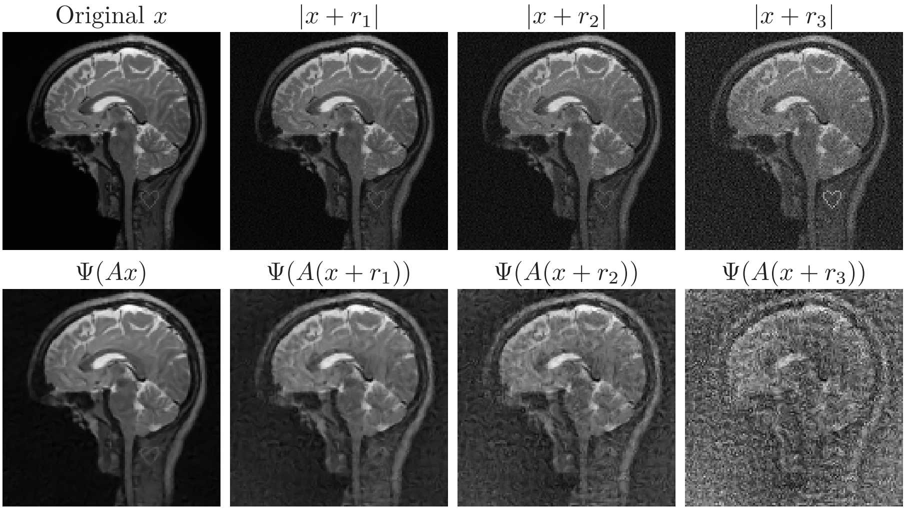


Figure 1.5: (**Unstable neural network**). The effect of small perturbations on the AUTOMAP network [84] for recovering an image x from its measurements $y = Ax$. Here $A \in \mathbb{C}^{m \times N}$ is a subsampled discrete Fourier transform, which is the standard mathematical model for MRI. The left column shows that the network, denoted by Ψ, recovers the original image x well. However, as shown in the second to fourth columns, small perturbations $\|r_1\| < \|r_2\| < \|r_3\|$ of x cause large artefacts in the recovered images $\Psi(A(x + r_i))$. This experiment is from Paper I.

for finding small perturbations which causes misclassification. A few examples of such small perturbations r for different triples $\{\mathcal{R}, \|\cdot\|_{\mathcal{R}}, T_r\}$ can be seen in Figure 1.4, where tiny perturbations makes high accuracy networks mispredict.

1.4 Are deep learning for inverse problems unstable?

Seeing that trained neural networks for classification are unstable to tiny perturbations of their input, one of the questions we investigate in this thesis is whether the same phenomenon occurs for deep learning in inverse problems. Before answering this question, we first recall that constructing a stable neural network is straightforward and can be achieved by picking the zero network, i.e., a network where all weights are zero. Thus, part of the subtlety is to produce a network which finds the right trade-off between stability and accuracy.

In Paper I, we make an empirical investigation of trained neural network's stability and accuracy compared to state-of-the-art (SoA) methods for inverse problems in imaging. These state-of-the-art methods are based on compressive sensing and provide a baseline for what type of accuracy we can achieve with standard methods. Often, these methods come with certain stability guarantees, such as the one we saw in Theorem 1.2.2. To test both the neural networks stability and accuracy, we ask the following:

(i) Are there tiny perturbations either in the image or in the sampling domain, such that the neural network produces a widely different output?

(ii) Is the network able to detect small structural perturbations (such a tumour) in the images?

To answer question (i), we develop an algorithm which, given a neural network $\Psi\colon \mathbb{C}^m \to \mathbb{C}^N$, a sampling matrix $A \in \mathbb{C}^{m\times N}$ and an image $x \in \mathbb{C}^N$, seeks an $r \in \mathbb{C}^N$ such that

$$\|\Psi(y + Ar)) - \Psi(y)\| \text{ is large, while } \|r\| \text{ is small.}$$

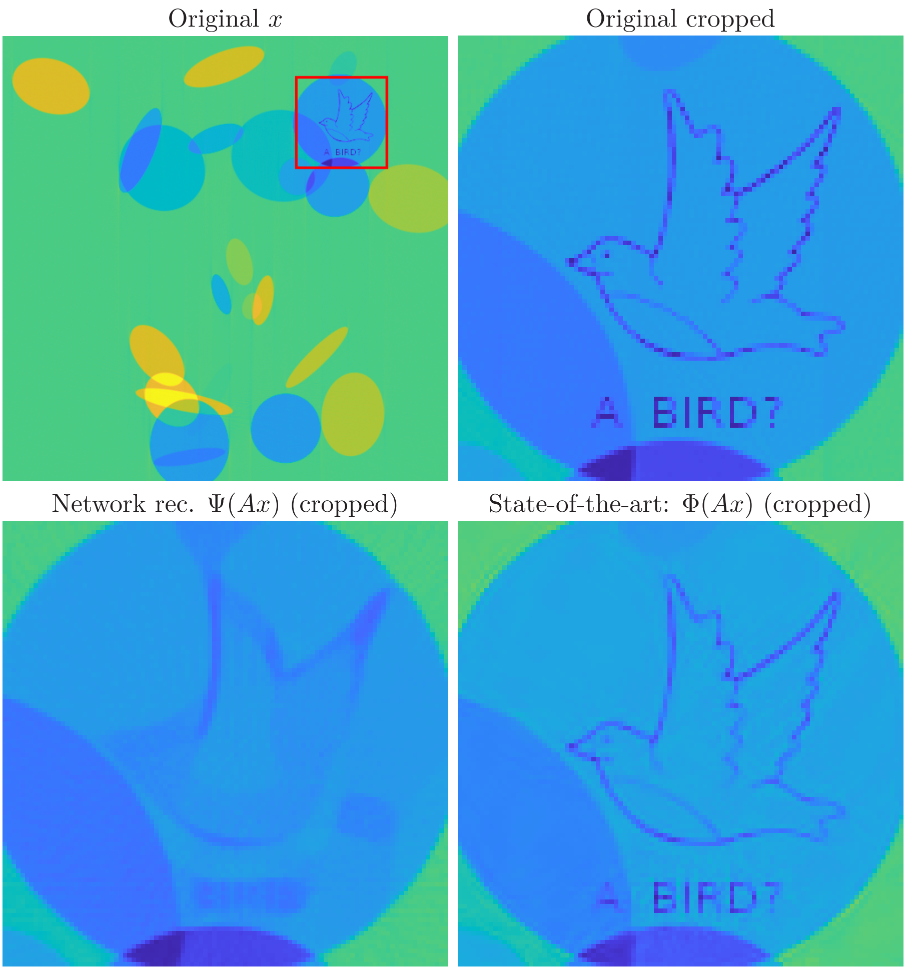
The perturbation r is deliberately computed in the image domain so that one can verify by visual inspection that it is tiny. However, a perturbation in the image domain may imply a perturbation in the sampling domain. For medical imaging applications, we note that these small perturbations r could occur due to slight movements of the patient, anatomical differences, or to the inevitable noise from the sampling device. Thus, the perturbations could be both random and structured. An example of such small perturbations, causing large instabilities in MRI, can be seen in Figure 1.5.

In question (ii) we consider small, clearly visible perturbations in the image domain. In a medical setting, this could be a tumour or small cracks in a bone structure. We test the network's accuracy by checking whether it can resolve fine, and potentially important, details. To create a sufficiently difficult task for the networks, we propose to insert short text strings and symbols in the images, to easily check whether the networks capture small structural changes in the images. An example can be seen in Figure 1.6, where a network, trained to reconstruct images of ellipses, completely washes out any fine details. This is not entirely surprising, as the network have never seen such fine details.

Remark 1.4.1. For both question (i) and (ii), we remark that it is important to test against a state-of-the-art method, to ensure that any instability is caused by the network itself and not because of ill-conditioning of the inverse problem. Indeed, it would certainly be possible to insert a detail so small that no method could stably reconstruct it.

1.5 Why instabilities occur – A motivating example

In Paper II, we investigate from a theoretical perspective why instabilities for inverse problems in imaging occur. To understand some of the mechanisms behind instabilities in inverse problems, we consider a simple example explaining why this phenomenon occurs. Hopefully, this sheds some light on why this problem is more prominent for data driven approaches than for compressive sensing, which is based on prior assumptions.


Original x

Original cropped

Network rec. $\Psi(Ax)$ (cropped)

State-of-the-art: $\Phi(Ax)$ (cropped)

Figure 1.6: (**Small structural perturbation/false negatives**). The FBPConvNet neural network $\Psi : \mathbb{R}^m \to \mathbb{R}^N$ from [44] is trained to recover images comprised of ellipses from a Radon sampling operator $A \in \mathbb{R}^{m \times N}$. Top left: The image x containing a bird and the letters 'A BIRD?'. This is a feature the network has not seen. Top right: Cropped original image. Lower left: The cropped FBPConvNet reconstruction from measurements Ax. Lower right: The cropped reconstruction of x from measurements Ax using a state-of-the-art decoder (based on compressive sensing) $\Phi : \mathbb{R}^m \to \mathbb{R}^N$. See Paper II for experiment details.

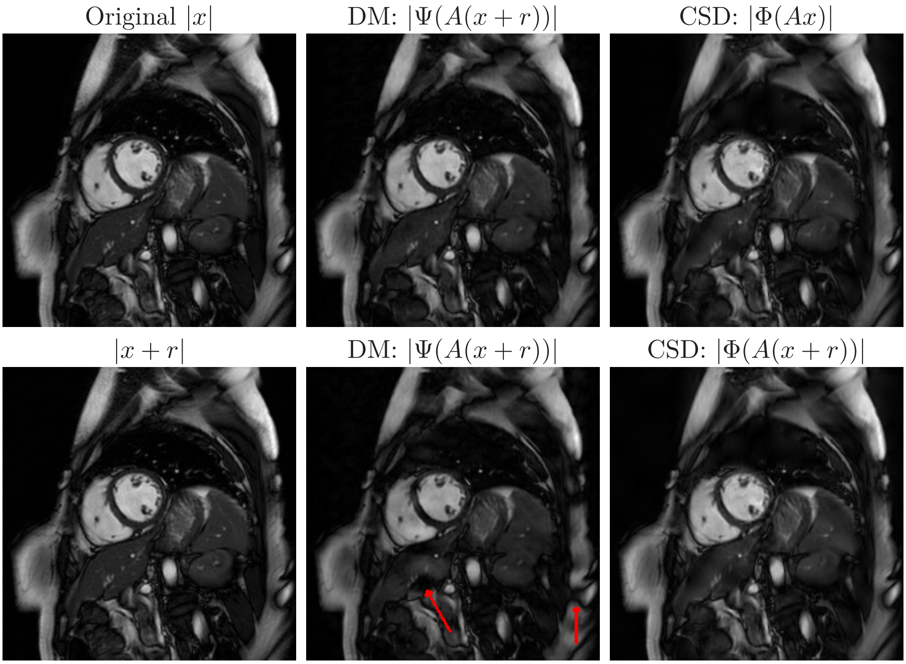


Figure 1.7: (**False positives**). Perturbations cause the Deep MRI neural network (DM) $\Psi : \mathbb{C}^m \to \mathbb{C}^N$ from [64] to reconstruct false positives (red arrows). Left: Original image x (top) and perturbed image $x+r$ (bottom). Middle: DM reconstructs the image x from measurements $y = Ax$ (top) and $\widetilde{y} = A(x+r)$ (bottom). Here $A \in \mathbb{C}^{m\times N}$ is a subsampled Fourier transform. Right: A compressive sensing decoder (CSD) $\Phi : \mathbb{C}^m \to \mathbb{C}^N$ reconstructs the image x from measurements $y = Ax$ (top) and $\widetilde{y} = A(x+r)$ (bottom). This experiment is from Paper I.

We consider the following imaging inverse problem,

$$\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = y, \qquad x_1, x_2, y \in \mathbb{R}, \tag{1.11}$$

where $A = \begin{bmatrix} 1 & 0 \end{bmatrix}$, and we see that $y = x_1$. We consider a data-driven approach for learning the inverse $\Psi \colon \mathcal{M}_2 \to \mathcal{M}_1$, where $\mathcal{M}_1 \subset \mathbb{R}^2$ and $\mathcal{M}_2 \subset \mathbb{R}$. For data-driven methods, we have access to a (potentially large) dataset

$$\mathcal{T} = \{(y^{(1)}, x^{(1)}), \ldots, (y^{(n)}, x^{(n)})\} \subset \mathbb{R} \times \mathbb{R}^2,$$

and it is common to assume $\mathcal{T} \subset \mathcal{M}_2 \times \mathcal{M}_1$, since the sets $\mathcal{M}_1$ and $\mathcal{M}_2$ are supposed to be learned from the data. The sets $\mathcal{M}_1$ and $\mathcal{M}_2$ are equipped with metrics, which we for simplicity assume are induced by norms in this example.

Suppose we after the training, have found a function $\Psi\colon \mathcal{M}_2 \to \mathcal{M}_1$ such that

$$\|\Psi(y^{(j)}) - x^{(j)}\| \leq \eta, \qquad \text{for } j \in S \subseteq \{1,\dots,n\}, \quad \text{where } |S| \geq 2. \tag{1.12}$$

Let $\mathcal{T}_S = \{(y^{(i)}, x^{(i)}) \in \mathcal{T} : i \in S\}$, and fix $(\widetilde{y},\widetilde{x}), (y', x') \in \mathcal{T}_S$. We write these as

$$\widetilde{x} = \begin{bmatrix} \widetilde{x}_1 \\ \widetilde{x}_2 \end{bmatrix} \quad \text{and} \quad x' = \begin{bmatrix} \widetilde{x}_1 + \epsilon \\ x_2' \end{bmatrix}, \quad \text{where } x_1' = \widetilde{x}_1 + \epsilon,$$

and notice that, by assumption, we have

$$\|\Psi(\widetilde{y}) - \widetilde{x}\| \leq \eta \quad \text{and} \quad \|\Psi(\widetilde{y} + \epsilon) - x'\| \leq \eta.$$

Thus, if

$$\|\widetilde{x} - x'\| > 2\eta \text{ is large, while } |\epsilon| > 0 \text{ is small,} \tag{1.13}$$

the mapping sought during training is necessarily sensitive to noise. Indeed, at $\widetilde{y}$ we only require a perturbation e of size $|\epsilon|$ in order for $\Psi(\widetilde{y} + e)$ to give a completely different output.

If (1.13) is fulfilled, what happens in the above example is that Ψ recover elements, whose difference $\widetilde{x} - x'$ lies close to the null space $\mathcal{N}(A)$. Thus, if $\|\widetilde{x} - x'\| > 2\eta$ is bounded from below, we can make the Lipschitz constant of the map Ψ arbitrarily large by choosing $|\epsilon| > 0$ small, i.e., letting $\widetilde{x} - x'$ lie closer to $\mathcal{N}(A)$. The example is of course overly simplified, yet the setup transfers to more advanced examples and can be seen in practice. See, e.g., Figure 1.7, where a tiny perturbation $r \in \mathbb{C}^N$ is mapped to a perturbation $e = Ar$ in the sampling domain, causing a potential false positive. That is, it adds potential false information, by creating certain dark areas in the image. Other examples can be found in Paper I and Paper II.

1.6 Kernel awareness – At the heart of compressive sensing

As the previous example explains, the root to instabilities for inverse problems in imaging is to recover elements whose difference lie close to the null space $\mathcal{N}(A)$ of the sampling operator. Seeing this, it is natural to ask how compressive sensing deals with the same problem. The answer can be summarised in two words, *kernel awareness*, and lies at the heart of compressive sensing theory.

To see this, we go back to the simplified compressive sensing setup from Section 1.2. We set $\Sigma_s = \{z \in \mathbb{C}^N : z \text{ is } s\text{-sparse}\}$ and recall that $W \in \mathbb{C}^{N\times N}$ is a unitary matrix. In compressive sensing, we choose the set $\mathcal{M}_1$ explicitly so that it is a good model for images. One example would be to let W be a discrete wavelet transform (DWT). Then $\mathcal{M}_1 = W^{-1}(\Sigma_s)$ is the set of all s-sparse images in wavelets.

To guarantee accurate and stable recovery on $\mathcal{M}_1$, Theorem 1.2.2 required the matrix AW^{-1} to satisfy the rNSP of order s. That is

$$\|P_\Omega x\|_{\ell^2} \leq \frac{\rho}{\sqrt{s}} \|P_\Omega^\perp x\|_{\ell^1} + \gamma \|AW^{-1} x\|_{\ell^2}, \qquad \text{for all } x \in \mathbb{C}^N,$$

and for any $\Omega \subset \{1, \dots, N\}$ with $|\Omega| \leq s$. We want to show that this implies

$$\|x - z\|_{\ell^2} \leq \frac{(3+\rho)\gamma}{1-\rho} \|AW^{-1}(x-z)\|_{\ell^2}, \quad \text{for all } x, z \in \Sigma_s, \tag{1.14}$$

i.e., that the difference between any two elements in Σ_s is bounded away from the kernel. In Section 1.5, it was the lack of a lower bound like this one which allowed us to map two completely different vectors to almost the same measurements. This caused the instabilities.

We also note that (1.14) implies

$$\|x - z\|_{\ell^2} \leq \frac{(3+\rho)\gamma}{1-\rho} \|A(x-z)\|_{\ell^2} \qquad \text{for all } x, z \in \mathcal{M}_1 = W^{-1}(\Sigma_s)$$

which means that the difference between all s-sparse images in wavelets is bounded away from the kernel. To show (1.14) we rely on the following lemma.

Lemma 1.6.1 ([31, Thm. 4.25]). *Suppose the matrix $AW^{-1} \in \mathbb{C}^{m \times N}$ satisfies the robust null space property of order s with constants $0 < \rho < 1$ and $\gamma > 0$. Then for any $x, z \in \mathbb{C}^N$,*

$$\|x - z\|_{\ell^2} \leq \frac{C}{\sqrt{s}} \left(\|z\|_{\ell^1} - \|x\|_{\ell^1} + 2\sigma_s(x)_1\right) + D\|AW^{-1}(x-z)\|_{\ell^2}$$

where $C = (1+\rho)^2/(1-\rho)$ and $D = (3+\rho)\gamma/(1-\rho)$.

To prove (1.14), let $x, z \in \Sigma_s$, and assume that $\|x\|_{\ell^1} \geq \|z\|_{\ell^1}$. Since $x, z \in \Sigma_s$ we have $\sigma_s(x)_1 = \sigma_s(z)_1 = 0$. Thus by Lemma 1.6.1, we have that $\|x - z\|_{\ell^2} \leq D\|A(x-z)\|_{\ell^2}$. If instead $\|x\|_{\ell^1} \leq \|z\|_{\ell^1}$, we notice that $\|x - z\|_{\ell^2} = \|z - x\|_{\ell^2}$. We can therefore reverse the argument. This proves the claim.

In practice, it might be more realistic to assume that x can be approximated well by an image which is s-sparse in wavelets. If x not exactly s-sparse, we saw that the error bound in Theorem 1.2.2 accounted for vectors which were only approximately sparse and measurements y, being contaminated with noise.

In Section 1.2.1 we mentioned that physical constrains of the sampling device, might make it impossible to design the matrix A in such a way that AW^{-1} satisfies the rNSP. In such cases, more refined models are needed. In the two next sections we introduce one of these models in infinite-dimensional function spaces.

1.7 An infinite-dimensional model for imaging

In infinite-dimensional compressive sensing, we are interested in approximating functions, rather than vectors. A continuous d-dimensional image is a function $f\colon [0,1]^d \to \mathbb{C}$. It can be discretised by evaluating f at distinct points $t_i \in [0,1]^d$. For simplicity, we consider $d = 1$, and let $x_i = f((i-1)/N)$ for $i = 1, \dots, N$.

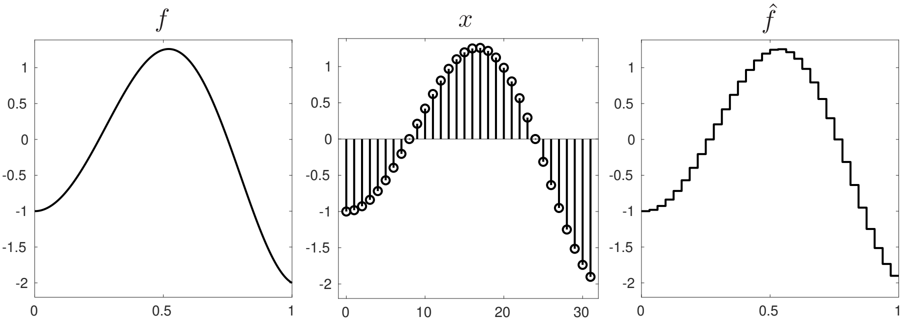


Figure 1.8: (**Continuous and discrete images**). Left: Continuous image $f\colon [0,1] \to \mathbb{C}$. Middle: discrete image $x \in \mathbb{C}^N$. Right: Continuous image $\hat{f}\colon [0,1] \to \mathbb{C}$ create from discrete measurements x.

The vector $x \in \mathbb{C}^N$ represents a discrete version of the image f. We can convert the discrete image x back to a continuous image as

$$\hat{f} = \sum_{i=1}^{N} x_i \chi_i.$$

for appropriate step functions $\chi_i\colon [0,1] \to \mathbb{R}$. An example of the three different kinds of images can be seen in Figure 1.8. Whenever we visualise a complex-valued image, we display the modulus of the complex numbers.

In the finite-dimensional compressive sensing model, we work with discrete images $x \in \mathbb{C}^N$, and we assume that each measurement y_k, $k = 1, \ldots, m$ is best modelled by using the discrete inner-product

$$y_k = \langle x, a_k \rangle_{\mathbb{C}^N} = \sum_{i=1}^{N} x_i (a_k)_i, \tag{1.15}$$

where $a_k \in \mathbb{C}^N$ is the k-th row in the matrix $A \in \mathbb{C}^{m \times N}$.

In the infinite-dimensional model, we consider a function f in an infinite-dimensional separable Hilbert space $\mathcal{H}$[3]. Let $\{s_k\}_{k\in\mathbb{N}}$ be an orthonormal basis for $\mathcal{H}$. We call $\{s_k\}_{k\in\mathbb{N}}$ the sampling basis and model the measurements as

$$y_k = \langle f, s_k \rangle_{\mathcal{H}}.$$

The motivation for using an infinite-dimensional model, is that many physical sampling devices are better modeled using integral transforms, where the function f is integrated over time or over an area. Thus, for certain choices of $\mathcal{H}$, such as $L^2([0,1]^d)$, the above inner-product is an integral transform, i.e.,

$$y_k = \langle f, s_k \rangle_{L^2([0,1]^d)} = \int_{[0,1]^d} f(t) s_k(t) \mathrm{d}t.$$

[3] We note that such a Hilbert space, does not need to be a function space, but we abuse notation slightly and say that $f \in \mathcal{H}$ is a function.

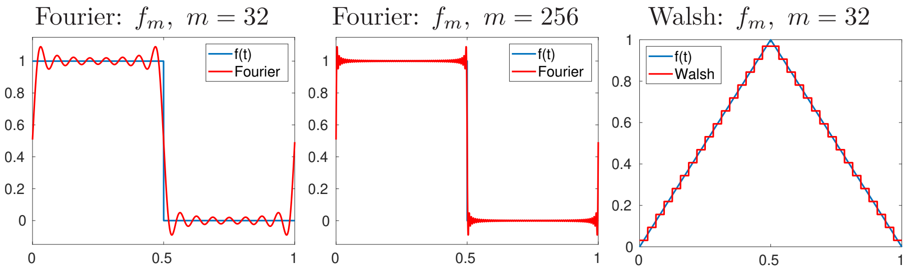


Figure 1.9: (**Undesirable artifacts**). The approximation f_m from (1.16) for both a Fourier and Walsh sampling basis. Left and middle: A m-term Fourier approximation to the discontinuous Haar wavelet. Right: A m-term Walsh approximation to the hat function.

In practice, we can only acquire a finite number of m measurements and since $\{s_k\}_{k\in\mathbb{N}}$ is orthonormal, we can approximate f immediately using

$$f_m = \sum_{i=1}^{m} y_k s_k = \sum_{i=1}^{m} \langle f, s_k\rangle_{\mathcal{H}}\, s_k. \tag{1.16}$$

Most physical devices are restricted to certain choices of basis $\{s_k\}_{k\in\mathbb{N}}$. If $\mathcal{H} = L^2([0,1])$, then examples include Fourier sampling used in MRI with $s_k(t) = e^{2\pi i k t}$ for $k \in \mathbb{Z}$, or Walsh sampling used for binary imaging, where $s_k = w_k$ for $k \in \mathbb{N}_0$. Here $w_k \colon [0,1] \to \{+1,-1\}$ is a Walsh function (see Def. III.4.1).

For both of these choices, the approximation f_m typically contains undesirable artefacts. In the case of Fourier sampling, f_m is a truncated Fourier series, exhibiting Gibbs oscillations at any discontinuity, whereas Walsh sampling gives a blocky approximation due its discontinuous nature. Examples of such artifacts can be seen in Figure 1.9.

1.8 Compressive sensing in infinite-dimensions

The idea of reconstruction techniques, such as infinite-dimensional compressive sensing (and also generalised sampling [2, 3]), is to improve upon the approximation f_m, by computing an approximation in an M-dimensional subspace $\mathcal{R}_M \subset \mathcal{H}$. The subspace $\mathcal{R}_M$ is chosen a priori, and it is designed so that a projection of f onto $\mathcal{R}_M$ is likely to be a good approximation to f. An example can be seen in Figure 1.10, where different choices of wavelet subspaces offer substantially better representations of the underlying functions. Let $\{r_k\}_{k\in\mathbb{N}}$ be an orthonormal basis for $\mathcal{H}$. We call $\{r_k\}_{k\in\mathbb{N}}$ the reconstruction basis, and we let $\mathcal{R}_M = \operatorname{span}\{r_k : 1 \le k \le M\}$.

In compressive sensing an additional requirement on $\{r_k\}_{k\in\mathbb{N}}$, is that the coefficients $x_k = \langle f, r_k\rangle$, $1 \le k \le M$, are sparse. To achieve this, we consider consider orthonormal wavelets bases as our preferred reconstruction bases in this

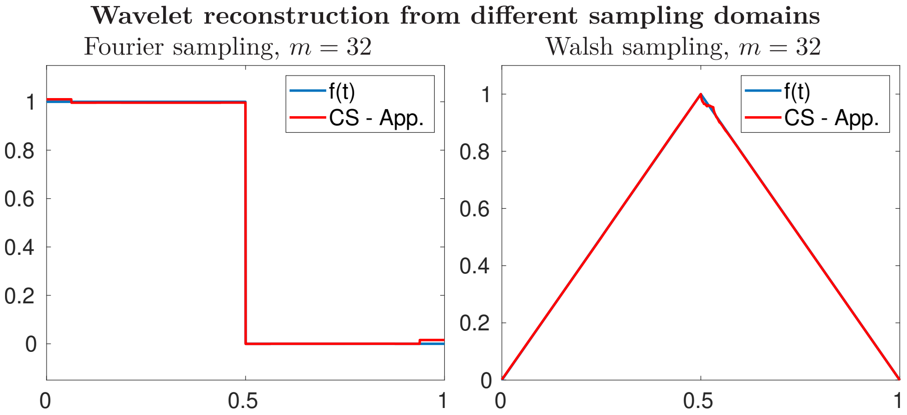


Figure 1.10: (**Improved reconstruction in $\mathcal{R}_M$**). We compute approximations to the functions $f(t)$ considered in Figure 1.9 using functions in the subspace $\mathcal{R}_M$. The approximations are computed using infinite dimensional compressive sensing from $m = 32$ samples in different domains. In the left image, we let $\mathcal{R}_M$ be spanned by the Haar wavelet basis ($M = 16$) and in the right image we consider $\mathcal{R}_M$ spanned by functions from the Daubechies 2 (DB2) wavelet basis ($M = 32$).

thesis. For wavelets, sparsity alone is often too general to obtain sharp estimates on the required number of samples m. In these cases, it is required to take the local structure of the wavelet coefficients into account. This motivates the definition of sparsity in levels. In this thesis, the levels $\mathbf{M}$ are chosen according to wavelet scales.

Definition 1.8.1 (Sparsity in levels)**.** Let $\mathbf{M} = [M_1, \dots, M_r] \in \mathbb{N}^r$, $M_0 = 0$, with $1 \leq M_1 < \cdots < M_r = M$ and let $\boldsymbol{s} = (s_1, \dots, s_r) \in \mathbb{N}^r$ with $s_l \leq M_l - M_{l-1}$, for $l = 1, \dots, r$. We say that the vector $x \in \mathbb{C}^M$ is sparse in levels if

$$|\operatorname{supp}(x) \cap \{M_{l-1}+1, \dots, M_l\}| \leq s_l \quad \text{for } l = 1, \dots, r.$$

In this case x is called $(\boldsymbol{s}, \mathbf{M})$-sparse.

Next, we consider how to compute a finite dimensional approximation in $\mathcal{R}_M$. Let $y_k = \langle f, s_k \rangle_{\mathcal{H}}$ and $x_k = \langle f, r_k \rangle_{\mathcal{H}}$. Then $y = (y_k)_{k\in\mathbb{N}} \in \ell^2(\mathbb{N})$ and $x = (x_k)_{k\in\mathbb{N}} \in \ell^2(\mathbb{N})$ represents the coefficients of f with respect to two different bases. Furthermore, we have that

$$y = Ux, \qquad \text{where } (U)_{ij} = \langle r_j, s_i \rangle_{\mathcal{H}}\,. \tag{1.17}$$

Here U is a bounded linear operator on $\ell^2(\mathbb{N})$, and can be viewed as a *change of coordinates matrix* between $\{r_k\}_{k\in\mathbb{N}}$ and $\{s_k\}_{k\in\mathbb{N}}$. We notice in particular that U is unitary as both bases are orthonormal.

Recall that for a set $\Omega \subseteq \{1, \dots, N\}$ we defined $P_\Omega \colon \mathbb{C}^N \to \mathbb{C}^N$ as the projection onto the canonical basis, that is $(P_\Omega z)_i = z_i$ if $i \in \Omega$ and zero otherwise. We extend P_Ω to act on sequences in $\ell^2(\mathbb{N})$ in the natural way. Moreover, we use the notation $P_N = P_{\{1,\dots,N\}}$.

The key to obtain a good approximation to a function $f \in \mathcal{H}$, is to discretise the infinite-dimensional problem in the right way. In infinite-dimensional compressive sensing, this is achieved by choosing a set $\Omega \subset \{1,\dots,N\}$, of size $|\Omega| = m$ (here $N \geq M$) and consider scaled measurements, namely

$$\tilde{y} = DP_\Omega y \in \mathbb{C}^m \quad \text{where } D \in \mathbb{R}^{m\times m} \text{ is a diagonal scaling matrix.}$$

Furthermore, we use the finite dimensional sampling matrix

$$A = DP_\Omega U P_M \in \mathbb{C}^{m\times M}, \tag{1.18}$$

as a model of our sampling modality.

The matrix D above is present for technical reasons, and it has explicit entries, depending on Ω, see Paper III and Equation (III.6) for its definition. We note that by replacing $P_\Omega U$ with $P_\Omega U P_M$ we introduce a truncation error

$$\tilde{y} - Ax = DP_\Omega U P_M^\perp x.$$

This error is inevitable, and occurs whenever we consider a finite dimensional approximation to a function in $\mathcal{H}$. However, if $\mathcal{R}_M$ is chosen so that f can be well approximated by functions in $\mathcal{R}_M$, then this truncation error is small. This is the case for most wavelet spaces, where the scaling space and the first few detail spaces normally capture the important characteristics of not overly irregular functions.

Next, let $\boldsymbol{\omega} = \{\omega_1,\dots,\omega_M\} \subset \mathbb{R}_+$ be positive weights, and define the weighted ℓ^1-norm on $\mathbb{C}^M$ as

$$\|z\|_{1,\boldsymbol{\omega}} = \sum_{i=1}^{M} \omega_i |z_i|, \quad z \in \mathbb{C}^M.$$

Let also

$$\hat{x} \in \operatorname{argmin}_{z\in\mathbb{C}^M} \|z\|_{1,\boldsymbol{\omega}} \quad \text{subject to} \quad \|Az - \tilde{y}\|_{\ell^2} \leq \eta + \eta', \tag{1.19}$$

where $\tilde{y} = DP_\Omega y + e$ and the noise parameters $\eta \geq \|e\|_{\ell^2}$ and $\eta' \geq \|DP_\Omega U P_M^\perp x\|_{\ell^2}$.

In Paper III we consider $\mathcal{H} = L^2([0,1])$, and we derive conditions for choosing the sampling set Ω, weights $\boldsymbol{\omega}$, diagonal matrix D, N in relation to M, so that any minimiser $\hat{x}$ of (1.19) satisfies the bound

$$\|P_M x - \hat{x}\|_{\ell^2} \leq (1 + r^{1/4}) \left(C \frac{\sigma_{\mathbf{s},\mathbf{M}}(P_M x)_{1,\boldsymbol{\omega}}}{\sqrt{r}} + D \frac{1}{\sqrt{\theta}} (\eta + \eta') \right)$$

with high probability. Here $C, D > 0$ are explicit constants, r is the number of sparsity levels, $0 < \theta < 1$ depends on the relation $N \geq M$, and

$$\sigma_{\mathbf{s},\mathbf{M}}(P_M x)_{1,\boldsymbol{\omega}} = \inf\{\|P_M x - z\|_{1,\boldsymbol{\omega}} : z \text{ is } (\mathbf{s},\mathbf{M})\text{-sparse}\}$$

is the weighted distance to a $(\mathbf{s}, \mathbf{M})$-sparse vector.

Notice that the structure of this recovery guarantee is similar to the recovery guarantee we have already seen in Theorem 1.2.2. As we shall see in Paper III, a sufficient condition for the bound in (1.19) to hold, is that the matrix A in (1.18) satisfies what we call the *weighted robust null-space property in levels* (see Def. III.5.1).

1.9 Wavelets

1.9.1 The wavelet setup

Wavelets are built around nested sets of subspaces. Its construction start with a scaling function satisfying the properties of a multiresolution analysis (see e.g. [52]).

Definition 1.9.1. A multiresolution analysis (MRA) is a sequence of closed subspaces V_j, $j \in \mathbb{Z}$ of $L^2(\mathbb{R})$ satisfying

(i) $V_j \subset V_{j+1}$ for all $j \in \mathbb{Z}$,

(ii) $f \in V_j$ if and only if $f(2\cdot) \in V_{j+1}$, for all $j \in \mathbb{Z}$,

(iii) $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$[4],

(iv) $\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$,

(v) there exists a function $\phi \in V_0$ such that $\{\phi(\cdot - n) : n \in \mathbb{Z}\}$ is an orthonormal basis of V_0[5].

Let $\phi_{j,n} = 2^{j/2}\phi(2^j \cdot - n)$ denote a translated and dilated version of a scaling function ϕ, satisfying the properties of an MRA. We denote the subspaces

$$V_j = \operatorname{span}(\phi_{j,n} : n \in \mathbb{Z}), \quad \text{and} \quad W_j = \operatorname{span}(\psi_{j,n} : n \in \mathbb{Z}),$$

where $\psi \colon \mathbb{R} \to \mathbb{R}$ is a wavelet we would like to construct and $\psi_{j,n} = 2^{j/2}\psi(2^j \cdot - n)$. The wavelet ψ is constructed so that $V_{j+1} = V_j \oplus W_j$ for all $j \in \mathbb{Z}$. Here $\oplus$ denotes the direct sum of two vector spaces. We assume this setup and to make the following discussion coherent with the previous sections, we also assume that the functions $\phi_{j,n}$ and $\psi_{j,n}$ are orthonormal, with compact support. In this case we have that

$$W_j = \{f \in V_{j+1} : f \perp g, \forall g \in V_j\}$$

is the orthogonal complement of V_j in V_{j+1}. Moreover, for any $j_0 \in \mathbb{Z}$ we have that

$$\{\phi_{j_0,n} : n \in \mathbb{Z}\} \cup \{\psi_{j,n} : j \geq j_0, n \in \mathbb{Z}\}$$

[4] This condition is implied by condition (i), (ii) and (v).

[5] Some authors require $\{\phi(\cdot - n) : n \in \mathbb{Z}\}$ to be a Riesz basis, but such a definition can be shown to be equivalent to the one above. See, e.g., [38, Sec. 2.1].

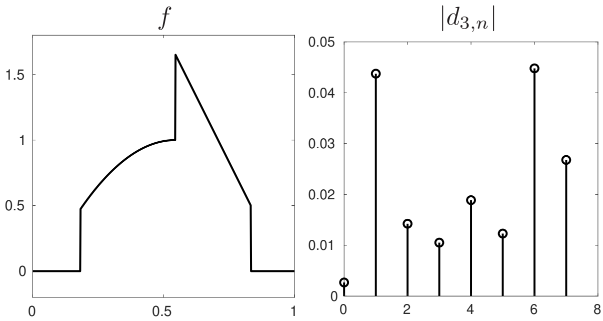


Figure 1.11: (**Wavelet coefficients are sparse at fine scales**). We show the piecewise smooth function f (left), and the magnitude of its wavelet coefficients $d_{j,n} = \langle f, \psi_{j,n} \rangle$ at scale $j = 3$ (middle) and scale $j = 6$ (right). The wavelet ψ is an orthonormal wavelet with $\nu = 2$ vanishing moments.

is an orthonormal basis for $L^2(\mathbb{R})$.

For a wavelet $\psi_{j,n}$, the coefficient $d_{j,n} = \langle f, \psi_{j,n} \rangle$ describes the local behaviour of $f \in L^2(\mathbb{R})$ in the area where $\psi_{j,n}$ is supported. As j grows, the support of $\psi_{j,n}$ shrinks. This means that, if j is large, then the coefficient $d_{j,n}$ contains more localised information about f, than if j is small. Generally, we shall be interested in sparse or compressible representations of f, and we seek a ψ so that the coefficients $\{d_{j,n}\}_{n\in\mathbb{Z}}$ are small at *fine* scales, i.e., for large j.

To achieve this, it is common to construct ψ so that it is orthogonal to all polynomials of degree $\nu - 1$. That is,

$$\int_{\mathbb{R}} t^k \psi(t) \mathrm{d}t = 0, \quad \text{for } k = 0, \dots, \nu - 1. \tag{1.20}$$

We say that ψ has ν *vanishing moments* if (1.20) is satisfied.

Now, notice that if f is sufficiently regular around $\operatorname{supp}(\psi_{j,n})$, then f can be well approximated by a polynomial of degree $\nu-1$, and we get $d_{j,n} = \langle f, \psi_{j,n} \rangle \approx 0$ as desired. However, if f is less regular, say discontinuous at some point $t \in \operatorname{supp}(\psi_{j,n})$, then the magnitude of the coefficient $d_{j,n}$ is typically large. An example can be seen in Figure 1.11, where we see that at the fine scale $j = 6$, we have $d_{j,n} = 0$ in any area where f is smooth. For $j = 3$, the support of $\psi_{j,n}$, is large enough for all of the $d_{j,n}$'s to notice the discontinuities of f, and we do not get a sparse representation. In images these type of irregularities are found around lines and edges. An example can be seen in Figure 1.2.

We note for completeness that if ψ have ν vanishing moments then the minimal support of this wavelet is $2\nu - 1$ (see, e.g., [24]). The same holds for ϕ, and we assume without loss of generality that $\operatorname{supp}(\psi) = \operatorname{supp}(\phi) = [-\nu + 1, \nu]$.

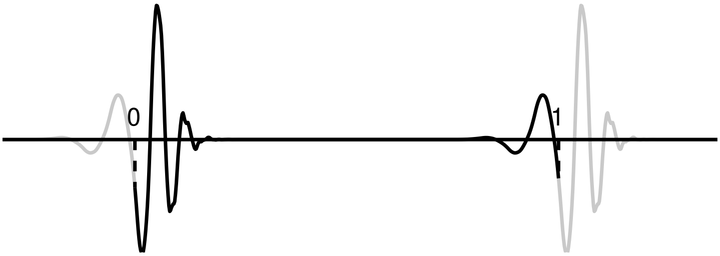


Figure 1.12: (**A periodically extended wavelet**). We see the orthonormal Daubechies wavelet $\psi^{\text{per}}_{4,0} = \phi_{4,0}|_{[0,1]} + \psi_{4,2^4}|_{[0,1]}$ for $\nu = 5$ vanishing moments.

1.9.2 Wavelets on the interval

In imaging, we typically work with functions on an interval, which we take to be $[0, 1]$ for simplicity. In this case, we need to create wavelet bases on $L^2([0, 1])$ rather than $L^2(\mathbb{R})$. It is natural to reuse as much of the old bases as possible, but to build orthonormal bases, it is necessary to replace all wavelets and scaling functions intersecting the boundary. To ensure that at least one $\phi_{j,n}$ is fully contained in the interval $[0, 1]$, let $j_0 \geq \log_2(2\nu)$[6], and only consider $j \geq j_0$. One way of retaining orthonormality is by periodically extending all functions intersecting the boundaries. That is, we define the periodically extended functions as

$$\begin{aligned} \phi^{\text{per}}_{j,n} &= \phi_{j,n}|_{[0,1]} + \phi_{j,2^j+n}|_{[0,1]} && \text{for } n = 0, \ldots, \nu - 2, \\ \phi^{\text{per}}_{j,n} &= \phi_{j,n}|_{[0,1]} + \phi_{j,n-2^j}|_{[0,1]} && \text{for } n = 2^j - \nu + 1, \ldots, 2^j - 1, \end{aligned}$$

and similarly for $\psi^{\text{per}}_{j,n}$. Here $|_{[0,1]}$ means the restriction to the interval $[0, 1]$. These periodically extended functions are orthonormal and thus, if we define

$$B^{\text{per}}_{\phi,j} = \{\phi^{\text{per}}_{j,n}\}_{n=0}^{\nu-2} \cup \{\phi_{j,n}\}_{n=\nu-1}^{2^j-\nu} \cup \{\phi^{\text{per}}_{j,n}\}_{n=2^j-\nu+1}^{2^j-1}$$

for each $j \geq j_0$, and similarly for $B^{\text{per}}_{\psi,j}$, then $B^{\text{per}}_{\phi,j} \cup \left(\bigcup_{s \geq j} B^{\text{per}}_{\psi,s}\right)$ is an orthonormal basis for $L^2([0, 1])$. An example of the wavelet $\psi^{\text{per}}_{4,0}$ can be seen in Figure 1.12, for $\nu = 5$ vanishing moments.

Whenever we restrict orthonormal wavelets to an interval, the easiest way of preserving orthonormality is to extend the wavelet basis periodically at the boundaries. The drawback, however, is that the wavelet basis no longer has the vanishing moments property. For non-periodic images, this creates a few extra high amplitude coefficients at each scale j. Moreover, as we can see in Figure 1.13, the underlying periodic assumption also leads to less accurate function approximation of non-periodic functions.

In [21] Cohen, Daubechies and Vial construct orthonormal wavelet bases on the interval which preserve vanishing moments. Similar extensions exist

[6] For $\nu = 1$ we can take $j_0 \geq 0$.

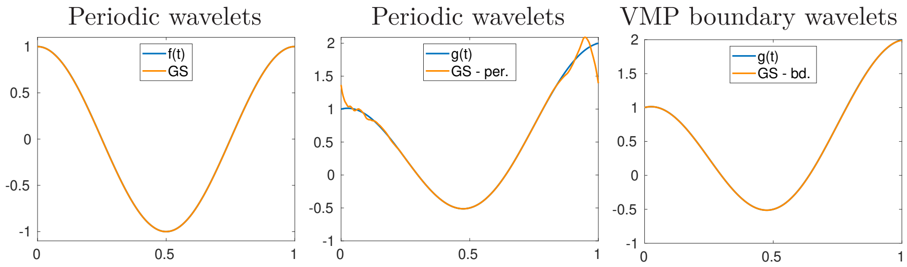


Figure 1.13: (**Periodic boundary conditions are not optimal for non-periodic functions**). We consider a Walsh sampling basis $\{s_k = w_{k-1}\}_{k\in\mathbb{N}}$, with $N = 32$ Walsh samples, and a wavelet reconstruction basis $\{r_k\}$ on $[0,1]$ with $\nu = 4$ vanishing moments and $M = 16$ basis functions. We consider periodic and vanishing moment preserving (VMP) boundary extensions. Let $y_k = \langle f, s_k\rangle_{L^2([0,1])}$, $k = 1,\dots,N$ and let $y \in \mathbb{C}^N$ denote the measurement vector. Let $\hat{x} = \operatorname{argmin}_{x\in\mathbb{C}^M} \|y - P_N U P_M x\|_{\ell^2}^2$, be the least-squares solution, where U is the matrix from (1.17). We then compute an approximation $\tilde{f} = \sum_{k=1}^{M} \hat{x}_k r_k$ to f, in $\mathcal{R}_M$. This is known as the generalised sampling approximation [2]. To the left we consider the 1-periodic function $f(t) = \cos(2\pi t)$. In the middle and right plots we consider $f(t) = \cos(2\pi t) + t$. What is clear is that for non-periodic functions, we get artefacts at the boundary due to the underlying discontinuity. Vanishing moment preserving wavelets avoid this and provide a more accurate reconstruction.

for biorthogonal wavelets [23, 60]. While these wavelets have attractive properties which might lead to better compression and more accurate function approximation, most implementations do not support them. Moreover, the few which actually do, all have certain restrictions, such as limited support for DWT computations of vectors with non-standard lengths (i.e., other than 2^m), or they are restricted to certain wavelets.

Paper IV unify the constructions of wavelets on the interval, both for orthonormal and biorthogonal wavelets. This is done in the language of linear algebra, which helps translating the constructions of these wavelets to software. The accompanying software is built on the implementation from [62, 63] and is made general enough to support a wide range of different vanishing moment preserving orthonormal and biorthogonal wavelets on the interval. It also accepts DWT computations of non-standard vector lengths.

1.10 False structures

In 1904, an old German schoolteacher, Herr von Osten, trained a horse named Hans to do arithmetics and light intellectual tasks. The horse could count up to one hundred, walk over to an object if you named it, or tell you the number of a playing card. The horse could not talk, but tapped its right foot n times, followed by a final tap with the left foot, to communicate the number n. For natural numbers the translation to taps was immediate, but for other tasks, such as naming playing cards, a translation was necessary, aces where one tap, kings two, and so forth. [15, p. 111–113]

During the summer of 1904, the horse became a hot topic in Berlin and in September news of the horse reached the New York times [39]. At the same time in Berlin, a commission was put together for examining the horse. The commission consisted of two teachers, two zoo directors, two military majors, a circus manager, a veterinarian, a horseman, and two academics, one of which was Professor Carl Stumpf, at that time director of the Psychological Institute of Berlin. [15, p. 177]. The commission mostly tested Hans accuracy and the limits of his intelligence with and without Herr von Osten being present. On 12 September the same year, the commission, somewhat hurried, issued a report saying that no trickery was found. Despite not finding any trickery, the commission did not rule out that any other factors were involved, and they could not decide whether Hans was clever or not [15, p. 177-118] (see also [78]).

During the autumn the investigation was continued by Prof. Stumpf's student, Oskar Pfungst, who started to investigate the relationship between the horse and humans. Pfungst discovered that Hans was unreliable whenever a new person asked him a question. However, if a person familiar to Hans was around, or he got to know the person asking the questions more thoroughly, Hans would answer correctly. Another test Pfungst made, was to check how Hans responded whenever the questioner did not know the answer. It turned out that in this case, the horse would consistently tap too many times. From this and several other tests, Pfungst concluded that Hans could not think independently of humans. What Pfungst discovered, was that the horse was picking up on delicate human-originated cues. These cues could be a person bending slightly forward when asking the question, or making a small unwanted upward jerk of the head, when the desired number of taps was reached[7].

We believe the same phenomenon, where the horse seems to be intelligent, but is in fact picking up a different structure, is currently happening in modern AI. This can cause high accuracy on a test set, yet incredible instabilities. In Paper V, we formally define what we mean by the word *structure*, and the term *false structure*, and we formulate a possible explanation to what AI algorithms actually learn as a conjecture. We note that this conjecture can never be proven in the traditional sense of the word, but must be established with a combination of theoretical and computational results, in the same way as one does in physics. We then discuss the potential consequences of the conjecture and provide two numerical examples demonstrating how difficult it is to learn the true structure.

[7] For a more thorough description of the experiments, followed by a discussion of whether these were adequate for drawing these conclusions we refer to [15, Ch. 5].

References

[1] Abdoli, S. et al. ‘Universal adversarial audio perturbations’. In: *arXiv:1908.03173* (2019).

[2] Adcock, B. and Hansen, A. C. ‘A generalized sampling theorem for stable reconstructions in arbitrary bases’. In: *Journal of Fourier Analysis and Applications* vol. 18, no. 4 (2012), pp. 685–716.

[3] Adcock, B. and Hansen, A. C. ‘Generalized sampling and infinite-dimensional compressed sensing’. In: *Foundations of Computational Mathematics* vol. 16, no. 5 (2016), pp. 1263–1323.

[4] Adcock, B., Hansen, A. C., Poon, C. and Roman, B. ‘Breaking the coherence barrier: A new theory for compressed sensing’. In: *Forum of Mathematics, Sigma*. Vol. 5. Cambridge University Press. 2017.

[5] Akhtar, N. and Mian, A. ‘Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey’. In: *IEEE Access* vol. 6 (2018), pp. 14410–14430.

[6] Antun, V., Renna, F., Poon, C., Adcock, B. and Hansen, A. C. ‘On instabilities of deep learning in image reconstruction and the potential costs of AI’. In: *Submitted* (2019).

[7] Arridge, S., Maass, P., Öktem, O. and Schönlieb, C.-B. ‘Solving inverse problems using data-driven models’. In: *Acta Numerica* vol. 28 (2019), pp. 1–174.

[8] Baker, N. et al. ‘Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence’. In: *U.S. Department of Energy Advanced Scientific Computing Research* (2019).

[9] Bastounis, A. and Hansen, A. C. ‘On the Absence of Uniform Recovery in Many Real-World Applications of Compressed Sensing and the Restricted Isometry Property and Nullspace Property in Levels’. In: *SIAM Journal on Imaging Sciences* vol. 10, no. 1 (2017), pp. 335–371.

[10] Behzadan, V. and Munir, A. ‘Vulnerability of deep reinforcement learning to policy induction attacks’. In: *International Conference on Machine Learning and Data Mining in Pattern Recognition*. Springer. 2017, pp. 262–275.

[11] Bojarski, M. et al. ‘End to end learning for self-driving cars’. In: *arXiv:1604.07316* (2016).

[12] Bredies, K., Kunisch, K. and Pock, T. ‘Total generalized variation’. In: *SIAM Journal on Imaging Sciences* vol. 3, no. 3 (2010), pp. 492–526.

[13] Candès, E. J., Romberg, J. and Tao, T. ‘Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information’. In: *IEEE Transactions Information Theory* vol. 52, no. 2 (2006), pp. 489–509.

[14] Candès, E. J. and Donoho, D. L. ‘Curvelets: A surprisingly effective nonadaptive representation for objects with edges’. In: *Curves and Surfaces*. Ed. by Schumaker, L. L. et al. Vanderbilt University Press, 1999.

[15] Candland, D. K. *Feral children and clever animals: Reflections on human nature*. Oxford University Press, 1993.

[16] Carlini, N. and Wagner, D. ‘Audio adversarial examples: Targeted attacks on speech-to-text’. In: *2018 IEEE Security and Privacy Workshops*. IEEE. 2018, pp. 1–7.

[17] Carlini, N. et al. ‘Hidden voice commands’. In: *25th {USENIX} Security Symposium*. 2016, pp. 513–530.

[18] Chávez, C. E., Alonzo-Atienza, F. and Alvarez, D. ‘Avoiding the inverse crime in the inverse problem of electrocardiography: estimating the shape and location of cardiac ischemia’. In: *Computing in Cardiology 2013*. IEEE. 2013, pp. 687–690.

[19] Chen, H. et al. ‘LEARN: Learned experts’ assessment-based reconstruction network for sparse-data CT’. In: *IEEE transactions on medical imaging* vol. 37, no. 6 (2018), pp. 1333–1347.

[20] Chen, T. Q., Rubanova, Y., Bettencourt, J. and Duvenaud, D. K. ‘Neural Ordinary Differential Equations’. In: *Advances in Neural Information Processing Systems 31*. Ed. by Bengio, S. et al. Curran Associates, Inc., 2018, pp. 6571–6583.

[21] Cohen, A., Daubechies, I. and Vial, P. ‘Wavelets on the interval and fast wavelet transforms’. In: *Applied and Computational Harmonic Analysis* vol. 1, no. 1 (1993), pp. 54–81.

[22] Dahl, G. E., Yu, D., Deng, L. and Acero, A. ‘Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition’. In: *IEEE Transactions on audio, speech, and language processing* vol. 20, no. 1 (2011), pp. 30–42.

[23] Dahmen, W., Kunoth, A. and Urban, K. ‘Biorthogonal Spline Wavelets on the Interval – Stability and Moment Conditions’. In: *Applied and Computational Harmonic Analysis* vol. 6, no. 2 (1999), pp. 132–196.

[24] Daubechies, I. *Ten lectures on wavelets*. Vol. 61. SIAM, 1992.

[25] Daws Jr, J. and Webster, C. G. ‘A Polynomial-Based Approach for Architectural Design and Learning with Deep Neural Networks’. In: *arXiv:1905.10457* (2019).

[26] Donoho, D. L. ‘Compressed sensing’. In: *IEEE Transactions Information Theory* vol. 52, no. 4 (2006), pp. 1289–1306.

