Seismic reflectivity inversion using deep learning and model-based methods

by

Kristian David Torres Bautista

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Geophysics

Department of Physics
University of Alberta

Abstract

This thesis advances seismic imaging and inversion, crucial for identifying hydrocarbon prospects and understanding the Earth's internal structure by retrieving rock parameters from seismic data. Facing challenges such as solution non-uniqueness, slow convergence, and high computational demands, this work integrates deep learning frameworks, classical regularization theory, and two-way wave equation propagators. We develop iterative and non-iterative deep learning methods, such as interlacing convolutional neural networks (CNNs) within traditional Least-Squares Reverse Time Migration (LSRTM) schemes, and employing deep autoencoders to refine inverse problem spaces, enhancing resolution and speeding up convergence. Additionally, Sparse Full Waveform Least-Squares Reverse Time Migration (Sparse FWLSRTM) is introduced, combining sparse regularization with the full wavefield vector reflectivity modeling engine to substantially improve imaging quality under complex geological settings. Furthermore, the thesis explores seismic broadband deconvolution using deep learning to derive full-band reflectivity from band-limited data, integrating learned null space components for better data consistency and resolution. Collectively, these methodologies significantly enhance the fidelity and efficiency of seismic imaging, merging advanced machine learning techniques with traditional approaches to offer a robust toolset for geophysical prospecting in challenging subsurface environments.

Preface

A version of Chapter 2 of this thesis has been published as a journal article: Torres, K., and M. D. Sacchi, 2022, Least-squares reverse time migration via deep learning-based updating operators, Geophysics, 87, no. 6, S315–S333.

A version of Chapter 3 has been published as a conference proceeding: Torres, K. and Sacchi, M., 2023, A deep-learning inverse Hessian preconditioning for iterative least-squares migration. In 84th EAGE Annual Conference & Exhibition (Vol. 2023, No. 1, pp. 1-5).

A version of Chapter 4 has been published as a conference proceeding: Torres, K. and Sacchi, M., 2024, Sparse vector reflectivity inversion with full-wavefield least-squares reverse time migration. In 85th EAGE Annual Conference & Exhibition (Vol. 2024, No. 1, pp. 1-5). An expanded version will be submitted for publication to Geophysical Prospecting in July 2024.

A version of Chapter 5 of this thesis has been published as a journal article: Torres, K. and Sacchi, M. D., 2023, Deep decomposition learning for reflectivity inversion, Geophysical Prospecting, 71(6), pp.963-982.

In the publications mentioned, I was the main author and the developer responsible for the coding aspects of the work. Dr. Sacchi contributed as the supervising author, offering direction for the project and assistance with editing the manuscripts.

To the memory of Professor Franciane Conceição Peters.

Acknowledgements

I must begin by expressing my deepest gratitude to Professor Mauricio D. Sacchi. Having him as my supervisor has been a rewarding and life-changing experience. His expertise, guidance, and support have been instrumental in my journey. Mauricio's passion for research is both captivating and inspiring. Every conversation with him offers new insights. I greatly admire his dedication to understanding and mastering complex yet relevant topics. Thank you for trusting me as a researcher, allowing me to pursue the ideas that sparked my curiosity, and providing me with many opportunities. Gracias, Boss!

I am grateful to my examination committee members, Professors Gabriel Fabien-Ouellet, Nilanjan Ray, Vadim Kravchinsky, and Jeff Gu, for their insightful comments and valuable feedback.

I sincerely appreciate all my colleagues and friends in the Signal Analysis and Imaging Group and the University of Alberta. You have made this experience more bearable, enriching, and fun. Special mentions go to Breno Bahia, Gian Matharau, Yi Guo, Fernanda Carozzi, Andres Ambros, Alejandro Quiaro, Joaquín Acedo, Zoë Vestrum, and Mauricio Reyes Canales. I am also thankful to the SEG and Alberta Innovates for providing scholarship funding during my PhD studies.

Heartfelt appreciation goes to all my mentors who have significantly contributed to my work and development as a scientist. In particular, I am grateful to Konstantin Osypov and the research team at Halliburton for allowing me to work alongside versatile geoscientists during my internship. I would also like to acknowledge my former MSc supervisors, Professors Webe João Mansur and Franciane Conceição Peters. I learned a great deal about inverse problems and numerical modelling from them. Although Professor Franciane left us too soon, I will always be grateful for her trust, motivation, and the profound impact she had on my life. I often find myself contemplating ideas and topics I wish to discuss with her or seek her advice on. Her unwavering determination and work ethic still inspire me today. Thank you for everything.