[27] E, W., Han, J. and Jentzen, A. ‘Deep Learning-Based Numerical Methods for High-Dimensional Parabolic Partial Differential Equations and Backward Stochastic Differential Equations’. In: *Communications in Mathematics and Statistics* vol. 5, no. 4 (2017), pp. 349–380.

[28] Eykholt, K. et al. ‘Robust Physical-World Attacks on Deep Learning Visual Classification’. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 1625–1634.

[29] Fawzi, A., Moosavi-Dezfooli, S.-M. and Frossard, P. ‘The robustness of deep networks: A geometrical perspective’. In: *IEEE Signal Processing Magazine* vol. 34, no. 6 (2017), pp. 50–62.

[30] Finlayson, S. G. et al. ‘Adversarial attacks on medical machine learning’. In: *Science* vol. 363, no. 6433 (2019), pp. 1287–1289.

[31] Foucart, S. and Rauhut, H. *A mathematical introduction to compressive sensing*. Birkhauser, 2013.

[32] Gao, H. and Oates, T. ‘Universal Adversarial Perturbation for Text Classification’. In: *arXiv:1910.04618* (2019).

[33] Guerquin-Kern, M., Lejeune, L., Pruessmann, K. P. and Unser, M. ‘Realistic analytical phantoms for parallel magnetic resonance imaging’. In: *IEEE Transactions on Medical Imaging* vol. 31, no. 3 (2012), pp. 626–636.

[34] Gupta, H., Jin, K. H., Nguyen, H. Q., McCann, M. T. and Unser, M. ‘CNN-based projected gradient descent for consistent CT image reconstruction’. In: *IEEE transactions on medical imaging* vol. 37, no. 6 (2018), pp. 1440–1453.

[35] Hamon, R., Junklewitz, H. and Sanchez, I. *Robustness and Explainability of Artificial Intelligence: From technical to policy solutions*. Tech. rep. European Commission, Jan. 2020.

[36] He, K., Zhang, X., Ren, S. and Sun, J. ‘Deep residual learning for image recognition’. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[37] Heaven, D. ‘Why deep-learning AIs are so easy to fool’. In: *Nature* vol. 574, no. 7777 (2019), p. 163.

[38] Hernández, E. and Weiss, G. *A first course on wavelets*. CRC press, 1996.

[39] Heyen, E. T. ‘Berlin’s wonderful horse; He Can Do Almost Everything but Talk – How He Was Taught’. In: *The New York Times* (Sept. 1904). https://timesmachine.nytimes.com/timesmachine/1904/09/04/101396572.pdf, visited 11.02.2020.

[40] Hinton, G. et al. ‘Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups’. In: *IEEE Signal processing magazine* vol. 29, no. 6 (2012), pp. 82–97.

[41] Huang, Y. et al. ‘Data consistent artifact reduction for limited angle tomography with deep learning prior’. In: *International Workshop on Machine Learning for Medical Image Reconstruction*. Springer. 2019, pp. 101–112.

[42] Huang, Y. et al. ‘Some investigations on robustness of deep learning in limited angle tomography’. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer. 2018, pp. 145–153.

[43] Ilahi, I. et al. ‘Challenges and Countermeasures for Adversarial Attacks on Deep Reinforcement Learning’. In: *arXiv:2001.09684* (2020).

[44] Jin, K. H., McCann, M. T., Froustey, E. and Unser, M. ‘Deep convolutional neural network for inverse problems in imaging’. In: *IEEE Transactions Image Processing* vol. 26, no. 9 (2017), pp. 4509–4522.

[45] Kanbak, C., Moosavi-Dezfooli, S.-M. and Frossard, P. ‘Geometric robustness of deep networks: analysis and improvement’. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2018, pp. 4441–4449.

[46] Krizhevsky, A., Sutskever, I. and Hinton, G. E. ‘Imagenet classification with deep convolutional neural networks’. In: *Advances in Neural Information Processing Systems.* 2012, pp. 1097–1105.

[47] LeCun, Y., Bengio, Y. and Hinton, G. ‘Deep learning’. In: *Nature* vol. 521, no. 7553 (2015), p. 436.

[48] Li, C. and Adcock, B. ‘Compressed sensing with local structure: uniform recovery guarantees for the sparsity in levels class’. In: *Applied and Computational Harmonic Analysis* vol. 46, no. 3 (2019), pp. 453–477.

[49] Liang, B. et al. ‘Deep text classification can be fooled’. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence.* 2017.

[50] Lim, W.-Q. ‘The discrete shearlet transform: A new directional transform and compactly supported shearlet frames’. In: *IEEE Transactions on image processing* vol. 19, no. 5 (2010), pp. 1166–1180.

[51] Lønning, K. et al. ‘Recurrent inference machines for reconstructing heterogeneous MRI data’. In: *Medical image analysis* vol. 53 (2019), pp. 64–78.

[52] Mallat, S. *A wavelet tour of signal processing: The sparse way.* Third. Academic Press, 2008.

[53] McCann, M. T., Jin, K. H. and Unser, M. ‘Convolutional neural networks for inverse problems in imaging: A review’. In: *IEEE Signal Processing Magazine* vol. 34, no. 6 (2017), pp. 85–95.

[54] Moosavi-Dezfooli, S., Fawzi, A., Fawzi, O. and Frossard, P. ‘Universal Adversarial Perturbations’. In: *IEEE Conference on computer vision and pattern recognition.* July 2017, pp. 86–94.

[55] Moosavi-Dezfooli, S., Fawzi, A. and Frossard, P. ‘DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks’. In: *IEEE Conference on Computer Vision and Pattern Recognition.* June 2016, pp. 2574–2582.

[56] Natterer, F. and Wübbeling, F. *Mathematical methods in image reconstruction.* Vol. 5. SIAM, 2001.

[57] Nguyen, A., Yosinski, J. and Clune, J. 'Deep neural networks are easily fooled: High confidence predictions for unrecognizable images'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 427–436.

[58] Pan, X., Sidky, E. Y. and Vannier, M. 'Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction?' In: *Inverse problems* vol. 25, no. 12 (2009), p. 123009.

[59] Pinkus, A. 'Approximation theory of the MLP model in neural networks'. In: *Acta numerica* vol. 8 (1999), pp. 143–195.

[60] Primbs, M. 'New Stable Biorthogonal Spline-Wavelets on the Interval'. In: *Results in Mathematics* vol. 57, no. 1-2 (2010), pp. 121–162.

[61] Rudy, S. H., Brunton, S. L., Proctor, J. L. and Kutz, J. N. 'Data-driven discovery of partial differential equations'. In: *Science Advances* vol. 3, no. 4 (2017).

[62] Ryan, Ø. *Linear algebra, signal processing, and wavelets. A unified approach. MATLAB version.* Springer, 2019.

[63] Ryan, Ø. *Linear algebra, signal processing, and wavelets. A unified approach. Python version.* Springer, 2019.

[64] Schlemper, J., Caballero, J., Hajnal, J. V., Price, A. and Rueckert, D. 'A deep cascade of convolutional neural networks for MR image reconstruction'. In: *International Conference on Information Processing in Medical Imaging.* Springer. 2017, pp. 647–658.

[65] Schwab, C. and Zech, J. 'Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ'. In: *Analysis and Applications* vol. 17, no. 01 (2019), pp. 19–55.

[66] Shepp, L. A. and Logan, B. F. 'The Fourier reconstruction of a head section'. In: *IEEE Transactions on nuclear science* vol. 21, no. 3 (1974), pp. 21–43.

[67] Silver, D. et al. 'Mastering the game of go without human knowledge'. In: *Nature* vol. 550, no. 7676 (2017), pp. 354–359.

[68] Song, D. et al. 'Physical adversarial examples for object detectors'. In: *12th {USENIX} Workshop on Offensive Technologies.* Aug. 2018.

[69] Song, P., Eldar, Y. C., Mazor, G. and Rodrigues, M. R. D. 'Magnetic Resonance Fingerprinting Using a Residual Convolutional Neural Network'. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* May 2019, pp. 1040–1044.

[70] Sriram, A. et al. 'GrappaNet: Combining Parallel Imaging with Deep Learning for Multi-Coil MRI Reconstruction'. In: *arXiv:1910.12325* (2019).

[71] Strack, R. ‘Imaging: AI transforms image reconstruction’. In: *Nature Methods* vol. 15, no. 5 (2018), p. 309.

[72] Studer, V. et al. ‘Compressive Fluorescence Microscopy for Biological and Hyperspectral Imaging’. In: *Proceedings of the National Academy of Sciences* vol. 109, no. 26 (2011), pp. 1679–1687.

[73] Su, J., Vargas, D. V. and Sakurai, K. ‘One Pixel Attack for Fooling Deep Neural Networks’. In: *IEEE Transactions on Evolutionary Computation* (2019).

[74] Sutskever, I., Vinyals, O. and Le, Q. V. ‘Sequence to sequence learning with neural networks’. In: *Advances in neural information processing systems.* 2014, pp. 3104–3112.

[75] Szegedy, C. et al. ‘Intriguing properties of neural networks’. In: *Proceedings of the International Conference on Learning Representations.* 2014.

[76] U.S. Food and Drug Administration. *510k premarket notification of Compressed Sensing Cardiac Cine (Siemens).* https://www.accessdata.fda.gov/cdrh_docs/pdf16/K163312.pdf. 2017.

[77] U.S. Food and Drug Administration. *510k premarket notification of HyperSense (GE Medical Systems).* https://www.accessdata.fda.gov/cdrh_docs/pdf16/K162722.pdf. 2017.

[78] Unknown Author. ‘“Clever Hans” Again. Expert Commission Decides That the Horse Actually Reasons’. In: *The New York Times* (Oct. 1904). https://timesmachine.nytimes.com/timesmachine/1904/10/02/120289067.pdf, visited 11.02.2020.

[79] Weinan, E. and Yu, B. ‘The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems’. In: *Communications in Mathematics and Statistics* vol. 6, no. 1 (2018), pp. 1–14.

[80] Xie, C. et al. ‘Adversarial examples for semantic segmentation and object detection’. In: *Proceedings of the IEEE International Conference on Computer Vision.* 2017, pp. 1369–1378.

[81] Xu, K. et al. ‘Representation Learning on Graphs with Jumping Knowledge Networks’. In: *Proceedings of Machine Learning Research.* Vol. 80. July 2018, pp. 5453–5462.

[82] Yarotsky, D. ‘Error bounds for approximations with deep ReLU networks’. In: *Neural Networks* vol. 94 (2017), pp. 103–114.

[83] Zhang, G. et al. ‘Dolphinattack: Inaudible voice commands’. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security.* 2017, pp. 103–117.

[84] Zhu, B., Liu, J. Z., Cauley, S. F., Rosen, B. R. and Rosen, M. S. ‘Image reconstruction by domain-transform manifold learning’. In: *Nature* vol. 555, no. 7697 (Mar. 2018), p. 487.

[85] Zomet, A. and Nayar, S. K. 'Lensless imaging with a controllable aperture'. In: *Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 1. IEEE. 2006, pp. 339–346.

Papers

Paper I

On instabilities of deep learning in image reconstruction and the potential costs of AI

Vegard Antun, Francesco Renna, Clarice Poon, Ben Adcock, Anders C. Hansen

Proceedings of the National Academy of Sciences, 2020.

Abstract

Deep learning, due to its unprecedented success in tasks such as image classification, has emerged as a new tool in image reconstruction with potential to change the field. In this paper we demonstrate a crucial phenomenon: deep learning typically yields unstable methods for image reconstruction. The instabilities usually occur in several forms: (1) certain tiny, almost undetectable perturbations, both in the image and sampling domain, may result in severe artefacts in the reconstruction, (2) a small structural change, for example a tumour, may not be captured in the reconstructed image and (3) (a counterintuitive type of instability) more samples may yield poorer performance. Our new stability test with algorithms and easy to use software detects the instability phenomena. The test is aimed at researchers to test their networks for instabilities and for government agencies, such as the Food and Drug Administration (FDA), to secure safe use of deep learning methods.

There are two paradigm changes currently happening: (1) AI is replacing humans in problem solving, however, (2) AI is also replacing the standard algorithms in computational science and engineering. Since reliable numerical calculations are paramount, algorithms for computational science are traditionally based on two pillars: accuracy and stability. This is in particular true of image reconstruction, which is a mainstay of computational science, providing fundamental tools in medical, scientific and industrial imaging. This paper is the first to demonstrate that the stability pillar is typically absent in current deep learning and AI-based algorithms for image reconstruction. This raises two fundamental questions: how reliable are such algorithms when applied in the sciences, and do AI-based algorithms have an unavoidable achilles heel:

Keywords: instability, deep learning, AI, image reconstruction, inverse problems.

instability? This paper introduces a comprehensive testing framework designed to demonstrate, investigate and ultimately answer these foundational questions.

The importance of stable and accurate methods for image reconstruction for inverse problems is hard to over estimate. These techniques form the foundation for essential tools across the physical and life sciences such as Magnetic Resonance Imaging (MRI), Computerised Tomography (CT), fluorescence microscopy, electron tomography, Nuclear Magnetic Resonance (NMR), radio interferometry, lensless cameras etc. Moreover, stability is traditionally considered a necessity in order to secure reliable and trustworthy methods used in, for example, cancer diagnosis. Hence, there is an extensive literature on designing stable methods for image reconstruction in inverse problems [11, 16, 19, 48].

Artificial intelligence (AI) techniques such as deep learning and neural networks [31] have provided a new paradigm with new techniques in inverse problems [18, 24, 32, 36, 37, 44, 47, 54, 56] that may change the field. In particular, the reconstruction algorithms learn how to best do the reconstruction based on training from previous data, and through this training procedure aim to optimise the quality of the reconstruction. This is a radical change from the current state of the art both from an engineering, physical and mathematical point of view.

AI and deep learning has already changed the field of computer vision and image classification [10, 14, 29, 55], where the performance is now referred to as super human [21]. However, the success comes with a price. Indeed, the methods are highly unstable. It is now well established [13, 26, 38, 39, 49] that high performance deep learning methods for image classification are subject to failure given tiny, almost invisible perturbation of the image. An image of a cat may be classified correctly, however, a tiny change, invisible to the human eye, may cause the algorithm to change its classification label from cat to a fire truck, or another label far from the original.

In this paper we establish the instability phenomenon of deep learning in image reconstruction for inverse problems. A potential surprising conclusion is that the phenomenon may be independent of the underlying mathematical model. For example, MRI is based on sampling the Fourier transform whereas CT is based on sampling the Radon transform. These are rather different models, yet the instability phenomena happen for both sampling modalities when using deep learning.

There is, however, a big difference between the instabilities of deep learning for image classification and our results on instabilities of deep learning for image reconstruction. Firstly, in the former case there is only one thing that could go wrong: a small perturbation results in a wrong classification. In image reconstruction there are several potential forms of instabilities. In particular, we consider three crucial issues: (1) *instabilities with respect to certain tiny perturbations*, (2) *instabilities with respect to small structural changes* (for example a brain image with or without a small tumour), (3) *instabilities with respect to changes in the number of samples*. Secondly, the two problems are totally unrelated. Indeed, the former problem is, in its simplest form, a decision problem, and hence the decision function ('is there a cat in the image?')

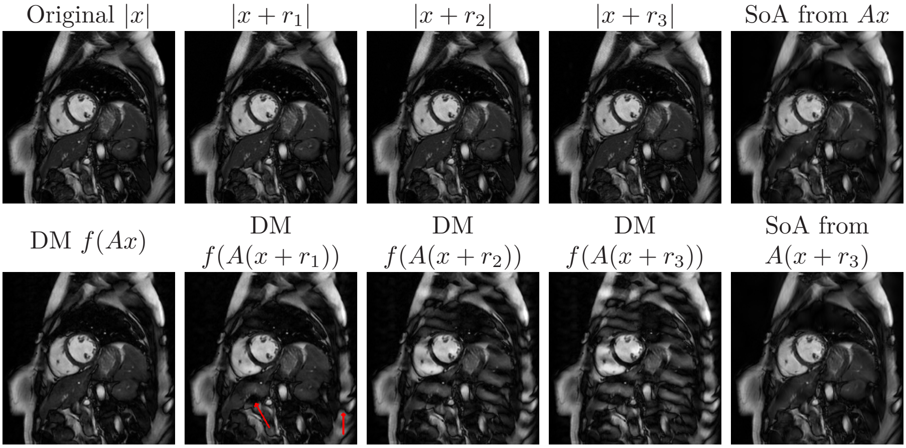


Figure I.1: Perturbations r_j (created to simulate worst-case effect) with $|r_1| < |r_2| < |r_3|$ are added to the image x. Upper row images (1)-(4): original image x and perturbations $x + r_j$. Lower row images (1)-(4) reconstruction from $A(x + r_j)$ using the Deep MRI (DM) network f, where A is a subsampled Fourier transform (33% subsampling), see Methods and SI for details. Rightmost column: reconstruction from Ax and $A(x + r_3)$ using a state-of-the-art (SoA) method, see Methods for details. Note how the artefacts (red arrows) are hard to dismiss as non-physical.

to be approximated is necessarily discontinuous. However, the problem of reconstructing an image from Fourier coefficients, as is the problem in MRI, is completely different. In this case there exist stable and accurate methods that depend continuously on the input. It is therefore paradoxical that deep learning leads to unstable methods for problems that can be solved accurately in a stable way (see §I.A.1.5 in the Supplementary Information (SI)).

The networks we have tested are unstable either in the form of (1) or (2) or both. Moreover, networks that are highly stable in one of the categories tend to be highly unstable in the other. The instability in form of (3), however, occur for some networks but not all. The new findings raise two fundamental questions:

(i) Does AI, as we know it, come at a cost? Is instability a necessary by-product of our current AI techniques?

(ii) Can reconstruction methods based on deep learning always be safely used in the physical and life sciences? Or, are there cases for which instabilities may lead to, for example, incorrect medical diagnosis if applied in medical imaging?

The scope of this paper is on the second question, as the first question is on foundations, and our stability test provides the starting point for answering

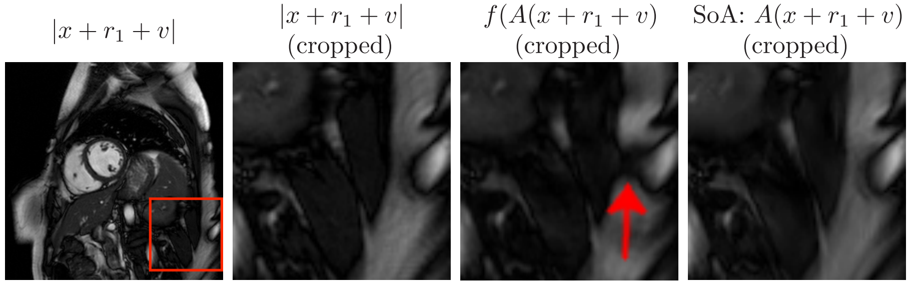

$|x + r_1 + v|$
$|x + r_1 + v|$ (cropped)
$f(A(x + r_1 + v)$ (cropped)
SoA: $A(x + r_1 + v)$ (cropped)

Figure I.2: A random Gaussian vector $e \in \mathbb{C}^m$ is computed by drawing (the real and imaginary part of) each component independently from the normal distribution $\mathcal{N}(0, 10)$. We let $v = A^*e$, and rescale v so that $\|v\|_2 = \frac{1}{4}\|r_1\|_2$, where r_1 is the perturbation from Figure I.1. The Deep MRI network f reconstructs from the measurements $A(x + r_1 + v)$ and shows the same artefact as was seen for r_1 in Figure I.1. Note that in this experiment $A \in \mathbb{C}^{m \times N}$ is a subsampled normalized discrete Fourier transform (33% subsampling), so that $AA^* = I$ i.e. $e = Av$.

question (ii). However, even if instabilities occur, this should not rule out the use of deep learning methods in inverse problems. In fact, one may be able to show, with large empirical statistical tests, that the artefacts caused by instabilities occur infrequently. As our test reveals, there is a myriad of different artefacts that may occur, as a result of the instabilities, suggesting vast efforts needed to answer (ii). A detailed account is in the conclusion.

I.1 The instability test

The instability test is based on the three instability issues mentioned above. We consider instabilities with respect to the following:

I.1.1 Tiny worst-case perturbations

The tiny perturbation could be in the image domain or in the sampling domain. When considering medical imaging, a perturbation in the image domain could come from a slight movement of the patient, small anatomic differences between people etc. The perturbation in the sampling domain may be caused by malfunctioning of the equipment or the inevitable noise dictated by the physical model of the scanning machine. However, a perturbation in the image domain may imply a perturbation in the sampling domain. Also, in many cases, the mathematical model of the sampling reveals that such a sampling process implies an operator that is surjective onto its range, and hence there exists a perturbation in the image domain corresponding to the perturbation in the sampling domain. Thus, a combination of all these factors may yield perturbations that in a worst case scenario may be quite specific, hard to model and hard to protect against unless one has a completely stable neural network.

The instability test includes algorithms that do the following. Given an image and a neural network, designed for image reconstruction from samples provided by a specific sampling modality, the algorithm searches for a perturbation of the image that makes the most severe change in the output of the network while still keeping the perturbation small. In a simple mathematical form this can be described as follows. Given an image $x \in \mathbb{R}^N$ (we interpret an image as a vector for simplicity), a matrix $A \in \mathbb{C}^{m\times N}$ representing the sampling modality (for example a discrete Fourier transform modelling MRI) and a neural network $f : \mathbb{C}^m \to \mathbb{C}^N$, the neural network reconstructs an approximation $\tilde{x}$ to x defined by $y = Ax$, where $\tilde{x} = f(y)$. The algorithm seeks an $r \in \mathbb{R}^N$ such that

$$\|f(y + Ar) - f(y)\| \text{ is large, while } \|r\| \text{ is small,}$$

see the methods section for details. However, the perturbation could, of course, be put on the measurement vector y instead.

I.1.2 Small structural changes in the image

By structural change we mean a change in the image domain that may not be tiny, and typically significant and clearly visible, however still small (for example a small tumour). The purpose is to check if the network can recover important details that are crucial in, for example, medical assessments. In particular, given the image $x \in \mathbb{R}^N$ we add a perturbation $r \in \mathbb{R}^N$, where r is a detail that is clearly visible in the perturbed image $x + r$, and check if r is still clearly visible in the reconstructed image

$$\hat{x} = f(A(x + r)).$$

In this paper we consider the symbols from cards as well as letters. In particular, we add the symbols $\spadesuit, \heartsuit, \diamondsuit, \clubsuit$ and the letters CAN U SEE IT to the image. The reason for this is that card symbols as well as letters are fine details that are hard to detect, and thus represent a reasonable challenge for the network. If the network is able to recover these small structural changes it is likely to recover other details of the same size. On the other hand, if the network fails on these basic changes, it is likely to fail on other details as well. The symbols can, of course, be specified depending on the specific application. Our choice is merely for illustration.

Important note: When testing stability, both with respect to tiny perturbations and with respect to small structural changes, the test is always done in comparison with a state-of-the-art (SoA abbreviated) stable method in order to check that any instabilities produced by the neural network is due to the network itself and not because of ill-conditioning of the inverse problem. The state-of-the-art methods used are based on compressed sensing and sparse regularisation [7, 9, 43]. These methods often come with mathematical stability guaranties [2], and are hence suitable as benchmarks (see the Methods for details).

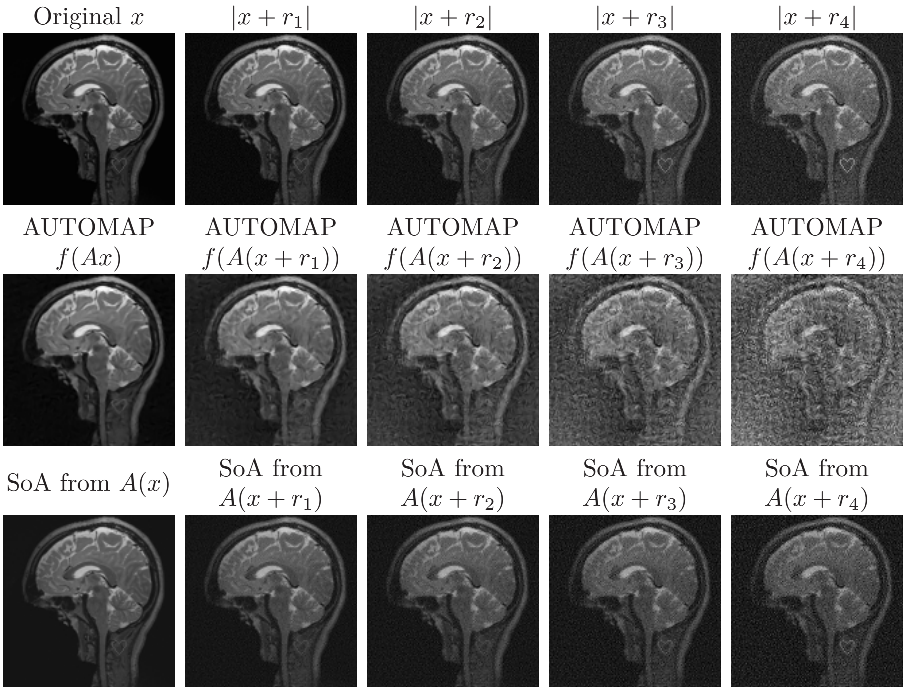


Figure I.3: Perturbations $\tilde{r}_j$ (created to simulate worst-case effect) are added to the measurements $y = Ax$, where $|\tilde{r}_1| < |\tilde{r}_2| < |\tilde{r}_3| < |\tilde{r}_4|$ and A is a subsampled Fourier transform (60% subsampling). To visualise we show $|x + r_j|$ where $y + \tilde{r}_j = A(x + r_j)$. Upper row: original image x with perturbations r_j. Middle row: reconstructions from $A(x + r_j)$ by the AUTOMAP network f. Lower row: reconstructions from $A(x + r_j)$ by a state of the art method (see Methods for details). A detail in form of a heart, with varying intensity, is added to visualise the loss in quality.

I.1.3 Changing the number of samples in the sampling device (such as the MRI or CT scanner)

Typical state-of-the-art methods share a common quality; more samples imply better quality of the reconstruction. Given that deep learning neural networks in inverse problems are trained given a specific sampling pattern, the question is how robust is the trained network with respect to changes in the sampling. The test checks whether the quality of the reconstruction deteriorates with more samples. This is a crucial question in applications. For example the recent implementation of compressed sensing on Philips MRI machines allows the user to change the under sampling ration for every scan. This means that if a network is trained on 25% subsampling, say, and suddenly the user changed the subsampling ratio to 35% one would want an improved recovery. If the quality deteriorates or stagnates with more samples, this means that one will have to produce networks trained for each and every combination of subsampling that the machine allows for. Finally, due to the other instability issues, every such network must individually be empirically statistically tested to detect whether the occurrence of instabilities is rare or not. It is not enough to test on only one neural network, as their instabilities may be completely different.

I.2 Testing the test

We test six deep learning neural networks selected based on their strong performance, wide range in architectures, difference in sampling patterns and subsampling ratios, as well as their difference in training data. The specific details about the architecture and the training data of the tested networks can be found in the supplementary information (SI).

Important note: The tests performed are not designed to test deep learning against state-of-the-art in terms of performance on specific images. The test is designed to detect the instability phenomenon. Hence, the comparison with state-of-the-art is only to verify that the instabilities are exclusive only to neural networks based on deep learning, and not due to an ill-conditioning of the problem itself. Moreover, as is clear from the images, in the unperturbed cases, the best performance varies between neural networks and state-of-the-art. The list of networks is as follows:

AUTOMAP [56]: This is a neural network for low resolution single coil MRI with 60% subsampling. The training set consists of brain images with added white noise to the Fourier samples.

DAGAN [54]: This network is for medium resolution single coil MRI with 20% subsampling, and is trained with a variety of brain images.

Deep MRI [44]: This neural network is for medium resolution single coil MRI with 33% subsampling. It is trained with detailed cardiac MR images.

Ell 50 [24]: Ell 50 is a network for CT or any Radon transform based inverse problem. It is trained on images containing solely ellipses (hence the name Ell 50). The number 50 refers to the number of lines used in the sampling in the sinogram.

Med 50 [24]: Med 50 has exactly the same architecture as Ell 50 and is used for CT, however, it is trained with medical images (hence the name Med 50) from the Mayo Clinic database. The number of lines used in the sampling from the sinogram is 50.

MRI-VN [18]: This network is for medium to high resolution parallel MRI with 15 coil elements and 15% subsampling. The training is done with a variety of knee images.

I.3 Stability with respect to tiny worst-case perturbations

Below follows the description of the test applied to some of the networks where we detect instabilities with respect to tiny perturbations.

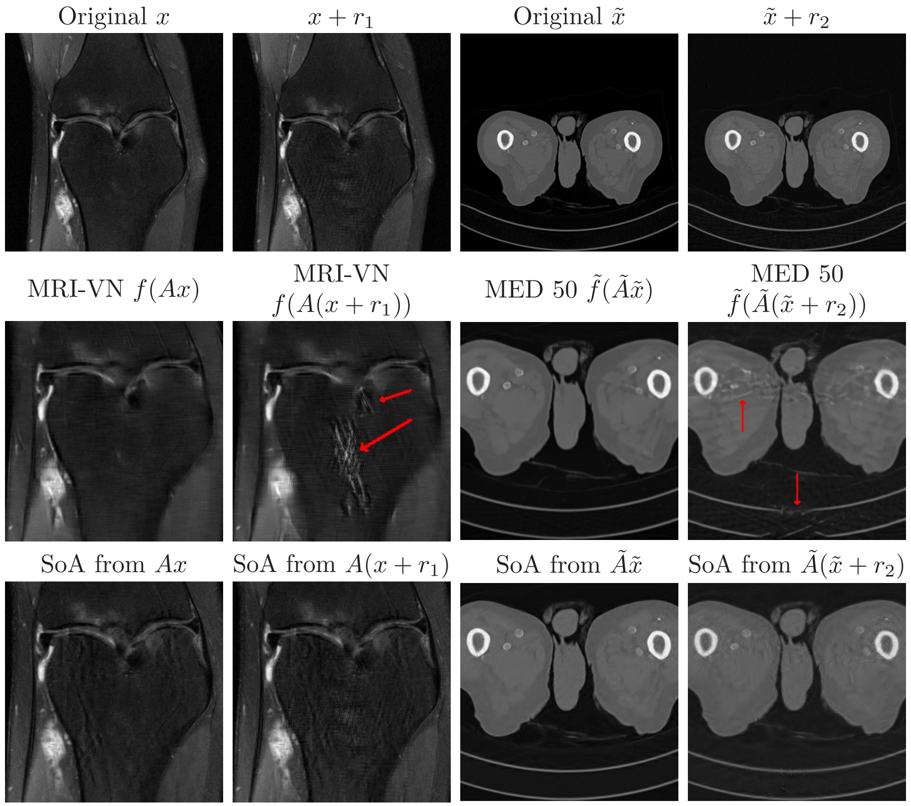

Original x
$x + r_1$
Original $\tilde{x}$
$\tilde{x} + r_2$
MRI-VN $f(Ax)$
MRI-VN $f(A(x + r_1))$
MED 50 $\tilde{f}(\tilde{A}\tilde{x})$
MED 50 $\tilde{f}(\tilde{A}(\tilde{x} + r_2))$
SoA from Ax
SoA from $A(x + r_1)$
SoA from $\tilde{A}\tilde{x}$
SoA from $\tilde{A}(\tilde{x} + r_2)$

Figure I.4: Perturbations r_1, r_2 (created to simulate worst-case effect) are added to the images x and $\tilde{x}$ in the first row. The reconstructions by the network f (MRI-VN), from Ax and $A(x + r_1)$, and the network $\tilde{f}$ (MED 50), from $\tilde{A}\tilde{x}$ and $\tilde{A}(\tilde{x} + r_2)$ are shown in the second row. A is a subsampled discrete Fourier transform and $\tilde{A}$ is a subsampled Radon transform. State-of-the-art (SoA) comparisons are shown in the last row.

Deep MRI: In this test we perturb the image x with a sequence of perturbations $\{r_j\}_{j=1}^{3}$ with $|r_1| < |r_2| < |r_3|$ in order to simulate how the instabilities continuously transform the reconstructed image from a very high quality reconstruction to an almost unrecognisable distortion. This is illustrated in the lower row of Figure I.1. Note that the perturbations are almost invisible to the human eye as demonstrated in the upper row of Figure I.1. The r_j perturbations are created by early stopping of the algorithm iterating to solve for the optimal worst case perturbation. The purpose of this experiment is to demonstrate how the gradual change in perturbation create artefacts that may be hard to verify as non-physical. Indeed, the worst case perturbation r_3 causes clearly a reconstruction that, in a real world situation, can be dismissed by a clinician as non-physical. However, for the smallest r_1 we have a perturbation

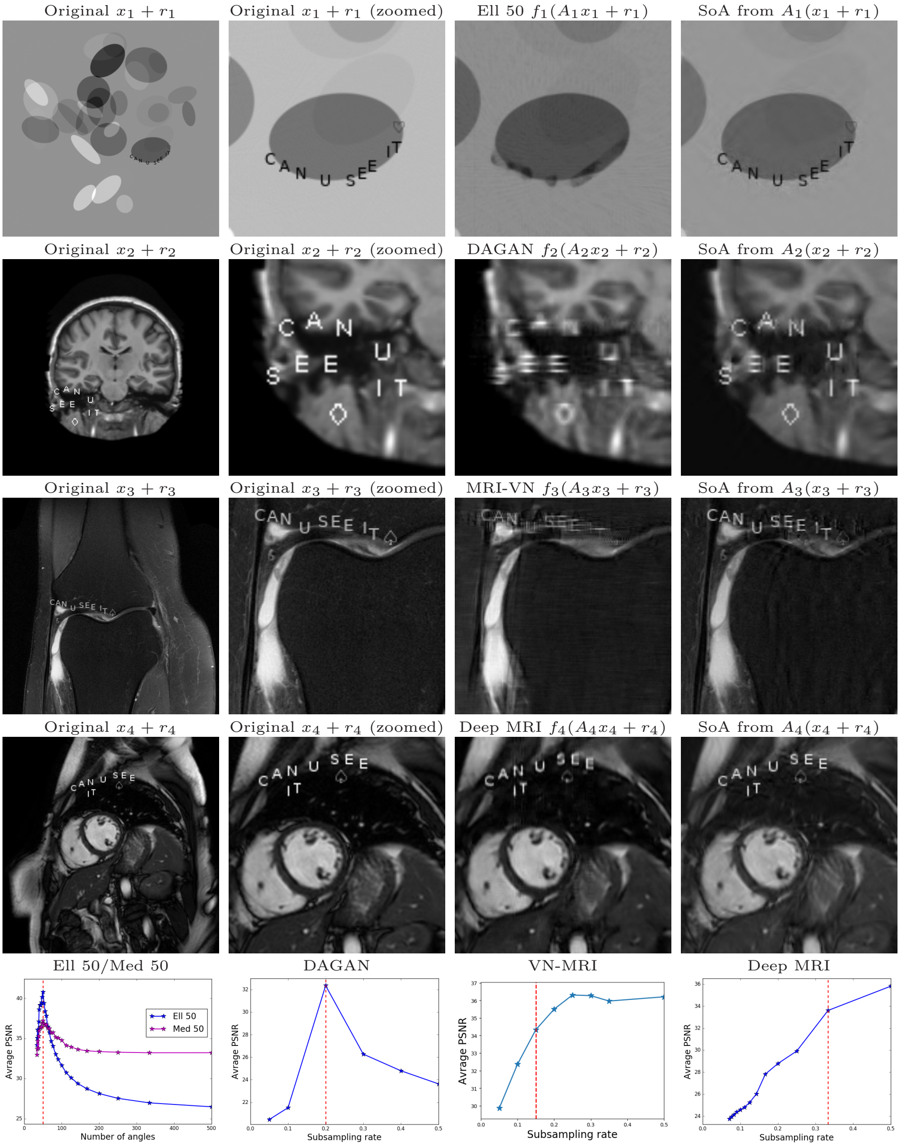


Figure I.5: First four rows: Images x_j plus structured perturbations r_j (in the form of text and symbols) are reconstructed from measurements $y_j = A_j(x_j + r_j)$ with neural networks f_j and state-of-the-art (SoA) methods. The networks are: f_1 = Ell 50, f_2 = DAGAN, f_3 = MRI-VN, f_4 = Deep MRI. The sampling modalities A_j are: A_1 is a subsampled discrete Radon transform, A_2 is a subsampled discrete Fourier transform (single coil simulation), A_3 is a superposition of subsampled discrete Fourier transforms (parallel MRI simulation with 15 coils elements), A_4 is a subsampled discrete Fourier transform (single coil). Note that Deep MRI has not been trained with images containing any of the letters or symbols used in the perturbation, yet it is completely stable with respect to the structural changes. The same is true for the AUTOMAP network (see first column of Figure I.3). Last row: The figures show PSNR as a function of the subsampling rate for different networks. The red line indicates the subsampling ratio that the networks were trained for.

that is completely invisible to the human eye, yet it results in a reconstruction that is hard to dismiss as non-physical, and provides an incorrect representation of the actual image. Such examples could potentially lead to incorrect medical diagnosis. Note that state-of-the-art methods are not affected by the perturbation as demonstrated in the rightmost column of Figure I.1. However, although this network is highly unstable with respect to certain tiny perturbations, it is highly stable with respect to small structured changes, see the 4th row of Figure I.5. Note also that the instabilities are actually stable. In particular, in Figure I.2 we demonstrate how a random Gaussian perturbation added to the perturbation r_1 still yields a substantial artefact (see also §I.A.1.8 in SI).

AUTOMAP: This experiment is similar to the one above, however, in this case we add $\tilde{r}_1, \dots, \tilde{r}_4$ to the measurements $y = Ax$, where $|\tilde{r}_1| < |\tilde{r}_2| < |\tilde{r}_3| < |\tilde{r}_4|$ and A is a subsampled discrete Fourier transform. In order to illustrate how small the perturbations are we have visualised $|x + r_j|$ in the first row of Figure I.3, where $y + \tilde{r}_j = A(x + r_j)$. To emphasise how the network reconstruction completely deforms the image we have, inspired by the second test on structural changes, added a small structural change in form of a heart that gradually disappears completely in the network reconstruction. This is demonstrated in the second row of Figure I.3, and the third row of Figure I.3 contains the reconstruction done by a state-of-the-art method. Note that the worst case perturbations are completely different to the ones failing the Deep MRI network. Hence, the artefacts are also completely different. These perturbations are white-noise like and the reconstructions from the network provide a similar impression. As this is a standard artefact in MRI, it is, however, not clear how to protect against the potential bad tiny noise. Indeed, a detail may be washed out, as shown in the experiment (note the heart inserted with slightly different intensities in the brain image), but the similarity between the standard artefact may make it difficult to judge that this is an untrustworthy image.

MRI-VN: In this case we add one perturbation r_1 to the image, where r_1 is produced by letting the algorithm searching for the worst perturbation run until it has converged. The results are shown in the first two columns of Figure I.4, and the conclusion is the same for the MRI-VN net as for Deep MRI and AUTOPMAP; perturbations barely visible to the human eye, even when zooming in, yield substantial misleading artefacts. Note also that the perturbation has no effect on the state-of-the-art-method.

Med-50: Here we add a perturbation r_2 that is also produced by running the algorithm until it has converged, and the results are visualised in the last two columns of Figure I.4. The Med-50 network is moderately unstable with respect to tiny perturbations compared to Deep MRI, AUTOMAP and MRI-VN, however, severe artefacts are clearly seen. It is worth noting that this network is used for the Radon transform, which is, from a stability point of view, a more unstable operator than the Fourier transform when considering its inverse.

I.4 Stability with respect to small structural changes

Instabilities with respect to small structural changes are documented below.

Ell-50: This network provides a stark example of instability with respect to structural perturbation. Indeed, none of the details are visible in the reconstruction as documented in the first row of Figure I.5. This may not be entirely surprising, given that the network is trained on ellipses.

DAGAN: This network is not as unstable as the Ell-50 network with respect to structural changes. However, as seen in the second row of Figure I.5 the blurring of the structural details are substantial, and the instability is still critical.

MRI-VN: This is an example of a moderately unstable network when considering structural changes. Note, however, how the instability coincides with the lack of ability to reconstruct details in general. This is documented in the third row of Figure I.5.

Deep MRI: To demonstrate how the stability with respect to small structured changes coincides with the ability to reconstruct details, we show how stable the Deep MRI network is. Observe also how well the details in the image are preserved in the fourth row of Figure I.5. Here we have lowered the subsampling ration to 25% even when the network is trained on 33% subsampling ratio. We want to point out that none of the symbols, nor any text, has been used in the training set.

I.5 Stability with respect to more samples

Certain convolutional neural networks will allow for the flexibility of changing the amount of sampling. In our test cases all of the networks except AUTOMAP have this feature, and we report on the stability with respect to changes in the amount of samples below and in the last row of Figure I.5:

Ell 50/Med 50: Ell 50 has the strongest and most fascinating decay in performance as a function of an increasing subsampling ratio. Med 50 is similar, however, with a less steep decline in reconstruction quality.

DAGAN: The reconstruction quality deteriorates with more samples similar to the Ell 50/Med 50 networks.

VN-MRI: This network provides reconstructions where the quality stagnates with more samples as opposed to the decay in performance witnessed in the other cases.

Deep MRI: This network is the only one that behaves aligned with standard state-of-the-art methods and provides better reconstructions when more samples are added.

I.6 Conclusion

The new paradigm of learning the reconstruction algorithm for image reconstruction in inverse problem, through deep learning, typically yields unstable methods. Moreover, our test reveals numerous instability phenomena, challenges and new research directions. In particular:

(1) Certain tiny perturbations lead to a myriad of different artefacts. Different networks yield different artefacts and instabilities, and as Figures I.1, I.3, I.4 reveal there is no common denominator. Moreover, the artefacts may be difficult to detect as non-physical. Thus, several key questions emerge: given a trained neural network, which types of artefacts may the network produce? How is the instability related to the network architecture, training set and also subsampling patterns?

(2) Variety in failure of recovering structural changes. There is a great variety in the instabilities with respect to structural changes as demonstrated in Figure I.4, ranging from complete removal of details to more subtle distortions and blurring of the features. How is this related to the network architecture and training set? Moreover, does the subsampling pattern play a role? It is important, however, to observe (as in the 4th row of Figure I.5 and 1st column of Figure I.3) that there are perfectly stable networks with respect to structural changes, even when the training set does not contain any images with such details.

(3) Networks must be retrained on any subsampling pattern. The fact that more samples may cause the quality of reconstruction to either deteriorate or stagnate means that each network has to be retrained on every specific subsampling pattern, subsampling ratio and dimensions used. Hence, one may in practice need hundreds of different network to facilitate the many different combinations of dimensions, subsampling ratios and sampling patterns.

(4) Instabilities are not necessarily rare events. A key question regarding instabilities with respect to tiny perturbations is whether they may occur in practice. The example in Figure I.2 suggests that there is a ball around a worst-case perturbation in which the severe artefacts are always witnessed. This suggests that the set of 'bad' perturbations have Lebesgue measure greater than zero, and thus, there will typically be a non-zero probability of a 'bad' perturbation. Estimating this probability may be highly non-trivial as the perturbation will typically be the sum of two random variables, where one variable comes from generic noise and one highly non-generic variable is due to patient movements, anatomic differences, apparatus malfunctions, etc. These predictions can also be theoretically verified, as discussed in §I.A.1.8 in SI.

(5) The instability phenomenon is not easy to remedy. We deliberately choose quite different networks in this paper to highlight the seeming ubiquity of the instability phenomenon. Theoretical insights (see §I.A.1.8 in SI on the next generation of methods [3, 4, 17, 41, 50]) also support the conclusion that this phenomenon is nontrivial to overcome. Finding effective remedies is an extremely important future challenge.

I.7 Acknowledgements

The authors would like to thank Kristian Monsen Haug for help with Figure S3 in SI. The authors would like to thank Dr. Cynthia McCollough, the Mayo Clinic, the American Association of Physicists in Medicine, and the National Institute of Biomedical Imaging and Bioengineering for allowing the use of their data in the experiments. F. Renna acknowledges support from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 655282 and the FCT grant SFRH/BPD/118714/2016. B. Adcock acknowledges support from the NSERC grant 611675. A. C. Hansen thanks NVIDIA for a GPU grant in form of a Titan X Pascal and acknowledges support from a Royal Society University Research Fellowship, the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/L003457/1 and a Leverhulme Prize 2017.

Code and data

All code available from https://github.com/vegarant/Invfool.

I.A Supplementary Information

I.A.1 Methods

The specific setups for deep learning and neural networks in inverse problems are typically rather specialised for each type of network. However, the main idea can be presented in a rather general way. Given an under-sampled inverse problem

$$Ax = y, \qquad A \in \mathbb{C}^{m \times N}, \qquad m < N \tag{I.1}$$

there is typically an easy linear way of approximating x from the measurement vector y. We will denote this linear operator by $H \in \mathbb{C}^{N \times m}$. In the MRI case, when A is a subsampled discrete Fourier transform, often $H = A^*$. Note that in the MRI case x is complex valued and we actually display the magnitude image $|x|$. An example is illustrated in Figure I.6. In the CT case H could be A^*, however, this gives very poor results (as demonstrated in Figure I.6), and thus H is usually a discretisation of the filtered back projection (FBP). The problem is, as displayed in Figure I.6, that the reconstruction $\tilde{x} = Hy$ may still be rather poor. The philosophy of deep learning is quite simple; improve this reconstruction by using learning. In particular, inspired by deep learning in image denoising [5], given training images $\{x_1, \ldots, x_n\}$ and poor reconstructions $\{HAx_1, \ldots, HAx_n\}$, train a neural network $f : \mathbb{C}^N \to \mathbb{C}^N$ such that

$$\|f(HAx_j) - x_j\| \ll \|HAx_j - x_j\|, \quad j = 1, \ldots, n. \tag{I.2}$$

The hope is that (I.2) will hold for other images as well, not just the training examples $\{x_1, \ldots, x_n\}$.

The construction process of the neural network f is typically done as follows. First one decides on a particular class (architecture) of neural networks $\mathcal{NN}$.

Then one decides on a cost function $\mathrm{Cost} : \mathcal{NN} \times \mathbb{C}^N \times \mathbb{C}^m \times \mathbb{C}^N \to \mathbb{R}$ and tries to solve the optimisation problem of finding

$$f \in \operatorname*{argmin}_{h \in \mathcal{NN}} \sum_{j=1}^{n} \mathrm{Cost}(h, HAx_j, Ax_j, x_j). \tag{I.3}$$

The task of finding a good class $\mathcal{NN}$ and a good cost function Cost is an engineering art on its own. All the networks we test, except for AUTOMAP, are trained with some form of a 'warm start' in form of a linear operator H, however, AUTOMAP is based on directly solving the problem

$$f \in \operatorname*{argmin}_{h \in \mathcal{NN}} \sum_{j=1}^{n} \mathrm{Cost}(h, Ax_j, x_j), \tag{I.4}$$

without any reference to the reconstructions HAx_j. We refer to Section I.A.2 for details regarding the training of the networks. Note that the instability phenomenon is independent of the choice of (I.3) or (I.4), and the operator H may be viewed as just adding a layer to the network. Thus, we will in general talk about a network f that takes the measurements $y = Ax$ as input.

I.A.1.1 Describing the test

Before describing the algorithm for creating the unstable perturbations, it is convenient to have a short review of the framework for establishing instabilities for neural networks for image classification. For a detailed review of such methods, see [13] and the references therein. The basic idea is as follows. Let $g \colon \mathbb{R}^d \to [0,1]^C$ be an image classification network with C different classes, so that $g(x)$ is a C-dimensional vector containing the probabilities associated to the different classes for a given input image x. Let $\hat{k}_g \colon \mathbb{R}^d \to \{1, \dots, C\}$ where $\hat{k}_g(x) = \mathrm{argmax}_i(g(x)_i)$ is the image classifier. For a given norm $\|\cdot\|$ on $\mathbb{R}^d$, we can then define the optimal, meaning smallest, unstable perturbation $r^* \in \mathbb{R}^d$, for an image $x \in \mathbb{R}^d$ as

$$r^*(x) \in \operatorname*{argmin}_{r} \|r\| \;\text{ subject to }\; \hat{k}_g(x+r) \neq \hat{k}_g(x), \tag{I.5}$$

where we write $r^*(x)$ to indicate that this is a perturbation for the image x.

It is clear that one cannot apply the approach in (I.5) to the problem of finding instabilities of neural networks for the inverse problem. Indeed, (I.5) is designed for a decision problem a la 'is there a cat in the image?'. In inverse problems there is no decision problem but rather the following, slightly simplified issue:

$$\text{Reconstruct } x \text{ from } y = Ax, \qquad A \in \mathbb{C}^{m \times N}. \tag{I.6}$$

Thus, if we are given a neural network $f \colon \mathbb{C}^m \to \mathbb{C}^N$ designed to solve (I.6), and we want to search for instabilities imitating (I.5), we would end up with the problem of finding

$$\hat{r}(x) \in \operatorname*{argmin}_{r} \|r\| \text{ subject to } \|f(y + Ar) - f(y)\| \geq \delta, \tag{I.7}$$

for some $\delta > 0$, where $y = Ax$ for some x. Note that (I.7) has a clear disadvantage in that it may be infeasible for different values of δ. Hence, a slightly different, constrained Lasso inspired variant, may be worth considering;

$$\tilde{r}(x) \in \underset{r}{\operatorname{argmax}} \|f(y + Ar) - f(y)\| \quad \text{subject to} \quad \|r\| \leq \theta, \tag{I.8}$$

for some $\tau > 0$. In the case of (I.8) we do not have any issues regarding infeasibility. However, a third option could be an unconstrained Lasso inspired version of (I.8) given by

$$r^*(y) \in \underset{r}{\operatorname{argmax}} \frac{1}{2}\|f(y + Ar) - f(y)\|_2^2 - \frac{\lambda}{2}\|r\|_2^2 \tag{I.9}$$

(here we have specified the norm), where $\lambda > 0$. Note that (I.9) is not the only possibility. In particular, one could consider the more general setting

$$r^*(y) \in \underset{r}{\operatorname{argmax}} \frac{1}{2}\|f(y + Ar) - p(x)\|_2^2 - \frac{\lambda}{2}\|r\|_2^2, \tag{I.10}$$

where $p : \mathbb{C}^N \to \mathbb{C}^N$. In this case r^* will obviously depend on p, and the quality of the artefacts produced by r^* may differ greatly as p changes. Indeed, this is the motivation for allowing this extra variable. In this paper we focus on (I.10) and consider $p(x) = f(Ax)$ (as in (I.9)) and $p(x) = x$.

However, the first part of our test could indeed be carried out by a different optimisation problem. Moreover, we anticipate that there will be other methods for creating instabilities for neural networks for inverse problems that will be as reliable and diverse as what we present here. Note that (I.10) is set up to find perturbations in the image domain. We do this deliberately as this provides an easy way to compare the original image with a perturbed image and deduce whether the reconstruction of the perturbed image is acceptable/unacceptable. However, one could set up (I.10) so that the perturbation is in the sampling domain as well. In the following we describe the test in detail and the methodology.

I.A.1.2 Stability with respect to tiny perturbations

The neural network $f \colon \mathbb{C}^m \to \mathbb{C}^N$ is a non-linear function. In practice this makes the problem of finding a global maximum of the optimization problem in (I.10) impossible, even for small values of m and N. In the following we will provide a method that aims at locating local maxima of (I.10) by using a gradient search method. In particular, given an image $x \in \mathbb{R}^N$, $A \in \mathbb{C}^{m\times N}$ and $y = Ax$ as in (I.1) let

$$Q_y^p(r) = \frac{1}{2}\|f(y + Ar) - p(x)\|_2^2 - \frac{\lambda}{2}\|r\|_2^2 \tag{I.11}$$

be the objective function. A most natural method to solve (I.10) is *gradient ascent with momentum*. Thus, the method uses the gradient of Q_y^p in conjunction with two parameters $\gamma > 0$ (the momentum) and $\eta > 0$ (the learning rate) in each step towards a local maximum.

Algorithm 1 Finding unstable perturbation for inverse problems

```
Input: Image: $x$, neural network: $f$, sampling matrix: $A$, maximum number of iterations: $M$.
Output: Perturbation $r_M$
Initialize: $y \leftarrow Ax$, $v \leftarrow 0$, $i \leftarrow 1$, $r_0 \sim \text{Unif}([0,1]^N)$, $0 < \lambda, \gamma, \eta, \tau$. Set $Q_y^p(r)$ as in Equation (I.11).
$r_0 \leftarrow \tau r_0$
while $i \leq M$ do
    $v_{i+1} \leftarrow \gamma v_i + \eta \nabla_r Q_y(r_i)$
    $r_{i+1} \leftarrow r_i + v_{i+1}$
    $i \leftarrow i + 1$
return $r_M$
```

This means that there are three parameters $\lambda > 0$, $\gamma > 0$ and $\eta > 0$ to be set, and hence the perturbation r found by the algorithm will depend on these. The complete algorithm is presented in Algorithm 1, where r_0 is initialised randomly. Note that the parameter τ used in Algorithm 1 is simply a scaling factor needed as the input images may have values in different ranges.

Note that for $u = y + Ar$, the gradient of Q_y^p is given by

$$\nabla_r Q_y^p = A^* \nabla_u g(u) - \lambda r, \quad g(u) := \|f(u) - p(x)\|_2^2 \tag{I.12}$$

where $\nabla_u g(u)$ can be computed efficiently using back propagation. Note also that at each iteration this gradient is left multiplied by the adjoint A^*.

Algorithm 2 Finding unstable perturbation for Radon problems

```
Input: Image: $x$, neural network: $f$, sampling matrix: $A$, FBP operator: $B$, maximum number of iterations: $M$.
Output: Perturbation $r_M$
Initialize: $y \leftarrow Ax$, $v \leftarrow 0$, $i \leftarrow 1$, $r_0 \sim \text{Unif}([0,1]^N)$, $0 < \lambda, \gamma, \eta, \tau$. Set $g(u)$ as in Equation (I.12).
$r_0 \leftarrow \tau r_0$
while $i \leq M$ do
    $v_{i+1} \leftarrow \gamma v_i + \eta B \nabla_u g(y + Ar_i) - \lambda r_i$
    $r_{i+1} \leftarrow r_i + v_{i+1}$
    $i \leftarrow i + 1$
return $r_M$
```

Just as in the case when training neural networks using stochastic gradient descent with momentum, choosing the parameters γ and η is an art of engineering. We are in a similar situation with our algorithm, and the optimal choices of

γ, η are based on empirical testing. Such experimenting with parameters also motivates experimenting with other parts of the algorithm. For example, when considering Radon measurements, we found that setting the optimal values of γ and η could be rather difficult. However, by replacing A^* in (I.12) by $B \in \mathbb{R}^{N \times m}$ being a discretisation of a filtered back projection (FBP), this problem could be overcome and we therefore use Algorithm 2 in the case of Radon samples.

It should be mentioned that there are different variations of discretisations of the filtered backprojection for Radon problems. The discretisation $B \in \mathbb{R}^{N \times m}$ used in our experiment is the one provided by MatLab R2018b.

I.A.1.3 State-of-the-art comparison method

All of our tests are done against state-of-the-art benchmark methods using established techniques based on sparse regularisation and compressed sensing [2, 7, 9, 43].

There are many variations in the literature using X-lets and Total Variation (TV) techniques separately or in combination. Our main algorithm is based on the re-weighting technique suggested in [8]. This idea was refined in [35] and [34], by combining both X-lets (shearlets in this case) and TV. This is our main algorithm of choice used in this paper. We refer the reader to [34] for details, however, a short summary can be described as follows. The algorithm allows for both Fourier and Radon sampling, however, the current implementation only allows for single coil MRI in the Fourier case. The idea is to solve iteratively the problem

$$\underset{z}{\text{minimise}} \sum_{j=1}^{J} \lambda_j \|W_j \Psi_j z\|_1 + \mathrm{TGV}_{\alpha}^2(z) \text{ subject to } Az = y,$$

where the λ_js are weights, W_j is a diagonal weighting matrix, Ψ_j is the j'th subband in a shearlet transform [34], and TGV_{α}^2, $\alpha = (\alpha_1, \alpha_2)$ is the second order Total Generalised Variation operator. The TGV_{α}^2 operator consist of a first order term (TV) weighted by α_1 and a second order (generalised) term weighted by α_2. In each iteration step the weights λ_j and weighting matrices W_j are updated.

In particular, the minimisation problem is casted into an unconstrained formulation

$$\underset{z}{\text{minimise}} \sum_{j=1}^{J} \lambda_j \|W_j \Psi_j z\|_1 + \mathrm{TGV}_{\alpha}^2(z) + \frac{\beta}{2} \|Az - y\|_2^2,$$

and solved via split Bregman iterations. This means that the problem is decoupled into two portions, one accounting for the ℓ_1-norm term and one for the ℓ_2-norm term.

In particular, on denoting by Ψ' a composite operator including (1) the effect of multi-scale X-lets transform in different levels including the weights λ_j, (2) the first order (TV) term of TGV_{α}^2 and (3) the second order term of the same operator, and by adding a further splitting variable $d = \Psi' z$, it is possible to write the k-th split Bregman iteration as

$$\left\{\begin{array}{rcl} (z_{k+1}, d_{k+1}) & = & \arg\min_{z,d} \|Wd\|_1 + \frac{\beta}{2}\|Az - y_k\|_2^2 \\ & & + \frac{\mu}{2}\|d - \Psi' z - b_k\|_2^2, \\ b_{k+1} & = & b_k + \Psi' z_{k+1} - d_{k+1}, \\ y_{k+1} & = & y_k + y - Az_{k+1}. \end{array}\right. \tag{I.13}$$

During each iteration, the (z, d)-minimisation problem is solved using one or multiple non-linear block Gauss-Seidel iterations, which alternate between minimising z and d. Also, in contrast with the re-weighting strategy originally presented in [8], the weights in W are updated not only after convergence to the solution of the ℓ_1 minimisation problem, but weight updates are incorporated in the split Bregman iterations.

In the above iterations we have allowed for a slight abuse of notation. We are using $\mu = (\mu_1, \mu_2, \mu_3)$ and split the sum $\|d - \Psi' z - b_k\|_2^2$, into three separate parts, depending on which of the terms of Ψ' they come from, and weight each partial sum separately with $\frac{\mu_1}{2}$, $\frac{\mu_2}{2}$ and $\frac{\mu_3}{2}$, respectively (see equation (15) in [34] for details).

This method has been used for reconstruction from Fourier and Radon measurements, using two different setups. For the case of Fourier measurements, discrete shearlets are generated with 3 scales and with directional parameters [1 2 2] (see [35] and [34] for details). The optimisation parameters are set as follows:

- $(\mu_1, \mu_2, \mu_3) : (5000, 10, 20)$,
- (α_1, α_2): $(1, 1)$,
- β: 10^5,
- ϵ: 10^{-5},

Where ϵ is a parameter which is added in the denominator, of the updating rule, for the weights W, to avoid division by zero. Similarly, image reconstruction from Radon measurements are obtained by using shearlets with 4 scales and directional parameters [0 0 1 1] and with the following parameter setup:

- $(\mu_1, \mu_2, \mu_3) : (500, 0, 0)$,
- (α_1, α_2): $(1, 0)$,
- β: 50,
- ϵ: 10^{-8},

Notice in particular that $\mu_3 = \alpha_2 = 0$, hence we are only using shearlets and a TV term as our regularizes. In all setups, we run the algorithm until convergence, i.e. between 50 and 500 iterations.

The above approach is used in all examples except for the tests using the MRI-VN network which is designed for parallel MRI. For this imaging modality we have the following reconstruction problem. Let $\Omega \subseteq \{1, \dots, N\}$, $|\Omega| = m$ and $P_\Omega \in \mathbb{R}^{m \times N}$ be the projection operator onto the canonical basis i.e. $P_\Omega x = (x_i)_{i \in \Omega}$. Let $F \in \mathbb{C}^{N \times N}$ be the discrete Fourier transform (DFT) matrix. The Fourier sampling matrix can then be written as $A_f = P_\Omega F$, for a given sampling pattern Ω. In parallel Fourier imaging we receive information from multiple coils elements at the same time. This is modeled by introducing diagonal sensitivity matrices $S_1, \dots S_c \in \mathbb{C}^{N \times N}$ which weight the measurements, based on the environmental conditions of the sensing problem. The corresponding measurement matrix is then written as

$$A_{pf} = \begin{bmatrix} P_\Omega F & & \\ & \ddots & \\ & & P_\Omega F \end{bmatrix} \begin{bmatrix} S_1 \\ \vdots \\ S_c \end{bmatrix} \in \mathbb{C}^{m' \times N}.$$

where $m' = cm$. Note that for this sampling operator we might have $m' > N$, which means that the corresponding linear system may be overdetermined. Given an image $x \in \mathbb{R}^N$ we let $y = A_{pf} x$ and use the SPGL1 algorithm [53] for solving

$$\underset{z}{\text{minimise}} \, \|z\|_1 \quad \text{subject to} \quad \|A_{pf} \Psi^{-1} z - y\|_2 \leq \delta,$$

where $\Psi \in \mathbb{R}^{N \times N}$ is the wavelet transform corresponding to the periodised Daubechies 2 wavelet with 3 levels. In all the experiments we set $\delta = 0.01$.

I.A.1.4 Non-uniqueness of the test – parameter dependency

Note that the test we provide can never become unique. Indeed, we choose to solve (I.10) with different choices of p (see Figure I.7), however, (I.7) and (I.8) could also be viable alternatives. Moreover, all of these approaches depend on parameters δ, θ and λ that have to be specified, and different values give different worst-case perturbations. In addition, Algorithm 1 and Algorithm 2 designed to solve (I.10) depend on the parameters γ, η and τ. Moreover, note also that there is no built-in halting criteria in Algorithm 1 and Algorithm 2, but rather the parameter M controlling the number of iterations. Thus, the stability test can never become a unique test, but instead a collection of algorithms depending on different parameters. Hence, an appropriate use of the test means running Algorithm 1 and Algorithm 2 varying the parameters. This is also what is done in this paper, however, only the results based on the final parameters are displayed in the figures. The final parameters chosen are listed in Table I.1.

In Figure I.9, we display different perturbations r_j produced by Algorithm 1 with different values of M corresponding to the experiments shown in Figures I.1, I.3 and I.7. The values of λ, γ, η and τ are as in Table I.1. Note the difference between the perturbations depending on the network. As the perturbations are tiny, they have been enlarged in order to get a visual impression.

I.A.1.5 Stability of state-of-the-art methods

The perturbation r computed by the algorithms from Section I.A.1.2 are constructed specifically for an image x, sampling operator A and neural network f. Hence it might not be too surprising that the state-of-the-art methods are unaffected by this perturbation. Though our use of state-of-the-art methods in this work have mainly been to ensure that the inverse problem in itself is not ill posed, it is tempting to see if similar instabilities can occur for sparse regularization algorithms as well. Indeed, it could be that sparse regularization techniques are equally unstable to a worst case perturbation r, but that such a perturbation is rare enough so as to not have yet arisen in practice. Compressive sensing techniques have after all, only been tested by the scientific community for a little more than a decade, and by clinicians for the last few years.

Extending the algorithms from Section I.A.1.2 so that they can be applied to general sparse regularization algorithms is a challenging task, as both algorithms need to compute the gradient of $g(u) = \|f(u) - p(x)\|_2^2$. Thus if f is not a neural network, we can no longer use the back propagation algorithm to easily compute this quantity. While some of the simplest sparse regularization algorithms can be written as neural networks, more sophisticated algorithms often contain internal *if-else* or *while* statements, making it hard to write them as neural networks.

Writing the above sparse regularization algorithms as neural networks are beyond the scope of this paper. Yet, we want to illustrate that sparse regularization algorithms can be tested if they are written as neural networks. To this end, we include an experiment from [1, 20], where an iterative algorithm for solving the following optimization problem

$$\underset{z\in\mathbb{C}^N}{\text{minimise}}\, \mu\|Wz\|_1 + \|A\Psi^{-1}z - y\|_2, \qquad \mu > 0, \tag{I.14}$$

have been unrolled, using 1000 iterations, as a neural network. Here $W \in \mathbb{R}^{N\times N}$ is a diagonal weighting matrix, $A = P_\Omega F \in \mathbb{C}^{m\times N}$ is a subsampled Fourier transform and $\Psi \in \mathbb{R}^{N\times N}$ is the discrete Haar wavelet transform. To test the stability of this network, we have used the same data x and sampling pattern Ω as for the AUTOMAP network. Using Algorithm 1, we then computed perturbations v_j, $j = 1, 2, 3, 4$ all with the same ℓ_2-norm as the r_j's computed for AUTOMAP in Figure I.3 in the main paper. As can be seen in Figure I.8, the network seems stable with respect to tiny perturbations.

I.A.1.6 Stability with respect to small structural changes

This part is fully explained in the main manuscript. However, we want to emphasise that the symbols used in the experiment are chosen in order to assure that networks can recover important details. These can of course be replaced by other symbols as long as the ability of an algorithm to reconstruct these symbols correlate with the ability to recover other small important details.

Important note: In our examples, it is irrelevant whether the symbols have been included in the training set or not. In fact, both the AUTOMAP and the Deep MRI networks have no problem recovering the symbols, see the first column of Figure I.3 and Figure I.7, where a heart is artificially added, and the fourth row of Figure I.5. Indeed, none of these networks have been trained on images containing the symbols, yet they can perfectly well recover them.

I.A.1.7 Stability with respect to more samples

All of the networks we have tested, except AUTOMAP, are convolutional neural networks (CNNs), which means that the trained weights come from convolutional layers. This has the advantage of reducing the number of parameters we need to learn, compared to fully connected layers (dense matrices), and may for large image sizes be the only alternative. Moreover, these CNNs can easily be adapted to other subsampling patterns as explained below. Thus, one can easily apply a network that is trained on 25% subsampling, say, to input that uses, for example, 35% subsampling. The question is whether the quality of the reconstruction is kept when increasing the subsampling ratio. The reason for the flexibility of the CNNs in our test is that they all depend on the operator $H \in \mathbb{C}^{N \times m}$, as described in (I.2), by considering it as a non-learnable first layer. As the H is non-learnable, this allows for flexibility in our choice of m, since we know how to construct H for various values of m.

In the last row of Figure 4 in the main manuscript we have varied the number of samples m and measured the image quality of the networks reconstruction using the peak signal-noise-ratio (PSNR) between the magnitude images of the original and the reconstructed image. Figure 4 shows all of the networks, except the AUTOMAP network, which learns a mapping directly from the measurement domain without using a non-learned layer H. Below follows the description of each of the experiments visualised in the last row of Figure 4.

Ell 50: We created 25 sinograms of images containing ellipses similar to the data in the network's training and test set. The sinograms were created with 1000 uniformly spaced angles (views) using the formula for the Radon transform of an ellipse. We then considered an acceleration factor $k \in \{2, \ldots, 30\}$, by subsampling every k-th line among the 1000 views. The FBP of the subsampled sinogram was given to the network and the PSNR of the reconstruction was computed against the FBP of all 1000 views.

Med 50: We used a test set, provided by the authors of [24], consisting of 25 CT images from the Mayo Clinic. These images were synthetically sampled, using a the discrete Radon transform from MatLab, at the same angles as the Ell 50 network. The subsampled sinograms were mapped back into the image domain using a FBP and reconstructed using the network. The PSNR values were computed with the original image as ground truth.

DAGAN: We used 20 MR images of brain tissue from the test set, and subsampled these images using the 1D Gaussian sampling patterns provided by the authors of [54]. These patterns have been generated for subsampling rates 1%, 5%, 10%, 20%, 30%, 40% and 50%.

MRI-VN: We used image data from the networks test set, and picked one image slice from 10 different patients. The image data was subsampled with uniformly spaced lines (center lines was always included), at subsampling rates

5%, 10%, 15%, 20%, 25%, 30%, 35%, and 50%. The PSNR was computed with the magnitude image of the fully sampled images as reference.

Deep-MRI: We used 30 image slices from one MRI scan, and subsampled each slice using lines sampled according to a Gaussian distribution. Extra caution was taken, so that all lines sampled at a low sampling rate, was included at higher sampling rates. We sampled with an acceleration rate $2, \ldots, 14$.

It should be noted that measuring image quality is a delicate issue. We point out that no comparison based on the last row of Figure I.5 on the reconstruction quality should be made between the networks, as the PSNR is unfit to measure image quality between different types of images [22]. However, we are only interested in the change in PSNR, as a function of subsampling percentage, for each specific network.

I.A.1.8 Theoretical aspects

Our contribution documents the instability phenomenon in deep learning methods for inverse problems. However, the instability phenomenon can be explained theoretically as well. Indeed, the recent paper [15] provides theoretical foundations that explain the reasons for the instabilities. Moreover, the theoretical results confirm that the instabilities are stable in the sense that there will always be balls around the 'bad' perturbations such that adding elements in the ball to the 'bad' perturbation will yield another 'bad' perturbation. If the perturbation

$$r_{\mathrm{pert}} = r^1_{\mathrm{pert}} + r^2_{\mathrm{pert}},$$

where r^1_{pert} and r^2_{pert} are random variables there will typically be a non-zero probability that r_{pert} is a 'bad' perturbation depending on the probability distribution of r^1_{pert} and r^2_{pert}. We may have that r^1_{pert} is a Gaussian vector but r^2_{pert} could have a probability distribution (coming from patient movement, anatomic differences, etc) that is incredibly difficult to estimate. Thus, deeming unstable neural networks safe for use in medical imaging based on a probabilistic estimate is far from trivial, if not impossible.

The theoretical developments also demonstrate how difficult it is to cure the instability phenomenon. As the theoretical results in [15] confirm, one can predict which attempts on remedies that will have limited effect. This includes adversarial training (a technique that has been thoroughly investigated as a remedy for instabilities occurring in image classification), training with multiple random sampling patterns, enforcing consistency, adding small random perturbations to the measurements etc. [3, 4, 17, 41, 50]

I.A.2 Technical details needed to reproduce the results

I.A.2.1 Overview

This section contains all the extra material on neural networks that is useful in order to understand and reproduce all the experiments done in the paper. In particular, the it displays the variety of different architectures and training sets used in the various experiments. The neural networks considered are:

(i) *AUTOMAP* [56]: The AUTOMAP neural network we test is for low resolution single coil MRI with 60% subsampling. In the paper [56] one mentions 40% subsampling, but this apparent discrepancy is simply due to different interpretation of the word subsampling. We use the traditional meaning in sampling theory referring to x% subsampling as describing that the total amount of samples used are x% of full sampling. The network used in our experiment is trained by the authors of [56]. The details of the architecture and training data are summarised in §I.A.2.2.

(ii) *DAGAN* [54]: This network is for medium resolution single coil MRI with 20% subsampling. The network weights are not available online, however, complete instructions on how to reproduce the network used in [54] are accessible. Based on these instruction, we have retrained a network that reproduces the results in [54]. The details of the training data and architecture are summarised in §I.A.2.3.

(iii) *Deep MRI* [44]: This neural network is for medium resolution single coil MRI with 33% subsampling. The network used in our experiments is trained by the authors of [44], can be found online and we summarise the details on training data and architecture in §I.A.2.4.

(iv) *Ell 50* [24]: Ell 50 is a network for CT or any Radon transform based inverse problem. The number 50 refers to the number of lines used in the sampling in the sinogram. The training of the network is done by the authors of [24]. The network can be obtained online, and all the details can be found in §I.A.2.5.

(v) *Med 50* [24]: Med 50 has exactly the same architecture as Ell 50 and is used for CT, however the training is done on a different dataset. The network is trained by the authors of [24]. Details are summarised in §I.A.2.5.

(vi) *MRI-VN* [18]: This network is for medium to high resolution parallel MRI with 15 coil elements and 15% subsampling. In order to show a variety of subsampling ratios we have trained this network on a smaller subsampling percentage than what the authors of [18] originally (25% and 33%) did in their paper. As we already have 33%, and 20%, we want a test on even lower subsampling rates. All the remaining parameters are kept as suggested in the code provided by the authors of [18], except for the subsampling ratios and batch size (due to memory limitations). All the details are documented in §I.A.2.6.

All network weights are available from https://github.com/vegarant/Invfool.

I.A.2.2 AUTOMAP

Network architecture The AUTOMAP network [56] is proposed for image reconstruction from Radon measurements, spatial non-Cartesian Fourier sampling, misaligned Fourier sampling and undersampled Cartesian Fourier samples. In this work we have tested the network trained for image reconstruction from undersampled Cartesian Fourier samples. In contrast with the other networks considered in this work, the AUTOMAP network provides a direct mapping of the Fourier measurements to the image domain without applying the adjoint operator H as a first step.

The authors of [56] have not made their code publicly available, and the weights from their paper [56] had not been stored. However, they kindly agreed to retrain their network for us and save the weights. The network architecture they trained deviates slightly in some of activation functions reported in their paper [56], however, the network was trained on the same data and sampling pattern. Below we describe the network architecture we received. The training parameters and data, are reported as in the paper [56].

The input of the AUTOMAP network, as described in [56] and in Figure I.10 takes a complex $n \times n$ image of measurements as input. In the case of subsampling, one may interpret the $n \times n$ image as being zero padded in the coordinates that are not sampled. In the tests, $n = 128$, and in the actual implementation the input is represented by the complex measurement data $y \in \mathbb{C}^m$ with $m = 9855$ (60% of n^2) in this experiment. Such data is reshaped into a vector of length $2m$ with real entries before being fed into the network. The first two layers of the network a fully connected matrices of size $25000 \times 2m$ and $n^2 \times 25000$. The first fully connected layer is followed by a hyperbolic tangent activation function. The second fully connected layer is followed by a layer which subtracts the mean from the output of the second layer. The output is then reshaped into an $n \times n$ image.

Next follows two convolutional layers with filter size 5×5, 64 feature maps and stride 1×1. The first convolutional layer is followed by a hyperbolic tangent function, while the other is followed by a rectified linear unit (ReLU). Finally, the output layer deconvolves the 64 feature maps provided by the second convolutional layer with 7×7 filters with stride 1×1. The output of the network is an $n \times n$ matrix representing the image magnitudes.

Training parameters The loss function used for training consisted of two terms, $\mathcal{L}_{\mathrm{SE}}$ and $\mathcal{L}_{\mathrm{PEN}}$. Here $\mathcal{L}_{SE}$ is the ℓ_2-norm of the difference between the ground truth magnitude image and the magnitude image predicted by the network. Similarly the $\mathcal{L}_{\mathrm{PEN}}$ is an ℓ_1-penalty on the outputs of the activations following the second convolutional layer ($C2$). The total loss was then computed as

$$\mathcal{L}_{\mathrm{TOTAL}} = \mathcal{L}_{\mathrm{SE}} + \lambda \mathcal{L}_{\mathrm{PEN}}$$

with $\lambda = 0.0001$. The network is trained using the RMSProp algorithm (see for example http://www.cs.toronto.edu/~ tijmen/csc321/slides/lecture_slides_ lec6.pdf as referred to in [56]) with minibatch size 100, learning rate 0.00002, momentum 0, and decay 0.9. The number of training epochs is 100.

Training data The training dataset consists of 50, 000 images taken from 131 different subjects from the MGH-USC HCP public dataset [12][1]. For each image, the central 256 × 256 pixels were cropped and subsampled to a resolution of 128 × 128 pixels. Before training, the images were preprocessed by normalizing the entire dataset to a constant value defined by the maximum intensity of the dataset. Fourier data were obtained by subsampling the Cartesian k-space using a Poisson-disk sampling pattern with 60% undersampling [51].

In order to increase the network robustness against translation, the following data augmentation scheme was applied. New images were created from each image in the training dataset by tiling together four reflections of the original image. Then, the so obtained 256×256 image was cropped to a random 128×128 selection. The routines used to implement the AUTOMAP network were written in TensorFlow[2].

I.A.2.3 DAGAN

Network architecture The DAGAN network was introduced in [54] to recover images from Fourier samples, with particular emphasis on MRI reconstruction applications. The DAGAN network assumes measurements $y = Ax$, where A is a subsampled discrete Fourier transform. The input of the network is represented by the noisy magnitude image $\tilde{x} = |Hy|$, which is obtained by direct inversion of the zero-filled Fourier data, in particular, $H = A^*$.

The recovery algorithm presented in [54] is based on a conditional generative adversarial network (GAN) model, which consists of a generator network, used for the image reconstruction, and a discriminator network, measuring the quality of the reconstructed image. The generator network adopted in [54] has a U-net structure, similar to that used in [24], and its objective is to produce the recovered image. In [54] the authors propose two almost identical architectures, and train them with different loss functions. Below we will describe their ‘refinement’ architecture trained with what is referred to as Pixel-Frequency-Perceptual-GAN-Refinement loss in the paper. The refined version is also our choice, as this architecture and training performed the best in the paper and in our tests as well. The network was not made publicly available, and based on advice from the authors of [54] we trained the network ourselves.

The architecture of the generator network, which is reported in Figure I.11, contains 8 convolutional layers and 8 deconvolutional layers each followed by batch normalization (BN) [23]. The batch normalization layers after the convolutional layers are followed by leaky ReLU (lReLU) activations with slope equal to 0.2 for $x < 0$, while the batch normalization layers after the deconvolutions are followed by a ReLU activation. The generator network also contains skip connections,

[1] https:// db.humanconnectome.org/

[2] https://www.tensorflow.org

i.e., connections that copy the output of a layer directly to the input of a layer further down in the hierarchy. The skip connections are used to concatenate mirrored layers (see Figure I.11). The filter kernels used for the convolutional and deconvolutional layers have size 4×4 with stride 2×2. The number of filters in each convolutional/deconvolutional layer increases/decreases according to Figure I.11.

The last deconvolutional layer is followed by a hyperbolic tangent activation function. A global skip connection, adding the input to the network and the output from the hyperbolic tangent function, is then followed by a ramp function clipping the output values of the image to the range $[-1, 1]$. Adding this last skip connection means that the network is actually approximating the residual error between the network input $\tilde{x} = Hy$ and the image of interest.

The generator network is trained jointly with a discriminator network, which aims to distinguish between the output of the generator network and ground truth images. For further information on this network, we refer to [54].

Training parameters The loss function used to train the DAGAN network consists of four different terms. First, an image domain mean square error (MSE) loss, $\mathcal{L}_{\text{iMSE}}$, which accounts for the ℓ_2 distance between the output of the generator network and the ground truth image. Second, a frequency domain MSE loss, $\mathcal{L}_{\text{fMSE}}$, which enforces consistency between the output of the generative network in the frequency domain and the acquired Fourier measurements. Third, a perceptual loss term, $\mathcal{L}_{\text{VGG}}$, which is computed by using a pretrained VGG-16 described in [46]. In particular, the VGG-16 network was trained over the ImageNet dataset[3] and the output of its conv4 layer was used to compute the loss term by considering the ℓ_2-norm of the difference between the VGG-16 output corresponding to the ground truth image and the generator network output. Finally, the fourth term, $\mathcal{L}_{\text{GEN}}$ is computed using a cross entropy loss on the output of the discriminator network. Adding these four terms together gives us the loss

$$\mathcal{L}_{\text{TOTAL}} = \alpha\mathcal{L}_{\text{iMSE}} + \beta\mathcal{L}_{\text{fMSE}} + \gamma\mathcal{L}_{\text{VGG}} + \tau\mathcal{L}_{\text{GEN}}, \qquad \alpha, \beta, \gamma, \tau > 0.$$

We used the same values for α, β, γ and τ as in [54], in particular, $\alpha = 15$, $\beta = 0.1$, $\gamma = 0.0025$ and $\tau = 1$. The generator and the discriminator network were jointly trained by alternating gradient optimization. In particular, the Adam [27] optimizer was adopted, with initial learning rate 0.0001, momentum 0.5, and minibatch size 25. The learning rate was halved every 5 epochs. We applied the same early stopping rule as given in their implementation[4]. This is based on measuring the $\mathcal{L}_{\text{iMSE}}$ loss between the training set and validation set. We used the early stopping number 10. In total this resulted in 15 epochs of training.

[3] http://www.image-net.org/

[4] https://github.com/nebulaV/DAGAN

Training data The DAGAN network was trained using data from a MICCAI 2013 grand challenge dataset[5]. We removed all images from the dataset where less than 10% of the pixel values were non-black. In total we therefore used 15912 images for training and 4977 images for validation. The dataset consisted of T_1-weighted MR images of different brain tissues.

The following data augmentation techniques were used to increase the amount of training data: image flipping, rotation, shifting, brightness adjustment, zooming, and elastic distortion [45].

The discrete Fourier transform of the training images were subsampled using 1D Gaussian masks, i.e., masks containing vertical lines of data in the k-space randomly located over the image according to a Gaussian distribution. In our tests we trained a network to do recovery from 20% subsampling. The code used to implement DAGAN was written using the TensorLayer[6] wrapper and TensorFlow, and was made publicly available by the authors of [54].

I.A.2.4 DeepMRINet

Network architecture The Deep MRI net is used to recover images from their subsampled Fourier measurements. Its architecture is built up as a cascade of neural networks, whose input is represented by the blurry image obtained by direct inversion of the measurements, i.e., $\tilde{x} = Hy$. The networks in the cascade are convolutional neural networks (CNN) designed as follows

$$CNN_i(\tilde{x}) = \tilde{x} + C_{rec}^{(i)}\rho(C_{rec-1}^{(i)} \cdots \rho(C_1^{(i)}\tilde{x} + b_1^{(i)}) \cdots + b_{rec-1}^{(i)}) + b_{rec}^{(i)},$$

where $\rho(z) = \max\{0, z\}$ is the ReLU activation function, whereas $C_k^{(i)}$ and $b_k^{(i)}$represent trainable convolutional operators and biases, respectively, for the ith network. These networks are then tied together and interleaved with *data consistency layers (DC)*, which have the objective to promote consistency between the reconstructed images and the Fourier measurements. The DC layers are defined as

$$DC_\lambda(\tilde{x}, y, \Omega) = F^{-1}g_\lambda(F\tilde{x}, y, \Omega), \quad \text{where} \quad g_\lambda(z, y, \Omega) = \begin{cases} z_k & k \notin \Omega \\ \frac{z_k + \lambda y_k}{1+\lambda} & k \in \Omega \end{cases}.$$

Here F represents the Fourier operator, and Ω is the set of indices corresponding to the measurements acquired in the k-space. We point out that in the limit $\lambda \to \infty$, the g_λ function simplifies to y_k if $k \in \Omega$ and z_k otherwise.

In practice, a DC layer performs a weighted average of the Fourier coefficients of the image obtained as the output of a CNN in the cascade and the true samples y. The parameter λ can either be trained or kept fixed. In [44], it is not specified whether λ is learned or not, however, from the code[7] it is clear that λ is chosen to be ∞.

[5] http://masiweb.vuse.vanderbilt.edu/workshop2013/index.php/

[6] http://tensorlayer.readthedocs.io

[7] https://github.com/js3611/Deep-MRI-Reconstruction

The complete network can now be written as

$$f(y,\Omega) = DC_\lambda(CNN_n(\cdots DC_\lambda((CNN_1(Hy)), y, \Omega)\cdots), y, \Omega),$$

and its architecture is reported in Figure I.12. In particular, the architecture used to produce the results in [44] and those reported in this paper contains 5 CNNs interleaved with 5 DC layers. Each CNN contains 5 convolutional layers, all with kernel size 3 × 3 and stride 1 × 1. The first 4 layers are using 64 filters and are followed by a ReLU activation function. The fifth convolutional layer in each CNN contains 2 filters, representing the real and imaginary part of the image. This fifth layer is not followed by any activation function, however its output is added to the input to the CNN using a skip connection.

Training parameters In our experiments we used a pre-trained network that was trained (and published online) by the authors of [44] with training parameters documented in the paper [44]. The DeepMRINet was trained using a loss function with two terms, $\mathcal{L}_{\text{MSE}}$ and $\mathcal{L}_{\text{WEIGHTS}}$. The $\mathcal{L}_{\text{MSE}}$ term computed the mean squared error (MSE) between the true (complex valued) image and the predicted (complex valued) image, while the $\mathcal{L}_{\text{WEIGHTS}}$ computed the ℓ_2-norm of the weights. The loss function was then computed as

$$\mathcal{L}_{\text{TOTAL}} = \mathcal{L}_{\text{MSE}} + 10^{-7}\mathcal{L}_{\text{WEIGHTS}}.$$

The network weights were initialized using He initialization [21] and the Adam [27] optimizer was used for training. This optimizer takes as input a learning rate (step size) α, and two exponential decay parameters β_1 and β_2 related to a momentum term. We refer to [27] for further explanations of these parameters. The network was trained with $\alpha = 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and batch size equal 10.

Training data The DeepMRINet was trained using data from five subjects from the MR dataset used in [6], which consists of 10 fully sampled short-axis cardiac cine scans. Each of these scans was then preprocessed, using the SENSE [40] software, into 30 temporal (complex-valued) frames of size 256 × 256. Synthetic MRI measurements were then obtained by sampling retrospectively the reconstructed images in k-space according to a Cartesian undersampling masks. During training, whereas a fixed undersampling rate of 33% was used, different undersampling masks were randomly generated in order to allow the network to recover images from measurements obtained with different undersampling masks. In particular, training images were fully sampled along the frequency-encoding direction but undersampled in the phase-encoding direction, according to the scheme described in [25] (center frequencies were always included in the subsampling patterns).

To prevent overfitting, data augmentation was performed by including rigid transformations of the considered images in the training datasets.

The code used to implement the DeepMRINet was written in Python using the Theano [8] and Lasagne[9] libaries.

I.A.2.5 FBPConvNet – The Ell 50 and Med 50 networks

The Ell 50 and Med 50 networks were proposed in [24] under the name FBPConvNet. The networks are trained to reconstruct images from Radon measurements. The networks have identical architecture and are trained using the same algorithm, with the same set of hyper parameters. The only difference between the training of the two networks, is the dataset they have been trained on. Below, we will describe the architecture and the training procedure of both the networks. We will then describe the datasets for the two networks in separate sections.

Network architecture The Ell 50 and Med 50 networks are trained for reconstructing x from measurements $y = Ax$ where $A \in \mathbb{R}^{m\times N}$ is a Radon[10] sampling operator, sampling 50 uniformly spaced radial lines. Rather than learning a mapping from y to x directly, the networks takes advantage of a discrete filtered back projection[11] $H \in \mathbb{R}^{N\times m}$, as described in the methods section, to obtain a noisy approximation $\tilde{x} = Hy$ to x. The operator H can be seen as a non-learnable first layer in the network.

The network contain several convolutional and deconvolutional layers, all of which (except the last) are followed by a batch normalization (BN) layer and a ReLU activation function. The (de)convolutional layers use filter size 3×3, stride 1×1 and a varying number of filters. We will not describe the full architecture in detail, as it can be seen in Figure I.13, with the relevant number of filters, skip connections, max-poolings (2×2) and concatenations. We do, however, point out that the network applies a final global skip connection, so that rather than learning a mapping from $\tilde{x}$ to x the network is trying to learn the 'noise' $x - \tilde{x}$.

Training parameters The network weights were provided by the authors of [24] and obtained based on the training procedure as described in their paper [24]. The loss function used to train the networks is the ℓ^2 difference between the network output and the ground truth, and the networks are trained using the stochastic gradient descent algorithm with momentum. The learning rate varies from 0.01 to 0.001, whereas the momentum is set to 0.99, and the minibatch size is equal to 1. During training, gradients are clipped to the interval $[-I_{\max}, I_{\max}]$ with $I_{\max} = 10^{-2}$, to prevent the divergence of the cost function. The networks

[8] http://deeplearning.net/software/theano

[9] https://lasagne.readthedocs.io/en/latest

[10] We used MatLabs `randon` command to represent this operator

[11] We used MatLabs `iradon` with linear interpolation and a 'Ram-Lak' filter to represent this operator

are trained for 101 epochs, and the code used to implement the networks is written in MatLab using the library MatConvNet[12].

Ell 50 – Training data The Ell 50 network is trained from the filtered back projection of 475 synthetic sinograms containing the Radon transform of ellipses of random intensity, size, and location. The dynamic range of the back projected images is adjusted so that image values are contained in the interval $[-500, 500]$. The Radon transform of an ellipse has an analytic formula, and hence this formula was used to create sinograms of such images using 1000 uniformly spaced lines (views). Measurement data are obtained by retaining 50 radial lines out of the 1000 views. The ground truth images were obtained by applying filtered back projection to fully sampled sinograms (i.e., 1000 radial lines). This approach is motivated by the fact that in applications, one will never have access to the underlying actual ground truth image. Data augmentation is also applied to the training data, by considering horizontal and vertical mirroring of the original images.

Med 50 – Training data Med 50 is trained on synthetic images obtained from 475 real in-vivo CT images from the Low-dose Grand challenge competition database provided by the Mayo Clinic. The sinograms used for this training were synthetically generated from high quality CT-images using MatLab `radon` command. The same approach as for the Ell 50 network was used, where one sampled 1000 view and used this as ground truth. The network was trained from 50 of these views.

I.A.2.6 MRI Variational Network (MRI-VN)

Network architecture The MRI Variational Network (MRI-VN) presented in [18] is designed to reconstruct images from undersampled MRI data, sampled using 15 coil elements. Thus, we use the sampling operator $A = A_{pf}$ as described in the methods section, with $c = 15$.

The network structure is inspired by the unfolding of a variational minimization problem including a fidelity term and a regularization term defined according the Fields of Experts model [42]. In particular, each iteration of the corresponding Landweber method [30] corresponds to a layer of the resulting neural network. More specifically, the implementation considered in this work consists of $T = 10$ layers/iterations that can be expressed as follows:

$$u^{t+1} = u^t - (K^t)^T \Psi^t (K^t u^t) + \lambda^t A^*(Au^t - y), \quad 0 \leq t < T \tag{I.15}$$

where $u^0 = Hy$ is the complex image obtained by applying $H = A_{pf}^*$. We will describe each of the remaining components of this network separately.

We start by noticing that the images $u^t \in \mathbb{C}^N$ (stacked as a vector in this simplified description) are complex valued, and can therefore described by its

[12] http://www.vlfeat.org/matconvnet

real and imaginary components u^t_{re} and u^t_{im}, respectively. We will alternate between the representations.

The operator $K^t \colon \mathbb{C}^N \to \mathbb{R}^{N \times N_k}$ acts as follows on u^t,

$$K^t u^t = K^t_{\text{re}} u^t_{\text{re}} + K^t_{\text{im}} u^t_{\text{im}},$$

where $K^t_{\text{re}}, K^t_{\text{im}} \colon \mathbb{R}^N \to \mathbb{R}^{N \times N_k}$, are learnable convolutional operators, with N_k filters (channels), filter size 11×11 and stride 1×1. We will comment on the value of N_k later.

The $\Psi^t \colon \mathbb{R}^{N \times N_k} \to \mathbb{R}^{N \times N_k}$ is a non-linear activation function in the network. For each filter/channel, $i = 1, \ldots, N_k$ it applies the non-linear function

$$\phi^t_i(z) = \sum_{j=1}^{N_w} w^t_{ij} \exp\left(-\frac{(z - \mu_j)^2}{2\sigma^2} \right),$$

pointwise to each component z in that channel. Here $\{w^t_{ij}\}_{i=1,j=1}^{N_k,N_w}$, with $N_w = 31$, are weights which are learnt during the training phase. The nodes μ_j are non-learnable, and distributed in an equidistant manner on the interval $[-I_{\max}, I_{\max}]$, for a fixed value $I_{\max}$, commented on below. The σ is also non-learnable and equals $\frac{2I_{\max}}{N_w - 1}$.

The operator $(K^t)^T \colon \mathbb{R}^{N \times N_k} \to \mathbb{C}^N$, maps $z \mapsto (K^t_{\text{re}})^T z + \mathrm{i}(K^t_{\text{im}})^T z$, where i is the imaginary unit, and $(K^t_{\text{re}})^T, (K^t_{\text{im}})^T$ are the transpose of $K^t_{\text{re}}, K^t_{\text{im}}$, respectively. The matrices A, A^* are the matrix A_{pf} and its adjoint, while λ^t is a learnable scalar. The remaining operations should be clear from Equation (I.15).

During training, each of the filters in K^t_{re} and K_{im} were restricted to have zero mean and have unit Euclidean norm. This was done to avoid a scaling problem with the weights w_{ij}.

To reproduce this network, we use the code published by the authors of [18][13]. Parts of this code uses slightly different parameters, than what was used in the original paper. In particular, the value $N_k = 24$ was chosen, rather than $N_k = 48$, as used in the paper. The value of $I_{\max}$, was also changed from 150 in the paper, to 1 in the code. The change of the $I_{\max}$ value is motivated by another change, also made in the published implementation, namely the scaling of the k-space values. In [18] the k-space volumes (with n_{sl} slices) was normalized by the factor $\sqrt{n_{sl}}10000/\|y_{\text{volume}}\|_2$, whereas in the code this have been changed to scaling each k-space slice y with $1/\|Hy\|_2$. This change has been made to make the their implementation more streamlined. Whenever there has been a conflict between the two sources, we have chosen the version found in the code.

Training parameters The MRI-VN network is trained using the ℓ_2-norm as loss function. In particular, since MRI reconstruction are typically assessed through magnitude images, the error is evaluated by comparing smoothed version

[13] https://github.com/VLOGroup/mri-variationalnetwork

of magnitude images

$$|x|_\epsilon = \sqrt{(\mathrm{Re}(x))^2 + (\mathrm{Im}(x))^2 + \epsilon},$$

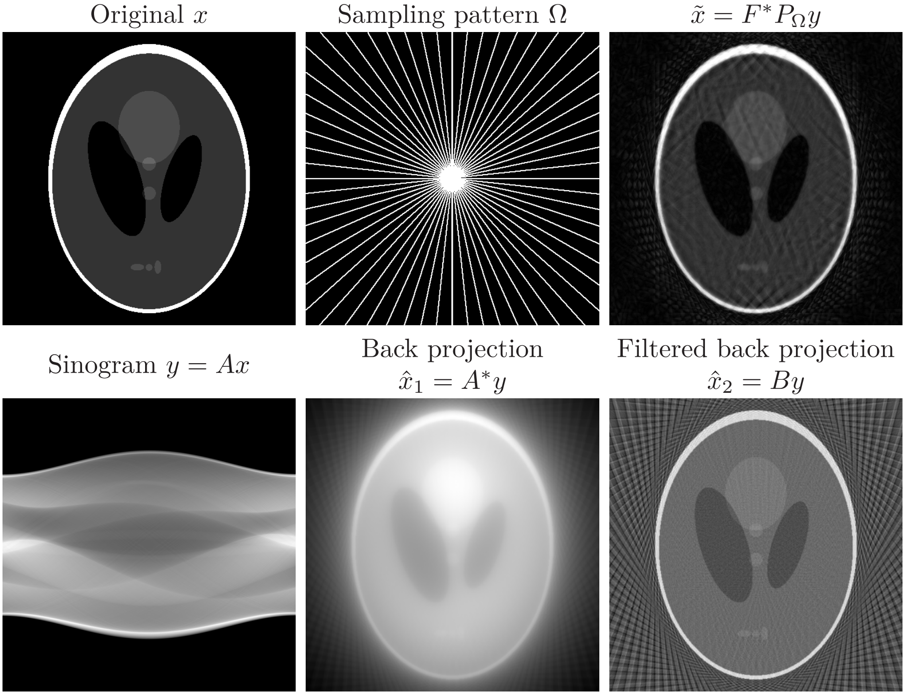
with $\epsilon = 10^{-12}$. The network parameters that minimize the loss function are determined using the inertial incremental proximal gradient (IIPG) optimizer (see [18, 28] for details). Optimization is performed for 1000 epochs, with a step size of 10^{-3}. Training data is arranged into minibatches of size 5. In the original paper, the batch size was set to 10, but due to memory limitations we had to adjust this.

Training data The authors in [18] considered 5 datasets for different types of parallel MR imaging protocols, and trained one VN for each dataset. In this work, we have trained a VN for one of these protocols, namely *Coronal Spin Density weighted with Fat Suppression*. The training data consisted of knee images from 10 patients. From each patient we used 20 slices of the knee images, making up a total of 200 training images. The raw k-space data for each slice consisted of 15, k-space images of size 640 × 368, each with a precomputed sensitivity map. The sensitivity map was computed by the authors of [18], using ESPIRiT [52].

The raw data was obtained using a clinical 3T system (Siemens Magnetom Skyra) using an off-the-shelf 15-element knee coil. The raw data was subsampled retrospectively by zeroing out 85% of the k-space data. In [18] they test both a regular sampling scheme and a variable density pattern as proposed in [33]. In our work, we used a regular sampling scheme, where the 28 first central k-space lines were sampled, and the remaining lines were placed equidistantly in k-space. No data augmentation was used.

The code was implemented in Python with a custom made version of Tensorflow[14], which was partly implemented in C++/CUDA with cuDNN support. All the code and data have been made available online by the authors of [18].

[14] https://github.com/VLOGroup/tensorflow-icg



Original x Sampling pattern Ω $\tilde{x} = F^* P_\Omega y$

Sinogram $y = Ax$ Back projection $\hat{x}_1 = A^* y$ Filtered back projection $\hat{x}_2 = By$

Figure I.6: (Under-sampled MRI and CT problem) We consider sampling $y = Ax$ for two different sampling modalities: the Fourier transform (upper figure simulating MRI) and the Radon transform (lower figure simulating CT) . Upper left: x is the original image. Upper middle: The white dots corresponds to the frequencies we sample i.e. the indices of the sampling pattern Ω. Upper right: The poorly reconstructed image $\tilde{x} = A^* y$, where $A = P_\Omega F$ and F is the 2-dimensional discrete Fourier transform transform and P_Ω is the projection onto the span of $\{e_j\}_{j \in \Omega}$ where the e_js denote the canonical basis. Lower left: Sinogram of Radon measurements using 729 radial lines, $y = Ax$ where A is the discrete Radon sampling matrix. Lower middle: The back projected blurry image $\hat{x} = A^* y$ obtained by using a radon matrix with 50 uniformly spaced lines. Lower right: The slightly sharper image $\hat{x}_2 = By$ using the filtered back projection with 50 uniformly spaced lines.

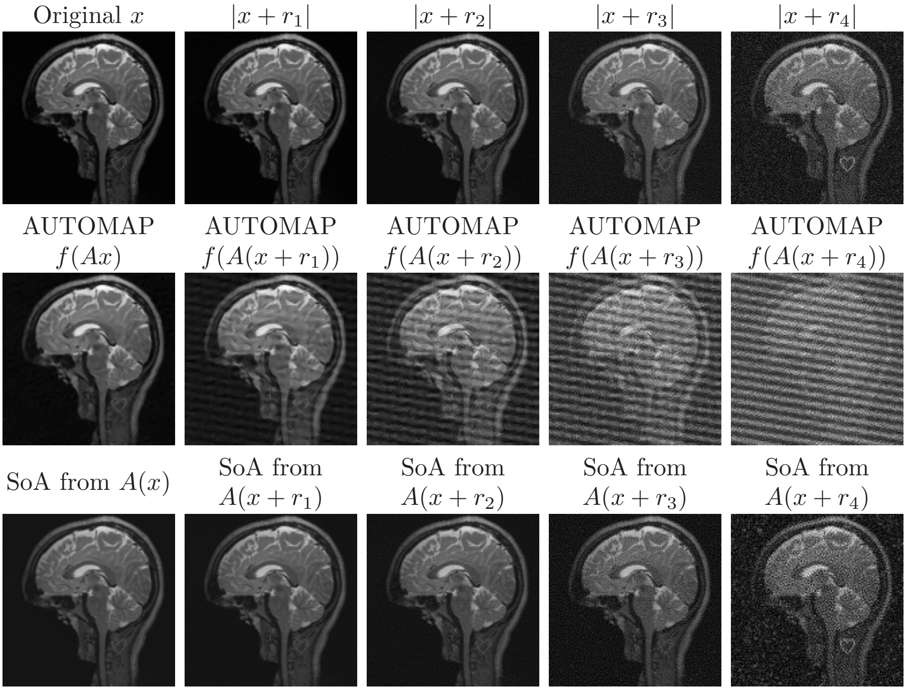


Figure I.7: The experiment from Figure I.3. is repeated, however, by using a different p in Algorithm 1. In particular, Figure I.3. is produced by using $p(x) = x$, however, this figure is produced by using $p(x) = f(Ax)$. Note the substantial difference in the quality of the artefacts in the AUTOMAP reconstruction compared to Figure I.3.

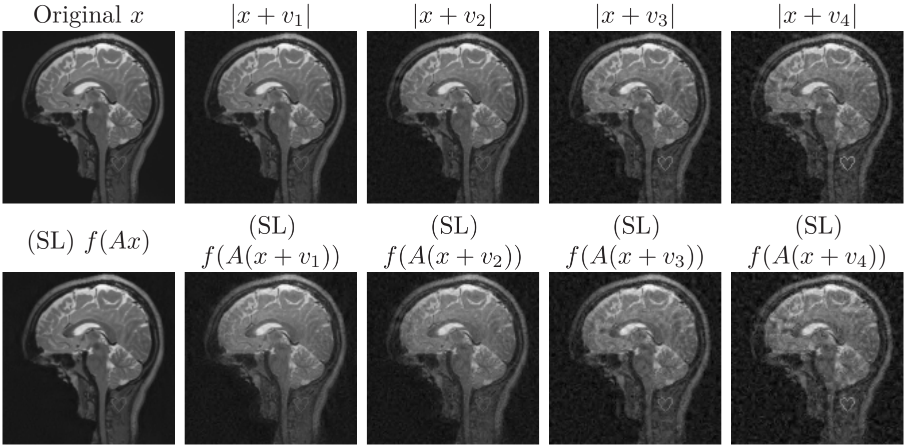


Figure I.8: The recovery mapping f is based on unraveling an optimisation algorithm for solving the (square root) LASSO (SL) optimization problem in (I.14) used in compressed sensing. The perturbations v_j have been chosen so that $\|v_j\|_2 = \|r_j\|_2$ for $j = 1, 2, 3, 4$, where r_j are the perturbations used for Figure 3 in the main paper. The sampling operator $A \in \mathbb{C}^{m \times N}$ is the same in both experiments. As is evident from the images, this compressed sensing type recovery mapping f has a local Lipschitz constant that is quite reasonable, whereas the AUTOMAP network from Figure 3 in the main paper has a very large local Lipschitz constant.

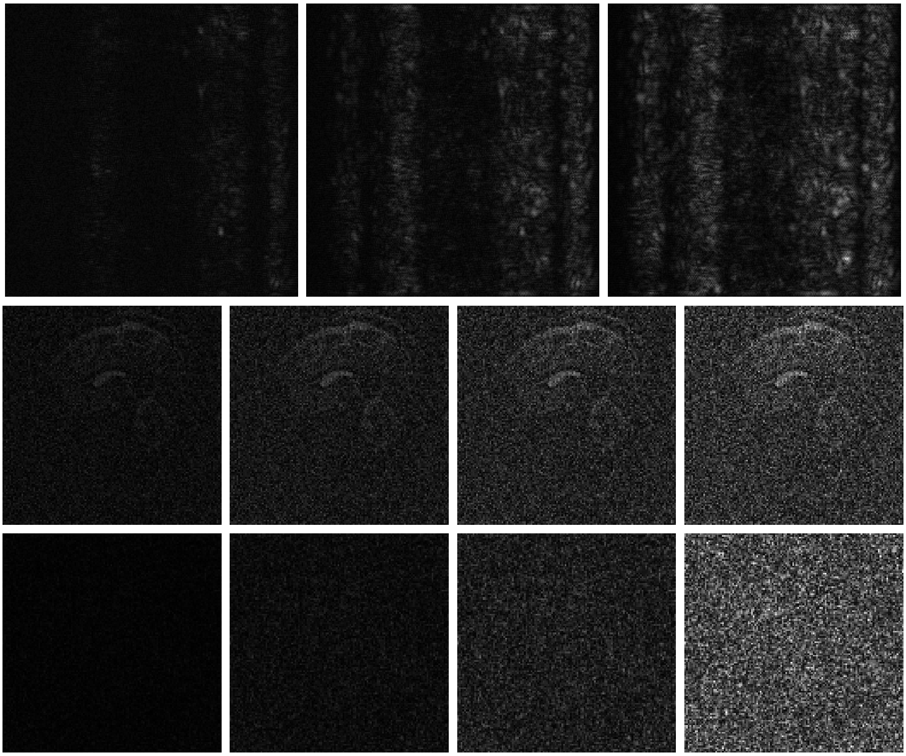


Figure I.9: First row: We visualise the perturbations $|r_1|, |r_2|$ and $|r_3|$ used in Figure I.1 in the main manuscript to create the instabilities for the Deep MRI network. These perturbations have been rescaled to all lie in the same intensity range. The number of iterations in Algorithm 1 is given by $M = 2000, 4000, 6000$. Second and third row: We visualise the perturbations $|r_1|, |r_2|, |r_3|$ and $|r_4|$ used in Figure I.3. (second row) and Figure I.7 (third row) to create the instabilities for the AUTOMAP network. These perturbations have been rescaled to all lie in the same intensity range. The number of iterations in Algorithm 1 is given by $M = 12, 16, 20, 24$, for the second row and $M = 160, 170, 177, 183$ for the third row. The values of λ, γ, η and τ used in in Algorithm 1 are given in Table I.1.

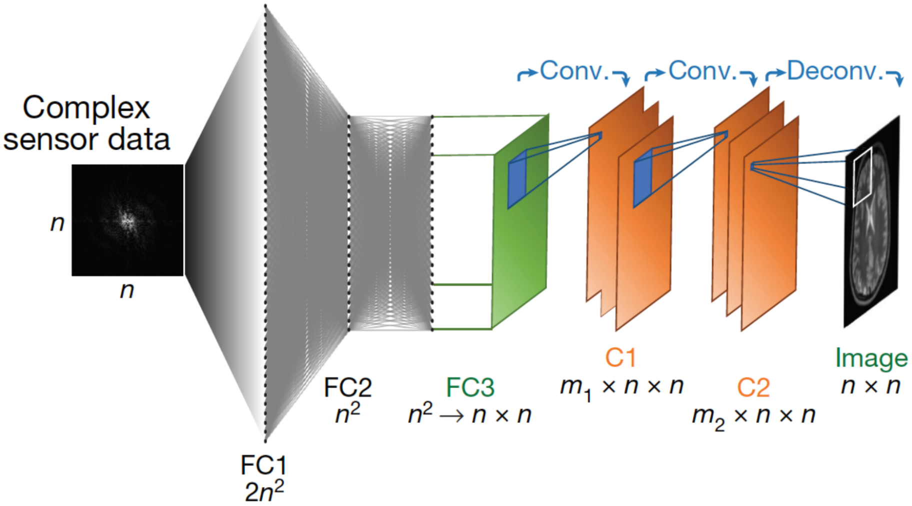


Figure I.10: The AUTOMAP architecture (figure from [56]).