I had to climb many mountains to reach this point. Each challenge was an opportunity

for self-discovery and personal growth while constantly reminding me of the support and importance of those around me. With this in mind, I will always be grateful to the Peters and Yanagihara families for being my family in Edmonton and São Paulo, respectively. Your friendship is a testament to God's love and is a true blessing.

Finally, I want to acknowledge my parents, Fernando and Soraya, my sister, Karina, and my cousin, Daniel. Thank you for your unconditional love, advice, care, and encouragement and for always being present, even from a distance, especially in the most challenging times. Your support means everything to me.

Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Seismic imaging and inversion are pivotal techniques for geophysicists, enabling them to uncover valuable hydrocarbon prospects and delve into the Earth's inner structure. These methods involve retrieving rock parameters and creating detailed subsurface images from recorded seismic data. At their core, they tackle a large-scale inverse problem: inferring an unknown model (or image)—a discrete representation of the subsurface, typically with up to a billion unknowns—from indirect and noisy measurements, known as observed data. The underlying inverse process reverses the causality direction of a known physical phenomenon embodying the seismic experiment, which derives observed data from a subsurface model. This is typically accomplished using a forward operator that simulates how seismic waves travel through the subsurface, interact with different geological structures, and then return to the surface, where they are recorded by receivers.

Solving the forward problem lays the foundation for approaching the corresponding inverse problem. Researchers have pioneered a variety of algorithms based on different approximations of a particular wave equation (Tarantola, 1984a; Woodward, 1992; Kühl and Sacchi, 2003; Sava and Biondi, 2004). The methods grounded in the two-way wave equation stand out for their enhanced precision in simulating wave propagation physics beneath the surface through a partial differential equation (PDE). This accuracy is especially beneficial in complex geological settings characterized by pronounced velocity contrasts and steep structural dips, where more simplistic methods might struggle to provide reliable results. In such contexts, a comprehensive treatment of wave phenomena is crucial for enhancing resolution and achieving higher imaging fidelity. This means that finer details within the subsurface can be resolved, leading to clearer and more detailed structures, which is crucial for identifying potential resources and understanding geological formations.

By the same token, the seismic imaging industry has leaned heavily on the principle of

acoustic approximations, treating the Earth as a fluid medium that exclusively supports the propagation of primary waves. This principle, rooted in the observation that both acoustic and elastic approximations yield indistinguishable travel times for primary waves, has shaped the standard approach to seismic data processing and imaging (Aki and Richards, 2002). Despite the oversimplified view of Earth's layers as shear-resistant fluids, this conception accurately predicts the arrival times of primary waves and provides an approximation of amplitude decay resulting from geometric spreading. Therefore, even though the acoustic approximation may technically employ an inexact model of Earth's physics, it consistently achieves remarkable success in seismic imaging applications where elastic assumptions are still computationally prohibitive. Building on this foundation, the majority of this thesis aims to leverage the acoustic approximation framework derived from the two-way wave equation, seeking to further advance subsurface imaging techniques.

Unlike the forward problem, solving the inverse problem is an unstable and inconsistent process. This instability arises because the subsurface can only be accessed from one side, that is, from the surface and/or a very limited number of well locations, with finite-aperture receiver arrays, leading to inadequate coverage in certain areas. Hence, there is not enough information in the data to uniquely determine a plausible subsurface model. As a matter of fact, many different subsurface models can result in identical surface measurements, a challenge that introduces a non-trivial null space—a family of model components that have no effect on the data—into the problem, and highlights the severe ill-posed nature of seismic inversion. This issue is exacerbated by the fact that the seismic sources used in field recordings are band-limited, restricting deep exploration and diminishing detail resolution due to the frequency gap. Moreover, the observed data is prone to noise and can only be partially reproduced due to approximation errors in the adopted forward modeling with respect to the true physics of the seismic experiment, which further complicates the task of model estimation. In the realm of seismic imaging, this translates into results with non-illuminated (shadow) regions, inconsistent amplitudes, imaging artifacts and low resolution.