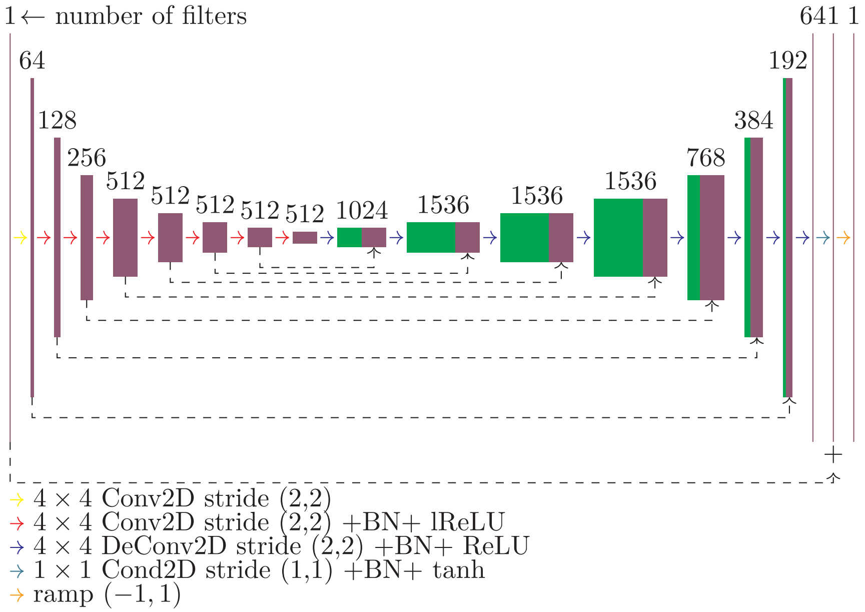

1← number of filters
64
128
256
512
512
512
512
512
1024
1536
1536
1536
768
384
192
641 1
+
→ 4 × 4 Conv2D stride (2,2)
→ 4 × 4 Conv2D stride (2,2) +BN+ lReLU
→ 4 × 4 DeConv2D stride (2,2) +BN+ ReLU
→ 1 × 1 Cond2D stride (1,1) +BN+ tanh
→ ramp (−1, 1)

Figure I.11: DAGAN architecture. Here lReLU is the leaky ReLU function with slope parameter equal to 0.2.

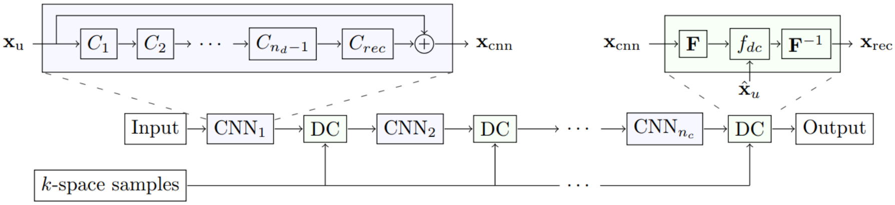


Figure I.12: The DeepMRINet architecture (figure from [44]).

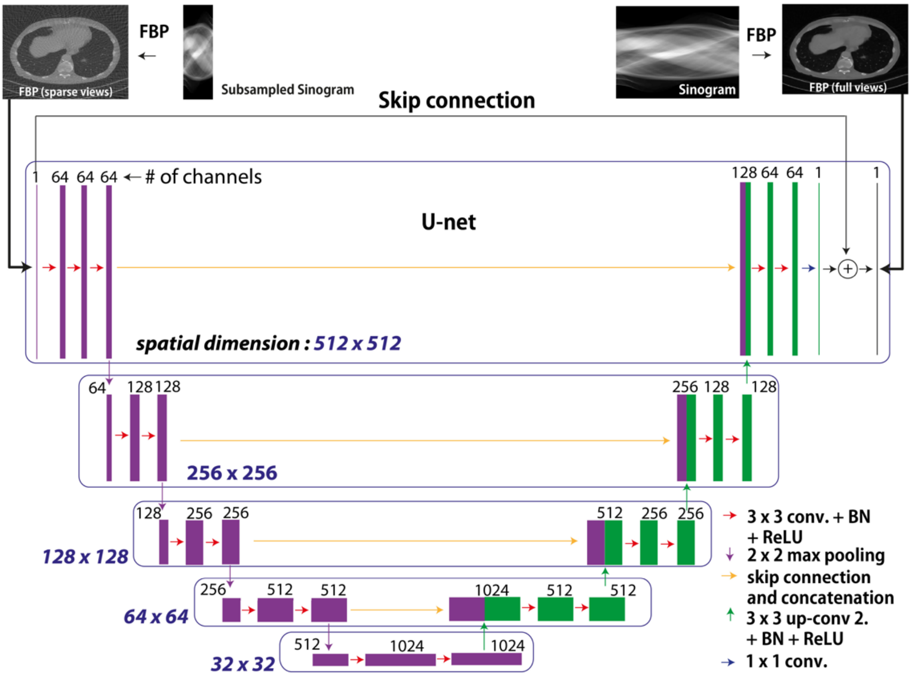


Figure I.13: The Ell 50 and Med 50 architecture (figure from [24]).

Table I.1: Summary of the different choices of parameters leading to the results reported in Figures I.1, I.3, I.4 and I.7.

Neural Network	$\boldsymbol{\lambda}$	$\boldsymbol{\gamma}$	$\boldsymbol{\eta}$	$\boldsymbol{\tau}$	$\boldsymbol{p(x)}$
Deep MRI	0.001	0.9	0.01	0.01	$f(Ax)$
AUTOMAP Fig. I.3	0.1	0.9	0.001	10^{-5}	$f(Ax)$
AUTOMAP Fig. I.7	0.1	0.9	0.001	10^{-5}	x
MRI-VN	1	0.9	0.005	0.001	$f(Ax)$
MED 50	20	0.9	0.005	0.005	$f(Ax)$

References

[1] Adcock, B. and Hansen, A. C. *Compressive imaging.* Cambridge University Press, 2020.

[2] Adcock, B., Hansen, A. C., Poon, C. and Roman, B. 'Breaking the coherence barrier: A new theory for compressed sensing'. In: *Forum of Mathematics, Sigma.* Vol. 5. Cambridge University Press. 2017.

[3] Adler, J. and Öktem, O. 'Solving ill-posed inverse problems using iterative deep neural networks'. In: *Inverse Problems* vol. 33, no. 12 (2017), p. 124007.

[4] Bigdeli, S. A., Zwicker, M., Favaro, P. and Jin, M. 'Deep mean-shift priors for image restoration'. In: *Advances in Neural Information Processing Systems.* 2017, pp. 763–772.

[5] Burger, H. C., Schuler, C. J. and Harmeling, S. 'Image denoising: Can plain neural networks compete with BM3D?' In: *IEEE Conference on Computer Vision and Pattern Recognition.* June 2012, pp. 2392–2399.

[6] Caballero, J., Price, A. N., Rueckert, D. and Hajnal, J. V. 'Dictionary learning and time sparsity for dynamic MR data reconstruction'. In: *IEEE Transactions on Medical Imaging* vol. 33, no. 4 (2014), pp. 979–994.

[7] Candès, E. J., Romberg, J. and Tao, T. 'Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information'. In: *IEEE Transactions Information Theory* vol. 52, no. 2 (2006), pp. 489–509.

[8] Candes, E. J., Wakin, M. B. and Boyd, S. P. 'Enhancing sparsity by reweighted ℓ^1 minimization'. In: *Journal of Fourier Analysis and Applications* vol. 14, no. 5-6 (2008), pp. 877–905.

[9] Donoho, D. L. 'Compressed sensing'. In: *IEEE Transactions Information Theory* vol. 52, no. 4 (2006), pp. 1289–1306.

[10] Elad, M. 'Deep, Deep Trouble - Deep Learning's Impact on Image Processing, Mathematics, and Humanity'. In: *SIAM News* vol. May (2017).

[11] Engl, H. W., Hanke, M. and Neubauer, A. *Regularization of inverse problems.* Vol. 375. Springer Science & Business Media, 1996.

[12] Fan, Q. et al. 'MGH–USC Human Connectome Project datasets with ultra-high b-value diffusion MRI'. In: *Neuroimage* vol. 124 (2016), pp. 1108–1114.

[13] Fawzi, A., Moosavi-Dezfooli, S.-M. and Frossard, P. 'The robustness of deep networks: A geometrical perspective'. In: *IEEE Signal Processing Magazine* vol. 34, no. 6 (2017), pp. 50–62.

[14] Girshick, R., Donahue, J., Darrell, T. and Malik, J. 'Rich feature hierarchies for accurate object detection and semantic segmentation'. In: *IEEE Conference on Computer Vision and Pattern Recognition.* June 2014, pp. 580–587.

[15] Gottschling, N. M., Antun, V., Adcock, B. and Hansen, A. C. 'The troublesome kernel: why deep learning for inverse problems is typically unstable'. In: *Submitted* (2020).

[16] Gull, S. F. and Daniell, G. J. 'Image reconstruction from incomplete and noisy data'. In: *Nature* vol. 272, no. 5655 (1978), p. 686.

[17] Gupta, H., Jin, K. H., Nguyen, H. Q., McCann, M. T. and Unser, M. 'CNN-based projected gradient descent for consistent CT image reconstruction'. In: *IEEE transactions on medical imaging* vol. 37, no. 6 (2018), pp. 1440–1453.

[18] Hammernik, K. et al. 'Learning a variational network for reconstruction of accelerated MRI data'. In: *Magnetic resonance in medicine* vol. 79, no. 6 (2018), pp. 3055–3071.

[19] Hansen, P. C. 'The L-Curve and its Use in the Numerical Treatment of Inverse Problems'. In: *Computational inverse problems in electrocardiology, advances in computational bioengineering*. WIT Press, 2000, pp. 119–142.

[20] Haug, K. M. 'Stability of Adaptive Neural Networks for Image Reconstruction'. https://www.duo.uio.no/handle/10852/69486. MA thesis. 2019.

[21] He, K., Zhang, X., Ren, S. and Sun, J. 'Delving deep into rectifiers: Surpassing human-level performance on imagenet classification'. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.

[22] Huynh-Thu, Q. and Ghanbari, M. 'Scope of validity of PSNR in image/video quality assessment'. In: *Electronics Letters* vol. 44, no. 13 (2008), pp. 800–801.

[23] Ioffe, S. and Szegedy, C. 'Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift'. In: *The 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. PMLR, July 2015, pp. 448–456.

[24] Jin, K. H., McCann, M. T., Froustey, E. and Unser, M. 'Deep convolutional neural network for inverse problems in imaging'. In: *IEEE Transactions Image Processing* vol. 26, no. 9 (2017), pp. 4509–4522.

[25] Jung, H., Ye, J. C. and Kim, E. Y. 'Improved k–t BLAST and k–t SENSE using FOCUSS'. In: *Physics in Medicine & Biology* vol. 52, no. 11 (2007), p. 3201.

[26] Kanbak, C., Moosavi-Dezfooli, S.-M. and Frossard, P. 'Geometric robustness of deep networks: analysis and improvement'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4441–4449.

[27] Kingma, D. and Ba, J. 'Adam: A method for stochastic optimization'. In: *International Conference on Learning Representations*. 2015.

[28] Kobler, E., Klatzer, T., Hammernik, K. and Pock, T. ‘Variational networks: connecting variational methods and deep learning’. In: *German Conference on Pattern Recognition.* Springer. 2017, pp. 281–293.

[29] Krizhevsky, A., Sutskever, I. and Hinton, G. E. ‘Imagenet classification with deep convolutional neural networks’. In: *Advances in Neural Information Processing Systems.* 2012, pp. 1097–1105.

[30] Landweber, L. ‘An iteration formula for Fredholm integral equations of the first kind’. In: *American Journal of Mathematics* vol. 73, no. 3 (1951), pp. 615–624.

[31] LeCun, Y., Bengio, Y. and Hinton, G. ‘Deep learning’. In: *Nature* vol. 521, no. 7553 (2015), p. 436.

[32] Lucas, A., Iliadis, M., Molina, R. and Katsaggelos, A. K. ‘Using deep neural networks for inverse problems in imaging: beyond analytical methods’. In: *IEEE Signal Processing Magazine* vol. 35, no. 1 (2018), pp. 20–36.

[33] Lustig, M., Donoho, D. and Pauly, J. M. ‘Sparse MRI: The application of compressed sensing for rapid MR imaging’. In: *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* vol. 58, no. 6 (2007), pp. 1182–1195.

[34] Ma, J. and März, M. ‘A multilevel based reweighting algorithm with joint regularizers for sparse recovery’. In: *arXiv:1604.06941* (2016).

[35] Ma, J. et al. ‘Shearlet-based compressed sensing for fast 3D cardiac MR imaging using iterative reweighting’. In: *Physics in Medicine & Biology* vol. 63, no. 23 (2018), p. 235004.

[36] Mardani, M. et al. ‘Neural proximal gradient descent for compressive imaging’. In: *Advances in Neural Information Processing Systems.* 2018, pp. 9573–9583.

[37] McCann, M. T., Jin, K. H. and Unser, M. ‘Convolutional Neural Networks for Inverse Problems in Imaging: A Review’. In: *IEEE Signal Processing Magazine* vol. 34, no. 6 (Nov. 2017), pp. 85–95.

[38] Moosavi-Dezfooli, S., Fawzi, A., Fawzi, O. and Frossard, P. ‘Universal Adversarial Perturbations’. In: *IEEE Conference on computer vision and pattern recognition.* July 2017, pp. 86–94.

[39] Moosavi-Dezfooli, S., Fawzi, A. and Frossard, P. ‘DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks’. In: *IEEE Conference on Computer Vision and Pattern Recognition.* June 2016, pp. 2574–2582.

[40] Pruessmann, K. P., Weiger, M., Scheidegger, M. B., Boesiger, P. et al. ‘SENSE: sensitivity encoding for fast MRI’. In: *Magnetic Resonance in Medicine* vol. 42, no. 5 (1999), pp. 952–962.

[41] Rick Chang, J., Li, C.-L., Poczos, B., Vijaya Kumar, B. and Sankaranarayanan, A. C. ‘One Network to Solve Them All–Solving Linear Inverse Problems Using Deep Projection Models’. In: *Proceedings of the IEEE International Conference on Computer Vision.* 2017, pp. 5888–5897.

[42] Roth, S. and Black, M. J. ‘Fields of experts’. In: *International Journal of Computer Vision* vol. 82, no. 2 (2009), p. 205.

[43] Rudin, L. I., Osher, S. and Fatemi, E. ‘Nonlinear total variation based noise removal algorithms’. In: *Physica D: Nonlinear Phenomena* vol. 60, no. 1-4 (1992), pp. 259–268.

[44] Schlemper, J., Caballero, J., Hajnal, J. V., Price, A. and Rueckert, D. ‘A deep cascade of convolutional neural networks for MR image reconstruction’. In: *International Conference on Information Processing in Medical Imaging*. Springer. 2017, pp. 647–658.

[45] Simard, P. Y., Steinkraus, D. and Platt, J. C. ‘Best practices for convolutional neural networks applied to visual document analysis’. In: *Seventh International Conference on Document Analysis and Recognition.* Aug. 2003, pp. 958–963.

[46] Simonyan, K. and Zisserman, A. ‘Very Deep Convolutional Networks for Large-Scale Image Recognition’. In: *International Conference on Learning Representations.* 2015.

[47] Strack, R. ‘Imaging: AI transforms image reconstruction’. In: *Nature Methods* vol. 15, no. 5 (2018), p. 309.

[48] Studer, V. et al. ‘Compressive Fluorescence Microscopy for Biological and Hyperspectral Imaging’. In: *Proceedings of the National Academy of Sciences* vol. 109, no. 26 (2011), pp. 1679–1687.

[49] Szegedy, C. et al. ‘Intriguing properties of neural networks’. In: *Proceedings of the International Conference on Learning Representations.* 2014.

[50] Tezcan, K. C., Baumgartner, C. F., Luechinger, R., Pruessmann, K. P. and Konukoglu, E. ‘MR Image Reconstruction Using Deep Density Priors’. In: *IEEE Transactions on Medical Imaging* vol. 38, no. 7 (July 2019), pp. 1633–1642.

[51] Uecker, M. et al. ‘Berkeley advanced reconstruction toolbox’. In: *Proceedings of the International Society for Magnetic Resonance in Medicine.* Vol. 23. 2015, p. 2486.

[52] Uecker, M. et al. ‘ESPIRiT-an eigenvalue approach to autocalibrating parallel MRI: where SENSE meets GRAPPA’. In: *Magnetic Resonance in Medicine* vol. 71, no. 3 (2014), pp. 990–1001.

[53] van den Berg, E. and Friedlander, M. P. ‘Probing the Pareto frontier for basis pursuit solutions’. In: *SIAM Journal on Scientific Computing* vol. 31, no. 2 (2008), pp. 890–912.

[54] Yang, G. et al. ‘DAGAN: Deep De-Aliasing Generative Adversarial Networks for Fast Compressed Sensing MRI Reconstruction’. In: *IEEE Transactions on Medical Imaging* (2017).

[55] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A. and Oliva, A. ‘Learning deep features for scene recognition using places database’. In: *Advances in Neural Information Processing Systems.* 2014, pp. 487–495.

[56] Zhu, B., Liu, J. Z., Cauley, S. F., Rosen, B. R. and Rosen, M. S. 'Image reconstruction by domain-transform manifold learning'. In: *Nature* vol. 555, no. 7697 (Mar. 2018), p. 487.

Paper III

Uniform recovery in infinite-dimensional compressed sensing and applications to structured binary sampling

Ben Adcock, Vegard Antun, Anders C. Hansen

Submitted for publication in *Applied and Computational Harmonic Analysis*.

Abstract

Infinite-dimensional compressed sensing deals with the recovery of analog signals (functions) from linear measurements, often in the form of integral transforms such as the Fourier transform. This framework is well-suited to many real-world inverse problems, which are typically modelled in infinite-dimensional spaces, and where the application of finite-dimensional approaches can lead to noticeable artefacts. Another typical feature of such problems is that the signals are not only sparse in some dictionary, but possess a so-called local sparsity in levels structure. Consequently, the sampling scheme should be designed so as to exploit this additional structure. In this paper, we introduce a series of uniform recovery guarantees for infinite-dimensional compressed sensing based on sparsity in levels and so-called multilevel random subsampling. By using a weighted ℓ^1-regularizer we derive measurement conditions that are sharp up to log factors, in the sense they agree with those of certain oracle estimators. These guarantees also apply in finite dimensions, and improve existing results for unweighted ℓ^1-regularization. To illustrate our results, we consider the problem of binary sampling with the Walsh transform using orthogonal wavelets. Binary sampling is an important mechanism for certain imaging modalities. Through carefully estimating the local coherence between the Walsh and wavelet bases, we derive the first known recovery guarantees for this problem.

Keywords: Infinite-dimensional compressed sensing, uniform recovery, Walsh sampling, wavelet recovery, sparsity in levels, local coherence

Mathematics Subject Classification (2010): 94A20, 42C40, 42C10, 15B52

III.1 Introduction

Compressive sensing (CS), introduced by Candes, Romberg & Tao in [10] and Donoho in [14], has been an area of substantial research during the last decade. The key assumption, which lays the foundation for this field of research, is that a sparse vector $x \in \mathbb{C}^M$ can be recovered from an underdetermined system of linear equations, using, for instance, convex optimization algorithms [15, 16].

Imaging has been one of the most successful areas of application of CS. However, in this area, the sparsity assumption is typically too general. Examples include all applications using Fourier samples – such as Magnetic Resonance Imaging (MRI) [22, 24, 25], surface scattering [21], Computerized Tomography (CT) and electron microscopy – as well as applications using binary sampling, e.g. fluorescence microscopy [29], lensless imaging [33] and numerous other optical imaging modalities [6, 17, 32]. Natural images, when sparsified via a wavelet (or more generally, X-let) transform, are not only sparse, but have specific sparsity structure [3, 27]. For wavelets, which will be our sparsifying transform in this paper, natural images have coefficients where most of the large entries are concentrated at the coarse scales, and progressively fewer at the fine scales (termed *asymptotic sparsity* in [3]).

In the presence of structured sparsity, it is natural to ask how best to promote this additional structure. In [3] it was proposed to do this via the sampling operator. Wavelets partition Fourier space into dyadic bands corresponding to distinct scales. Hence, by choosing Fourier samples in these bands corresponding to the local sparsities, one obtains as structured sampling scheme – a so-called *multilevel sampling* scheme – which promotes the asymptotic sparsity structure. The practical benefits of such schemes have been demonstrated in [27] for various different imaging modalities, including MRI, Nuclear Magnetic Resonance (NMR) spectroscopy, fluorescence microscopy and Helium Atom Scattering. Theoretical analysis has been presented in [3] (nonuniform recovery) and [7, 23] (uniform recovery in the finite-dimensional setting).

III.1.1 Main results

This paper has two main objectives. First, we generalize existing uniform recovery guarantees [7, 23] from the finite-dimensional to the infinite-dimensional setting. This extension is important for practical imaging. Although much of the compressive imaging literature considers the recovery of discrete images (i.e. finite-dimensional arrays) from discrete measurements (e.g. the discrete Fourier transform), modalities such as MRI, NMR and others are naturally analog, and hence better modelled over the continuum (i.e. functions, and the continuous Fourier transform). Indeed, as we will see in Section III.2.3, discretizing such a problem leads to measurement mismatch [11], and in the case of wavelet recovery, the wavelet crime [28, p. 232], both of which can introduce artefacts in the reconstruction [19]. In this paper, we consider signals as functions $f \in L^2([0,1))$ and work with continuous integral transforms, thus avoiding these pitfalls.

In our theoretical analysis, we also improve the uniform recovery guarantee given in previous works [7, 23]. Unlike previous results, our recovery guarantees are, up to log factors, optimal: specifically, they agree with those of the oracle least-square estimator based on *a priori* knowledge of the support [1]. We do this by replacing the standard ℓ^1-minimization decoder by a certain weighted ℓ^1-minimization decoder; an idea originally proposed in [31].

Our second objective is to consider binary sampling. Previous works have addressed the case of (discrete or continuous) Fourier sampling. Yet many imaging modalities, e.g. fluorescence microscopy and lensless imaging, require binary sampling operators. To do so, we replace the Fourier transform

$$\mathcal{F}f(\omega) \coloneqq \int_{[0,1)} f(x)e^{-2\pi\omega x}\,\mathrm{d}x, \qquad f \in L^2([0,1)),$$

by the binary *Walsh transform*

$$\mathcal{W}f(n) \coloneqq \int_{[0,1)} f(x)w_n(x)\,\mathrm{d}x, \qquad f \in L^2([0,1))$$

where $w_n\colon [0,1) \to \{+1,-1\}$, $n \in \mathbb{Z}_+ \coloneqq \{0,1,\dots\}$ denote the Walsh functions. This is a widely used sampling operator in binary imaging [29, 33], and often goes under the name of Hadamard sampling in the discrete case. Working with this continuous transform, we provide analogous guarantees for binary sampling to those for Fourier sampling. As a side note, we remark that working in the continuous setting also simplifies the analysis (specifically, the derivation of so-called *local coherence* estimates) over working directly with the discrete setup.

We note that in this paper, we only consider recovery guarantees for one dimensional functions. We expect that the setup for higher dimensional function will deviate slightly from what we present here, and we will save this discussion for future work.

The outline of the remainder of this paper is as follows. We commence in Section III.2 by reviewing previous work, and in particular, the existing finite-dimensional theory. We then introduce an abstract infinite-dimensional model for isometries U acting on $\ell^2(\mathbb{N})$ in Section III.3. Here we will derive sufficient conditions for such operators to provide uniform recovery guarantees. In Section III.4 we continue this work by finding conditions for which the cross-Gramian U between a wavelet and Walsh basis satisfies these conditions. Finally in Section III.5, III.6 and III.6.6 we will present proofs of our main results.

III.2 Sparsity in levels in finite dimensions

III.2.1 Notation

For $N \in \mathbb{N}$ and $\Omega \subseteq \{1,\dots,N\}$ we let $P_\Omega \in \mathbb{C}^{N\times N}$ denote the projection onto the linear span of the associated subset of the canonical basis, i.e. for $x \in \mathbb{C}^N$, we have $(P_\Omega x)_i = x_i$ if $i \in \Omega$ and $(P_\Omega x)_i = 0$ if $i \notin \Omega$. Sometimes, we will abuse this notation slightly by assuming $P_\Omega \in \mathbb{C}^{|\Omega|\times N}$, and discard all the zero entries in $P_\Omega x$. Whether we mean $P_\Omega \in \mathbb{C}^{N\times N}$ or $P_\Omega \in \mathbb{C}^{|\Omega|\times N}$ will be clear from the context. If $\Omega = \{N_{k-1}+1,\dots,N_k\}$ we simply write $P_{N_k}^{N_{k-1}} = P_{\{N_{k-1}+1,\dots,N_k\}}$, and simply P_{N_k} if $N_{k-1}=0$.

We call a vector $x \in \mathbb{C}^N$ s-sparse if $|\operatorname{supp}(x)| \leq s$, where $\operatorname{supp}(x) = \{i : x_i \neq 0\}$. We write $A \lesssim B$ if there exits a constant $C>0$ independent of all relevant parameters, so that $A \leq CB$, and similarly for $A \gtrsim B$.

III.2.2 Finite model

Let $V \in \mathbb{C}^{N\times N}$ be a measurement matrix e.g. a Fourier of Hadamard matrix, denoted V_{Four} and V_{Had}, respectively, and let $\Omega \subset \{1,\dots N\}$ with $|\Omega| = m < N$. In a typical finite-dimensional CS setup we consider the recovery of a signal $x \in \mathbb{C}^N$ from measurements $y = P_\Omega V x + e \in \mathbb{C}^m$, where $e \in \mathbb{C}^m$ is a vector of measurement error. If x is sparse in a discrete wavelet basis, one then recovers its coefficients by solving the optimization problem

$$\underset{z\in\mathbb{C}^N}{\text{minimise}} \|z\|_1 \quad \text{subject to} \quad \|P_\Omega V \Psi^{-1} z - y\|_2 \leq \eta \tag{III.1}$$

where $\Psi \in \mathbb{C}^{N\times N}$ is a discrete wavelet transform and $\eta \geq \|e\|_2$ is a noise parameter. Usually one would scale $V \in \mathbb{C}^{N\times N}$ so that it becomes orthonormal and choose an orthonormal wavelet basis, so that the matrix $U = V\Psi^{-1} = V\Psi^T$ acts as an isometry on $\mathbb{C}^N$.

Suppose that U is indeed an isometry. To obtain a uniform recovery guarantee for the above system, one typically first shows that the matrix $A = \frac{1}{\sqrt{p}} P_\Omega U \in \mathbb{C}^{m\times N}$, with $p = \frac{m}{N}$, satisfies the *Restricted Isometry Property* (RIP) with high probability.

Definition III.2.1 (RIP)**.** Let $1 \leq s \leq N$ and $A \in \mathbb{C}^{m\times N}$. The *Restricted Isometry Constant (RIC)* of order s is the smallest $\delta \geq 0$ such that

$$(1-\delta)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1+\delta)\|x\|_2^2 \quad \forall x \in \Sigma_s,$$

where Σ_s denotes the set of s-sparse vectors in $\mathbb{C}^N$. If $0 \leq \delta < 1$ we say that A has the *Restricted Isometry Property (RIP)* of order s.

Theorem III.2.2 ([16, Thm. 6.12])**.** *Suppose the RIC δ_{2s} of a matrix $A \in \mathbb{C}^{m\times N}$ satisfies $\delta_{2s} < 4/\sqrt{41} \approx 0.62$. Then for any $x \in \mathbb{C}^N$ and $e \in \mathbb{C}^m$ with $\|e\|_2 \leq \eta$, any solution $\hat{x} \in \mathbb{C}^N$ of*

$$\underset{z\in\mathbb{C}^N}{\text{minimise}} \|z\|_1 \quad \textit{subject to} \quad \|z - (Ax+e)\|_2 \leq \eta$$

satisfies

$$\|x - \hat{x}\|_2 \leq \frac{C}{\sqrt{s}} \sigma_s(x)_1 + D\eta$$

where $C, D > 0$ are constants dependent on δ_{2s} only and $\sigma_s(x)_1 = \inf\{\|x-z\|_1 : z \in \Sigma_s\}$.

For an isometry $U \in \mathbb{C}^{N\times N}$the question of whether or not $P_\Omega U$ satisfies the RIP is related to the so-called *coherence* of U:

Definition III.2.3 (Coherence)**.** Let $U \in \mathbb{C}^{N\times N}$ be an isometry. The *coherence* of U is

$$\mu(U) = \max_{i,j=1,\dots,N} |U_{ij}|^2 \in [N^{-1}, 1].$$

Theorem III.2.4 ([16, Thm. 12.32])**.** *Let $U \in \mathbb{C}^{N\times N}$ be an isometry and let $0 < \delta, \epsilon < 1$. Suppose $\Omega = \{t_1, \dots t_m\} \subseteq \{1, \dots, N\}$ where each t_k is chosen uniformly and independently at random from the set $\{1, \dots, N\}$. If*

$$m \gtrsim \delta^{-2} \cdot s \cdot N \cdot \mu(U) \cdot \big(\log(2m)\log(2N)\log^2(2s) + \log(\epsilon^{-1})\big)$$

then with probability $1-\epsilon$ the matrix $A = \frac{1}{\sqrt{p}} P_\Omega U \in \mathbb{C}^{m\times N}$, with $p = \frac{m}{N}$, satisfies the RIP of order s with $\delta_s \leq \delta$.

(We slightly abuse notation here in that we allow for possible repeats of the values t_i that make up Ω). Thus if the coherence $\mu(U) \approx N^{-1}$ we obtain the RIP of order s using approximately s measurements up to constants and log factors.

There are, however, two problems with this approach. First, in our setup, where $U = V\Psi^T$ is the product of a Fourier or Hadamard matrix and a discrete wavelet transform, the coherence $\mu(U) \approx 1$. Hence satisfying the RIP requires at least $m \approx N$ measurements. Second, the RIP asserts recovery for *all* s-sparse vectors of wavelet coefficients and thus does not exploit any additional structure these coefficients possess. However, as stated, wavelet coefficients are highly structured: large wavelet coefficients tend to cluster at coarse scales, with coefficients at fine scales being increasingly sparse.

Motivated by this, the following structured sparsity model was introduced in [3]:

Definition III.2.5 (Sparsity in levels)**.** Let $\mathbf{M} = [M_1, \dots, M_r] \in \mathbb{N}^r$, $M_0 = 0$, with $1 \leq M_1 < \cdots < M_r = M$ and let $\boldsymbol{s} = (s_1, \dots, s_r) \in \mathbb{N}^r$ with $s_l \leq M_l - M_{l-1}$, for $l = 1, \dots, r$. We say that the vector $x \in \mathbb{C}^M$ is sparse in levels if

$$|\operatorname{supp}(x) \cap \{M_{l-1}+1, \dots, M_l\}| \leq s_l \quad \text{for } l = 1, \dots, r.$$

In which case we call x, $(\boldsymbol{s}, \mathbf{M})$-sparse, where $\boldsymbol{s}$ and $\mathbf{M}$ are called the local sparsities and sparsity levels, respectively. We denote the set of all $(\boldsymbol{s}, \mathbf{M})$-sparse vectors by $\Sigma_{\boldsymbol{s},\mathbf{M}}$.

As noted above, randomly subsampling an isometry U is a poor measurement protocol for coherent problems such as Fourier–Wavelets. Instead, in [3] it was proposed to sample in the following structured way:

Definition III.2.6 (Multilevel random subsampling)**.** Let $\mathbf{N} = [N_1, \dots, N_r] \in \mathbb{N}^r$, where $1 \leq N_1 < \cdots < N_r = N$ and $\boldsymbol{m} = (m_1, \dots, m_r) \in \mathbb{N}^r$ with $m_k \leq N_k - N_{k-1}$ for $k = 1, \dots, r$, and $N_0 = 0$. For each $k = 1, \dots, r$, let $\Omega_k = \{N_{k-1}+1, \dots, N_k\}$ if $m_k = N_k - N_{k-1}$ and if not, let $t_{k,1}, \dots, t_{k,m_k}$

be chosen uniformly and independently from the set $\{N_{k-1}+1,\dots,N_k\}$, and set $\Omega_k = \{t_{k,1},\dots,t_{k,m_k}\}$. If $\Omega = \Omega_{\mathbf{N},\boldsymbol{m}} = \Omega_1 \cup \dots \cup \Omega_r$ we refer to Ω as an $(\mathbf{N},\boldsymbol{m})$-*multilevel subsampling scheme.*

For this structured model, the following extensions of the RIP was first introduced in [7].

Definition III.2.7 (RIPL)**.** Let $\boldsymbol{s},\mathbf{M} \in \mathbb{N}^r$ be given local sparsities and sparsity levels, respectively. For a matrix $A \in \mathbb{C}^{m\times N}$ the *Restricted Isometry Constant in Levels (RICL)* of order $(\boldsymbol{s},\mathbf{M})$, denoted $\delta_{\boldsymbol{s},\mathbf{M}}$, is the smallest $\delta \geq 0$ such that

$$(1-\delta)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1+\delta)\|x\|_2^2 \quad \forall x \in \Sigma_{\boldsymbol{s},\mathbf{M}}.$$

We say that A has the *Restricted Isometry Property in Levels (RIPL)* if $0 \leq \delta < 1$.

We shall see that this leads to uniform recovery of all $(\boldsymbol{s},\mathbf{M})$-sparse vectors, but first we define the *best* $(\boldsymbol{s},\mathbf{M})$-*term approximation error* of $x \in \mathbb{C}^N$. That is

$$\sigma_{\boldsymbol{s},\mathbf{M}}(x)_p \coloneqq \inf\{\|x-z\|_p : z \in \Sigma_{\boldsymbol{s},\mathbf{M}}\}.$$

Theorem III.2.8 ([7, Thm. 4.4])**.** *Let $\boldsymbol{s},\mathbf{M} \in \mathbb{N}^r$ be local sparsities and sparsity levels, respectively. Let $\alpha_{\boldsymbol{s},\mathbf{M}} = \max_{k,l=1,\dots,r} s_l/s_k$ and $s = s_1+\dots+s_r$. Suppose that the RICL $\delta_{2\boldsymbol{s},\mathbf{M}} \geq 0$ for the matrix $A \in \mathbb{C}^{m\times M}$ satisfies*

$$\delta_{2\boldsymbol{s},\mathbf{M}} < \frac{1}{\sqrt{r(\sqrt{\alpha_{\boldsymbol{s},\mathbf{M}}}+\frac{1}{4})^2+1}}. \tag{III.2}$$

Then, for $x \in \mathbb{C}^M$ and $e \in \mathbb{C}^m$ with $\|e\|_2 \leq \eta$, any solution $\hat{x}$ of

$$\operatorname*{minimise}_{z\in\mathbb{C}^M} \|z\|_1 \quad \textit{subject to} \quad \|z-(Ax+e)\|_2 \leq \eta$$

satisfies

$$\|x-\hat{x}\|_2 \leq (C+C'(r\alpha_{\boldsymbol{s},\mathbf{M}})^{1/4})\frac{\sigma_{s,\mathbf{M}}(x)_1}{\sqrt{s}} + (D+D'(r\alpha_{\boldsymbol{s},\mathbf{M}})^{1/4})\eta$$

where $C,C',D,D' > 0$ are constants which only dependent on $\delta_{2\boldsymbol{s},\mathbf{M}}$.

In [23] the authors investigated conditions under which a subsampled isometry $U \in \mathbb{C}^{N\times N}$ satisfies the RIPL. In was shown that the number of samples required to satisfy the RIPL was related to the so-called *local coherence* properties of U:

Definition III.2.9. Let $U \in C^{N\times N}$ be an isometry and $\mathbf{N},\mathbf{M} \in \mathbb{N}^r$ be given sampling and sparsity levels. The *local coherence* of U is

$$\mu_{k,l} = \mu_{k,l}(\mathbf{N},\mathbf{M}) = \{\max|U_{ij}|^2 : i = N_{k-1}+1,\dots,N_k, j = M_{l-1}+1,\dots,M_l\}.$$

Theorem III.2.10 ([23, thm. 3.2])**.** *Let $U \in \mathbb{C}^{N\times N}$ be an isometry. Let $r \in \mathbb{N}$, $0 < \delta,\epsilon < 1$, and $0 \leq r_0 \leq r$. Let $\Omega = \Omega_{\mathbf{N},\boldsymbol{m}}$ be an $(\mathbf{N},\boldsymbol{m})$-multilevel random* *subsampling scheme. Let $\tilde{m} = m_{r_0+1} + \ldots + m_r$ and $s = s_1 + \ldots + s_r$. Suppose that the m_ks satisfy*

$$m_k = N_k - N_{k-1}, \quad \textit{for } k = 1, \ldots, r_0, \tag{III.3}$$

and

$$m_k \gtrsim \delta^{-2} \cdot (N_k - N_{k-1}) \cdot \left(\sum_{l=1}^{r} s_l \mu_{k,l} \right) \cdot \left(r \log(2\tilde{m}) \log(2N) \log^2(2s) + \log(\epsilon^{-1}) \right) \tag{III.4}$$

for $k = r_0 + 1, \ldots, r$. Then the matrix

$$A = \begin{bmatrix} \frac{1}{\sqrt{p_1}} P_{\Omega_1} U \\ \vdots \\ \frac{1}{\sqrt{p_r}} P_{\Omega_r} U \end{bmatrix} \quad \textit{where } p_k = \frac{m_k}{N_k - N_{k-1}} \quad \textit{for } k = 1, \ldots, r \tag{III.5}$$

satisfies the RIPL of order $(\boldsymbol{s}, \mathbf{M})$ with constant $\delta_{\boldsymbol{s},\mathbf{M}} \leq \delta$.

This theorem characterizes the number of local measurements m_k needed to ensure uniform recovery explicitly in terms of local sparsities s_k and local coherences $\mu_{k,l}$. In particular, if the local coherences are suitably well-behaved, then recovery may still be possible from highly subsampled measurements, even though the global coherence may be high (see next). Note that the condition (III.3), whereby the first r_0 sampling levels are saturated, models practical imaging scenarios where the low Fourier frequencies are typically fully sampled.

To illustrate this theorem, in [4] the authors consider the one-dimensional discrete Fourier sampling problem with sparsity in Haar wavelets. For the Haar wavelet basis we choose an ordering where the first level $\{M_0 + 1, M_1\} = \{1, 2\}$ consists of the scaling function and mother wavelet and the subsequent levels are chosen so that $\{M_{l-1} + 1, \ldots, M_l\} = \{2^{l-1} + 1, \ldots, 2^l\}$ consists of the wavelets at scale $l - 1$. This gives the sparsity levels

$$\mathbf{M} = [2^1, 2^2, \ldots, 2^r],$$

where $r = \log_2(N)$ (assumed to be an integer). Next we define the entries in the Fourier matrix $V_{\text{Four}} \in \mathbb{C}^{N \times N}$ as

$$(V_{\text{Four}})_{\omega=-N/2+1,\ j=1}^{N/2,\ N} = \frac{1}{\sqrt{N}} \exp(2\pi i (j-1)\omega/N),$$

where we have started the ordering of the rows with negative indices for convenience. We define the sampling levels for the frequencies ω in dyadic bands with $W_1 = \{0, 1\}$ and

$$W_{k+1} = \{-2^k + 1, \ldots, -2^{k-1}\} \cup \{2^{k-1} + 1, \ldots, 2^k\}, \quad k = 1, \ldots, r-1.$$

Notice that for a suitable reordering of the rows of V_{Four} these bands corresponds to the sampling levels $\mathbf{N} = [2^1, 2^2, \ldots, 2^r]$.

Theorem III.2.11 ([23, Cor. 3.3])**.** *Let $N = 2^r$ for some $r \geq 1$ and let $U = V_{Four}\Psi^{-1} \in \mathbb{C}^{N\times N}$, where Ψ is the Haar wavelet matrix. Let $0 < \delta, \epsilon < 1$ and let $\mathbf{N} = \mathbf{M} = [2^1, \dots, 2^r]$. Let $m = m_1 + \cdots m_r$ and $s = s_1 + \cdots s_r$. For each $k = 1, \dots, r$ suppose we draw m_k Fourier samples from band W_k randomly and independently, where*

$$m_k \gtrsim \delta^{-2} \cdot \left(\sum_{l=1}^{r} 2^{-|k-l|} s_l \right) \left(r \log(2m) \log(2N) \log^2(2s) + \log(\epsilon^{-1}) \right).$$

Then with probability at least $1 - \epsilon$ the matrix (III.5) satisfies the RIPL with constant $\delta_{s,\mathbf{M}} \leq \delta$.

Here, for convenience, we have taken $r_0 = 0$; see [23] for further discussion on this point.

III.2.3 Shortcomings

These results have two primary shortcomings, which we now discuss in further detail. The key issue is that they are limited to finite dimensions. As noted in Section III.1, applying finite-dimensional recovery procedures to analog problems can result in artefacts. For simplicity, let $N = 2^p$. We have argued that analog signals should be modelled as elements in $L^2([0,1))$, rather than $\mathbb{C}^N$. Yet, above we have tried to use discrete tools for recovering the signal $f \in L^2([0,1))$ by replacing $\mathcal{W}f$ and $\mathcal{F}f$ with V_{Had} and V_{Four}, respectively. Next, we argue that this construction leads to both *measurement mismatch* and the *wavelet crime.*

Let $\chi_{[a,b)}$ denote step functions on the interval $[a,b)$ and set $\Delta_{k,p} = [k2^{-p}, (k+1)2^{-p})$. We see that replacing $\mathcal{W}f$ with $V_{\mathrm{Had}} \in \mathbb{C}^{N\times N}$ is equivalent to replacing f by e.g. $\tilde{f} = \sum_{k=0}^{N-1} c_k \chi_{\Delta_{k,r}}$ for some $c \in \mathbb{C}^N$, since $\mathcal{W}\tilde{f} = V_{\mathrm{Had}} c$. Clearly, $\mathcal{W}\tilde{f}$ will be a poor approximation to $\mathcal{W}f$. We refer to this as measurement mismatch.

Next let ϕ^0, ϕ^1 denote a scaling function and wavelet, respectively, and set $\phi^s_{j,k} = 2^{j/2}\phi^s(2^j \cdot - k)$ for $s \in \{0,1\}$. By construction the solution $\hat{x}$ of (2.1) will be the coefficients of a function $\hat{f}$ written in a basis consisting of both wavelets and scaling functions. Equivalently we can represent $\hat{f}$ in the basis $\{\phi^0_{j,k}\}_{k=0}^{N-1}$ using the coefficients $c = \Psi^{-1}\hat{x} \in \mathbb{C}^N$. The wavelet crime is whenever we let c, represent pointwise samples of f i.e. $c_k = f(k/N)$.

What does this mean for reconstruction? To illustrate the issue, we provide a similar example to the first numerical simulation in [2], showing how finite-dimensional compressed sensing fails to recover even a function that is 1-sparse (meaning it has only one non-zero coefficient) in its wavelet decomposition. Indeed, in Figure III.1 we consider the problem of recovering a function f from samples of the continuous Walsh transform. In particular, we choose $f(t) = \phi_{4,4}(t)$, where ϕ is the Daubechies scaling function, corresponding to the wavelet with four vanishing moments. Figure III.1 shows the poor performance of CS using the discrete finite-dimensional setup when applied to a continuous problem. Conversely, the infinite-dimensional CS approach, which we develop in the next sections, gives a much higher fidelity reconstruction from exactly the same samples as used in the finite-dimensional case. In fact, the infinite-dimensional CS reconstruction recovers f perfectly up to numerical errors occurring from solving the optimization problem. We also observe the slightly paradoxical phenomenon in the finite-dimensional case: more samples do not improve performance. This is due to the fact that the finite-dimensional CS solution with full sampling coincides with the truncated Walsh series (direct inversion) approximation. This approximation is clearly highly suboptimal, as demonstrated in Figure III.1.

We note in passing that the above crimes stem from too early a discretization of the inverse problem. Our infinite-dimensional CS approach replaces $V_{\mathrm{Had}}\Psi^{-1}$ by a finite section of the an isometry $U \in \mathcal{B}(\ell^2(\mathbb{N}))$ representing change of basis between the continuous Fourier or Walsh transform and wavelet basis.

On a related note, even if one were to ignore the above issues, estimating the local coherences $\mu_{k,l}$ in the discrete setting for anything but the Haar wavelet becomes extremely complicated. Conversely, by moving to the continuous setting, these estimates become much easier to derive. We do this later in the paper for arbitrary Daubechies' wavelets with the Walsh transform.

The second shortcoming relates to Theorem III.2.8. It says that we can guarantee recovery of all sparse signals provided the matrix $A \in \mathbb{C}^{m\times M}$ satisfies the RIPL with constant

$$\delta_{2\boldsymbol{s},\mathbf{M}} < \frac{1}{\sqrt{r(\sqrt{\alpha_{\boldsymbol{s},\mathbf{M}}} + \frac{1}{4})^2 + 1}}.$$

Here r is the number of levels and $\alpha_{\boldsymbol{s},\mathbf{M}} = \max_{k,l=1,\dots,r} s_l/s_k$ is the sparsity ratio. Inserting the above inequality into Theorem III.2.10 gives a sampling condition of the form

$$m_k \gtrsim r \cdot \alpha_{\boldsymbol{s},\mathbf{M}} \cdot (N_k - N_{k-1}) \cdot \left(\sum_{l=1}^{r} \mu_{k,l} s_l \right) \cdot L$$

where L is the log factors. This means that the sparsity ratio $\alpha_{\boldsymbol{s},\mathbf{M}}$ will affect the sampling condition in all sampling levels. Thus for signals where we expect the local sparsities to vary greatly from level to level (e.g. wavelets) this will lead to a unreasonably high number of samples.

To overcome this problem, using an idea from [31], we replace the ℓ^1-regularizer in the optimization problem (III.1) with a weighted ℓ^1-regularizer. For a suitable choice of weights, this removes the factor of $\alpha_{\boldsymbol{s},\mathbf{M}}$ in the various measurement conditions. As we show, these guarantees are optimal up to constants and log factors.

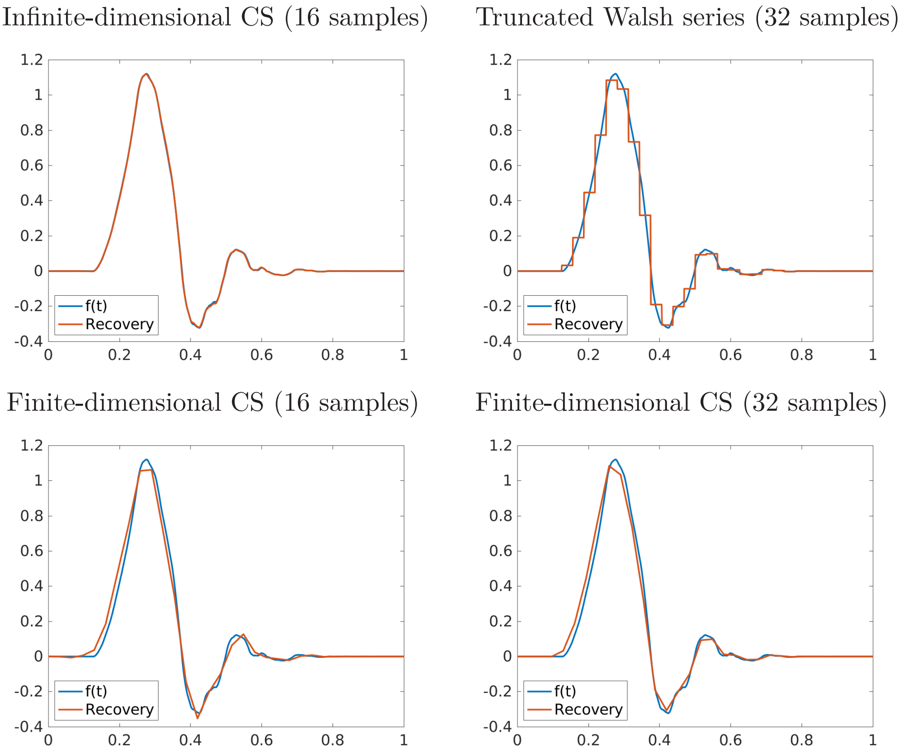


Figure III.1: Reconstructions (using Walsh samples) of $f(t) = \phi_{4,4}(t)$, where ϕ is the Daubechies scaling function, corresponding the wavelets with four vanishing moments. Upper left: Reconstruction from the first 16 Walsh samples using an infinite-dimensional CS model. Upper right: Truncated Walsh series based on the first 32 Walsh samples. Lower left: Reconstruction from the first 16 Walsh samples using the finite-dimensional (32×32) CS model. Lower right: Reconstruction from the first 32 Walsh samples using the finite-dimensional (32×32) CS model. In theory, the right images should be the same, however, numerical errors in the optimisation cause the difference.

III.3 Extensions to infinite dimensions

III.3.1 Setup

We will continue with the notation we introduced above, extended to infinite dimensions. That is, we assume that the signal f is an element of $L^2([0,1))$. We still let P_Ω denote the projection onto the canonical basis, but we now let it be an element in either $\mathcal{B}(\ell^2(\mathbb{N}))$ or $\mathcal{B}(\ell^2(\mathbb{N}), \mathbb{C}^{|\Omega|})$. Similarly we call a vector $x \in \ell^2(\mathbb{N})$ $(\boldsymbol{s}, \mathbf{M})$-sparse if $P_M x$ is $(\boldsymbol{s}, \mathbf{M})$-sparse and $P_M^\perp x = 0$. Here $M = M_r$ and we refer to it as the *sparsity bandwidth* of x. For an isometry $U \in \mathcal{B}(\ell^2(\mathbb{N}))$

we define the coherence of U as $\mu(U) = \sup\{|U_{ij}|^2 : i, j \in \mathbb{N}\}$.

Next we describe the setup for a general sampling basis $B_{\mathrm{sa}} = \{b_1^{\mathrm{sa}}, b_2^{\mathrm{sa}}, b_3^{\mathrm{sa}}, \dots,\}$ and a sparsifying basis $B_{\mathrm{sp}} = \{b_1^{\mathrm{sp}}, b_2^{\mathrm{sp}}, b_3^{\mathrm{sp}}, \dots,\}$, both assumed to be orthonormal bases of $L^2([0,1))$. In Section III.4, we will specialize this so that B_{sa} is the Walsh sampling basis and B_{sp} is a wavelet sparsifying basis. This will enable us to derive concrete recovery guarantees for f. The setup below is, however, completely general.

For the two bases B_{sa} and B_{sp} we can represent f using the coefficients $y = \{\langle f, b_n^{\mathrm{sa}}\rangle\}_{n\in\mathbb{N}}$ and $x = \{\langle f, b_n^{\mathrm{sp}}\rangle\}_{n\in\mathbb{N}}$, respectively. To change the representation from B_{sa} to B_{sp} we define the following matrix.

Definition III.3.1. Let $B_{\mathrm{sa}} = \{b_1^{\mathrm{sa}}, b_2^{\mathrm{sa}}, b_3^{\mathrm{sa}}, \dots,\}$ and $B_{\mathrm{sp}} = \{b_1^{\mathrm{sp}}, b_2^{\mathrm{sp}}, b_3^{\mathrm{sp}}, \dots,\}$ be orthonormal bases for $L^2([0,1))$. The *change of basis matrix* $U \in \mathcal{B}(\ell^2(\mathbb{N}))$ between B_{sa} and B_{sp} is the infinite matrix with entries

$$U_{ij} = \left\langle b_j^{\mathrm{sp}}, b_i^{\mathrm{sa}} \right\rangle$$

We will denote this matrix by $U = [B_{\mathrm{sa}}, B_{\mathrm{sp}}]$.

Notice in particular that since B_{sa} and B_{sp} are orthonormal, $U = [B_{\mathrm{sa}}, B_{\mathrm{sp}}]$ is an isometry on $\ell^2(\mathbb{N})$ and we can write $y = Ux$.

Next let $\Omega = \Omega_{\boldsymbol{m},\mathbf{N}}$ be a given multilevel random sampling scheme with $|\Omega| = m$. We refer to $N = N_r$ as the *sampling bandwidth* of Ω (as discussed later, this will be chosen in terms of sampling bandwidth to ensure stable truncation of U). Now define the matrix

$$H := \begin{bmatrix} 1/\sqrt{p_1} P_{\Omega_1} U \\ 1/\sqrt{p_2} P_{\Omega_2} U \\ \cdots \\ 1/\sqrt{p_r} P_{\Omega_r} U \end{bmatrix} \in \mathbb{C}^{m\times\infty}, \quad \text{where} \quad p_k = m_k/(N_k - N_{k-1}) \tag{III.6}$$

and we use the slightly unusual notation $\mathbb{C}^{m\times\infty}$ for the operators $\mathcal{B}(\ell^2(\mathbb{N}), \mathbb{C}^m)$. Due to the scaling factors $1/\sqrt{p_k}$ we consider scaled noisy measurements

$$\tilde{y} = DP_\Omega y + e \in \mathbb{C}^m \tag{III.7}$$

where D is a diagonal matrix with the corresponding scaling factors found in H along the diagonal and e is the measurement noise.

Suppose that x is approximately $(\boldsymbol{s}, \mathbf{M})$-sparse with sparsity bandwidth M. It is tempting to form the finite matrix $A = HP_M \in \mathbb{C}^{m\times M}$ and solve the minimization problem

$$\text{minimise } \|z\|_1 \quad \text{subject to} \quad \|Az - \tilde{y}\|_2 \leq \eta.$$

However, note that the truncation of H to A introduces an additional truncation error $HP_M^\perp x$. Indeed,

$$Ax - \tilde{y} = HP_M^\perp x + e,$$

and this poses a problem since for the above decoder we require $\eta \geq \|HP_M^\perp x + e\|_2$ in order for $P_M x$ to be a feasible point. For some applications we might have a rough estimate of $\|e\|_2$, but any estimate of $\|HP_M^\perp x\|_2$ would require a priori knowledge of x, the signal we are trying to recover. This is generally impossible. (We note in passing that there is some recent work [8] which derives CS recovery guarantees in the absence of feasibility of the target vector $P_M x$, but the application of this work to the sparse in levels model is not clear).