A partial solution to address the inherent ill-posedness of inverse problems involves integrating prior information about the image space to achieve unique and enhanced results. Due to the problem size in seismic inversion, this integration is typically realized through regularization strategies within PDE-constrained optimization methods applied to least-squares imaging formulations. These methods iteratively minimize a cost function indicating the discrepancy between measured and predicted data derived from the established forward model, which is often a linear operator. Desirably, the null spaces of the regularizer and the forward operator should not overlap to reduce the null-space ambiguity. Moreover, the regularized reconstruction is deterministic, producing a single subsurface model that approximately fits the noisy seismic data, among various possible models. It is important to note that under specific conditions, this optimization problem can also be framed within a Bayesian context.

In this probabilistic interpretation, the solution that minimizes the optimization problem aligns with the maximum a posteriori (MAP) parameter estimate.

Additionally, preconditioning is a common technique employed alongside or in lieu of regularization. This involves encoding additional information into the solution space by enforcing features in specific mathematical bases. From a linear algebra perspective, preconditioning the system derived from the inverse problem formulation enhances its spectral properties, leading to accelerated convergence due to the improved distribution and clustering of large eigenvalues (Saad, 2011). Therefore, optimization methods will require fewer iterations to minimize the cost function. Moreover, the exclusive use of preconditioning allows for the avoidance of explicitly including extra regularization into the cost function, which saves the time needed to determine an appropriate trade-off parameter. When using popular semi-iterative solvers, the number of iterations also play the role of regularizing the data fitting process. A carefully chosen stopping point ensures that the iteration halts before the smaller singular values can significantly affect the solution (Hanke and Hansen, 1993).

Traditionally, regularization and preconditioning in seismic imaging have relied on hand-crafted operators, employing techniques such as sparsity enforcement or the application of Tikhonov and Total Variation methods (e.g. Anagaw and Sacchi, 2012). These methods aim to establish structured priors based on predefined notions of what constitutes a plausible subsurface model. However, despite their utility, regularized iterative approaches often struggle with the complicated nature of seismic data, leading to unrealistic reconstructions that fail to capture complex subsurface parameter statistics, slow convergence rates, and significant computational burdens due to the high costs of applying imaging operators. Although hand-designed priors can mitigate some artifacts, they often fall short in capturing realistic details and can be challenging to define due to their domain-specific nature.

To address these challenges, there has been a significant shift towards the use of learned-based operators, driven by advancements in deep learning technologies. These modern methods utilize extensive training data to learn complex prior distributions, enabling a more nuanced exploration of underlying geological features. In this thesis, leveraging deep learning frameworks, in particular through the application of Convolutional Neural Networks (CNN), we explore how seismic imaging can benefit from adaptive learning directly from the data itself. This results in tailored reconstruction schemes that dynamically adjusts to the specific intricacies of each dataset. Under the premise that we can use machine learning to extract this information from data efficiently, we develop solid protocols that integrate deep learning frameworks to solve complex geophysical inverse problems in imaging. Specifically, by combining theoretical concepts of wave propagation phenomena, seismic data processing, high-performance imaging techniques, and novel ideas arising from the field of deep learning, we study how data-driven and model-based methods can achieve substantial evolution

in seismic reflectivity imaging techniques, offering faster convergence and more accurate reconstructions of the subsurface.

As a starting point and for the sake of clarity, the next section presents the fundamental mathematical expressions of wave-equation-based seismic imaging. The remaining of this introduction briefly explains the novelty and the main contributions that this thesis brings to the field.

1.1 Seimic imaging

Model-based reconstruction approaches use the forward and adjoint operators of the imaging problem directly in the inversion algorithm, with a regularization scheme that implements handcrafted operators based on prior knowledge about the model space, or whose parameters are learned by a trainable operator from a dataset in a training phase. These methodologies are crucial in enhancing the fidelity of seismic images by ensuring that the adjustments made to the subsurface model are both accurate and reflective of the true geological structures.

The ultimate goal of seismic imaging is to retrieve a high-resolution image of the subsurface reflectivity. This process is conducted using reflection migration algorithms that utilize a previously estimated background velocity model and recorded data as inputs. Depth migration algorithms geometrically relocate the recorded seismic measurements to the depth locations where the reflection events originated in the subsurface, effectively collapsing the wavefronts to enhance the image of structures, provided that the background velocity model is sufficiently accurate. Demigration, on the other hand, reverses this process, taking an image and predicting how it would appear in recorded seismic data. In acoustic wave-equation imaging techniques, velocity is commonly used as the model parameter. In this context, the subsurface reflectivity model can be defined as the high-wavenumber velocity perturbation (Tarantola, 1984a). Therefore, we employ the acoustic wave equation to represent the underlying physical phenomena that links the mentioned reflectivity parameter to the observed reflection data.

In this section, we briefly describe the theory behind wave-equation-based seismic imaging. We begin by writing the scalar wave equation for both continuous and discrete cases. Next, we establish the linearized versions of the scalar wave equation and construct the associated forward and adjoint Born operators. These constitute the essential operators used in imaging techniques such as Reverse Time Migration (RTM) and the convex optimization problem known as Least-squares Reverse Time Migration (LSRTM), discussed in Chapters 2, 3, and 4 of this thesis.

1.1.1 The acoustic wave equation

The wave equation can be written in its acoustic isotropic (i.e., scalar) form as the following PDE:

$$\frac{\partial^2 p(\mathbf{x},t)}{\partial t^2} - v_p^2(\mathbf{x})\rho(\mathbf{x})\nabla\cdot\left(\frac{1}{\rho(\mathbf{x})}\nabla p(\mathbf{x},t)\right) = s(\mathbf{x},t), \tag{1.1}$$

where $\mathbf{x}$ denotes an observation point with Cartesian coordinates (x, y, z) in a three-dimensional space, $p(\mathbf{x}, t)$ is the pressure wavefield, $v_p(\mathbf{x})$ is the seismic velocity field, $\rho(\mathbf{x})$ is the density, $s(\mathbf{x}, t)$ is the source wavefield, ∇ is the gradient operator, and $\nabla\cdot$ is the divergence. Under the assumption of a spatially constant density function ρ, we can further simplify the wave equation and characterize the subsurface by a single parameter, namely the squared slowness $m = 1/v_p^2$, becoming

$$m(\mathbf{x})\frac{\partial^2 p(\mathbf{x},t)}{\partial t^2} - \nabla^2 p(\mathbf{x},t) = s'(\mathbf{x},t), \tag{1.2}$$

with $s' = s/v_p^2$. This simplification allows a more computationally efficient way to model seismic wavefields.

In chapters 2 and 3 of this thesis, we restrict ourselves to the setup of constant-density acoustics. In chapter 4, we present a reparameterization of equation 1.1 in terms of the acoustic velocity and vector reflectivity variables. We also limit our analysis to 2D acoustic models, but the ideas presented hereafter apply in principle to virtually any class of earth models.

One common additional assumption is to consider the source term as a point source described by a distinct source signature $w(t)$. For example, one can express the source term in equation 1.2 as

$$s'(\mathbf{x},t) = \delta(\mathbf{x}-\mathbf{x}_s)w(t), \tag{1.3}$$

where δ signifies the Dirac delta, and $\mathbf{x}_s$ denotes the location of the source. In practice, the wavefield variable p is not known across all subsurface points; rather, it is captured by receivers like geophones or hydrophones. The recorded data $d(\mathbf{x}_r, t)$ are defined by

$$d(\mathbf{x}_r,t) = \int_\Omega \delta(\mathbf{x}-\mathbf{x}_r)p(\mathbf{x},t)d\mathbf{x}, \tag{1.4}$$

where $\mathbf{x}_r$ is the coordinate of the receivers and Ω signifies the propagation domain.

To synthetically model the wave equation, the previously stated PDEs are discretized using an explicit finite-difference (FD) approach. For the constant density acoustic wave equation, a central-grid finite-difference technique is usually employed, leading to the expression

$$\left[\mathbf{M}\mathbf{D}_t^2 - \boldsymbol{\nabla}^2\right]\mathbf{p} = \mathbf{s}, \tag{1.5}$$