To overcome this issue, we will introduce a *data fidelity parameter* $K \geq M$ and assume we know $\|e\|_2$ so that we can let $\eta > \|e\|_2$. Then there will always exits a $K' \geq M$ such that $P_K x$ lies in the feasible set $\{z \in \mathbb{C}^K : \|Az - \tilde{y}\|_2 \leq \eta\}$ corresponding to the augmented matrix

$$A = HP_K \tag{III.8}$$

for all $K \geq K'$. In practice (for the general case) it will also be impossible determine a sufficient value for K, but for fixed $\eta > \|e\|_2$ there will always exist such a K. It should, however, be noted that there are special cases, such as Walsh sampling and wavelet recovery, where sufficient values for K are known; see Remark III.4.9.

This aside, as previously mentioned, we also now modify the optimization problem to include weights. Specifically, let $\mathbf{M}, \boldsymbol{s} \in \mathbb{N}^r$ be given sparsity levels and local sparsities respectively. For positive weights $\boldsymbol{\omega} = (\omega_1, \dots, \omega_{r+1})$ we define

$$\|x\|_{1,\boldsymbol{\omega}} \coloneqq \sum_{l=1}^{r+1} \omega_l \|P_{M_l}^{M_{l-1}} x\|_1,$$

with $M_{r+1} = K$ for $x \in \mathbb{C}^K$. Notice that this weighted regularizer assigns constant weights on each sparsity level. With this in hand, our recovery procedure is

$$\text{minimise} \, \|z\|_{1,\boldsymbol{\omega}} \quad \text{subject to} \quad \|Az - \tilde{y}\|_2 \leq \eta,$$

with A as in (III.8) and $\eta \geq \|Ax - \tilde{y}\|_2$.

III.3.2 The balancing property

We now discuss the relationship between the sampling and sparsity bandwidths N and M. From generalized sampling theory [2] we know that we must choose $N \geq M$ to obtain a stable mapping between the first N sampling basis functions and the first M sparsity basis functions. The degree of stability for this solution will depend on the so-called *balancing property*:

Definition III.3.2. Let $U \colon \ell^2(\mathbb{N}) \to \ell^2(\mathbb{N})$ be an isometry. Let $0 < \theta < 1$ and $N \geq M \geq 1$. Then U has the *balancing property* with constant θ if

$$\|P_M U^* P_N U P_M - P_M\|_2 \leq 1 - \theta.$$

Note that the balancing property may not hold for any $N \geq M$. However, it always holds for sufficiently large N (for fixed M). Indeed, $P_M U^* P_N U P_M \to P_M U^* U P_M \equiv P_M$ in the operator norm, hence the balancing property holds with θ arbitrarily close to 1 for large enough N.

Below we shall see that this property will also affect our recovery guarantees, but it will be camouflaged as the quantity $\|G^{-1}\|_2$, where $G = \sqrt{P_M U^* P_N U_P M}$. This gives the following relation.

Lemma III.3.3. *Let $U \in \mathcal{B}(\ell^2(\mathbb{N}))$ be an isometry satisfying the balancing property of order $0 < \theta < 1$ for $M, N \in \mathbb{N}$. Let $G = \sqrt{P_M U^* P_N U P_M}$ be self-adoint and nonnegative definite. Then G is invertible and*

$$\|G^{-1}\|_2 \leq 1/\sqrt{\theta} \tag{III.9}$$

III.3.3 G-adjusted Restricted Isometry Property in Levels (G-RIPL)

Our theoretical analysis requires a RIP-type property for the matrix HP_M. However, as implied in the previous discussion, the finite matrix $P_N U P_M \in \mathbb{C}^{N \times M}$ (from which AP_M is constructed) is not an isometry for any $N \geq M$. In particular, unlike in finite dimensions $\mathbb{E}(P_M H^* H P_M) = P_M U^* P_N U P_M = G^2$ is not the identity. In order to handle this situation, we introduce the following generalization of the RIP:

Definition III.3.4 (G-RIPL)**.** Let $A \in \mathbb{C}^{m \times M}$, $G \in \mathbb{C}^{M \times M}$ be invertible, $\mathbf{M} = (M_1, \ldots, M_r)$ be sparsity levels and $\mathbf{s} = (s_1, \ldots, s_r)$ be local sparsities. The $\mathbf{s}^{\text{th}}$ *G-adjusted Restricted Isometry Constant in Levels (G-RICL)* $\delta_{\mathbf{s},\mathbf{M}}$ is the smallest $\delta \geq 0$ such that

$$(1-\delta)\|Gx\|_2^2 \leq \|Ax\|_2^2 \leq (1+\delta)\|Gx\|_2^2, \quad \forall x \in \Sigma_{\mathbf{s},\mathbf{M}}.$$

If $0 < \delta_{\mathbf{s},\mathbf{M}} < 1$ we say that the matrix A satisfies the *G-adjusted Restricted Isometry Property in Levels (G-RIPL)* of order $(\mathbf{s}, \mathbf{M})$.

The G-RIPL is of course completely general and can be stated for any G. However, in the following we will let $G = \sqrt{P_M U^* P_N U P_M}$ and show that the matrix $A = HP_K$ (or equivalently, HP_M – note that $\Sigma_{\boldsymbol{s},\mathbf{M}}$ consists of vectors z with $P_M^{\perp} z = 0$) satisfies the G-RIPL for this particular G.

First, however, we show that the G-RIPL implies uniform recovery. For this, we introduce the following notation:

$$S_{\boldsymbol{\omega},\boldsymbol{s}} := \sum_{l=1}^{r} \omega_l^2 s_l \quad \text{and} \quad \zeta_{\boldsymbol{s},\boldsymbol{\omega}} = \min_{l \in \{1,\ldots,r\}} \omega_l^2 s_l.$$

Notice in particular that for the choice $\boldsymbol{\omega} = (1, \ldots, 1, \omega_{r+1})$ we have $S_{\boldsymbol{\omega},\boldsymbol{s}} = s_1 + \ldots + s_r$ and for the choice $\boldsymbol{\omega} = (s_1^{-1/2}, \ldots, s_r^{-1/2}, \omega_{r+1})$ we have $S_{\boldsymbol{\omega},\boldsymbol{s}} = r$. Finally, we let $\kappa(G) = \|G\|_2 \|G^{-1}\|_2$ denote the condition number of G.

Theorem III.3.5. *Let $A \in \mathbb{C}^{m \times K}$, $G \in \mathbb{C}^{M \times M}$ with $K \geq M$ and let $\mathbf{M}, \boldsymbol{s} \in \mathbb{N}^r$ be given sparsity levels and local sparsities, respectively. Let $\boldsymbol{\omega} \in \mathbb{R}^{r+1}$ be positive weights. Suppose AP_M satisfies the G-RIPL of order $(\boldsymbol{t}, \mathbf{M})$ with constant $\delta_{\boldsymbol{t},\mathbf{M}} \leq 1/2$ and*

$$t_l = \min\left\{ M_l - M_{l-1},\, 2\left\lceil \frac{4\kappa(G)^2 S_{\boldsymbol{\omega},\boldsymbol{s}}}{\omega_l^2} \right\rceil \right\} \quad \textit{for } l = 1,\dots,r. \tag{III.10}$$

Let

$$\omega_{r+1} \geq \sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\big(\tfrac{1}{3}(1+(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4})^{-1} + 2\sqrt{2}\|AP_K^M\|_{1\to 2}\|G^{-1}\|_2\big).$$

Let $\eta \geq 0$, $x \in \mathbb{C}^K$, $e \in \mathbb{C}^m$ with $\|e\|_2 \leq \eta$ and set $y = Ax + e$. Then any solution $\hat{x}$ of the optimization problem

$$\operatorname*{minimise}_{z\in\mathbb{C}^K} \|z\|_{1,\boldsymbol{\omega}} \quad \textit{subject to} \quad \|Az - y\|_2 \leq \eta \tag{III.11}$$

satisfies

$$\|x - \hat{x}\|_{1,\boldsymbol{\omega}} \leq C\sigma_{\boldsymbol{s},\mathbf{M}}(x)_{1,\boldsymbol{\omega}} + D\|G^{-1}\|_2\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\eta \tag{III.12}$$

$$\|x - \hat{x}\|_2 \leq (1+(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4})\left(C\frac{\sigma_{\boldsymbol{s},\mathbf{M}}(x)_{1,\boldsymbol{\omega}}}{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}} + D\|G^{-1}\|_2\eta \right) \tag{III.13}$$

where $C = 2(2+\sqrt{3})/(2-\sqrt{3})$, $D = 8\sqrt{2}/(2-\sqrt{3})$ and $\sigma_{\boldsymbol{s},\mathbf{M}}(x)_{1,\boldsymbol{\omega}} = \inf\{\|x-z\|_{1,\boldsymbol{\omega}} : z \in \Sigma_{\boldsymbol{s},\mathbf{M}}\}$.

Notice that the condition on δ in the above theorem is fundamentally different from the condition found in Theorem III.2.8. In the latter one requires $\delta_{2\boldsymbol{s},\mathbf{M}} < (r(\sqrt{\alpha_{\boldsymbol{s},\mathbf{M}}}+\frac{1}{4})^2+1)^{-1/2}$ where $\alpha_{\boldsymbol{s},\mathbf{M}} = \max_{k,l=1,\dots,r} s_k/s_l$ is the sparsity ratio. Thus for sparsity levels where the local sparsities vary greatly, this bound will be unreasonably small.

In the above theorem we have removed this sparsity ratio term, by setting $\delta = 1/2$, and require $\delta_{\boldsymbol{t},\mathbf{M}} \leq \delta$ where $t_l \geq 2\left\lceil 4\kappa(G)S_{\boldsymbol{\omega},\boldsymbol{s}}w_l^{-2}\right\rceil$. For the unweighted case this leads to a condition of the form

$$t_l \geq 2\left\lceil 4\kappa(G)^2(s_1 + \ldots + s_r)\right\rceil$$

which could be difficult to fulfill in practice, since each t_l would have to be greater than the total sparsity of the signal. However, by considering the weights $\boldsymbol{\omega} = (s_1^{-1/2}, \dots, s_r^{-1/2}, \omega_{r+1})$ we obtain a condition of the form

$$t_l \geq 2\left\lceil 4\kappa(G)^2 r s_l\right\rceil,$$

where t_l is independent of s_k for $k \neq l$. This means that we can write the requirement as $\delta_{2\lceil 4\kappa(G)^2 r\boldsymbol{s}\rceil,\mathbf{M}} \leq 1/2$, and ignore any dependence between the $\boldsymbol{s}$-values, as was the problem in Theorem III.2.8.

III.3.4 Sufficient condition for the G-RIPL

In Definition III.2.9 we defined the local coherence $\mu_{k,l}$ of an isometry $U \in \mathbb{C}^{N\times N}$. We extend this to isometries $U \in \mathcal{B}(\ell^2(\mathbb{N}))$ in the exact same way

$$\mu_{k,l} = \mu_{k,l}(\mathbf{N},\mathbf{M}) = \{\max|U_{ij}|^2 : i = N_{k-1}+1,\dots,N_k, j = M_{l-1}+1,\dots,M_l\}.$$

This yields the following theorem.

Theorem III.3.6 (Subsampled isometries and the G-RIPL)**.** *Let $U \in \mathcal{B}(\ell^2(\mathbb{N}))$ be an isometry, and let $\Omega = \Omega_{\mathbf{N},\mathbf{m}}$ be an $(\mathbf{N},\mathbf{m})$-multilevel sampling scheme with r levels. Let $\mathbf{M}, \boldsymbol{s} \in \mathbb{N}^r$ be sparsity levels and local sparsities, respectively. Let $\epsilon, \delta \in (0,1)$ and let $0 \leq r_0 \leq r$, with $\tilde{m} = m_{r_0+1} + \cdots + m_r$. Let $s = s_1 + \cdots + s_r$ and $L = r \cdot \log(2\tilde{m}) \cdot \log(2N) \cdot \log^2(2s) + \log(\epsilon^{-1})$. Suppose $G = \sqrt{P_M U^* P_N U P_M}$ is non-singular. If*

$$m_k = N_k - N_{k-1}, \quad k = 1, \ldots, r_0, \tag{III.14}$$

and

$$m_k \gtrsim \delta^{-2} \cdot \|G^{-1}\|_2^2 \cdot (N_k - N_{k-1}) \cdot \left(\sum_{l=1}^{r} \mu_{k,l} \cdot s_l \right) \cdot L, \tag{III.15}$$

for $k = r_0 + 1, \ldots, r$ then with probability at least $1 - \epsilon$, the matrix

$$A = \begin{bmatrix} 1/\sqrt{p_1} P_{\Omega_1} U P_M \\ \vdots \\ 1/\sqrt{p_r} P_{\Omega_r} U P_M \end{bmatrix} \quad \text{where } p_k = \frac{m_k}{N_k - N_{k-1}} \quad \text{for } k = 1, \ldots, r \tag{III.16}$$

satisfies the G-RIPL of order $(\mathbf{s}, \mathbf{M})$ with constant $\delta_{\mathbf{s},\mathbf{M}} \leq \delta$.

III.3.5 Overall recovery guarantee

Theorem III.3.5 and Theorem III.3.6 yield the next results.

Corollary III.3.7. *Let $U \in \mathcal{B}(\ell^2(\mathbb{N}))$ be an isometry, and let $\Omega = \Omega_{\mathbf{N},\mathbf{m}}$ be an $(\mathbf{N},\mathbf{m})$-multilevel sampling scheme with r levels. Let $\mathbf{M}, \boldsymbol{s} \in \mathbb{N}^r$ be sparsity levels and local sparsities, respectively, and let $\boldsymbol{\omega} = [s_1^{-1/2}, \ldots, s_r^{-1/2}, \omega_{r+1}]$ be weights. Let $\epsilon, \delta \in (0,1)$ and $0 \leq r_0 \leq r$. Let $m = m_1 + \ldots + m_r$, $\tilde{m} = m_{r_0+1} + \cdots + m_r$, $s = s_1 + \cdots + s_r$, and $L = r \cdot \log(2\tilde{m}) \cdot \log(2N) \cdot \log^2(2s) + \log(\epsilon^{-1})$. Let $H \in \mathbb{C}^{m \times \infty}$ be as in (III.6) and set $A = HP_K$. Let $x \in \ell^2(\mathbb{N})$, $e_1 \in \mathbb{C}^m$ and $\eta > 0$. Set $e = HP_K^{\perp} x + e_1$ and $\tilde{y} = Ax + e$. Suppose*

(i) *we choose M and N so that U satisfies the balancing property of order $0 < \theta < 1$,*

(ii) *we choose $\eta \geq \|e_1\|_w$ and K so that $\|HP_K^{\perp} x\|_2 \leq \eta'$,*

(iii) *the weight ω_{r+1} satisfies*

$$\omega_{r+1} \geq \sqrt{r} \left(\frac{1}{3(1 + r^{1/4})} + 2\sqrt{\frac{2}{\theta}} \|AP_K^M\|_{1 \to 2} \right),$$

(iv) *the m_k's satisfy $m_k = N_k - N_{k-1}$ for $k = 1, \ldots, r_0$ and*

$$m_k \gtrsim \theta^{-2} \cdot r \cdot (N_k - N_{k-1}) \cdot \left(\sum_{l=1}^{r} \mu_{k,l} s_l \right) \cdot L \quad \text{for } k = r_0 + 1, \ldots, r. \tag{III.17}$$

Then with probability $1-\epsilon$ *any solution* $\hat{x}$ *of the optimization problem*

$$\operatorname*{minimise}_{z\in\mathbb{C}^K} \|z\|_{1,\boldsymbol{\omega}} \quad \textit{subject to} \quad \|Az-\tilde{y}\|_2 \leq \eta+\eta'$$

satisfies

$$\|P_K x-\hat{x}\|_{1,\boldsymbol{\omega}} \leq C\sigma_{\boldsymbol{s},\mathbf{M}}(P_K x)_{1,\boldsymbol{\omega}} + D\frac{\sqrt{r}}{\sqrt{\theta}}(\eta+\eta') \tag{III.18}$$

$$\|P_K x-\hat{x}\|_2 \leq (1+r^{1/4})\left(C\frac{\sigma_{\boldsymbol{s},\mathbf{M}}(P_K x)_{1,\boldsymbol{\omega}}}{\sqrt{r}} + D\frac{1}{\sqrt{\theta}}(\eta+\eta')\right) \tag{III.19}$$

where $C=2(2+\sqrt{3})/(2-\sqrt{3})$ *and* $D=8\sqrt{2}/(2-\sqrt{3})$.

Suppose that x is exactly $(\boldsymbol{s},\mathbf{M})$-sparse. Then the above theorem guarantees exact recovery of x via weighted ℓ^1 minimization subject to the corresponding measurement condition. We note in passing this measurement condition is optimal up to log factors, in the sense that it is the same of that of the oracle estimator based on *a priori* knowledge of $\operatorname{supp}(x)$. See [1].

III.4 Recovery guarantees for Walsh sampling with wavelet reconstruction

Having presented the abstract infinite-dimensional CS framework in full generality, the remainder of the paper is devoted to its application to the case of binary sampling with the Walsh transform with sparsity in orthogonal wavelet bases. We first describe the setup, before presenting the main recovery guarantees in Sections III.4.3 and III.4.4.

III.4.1 Walsh functions

For any number $n\in\mathbb{Z}_+=\{0,1,2,\ldots\}$ there exits a unique dyadic expansion

$$n=n_1 2^0+n_2 2^1+\ldots+n_j 2^{j-1}+\cdots$$

where $n_j\in\{0,1\}$ for $j\in\mathbb{N}$. Similarly any $x\in[0,1)$ can be written in its dyadic form as

$$x=x_1 2^{-1}+x_2 2^{-2}+\cdots+x_j 2^{-j}$$

with $x_j\in\{0,1\}$ for all $j\in\mathbb{N}$. For a dyadic rational number x this expansion is not unique, as one may use either a finite expansion, or an infinite expansion where $x_i=1$ for all $i\geq k$ for some $k\in\mathbb{N}$. In such cases we always consider the finite expansion. In practice this means that we have removed countably many singletons from $[0,1)$.

Definition III.4.1. Let $n\in\mathbb{Z}_+$ and $x\in[0,1)$. The *Walsh function* $w_n\colon[0,1)\to\{+1,-1\}$ is given by

$$w_n(x) \coloneqq (-1)^{\sum_{j=1}^{\infty}(n_j+n_{j+1})x_j} \tag{III.20}$$

On the interval $[0, 1)$ the Walsh function w_n has n sign changes, n is therefore often denoted the *frequency* of w_n. The 2^r first Walsh functions gives rise to the entries in the sequency ordered Hadamard matrix

$$(V_{\text{Had}})_{i,j} = w_{i-1}((j-1)/2^r)$$

where $i, j = 1, \dots, 2^r$.

Definition III.4.2 (Walsh basis)**.** Define the *Walsh basis* as

$$B_{\text{wh}} \coloneqq \{w_n : n \in \mathbb{Z}_+\}$$

where 'wh' is an abbreviation for Walsh-Hadamard.

Note that this is an orthonormal basis of $L^2([0, 1))$.

III.4.2 Wavelet transform

Let $\phi\colon \mathbb{R} \to \mathbb{R}$ and $\psi\colon \mathbb{R} \to \mathbb{R}$ be a orthonormal scaling function and wavelet [13], respectively, with minimal support, corresponding to an multiresolution analysis (MRA). Note that this could both be the classical 'Daubechies wavelet' with a minimum-phase or 'symlets' which are close to being symmetric, but with a larger phase [26, p. 294]. Let

$$\phi_{j,k}(x) \coloneqq 2^{j/2}\phi(2^j x - k) \quad \text{and} \quad \psi_{j,k}(x) \coloneqq 2^{j/2}\psi(2^j x - k) \tag{III.21}$$

denote the scaled and translated versions.

A wavelet ψ is said to have ν vanishing moments if

$$\int_{-\infty}^{\infty} x^k \psi(x)\,\mathrm{d}x = 0 \qquad \text{for } 0 \le k < \nu.$$

For for orthogonal wavelets with minimum support, the support depends on the number of vanishing moments. That is

$$\operatorname{supp}(\phi) = \operatorname{supp}(\psi) = [-\nu + 1, \nu]. \tag{III.22}$$

While this system constitutes an orthonormal basis of $L^2(\mathbb{R})$, in our case we require an orthonormal basis of $L^2([0, 1))$. There exists several construction of wavelets on the interval, but we will only consider periodic extensions and the orthogonal *boundary wavelets* introduced by Cohen, Daubechies and Vial in [12], which preserves the number of vanishing moments.

For wavelets on the interval, we need to replace the 2ν wavelets/scaling functions intersecting the boundaries at each scale, with their corresponding boundary-corrected counterparts. We postpone the formal definition of periodic and boundary wavelets until we need it, in the proof sections. But to simplify the notation let

$$\phi^0_{j,k} := \begin{cases} \phi^{\text{boundary}}_{j,k} & \text{for } k \in \{0,\dots,\nu-1\} \\ \phi_{j,k} & \text{for } k \in \{\nu,\dots,2^j-\nu-1\} \\ \phi^{\text{boundary}}_{j,k} & \text{for } k \in \{2^j-\nu,\dots,2^j-1\} \end{cases},$$

$$\phi^1_{j,k} := \begin{cases} \psi^{\text{boundary}}_{j,k} & \text{for } k \in \{0,\dots,\nu-1\} \\ \psi_{j,k} & \text{for } k \in \{\nu,\dots,2^j-\nu-1\} \\ \psi^{\text{boundary}}_{j,k} & \text{for } k \in \{2^j-\nu,\dots,2^j-1\} \end{cases},$$

where $\phi^{\text{boundary}}_{j,k}$ and $\psi^{\text{boundary}}_{j,k}$ are either a periodic wavelet/scaling function or the boundary wavelet/scaling functions introduced in [12]. For the former extension we say that $\phi^s_{j,k}$, $s \in \{0,1\}$ 'originate from a *periodic wavelet*' while for the latter we say that it 'originate from a *boundary wavelet*'.

We will throughout assume $J_0 \in \mathbb{Z}_+$ satisfies $2^{J_0} \geq 2\nu$ for $\nu \geq 2$ and $J_0 \geq 0$ for $\nu = 1$. This will ensure that there exits at least one $k \in \{0,\dots,2^j-1\}$ such that $\operatorname{supp}(\phi_{j,k}) = \operatorname{supp}(\psi_{j,k}) \subseteq [0,1)$ for all $j \geq J_0$.

Definition III.4.3. For a fixed number of vanishing moments ν, minimum wavelet decomposition J_0 and a boundary extension which is either periodic or boundary wavelets, let $\phi^s_{j,k}$ be the corresponding wavelets and scaling functions. We define

$$B^{J_0,\nu}_{\text{wave}} = \left\{\phi^0_{J_0,0},\dots,\phi^0_{J_0,2^{J_0}-1},\phi^1_{J_0,0},\dots,\phi^1_{J_0,2^{J_0}-1},\phi^1_{J_0+1,0},\dots,\phi^1_{J_0+1,2^{J_0+1}-1},\dots\right\}$$

Both B_{wh} and $B^{J_0,\nu}_{\text{wave}}$ are orthonormal bases for $L^2([0,1))$.

III.4.3 Recovery guarantees

From Section III.3 there are four unknown factors depending on U which need to be estimated. These are the local coherences $\mu_{k,l}$, the norm $\|HP^M_K\|_{1\to 2}$ where H is given by (III.6), the condition number $\kappa(G) = \|G\|_2\|G^{-1}\|_2$ and the factor $\|G^{-1}\|_2$ found in condition (III.15).

For the two latter factors we have $G = \sqrt{P_M U^* P_N U P_M}$. Furthermore we know that $\|G\|_2 \leq 1$ since U is an isometry. In practice we therefore only need to determine an upper bound $\|G^{-1}\|_2$ and from Lemma III.3.3 we know that $\|G^{-1}\|_2 \leq 1/\sqrt{\theta}$, where $0 < \theta < 1$ is the balancing property constant. In other words, it suffices to determine when the balancing property holds with a given θ.

The following three propositions estimate these quantities for the case $U = [B_{\text{wh}}, B^{J_0,\nu}_{\text{wave}}]$.

Proposition III.4.4. *Let $U = [B_{\text{wh}}, B^{J_0,\nu}_{\text{wave}}]$. For each $\theta \in (0,1)$, there exits a constant $q_\theta \geq 0$, such that whenever $N = 2^{k+q_\theta} \geq 2^k = M$ then U satisfies the balancing property of order θ for all $k \in \mathbb{N}$.*

Note that Proposition III.4.4 is a consequence of Theorem 1.1 in [20].

Proposition III.4.5. *Let $U = [B_{\text{wh}}, B^{J_0,\nu}_{\text{wave}}]$ with $\nu \geq 3$ and let*

$$\mathbf{M} = [2^{J_0+1},\dots,2^{J_0+r}] \textit{ and } \mathbf{N} = [2^{J_0+1},\dots,2^{J_0+r-1},2^{J_0+r+q}] \textit{ with } q \geq 0,$$

be sparsity and sampling levels, respectively. Then the local coherences of U scales like

$$\mu_{k,l} \lesssim 2^{-J_0-k}2^{-|l-k|}.$$

Proposition III.4.6. *Let $U = [B_{\mathrm{wh}}, B_{\mathrm{wave}}^{J_0,\nu}]$ and let $\mathbf{M}, \mathbf{N} \in \mathbb{N}^r$ be sparsity and sampling levels. Let $\Omega = \Omega_{\boldsymbol{m},\mathbb{N}}$ be a multilevel random sampling scheme, and let H be as in (III.6). Then*

$$\|HP_K^{\perp}\|_{1\to 2} \lesssim \sqrt{\frac{N}{K}}.$$

We can now present the two main theorems in this section. We point out that these are only valid for $\nu \geq 3$ vanishing moments. For $\nu = 1$, the corresponding wavelet is the Haar wavelet and will be considered in the next subsection. For $\nu = 2$, the coherence of $U = [B_{\mathrm{wh}}, B_{\mathrm{wave}}^{J_0,2}]$ does not decay as fast as for the other wavelets. Whether this is because our coherence bounds are not sharp enough for this wavelet or if it is because the coherence of $U = [B_{\mathrm{wh}}, B_{\mathrm{wave}}^{J_0,2}]$ actually decays more slowly is not known. We do, however, present some numerics in Section III.6.5 which indicate that it is potentially the latter.

Theorem III.4.7. *Let $U = [B_{\mathrm{wh}}, B_{\mathrm{wave}}^{J_0,\nu}]$ with $\nu \geq 3$ and let*

$$\mathbf{M} = [2^{J_0+1}, \ldots, 2^{J_0+r}] \text{ and } \mathbf{N} = [2^{J_0+1}, \ldots, 2^{J_0+r-1}, 2^{J_0+r+q}] \text{ with } q \geq 0,$$

be sparsity and sampling levels, respectively. Let $\boldsymbol{s} \in \mathbb{N}^r$ be local sparsities. Suppose q is chosen so that U satisfies the balancing property with constant $0 < \theta < 1$ and set $G = \sqrt{P_M U^* P_N U P_M}$. Let $\epsilon, \delta \in (0,1)$ and let $0 \leq r_0 \leq r$, with $\tilde{m} = m_{r_0+1} + \cdots + m_r$. Let $s = s_1 + \cdots + s_r$ and $L = r \cdot \log(2\tilde{m}) \cdot \log(2N) \cdot \log^2(2s) + \log(\epsilon^{-1})$. If

$$m_k = N_k - N_{k-1}, \quad k = 1, \ldots, r_0, \tag{III.23}$$

and

$$m_k \gtrsim \delta^{-2} \cdot \theta^{-1} \cdot 2^{q\max\{k+1-r,0\}} \cdot \left(\sum_{l=1}^{r} 2^{-|k-l|} s_l\right) \cdot L$$

for $k = r_0+1, \ldots, r$, then with probability at least $1-\epsilon$, the matrix in (III.16) satisfies the G-RIPL of order $(\mathbf{s}, \mathbf{M})$ with constant $\delta_{\mathbf{s},\mathbf{M}} \leq \delta$.

With this in hand, we now present our main result:

Theorem III.4.8. *Let $U = [B_{\mathrm{wh}}, B_{\mathrm{wave}}^{J_0,\nu}]$ with $\nu \geq 3$ and let*

$$\mathbf{M} = [2^{J_0+1}, \ldots, 2^{J_0+r}] \text{ and } \mathbf{N} = [2^{J_0+1}, \ldots, 2^{J_0+r-1}, 2^{J_0+r+q}], \text{ with } q \geq 0$$

be sparsity and sampling levels, respectively. Let $\boldsymbol{s} \in \mathbb{N}^r$ be local sparsities, $\boldsymbol{\omega} = (s_1^{-1/2}, \ldots, s_r^{-1/2}, \omega_{r+1})$ be weights and let $\boldsymbol{m} \in \mathbb{N}^r$ be sampling densities. Let $\epsilon \in (0,1)$ and let $0 \leq r_0 \leq r$. Let $m = m_1 + \ldots + m_r$, $\tilde{m} = m_{r_0+1} + \cdots + m_r$, $s = s_1 + \ldots + s_r$, and $L = r \cdot \log(2\tilde{m}) \cdot \log(2N) \cdot \log^2(2s) + \log(\epsilon^{-1})$.

Let $H \in \mathbb{C}^{m\times\infty}$ be as in (III.6) and set $A = HP_K$. Let $x \in \ell^2(\mathbb{N})$, $e_1 \in \mathbb{C}^m$ and $\eta > 0$. Set $e = HP_K^{\perp}x + e_1$ and $\tilde{y} = Ax + e$. Suppose

(i) *we choose $q = q_\theta$ as in Proposition III.4.4 so that U satisfies the balancing property of order $0 < \theta < 1$,*

(ii) *we choose $\eta \geq \|e_1\|$ and K so that $\|HP_K^\perp x\|_2 \leq \eta'$,*

(iii) *the weight ω_{r+1} satisfies*

$$\omega_{r+1} \geq \sqrt{r}\left(\frac{1}{3(1+r^{1/4})} + 2\sqrt{\frac{2}{\theta}}\|AP_K^M\|_{1\to 2}\right),$$

(iv) *the m_k's satisfy $m_k = N_k - N_{k-1}$ for $k = 1, \ldots, r_0$ and*

$$m_k \gtrsim \theta^{-2} \cdot r \cdot 2^{q\max\{k+1-r,0\}}\left(\sum_{l=1}^{r} 2^{-|k-l|}s_l\right) \cdot L \quad \text{for } k = r_0+1, \ldots, r. \tag{III.24}$$

Then with probability $1-\epsilon$ any solution $\hat{x}$ of the optimization problem

$$\operatorname*{minimise}_{z\in\mathbb{C}^K} \|z\|_{1,\boldsymbol{\omega}} \quad \text{subject to} \quad \|Az - \tilde{y}\|_2 \leq \eta + \eta'$$

satisfies

$$\|P_K x - \hat{x}\|_{1,\boldsymbol{\omega}} \leq C\sigma_{\boldsymbol{s},\mathbf{M}}(P_K x)_{1,\boldsymbol{\omega}} + D\frac{\sqrt{r}}{\sqrt{\theta}}(\eta + \eta') \tag{III.25}$$

$$\|P_K x - \hat{x}\|_2 \leq (1 + r^{1/4})\left(C\frac{\sigma_{\boldsymbol{s},\mathbf{M}}(P_K x)_{1,\boldsymbol{\omega}}}{\sqrt{r}} + D\frac{1}{\sqrt{\theta}}(\eta + \eta')\right) \tag{III.26}$$

where $C = 2(2+\sqrt{3})/(2-\sqrt{3})$ and $D = 8\sqrt{2}/(2-\sqrt{3})$.

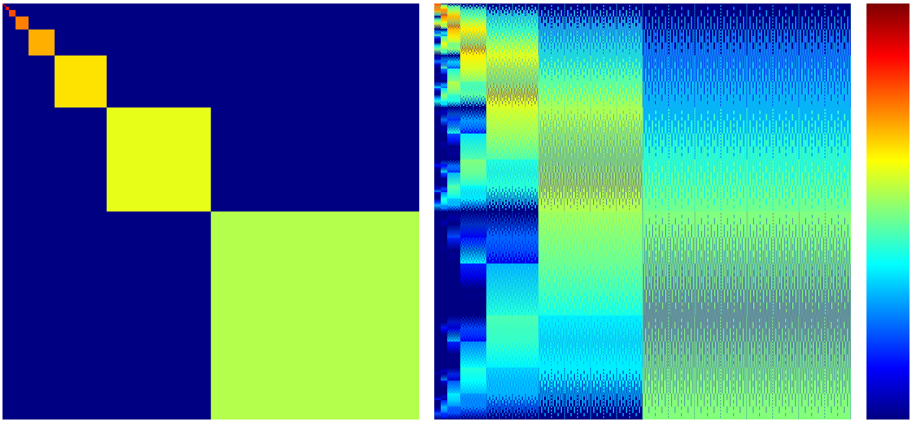
Remark III.4.9. Note that the second condition (ii) can be guaranteed using Proposition III.4.6. Indeed, it suffices for K to satisfy

$$\frac{\|P_K^\perp x\|_1}{\sqrt{K}} \lesssim \frac{\eta'}{\sqrt{N}}.$$

Hence, given any *a priori* estimates on the decay of the coefficients x (such as in the case of wavelets), one can use this to determine a suitable K.

III.4.4 Uniform recovery for Haar wavelets

Below we shall see that for the Haar wavelet, $P_N U P_N$ will be an isometry for $N = 2^r$ where $r \in \mathbb{N}$. This can also be seen from Figure III.2, where $U = [B_{\text{wh}}, B_{\text{wave}}^{J_0,\nu}]$ is perfectly block diagonal for $\nu = 1$. This means that the G-RIPL, reduces to the I-adjusted RIPL, or simply the RIPL, which we know from the finite dimensional case. Notice in particular that we also avoid any considerations where $K > M = N$ as above, since $HP_M^\perp = 0$.



Haar (DB1)

DB4

Figure III.2: The absolute values in log scale of the matrix $P_M U P_M$ for $U = [B_{\text{wh}}, B_{\text{wave}}^{J_0,\nu}]$, with $\nu = 1$ (left) and $\nu = 4$ (middle). The rightmost image is the colorbar.

Proposition III.4.10. *Let $U = [B_{\text{wh}}, B_{\text{wave}}^{J_0,1}]$ and let $N = 2^k$, for some $k \in \mathbb{N}$ with $k \geq J_0 + 1$. Then $P_N U P_N$ is an isometry on $\mathbb{C}^N$.*

Proposition III.4.11. *Let $U = [B_{\text{wh}}, B_{\text{wave}}^{J_0,1}]$ and let $\mathbf{M} = \mathbf{N} = [2^{J_0+1}, \ldots, 2^{J_0+r}]$ be sparsity and sampling levels, respectively. Then the local coherences of U are*

$$\mu_{kl} = \begin{cases} 2^{-J_0-k+1} & \text{if } k = l \\ 0 & \text{if } k \neq l \end{cases}$$

It is now straightforward to derive the following:

Theorem III.4.12. *Let $U = [B_{\text{wh}}, B_{\text{wave}}^{J_0,1}]$ and let $\mathbf{M} = \mathbf{N} = [2^{J_0+1}, \ldots, 2^{J_0+r}]$ be sparsity and sampling levels. Let $s \in \mathbb{N}^r$ be local sparsities and $\boldsymbol{m} \in \mathbb{N}^r$ be local sampling densities. Let $\epsilon, \delta \in (0,1)$ and $0 \leq r_0 \leq r$. Let $\tilde{m} = m_{r_0+1} + \ldots + m_r$ and $s = s_1 + \ldots + s_r$. Suppose that the m_k's satisfies $m_k = N_k - N_{k-1}$ for $k = 1, \ldots, r_0$ and*

$$m_k \gtrsim \delta^{-2} s_k \left(r \log(2\tilde{m}) \log(2N) \log^2(2s) + \log(\epsilon^{-1}) \right), \quad \text{for } k = r_0 + 1, \ldots, r. \tag{III.27}$$

Then with probability $1 - \epsilon$ the matrix (III.16) satisfies the RIPL with constant $\delta_{\boldsymbol{s},\mathbf{M}} \leq \delta$.

Proof. Using Proposition III.4.10 we know that $P_N U P_N$ is an isometry. Thus inserting the local coherences from Proposition III.4.11 into (III.4) in Theorem III.2.10 gives to the result. ■

Theorem III.4.13. *Let $U = [B_{\mathrm{wh}}, B_{\mathrm{wave}}^{J_0,1}]$ and let $\mathbf{M} = \mathbf{N} = [2^{J_0+1}, \dots, 2^{J_0+r}]$ be sparsity and sampling levels. Let $\boldsymbol{s} \in \mathbb{N}^r$ be local sparsities, $\boldsymbol{\omega} = (s_1^{1/2}, \dots, s_r^{1/2})$ be weights and $\boldsymbol{m} \in \mathbb{N}^r$ be local sampling densities. Let $\epsilon \in (0,1)$ and let $0 \leq r_0 \leq r$. Let $m = m_1 + \ldots + m_r$, $\tilde{m} = m_{r_0+1} + \cdots + m_r$ and $s = s_1 + \ldots + s_r$. Suppose we sample $m_k = N_k - N_{k-1}$ for $k = 1, \dots, r_0$ and*

$$m_k \gtrsim r \cdot s_k \cdot \left(r \log(2\tilde{m}) \log(2N) \log^2(2s) + \log(\epsilon^{-1})\right),$$

for $k = r_0 + 1, \dots, r$. Let $H \in \mathbb{C}^{m \times \infty}$ be as in (III.6) with $A = HP_M$. Let $x \in \ell^2(\mathbb{N})$ and $e \in \mathbb{C}^m$ with $\|e\|_2 \leq \eta$ for some $\eta \geq 0$. Set $\tilde{y} = Ax + e$. Then any solution $\hat{x}$ of the optimization problem

$$\underset{z \in \mathbb{C}^M}{\operatorname{minimise}} \|z\|_{1,\boldsymbol{\omega}} \quad \textit{subject to} \quad \|Az - \tilde{y}\|_2 \leq \eta$$

satisfies

$$\begin{aligned} \|P_M x - \hat{x}\|_{1,\boldsymbol{\omega}} &\leq C \sigma_{\boldsymbol{s},\mathbf{M}}(P_M x)_{1,\boldsymbol{\omega}} + D\sqrt{r}\eta \\ \|P_M x - \hat{x}\|_2 &\leq (1 + r^{1/4}) \left(C \frac{\sigma_{\boldsymbol{s},\mathbf{M}}(P_M x)_{1,\boldsymbol{\omega}}}{\sqrt{r}} + D\eta \right) \end{aligned}$$

with probability $1 - \epsilon$, where $C = 2(2 + \sqrt{3})/(2 - \sqrt{3})$ and $D = 8\sqrt{2}/(2 - \sqrt{3})$.

Proof. Proposition III.4.10 gives $G = \sqrt{P_M U^* P_N U P_M} = \sqrt{I} = I$. Next notice that $S_{\boldsymbol{\omega},\boldsymbol{s}} = r$ and that $P_M x \in \{z \in \mathbb{C}^M : \|Az - \tilde{y}\|_2 \leq \eta\}$ since $\|HP_M^{\perp}\| = 0$. Using Theorem III.3.5 we see that we can guarantee recovery of $(\boldsymbol{s}, \mathbf{M})$-sparse vectors, if A satisfies the RIPL with constant $\delta_{\boldsymbol{t},\mathbf{M}} \leq 1/2$, where $t_l = \min\{M_l - M_{l-1}, 8rs_l\}$. Using Theorem III.4.12 gives the result. ■

III.5 Proofs of results in Section III.3

When deriving uniform recovery guarantees via the RIP, it is typical to proceed as follows. First, one shows that the RIP implies the so-called *robust Null space Property (rNSP)* of order s (see Def. 4.17 in [16]). Second, one shows that the rNSP implies stable and robust recovery. Thus the line of implications reads

$$(\text{RIP}) \implies (\text{rNSP}) \implies (\text{uniform recovery}).$$

A similar line of implications holds for the RIPL and the corresponding *robust Null Space Property in levels (rNSPL)*; see Def. 3.6 in [7]).

Both of the recovery guarantees for matrices satisfying the rNSP and rNSPL consider minimizers of the unweighed quadratically-constrained basis pursuit (QCBP) optimization problem. In our setup, we consider minimizers of the weighted QCBP. We have therefore generalized the rNSPL to what we call the weighted robust null space property in levels.

For the sufficient condition for the G-RIPL in Theorem III.3.6, the proof follows along similar lines as in [23]. We only sketch the main differences here.

III.5.1 The weighted rNSPL and norm bounds

For a set $\Theta \subseteq \{1,\dots,M\}$ and a vector $x \in \mathbb{C}^M$ we let the vector x_Θ be given by

$$(x_\Theta)_i = \begin{cases} x_i & i \in \Theta \\ 0 & i \notin \Theta \end{cases}.$$

We also define

$$E_{\boldsymbol{s},\mathbf{M}} = \{\Theta \subseteq \{1,\dots,M\} : |\Theta \cap \{M_{l-1}+1,\dots,M_l\}| \le s_l, \text{ for } l = 1,\dots,r\}.$$

Definition III.5.1 (weigthed rNSP in levels)**.** Let $\mathbf{M}, \boldsymbol{s} \in \mathbb{N}^r$ be sparsity levels and local sparsities, respectively. For positive weights $\boldsymbol{\omega} \in \mathbb{R}^{r+1}$, we say that $A \in \mathbb{C}^{m\times M}$ satisfies the *weighted robust Null Space Property in Levels* (weighted rNSPL) of order $(\boldsymbol{s},\mathbf{M})$ with constants $0 < \rho < 1$ and $\gamma > 0$ if

$$\|x_\Theta\|_2 \le \frac{\rho\|x_{\Theta^c}\|_{1,\boldsymbol{\omega}}}{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}} + \gamma\|Ax\|_2 \tag{III.28}$$

for all $x \in \mathbb{C}^M$ and all $\Theta \in E_{\boldsymbol{s},\mathbf{M}}$.

Lemma III.5.2 (weighted rNSPL implies $\ell^{(1,\boldsymbol{\omega})}$-distance bound)**.** *Suppose that $A \in \mathbb{C}^{m\times M}$ satisfies the weighted rNSPL of order $(\boldsymbol{s},\mathbf{M})$ with constants $0 < \rho < 1$ and $\gamma > 0$. Let $x, z \in \mathbb{C}^M$. Then*

$$\|z - x\|_{1,\boldsymbol{\omega}} \le \frac{1+\rho}{1-\rho}\left(2\sigma_{\boldsymbol{s},\mathbf{M}}(x)_{1,\boldsymbol{\omega}} + \|z\|_{1,\boldsymbol{\omega}} - \|x\|_{1,\boldsymbol{\omega}}\right) + \frac{2\gamma}{1-\rho}\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\|A(z-x)\|_2. \tag{III.29}$$

Proof. Let $v = z - x$ and $\Theta \in E_{\boldsymbol{s},\mathbf{M}}$ be such that $\|x_{\Theta^c}\|_{1,\boldsymbol{\omega}} = \sigma_{\boldsymbol{s},\mathbf{M}}(x)_{1,\boldsymbol{\omega}}$. Then

$$\begin{aligned}\|x\|_{1,\boldsymbol{\omega}} + \|v_{\Theta^c}\|_{1,\boldsymbol{\omega}} &\le 2\|x_{\Theta^c}\|_{1,\boldsymbol{\omega}} + \|x_\Theta\|_{1,\boldsymbol{\omega}} + \|z_{\Theta^c}\|_{1,\boldsymbol{\omega}} \\ &= 2\|x_{\Theta^c}\|_{1,\boldsymbol{\omega}} + \|x_\Theta\|_{1,\boldsymbol{\omega}} + \|z\|_{1,\boldsymbol{\omega}} - \|z_\Theta\|_{1,\boldsymbol{\omega}} \\ &\le 2\sigma_{\mathbf{s},\mathbf{M}}(x)_{1,\boldsymbol{\omega}} + \|v_\Theta\|_{1,\boldsymbol{\omega}} + \|z\|_{1,\boldsymbol{\omega}},\end{aligned}$$

which implies that

$$\|v_{\Theta^c}\|_{1,\boldsymbol{\omega}} \le 2\sigma_{\mathbf{s},\mathbf{M}}(x)_{1,\boldsymbol{\omega}} + \|z\|_{1,\boldsymbol{\omega}} - \|x\|_{1,\boldsymbol{\omega}} + \|v_\Theta\|_{1,\boldsymbol{\omega}}. \tag{III.30}$$

Now consider $\|v_\Theta\|_{1,\boldsymbol{\omega}}$. By the weighted rNSPL, we have

$$\|v_\Theta\|_{1,\boldsymbol{\omega}} \le \sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\|v_\Theta\|_2 \le \rho\|v_{\Theta^c}\|_{1,\boldsymbol{\omega}} + \sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\gamma\|Av\|_2.$$

Hence (III.30) gives

$$\|v_\Theta\|_{1,\boldsymbol{\omega}} \le \rho\left(2\sigma_{\mathbf{s},\mathbf{M}}(x)_{1,\boldsymbol{\omega}} + \|z\|_{1,\boldsymbol{\omega}} - \|x\|_{1,\boldsymbol{\omega}} + \|v_\Theta\|_{1,\boldsymbol{\omega}}\right) + \sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\gamma\|Av\|_2,$$

and after rearranging we get

$$\|v_\Theta\|_{1,\boldsymbol{\omega}} \leq \frac{\rho}{1-\rho}\left(2\sigma_{\mathbf{s},\mathbf{M}}(x)_{1,\boldsymbol{\omega}} + \|z\|_{1,\boldsymbol{\omega}} - \|x\|_{1,\boldsymbol{\omega}}\right) + \frac{\gamma}{1-\rho}\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\|Av\|_2.$$

Therefore, using this and (III.30) once more, we deduce that

$$\begin{aligned}\|z-x\|_{1,\boldsymbol{\omega}} &= \|v_\Theta\|_{1,\boldsymbol{\omega}} + \|v_{\Theta^c}\|_{1,\boldsymbol{\omega}} \\ &\leq 2\|v_\Theta\|_{1,\boldsymbol{\omega}} + \left(2\sigma_{\mathbf{s},\mathbf{M}}(x)_{1,\boldsymbol{\omega}} + \|z\|_{1,\boldsymbol{\omega}} - \|x\|_{1,\boldsymbol{\omega}}\right) \\ &\leq \frac{1+\rho}{1-\rho}\left(2\sigma_{\mathbf{s},\mathbf{M}}(x)_{1,\boldsymbol{\omega}} + \|z\|_{1,\boldsymbol{\omega}} - \|x\|_{1,\boldsymbol{\omega}}\right) + \frac{2\gamma}{1-\rho}\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\|A(z-x)\|_2,\end{aligned}$$

which gives the result. ■

Lemma III.5.3 (weighted rNSPL implies ℓ^2 distance bound)**.** *Suppose that $A \in \mathbb{C}^{m\times M}$ satisfies the weighted rNSPL of order $(\mathbf{s},\mathbf{M})$ with constants $0<\rho<1$ and $\gamma>0$. Let $x, z \in \mathbb{C}^M$. Then*

$$\begin{aligned}\|z-x\|_2 \leq &\left(\rho + (1+\rho)(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}/2\right)\frac{\|z-x\|_{1,\boldsymbol{\omega}}}{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}} \\ &+ \left(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}/2\right)\gamma\|A(z-x)\|_2.\end{aligned} \tag{III.31}$$

Proof. Let $v = z - x$ and $\Theta = \Theta_1 \cup \cdots \cup \Theta_r$, where $\Theta_l \subseteq \{M_{l-1}+1, \dots, M_l\}$, $|\Theta_l| = s_l$ is the index set of the largest s_l coefficients of $P^{M_{l-1}}_{M_l} v$ in absolute value. Then

$$\|v_{\Theta_l}\|_2 = \sqrt{\sum_{i\in\Theta_l}|v_i|^2} \geq \sqrt{s_l}\min_{i\in\Theta_l}|v_i| \geq \sqrt{s_l}\max_{\substack{M_{l-1}<i\leq M_l\\ i\notin\Theta_l}}|v_i|, \quad l = 1,\dots,r,$$

which gives

$$\begin{aligned}\|v_{\Theta^c}\|_2^2 &= \sum_{l=1}^r \sum_{\substack{M_{l-1}<i\leq M_l\\ i\notin\Theta_l}} |v_i|^2 \leq \sum_{l=1}^r \max_{\substack{M_{l-1}<i\leq M_l\\ i\notin\Theta_l}}|v_i| \sum_{\substack{M_{l-1}<i\leq M_l\\ i\notin\Theta_l}}|v_i| \\ &\leq \sum_{l=1}^r \frac{\|v_{\Theta_l}\|_2}{\sqrt{s_l}} \sum_{\substack{M_{l-1}<i\leq M_l\\ i\notin\Theta_l}}|v_i| \leq \max_{l=1,\dots,r}\left\{\frac{\|v_{\Theta_l}\|_2}{\omega_l\sqrt{s_l}}\right\}\sum_{l=1}^r \omega_l \sum_{\substack{M_{l-1}<i\leq M_l\\ i\notin\Theta_l}}|v_i| \\ &\leq \max_{l=1,\dots,r}\left\{\frac{\|v_{\Theta_l}\|_2}{\omega_l\sqrt{s_l}}\right\}\|v_{\Theta^c}\|_{1,\boldsymbol{\omega}}\end{aligned}$$

Since $\|v_{\Theta_l}\|_2 \leq \|v_\Theta\|_2$ we deduce that

$$\|v_{\Theta^c}\|_2 \leq \sqrt{\frac{\|v_\Theta\|_2\|v_{\Theta^c}\|_{1,\boldsymbol{\omega}}}{\min_{l=1,\dots,r}\{\omega_l\sqrt{s_l}\}}} = \sqrt{\frac{\|v_\Theta\|_2\|v_{\Theta^c}\|_{1,\boldsymbol{\omega}}}{\sqrt{\zeta_{\boldsymbol{s},\boldsymbol{\omega}}}}}.$$

Applying Young's inequality $ab \leq \frac{1}{2}a^2 + \frac{1}{2}b^2$, we obtain

$$\|v_{\Theta^c}\|_2 \leq \frac{(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}}{2} \frac{\|v_{\Theta^c}\|_{1,\boldsymbol{\omega}}}{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}} + \frac{(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}}{2} \|v_\Theta\|_2.$$

Hence

$$\|v\|_2 \leq \|v_\Theta\|_2 + \|v_{\Theta^c}\|_2 \leq \left(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}/2\right) \|v_\Theta\|_2 + \frac{(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}}{2} \frac{\|v_{\Theta^c}\|_{1,\boldsymbol{\omega}}}{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}}.$$

We now use the weighted rNSPL to get

$$\|v\|_2 \leq \left(\rho + (1+\rho)(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}/2\right) \frac{\|v_{\Theta^c}\|_{1,\boldsymbol{\omega}}}{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}} + \left(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}/2\right) \gamma \|Av\|_2.$$

To complete the proof, we use the inequality $\|v_{\Theta^c}\|_{1,\boldsymbol{\omega}} \leq \|v\|_{1,\boldsymbol{\omega}}$. ■

III.5.2 Weighted rNSPL implies uniform recovery

Theorem III.5.4. *Let $\mathbf{M}, \boldsymbol{s} \in \mathbb{N}^r$ be sparsity levels and local sparsities, respectively, and let $\boldsymbol{\omega} \in \mathbb{R}^{r+1}$ be positive weights. Let $x \in \mathbb{C}^K$, with $K > M$ and $e \in \mathbb{C}^m$ with $\|e\|_2 \leq \eta$. Set $y = Ax + e$. Let $A \in \mathbb{C}^{m \times K}$ and suppose that AP_M satisfies the weighted rNSP in levels of order $(\boldsymbol{s}, \mathbf{M})$ with constants $\rho = \sqrt{3}/2$ and $\gamma > 0$. If*

$$\omega_{r+1} \geq \sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}} \left(\frac{1}{3(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4})} + 2\gamma \|AP_K^M\|_{1\to2} \right) \tag{III.32}$$

then any solution $\hat{x}$ of the optimization problem

$$\underset{z \in \mathbb{C}^K}{\text{minimise}} \, \|z\|_{1,\boldsymbol{\omega}} \quad \textit{subject to} \quad \|Az - y\|_2 \leq \eta \tag{III.33}$$

satisfies

$$\|x - \hat{x}\|_{1,\boldsymbol{\omega}} \leq C\sigma_{\boldsymbol{s},\mathbf{M}}(x)_{1,\boldsymbol{\omega}} + D\gamma\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\eta$$
$$\|x - \hat{x}\|_2 \leq \left(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}\right) \left(C\frac{\sigma_{\boldsymbol{s},\mathbf{M}}(x)_{1,\boldsymbol{\omega}}}{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}} + D\gamma\eta \right),$$

where $C = 2(2+\sqrt{3})/(2-\sqrt{3})$ and $D = 8/(2-\sqrt{3})$.