where $\mathbf{D}_t^2$ is a second-order central-difference operator in time, $\boldsymbol{\nabla}^2$ denotes a high-order finite difference representation of the Laplacian operator, and $\mathbf{M}$ is a diagonal matrix that depends on the squared slowness vector $\mathbf{m} \in \mathbb{R}^{N_n}$. The vectorized pressure wavefield $\mathbf{p}$ and the source vector $\mathbf{s}$ are both part of $\mathbb{R}^N$ with $N = N_n \times N_t$, being $N_n = N_x \times N_y \times N_z$ the number of grid-points of the discrete spatial dimensions, and N_t is the number of time samples. The process that describes the propagation of the source vector in the computational domain can be encapsulated as

$$\mathbf{p} = \mathbf{A}^{-1}(\mathbf{m})\mathbf{s}. \tag{1.6}$$

Here, $\mathbf{A}(\mathbf{m})$ represents a square matrix containing the operator $\left[\mathbf{M}\mathbf{D}_t^2 - \boldsymbol{\nabla}^2\right]$, which can be inverted with a forward substitution strategy when initial time-boundary conditions are defined. Finally, we can represent the discretization of the delta function in equation 1.4 for a single-source experiment as

$$\mathbf{d}(\mathbf{m}) = \mathbf{P}_r\mathbf{p}(\mathbf{m}), \tag{1.7}$$

with $\mathbf{d} \in \mathbb{R}^{N_r \times N_t}$ as the discrete observed data vector, and $\mathbf{P}_r \in \mathbb{R}^{(N_r \times N_t) \times N}$ acting as a "projection-onto-receivers" operator. N_r represents the number of receivers. Acoustic data modeling for a multi-source experiment can be easily extended as

$$\begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_{N_s} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{r_1}\mathbf{p}_1(\mathbf{m}) \\ \vdots \\ \mathbf{P}_{r_{N_s}}\mathbf{p}_{N_s}(\mathbf{m}) \end{bmatrix}, \tag{1.8}$$

where $\mathbf{P}_{r_i}$ and $\mathbf{p}_i$ refer to the sampling operator and simulated wavefield vector for the i-th source, respectively.

1.1.2 Migration and demigration operators

The industry-standard approach for seismic reflection imaging and inversion relies upon the linearized model of acoustic single scattering, which results from applying a suitable first-order perturbation approximation to the acoustic wave equation. Finding the linearized form of equation 1.2 facilitates the linear mapping of wave velocity perturbations to seismic data perturbations. Once the linearized acoustic wave equation is discretized, its adjoint form can be derived, establishing a linear relationship between data perturbations and wave velocity perturbations. This connection enables imaging through two-way wave-equation-based algorithms like RTM and the inversion of wave velocity and reflectivity models from seismic data using Full Waveform Inversion (FWI) and LSRTM, respectively.

To express the linear operators, we start with a key and widespread assumption of seismic migration and linearized waveform inversion: the true velocity model, parameterized in terms of the squared slowness, m_{true}, admits a scale separation as a sum of a low-wavenumber component (the true background model), m_0, and a high-wavenumber component (the true reflectivity model), δm, so that the following equation holds

$$m_{\text{true}}(\mathbf{x}) = m_0(\mathbf{x}) + \delta m(\mathbf{x}). \tag{1.9}$$

Respectively, the full pressure wavefield for a single-source experiment, p, can be decomposed as the sum of a background wavefield, p_0, and a perturbed wavefield, δp,

$$p(\mathbf{x}, t) = p_0(\mathbf{x}, t) + \delta p(\mathbf{x}, t), \tag{1.10}$$

such that we can rewrite the acoustic wave equation as

$$[m_0(\mathbf{x}) + \delta m(\mathbf{x})]\frac{\partial^2[p_0(\mathbf{x}, t) + \delta p(\mathbf{x}, t)]}{\partial t^2} - \nabla^2[p_0(\mathbf{x}, t) + \delta p(\mathbf{x}, t)] = s'(\mathbf{x}, t). \tag{1.11}$$

Expanding equation 1.11 and considering that

$$\delta m\frac{\partial^2(p_0 + \delta p)}{\partial t^2} = \delta m\frac{\partial^2 p}{\partial t^2} \tag{1.12}$$

$$\approx \delta m\frac{\partial^2 p_0}{\partial t^2} \tag{1.13}$$

via Born's approximation, results in the following system

$$m_0(\mathbf{x})\frac{\partial^2 p_0(\mathbf{x}, t)}{\partial t^2} - \nabla^2 p_0(\mathbf{x}, t) = s'(\mathbf{x}, t), \tag{1.14}$$

$$m_0(\mathbf{x})\frac{\partial^2 \delta p(\mathbf{x}, t)}{\partial t^2} - \nabla^2 \delta p(\mathbf{x}, t) = -\delta m(\mathbf{x})\frac{\partial^2 p_0(\mathbf{x}, t)}{\partial t^2}. \tag{1.15}$$

This approximation assumes that the reflections are due to local discontinuities of the medium only, and the effect of multiple reflections will be neglected.