Proof. Recall that $\rho = \sqrt{3}/2$, and notice that this gives $C/2 = (1+\rho)/(1-\rho)$ and $D/2 = 2/(1-\rho)$. Next we consider the bound (III.32), and note that this bound implies

$$\omega_{r+1} \geq \gamma\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\|AP_K^M\|_{1\to2}/\rho \tag{III.34}$$
$$1 + 2\rho \geq 1 + 2\gamma\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\|AP_K^M\|_{1\to2}/\omega_{r+1} \tag{III.35}$$
$$1 + \rho \geq 1 - \rho + 2\gamma\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\|AP_K^M\|_{1\to2}/\omega_{r+1} \tag{III.36}$$
$$\frac{C}{2} \geq 1 + \frac{D}{2}\gamma\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\|AP_K^M\|_{1\to2}/\omega_{r+1}. \tag{III.37}$$

We also note that (III.32) implies

$$\begin{aligned}\omega_{r+1} &\geq \left(\frac{1}{3(1+(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4})} + 2\gamma\|AP_M^{\perp}\|_{1\to2}\right)\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\\ &\geq \left(\frac{2}{C(1+(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4})} + \frac{D}{C}\gamma\|AP_K^M\|_{1\to2}\right)\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\end{aligned}$$

which can be written as

$$(1+(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4})(C/2)\frac{1}{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}} \geq \Big((D/2)(1+(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4})\gamma\|AP_K^M\|_{1\to2}+1\Big)/\omega_{r+1}. \tag{III.38}$$

Next set $v = x - \hat{x}$ and consider the $\ell^{(1,\boldsymbol{\omega})}$-bound. First notice that since AP_M satisfies the weighted rNSPL, Lemma III.5.2 gives

$$\|P_M v\|_{1,\boldsymbol{\omega}} \leq (C/2)\left(2\sigma_{\boldsymbol{s},\mathbf{M}}(P_M x)_{1,\boldsymbol{\omega}} + \|P_M\hat{x}\|_{1,\boldsymbol{\omega}} - \|P_M x\|_{1,\boldsymbol{\omega}}\right) + (D/2)\gamma\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\|AP_M v\|_2. \tag{III.39}$$

Here the last term can be bounded by

$$\|AP_M v\|_2 \leq \|Av + y - y\|_2 + \|AP_K^M v\|_2 \leq 2\eta + \frac{\|AP_K^M\|_{1\to2}}{\omega_{r+1}}\|P_K^M v\|_{1,\boldsymbol{\omega}} \tag{III.40}$$

$$\leq 2\eta + \frac{\|AP_K^M\|_{1\to2}}{\omega_{r+1}}\left(\|P_K^M x\|_{1,\boldsymbol{\omega}} + \|P_K^M\hat{x}\|_{1,\boldsymbol{\omega}}\right), \tag{III.41}$$

since both x and $\hat{x}$ are feasible. Combining (III.37), (III.39) and (III.41) gives

$$\begin{aligned}\|v\|_{1,\boldsymbol{\omega}} \leq& \|P_M v\|_{1,\boldsymbol{\omega}} + \|P_K^M x\|_{1,\boldsymbol{\omega}} + \|P_K^M\hat{x}\|_{1,\boldsymbol{\omega}}\\ \leq& (C/2)\big(2\sigma_{\boldsymbol{s},\mathbf{M}}(P_M x)_{1,\boldsymbol{\omega}} + \|P_M\hat{x}\|_{1,\boldsymbol{\omega}} - \|P_M x\|_{1,\boldsymbol{\omega}}\big) + \|P_K^M x\|_{1,\boldsymbol{\omega}} + \|P_K^M\hat{x}\|_{1,\boldsymbol{\omega}}\\ &+ (D/2)\gamma\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\|AP_M v\|_2\\ \leq& (C/2)\big(2\sigma_{\boldsymbol{s},\mathbf{M}}(P_M x)_{1,\boldsymbol{\omega}} + \|P_M\hat{x}\|_{1,\boldsymbol{\omega}} - \|P_M x\|_{1,\boldsymbol{\omega}}\big) + D\gamma\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\eta\\ &+ \left(1 + (D/2)\gamma\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\frac{\|AP_K^M\|_{1\to2}}{\omega_{r+1}}\right)\left(\|P_K^M x\|_{1,\boldsymbol{\omega}} + \|P_K^M\hat{x}\|_{1,\boldsymbol{\omega}}\right)\\ \leq& (C/2)\big(2\sigma_{\boldsymbol{s},\mathbf{M}}(x)_{1,\boldsymbol{\omega}} + \|\hat{x}\|_{1,\boldsymbol{\omega}} - \|x\|_{1,\boldsymbol{\omega}}\big) + D\gamma\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\eta.\end{aligned}$$

Using that $\hat{x}$ is a minimizer of (III.33) gives the desired bound.

We now consider the ℓ^2-bound. First note that

$$\|v\|_2 \leq \|P_M v\|_2 + \|P_K^M v\|_2 \leq \|P_M v\|_2 + \frac{1}{\omega_{r+1}}\|P_K^M v\|_{1,\boldsymbol{\omega}}. \tag{III.42}$$

We shall also need

$$\begin{aligned}
&(\rho + (1+\rho)(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}/2)\frac{2}{1-\rho} + (1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}/2)\\
=&(D/4)\big(2\rho + (1+\rho)(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4} + (1-\rho) + (1-\rho)(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}/2\big)\\
=&(D/4)\big((1+\rho) + \tfrac{1}{2}(3+\rho)(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}\big)\\
\leq&(D/2)\big(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}\big)
\end{aligned} \tag{III.43}$$

Again, since AP_M satisfies the weighted rNSPL we can apply Lemma III.5.3, Lemma III.5.2 and inequality (III.43) to obtain the bound

$$\begin{aligned}
\|P_M v\|_2 \leq& \left(\rho + (1+\rho)(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}/2\right)\frac{\|P_M v\|_{1,\boldsymbol{\omega}}}{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}}\\
&+ \left(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}/2\right)\gamma\|AP_M v\|_2\\
\leq& \left(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}\right)(C/2)\frac{2\sigma_{\boldsymbol{s},\mathbf{M}}(P_M x)_{1,\boldsymbol{\omega}} + \|P_M\hat{x}\|_{1,\boldsymbol{\omega}} - \|P_M x\|_{1,\boldsymbol{\omega}}}{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}}\\
&+ \left(\rho + (1+\rho)(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}/2\right)\frac{2\gamma}{1-\rho}\|AP_M v\|_2\\
&+ \left(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}/2\right)\gamma\|AP_M v\|_2\\
\leq& \left(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}\right)(C/2)\frac{2\sigma_{\boldsymbol{s},\mathbf{M}}(P_M x)_{1,\boldsymbol{\omega}} + \|P_M\hat{x}\|_{1,\boldsymbol{\omega}} - \|P_M x\|_{1,\boldsymbol{\omega}}}{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}}\\
&+(D/2)\left(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}\right)\gamma\|AP_M v\|_2.
\end{aligned} \tag{III.44}$$

Combining (III.38), (III.41), (III.42), (III.44) and now gives

$$\begin{aligned}
\|v\|_2 \leq& \left(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}\right)(C/2)\frac{2\sigma_{\boldsymbol{s},\mathbf{M}}(P_M x)_{1,\boldsymbol{\omega}} + \|P_M\hat{x}\|_{1,\boldsymbol{\omega}} - \|P_M x\|_{1,\boldsymbol{\omega}}}{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}}\\
&+ (D/2)\left(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}\right)\gamma\|AP_M v\|_2 + \frac{1}{\omega_{r+1}}\|P_K^M v\|_{1,\boldsymbol{\omega}}\\
\leq& \left(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}\right)(C/2)\frac{2\sigma_{\boldsymbol{s},\mathbf{M}}(P_M x)_{1,\boldsymbol{\omega}} + \|P_M\hat{x}\|_{1,\boldsymbol{\omega}} - \|P_M x\|_{1,\boldsymbol{\omega}}}{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}}\\
&+ \left((D/2)\left(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}\right)\gamma\|AP_K^M\|_{1\to 2} + 1\right)\frac{\|P_K^M x\|_{1,\boldsymbol{\omega}} + \|P_K^M\hat{x}\|_{1,\boldsymbol{\omega}}}{\omega_{r+1}}\\
&+ \left(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}\right)D\gamma\eta\\
\leq& \left(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}\right)(C/2)\frac{2\sigma_{\boldsymbol{s},\mathbf{M}}(x)_{1,\boldsymbol{\omega}} + \|\hat{x}\|_{1,\boldsymbol{\omega}} - \|P_K x\|_{1,\boldsymbol{\omega}}}{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}}\\
&+ \left(1 + (S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4}\right)D\gamma\eta
\end{aligned}$$

Using that $\hat{x}$ is a minimizer of (III.33) completes the proof. ■

III.5.3 G-RIPL implies weighted rNSPL

Theorem III.5.5. *Let $A \in \mathbb{C}^{m\times M}$ and let $G \in \mathbb{C}^{M\times M}$ be invertible. Let $\mathbf{M} \in \mathbb{N}^r$ be sparsity levels, $\boldsymbol{s}, \boldsymbol{t} \in \mathbb{N}^r$ be local sparsities and let $\boldsymbol{\omega} \in \mathbb{R}^r$ be positive weights. Suppose that A satisfies the G-RIPL of order $(\boldsymbol{t}, \mathbf{M})$ with constant $0 < \delta_{\boldsymbol{t},\mathbf{M}} < 1$, where*

$$t_l = \min\left\{M_l - M_{l-1}, 2\left\lceil \left(\frac{1+\delta}{1-\delta}\right)\frac{\kappa(G)^2}{\rho^2\omega_l^2}S_{\boldsymbol{\omega},\boldsymbol{s}}\right\rceil\right\}, \quad \textit{for } l = 1,\dots,r. \quad \text{(III.45)}$$

Then A satisfies the weighted rNSP in levels of order $(\boldsymbol{s}, \mathbf{M})$ with constants $0 < \rho < 1$ and $\gamma = \|G^{-1}\|_2/\sqrt{1-\delta}$.

Proof. Let $x \in \mathbb{C}^K$ be such that $P_M^{\perp}x = 0$ and let $\Theta = \Theta_1 \cup \cdots \cup \Theta_r$, where Θ_l is the set of the largest s_l indices of $P_{M_l}^{M_{l-1}}x$ in absolute value. If $t_l = M_l - M_{l-1}$, let $T_{l,0} = \{M_{l-1}+1,\dots,M_l\}$ and let $T_{l,k} = \emptyset$ for $k \geq 1$. For $t_l < M_l - M_{l-1}$ let $T_{l,0}$ be the index set of the largest $t_l/2$ values of $|P_{M_l}^{M_{l-1}}x|$, and let $T_{l,1}$ be the index set of the next $t_l/2$ largest values and so forth. In the case where there are less than $t_l/2$ values left at iteration k, we let $T_{l,k}$ be the remaining indices. Let $T_k = T_{1,k} \cup \cdots \cup T_{r,k}$ and let $T_{\{0,1\}} = T_0 \cup T_1$. Since $\Theta \subseteq T_{\{0,1\}}$ we have

$$\|x_\Theta\|_2^2 \leq \|x_{T_{\{0,1\}}}\|_2^2 \leq \|G^{-1}\|_2^2\|Gx_{T_{\{0,1\}}}\|_2^2 \leq \frac{\|G^{-1}\|_2^2}{1-\delta}\|Ax_{T_{\{0,1\}}}\|_2^2$$

where $\delta = \delta_{\boldsymbol{t},\mathbf{M}}$. Note that

$$Ax_{T_{\{0,1\}}} = Ax - \sum_{k\geq 2} Ax_{T_k},$$

Then

$$\begin{aligned}
\|Ax_{T_{\{0,1\}}}\|_2^2 &= \langle Ax_{T_{\{0,1\}}}, Ax\rangle - \sum_{k\geq 2}\langle Ax_{T_{\{0,1\}}}, Ax_{T_k}\rangle \\
&\leq \|Ax_{T_{\{0,1\}}}\|_2\|Ax\|_2 + \|Ax_{T_{\{0,1\}}}\|_2\sum_{k\geq 2}\|Ax_{T_k}\|_2 \\
&\leq \|Ax_{T_{\{0,1\}}}\|_2\|Ax\|_2 + \sqrt{1+\delta}\|Ax_{T_{\{0,1\}}}\|_2\sum_{k\geq 2}\|Gx_{T_k}\|_2 \\
&\leq \|Ax_{T_{\{0,1\}}}\|_2\|Ax\|_2 + \sqrt{1+\delta}\|G\|_2\|Ax_{T_{\{0,1\}}}\|_2\sum_{k\geq 2}\|x_{T_k}\|_2
\end{aligned}$$

Set $\Delta = \{l \in \{1,\dots,r\} : t_l < M_l - M_{l-1}\}$ and notice that $T_{l,k} = \emptyset$ for $l \in \{1,\dots,r\}\setminus\Delta$ and $k \geq 1$. Thus for $k \geq 2$ we get

$$\begin{aligned}
\|x_{T_k}\|_2^2 &= \sum_{l\in\Delta}\|x_{T_{l,k}}\|_2^2 \leq \sum_{l\in\Delta}\frac{2\|x_{T_{l,k-1}}\|_1^2}{t_l} = \sum_{l\in\Delta}\frac{2\|x_{T_{l,k-1}}\|_1^2\omega_l^2}{t_l\omega_l^2} \\
&\leq \frac{\sum_{l\in\Delta}2\|x_{T_{l,k-1}}\|_{1,\boldsymbol{\omega}}^2}{\min_{l\in\Delta}\{\omega_l^2 t_l\}} \leq \frac{2\|x_{T_{k-1}}\|_{1,\boldsymbol{\omega}}^2}{\min_{l\in\Delta}\{\omega_l^2 t_l\}}.
\end{aligned}$$

Therefore

$$\begin{aligned}\|Ax_{T_{\{0,1\}}}\|_2 &\le \|Ax\|_2 + \frac{\sqrt{2(1+\delta)}\|G\|_2}{\sqrt{\min_{l\in\Delta}\{\omega_l^2 t_l\}}}\sum_{k\ge 2}\|x_{T_{k-1}}\|_{1,\boldsymbol{\omega}}\\ &\le \|Ax\|_2 + \frac{\sqrt{2(1+\delta)}\|G\|_2}{\sqrt{\min_{l\in\Delta}\{\omega_l^2 t_l\}}}\|x_{T_0^c}\|_{1,\boldsymbol{\omega}}\\ &\le \|Ax\|_2 + \frac{\sqrt{1+\delta}\|G\|_2}{\min_{l\in\Delta}\{\omega_l\sqrt{t_l/2}\}}\|x_{\Theta^c}\|_{1,\boldsymbol{\omega}}.\end{aligned}$$

This results in

$$\begin{aligned}\|x_\Theta\|_2 \le&\sqrt{\frac{1+\delta}{1-\delta}}\|G\|_2\|G^{-1}\|_2\frac{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}}{\min_{l\in\Delta}\{\omega_l\sqrt{t_l/2}\}}\frac{\|x_{\Theta^c}\|_{1,\boldsymbol{\omega}}}{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}}+\frac{\|G^{-1}\|_2}{\sqrt{1-\delta}}\|Ax\|_2\\ \le&\rho\frac{\|x_{\Theta^c}\|_{1,\boldsymbol{\omega}}}{\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}}+\frac{\|G^{-1}\|_2}{\sqrt{1-\delta}}\|Ax\|_2\end{aligned}$$

which establishes the weighted rNSPL of order $(\boldsymbol{s},\mathbf{M})$ with $0<\rho<1$ and $\gamma=\|G^{-1}\|_2/\sqrt{1-\delta}$. ■

III.5.4 Proof of Theorem III.3.5

Proof of Theorem III.3.5. First notice that for $0<\delta\le 1/2$ we have

$$\frac{1+\delta}{1-\delta}\le 3 \quad\text{and}\quad \frac{1}{\sqrt{1-\delta}}\le\sqrt{2}.$$

Hence using Theorem III.5.5 with $0<\delta_{\boldsymbol{t},\mathbf{M}}\le\delta\le 1/2$ and $\rho=\sqrt{3}/2$ we see that Equation (III.45), simplifies to Equation (III.10). This implies that AP_M satisfies the weighted rNSPL of order $(\boldsymbol{s},\mathbf{M})$, with constants $\rho=\sqrt{3}/2$ and $\gamma=\sqrt{2}\|G^{-1}\|_2$. Now since

$$\omega_{r+1}\ge\sqrt{S_{\boldsymbol{\omega},\boldsymbol{s}}}\left(\tfrac{1}{3}(1+(S_{\boldsymbol{\omega},\boldsymbol{s}}/\zeta_{\boldsymbol{s},\boldsymbol{\omega}})^{1/4})^{-1}+2\sqrt{2}\|AP_K^M\|_{1\to 2}\|G^{-1}\|_2\right)$$

we know from Theorem III.5.4 that any solution $\hat{x}$ of (III.11) satisfies (III.12) and (III.13). ■

III.5.5 Proof of Theorem III.3.6

Proof of Theorem III.3.6. We recall that $U\in\mathcal{B}(\ell^2)$ is an isometry and that

$$A=\begin{bmatrix}1/\sqrt{p_1}P_{\Omega_1}UP_M\\ 1/\sqrt{p_2}P_{\Omega_2}UP_M\\ \vdots\\ 1/\sqrt{p_r}P_{\Omega_r}UP_M\end{bmatrix}\in\mathbb{C}^{m\times M},\qquad\text{where}\qquad p_k=m_k/(N_k-N_{k-1}),$$

and $m = m_1 + \ldots + m_r$. Note that

$$\|Ax\|^2 - \|Gx\|^2 = \langle (A^*A - G^*G)x, x \rangle,$$

and therefore

$$\delta = \sup_{\Theta \in E_{\mathbf{s},\mathbf{M}}} \|P_\Theta (A^*A - G^*G) P_\Theta\|_2.$$

Notice also that $p_k = 1$ and $\Omega_k = \{N_{k-1}+1, \ldots, N_k\}$ for $k = 1, \ldots, r_0$. Next notice that the matrix P_{Ω_k} can be written as

$$P_{\Omega_k} = \sum_{i=1}^{m_k} e_{t_{k,i}} e_{t_{k,i}}^*,$$

where $\{e_i\}_{i=1}^\infty$ is the standard basis on $\ell^2(\mathbb{N})$. It now follows that

$$\begin{aligned} A^*A &= \sum_{k=1}^{r} \frac{1}{p_k} P_M U^* P_{\Omega_k} U P_M = \sum_{k=1}^{r} \frac{1}{p_k} \sum_{i=1}^{m_k} P_M U^* e_{t_{k,i}} e_{t_{k,i}}^* U P_M \\ &= P_M U^* P_{N_{r_0}} U P_M + \sum_{k=r_0+1}^{r} \sum_{i=1}^{m_k} X_{k,i} X_{k,i}^*, \end{aligned}$$

where $X_{k,i}$ are random vectors given by $X_{k,i} = \frac{1}{\sqrt{p_k}} P_M U^* e_{t_{k,i}}$. Note that the $X_{k,i}$ are independent, and also that

$$\begin{aligned} \mathbb{E}(A^*A) &= P_M U^* P_{N_{r_0}} U P_M + \sum_{k=r_0+1}^{r} \sum_{i=1}^{m_k} \mathbb{E}\left(X_{k,i} X_{k,i}^*\right) \\ &= P_M U^* P_{N_{r_0}} U P_M + \sum_{k=r_0+1}^{r} \frac{m_k}{p_k(N_k - N_{k-1})} \sum_{j=N_{k-1}+1}^{N_k} P_M U^* e_j e_j^* U P_M \\ &= P_M U^* P_{N_{r_0}} U P_M + P_M U^* P_{N_r}^{N_{r_0}} U P_M \\ &= P_M U^* P_N U P_M && \text{(III.46)} \\ &= G^2, && \text{(III.47)} \end{aligned}$$

where $G \in \mathbb{C}^{M \times M}$ is non-singular by assumption. Let

$$D_{\mathbf{s},\mathbf{M},G} = \left\{ \eta \in \mathbb{C}^M : \|G\eta\|_2 \leq 1,\ |\operatorname{supp}(\eta) \cap \{M_{k-1}+1, \ldots, M_k\}| \leq s_k,\ k = 1, \ldots, r \right\}.$$

We now define the following seminorm on $\mathbb{C}^{M \times M}$:

$$\|B\|_{\mathbf{s},\mathbf{M},G} := \sup_{z \in D_{\mathbf{s},\mathbf{M},G}} |\langle Bz, z \rangle|,$$

so that

$$\delta_{\mathbf{s},\mathbf{M}} = \|A^*A - G^*G\|_{\mathbf{s},\mathbf{M}}.$$

Due to (III.46) and (III.47), we may rewrite this as

$$\delta_{\mathbf{s},\mathbf{M}} = \| \sum_{k=r_0+1}^{r} \sum_{i=1}^{m_k} \left(X_{k,i}X_{k,i}^* - \mathbb{E}(X_{k,i}X_{k,i}^*) \right) \|_{\mathbf{s},\mathbf{M}}. \tag{III.48}$$

Having detailed the setup, the remainder of the proof now follows along very similar lines to that of [23, Thm. 3.2]. Hence we only sketch the details.

The first step is to estimate $\mathbb{E}\left(\delta_{\mathbf{s},\mathbf{M}}\right)$. Using the standard techniques of symmetrization, Dudley's inequality, properties of covering numbers, and arguing as in [23, Sec. 4.2], we deduce that

$$\mathbb{E}\left(\delta_{\mathbf{s},\mathbf{M}}\right) \leq D + D^2, \qquad D = C_1\sqrt{\frac{rQ\|G^{-1}\|_2^2 \log(2\tilde{m})\log(2M)\log(2s)}{m}}, \tag{III.49}$$

where $C_1 > 0$ is a universal constant, $\tilde{m} = \sum_{k=r_0+1}^{r} m_k$, and

$$Q = \max_{k=r_0+1,\dots,r} \sum_{l=1}^{r} \frac{\mu_{k,l}s_l}{p_k}. \tag{III.50}$$

In particular,

$$\mathbb{E}\left(\delta_{\mathbf{s},\mathbf{M}}\right) \leq \delta/2,$$

provided

$$C_2 Q\left\|G^{-1}\right\|_2^2 \delta^{-2} r \log(2\tilde{m})\log(2M)\log^2(2s) \leq 1, \tag{III.51}$$

where $C_2 > 0$ is a constant. Using this, Talagrand's theorem and using the fact that $\|P_N U P_M\|_2 \leq \|U\|_2 = 1$ (see [23, Sec. 4.3]) we deduce that

$$\mathbb{P}(\delta_{\mathbf{s},\mathbf{M}} \geq \delta) \leq \exp\left(-3\delta^2/(8(3+7\delta)Q\left\|G^{-1}\right\|_2^2)\right).$$

In particular,

$$\mathbb{P}(\delta_{\mathbf{s},\mathbf{M}} \geq \delta) \leq \epsilon,$$

provided

$$\frac{80}{3}Q\left\|G^{-1}\right\|_2^2 \delta^{-2}\log(\epsilon^{-1}) \leq 1.$$

Combining this with (III.50) and (III.51) now completes the proof. ∎

III.5.6 Proof of Corollary III.3.7 and Lemma III.3.3

Proof of Corollary III.3.7. We must ensure that all the conditions are met to be able to apply Theorem III.3.5 with $P_K x$.

First notice that for weights $\boldsymbol{\omega} = (s_1^{-1/2}, \dots, s_r^{-1/2}, \omega_{r+1})$ we have $S_{\boldsymbol{\omega},\boldsymbol{s}} = r$ and $\zeta_{\boldsymbol{s},\boldsymbol{\omega}} = 1$. Next we note that condition (ii) implies that $P_K x$ is a feasible point since $\|HP_K x - \tilde{y}\|_2 \leq \|HP_K^{\perp} x\|_2 + \|e_1\|_2 = \eta + \eta'$.

Let $G = \sqrt{P_M U^* P_N U P_M}$. Combining condition (i) and Lemma III.3.3 gives $\|G^{-1}\|_2 \leq 1/\sqrt{\theta}$ and since $\|G\|_2 \leq 1$ we also have $\kappa(G) = \|G\|_2 \|G^{-1}\|_2 \leq 1/\sqrt{\theta}$. Inserting the above equalities and inequalities into the weight condition for ω_{r+1} in Theorem III.3.5 gives condition (iii).

Next we must ensure that AP_M satisfies the G-RIPL of order $(\boldsymbol{t}, \mathbf{M})$ with $\delta_{\boldsymbol{t},\mathbf{M}} \leq 1/2$ where

$$t_l = \min\left\{M_l - M_{l-1}, 2\left\lceil 4\theta^{-1} r s_l \right\rceil\right\}. \tag{III.52}$$

According to Theorem III.3.6 this occurs if the m_k's satisfies condition (iv). The error bounds (III.12) and (III.13) now follows directly from Theorem III.3.5. ■

Proof of lemma III.3.3. First notice that the balancing property is equivalent to requiring

$$\sigma_M(P_N U P_M) \geq \sqrt{\theta} \tag{III.53}$$

where $\sigma_M(P_N U P_M)$ is the Mth largest singular value of $P_N U P_M$. Indeed, since U is an isometry, the matrix $P_M - P_M U^* P_N U P_M$ is nonnegative definite, and therefore

$$\begin{aligned}
\|P_M U^* P_N U_P M - P_M\|_2 &= \sup_{x \in \mathbb{C}^M, \|x\|_2 \leq 1} \langle (P_M U^* P_N U_P M - P_M)x, x\rangle && \text{(III.54)}\\
&= \sup_{x \in \mathbb{C}^M, \|x\|_2 \leq 1} (\|P_M x\|_2 - \|P_N U P_M x\|_2) && \text{(III.55)}\\
&= 1 - \inf_{x \in \mathbb{C}^M, \|x\|_2 = 1} \|P_N U P_M x\|_2 && \text{(III.56)}
\end{aligned}$$

This gives (III.53). Next let $G = \sqrt{P_M U^* P_N U P_M}$ and notice that $\sigma_M(G) = \sigma_M(P_N U P_M)$. This gives $\|G^{-1}\|_2 = 1/\sigma_M(G) \leq 1/\sqrt{\theta}$. ■

III.6 Proofs of results in Section III.4

In Section III.4 we found concrete recovery guarantees for the Walsh sampling and wavelet reconstruction, using the theorems in Section III.3. The key to deriving Walsh-wavelet recovery guarantees boils down to estimating the quantities $\mu_{k,l}$, $||HP_K^M||_{1\to 2}$ and $||G^{-1}||_2 \leq \frac{1}{\sqrt{\theta}}$. All of these quantities depend directly $U = [B_{\text{wh}}, B_{\text{wave}}^{J_0,\nu}]$, and to control them we will have to estimate how the entries of U changes for varying n, j, k and s. We will therefore start this section by setting up notation for wavelets on the interval and stating some useful properties of Walsh functions. Then in Section III.6.3 and III.6.4 we will estimate $\mu_{k,l}$, followed by a discussion of the sharpness of this estimate for $\nu = 2$ in Section III.6.5. We will then finish in Section III.6.6 by estimating $||HP_K^M||_{1\to 2}$, show how θ scales for varying M and N, and prove Theorem III.4.7 and III.4.8.

III.6.1 Wavelets on the interval and regularity

In section III.4.2 we introduced orthogonal wavelets on the real line, but we did not make any formal definitions of the wavelets we used at the boundaries of the interval $[0,1)$. Next we consider the two boundary extensions, *periodic* and *boundary wavelets*. To simplify the exposition, we define the following sets

$$\begin{aligned}
&\Lambda_{\nu,j,\text{left}} := \{0,\dots,\nu-1\}, && \Lambda_{\nu,j,\text{mid}} := \{\nu,\dots,2^j-\nu-1\},\\
&\Lambda_{\nu,j,\text{right}} := \{2^j-\nu,\dots,2^j-1\} && \Lambda_j = \Lambda_{\nu,j,\text{left}} \cup \Lambda_{\nu,j,\text{mid}} \cup \Lambda_{\nu,j,\text{right}}
\end{aligned}$$

At each scale $j \geq J_0$, the periodic wavelet basis consists of the usual wavelets and scaling functions $\psi_{j,k}$, $\phi_{j,k}$ for $k \in \Lambda_{\nu,j,\text{mid}}$ and the periodic extended functions $\phi^{\text{per}}_{j,k}$ and $\psi^{\text{per}}_{j,k}$ for $k \in \Lambda_{\nu,j,\text{left}} \cup \Lambda_{\nu,j,\text{right}}$. These are defined as

$$\phi^{\text{per}}_{j,k} := \phi_{j,k}|_{[0,1)} + \phi_{j,2^j+k}|_{[0,1)} \qquad \text{for } k \in \Lambda_{\nu,j,\text{left}} \tag{III.57}$$

$$\phi^{\text{per}}_{j,k} := \phi_{j,2^j-\nu-k}|_{[0,1)} + \phi_{j,k}|_{[0,1)} \qquad \text{for } k \in \Lambda_{\nu,j,\text{right}} \tag{III.58}$$

and similarly for $\psi^{\text{per}}_{j,k}$. Strictly speaking we could have defined these periodic extensions only for $k = 0,\dots,\nu-2$ and $k = 2^j-\nu+1,\dots,2^j-1$, but to unify the notation for both boundary extensions we have chosen the former.

Next we have the boundary wavelet basis with ν vanishing moments. This wavelet basis consists of the same interior wavelets as the periodic basis, but with 2ν boundary scaling and wavelet functions.

$$\phi^{\text{left}}_k, \phi^{\text{right}}_k, \psi^{\text{left}}_k, \psi^{\text{right}}_k, \qquad \text{for } k = 0,\dots,\nu-1.$$

As for the interior functions we also define the scaled versions as

$$\begin{aligned}
\phi^{\text{left}}_{j,k}(x) &:= 2^{j/2}\phi^{\text{left}}_k(2x), \quad & \phi^{\text{right}}_{j,k}(x) &:= 2^{j/2}\phi^{\text{right}}_k(2x),\\
\psi^{\text{left}}_{j,k}(x) &:= 2^{j/2}\psi^{\text{left}}_k(2x), \quad & \psi^{\text{right}}_{j,k}(x) &:= 2^{j/2}\psi^{\text{right}}_k(2x).
\end{aligned} \tag{III.59}$$

The names 'left' and 'right' corresponds to the support of these functions. That is

$$\begin{aligned}
\operatorname{supp}\phi^{\text{left}}_{j,k} &= [0, 2^{-j}(\nu+k)]\\
\operatorname{supp}\phi^{\text{right}}_{j,k} &= [2^{-j}(2^j-\nu-k), 1]
\end{aligned}$$

for $k = 0,\dots,\nu-1$.

In the following we shall see that all of our results holds for both periodic and boundary wavelets, but their treatment in some of the proofs differs slightly. To make the treatment as unified as possible we make the following definition.

Definition III.6.1. We say that $\phi^s_{j,k}$, $s \in \{0,1\}$ 'originates from a periodic wavelet' if

$$\phi^0_{j,k} := \begin{cases} \phi^{\text{per}}_{j,k} & \text{for } k \in \Lambda_{\nu,j,\text{left}} \\ \phi_{j,k} & \text{for } k \in \Lambda_{\nu,j,\text{mid}} \\ \phi^{\text{per}}_{j,k} & \text{for } k \in \Lambda_{\nu,j,\text{right}} \end{cases}, \qquad \phi^1_{j,k} := \begin{cases} \psi^{\text{per}}_{j,k} & \text{for } k \in \Lambda_{\nu,j,\text{left}} \\ \psi_{j,k} & \text{for } k \in \Lambda_{\nu,j,\text{mid}} \\ \psi^{\text{per}}_{j,k} & \text{for } k \in \Lambda_{\nu,j,\text{right}} \end{cases}.$$

We say that $\phi^s_{j,k}$ 'originates from a boundary wavelet' if

$$\phi^0_{j,k} := \begin{cases} \phi^{\text{left}}_{j,k} & \text{for } k \in \Lambda_{\nu,j,\text{left}} \\ \phi_{j,k} & \text{for } k \in \Lambda_{\nu,j,\text{mid}} \\ \phi^{\text{right}}_{j,2^j-1-k} & \text{for } k \in \Lambda_{\nu,j,\text{right}} \end{cases}, \qquad \phi^1_{j,k} := \begin{cases} \psi^{\text{left}}_{j,k} & \text{for } k \in \Lambda_{\nu,j,\text{left}} \\ \psi_{j,k} & \text{for } k \in \Lambda_{\nu,j,\text{mid}} \\ \psi^{\text{right}}_{j,2^j-1-k} & \text{for } k \in \Lambda_{\nu,j,\text{right}} \end{cases}.$$

With these functions defined now for both boundary extensions, the definition of $B_{\text{wave}}^{J_0,\nu}$ is also clear. Next we make a note on the regularity of these orthogonal wavelets.

Definition III.6.2. Let $\alpha = k + \beta$, where $k \in \mathbb{Z}_+$ and $0 < \beta < 1$. A function $f\colon \mathbb{R} \to \mathbb{R}$ is said to be uniformly Lipschitz α if f is k-times continuously differentiable and for which the k^{th} derivative $f^{(k)}$ is Hölder continuous with exponent β, i.e.

$$|f^{(k)}(x) - f^{(k)}(y)| < C|x-y|^{\beta}, \quad \forall x, y \in \mathbb{R}$$

for some constant $C > 0$.

In particular the Daubechies wavelet with 1 vanishing moment (i.e., the Haar wavelet) is not uniformly Lipschitz as it is not continuous, whereas for $\nu \geq 2$ we have the constants found in table III.1 [13, p. 239]. For large ν, α grows as 0.2ν [26, p. 294]. Also note that each of the boundary functions $\phi_k^{\text{left}}, \phi_k^{\text{right}}$ and $\psi_k^{\text{left}}, \phi_k^{\text{right}}$ are constructed as finite linear combinations of the interior scaling function ϕ and wavelet ψ. Thus all of these boundary functions has the same regularity as ϕ and ψ.

ν	α
2	0.55
3	1.08
4	1.61

Table III.1: The Lipschitz regularity of Daubechies wavelets with ν vanishing moments.

III.6.2 Properties of Walsh functions

Definition III.6.3. Let $x = \{x_i\}_{i=1}^{\infty}$ and $y = \{y_i\}_{i=1}^{\infty}$ be sequences consisting of only binary numbers. That is $x_i, y_i \in \{0, 1\}$ for all $i \in \mathbb{N}$. The operation $\oplus$ applied to these sequences gives

$$x \oplus y \coloneqq \{|x_i - y_i|\}_{i=1}^{\infty}. \tag{III.60}$$

For two binary numbers $x_i, y_i \in \{0, 1\}$, we let $x_i \oplus y_i = |x_i - y_i|$.

Proposition III.6.4. *For $j, m, n \in \mathbb{Z}_+$ and $x, y \in [0, 1)$, the Walsh function satisfies the the following properties*

$$\int_0^1 w_n(x) w_m(x)\ dx = \begin{cases} 1 & \text{if } m = n \\ 0 & \text{otherwise} \end{cases} \tag{III.61}$$

$$w_n(x \oplus y) = w_n(x) w_n(y) \tag{III.62}$$

$$w_n(2^{-j}x) = w_{\lfloor n/2^j \rfloor}(x) \tag{III.63}$$

Proof. Equation (III.62) and (III.61) can be found in any standard text on Walsh functions e.g., [18], whereas the last follows by inserting j zeros in front of x's dyadic expansion. ■

III.6.3 Bounding the inner product $|\langle \phi^s_{j,k}, w_n\rangle|$

The entries in $U = [B_{\mathrm{wh}}, B^{J_0,\nu}_{\mathrm{wave}}]$, consists of $\langle \phi^s_{j,k}, w_n\rangle$ for different values of j, k, s and n. Thus in order to determine the local coherences we need to find an upper bound of this inner product. Next we derive such an bound for $\nu \geq 2$ vanishing moments and discusses its sharpness. For $\nu = 1$ we determine the magnitude of each matrix entry explicitly.

Lemma III.6.5. *Let* $w_n \in B_{\mathrm{wh}}$ *and let* $\phi^s_{j,k} \in B^{J_0,\nu}_{\mathrm{wave}}$ *for* $\nu \geq 2$. *For* $j \geq J_0$, $s \in \{0,1\}$ *and* $k \in \Lambda_j$ *we have*

$$\left|\langle \phi^s_{j,k}, w_n\rangle\right| \leq 2^{-j/2} 2\nu \max_{l\in\Gamma_k} \left\{ \left| \mathcal{W}\phi^s(\cdot + l)\right|_{[0,1)} \left(\left\lfloor \frac{n}{2^j} \right\rfloor \right) \right| \right\} \tag{III.64}$$

where

$$\Gamma_k = \begin{cases} \{0, \dots, \nu + k - 1\} & \text{for } k \in \Lambda_{\nu,j,\mathrm{left}}; \\ \{-\nu + 1, \dots, \nu - 1\} & \text{for } k \in \Lambda_{\nu,j,\mathrm{mid}}; \\ \{k - \nu + 1, \dots, 2^j - 1\} & \text{for } k \in \Lambda_{\nu,j,\mathrm{right}}. \end{cases}$$

if $\phi^s_{j,k}$ *originates from a boundary wavelet and*

$$\Gamma_k = \{-\nu + 1, \dots, \nu - 1\}$$

if $\phi^s_{j,k}$ *originates from a periodic wavelet.*

Proof. First notice that for any $x \in [0,1)$ we have

$$\begin{aligned} \frac{x}{2^j} + \frac{k}{2^j} &= \sum_{i=j}^{\infty} x_{i-j+1} 2^{-i-1} + \sum_{i=1}^{j} k_i 2^{-j-1+i} \\ &= \sum_{i=j}^{\infty} x_{i-j+1} 2^{-i-1} \oplus \sum_{i=1}^{j} k_i 2^{-j-1+i} = \frac{x}{2^j} \oplus \frac{k}{2^j}. \end{aligned} \tag{III.65}$$

Next, we only consider the interior wavelets $\phi^s_{j,k}$ i.e. $k \in \Lambda_{\nu,j,\mathrm{mid}}$. For $k \in \Lambda_{\nu,j,\mathrm{left}} \cup \Lambda_{\nu,j,\mathrm{right}}$, we need to handle the two cases where $\phi^s_{j,k}$ orignates from a periodic and boundary wavelet seperately. The arguments/calculations for the two different boundary extensions are analogous. Also, both of these extensions will have support less than 2ν.

For $k \in \Lambda_{\nu,j,\mathrm{mid}}$, notice that $\operatorname{supp}(\phi^s_{j,k}) = [2^{-j}(-\nu+1+k), 2^{-j}(\nu+k)]$. This gives

$$\begin{aligned}
\left\langle \phi^s_{j,k}, w_n \right\rangle &= \int_0^1 \phi^s_{j,k}(x) w_n(x) \; \mathrm{d}x \\
&= \int_{2^{-j}(-\nu+1+k)}^{2^{-j}(\nu+k)} 2^{j/2} \phi^s(2^j x - k) w_n(x) \; \mathrm{d}x \\
&= 2^{-j/2} \int_{-\nu+1}^{\nu} \phi^s\left(x\right) w_n\left(\frac{x+k}{2^j}\right) \; \mathrm{d}x \\
&= 2^{-j/2} \sum_{l=-\nu+1}^{\nu-1} \int_0^1 \phi^s\left(x+l\right) w_n\left(\frac{x+l+k}{2^j}\right) \; \mathrm{d}x.
\end{aligned}$$

Using Equation (III.65) and Proposition III.6.4 now gives

$$\begin{aligned}
\left\langle \phi^s_{j,k}, w_n \right\rangle &= 2^{-j/2} \sum_{l=-\nu+1}^{\nu-1} \int_0^1 \phi^s\left(x+l\right) w_n\left(\frac{x}{2^j} \oplus \frac{l+k}{2^j}\right) \; \mathrm{d}x \\
&= 2^{-j/2} \sum_{l=-\nu+1}^{\nu-1} w_n\left(\frac{l+k}{2^j}\right) \int_0^1 \phi^s\left(x+l\right) w_n\left(\frac{x}{2^j}\right) \; \mathrm{d}x \\
&= 2^{-j/2} \sum_{l=-\nu+1}^{\nu-1} w_n\left(\frac{l+k}{2^j}\right) \mathcal{W}\phi^s_{0,-l}\big|_{[0,1)} \left(\left\lfloor \frac{n}{2^j} \right\rfloor\right) \\
&\leq 2^{-j/2} 2\nu \max_{l \in \Gamma_k} \left\{ \left| \mathcal{W}\phi^s(\cdot + l)\big|_{[0,1)} \left(\left\lfloor \frac{n}{2^j} \right\rfloor\right) \right| \right\}.
\end{aligned}$$

■

Lemma III.6.6 ([9]). *Let $f : [0,1) \to \mathbb{R}$ be uniformly Lipschitz $0 < \alpha \leq 1$ then*

$$|\mathcal{W}f(n)| = \left| \int_0^1 f(x) w_n(x) \; dx \right| \lesssim (n+1)^{-\alpha}$$

for $n \in \mathbb{Z}_+$.

Theorem III.6.7. *Let $\phi^s_{l,t} \in B^{J_0,\nu}_{\mathrm{wave}}$ with $\nu \geq 3$ and let $w_n \in B_{\mathrm{wh}}$. For $l \geq J_0$ and $2^k \leq n < 2^{k+1}$ with $k \in \mathbb{Z}_+$, we have*

$$|\left\langle \phi^s_{l,t}, w_n \right\rangle|^2 \lesssim 2^{-k} 2^{-|l-k|}$$

for all $t \in \Lambda_l$ and $s \in \{0,1\}$. For $n = 0$ the bound hold with $k = 0$.

Proof. To obtain the bound above we will combine Lemma III.6.5 and Lemma III.6.6. We start by arguing that $\phi^s_{l,t}$ have the same regularity regardless of boundary extension. Let $a \in \Gamma_t$ where Γ_t is as in lemma III.6.5.

If $\phi^s_{l,t}$ originates from a periodic wavelet, $\phi^s_{0,-a}|_{[0,1)}$, will have Lipschitz regularity $\alpha > 0$, since both ϕ and ψ have this regularity. Next if $\phi^s_{l,t}$ originates from a boundary wavelet and $t \in \Lambda_{\nu,l,\text{mid}}$, $\phi^s_{0,-a}|_{[0,1)}$ will have Lipschitz regularity α, by the same argument as above. If $t \in \Lambda_{\nu,l,\text{left}} \cup \Lambda_{\nu,l,\text{right}}$ we know from the construction of the boundary functions [12] that these are finite linear combinations of $\phi_{l,t}$ and $\psi_{l,t}$. These function will therefore posses the same regularity α as the interior function.

Next notice from table III.1 that for $\nu \geq 3$ vanishing moments, we known that $\alpha \geq 1$. Applying Lemma III.6.5 and Lemma III.6.6 then gives

$$\left|\left\langle \phi^s_{l,t}, w_n \right\rangle\right|^2 \leq 2^{-l} 4\nu^2 \max_{a \in \Gamma_t} \left\{ \left| \mathcal{W}\phi^s(\cdot + a)\big|_{[0,1)} \left(\left\lfloor \frac{n}{2^l} \right\rfloor \right) \right|^2 \right\} \tag{III.66}$$

$$\lesssim 2^{-l} \frac{1}{(\lfloor \frac{n}{2^l} \rfloor + 1)^2} \leq 2^{-l} \frac{1}{(\lfloor 2^{k-l} \rfloor + 1)^2} \leq 2^{-k} 2^{-|l-k|} \tag{III.67}$$

where Γ_t depends on the boundary extension. ■

Theorem III.6.8. *Let $w_n \in B_{\text{wh}}$ and let $\phi^s_{l,t} \in B^{J_0,1}_{\text{wave}}$ for $l \geq 0$ and $t \in \Lambda_l$. Then*

$$|\left\langle \phi^0_{l,t}, w_n \right\rangle|^2 = \begin{cases} 2^{-l} & \textit{if } n < 2^l \\ 0 & \textit{otherwise} \end{cases}$$

$$|\left\langle \phi^1_{l,t}, w_n \right\rangle|^2 = \begin{cases} 2^{-l} & \textit{if } 2^l \leq n < 2^{l+1} \\ 0 & \textit{otherwise} \end{cases} .$$

Proof. These equalities can be found in either [5] or [30]. ■

III.6.4 Proof of Proposition III.4.5, III.4.10 and III.4.11

Using the above results we are now able to determine the local coherences of $U = [B_{\text{wh}}, B^{J_0,\nu}_{\text{wave}}]$.

Proof of Proposition III.4.5. We use the bound found in Theorem III.6.7. Recall that $\mathbf{M} = [2^{J_0+1}, \ldots, 2^{J_0+r}]$ and $\mathbf{N} = [2^{J_0+1}, \ldots, 2^{J_0-1+r}, 2^{J_0+r+q}]$. For fixed $l \in \{1, \ldots, r\}$ and $k \in \{2, \ldots, r\}$ we have

$$\begin{aligned}\mu_{k,l} &= \max \left\{ |\left\langle \phi^s_{J_0-1+l,t}, w_n \right\rangle|^2 : \substack{N_{k-1} \leq n < N_k \\ t \in \Lambda_{J_0-1+l}, s \in \{0,1\}, \text{ if } l=1, \\ s=1 \text{ if } l>1} \right\} \\ &\lesssim 2^{-(J_0-1+k)} 2^{-|(J_0-1+l)-(J_0-1+k)|} \lesssim 2^{-J_0-k} 2^{-|l-k|}\end{aligned}$$

For $l \in \{1, \ldots, r\}$ and $k = 1$ we have $N_0 = 0$. This gives

$$\begin{aligned}\mu_{k,l} &= \max \left\{ |\left\langle \phi^s_{J_0-1+l,t}, w_n \right\rangle|^2 : \substack{0 \leq n < N_1 \\ t \in \Lambda_{J_0-1+l}, s \in \{0,1\}, \text{ if } l=1, \\ s=1 \text{ if } l>1} \right\} \\ &\lesssim 2^{-(J_0-1+l)} \lesssim 2^{-J_0-k} 2^{-|l-k|}\end{aligned}$$

■

Proof of Proposition III.4.10. Since both $B_{\text{wave}}^{J_0,1}$ and B_{wh} are orthonormal, $U = [B_{\text{wh}}, B_{\text{wave}}^{J_0,1}]$ is an isometry on $\ell^2(\mathbb{N})$ i.e. $U^*U = I \in \mathcal{B}(\ell^2(\mathbb{N}))$. Let $N = 2^k$ for some $k \in \mathbb{N}$ with $k \geq J_0 + 1$. Using Theorem III.6.8 we see that

$$P_N^{\perp} U P_N = \left\{ \left\langle \phi_{j,t}^s, w_n \right\rangle : {}^{\quad n \geq 2^k}_{\substack{s=1, J_0 \leq j < k, t \in \Lambda_j \\ s=0, j=J_0, t\in\Lambda_{J_0}}} \right\} = 0$$

which means that

$$\begin{aligned}(P_N U P_N)^*(P_N U P_N) &= ((P_N + P_N^{\perp})UP_N)^*((P_N + P_N^{\perp})UP_N) \\ &= (UP_N)^*(UP_N) = P_N U^* U P_N = I \in \mathbb{C}^{N\times N}.\end{aligned}$$

■

Proof of Proposition III.4.11. We use the bound found in Theorem III.6.8. Recall that $\mathbf{M} = \mathbf{N} = [2^{J_0+1}, \dots, 2^{J_0+r}]$. For fixed $k, l \in \{1, \dots, r\}$ we have that

$$\begin{aligned}\mu_{k,l} &= \max\left\{ \left| \left\langle \phi_{J_0-1+l,t}^s, w_n \right\rangle \right|^2 : \substack{N_{k-1} \leq n < N_k \\ t \in \Lambda_{J_0-1+l}, s\in\{0,1\}, \text{ if } l=1, \\ s=1 \text{ if } l>1} \right\} \\ &= \begin{cases} 2^{-J_0-l+1} & \text{if } l = k \\ 0 & \text{otherwise} \end{cases}.\end{aligned}$$

■

III.6.5 About the sharpness of the local coherence bounds

As can be seen from Proposition III.4.11, the coherence bounds for $\nu = 1$ are sharp. However, for $\nu \geq 2$, we have not discussed their sharpness. In fact, none of the results in this paper consider the case for $\nu = 2$ vanishing moments. The reason for this is that these wavelets have a Lipschitz regularity $\alpha \approx 0.55$, which means that the bound in Theorem III.6.7 would have less rapid decay if we included these wavelets in the theorem. To simplify the presentation, we have chosen to exclude them.

We will argue that Theorem III.6.7 does not seem to extend to wavelets with $\nu = 2$ vanishing moments. Let $\mathbf{M} = \mathbf{N} = [2^{J_0+1}, \dots, 2^{J_0+r}]$ and $U = [B_{\text{wh}}, B_{\text{wave}}^{J_0,\nu}]$ for $\nu \geq 2$. Notice that setting $\nu = 2$ does only affect the local coherence estimates $\mu_{k,l}$ for $k \geq l$. For $k < l$, the local coherences are unaffected by the regularity of the wavelet. This follows from Lemma III.6.5, by setting $|\mathcal{W}\phi^s(\cdot + l)(0)| \approx 1$. Next consider the case where $k \geq l$, then Theorem III.6.7 suggests that $\mu_{k,l}/\mu_{k+1} \approx 4$ for $\nu \geq 3$.

We now consider table III.2 and notice that for $\nu = 2$, all of the 18 entries in table III.2 have values less than 4. This suggests that the bound in Theorem III.6.7 does not extend to the case of $\nu = 2$ vanishing moments. From the same table we also observe that for $\nu = 4$, the bound in Theorem III.6.7 seem to be quite sharp. While there are a few entries that are less than 4, most are very close, if not larger than this value.

$\mu_{k,l}/\mu_{k+1,l}$	$l=1$	$l=2$	$l=3$
$k=2$	3.017		
$k=3$	2.532	1.854	
$k=4$	3.292	2.532	1.846
$k=5$	3.653	3.293	2.534
$k=6$	3.828	3.653	3.293
$k=7$	3.914	3.828	3.654
$k=8$	3.957	3.914	3.828

$\mu_{k,l}/\mu_{k+1,l}$	$l=1$	$l=2$	$l=3$
$k=2$	4.342		
$k=3$	6.160	3.439	
$k=4$	3.643	6.202	3.503
$k=5$	4.060	3.639	6.286
$k=6$	3.961	4.064	3.632
$k=7$	4.004	3.960	4.070
$k=8$	3.996	4.004	3.960

Table III.2: Left: Fraction between the local coherences for $U = [B_{\text{wh}}, B_{\text{wave}}^{3,2}]$ and $\mathbf{M} = \mathbf{N} = [2^4, \dots, 2^{11}]$. Right: Fraction between the local coherences for $U = [B_{\text{wh}}, B_{\text{wave}}^{4,4}]$ and $\mathbf{M} = \mathbf{N} = [2^5, \dots, 2^{12}]$.

III.6.6 Proof of remaining results in Section III.4

Proof of Proposition III.4.4. This proposition is a consequence of Theorem 1.1 in [20]. Let $\mathcal{S}_N = \{w_n : n = 0, \dots, N-1\}$ and $\mathcal{R}_M$ be the M first function in $B_{\text{wave}}^{J_0,\nu}$. The subspace cosine angle between $\mathcal{S}_N$ and $\mathcal{R}_M$ is defined as

$$\cos(\omega(\mathcal{R}_M, \mathcal{S}_N)) = \inf_{f \in \mathcal{R}_M, \|f\|=1} \|P_{\mathcal{S}_N} f\| \quad \text{where} \quad \omega(\mathcal{R}_M, \mathcal{S}_N) \in [0, \pi/2],$$

and $P_{\mathcal{S}_N}$ is the projection operator onto $\mathcal{S}_N$. As both B_{wh} and $B_{\text{wave}}^{J_0,\nu}$ are orthonormal bases, the synthesis and analysis operators are unitary. We therefore have

$$\inf_{f \in \mathcal{R}_M, \|f\|=1} \|P_{\mathcal{S}_N} f\| = \inf_{x \in \mathbb{C}^M, \|x\|_2=1} \|P_N U P_M x\|_2 \tag{III.68}$$

Furthermore notice that by equation (III.56) and the definition of the balancing property, we have

$$\cos(\omega(\mathcal{R}_M, \mathcal{S}_N)) = \inf_{x \in \mathbb{C}^M, \|x\|_2=1} \|P_N U P_M x\|_2 \geq \theta. \tag{III.69}$$

Hence if U satisfies the balancing property of order $\theta \in (0,1)$ for N and M, then $1/\cos(\omega(\mathcal{R}_M, \mathcal{S}_N)) \leq 1/\theta$, where $1/\theta > 1$. Next for $M \in \mathbb{N}$ and $\gamma > 1$ we define the *stable sampling rate* as

$$\Theta(M, \gamma) = \min(N \in \mathbb{N} : 1/\cos(\omega(\mathcal{R}_M, \mathcal{S}_N)) < \gamma).$$

Rearranging the terms we see that if N, M satisfies the stable sampling rate of order $\gamma = 1/\theta > 1$ then U satisfies the balancing property of order θ for N and M.

Theorem 1.1 in [20] states that for $M = 2^r$, $r \in \mathbb{N}$ and for all $\gamma > 1$ there exists a constant $S_\gamma > 1$ (dependent on γ), such that whenever $N \geq S_\gamma M$, then $1/\cos(\omega(\mathcal{R}_M, \mathcal{S}_N)) < \gamma$. Moreover, we have the relation $\Theta(M, \gamma) \leq S_\gamma M = \mathcal{O}(M)$. Hence if $q = \lceil \log_2 S_{1/\theta} \rceil$ we see that the proposition hold with $N = 2^{k+q} \geq S_{1/\theta} 2^k > 2^k = M$. ■

Proof of Proposition III.4.6. Using Theorem III.6.7, we see that $\mu(P_N U P_K^{\perp}) \lesssim K^{-1}$. This gives

$$\begin{aligned}\|HP_K^{\perp}x\|_2^2 &= \sum_{k=1}^{r} \frac{N_k - N_{k-1}}{m_k} \sum_{i\in\Omega_k} \Big| \sum_{j>K} U_{ij}x_j \Big|^2 \\ &\leq \sum_{k=1}^{r} \frac{N_k - N_{k-1}}{m_k} \sum_{i\in\Omega_k} \Big(\sum_{j>K} \sqrt{\mu(P_N U P_K^{\perp})} |x_j| \Big)^2 \\ &\leq \sum_{k=1}^{r} (N_k - N_{k-1}) \mu(P_N U P_K^{\perp}) \Big(\sum_{j>K} |x_j| \Big)^2 \\ &\leq N\mu(P_N U P_K^{\perp}) \Big(\sum_{j>K} |x_j| \Big)^2 \lesssim \frac{N}{K} \|x\|_1^2.\end{aligned}$$

■

Proof of Theorem III.4.7. First recall that $\mathbf{M} = [2^{J_0+1}, \dots, 2^{J_0+r}]$ and $\mathbf{N} = [2^{J_0+1}, \dots, 2^{J_0+r-1}, 2^{J_0+r+q}]$ where $q \geq 0$ is chosen so that G satisfies the balancing property of order $0 < \theta < 1$. From Lemma III.3.3 we therefore have $\|G^{-1}\|_2 \leq 1/\sqrt{\theta}$.

From Theorem III.3.6 we know that the matrix A in equation (III.16) satisfies the G-RIPL with $\delta_{\boldsymbol{s},\mathbf{M}} \leq \delta$, provided the sample densities $\boldsymbol{m} \in \mathbb{N}^r$ satisfies $m_k = N_k - N_{k-1}$ for $k = 1, \dots, r_0$, and

$$m_k \gtrsim \delta^{-2} \cdot \|G^{-1}\|_2^2 \cdot (N_k - N_{k-1}) \cdot \left(\sum_{l=1}^{r} \mu_{k,l} \cdot s_l \right) \cdot L, \tag{III.70}$$

for $k = r_0 + 1, \dots, r$. Next notice that $N_k - N_{k-1} = 2^{J_0+k-1}$ for $k = 2, \dots, r-1$, while $N_r - N_{r-1} = 2^{J_0+r}(2^q - 2^{-1})$ and $N_1 - N_0 = 2^{J_0+1}$. Using the local coherences $\mu_{k,l}$ from Proposition III.4.5 we obtain

$$\begin{aligned}(N_k - N_{k-1}) \left(\sum_{l=1}^{r} \mu_{k,l} s_l \right) &\lesssim 2^{J_0+k} 2^{q \max\{k+1-r,0\}} \left(\sum_{l=1}^{r} 2^{-J_0-k} 2^{-|l-k|} s_l \right) \\ &= 2^{q \max\{k+1-r,0\}} \left(\sum_{l=1}^{r} 2^{-|k-l|} s_l \right).\end{aligned}$$

Inserting this and $\|G^{-1}\|_2^2 \leq \theta^{-1}$ into (III.70) leads to the sampling condition in Theorem III.4.7. ■

Proof of Theorem III.4.8. The theorem is identical to Corollary III.3.7, except that we have fixed $\mathbf{M}$ and $\mathbf{N}$. The concrete values for these have been inserted in condition (iv) together with the local coherences $\mu_{k,l}$. The computation of this can be found in the proof above. ■

Acknowledgements

The authors would like to thank Simone Brugiapaglia, Simon Foucart, Remi Gribonval, Øyvind Ryan and Laura Thesing for useful discussions and comments. BA acknowledges support from the Natural Sciences and Engineering Research Council of Canada through grant 611675. ACH acknowledges support from the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/L003457/1, a Royal Society University Research Fellowship, and the Philip Leverhulme Prize (2017).

References

[1] Adcock, B., Boyer, C. and Brugiapaglia, S. ‘On oracle-type local recovery guarantees in compressed sensing’. In: *arXiv:1806.03789* (2018).

[2] Adcock, B. and Hansen, A. C. ‘Generalized sampling and infinite-dimensional compressed sensing’. In: *Foundations of Computational Mathematics* vol. 16, no. 5 (2016), pp. 1263–1323.

[3] Adcock, B., Hansen, A. C., Poon, C. and Roman, B. ‘Breaking the coherence barrier: A new theory for compressed sensing’. In: *Forum of Mathematics, Sigma*. Vol. 5. Cambridge University Press. 2017.

[4] Adcock, B., Hansen, A. C. and Roman, B. ‘A note on compressed sensing of structured sparse wavelet coefficients from subsampled Fourier measurements’. In: *IEEE Signal Processing Letters* vol. 23, no. 5 (2016), pp. 732–736.

[5] Antun, V. *Coherence estimates between hadamard matrices and daubechies wavelets*. Master’s thesis, *University of Oslo*. 2016.

[6] Arce, G. R., Brady, D. J., Carin, L., Arguello, H. and Kittle, D. S. ‘Compressive coded aperture spectral imaging: An introduction’. In: *IEEE Signal Processing Magazine* vol. 31, no. 1 (2014), pp. 105–115.

[7] Bastounis, A. and Hansen, A. C. ‘On the Absence of Uniform Recovery in Many Real-World Applications of Compressed Sensing and the Restricted Isometry Property and Nullspace Property in Levels’. In: *SIAM Journal on Imaging Sciences* vol. 10, no. 1 (2017), pp. 335–371.

[8] Brugiapaglia, S. and Adcock, B. ‘Robustness to unknown error in sparse regularization’. In: *IEEE Transactions on Information Theory* vol. 64, no. 10 (2018), pp. 6638–6661.

[9] Butzer, P. L. and Wagner, H. J. ‘On dyadic analysis based on the pointwise dyadic derivative’. In: *Analysis Mathematica* vol. 1, no. 3 (Sept. 1975), pp. 171–196.

[10] Candès, E. J., Romberg, J. and Tao, T. ‘Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information’. In: *IEEE Transactions Information Theory* vol. 52, no. 2 (2006), pp. 489–509.

[11] Chi, Y., Scharf, L. L., Pezeshki, A. and Calderbank, A. R. 'Sensitivity to basis mismatch in compressed sensing'. In: *IEEE Transactions on Signal Processing* vol. 59, no. 5 (2011), pp. 2182–2195.

[12] Cohen, A., Daubechies, I. and Vial, P. 'Wavelets on the interval and fast wavelet transforms'. In: *Applied and Computational Harmonic Analysis* vol. 1, no. 1 (1993), pp. 54–81.

[13] Daubechies, I. *Ten lectures on wavelets*. Vol. 61. SIAM, 1992.

[14] Donoho, D. L. 'Compressed sensing'. In: *IEEE Transactions Information Theory* vol. 52, no. 4 (2006), pp. 1289–1306.

[15] Eldar, Y. C. and Kutyniok, G. *Compressed sensing: theory and applications*. Cambridge University Press, 2012.

[16] Foucart, S. and Rauhut, H. *A mathematical introduction to compressive sensing*. Birkhauser, 2013.

[17] Gehm, M. E. and Brady, D. J. 'Compressive sensing in the EO/IR'. In: *Applied Optics* vol. 54, no. 8 (2015), pp. C14–C22.

[18] Golubov, B., Efimov, A. and Skvortsov, V. *Walsh series and transforms: theory and applications*. Vol. 64. Springer Science & Business Media, 1991.

[19] Guerquin-Kern, M., Lejeune, L., Pruessmann, K. P. and Unser, M. 'Realistic analytical phantoms for parallel magnetic resonance imaging'. In: *IEEE Transactions on Medical Imaging* vol. 31, no. 3 (2012), pp. 626–636.

[20] Hansen, A. C. and Thesing, L. 'On the stable sampling rate for binary measurements and wavelet reconstruction'. In: *Applied and Computational Harmonic Analysis* vol. 48, no. 2 (2020), pp. 630–654.

[21] Jones, A., Tamtögl, A., Calvo-Almazán, I. and Hansen, A. 'Continuous compressed sensing for surface dynamical processes with helium atom scattering'. In: *Scientific Reports* vol. 6:27776 (2016).

[22] Larson, P. E. et al. 'Fast dynamic 3D MR spectroscopic imaging with compressed sensing and multiband excitation pulses for hyperpolarized 13C studies'. In: *Magnetic Resonance in Medicine* vol. 65, no. 3 (2011), pp. 610–619.

[23] Li, C. and Adcock, B. 'Compressed sensing with local structure: uniform recovery guarantees for the sparsity in levels class'. In: *Applied and Computational Harmonic Analysis* vol. 46, no. 3 (2019), pp. 453–477.

[24] Lustig, M., Donoho, D. L., Santos, J. M. and Pauly, J. M. 'Compressed sensing MRI'. In: *IEEE Signal Processing Magazine* vol. 25, no. 2 (2008), pp. 72–82.

[25] Lustig, M., Donoho, D. and Pauly, J. M. 'Sparse MRI: The application of compressed sensing for rapid MR imaging'. In: *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* vol. 58, no. 6 (2007), pp. 1182–1195.

[26] Mallat, S. *A wavelet tour of signal processing: The sparse way.* Third. Academic Press, 2008.

[27] Roman, B., Hansen, A. C. and Adcock, B. 'On asymptotic structure in compressed sensing'. In: *arXiv:1406.4178* (2014).

[28] Strang, G. and Nguyen, T. *Wavelets and filter banks.* Wellesley-Cambridge Press, 1996.

[29] Studer, V. et al. 'Compressive Fluorescence Microscopy for Biological and Hyperspectral Imaging'. In: *Proceedings of the National Academy of Sciences* vol. 109, no. 26 (2011), pp. 1679–1687.

[30] Thesing, L. and Hansen, A. C. 'Linear reconstructions and the analysis of the stable sampling rate'. In: *Sampling Theory in Signal and Image Processing* (2018).

[31] Traonmilin, Y. and Gribonval, R. 'Stable recovery of low-dimensional cones in Hilbert spaces: One RIP to rule them all'. In: *Applied and Computational Harmonic Analysis* vol. 45, no. 1 (2018), pp. 170–205.

[32] Willett, R. M., Marcia, R. F. and Nichols, J. M. 'Compressed sensing for practical optical imaging systems: a tutorial'. In: *Optical Engineering* vol. 50, no. 7 (2011).

[33] Zomet, A. and Nayar, S. K. 'Lensless imaging with a controllable aperture'. In: *Computer Society Conference on Computer Vision and Pattern Recognition.* Vol. 1. IEEE. 2006, pp. 339–346.

Paper IV

On the unification of schemes for wavelets on the interval

Vegard Antun, Øyvind Ryan

Submitted for publication in *Acta Applicandae Mathematicae*.

Abstract

We revisit the construction of wavelets on the interval with various degrees of polynomial exactness and present a unified scheme for constructing these, applicable to compactly supported delay-normalized wavelets. This extends previous constructions applicable to orthogonal- or Spline wavelets. It is explained how the corresponding discrete wavelet transform can be made flexible w.r.t. the length of the input and the method of stable completion is restated in the unified scheme. The contribution differs substantially from previous ones in how results are stated and deduced: linear algebra notation is exploited more heavily, and the use of sums and complex index notation is reduced. This extended use of linear algebra eases translation to software, and a general open source implementation that exploits the presented results has been developed.

IV.1 Introduction

Wavelets on the interval are well studied, and several constructions exist for various degrees of polynomial exactness at the primal and dual sides [1, 5, 7, 8, 13, 15]. Their practical use has been limited, however: Software involving wavelet transforms typically abandon polynomial exactness in favour of simpler extension strategies at the boundaries, such as periodic or symmetric extensions [22]. One reason for this is that polynomial exactness on the interval only may reduce spikes in wavelet coefficients near the boundaries. Therefore, little is obtained in terms of compression, one of the main applications of wavelets. Also, to the best of the authors knowledge, there does not exist openly available software handling such polynomial exactness on the interval[1].

Keywords: Wavelets, wavelets on the interval, boundary wavelets, polynomial exactness
Mathematics Subject Classification (2010): 42C40, 65T60

[1] There do, however, exist a few implementations which support a limited set of wavelets (such as certain orthogonal Daubechies wavelets), with coefficients precomputed from [7], see also [3, 11, 12]. The code for computing these coefficients are, however, not available.

This contribution is an attempt to address this lack of software, and it is therefore closely tied to an open source implementation. In terms of linear algebra and notation (see Section IV.1.1) the proofs are largely rewritten from those commonly found in the literature to be minimalistic, so that translation to software is straightforward. While many results in the literature restrict to specific cases such as the orthogonal- or Spline cases, we show that the deductions make sense for much more general compactly supported wavelets as well – one only needs to assume delay-normalized wavelets (see Section IV.2), and a common centre of support for the scaling function and its dual counterpart. Also, the method of stable completion [4] can be generalized to this setting. The deductions can be thought of as an extension of the Spline-based deductions from [8] to the delay-normalized case, with substantial changes to the notation.

We remark that [16] establishes Spline wavelets on the interval in an improved way, by keeping the primal boundary functions fixed, only modifying the dual boundary functions. We translate the main results from [16] to the delay-normalized setting as well. [16] makes the point that staggered supports (see Section IV.4.1) seem to produce wavelet bases with favourable Riesz bounds. The software implementation takes this into account. It is, however, straightforward to modify this not to use staggered supports[2], or use other bases (see [9] for recommendations).

Although polynomial exactness at the boundaries may give little extra in terms of applications, the software implementation hopefully can serve as a playground for researchers experimenting with wavelets. Exploiting some flexibility noted in [8], the implementation also opens for discrete wavelet transforms on the interval applicable to any input size and number of resolutions. This flexibility comes at the cost of numerical stability, however, so that putting constraints on the length of the input is natural. (such constraints are implicit in contributions such as [7, 16], since they operate with fixed boundary functions at the primal side: As an example, an m-level DWT as defined in [7] is possible only if the vector length is divisible with 2^m).

IV.1.1 Notation

The paper follows notation in [19, 20], which introduce the reader to signal processing and wavelets in a linear algebra friendly way, and in a style very different from that common for wavelets. The books also use the same software implementation and actually build it from scratch. The *interval notation* $[a, b] = \{a, a+1, \ldots, b\}$ will be used to denote all integers between the two integers a and b. If $b < a$, $[a, b] = \emptyset$. Similarly, $[a, b)$ denotes the set $\{a, a+1, \ldots, b-1\}$. Furthermore, for $k \in \mathbb{Z}$, one defines in the obvious way

$$k[a, b] = \{ka, k(a+1), \ldots, kb\} \qquad k + [a, b] = \{k+a, k+a+1, \ldots, k+b\}.$$

These can also be combined, i.e., for $k_1, k_2 \in \mathbb{Z}$, one has

$$k_1 + k_2[a, b] = \{k_1 + k_2 a, k_1 + k_2(a+1), \ldots k_1 + k_2 b\}.$$

[2] Note, however, that stable completions are made in the setting of staggered supports

This notation will be used to refer to segments of matrices and vectors. It should be clear from the context whether a range of integers, or an actual interval on the real line, is meant. This notation will eliminate much of the extensive indexing in wavelet literature. In particular, it will simplify referring to segments of the DWT/IDWT matrices, as will often be needed. In the literature, a DWT/IDWT is often expressed in terms of the filter coefficients, since these represent all entries in those matrices. This brings one away from simple matrix-vector expressions, and our deductions will, therefore, avoid this.

Wavelet bases for $L^2(\mathbb{R})$ contain an infinite number of basis functions at each resolution, whereas wavelet bases on the interval have finitely many. It will therefore be convenient to mix notation for finite and infinite matrices, and allowing finite matrices and vectors to have any given legal range of row- and column indices. In particular, in an expression on the form

$$\begin{pmatrix} \phi^b_{0,0} \\ \vdots \\ \phi^b_{0,N-1} \end{pmatrix} = C^T \begin{pmatrix} \phi_{0,-R+1}|_{[0,\infty)} \\ \vdots \\ \phi_{0,N-1}|_{[0,\infty)} \end{pmatrix},$$

it will be assumed that the column vector on the left hand side has row indices $[0, N-1]$, and that the column vector on the right hand side has row indices $[-R+1, N-1]$. The matrix C can be any infinite matrix, but when written as above, it is assumed that the range of column- and row indices in C are $[0, N-1] \times [-R+1, N-1]$, i.e., that the indices match. Since any range of row- and column-indices may be legal, entries with index 0 or $(0,0)$ will occasionally be underlined (as in filter notation in signal processing), to make positions clear. The MatLab notation that a simple colon denotes all elements along an axis, will also be followed.

IV.1.2 Organization of the paper

The paper is organized as follows. In Section IV.2 the general setup for wavelets is introduced, and in Section IV.3 the setup is specialized to the interval. In sections IV.4 and IV.5 the scaling functions and the corresponding mother wavelets are constructed. While those sections were adapted to the left end of the interval, Section IV.6 explains how delay-normalizedness ensures that the construction at the right end can be obtained from a simple mirroring operation of the left end. In Section IV.7 the result in [16] are put into the context of this contribution. Some notes on the software implementation can be found in Section IV.8. A more detailed explanation of this implementation can be found in the technical report [2].

IV.2 Setup for wavelets on the entire real line

Let ϕ and ψ be the scaling function and the mother wavelet of a compactly supported wavelet. Assume also that ϕ is *exact* of order N (meaning that all polynomials of degree less than N can be written as linear combinations of the translates $\{\phi(t-n)\}_n$). Similarly let $\tilde{\phi}$, $\tilde{\psi}$, and $\tilde{N}$ be the corresponding quantities for the *dual wavelet.* The *resolution space* V_0 is the space spanned by the translates $\phi_{0,n}(t) = \phi(t-n)$, while the *detail space* W_0 is the space spanned by $\psi_{0,n}(t) = \psi(t-n)$. For $m > 0$, the resolution- and detail spaces V_m and W_m are the spaces spanned by the dilated functions

$$\phi_{m,n}(t) = 2^{m/2}\phi_{0,n}(2^m t) \qquad\qquad \psi_{m,n}(t) = 2^{m/2}\psi_{0,n}(2^m t), \tag{IV.1}$$

respectively. Similar definitions apply for $\tilde{\phi}$ and $\tilde{\psi}$. One also writes

$$\boldsymbol{\phi}_m = \{\phi_{m,n}\}_{n=-\infty}^{\infty} \qquad\qquad \boldsymbol{\psi}_m = \{\psi_{m,n}\}_{n=-\infty}^{\infty},$$

so that $V_m = \operatorname{span}(\boldsymbol{\phi}_m)$ and $W_m = \operatorname{span}(\boldsymbol{\psi}_m)$. When ϕ gives rise to a *multiresolution analysis* the V_m are nested (i.e., $V_m \subset V_{m+1}$), and $V_m = V_{m-1} \oplus W_{m-1}$, so that

$$\mathcal{C}_m = \{\phi_{m-1,n}, \psi_{m-1,n}\}_{n=-\infty}^{\infty}$$

(i.e., where the dilated scaling functions and mother wavelets are listed in alternating order) is also a basis for V_m. This alternating order of the basis functions is non-standard in wavelet literature, where all $\phi_{m-1,n}$-functions usually preceed the $\psi_{m-1,n}$. This reordering has the advantage that the index n into the basis $\mathcal{C}_m$ represents time, and that change of coordinate matrices involving those bases will be banded.

On the dual side one similarly defines $\tilde{\boldsymbol{\phi}}_m$, $\tilde{\boldsymbol{\psi}}_m$, $\tilde{V}_m$, $\tilde{W}_m$, and $\tilde{\mathcal{C}_m}$. The *Gramm matrix* of two bases $\mathcal{B} = \{\boldsymbol{b}_i\}_i$ and $\mathcal{C} = \{\boldsymbol{c}_j\}_j$, denoted $(\langle\mathcal{B},\mathcal{C}\rangle)$, is the matrix with entries $\langle \boldsymbol{b}_i, \boldsymbol{c}_j\rangle$. If $(\langle\mathcal{B},\mathcal{C}\rangle) = I$ one also says that $\mathcal{B}$ and $\mathcal{C}$ are *biorthogonal.* A wavelet is called *biorthogonal* if the corresponding bases are biorthogonal, i.e., $(\langle\boldsymbol{\phi}_m, \tilde{\boldsymbol{\phi}}_m\rangle) = (\langle\mathcal{C}_m, \tilde{\mathcal{C}_m}\rangle) = I$. Some of the most used *biorthogonal wavelets* were established in [6]. Some of the most used *orthonormal wavelets*, for which $\phi = \tilde{\phi}$, $\psi = \tilde{\psi}$, and $(\langle\boldsymbol{\phi}_m, \boldsymbol{\phi}_m\rangle) = (\langle\mathcal{C}_m, \mathcal{C}_m\rangle) = I$ (i.e., both $\boldsymbol{\phi}_m$ and $\mathcal{C}_m$ are orthonormal bases for V_m) were established in [10]. Denoting by $\operatorname{supp}(f)$ the support interval of the function f, a convention therein is that

$$\operatorname{supp}(\phi) = \operatorname{supp}(\psi) = [-N+1, N]. \tag{IV.2}$$

The change of coordinates from $\boldsymbol{\phi}_m$ to $\mathcal{C}_m$ is called the (one-level) Discrete Wavelet Transform, or DWT, and denoted H (i.e., $H = P_{\mathcal{C}_m \leftarrow \boldsymbol{\phi}_m}$). Its inverse is the IDWT, denoted by G (i.e., $G = P_{\boldsymbol{\phi}_m \leftarrow \mathcal{C}_m}$), and can be written as

$$G = \begin{pmatrix} \cdots & \underline{[\phi_{m-1,0}]}_{\boldsymbol{\phi}_m} & [\psi_{m-1,0}]_{\boldsymbol{\phi}_m} & [\phi_{m-1,1}]_{\boldsymbol{\phi}_m} & [\psi_{m-1,1}]_{\boldsymbol{\phi}_m} & \cdots \end{pmatrix}. \tag{IV.3}$$

Since the bases here are doubly infinite, the component with index zero is emphasized by underlining it, i.e., the coordinate vector of $f(t) = c_{-1}\phi_{0,-1} + c_0\phi_{0,0} + c_1\phi_{0,1}$ in $\boldsymbol{\phi}_0$ will be written as $[f]_{\boldsymbol{\phi}_0} = (c_{-1}, \underline{c_0}, c_1)$. Coefficients which are zero were not listed here, as is common in signal processing filter notation.

H and G can be expressed in terms of filters as follows [19, 20, Chapter 3]:

1. The even-indexed rows of H coincide with those of a (low-pass) filter matrix, denoted H_0.
2. The odd-indexed rows of H coincide with those of a (high-pass) filter matrix, denotes H_1.
3. The even-indexed columns of G coincide with those of a (low-pass) filter matrix, denoted G_0.
4. The odd-indexed columns of G coincide with those of a (high-pass) filter matrix, denoted G_1.

Thus, H can be alternatively defined as the unique matrix compatible with filters H_0 and H_1, and G as the unique matrix compatible with filters G_0 and G_1. It is known (Exercise 5.10 in [19, 20]) that if the filters of a wavelet are finite impulse response (FIR), then there exist an integer d and $\alpha \in \mathbb{R}$ so that

$$(H_1)_n = (-1)^n \alpha^{-1} (G_0)_{n-2d} \qquad (G_1)_n = (-1)^n \alpha (H_0)_{n+2d}. \tag{IV.4}$$

Since the alternating sign corresponds to a shift in frequency by π, this says that, up to multiplication with a scalar,

1. H_1 is the high-pass filter corresponding to the low-pass filter G_0,
2. G_1 is the high-pass filter corresponding to the low-pass filter H_0.

When $d = 0$ in (IV.4), (ϕ, ψ) is said to be *delay-normalized* [22]. Clearly there is no loss in generality in assuming this, as changing d simply reorders the mother wavelet basis functions with a shift. Delay-normalized wavelets will be assumed in the following, as this will simplify some proofs. Wavelets with symmetric filters are clearly delay-normalized.

The dual wavelet transforms, denoted by $\tilde{H}$ and $\tilde{G}$, are the matrices compatible with the filters, $\tilde{H}_0 = G_0^T$ and $\tilde{H}_1 = G_1^T$, and $\tilde{G}_0 = H_0^T$ and $\tilde{G}_1 = H_1^T$. Let $[L, R] = \text{supp}(\phi)$, and $[\tilde{L}, \tilde{R}] = \text{supp}(\tilde{\phi})$ denote the left and right supports of ϕ and $\tilde{\phi}$. Defining the support of a filter as the smallest interval containing the nonzero filter indices, one has that $\text{supp}(G_0) = \text{supp}(\phi) = [L, R]$, and $\text{supp}(\tilde{G}_0) = \text{supp}(\tilde{\phi}) = [\tilde{L}, \tilde{R}]$. When the wavelet is delay-normalized one has that

$$\begin{aligned} \text{supp}(G_1) &= \text{supp}(H_0) = \text{supp}(\tilde{G_0}^T) = [-\tilde{R}, -\tilde{L}] \\ \text{supp}(\tilde{G}_1) &= \text{supp}(H_1^T) = \text{supp}(G_0^T) = [-R, -L]. \end{aligned}$$

These formulas tell us which scaling functions at scale 1 contribute in ψ and $\tilde{\psi}$, a fact which will be useful. It is straightforward to find the supports of the mother wavelets from the supports of the filters (see for instance Exercise 5.16 in [19, 20]). In particular, a delay-normalized wavelet can be recognized in terms of the supports by the requirement

$$\text{supp}(\psi) = [(L - \tilde{R} + 1)/2, (R - \tilde{L} + 1)/2]. \tag{IV.5}$$

Clearly $(\phi_{0,n}, \psi)$, as well as $(\tilde{\phi}_{0,n}, \tilde{\psi})$, are also delay-normalized for any n, as translating ϕ and $\tilde{\phi}$ with the same n gives scaling functions for a new biorthogonal wavelet.

For an orthonormal wavelet the filters and the dual filters equal, and H is orthogonal. From the deductions above, one sees that $\text{supp}(G_0) = \text{supp}(G_1^T)$, in order for an orthonormal wavelet to be delay-normalized. It is straightforward to check that assumption (IV.2) implies that (IV.5) holds, so that this support assumption guarantees a delay-normalizedness.

IV.3 Setup for wavelets on the interval

When restricting to an interval of the form $[0, M]$, the wavelet bases $\boldsymbol{\phi}_m$, $\boldsymbol{\psi}_m$, and the functions $\phi_{m,n}$, $\psi_{m,n}$, will be replaced with new bases $\boldsymbol{\phi}_m^b$, $\boldsymbol{\psi}_m^b$, and modified functions $\phi_{m,n}^b$, $\psi_{m,n}^b$. Here b is short for boundary, and only those functions supported near the boundaries are modified (called *left- and right edge functions*). Initial candidates for the left edge scaling functions will first be defined. It will then be seen how changes of coordinates can be applied to make those functions orthonormal/biorthogonal. The right edge functions will be obtained by repeating the left edge analysis, following a mirroring operation. The following definition is a generalization of that in [7].

Definition IV.3.1 (Initial left edge functions)**.** Let $\{c_k\}_{k=0}^{N-1}$ and $\{\tilde{c}_k\}_{k=0}^{\tilde{N}-1}$ be bases for the polynomials of degree at most $N-1$ and $\tilde{N}-1$, respectively, and let $K \geq \max(-L, N)$, $\tilde{K} \geq \max(-\tilde{L}, \tilde{N})$ be integers so that $N - K = \tilde{N} - \tilde{K}$. The *initial left edge scaling* functions are defined on $[0, \infty)$ by

$$\phi_{0,k}^b(t) = \begin{cases} \sum_{n=-R+1}^{K-1} c_k(n)\phi_{0,n}(t) & \text{for } 0 \leq k < N \\ \phi_{0,k+K-N}(t) & \text{for } N \leq k \end{cases} \tag{IV.6}$$

$$\tilde{\phi}^b{}_{0,k}(t) = \begin{cases} \sum_{n=-\tilde{R}+1}^{\tilde{K}-1} \tilde{c}_k(n)\tilde{\phi}_{0,n}(t) & \text{for } 0 \leq k < \tilde{N} \\ \tilde{\phi}_{0,k+\tilde{K}-\tilde{N}}(t) & \text{for } \tilde{N} \leq k, \end{cases} \tag{IV.7}$$

and the sets $\boldsymbol{\phi}_0^b = \{\phi_{0,k}^b\}_{k\geq 0}$ and $\tilde{\boldsymbol{\phi}}_0^b = \{\tilde{\phi}^b{}_{0,k}\}_{k\geq 0}$. V_0^b and $\tilde{V}_0^b$ will denote the spaces spanned by $\boldsymbol{\phi}_0^b$ and $\tilde{\boldsymbol{\phi}}_0^b$, respectively.

Some comments are in order.

- The first part of these functions are replacements of the $\{\phi_{0,k}\}_{k=K-N}^{K-1}$. Moreover, $\text{span}\left(\{\phi_{0,k}^b\}_{k\geq 0}\right)$ is independent of the choice of $\{c_k\}_{k=0}^{N-1}$, and will contain all polynomials of degree $< N$ on $(0, \infty)$. This follows since $\sum_{n=-\infty}^{\infty} c_k(n)\phi_{0,n}$ is a polynomial on $(-\infty, \infty)$, and its restriction to $(0, \infty)$ can be written as

$$\phi_{0,k}^b + \sum_{n\geq K} c_k(n)\phi_{0,n} = \phi_{0,k}^b + \sum_{n\geq N} c_k(n+K-N)\phi_{0,n}^b \in \text{span}\left(\{\phi_{0,k}^b\}_{k\geq 0}\right).$$

- The second part of these functions are translates of the scaling function, all supported on $[0,\infty)$ since $K \geq -L$, $\tilde{K} \geq -\tilde{L}$. They are called *internal functions*. The index shift $K-N = \tilde{K}-\tilde{N}$ is present for technical reasons: In order to compute an m-level DWT on the interval, we will see that K must be chosen accordingly. The $\{\phi_{0,k}\}_{k=0}^{K-N-1}$ in $\boldsymbol{\phi}_0$ have no counterpart in $\boldsymbol{\phi}_0^b$. This is reflected in the IDWT matrix in that rows are shifted $K-N$ entries downwards. However, $K-N$ basis functions will be added later, so that the net effect is that there is no shift.

- $N-K = \tilde{N}-\tilde{K}$ secures the same alignment of the shifted basis functions in $\boldsymbol{\phi}_0^b$ and $\tilde{\boldsymbol{\phi}}_0^b$, as in $\boldsymbol{\phi}_0$ and $\tilde{\boldsymbol{\phi}}_0$. One has flexibility in choosing K and $\tilde{K}$.

- $\{\phi_{0,n}^b\}_{n\geq N}$ can be expressed in terms of $\{\phi_{1,n}^b\}_{n\geq N}$, so that the internal functions inherit a refinability relation. This follows since $\phi_{0,N}^b = \phi_{0,K}$ can be expressed in terms of $\{\phi_{1,n}\}_{n\geq 2K+L} = \{\phi_{1,n}^b\}_{n\geq N+K+L}$, and since $K+L\geq 0$.

- $\phi_{m,n}^b$ are defined from $\phi_{0,n}^b$ using (IV.1) for $m>0$. Bases $\boldsymbol{\phi}_m^b$ and spaces V_m^b are defined similarly.

Let C be the matrix with entries $C_{n,k} = c_k(n)$ for $(n,k) \in [-R+1,K) \times [0,N)$, and $\tilde{C}$ the matrix with entries $\tilde{C}_{k,n} = \tilde{c}_k(n)$ for $(n,k) \in [-\tilde{R}+1,\tilde{K}) \times [0,\tilde{N})$. Definition IV.3.1 says that

$$\begin{pmatrix} \phi_{0,0}^b \\ \vdots \\ \phi_{0,N-1}^b \end{pmatrix} = C^T \begin{pmatrix} \phi_{0,-R+1}|_{[0,\infty)} \\ \vdots \\ \phi_{0,K-1}|_{[0,\infty)} \end{pmatrix} \quad \text{and}$$
$$\begin{pmatrix} \tilde{\phi}^b{}_{0,0} \\ \vdots \\ \tilde{\phi}^b{}_{0,\tilde{N}-1} \end{pmatrix} = \tilde{C}^T \begin{pmatrix} \tilde{\phi}_{0,-\tilde{R}+1}|_{[0,\infty)} \\ \vdots \\ \tilde{\phi}_{0,\tilde{K}-1}|_{[0,\infty)} \end{pmatrix}. \tag{IV.8}$$

C clearly has linearly independent columns, and thus full rank N. Any N rows of C give a nonsingular matrix since any polynomial of degree $N-1$ is uniquely identified from N distinct points.

Lemma IV.3.2. *The $\{\phi_{0,k}^b\}_{k=0}^{N-1}$ are*

1. *linearly independent on $[0,\infty)$, and linearly independent from the $\{\phi_{0,k}^b\}_{k\geq N}$,*

2. *orthogonal to the $\{\tilde{\phi}^b{}_{0,k}\}_{k\geq\tilde{N}}$,*

3. *orthogonal to the $\{\tilde{\psi}_{0,k}\}_k$ supported on $[0,\infty)$.*

Proof. If $\{\phi_{0,k}^b\}_{k=0}^{N-1}$ are linearly dependent on $[0,\infty)$ there must exist a non-zero linear combination so that $\sum_{k=0}^{N-1}\alpha_k\phi_{0,k}^b(t) = 0$ for all $t\in[0,\infty)$. Letting $\boldsymbol{\alpha}$ be the column vector with entries α_i, and using (IV.8) one gets

$$\boldsymbol{\alpha}^T \begin{pmatrix} \phi^b_{0,0} \\ \vdots \\ \phi^b_{0,N-1} \end{pmatrix} = \boldsymbol{\alpha}^T C^T \begin{pmatrix} \phi_{0,-R+1}|_{[0,\infty)} \\ \vdots \\ \phi_{0,K-1}|_{[0,\infty)} \end{pmatrix} = (C\boldsymbol{\alpha})^T \begin{pmatrix} \phi_{0,-R+1}|_{[0,\infty)} \\ \vdots \\ \phi_{0,K-1}|_{[0,\infty)} \end{pmatrix}.$$

Now, since $\{\phi_{0,n}|_{[0,\infty)}\}_{n=-R+1}^{K-1}$ are linearly independent on $[0,\infty)$, it is clear that $C\boldsymbol{\alpha} = \mathbf{0}$. But since C has linearly independent columns it follows that $\boldsymbol{\alpha} = \mathbf{0}$, so that the $\{\phi^b_{0,k}\}_{k=0}^{N-1}$ are linearly dependent as well. 1. now follows from the obvious fact that $\{\phi^b_{0,k}\}_{k=0}^{N-1}$ and $\{\phi_{0,k}\}_{k\geq K} = \{\phi^b_{0,k}\}_{k\geq N}$ are linearly independent on $[0,\infty)$. 2. and 3. follow also easily, since $\{\tilde{\phi^b}_{0,k}\}_{k\geq\tilde{N}}$ are supported on $[0,\infty)$. ■

It is known that the modified edge functions satisfy a refinement relation (This fact will be reproved in our setting in the following), so that the new spaces V_m^b also give rise to a multiresolution analysis. One can thus define change of coordinate matrices H^b and G^b as before, replacing the counterparts on the entire real line. If the first N scaling functions/mother wavelets need modification, (IV.3) translates to

$$G^b = \left([\phi^b_{0,0}]_{\boldsymbol{\phi}^b_1} \quad [\psi^b_{0,0}]_{\boldsymbol{\phi}^b_1} \quad \cdots \quad [\phi^b_{0,N-1}]_{\boldsymbol{\phi}^b_1} \quad [\psi^b_{0,N-1}]_{\boldsymbol{\phi}^b_1} \quad [\phi^b_{0,N}]_{\boldsymbol{\phi}^b_1} \quad [\psi^b_{0,N}]_{\boldsymbol{\phi}^b_1} \quad \cdots \right),$$

with all but the last two listed functions being modified versions. Since the unmodified functions inherit known refinability relations, the two columns listed last above are known. For the first columns one will write

$$\left([\phi^b_{0,0}]_{\boldsymbol{\phi}^b_1} \quad [\psi^b_{0,0}]_{\boldsymbol{\phi}^b_1} \quad \cdots \quad [\phi^b_{0,N-1}]_{\boldsymbol{\phi}^b_1} \quad [\psi^b_{0,N-1}]_{\boldsymbol{\phi}^b_1} \right) = \begin{pmatrix} X \\ Z \end{pmatrix},$$

with X representing the contribution of the modified functions, Z that of the internal functions, i.e.

$$\begin{pmatrix} \phi^b_{0,0} \\ \psi^b_{0,0} \\ \vdots \\ \phi^b_{0,N-1} \\ \psi^b_{0,N-1} \end{pmatrix} = X^T \begin{pmatrix} \phi^b_{1,0} \\ \vdots \\ \phi^b_{1,N-1} \end{pmatrix} + Z^T \begin{pmatrix} \phi^b_{1,N} \\ \vdots \end{pmatrix}. \tag{IV.9}$$

Denoting the even- and odd-indexed columns in X and Z, by X_e, Z_e and X_o, Z_o (throughout the paper the letters e and o will indicate even and odd indices), respectively, (IV.9) is equivalent to

$$\begin{pmatrix} \phi^b_{0,0} \\ \vdots \\ \phi^b_{0,N-1} \end{pmatrix} = (X_e)^T \begin{pmatrix} \phi^b_{1,0} \\ \vdots \\ \phi^b_{1,N-1} \end{pmatrix} + (Z_e)^T \begin{pmatrix} \phi^b_{1,N} \\ \vdots \end{pmatrix} \tag{IV.10}$$

$$\begin{pmatrix} \psi^b_{0,0} \\ \vdots \\ \psi^b_{0,N-1} \end{pmatrix} = (X_o)^T \begin{pmatrix} \phi^b_{1,0} \\ \vdots \\ \phi^b_{1,N-1} \end{pmatrix} + (Z_o)^T \begin{pmatrix} \phi^b_{1,N} \\ \vdots \end{pmatrix}. \tag{IV.11}$$

Since the left edge functions span the same space, regardless of the choice of polynomials, it makes sense to consider changes of coordinates between different candidates for edge functions. One can apply several such coordinate changes, in order to obtain functions $\phi^b_{0,n}$ with desired properties. The following result concerns how X_e, Z_e, and C are updated by such coordinate changes (note that Lemma IV.3.2 guarantees the uniqueness of X_e and Z_e in a factorization of the form (IV.10)).

Lemma IV.3.3. *Assume that a change of coordinates is applied to the left edge functions. Let $\{\phi^{b,1}_{0,k}\}_{k=0}^{N-1}$ and $\{\phi^{b,2}_{0,k}\}_{k=0}^{N-1}$ be bases, $P = P_{\{\phi^{b,1}_{0,k}\}\leftarrow\{\phi^{b,2}_{0,k}\}}$ the change of coordinate matrix from the second to the first basis. If (IV.13) holds for $\{\phi^{b,1}_{0,k}\}_{k=0}^{N-1}$, then (IV.13) also holds for $\{\phi^{b,2}_{0,k}\}_{k=0}^{N-1}$, and the change of coordinates from $\{\phi^{b,1}_{0,k}\}_{k=0}^{N-1}$ to $\{\phi^{b,2}_{0,k}\}_{k=0}^{N-1}$ transforms X_e, Z_e, and C according to*

$$X_e \to P^{-1}X_eP \qquad Z_e \to Z_eP \qquad C \to CP. \tag{IV.12}$$

Proof. One has that

$$\begin{aligned}
\begin{pmatrix} \phi^{b,2}_{0,0} \\ \vdots \\ \phi^{b,2}_{0,N-1} \end{pmatrix} &= P^T \begin{pmatrix} \phi^{b,1}_{0,0} \\ \vdots \\ \phi^{b,1}_{0,N-1} \end{pmatrix} = P^T \left(X_e^T \begin{pmatrix} \phi^{b,1}_{1,0} \\ \vdots \\ \phi^{b,1}_{1,N-1} \end{pmatrix} + Z_e^T \begin{pmatrix} \phi^b_{1,N} \\ \vdots \\ \phi^b_{1,K+R+N-2} \end{pmatrix} \right) \\
&= P^T \left(X_e^T (P^{-1})^T \begin{pmatrix} \phi^{b,2}_{1,0} \\ \vdots \\ \phi^{b,2}_{1,N-1} \end{pmatrix} + Z_e^T \begin{pmatrix} \phi_{1,K} \\ \vdots \\ \phi_{1,2K+R-2} \end{pmatrix} \right) \\
&= (P^{-1}X_eP)^T \begin{pmatrix} \phi^{b,2}_{1,0} \\ \vdots \\ \phi^{b,2}_{1,N-1} \end{pmatrix} + (Z_eP)^T \begin{pmatrix} \phi^b_{1,N} \\ \vdots \\ \phi^b_{1,K+R+N-2} \end{pmatrix}.
\end{aligned}$$

One also has that

$$\begin{aligned}
\begin{pmatrix} \phi^{b,2}_{0,0} \\ \vdots \\ \phi^{b,2}_{0,N-1} \end{pmatrix} &= P^T \begin{pmatrix} \phi^{b,1}_{0,0} \\ \vdots \\ \phi^{b,1}_{0,N-1} \end{pmatrix} = P^T C^T \begin{pmatrix} \phi_{0,-R+1}|_{[0,\infty)} \\ \vdots \\ \phi_{0,K-1}|_{[0,\infty)} \end{pmatrix} \\
&= (CP)^T \begin{pmatrix} \phi_{0,-R+1}|_{[0,\infty)} \\ \vdots \\ \phi_{0,K-1}|_{[0,\infty)} \end{pmatrix}.
\end{aligned}$$

(IV.12) follows. ∎

IV.4 Finding the left edge scaling functions

First the refinement relations satisfied by the modified edge functions is established.

Lemma IV.4.1. *For each choice of polynomial basis* $\{c_k\}_{k=0}^{N-1}$ *one has that*

$$\begin{pmatrix} \phi^b_{0,0} \\ \vdots \\ \phi^b_{0,N-1} \end{pmatrix} = (X_e)^T \begin{pmatrix} \phi^b_{1,0} \\ \vdots \\ \phi^b_{1,N-1} \end{pmatrix} + (Z_e)^T \begin{pmatrix} \phi^b_{1,N} \\ \vdots \\ \phi^b_{1,K+R+N-2} \end{pmatrix}, \tag{IV.13}$$

with

$$X_e = C^\dagger G_{I_X} C \qquad\qquad Z_e = G_{I_Z} C, \tag{IV.14}$$

where

- X_e *has indices from* $[0, N) \times [0, N)$,
- Z_e *has indices from* $[N, K + R + N - 2] \times [0, N)$,
- $I_X = [-R + 1, K) \times 2[-R + 1, K)$,
- $I_Z = [K, 2K + R - 2] \times 2[-R + 1, K)$,

and where $C^\dagger$ *is the generalized inverse of* C. X_e *is nonsingular.*

Here it is assumed that $C^\dagger$ has row indices equal to the column indices of C, and vice versa.

Proof. The first part of this proof corresponds to Lemma 3.1 in [8]. Since V_0^b contains all polynomials of degree less than N, C can be chosen so that

$$\phi^b_{0,k}(t) + \sum_{n \geq K} c_k(n)\phi_{0,n}(t) = t^k$$

on $[0, \infty)$. Inserting $2t$ for t and multiplying with $\sqrt{2}$ one also has

$$\phi^b_{1,k}(t) + \sum_{n \geq K} c_k(n)\phi_{1,n}(t) = \sqrt{2}(2t)^k.$$

Comparing these and using matrix notation one sees that

$$\begin{pmatrix} \phi^b_{0,0} \\ \vdots \\ \phi^b_{0,N-1} \end{pmatrix} + C^T \begin{pmatrix} \phi_{0,K} \\ \vdots \end{pmatrix} = D \left(\begin{pmatrix} \phi^b_{1,0} \\ \vdots \\ \phi^b_{1,N-1} \end{pmatrix} + C^T \begin{pmatrix} \phi_{1,K} \\ \vdots \end{pmatrix} \right), \tag{IV.15}$$

where D is $N \times N$ and diagonal with $\{2^{-k-1/2}\}_{k=0}^{N-1}$ on the diagonal. Since

- $\phi_{0,K} \in \text{Span}(\{\phi_{1,n}\}_n \geq K)$ (alternatively, $\phi^b_{0,N} \in \text{Span}(\{\phi^b_{1,n}\}_{n \geq N})$),

- $\operatorname{supp}(\phi^b_{0,N-1})$ ends to the right at $R+K-1$,
- $\operatorname{supp}(\phi_{1,2K+R-2}) = \operatorname{supp}(\phi^b_{1,K+R+N-2})$ also ends to the right at $R+K-1$,

replacing $\{\phi_{1,k}\}_{k\geq K}$ with $\{\phi^b_{1,k}\}_{k\geq N}$ gives (IV.13) for this choice of $\phi^b_{0,k}$, with X_e and Z_e having the stated indices. Since clearly $X_e = D$, it must be nonsingular. Since (IV.13) holds and X_e is nonsingular for this initial basis, Lemma IV.3.3 says that this will be the case for any other polynomial basis as well.

Now, (IV.8) can be written

$$\begin{aligned}
\begin{pmatrix} \phi^b_{0,0} \\ \vdots \\ \phi^b_{0,N-1} \end{pmatrix} &= C^T \begin{pmatrix} \phi_{0,-R+1}|_{[0,\infty)} \\ \vdots \\ \phi_{0,K-1}|_{[0,\infty)} \end{pmatrix} \\
&= C^T (G_{[-R+1,2K+R-2],2[-R+1,K)})^T \begin{pmatrix} \phi_{1,-R+1}|_{[0,\infty)} \\ \vdots \\ \phi_{1,2K+R-2}|_{[0,\infty)} \end{pmatrix} \\
&= C^T (G_{I_X})^T \begin{pmatrix} \phi_{1,-R+1}|_{[0,\infty)} \\ \vdots \\ \phi_{1,K-1}|_{[0,\infty)} \end{pmatrix} + C^T (G_{I_Z})^T \begin{pmatrix} \phi_{1,K} \\ \vdots \\ \phi_{1,2K+R-2} \end{pmatrix} \\
&= (G_{I_X} C)^T \begin{pmatrix} \phi_{1,-R+1}|_{[0,\infty)} \\ \vdots \\ \phi_{1,K-1}|_{[0,\infty)} \end{pmatrix} + (G_{I_Z} C)^T \begin{pmatrix} \phi^b_{1,N} \\ \vdots \\ \phi^b_{1,K+N+R-2} \end{pmatrix},
\end{aligned} \tag{IV.16}$$

where the matrix product with G^T was split into two parts on the third line. Noticing that (IV.13) can also be rewritten as

$$\begin{pmatrix} \phi^b_{0,0} \\ \vdots \\ \phi^b_{0,N-1} \end{pmatrix} = (X_e)^T C^T \begin{pmatrix} \phi_{1,-R+1} \\ \vdots \\ \phi_{1,K-1} \end{pmatrix} + (Z_e)^T \begin{pmatrix} \phi^b_{1,N} \\ \vdots \\ \phi^b_{1,K+N+R-2} \end{pmatrix}.$$

Comparing with (IV.16) and using the linear independence of $\{\{\phi_{1,n}\}_{n\geq -R+1}\}$ on $[0,\infty)$, one sees that

$$G_{I_X} C = C X_e \qquad\qquad G_{I_Z} C = Z_e.$$

Multiplying with $C^{\dagger}$ to the left in the first equation gives the first equation in (IV.14). ■

Some remarks on the initial choice of polynomials can be found in the technical report [2].

IV.4.1 Change of coordinates for staggered supports

We will now try to make a change of coordinates so that the new bases satisfy

$$\operatorname{supp}(\phi_{0,k+K-N}) \cap [0,\infty) = \operatorname{supp}(\phi^b_{0,k}). \tag{IV.17}$$

One says that the supports are *staggered*. For $n \geq N$ (IV.17) follows by definition. For all $n < N$, staggeredness is seen to be equivalent to the lower $N \times N$-block of C being upper triangular. The subspace of span $\left(\{\phi^b_{0,k}\}_{k=0}^{N-1}\right)$ consisting of functions on the form $\sum_{n=-R+1}^{K-1} c(n)\phi_{0,n}$ with c so that $c(K-N+k) = \cdots = c(K-1) = 0$, clearly has dimension k. If the $\phi^b_{0,k}$ have staggered supports, $\phi^b_{0,k}$ will lie in this k-dimensional subspace, so that standard orthogonalization procedures give us a unique (up to signs) orthonormal basis of functions with staggered supports. Staggered supports can thus be used to single out unique boundary functions (as partially noted in [7]).

More generally we will say that $\{f_i\}_i$ have staggered supports if $i < j$, $\operatorname{supp}(f_i) = [0,A]$, and $\operatorname{supp}(f_j) = [0,B]$ implies that $A < B$. This more general definition also comprises the setting in [16], and also possible supports for the mother wavelets. A coordinate change transforming the f_i to functions with staggered support can clearly be interpreted in terms of bringing a matrix to echelon form.

To obtain bases with staggered supports, Lemma IV.3.3 says that one needs to find a change of coordinates P so that the lower $N \times N$-block of CP is upper triangular. Clearly, this can be achieved by means of a QR-factorization, or an LU factorization. The technical report [2] contains further details.

IV.4.2 Change of coordinates for orthogonalization

In the following we will assume that $N = \tilde{N}$ (the deductions are a bit more complicated when $N \neq \tilde{N}$, see the technical report [2] for further details). Since

- $\{\phi^b_{0,k}\}_{k\geq N}$ and $\{\tilde{\phi^b}_{0,k}\}_{k\geq N}$ are biorthogonal,
- $\{\phi^b_{0,k}\}_{k=0}^{N-1}$ and $\{\tilde{\phi^b}_{0,k}\}_{k\geq \tilde{N}}$ are mutually orthogonal,
- $\{\phi^b_{0,k}\}_{k\geq N}$ and $\{\tilde{\phi^b}_{0,k}\}_{k=0}^{\tilde{N}-1}$ are mutually orthogonal,

in order to obtain biorthogonal bases for V^b_m and $\tilde{V}^b_m$, it is enough to find coordinate changes ensuring biorthogonality of $\{\phi^b_{0,k}\}_{k=0}^{N-1}$ and $\{\tilde{\phi^b}_{0,k}\}_{k=0}^{\tilde{N}-1}$ [3]. One of the two sets may here contain internal scaling functions. We will see that a coordinate change can be made so that it does not affect these. With $Y = \left(\left\langle \{\phi^{b,1}_{0,k}\}_{k=0}^{N-1}, \{\phi^{\tilde{b},1}{}_{0,k}\}_{k=0}^{\tilde{N}-1} \right\rangle\right)$ the Gramm matrix of the initial bases, one

[3] We do not go into details on when the Gramm matrix of these bases is invertible, but see [8, 18] for proofs for those considered in the literature

sees that, when $N = \tilde{N}$ (which will be assumed for simplicity in the following) ,

$$\langle \phi^{b,1}_{0,k}, \phi^{\tilde{b},1}{}_{0,l} \rangle = \sum_{1 \le r,s < N} (X_e)_{r,k} (\tilde{X}_e)_{s,l} \langle \phi^{b,1}_{0,r}, \phi^{\tilde{b},1}{}_{0,s} \rangle + \sum_{r \ge 0} (Z_e)_{r,k} (\tilde{Z}_e)_{r,l},$$

where it was used that $\langle \phi^{b,1}_{0,r}, \tilde{\phi}^{b,1}_{0,s} \rangle = \langle \phi^{b,1}_{1,r}, \tilde{\phi}^{b,1}_{1,s} \rangle$. This gives

$$Y = (X_e)^T Y \tilde{X}_e + (Z_e)^T \tilde{Z}_e, \tag{IV.18}$$

Solving $AVB^T = F$ is equivalent to solving the linear system $(A \otimes B)\text{vec}(V) = \text{vec}(F)$ [14], where $\otimes$ is the (left) Kronecker product, and where $\text{vec}(X)$ is the vector where the rows of X have been stacked horizontally and then transposed to a column vector. Equation (IV.18) can therefore be written as

$$(I - (X_e)^T \otimes (\tilde{X}_e)^T)\text{vec}(Y) = \text{vec}((Z_e)^T \tilde{Z}_e) \tag{IV.19}$$

(see also Theorem 3.2 in [17]. This paper also elaborates on the general computation of Gramm matrices), where I is the $N^2 \times N^2$ identity matrix[4].

Denote bases by

$$\mathcal{B} = \{\phi^{b,1}_{0,k}\}_{k=0}^{N-1} \qquad \mathcal{C} = \{\phi^{b,2}_{0,k}\}_{k=0}^{N-1}$$
$$\tilde{\mathcal{B}} = \{\phi^{\tilde{b},1}{}_{0,k}\}_{k=0}^{\tilde{N}-1} \qquad \tilde{\mathcal{C}} = \{\phi^{\tilde{b},2}{}_{0,k}\}_{k=0}^{\tilde{N}-1},$$

and let $P = P_{\mathcal{B} \leftarrow \mathcal{C}}$ and $\tilde{P} = P_{\tilde{\mathcal{B}} \leftarrow \tilde{\mathcal{C}}}$ be the corresponding coordinate changes (i.e., from the old to the new bases). It is straightforward to show that

$$(\langle \mathcal{C}, \tilde{\mathcal{C}} \rangle) = P^T (\langle \mathcal{B}, \tilde{\mathcal{B}} \rangle) \tilde{P}. \tag{IV.20}$$

Since one wants bases $\mathcal{C}$ and $\tilde{\mathcal{C}}$ so that $(\langle \mathcal{C}, \tilde{\mathcal{C}} \rangle) = I$, and since upper triangular coordinate changes preserve staggered supports, one seeks upper triangular matrices P and $\tilde{P}$ so that $P^T (\langle \mathcal{B}, \tilde{\mathcal{B}} \rangle) \tilde{P} = I$, i.e., so that

$$Y = (P^T)^{-1} \tilde{P}^{-1}.$$

Now, if $Y = LU$ is an L_1U-factorization [5] of Y, one can choose our upper triangular coordinate changes as $P = (L^{-1})^T$ and $\tilde{P} = U^{-1}$. In the orthogonal case where $\mathcal{B} = \tilde{\mathcal{B}}$ and $\mathcal{C} = \tilde{\mathcal{C}}$, Y is positive semidefinite, and thus has a unique Cholesky factorization $Y = LL^T$, so that one can choose $P = \tilde{P} = (L^{-1})^T$ as our coordinate change[6].

In the following, it will always be assumed that $\boldsymbol{\phi}^b_0$ and $\tilde{\boldsymbol{\phi}}^b_0$ are biorthogonal, both with staggered supports.

[4] X_e has eigenvalues $\{2^{-k-1/2}\}_{k=0}^{N-1}$. It follows that $\rho(X_e) < 1$, so that $I - (X_e)^T \otimes (\tilde{X}_e)^T$ is nonsingular, so that Y is unique.

[5] It is a major issue whether Y is nonsingular in the general case. Some special cases are handled in [8, 17, 18]. For Y to have a unique L_1U-factorization one also needs the principal leading submatrices to be nonsingular. The software implementation handles these issues only numerically. Other factorizations such as a LU_1 or LDU could also be chosen.

[6] It has been noted in the literature that Y can be badly conditioned. [9] proposes to use a Singular Value Decomposition of Y to address this problem. [9] does not assume staggered supports.

IV.5 Stable completion and the left edge mother wavelets

We define $\psi^b_{0,k} = \psi_{0,k+K-N}$ and $\tilde{\psi^b}_{0,k} = \tilde{\psi}_{0,k+\tilde{K}-\tilde{N}}$. This aligns $(\phi^b_{0,n}, \psi^b_{0,n})$ in the same way as $(\phi_{0,n}, \psi_{0,n})$. First the ψ^b- and $\tilde{\psi^b}$-functions that satisfy old refinement relations will be characterized.

Lemma IV.5.1. *The following hold.*

1. *If N_0 is chosen so that $2N_0 \geq 2N - K - 1 + \tilde{R}$, then $\{\psi^b_{0,n}\}_{n \geq N_0}$ can be expressed in terms of $\{\phi^b_{1,k}\}_{k \geq N}$.*

2. *If $\tilde{N}_0$ is chosen so that $2\tilde{N}_0 \geq 2\tilde{N} - \tilde{K} - 1 + R$, then $\{\tilde{\psi^b}_{0,n}\}_{n \geq \tilde{N}_0}$ can be expressed in terms of $\{\tilde{\phi^b}_{1,k}\}_{k \geq \tilde{N}}$.*

Proof. Since $\mathrm{supp}(G_1) = [-\tilde{R}, -\tilde{L}]$ when the wavelet is delay-normalized, only $\{\phi_{1,k}\}_{k \geq -\tilde{R}+1}$ contribute in ψ. $\psi^b_{0,N_0} = \psi_{0,N_0+K-N}$ can therefore be expressed in terms of

$$\{\phi_{1,k}\}_{k \geq 2(N_0+K-N)-\tilde{R}+1} = \{\phi^b_{1,k}\}_{k \geq 2N_0+K-N+1-\tilde{R}}.$$

$2N_0 + K - N + 1 - \tilde{R} \geq N$ thus implies that ψ^b_{0,N_0} can be expressed in terms of $\{\phi^b_{1,k}\}_{k \geq N}$, and 1. follows. 2. follows in the same way. ■

In the following it will be assumed that N_0 and $\tilde{N}_0$ satisfy the properties in Lemma IV.5.1, and we will set $N_0' = \max(N_0, \tilde{N}_0)$. The set $\{\psi^b_{0,k}\}_{k \geq N_0'} = \{\psi_{0,k}\}_{k \geq N_0'+K-N}$ accounts for all but the first $N_0' + K - N$ mother wavelets $\{\psi_{0,k}\}_{k \geq 0}$ in V_1^b, and similarly on the dual side. Define

$$\mathcal{A} = \{\{\phi^b_{0,n}\}_{n \geq 0}, \{\psi^b_{0,n}\}_{n \geq N_0'}\} \qquad \tilde{\mathcal{A}} = \{\{\tilde{\phi^b}_{0,n}\}_{n \geq 0}, \{\tilde{\psi^b}_{0,n}\}_{n \geq N_0'}\},$$

and write $A = \mathrm{span}(\mathcal{A})$, $\tilde{A} = \mathrm{span}(\tilde{\mathcal{A}})$. The previous lemma states that $A \subset V_1^b$, and $\tilde{A} \subset \tilde{V}_1^b$ and $\mathcal{A}$ and $\tilde{\mathcal{A}}$, are clearly biorthogonal by construction. Denote by $A^\perp$ the orthogonal complement of A in $\tilde{V}_1^b$.

If $g \in \mathcal{A}$, $\tilde{g}$ will denote the vector in $\tilde{\mathcal{A}}$ with the same index, and vice versa. We define, for $f \in A$ and $\tilde{f} \in \tilde{A}$,

$$P(f) = \sum_{g \in \mathcal{A}} \langle f, \tilde{g} \rangle g \qquad \tilde{P}(\tilde{f}) = \sum_{\tilde{g} \in \tilde{\mathcal{A}}} \langle \tilde{f}, g \rangle \tilde{g}.$$

Note that

- P equals the identity on A, and equals zero on $\tilde{A}^\perp$.

- $I - P$ equals the identity on $\tilde{A}^\perp$, and equals zero on A.

- The spaces A and $\tilde{A}^\perp$ are linearly independent: If $\boldsymbol{v} \in V_1^b$ lies in both these spaces it must be on the form $\boldsymbol{v} = \sum_{g \in \mathcal{A}} \alpha_g g$, and must for all $\tilde{h} \in \tilde{\mathcal{A}}$ satisfy $\langle \sum_{g \in \mathcal{A}} \alpha_i g, \tilde{h} \rangle = \alpha_h = 0$, where biorthogonality was used.

- $V_1^b = A \oplus \tilde{A}^\perp$, and $\boldsymbol{v} = P(\boldsymbol{v}) + (\boldsymbol{v} - P(\boldsymbol{v}))$ is the unique decomposition of $\boldsymbol{v} \in V_1^b$ in $A \oplus \tilde{A}^\perp$. P is thus a generalization of orthogonal projection, for which $A = \tilde{A}$.

Lemma IV.5.2. *Assume that $N_0 \geq R$ and $\tilde{N}_0 \geq \tilde{R}$, and that they satisfy the properties in Lemma IV.5.1. Assume also that the supports of $\boldsymbol{\phi}_1^b$ and $\tilde{\boldsymbol{\phi}}_1^b$ are staggered. Then the following hold.*

1. *Let*
$$S = [0, K - N + R) \cup \big(K - N + R - 1 + 2[1, N_0 - R]\big)$$
Then $|S| = N_0 + K - N$ and $\Big\{\{(I-P)\phi_{1,k}^b\}_{k\in S}, \{\psi_{0,n}^b\}_{n=N_0}^{N_0'-1}, \mathcal{A}\Big\}$ is a linearly independent set.

2. *Let*
$$\tilde{S} = [0, \tilde{K} - \tilde{N} + \tilde{R}) \cup \big(\tilde{K} - \tilde{N} + \tilde{R} - 1 + 2[1, \tilde{N}_0 - \tilde{R}]\big)$$
Then $|\tilde{S}| = \tilde{N}_0 + \tilde{K} - \tilde{N}$ and $\Big\{\{(I-\tilde{P})\tilde{\phi}^b{}_{1,k}\}_{k\in \tilde{S}}, \{\tilde{\psi}^b{}_{0,n}\}_{n=\tilde{N}_0}^{N_0'-1}, \tilde{\mathcal{A}}\Big\}$ is a linearly independent set.

Remark IV.5.3. The requirements $N_0 \geq R$, $\tilde{N}_0 \geq \tilde{R}$) are a bit difficult to grasp. If these are not fulfilled, the set $[0, K - N + R)$ will not be contained in S (the way this set is defined), and this will imply that one can't find enough functions (using the strategy of the proof) to find a new basis for V^b. Since one would like N_0 and $\tilde{N}_0$ to be as small as possible (to inherit as many of the old refinement relations as possible), the implementation computes these as

$$\begin{aligned} N_0 &= \max\left(\left\lceil \frac{2N - K - 1 + \tilde{R}}{2} \right\rceil, R\right), \\ \tilde{N}_0 &= \max\left(\left\lceil \frac{2\tilde{N} - \tilde{K} - 1 + R}{2} \right\rceil, \tilde{R}\right). \end{aligned} \tag{IV.21}$$

Proof. Only the first statement is considered since the second statement follows from the same line of arguments.

It is easily checked that $|S| = N_0 + K - N$, and that the last and largest entry of S is

$$K - N + R - 1 + 2(N_0 + K - N - (K - N + R)) = 2N_0 + K - N - R - 1$$

Due to staggered supports and Lemma IV.4.1, the highest index among boundary basis functions at resolutions 1 which contribute in $\phi_{0,0}^b$ is

$$K + R + N - 2 - 2(N-1) = K - N + R.$$

Also, the highest index among boundary basis functions at resolutions 1 which contribute in $\phi_{0,k}^b$ is $K - N + R + 2k$. It follows that the coordinate matrix of $\{\phi_{0,k}^b\}_{k\geq 0}$ and $\{\phi_{1,k}^b\}_{k\in S}$ relative to $\boldsymbol{\phi}_1^b$ has different highest contributing indices.

In particular, any finite set of these columns must be linearly independent, a fact which will be used in the final part of the proof.

Since clearly $\{\psi^b_{0,k}\}_{n=N_0}^{N_0'-1} \subset \tilde{A}^\perp$, also $\{\{(I-P)\phi^b_{1,k}\}_{k\in S}, \{\psi^b_{0,k}\}_{n=N_0}^{N_0'-1}\} \subset \tilde{A}^\perp$, so that linear independence of $\mathrm{Span}\{\{(I-P)\phi^b_{1,k}\}_{k\in S}, \{\psi^b_{0,k}\}_{n=N_0}^{N_0'-1}\}$ and $\mathcal{A}$ is immediate. It remains to show that $\{\{(I-P)\phi^b_{1,k}\}_{k\in S}, \{\psi^b_{0,k}\}_{n=N_0}^{N_0'-1}\}$ are linearly independent. Assume that

$$\sum_{k\in S}\alpha_k(I-P)\phi^b_{1,k} - \sum_{n=N_0}^{N_0'-1}\gamma_n\psi^b_{0,n} = 0,$$

where $\boldsymbol{\alpha}$ is a vector indexed over S, $\boldsymbol{\gamma}$ a vector indexed over $[N_0, N_0')$. But then

$$\begin{aligned}\sum_{k\in S}\alpha_k\phi^b_{1,k} - \sum_{n=N_0}^{N_0'-1}\gamma_n\psi^b_{0,n} &= \sum_{k\in S}\alpha_k P\phi^b_{1,k} + \sum_{k\in S}\alpha_k(I-P)\phi^b_{1,k} - \sum_{n=N_0}^{N_0'-1}\gamma_n\psi^b_{0,n}\\ &= \sum_{k\in S}\alpha_k P\phi^b_{1,k} \in A.\end{aligned}$$

This means that

$$\begin{aligned}\sum_{k\in S}\alpha_k\phi^b_{1,k} &= \sum_{n\geq 0}\beta_n\phi^b_{0,n} + \sum_{n\geq N_0'}\gamma_n\psi^b_{0,n} + \sum_{n=N_0}^{N_0'-1}\gamma_n\psi^b_{0,n}\\ &= \sum_{n\geq 0}\beta_n\phi^b_{0,n} + \sum_{n\geq N_0}\gamma_n\psi^b_{0,n},\end{aligned} \tag{IV.22}$$

for some vector $\boldsymbol{\beta}$, and where $\boldsymbol{\gamma}$ was expanded to a vector with indices from $[N_0, \infty)$. Viewing $\phi^b_{0,n}$ as functions on $[L-R+1, \infty)$ (see Equation (IV.6)), and similarly for $\phi^b_{1,k}$, we can ensure that (IV.22) holds also on $(-\infty, \infty)$, by adding a finite linear combination of functions $\phi_{1,k}$, $k \leq -R$ (i.e., scaling functions supported on $(-\infty, 0)$ on the left hand side:

$$\begin{aligned}\sum_{k\leq -R} r_k\phi_{1,k} + \sum_{k\in S}\alpha_k\phi^b_{1,k} &= \sum_{n\geq 0}\beta_n\phi^b_{0,n} + \sum_{n\geq N_0'}\gamma_n\psi^b_{0,n} + \sum_{n=N_0}^{N_0'-1}\gamma_n\psi^b_{0,n}\\ &= \sum_{n\geq 0}\beta_n\phi^b_{0,n} + \sum_{n\geq N_0}\gamma_n\psi^b_{0,n}.\end{aligned}$$

Taking inner products with $\widetilde{\phi^b}_{0,n}$ one sees that $\beta_n = 0$ for n sufficiently large. Taking inner products over $(-\infty, \infty)$ with $\{\tilde{\psi}_{0,n}\}_{n\geq N_0+K-N}$ on both sides (these may not be supported on $[0, \infty)$), one has that $\langle\phi^b_{1,k}, \tilde{\psi}_{0,N_0+K-N}\rangle_{(-\infty,\infty)} = 0$ for k in S and for $k \leq -R$ since

1. the largest entry of S is $2N_0 + K - N - R - 1$, so that the largest $\phi_{1,k}$ contributing on the left hand side is $\phi_{1,2(N_0+K-N)-R-1}$,

2. only $\{\tilde{\phi}_{1,k}\}_{k \geq 2(N_0+K-N)-R+1}$ contribute in $\tilde{\psi}_{0,N_0+K-N}$ (see the proof of Lemma IV.5.1),

3. $\boldsymbol{\phi}_1$ and $\tilde{\boldsymbol{\phi}}_1$ are biorthogonal.

Since also $\langle \phi^b_{0,n}, \tilde{\psi^b}_{0,k}\rangle_{(-\infty,\infty)} = 0$ for all k, it follows that $\gamma_n = 0$ for $n \geq N_0$. That also $\alpha_k = \beta_n = 0$ for $k \in S$, and for smaller n, follows since, as noted above, from the fact that any finite set of columns in the coordinate matrix of $\Big\{\{\phi^b_{1,k}\}_{k\in S}, \{\phi^b_{0,n}\}_{n\geq 0}\Big\}$ relative to $\boldsymbol{\phi}^b_1$ are linearly independent. It follows that $\Big\{\{(I-P)\phi^b_{1,k}\}_{k\in S}, \{\psi^b_{0,n}\}_{n=N_0}^{N_0'-1}, \mathcal{A}\Big\}$ is a linearly independent set. ■

In the next section, this analysis is repeated at the right edge, and it follows from a simple dimension count that the two mentioned linearly independent sets are in fact bases for V^b_1 and $\tilde{V}^b_1$. If $S = \{k_1, \ldots, k_{|S|}\}$, one therefore defines for $N - K \leq n < N_0$,

$$\psi^b_{0,n} = (I-P)\phi^b_{1,k_n} \qquad\qquad \tilde{\psi^b}_{0,n} = (I-\tilde{P})\tilde{\phi^b}_{1,k_n}.$$

We point out that when $N - K < 0$ this gives negative index sets. In particular the matrix G^b takes the form

$$G^b = \begin{pmatrix} [\psi^b_{0,N-K}]_{\boldsymbol{\phi}^b_1} & \cdots & [\psi^b_{0,-1}]_{\boldsymbol{\phi}^b_1} & \underline{[\phi^b_{0,0}]_{\boldsymbol{\phi}^b_1}} & [\psi^b_{0,0}]_{\boldsymbol{\phi}^b_1} & [\phi^b_{0,1}]_{\boldsymbol{\phi}^b_1} & [\psi^b_{0,1}]_{\boldsymbol{\phi}^b_1} & \cdots \end{pmatrix} \tag{IV.23}$$

Lemma IV.5.4. *The coordinate matrix of the* $\{\{(I-P)\phi^b_{1,k}\}_{k\in S}$ *relative to* $\boldsymbol{\phi}^b_1$ *is*

$$I_{:,S} - G^b_{:,2[0,T]}(\tilde{G}^b_{S,2[0,T]})^T,$$

where T *is the largest integer so that* $2T \leq 2N_0 - R - \tilde{L} - 1$*, i.e.,*

$$T = N_0 + \left\lfloor -\frac{R+\tilde{L}+1}{2} \right\rfloor \tag{IV.24}$$

Proof. In the proof above it was shown that $\langle \phi^b_{1,k}, \tilde{\psi^b}_{0,n}\rangle = 0$ for $k \in S$ and $n \geq N_0'$, so that

$$P(\phi^b_{1,k}) = \sum_{n\geq 0} \langle \phi^b_{1,k}, \tilde{\phi^b}_{0,n}\rangle \phi^b_{0,n}.$$

Now, the $\tilde{\phi^b}_{0,n}$ can be expressed in terms of $\{\tilde{\phi^b}_{1,r}\}_{r \geq 2n+\tilde{K}-\tilde{N}+\tilde{L}}$, and since the largest index in S is $2N_0 + K - N - R - 1$, one can have contribution in the sum above only when n satisfies

$$2N_0 + K - N - R - 1 \geq 2n + \tilde{K} - \tilde{N} + \tilde{L}.$$

Since $N-K=\tilde{N}-\tilde{K}$, this occurs when $2n \leq 2N_0 - R - \tilde{L} - 1$. This gives the expression for T. One obtains

$$\begin{aligned} P(\phi^b_{1,k}) &= \sum_{n=0}^{T} \langle \phi^b_{1,k}, \tilde{\phi}^b{}_{0,n} \rangle \phi^b_{0,n} = \begin{pmatrix} \phi^b_{0,0} & \cdots & \phi^b_{0,T} \end{pmatrix} (\tilde{G}^b_{k,2[0,T]})^T \\ &= \begin{pmatrix} \phi^b_{1,0} & \phi^b_{1,1} & \cdots \end{pmatrix} G^b_{:,2[0,T]} (\tilde{G}^b_{k,2[0,T]})^T. \end{aligned}$$

It follows that

$$(I-P)\phi^b_{1,k} = \phi^b_{1,k} - \begin{pmatrix} \phi^b_{1,0} & \phi^b_{1,1} & \cdots \end{pmatrix} G^b_{:,2[0,T]} (\tilde{G}^b_{k,2[0,T]})^T.$$

This gives the individual columns in the coordinate matrix of the $\{\{(I-P)\phi^b_{1,k}\}_{k\in S}$ relative to $\boldsymbol{\phi}^b_1$, so that this matrix is $I_{:,S} - G^b_{:,2[0,T]}(\tilde{G}^b_{S,2[0,T]})^T$. ■

Remark IV.5.5. A $\tilde{T}$ also needs to be computed for the dual wavelet (i.e., $\tilde{T} = \tilde{N}_0 + \left\lfloor -\frac{\tilde{R}+L+1}{2} \right\rfloor$).

Remark IV.5.6. The previous lemma does not provide any row limits. To deduce such limits, note first that $\phi^b_{0,T}$ can be expressed in terms of $\{\phi^b_{1,k}\}_{k=0}^{2T+K-N+R}$. Note also that, since by definition $2T \geq 2N_0 - R - \tilde{L} - 2$, it follows that

$$2N_0 + K - N - R - 1 \leq 2T + R + \tilde{L} + 2 + K - N - R - 1 = 2T + 1 + \tilde{L} + K - N.$$

Since $\tilde{L}+1 \leq R$, this is $\leq 2T + R + K - N$. Since the largest entry in S is $2N_0 + K - N - R - 1$, and after dropping rows that are zero, this proves that the coordinate matrix of the $\{\{(I-P)\phi^b_{1,k}\}_{k\in S}$ can also be written relative to $\{\phi^b_{1,0}, \ldots, \phi^b_{1,2T+K-N+R}\}$, and as

$$I_{[0,2T+K-N+R],S} - G^b_{[0,2T+K-N+R],2[0,T]}(\tilde{G}^b_{S,2[0,T]})^T.$$

One now redefines $\{\psi^b_{0,k}\}_{k=N-K}^{N_0-1}$ as the functions $\{(I-P)\phi^b_{1,k}\}_{k\in S}$, and set $\boldsymbol{\psi}^b_0 = \{\psi^b_{0,k}\}_{k\geq N-K}$, and

$$\mathcal{C}^b_1 = \{\{\psi^b_{0,k}\}_{k=N-K}^{-1}, \phi^b_{0,0}, \psi^b_{0,0}, \phi^b_{0,1}, \psi^b_{0,1}, \ldots\},$$

with the 1-level DWT the change of coordinates from $\boldsymbol{\phi}^b_1$ to $\mathcal{C}^b_1$. Set now

$$\begin{pmatrix} X_o \\ Z_o \end{pmatrix} = I_{:,S} - G^b_{:,2[0,T]}(\tilde{G}^b_{S,2[0,T]})^T, \tag{IV.25}$$

i.e.,

$$\begin{pmatrix} \psi^b_{0,N-K} \\ \vdots \\ \psi^b_{0,N_0-1} \end{pmatrix} = (X_o)^T \begin{pmatrix} \phi^b_{1,0} \\ \vdots \\ \phi^b_{1,N-1} \end{pmatrix} + (Z_o)^T \begin{pmatrix} \phi^b_{1,N} \\ \vdots \end{pmatrix}. \tag{IV.26}$$

The column indices of X_o and Z_o are in $[N-K, N_0-1]$. As for the scaling functions, we address how coordinate changes affect X_o and Z_o.

Lemma IV.5.7. *Assume that a change of coordinates is applied to the left edge mother wavelets without altering the left edge scaling functions, and let* $\{\psi^{b,1}_{0,k}\}_{k=N-K}^{N_0-1}$ *and* $\{\psi^{b,2}_{0,k}\}_{k=N-K}^{N_0-1}$ *be bases, and* $P = P_{\{\psi^{b,1}_{0,k}\}\leftarrow\{\psi^{b,2}_{0,k}\}}$ *the change of coordinate matrix from the second to the first basis. The change of coordinates from* $\{\psi^{b,1}_{0,k}\}_{k=N-K}^{N_0-1}$ *to* $\{\psi^{b,2}_{0,k}\}_{k=N-K}^{N_0-1}$ *transforms* X_o *and* Z_o *according to*

$$\begin{pmatrix} X_o \\ Z_o \end{pmatrix} \to \begin{pmatrix} X_o \\ Z_o \end{pmatrix} P. \tag{IV.27}$$

Proof. One gets as above

$$\begin{pmatrix} \psi^{b,2}_{0,N-K} \\ \vdots \\ \psi^{b,2}_{0,N_0-1} \end{pmatrix} = P^T \begin{pmatrix} \psi^{b,1}_{0,N-K} \\ \vdots \\ \psi^{b,1}_{0,N_0-1} \end{pmatrix} = P^T \left(X_o^T \begin{pmatrix} \phi^b_{1,0} \\ \vdots \\ \phi^b_{1,N-1} \end{pmatrix} + Z_o^T \begin{pmatrix} \phi_{1,N} \\ \vdots \end{pmatrix} \right)$$

$$= (X_o P)^T \begin{pmatrix} \phi^b_{1,0} \\ \vdots \\ \phi^b_{1,N-1} \end{pmatrix} + (Z_o P)^T \begin{pmatrix} \phi_{1,N} \\ \vdots \end{pmatrix}.$$

(IV.27) follows. ■

(IV.27) is applied twice. First the supports of $\{\psi^b_{0,k}\}_{k=0}^{N-1}$ is made staggered by finding a coordinate change which brings $\begin{pmatrix} X_o \\ Z_o \end{pmatrix}$ to echelon form. Secondly, the mother wavelets at the left edge are bi-orthogonalized, preserving their staggeredness. One now needs the $(N_0+K-N)\times(\tilde{N}_0+K-N)$ Gramm matrix

$$Y = \left(\left\langle \{\psi^b_{0,k}\}_{k=N-K}^{N_0-1}, \{\tilde{\psi^b}_{0,k}\}_{k=N-K}^{\tilde{N}_0-1} \right\rangle\right).$$

Similarly to Equation (IV.18) it follows that $Y = (X_o)^T\tilde{X}_o + (Z_o)^T\tilde{Z}_o$ (there is no Y on the right hand side here, however, since there are no ψ-functions on the right hand side). With Y found, one proceeds as in the end of Section IV.4 to find the required changes of coordinates.

IV.6 The right edge

Up to now K and $\tilde{K}$ have denoted the number of scaling functions at resolution 1 needed to synthesize $\{\phi^b_{0,k}\}_{k=0}^{N-1}$ and $\{\tilde{\phi^b}_{0,k}\}_{k=0}^{\tilde{N}-1}$. In analyzing the right edge, the flexibility in these numbers needs to be exploited, in order to obtain a Discrete Wavelet Transform on the interval for the dimension in question. K and $\tilde{K}$ will now be assigned different values at the left and right edge, and we write K_L and $\tilde{K}_L$ for the left edge values, K_R and $\tilde{K}_R$ for the right edge values. Since the value N_0 depended on K as well, its notation is changed to $N_{0,L}$ and $N_{0,R}$.

All functions are assumed to be defined on $[0, M]$. The operation $m(f) = f(M-t)$ ”mirrors” functions on $[0, M]$ so that the left and right edges are swapped. For the right edge, we reuse the left edge analysis, and the right edge functions are found so that their mirrors are on the form that has been considered, i.e., delay-normalized, possibly with the same supports. If this is the case, the right edge analysis simply boils down to repeating the left edge analysis with reversed filter coefficients. The following result addresses this.

Theorem IV.6.1. *Assume that (ϕ, ψ) is delay-normalized. Then*

1. $(m(\phi_{0,n}), m(\psi_{0,M-1}))$ *is delay-normalized for any n.*

2. $m(\psi_{0,M-1}))$ *has the same support as ψ if and only if $L + R = \tilde{L} + \tilde{R}$ (i.e.,* $\operatorname{supp}(\phi)$ *and* $\operatorname{supp}(\tilde{\phi})$ *have the same midpoints).*

3. $m(\phi_{0,n})$ *has the same support as ϕ if and only if $n = M - (L + R)$.*

Some comments are in order.

1. The case $L + R = \tilde{L} + \tilde{R} = 0$ includes wavelets with symmetric filters, while the case $L + R = \tilde{L} + \tilde{R} = 1$ includes orthogonal wavelets with the support assumption (IV.2).

2. Recall that $(\phi_{0,n}, \psi)$ is delay-normalized for any n, as long as (ϕ, ψ) is. This means that, when $L + R = \tilde{L} + \tilde{R}$, one can always assume that the common value is 0 or 1 (by changing the scaling function at the start). Moreover, the result says that

- when $L + R = 0$, $(m(\phi_{0,M}), m(\psi_{0,M-1})))$ is delay-normalized, with the same supports as (ϕ, ψ).

- when $L + R = 1$, $(m(\phi_{0,M-1}), m(\psi_{0,M-1})))$ is delay-normalized, with the same supports as (ϕ, ψ).

In both cases the new pairs of delay-normalized functions are adjacent basis functions, the only difference being that their internal order differs (something an implementation must take into account). In summary, when $L + R = \tilde{L} + \tilde{R}$ one can always assume equal supports at the left and right edge, and that the mirroring process preserves the ordering of the basis functions.

3. It is straightforward to prove that $m(\phi_{m,2^m M-(L+R)})$ has the same support as $\phi_{m,0}$, in the same way one proves the third statement. If $L + R = \tilde{L} + \tilde{R}$ and (ϕ, ψ) is delay-normalized, it also follows that $m(\psi_{m,2^m M-1})$ has the same support as $\psi_{m,0}$. In other words, when constructing a multiresolution analysis it is desirable to consider the sets $\{\phi_{m,n}\}_{n=0}^{2^m M-(L+R)}$ and $\{\psi_{m,n}\}_{n=0}^{2^m M-1}$, since a mirroring operation on each of these simply produce new functions with the same supports.

4. When $L + R \neq \tilde{L} + \tilde{R} = 0$ the right edge analysis is still possible, but the supports at the right edge will be different from those at the left edge. Also, the relative ordering at the right edge needs to be handled differently for the wavelet and the dual wavelet, making things more complicated. The software implementation, therefore, handles only the case $L + R = \tilde{L} + \tilde{R}$, and this will be assumed in the following.

Note that, assuming that $L+R=0$ or 1, from the two filters G_0 and G_1 one can compute everything (L, R, $\tilde{L}$, and $\tilde{R}$, as well as H_0 and H_1, N and $\tilde{N}$). In particular, there is no need to specify the positions of the filter coefficients, since these can also be inferred. The software implementation takes advantage of this fact.

Proof of Theorem IV.6.1. One has that

$$\operatorname{supp}(m(\phi_{0,M})) = [-R,-L] \qquad \operatorname{supp}(m(\tilde{\phi}_{0,M})) = [-\tilde{R},-\tilde{L}].$$

In order for $m(\psi_{0,n})$ to ensure delay-normalized mirrors, Equation (IV.5) says that $\operatorname{supp}(m(\psi_{0,n})$ must equal $[(-R+\tilde{L}+1)/2, (-L+\tilde{R}+1)/2]$. Also, $m(\psi_{0,n})$ has support $M-n+[(\tilde{L}-R-1)/2, (\tilde{R}-L-1)/2]$. Comparing one obtains that $1/2 = M-n-1/2$, so that $n = M-1$. This proves the first claim.

Equal supports for $\operatorname{supp}(m(\psi_{0,M-1})$ and ψ requires that (using Equation(IV.5) again)

$$[(L-\tilde{R}+1)/2, (R-\tilde{L}+1)/2] = [(-R+\tilde{L}+1)/2, (-L+\tilde{R}+1)/2],$$

i.e., $L+R = \tilde{L}+\tilde{R}$. Equal supports for ϕ and $m(\phi_{0,n})$ requires that $[L,R] = M-n-[R,L]$, so that $n = M-(L+R)$. ■

The sets $\boldsymbol{\phi}_m^b$ and $\boldsymbol{\psi}_m^b$ are now redefined as follows:

1. $\boldsymbol{\phi}_m^b$: Take $\{\phi_{m,n}\}_{n=0}^{2^m M-(L+R)}$, remove the first K_L functions and replace them with N modified functions, remove the last K_R functions and replace them with N modified functions.
2. $\boldsymbol{\psi}_m^b$: Take $\{\psi_{m,n}\}_{n=0}^{2^m M-1}$, remove the first $N_{0,L}$ functions and replace them with $K_L - N + N_{0,L}$ modified functions, remove the last $N_{0,R}$ functions and replace them with $K_R - N + N_{0,R}$ modified functions.

Both these are linearly independent (simply repeat the arguments from Section IV.5 by also including modified functions at the right edge), and one can define $V_m^b = \operatorname{Span}(\boldsymbol{\phi}_m^b)$, $W_m^b = \operatorname{Span}(\boldsymbol{\psi}_m^b)$ (and their duals, and one still has biorthogonality) as before. A simple count gives that

$$\dim(V_m^b) = 2N + 2^m M - L - R - K_R - K_L + 1 \quad \dim(W_m^b) = 2^m M. \tag{IV.28}$$

It is easily checked that $\dim(V_m^b)+\dim(W_m^b) = \dim(V_{m+1}^b)$, and since $V_m^b \cup W_m^b \subseteq V_{m+1}^b$, it follows that $\mathcal{C}_m^b$ and $\boldsymbol{\phi}_m^b$ are alternative bases for V_m^b (redefining $\mathcal{C}_m^b$ in the obvious way), so that one can define the DWT/IDWT as before. The following result provides a requirement on $K_L + K_R$ in order for an m-level DWT to be computable.

Theorem IV.6.2. *Let $\dim(\boldsymbol{x})$ be the number of components in $\boldsymbol{x}$. An m-level DWT and an m-level dual DWT of $\boldsymbol{x}$ are possible only if*

$$\dim(\boldsymbol{x}) + L + R - 2N + K_L + K_R - 1 \text{ is divisible by } 2^m. \tag{IV.29}$$

Remark IV.6.3. The value of M is not needed in the computations, and is eliminated in favour of $\dim(\boldsymbol{x})$. The proof of this theorem makes it clear that M must be chosen as

$$M = 2^{-m}(\dim(\boldsymbol{x}) + L + R - 2N + K_L + K_R - 1) \tag{IV.30}$$

in order for $\boldsymbol{x}$ to be the coordinates in a wavelet basis on $[0, M]$. In addition to Equation (IV.29), we need also take into account that K_L and K_R must satisfy the requirements of Definition IV.3.1.

Proof of Theorem IV.6.2. Since $\boldsymbol{x}$ are coordinates in $\boldsymbol{\phi}_m^b$, (IV.28) says that

$$\dim(\boldsymbol{x}) = 2N + 2^m M - L - R - K_R - K_L + 1,$$

so that an m-level DWT is possible if and only if $\dim(\boldsymbol{x}) + L + R - 2N + K_L + K_R - 1$ is divisible by 2^m. Equation (IV.30) follows by reorganizing this. By combining (IV.28) for $k = 1$ and $k = m$ and eliminating M, one also deduces

$$\dim(\boldsymbol{\phi}_1^b) = 2^{1-m}\dim(\boldsymbol{x}) + (1 - 2^{1-m})(2N - L - R - K_L - K_R + 1), \tag{IV.31}$$

which we also will have use for. Since $K_L - N = \tilde{K_L} - \tilde{N}$, and $L + R = \tilde{L} + \tilde{R}$, the same requirement is needed for a dual m-level DWT as well. ■

The condition from Theorem IV.6.2 is not sufficient for an m-level DWT to be computable, however: It may be that the smallest resolution spaces do not have room for all the boundary functions needed in the construction. Details on this can be found in [2].

In summary, the software first finds (for a given m and $\dim(\boldsymbol{x})$) $K_L + K_R$ from Equation (IV.29), and then chooses K_L and K_R so that they are as equal as possible.

Of particular interest are the cases where no shift in the basis functions is needed, i.e., when we can set $K_L = K_R = N$. Important cases are

1. $\dim(\boldsymbol{x}) = M2^m + 1$ when $L + R = \tilde{L} + \tilde{R} = 0$ (for instance wavelets with symmetric filters),
2. $\dim(\boldsymbol{x}) = M2^m$ when $L + R = \tilde{L} + \tilde{R} = 1$ (for instance orthonormal wavelets).

IV.7 Biorthogonal Spline Wavelets on the interval as defined in [16]

[16] gives a more refined construction of Biorthogonal Spline wavelets on the interval where the primal boundary functions are fixed, and directly defined from the Schoenberg-Spline basis with equidistant knots on the interval [21]. All internal primal scaling functions are included, thereby changing the multiresolution minimally. This differs from the previous part of the paper, where internal functions may be absorbed in the boundary functions, leading to

more changes to the multiresolution at the boundaries (in terms of boundary functions with wilder oscillations). This is particularly the case when K and $\tilde{K}$ are increased to adapt the multiresolution to a given vector length. The strategy in [16], however, is not adaptable to different vector lengths: For this, a combination with the strategy from the previous part of the paper is needed.

Let us explain how one can extend the results from [16] to the general delay-normalized case, including how to make stable completions. One clearly needs $s = \min(R - L - 1, N)$ boundary adapted functions at the primal side, and $\tilde{s} = \min(\tilde{R} - \tilde{L} - 1, \tilde{N})$ boundary adapted functions at the dual side. With $d := L + s - \tilde{L} - \tilde{s} \geq 0$, this modifies equations (IV.6) and (IV.7) in Definition IV.3.1 to

$$\phi^b_{0,k}(t) = \begin{cases} \sum_{n=-R+1}^{-L-1} c_k(n)\phi_{0,n}(t) & \text{for } 0 \leq k < s \\ \phi_{0,k-L-s}(t) & \text{for } k \geq s \end{cases} \tag{IV.32}$$

$$\tilde{\phi}^b{}_{0,k}(t) = \begin{cases} \sum_{n=-\tilde{R}+1}^{-\tilde{L}-1} \tilde{c}_k(n)\tilde{\phi}_{0,n}(t) & \text{for } d \leq k < d + \tilde{s} \\ \tilde{\phi}_{0,k-L-s}(t) & \text{for } k \geq d + \tilde{s}, \end{cases} \tag{IV.33}$$

so that we need to construct d functions $\{\tilde{\phi}^b\}_{k=0}^{d-1}$ on the dual side (it is easily checked that $d = N - 2$ in the case of Spline wavelets, agreeing with the construction in [16]). These are defined recursively as

$$\begin{pmatrix} \tilde{\phi}^b{}_{0,0} \\ \vdots \\ \tilde{\phi}^b{}_{0,d-1} \end{pmatrix} = (\tilde{G}^b_{:,2[0,d)})^T \left(\begin{array}{c} \tilde{\phi}^b{}_{1,0} \\ \vdots \\ \tilde{\phi}^b{}_{1,d-1} \\ \hline \tilde{\phi}^b{}_{1,d} \\ \vdots \\ \tilde{\phi}^b{}_{1,d+\tilde{s}-1} \\ \hline \tilde{\phi}^b{}_{1,d+\tilde{s}} \\ \vdots \end{array} \right).$$

This is simply a restatement of Equation (4.3) in [16] with all values of k there combined: $\tilde{G}^b_{:,2[0,d)}$ represents the refinement relations for the functions $\{\tilde{\phi}^b\}_{k=0}^{d-1}$ we seek to construct (these refinement relations were previously denoted by $\tilde{X}_e$ and $\tilde{Z}_e$). We will assume that the nonzero indices in column n of $\tilde{G}^b_e$ come at indices $[n+1, x+2n]$ (i.e., we assume staggered supports, and that the $\phi^b_{0,n}$ can be defined in terms of $\phi^b_{0,n+1}, \ldots, \phi^b_{0,d-1}$, i.e., iteratively in terms of previously constructed functions). This is again all compatible with [16], with the additional complication of finding a number x so that unique refinement relations can be found. The following result is the delay-normalized generalization of Theorem 4.13 in [16], and explains how to find the columns in $\tilde{G}^b$ iteratively, and the value for x.

Theorem IV.7.1. *Define $t = n + \left\lfloor \frac{x+s+1}{2} \right\rfloor$, and assume that $\boldsymbol{z}$ is a solution to the linear system*

$$\left(\left\langle \begin{pmatrix} \phi^b_{0,0} \\ \vdots \\ \phi^b_{0,t-1} \end{pmatrix}, \begin{pmatrix} \tilde{\phi}^b{}_{1,n+1} \\ \vdots \\ \tilde{\phi}^b{}_{1,x+2n} \end{pmatrix} \right\rangle \right) \boldsymbol{z} = \boldsymbol{e}_n.$$

Define column n of $\tilde{G}^b_e$ so that $(\tilde{G}^b_e)_{[n+1,x+2n],n} = \boldsymbol{z}$, and zero elsewhere, and define

$$\tilde{\phi}^b{}_{0,n} = ((\tilde{G}^b_e)_{[n+1,x+2n],n})^T \begin{pmatrix} \tilde{\phi}^b{}_{1,n+1} \\ \vdots \\ \tilde{\phi}^b{}_{1,x+2n} \end{pmatrix}$$

Then $\langle \phi^b_{0,k}, \tilde{\phi}^b{}_{0,n} \rangle = \delta_{k,n}$ for all k. Moreover, the linear system is square if $x = s+1$ (in which case $t = n+s+1$).

The value of x here is again compatible with the statements for the Spline case in [16]. We do not prove that the coefficient matrix of the above system is invertible. As commented in [16], numerical experiments show this to be true for the most common values of N an $\tilde{N}$ in the Spline case, but a general proof for this was not given.

Proof. If $k \geq s$ (i.e., $\phi^b_{0,k}$ is internal), only the $\{\phi_{1,t}\}_{t \geq 2k-2s-L}$ contribute in $\phi^b_{0,k} = \phi_{0,k-L-s}$. If $n < d$ (i.e., $\tilde{\phi}^b{}_{0,n}$ is one of the functions we seek to construct), only the

$$\{\tilde{\phi}^b{}_{1,t}\}_{t=n+1}^{x+2n} = \{\tilde{\phi}^b{}_{1,t}\}_{t=n+1}^{d-1} \cup \{\tilde{\phi}^b{}_{1,t}\}_{d \leq t \leq x+2n}$$

contribute in $\tilde{\phi}^b{}_{0,n}$. In $\{\tilde{\phi}^b{}_{1,t}\}_{d \leq t \leq x+2n}$ only the $\{\tilde{\phi}_{1,t}\}_{t \leq x+2n-L-s}$ contribute. Thus, if $2k-2s-L > x+2n-L-s$ (i.e., $2k > 2n+x+s$), we have that

$$\langle \phi^b_{0,k}, \tilde{\phi}^b{}_{0,n} \rangle = \left\langle \phi^b_{0,k}, ((\tilde{G}^b_e)_{[n+1,d-1],n})^T \begin{pmatrix} \tilde{\phi}^b{}_{1,n+1} \\ \vdots \\ \tilde{\phi}^b{}_{1,d-1} \end{pmatrix} \right\rangle.$$

In particular, $\langle \phi^b_{0,k}, \tilde{\phi}^b{}_{0,d-1} \rangle = 0$ when $2k > 2d-2+x+s$.

Assume now that $\langle \phi^b_{0,k}, \tilde{\phi}^b{}_{0,n'} \rangle = 0$ for $n' = n+1, \ldots, d-1$, $k > x$. Using the above, for $2k > 2n+x+s$ (i.e., $k \geq n + \left\lfloor \frac{x+s+1}{2} \right\rfloor =: t$),

$$\begin{aligned} \langle \phi^b_{0,k}, \tilde{\phi}^b{}_{0,n} \rangle &= \left\langle \phi^b_{0,k}, ((\tilde{G}^b_e)_{[n+1,d-1],n})^T \begin{pmatrix} \tilde{\phi}^b{}_{1,n+1} \\ \vdots \\ \tilde{\phi}^b{}_{1,d-1} \end{pmatrix} \right\rangle \\ &= \left\langle ((G^b_e)_{2k-L-s+[L,R],k})^T \begin{pmatrix} \phi^b_{1,2k-s} \\ \vdots \\ \phi^b_{1,2k-s-L+R} \end{pmatrix}, ((\tilde{G}^b_e)_{[n+1,d-1],n})^T \begin{pmatrix} \tilde{\phi}^b{}_{1,n+1} \\ \vdots \\ \tilde{\phi}^b{}_{1,d-1} \end{pmatrix} \right\rangle, \end{aligned}$$

which is zero by assumption (since $2k - s > 2n + x \geq x$). Note that $t = x + n$ if and only if

$$x + n = n + \left\lfloor \frac{x+s+1}{2} \right\rfloor,$$

which is seen to hold if $x = s+1$ (this agrees with $x = N$ in the spline case). We have that

$$\begin{aligned}\left\langle \begin{pmatrix} \phi^b_{0,0} \\ \vdots \\ \phi^b_{0,t-1} \end{pmatrix}, \tilde{\phi}^b{}_{0,n} \right\rangle &= \left\langle \begin{pmatrix} \phi^b_{0,0} \\ \vdots \\ \phi^b_{0,t-1} \end{pmatrix}, ((\tilde{G}^b_e)_{[n+1,x+2n],n})^T \begin{pmatrix} \tilde{\phi}^b{}_{1,n+1} \\ \vdots \\ \tilde{\phi}^b{}_{1,x+2n} \end{pmatrix} \right\rangle \\ &= \left(\left\langle \begin{pmatrix} \phi^b_{0,0} \\ \vdots \\ \phi^b_{0,t-1} \end{pmatrix}, \begin{pmatrix} \tilde{\phi}^b{}_{1,n+1} \\ \vdots \\ \tilde{\phi}^b{}_{1,x+2n} \end{pmatrix} \right\rangle \right) (\tilde{G}^b_e)_{[n+1,x+2n],n},\end{aligned}$$

and we have biorthogonality if this equals $\boldsymbol{e}_n$ (since orthogonality with $\{\phi^b_{0,r}\}_{r \geq t}$ was shown to hold above). This implies that $(\tilde{G}^b_e)_{[n+1,x+2n],n}$ (and thus the entire column n of $\tilde{G}^b_e$) can be found by solving the stated linear system. ■

To obtain the stated coefficient matrix one can compute

$$\begin{aligned}&\left(\left\langle \begin{pmatrix} \phi^b_{0,0} \\ \vdots \\ \phi^b_{0,n+s} \end{pmatrix}, \begin{pmatrix} \tilde{\phi}^b{}_{1,n+1} \\ \vdots \\ \tilde{\phi}^b{}_{1,s+1+2n} \end{pmatrix} \right\rangle \right) \\ =&((G_e)^b_{:,[0,n+s]})^T \left(\left\langle \begin{pmatrix} \phi^b_{1,0} \\ \vdots \end{pmatrix}, \begin{pmatrix} \tilde{\phi}^b{}_{1,n+1} \\ \vdots \\ \tilde{\phi}^b{}_{1,s+1+2n} \end{pmatrix} \right\rangle \right) \\ =&((G^b_e)_{:,[0,s+n]})^T \left(\left\langle \begin{pmatrix} \phi^b_{1,0} \\ \vdots \end{pmatrix}, \begin{pmatrix} \tilde{\phi}^b{}_{1,n+1} \\ \vdots \\ \tilde{\phi}^b{}_{1,d-1} \end{pmatrix} \right\rangle \right. \\ &\left. \left\langle \begin{pmatrix} \phi^b_{1,0} \\ \vdots \end{pmatrix}, \begin{pmatrix} \tilde{\phi}^b{}_{1,d} \\ \vdots \\ \tilde{\phi}^b{}_{1,d+\tilde{s}-1} \end{pmatrix} \right\rangle \left\langle \begin{pmatrix} \phi^b_{1,0} \\ \vdots \end{pmatrix}, \begin{pmatrix} \tilde{\phi}^b{}_{1,d+\tilde{s}} \\ \vdots \\ \tilde{\phi}^b{}_{s+1+2n} \end{pmatrix} \right\rangle \right) \\ =&((G_e)^b_{:,[0,s+n]})^T \left(\begin{pmatrix} 0_{n+1,d-n-1} \\ I_{d-n-1} \end{pmatrix} \quad \begin{pmatrix} Y \\ \tilde{C}_{[-L+1,-\tilde{L}),:} \end{pmatrix} \quad \begin{pmatrix} 0_{d+\tilde{s},\tilde{L}-L+2n+2} \\ I_{\tilde{L}-L+2n+2} \end{pmatrix} \right),\end{aligned}$$

where we set

$$Y = \left\langle \begin{pmatrix} \phi^b_{0,0} \\ \vdots \\ \phi^b_{0,s-1} \end{pmatrix}, \begin{pmatrix} \tilde{\phi}^b{}_{0,d} \\ \vdots \\ \tilde{\phi}^b{}_{0,d+\tilde{s}-1} \end{pmatrix} \right\rangle.$$

This can be found as in Section IV.4.2, with further in the technical report [2], and where $\tilde{C}$ was defined from (IV.33) as in Section IV.3. It is now clear that

$$\left\langle \begin{pmatrix} \phi^b_{0,0} \\ \vdots \\ \phi^b_{0,d+\tilde{s}-1} \end{pmatrix}, \begin{pmatrix} \tilde{\phi}^b{}_{0,0} \\ \vdots \\ \tilde{\phi}^b{}_{0,d+\tilde{s}-1} \end{pmatrix} \right\rangle = \begin{pmatrix} I_d & Y \\ 0_{\tilde{s},d} & \tilde{C}_{[-L+1,-\tilde{L}),:} \end{pmatrix}.$$

Since this is upper triangular, one can proceed as in [16] to biorthogonalize while preserving staggered supports.

It is not too hard to generalize the arguments from Section IV.5 in order to obtain a stable completion also here. Simply note that the quantity $K - N$ needs to be replaced by $-L - s$ (compare Definition IV.3.1 and (IV.33)). This modifies the definition of the sets S and $\tilde{S}$ as well. The set $\tilde{S}$ is 'punctured' with d values due to the addition of the $\{\tilde{\phi}^b{}_{0,k}\}_{k=0}^{d-1}$.

Currently, the software implementation does not support a delay-normalized generalization of the results in [16]. There are two reasons for this. First of all, keeping the primal boundary functions fixed is not compatible with adaptability to the input length, one of the main features of the software implementation (and there is no consensus on how to combine the strategy from [16] with absorption of inner functions in the boundary functions). Secondly, one would need some canonical boundary functions at the primal side in the general delay-normalized case, similarly to those obtained from the Schoenberg Spline basis in the Spine case.

IV.8 Notes on the implementation

This paper spares the reader for many tedious calculations needed for the software implementation. The necessary details for this can be found in the technical report [2], and can be summarized as follows.

- What are the smallest vector sizes so that a DWT/IDWT is possible (i.e., so that the left and right boundary functions do not interact)?
- Adoption of a lifting-based approach to the interval.
- Preconditioning (this was addressed in [7] for the orthogonal case, but [2] addresses this more generally).
- One has freedom in how to scale the modified boundary functions. [2] computes Gramm matrices as described in [17], in order to find the norms of the modified functions, and uses these to scale them accordingly.
- Computation of the Gramm matrices when $N \neq \tilde{N}$.

All code related to the paper can be found at https://github.com/oyvindry/wl

References

[1] Andersson, L., Hall, N., Jawerth, B. and Peters, G. 'Wavelets on Closed Subsets of the Real Line'. In: *Topics in the Theory and Applications of Wavelets* (1994), pp. 1–61.

[2] Antun, V. and Ryan, Ø. *Technical report for the paper "On the unification of schemes for wavelets on the interval"*. Tech. rep. Link: http://folk.uio.no/oyvindry/waveletsinterval/technicalreport.pdf. UIO, 2019.

[3] Buckheit, J., Chen, S., Donoho, D. L., Johnstone, I. and Scargle, J. 'About wavelab'. In: *Handbook of WaveLab Version* vol. 850 (1995), pp. 1–37.

[4] Carnicer, J. M., Dahmen, W. and Peña, J. M. 'Local decomposition of refinable spaces and wavelets'. In: *Applied and Computational Harmonic Analysis* vol. 3, no. 2 (1996), pp. 127–153.

[5] Chui, C. K. and Quak, E. 'Wavelets on a bounded interval'. In: *Numerical methods in approximation theory, Vol. 9*. Springer, 1992, pp. 53–75.

[6] Cohen, A., Daubechies, I. and Feauveau, J.-C. 'Biorthogonal bases of compactly supported wavelets'. In: *Communications on Pure and Applied Mathematics* vol. 45, no. 5 (June 1992), pp. 485–560.

[7] Cohen, A., Daubechies, I. and Vial, P. 'Wavelets on the interval and fast wavelet transforms'. In: *Applied and Computational Harmonic Analysis* vol. 1, no. 1 (1993), pp. 54–81.

[8] Dahmen, W., Kunoth, A. and Urban, K. 'Biorthogonal Spline Wavelets on the Interval – Stability and Moment Conditions'. In: *Applied and Computational Harmonic Analysis* vol. 6, no. 2 (1999), pp. 132–196.

[9] Dahmen, W., Kunoth, A. and Urban, K. 'Wavelets in numerical analysis and their quantitative properties'. In: *Proceedings of Chamonix*. 1996, p. 1.

[10] Daubechies, I. 'Orthonormal bases of compactly supported wavelets'. In: *Communications on Pure and Applied Mathematics* vol. 41, no. 7 (Oct. 1988), pp. 909–996.

[11] Gataric, M. and Poon, C. 'A practical guide to the recovery of wavelet coefficients from Fourier measurements'. In: *SIAM Journal on Scientific Computing* vol. 38, no. 2 (2016), A1075–A1099.

[12] Jacobsen, R. D., Nielsen, M. and Rasmussen, M. G. 'Generalized sampling in Julia'. In: *arXiv:1607.04091* (2016).

[13] Levaggi, L. and Tabacco, A. 'Wavelets on the interval and related topics'. In: *Rendiconti del Seminario Matematico* vol. 2 (Jan. 1999).

[14] Lyche, T. *Numerical Linear Algebra and Matrix Factorizations*. Springer, 2020.

[15] Meyer, Y. 'Ondelettes sur l'intervalle'. In: *Revista Matematica Iberoamericana* vol. 7, no. 2 (1991), pp. 115–133.

[16] Primbs, M. ‘New Stable Biorthogonal Spline-Wavelets on the Interval’. In: *Results in Mathematics* vol. 57, no. 1-2 (2010), pp. 121–162.

[17] Primbs, M. ‘On the computation of Gramian matrices for refinable bases on the interval’. In: *International Journal of Wavelets, Multiresolution and Information Processing* vol. 6, no. 03 (2008), pp. 459–479.

[18] Primbs, M. *Technical Report for the Paper: ’New stable biorthogonal Spline-wavelets on the Interval’.* Tech. rep. 2015.

[19] Ryan, Ø. *Linear algebra, signal processing, and wavelets. A unified approach. MATLAB version.* Springer, 2019.

[20] Ryan, Ø. *Linear algebra, signal processing, and wavelets. A unified approach. Python version.* Springer, 2019.

[21] Schumaker, L. L. *Spline Functions: Basic Theory.* Wiley, New York, 1981.

[22] Taubman, D. S. and Marcellin, M. W. *JPEG2000 Image Compression. Fundamentals, Standards and Practice.* Kluwer Academic Publishers, 2002.