Similar to equation 1.4, the registered scattered wavefield at the receivers position is expressed as

$$\delta d(\mathbf{x}_r, t) = \int_\Omega \delta(\mathbf{x} - \mathbf{x}_r)\delta p(\mathbf{x}, t)\, d\mathbf{x}. \tag{1.16}$$

Discretizing equations 1.14 to 1.16, we define the linear operator that connects a squared slowness perturbation vector $\mathbf{\Delta m} \in \mathbb{R}^{N_n}$ with a perturbation in the data space $\mathbf{\Delta d} \in \mathbb{R}^{N_m}$

as follows:

$$\Delta \mathbf{d} = \mathbf{P}_r \mathbf{A}^{-1}(\mathbf{m}_0) \ddot{\mathbf{P}}_0(\mathbf{m}_0) \Delta \mathbf{m}, \tag{1.17}$$

$$= \mathbf{P}_r \Delta \mathbf{p} \tag{1.18}$$

where $\mathbf{m}_0 \in \mathbb{R}^{N_n}$ is the background squared slowness, and $\ddot{\mathbf{P}}_0 \in \mathbb{R}^{N \times N_n}$ is a matrix representing the discretization of the term $-\frac{\partial^2 p_0(\mathbf{x},t)}{\partial t^2}$. This matrix is derived from taking the second derivative and scaling the solution p_0 in equation 1.14, often referred to as the source or background wavefield in migration jargon, obtained through the solution of the system

$$\mathbf{p}_0 = \mathbf{A}^{-1}(\mathbf{m}_0)\mathbf{s}. \tag{1.19}$$

The discrete scattered wavefield can be obtained by following an analogous process via the finite-difference and forward-substitution approach but using the product between the squared slowness perturbation and the scaled background wavefield as the source term, which results in

$$\Delta \mathbf{p} = \mathbf{A}^{-1}(\mathbf{m}_0) \ddot{\mathbf{P}}_0(\mathbf{m}_0) \Delta \mathbf{m}. \tag{1.20}$$

The linear mapping between perturbations in the model and data space can be explictly obtained by rewriting equation 1.17 as

$$\Delta \mathbf{d} = \mathbf{L} \Delta \mathbf{m}, \tag{1.21}$$

with $\mathbf{L} \in \mathbb{R}^{N_m \times N_n}$ defined as

$$\mathbf{L} = \mathbf{P}_r \mathbf{A}^{-1}(\mathbf{m}_0) \ddot{\mathbf{P}}_0(\mathbf{m}_0), \tag{1.22}$$

commonly known as Born modeling or demigration operator. Dropping the non-linear dependency on the background model for notation simplicity, we can write the adjoint of the Born operator, also known as the RTM operator, $\mathbf{L}^T \in \mathbb{R}^{N_n \times N_m}$, as

$$\mathbf{L}^T = \ddot{\mathbf{P}}_0^T (\mathbf{A}^T)^{-1} \mathbf{P}_r^T. \tag{1.23}$$

Equations 1.22 and 1.23 are applied throughout this thesis (unless otherwise noted) as the algorithms to generate seismic pressure data from seismic images (demigration) and create seismic images from the observed reflection data (migration). The latter is accomplished by

solving

$$\begin{aligned}\mathbf{m}_{\text{RTM}} &= \mathbf{L}^T\mathbf{d} \\ &= \ddot{\mathbf{P}}_0^T(\mathbf{A}^T)^{-1}\mathbf{P}_r^T\mathbf{d} \\ &= \ddot{\mathbf{P}}_0^T\mathbf{q}\end{aligned} \tag{1.24}$$

Breaking down the individual terms that result in equation 1.24, it is possible to obtain a physical interpretation of how the adjoint of the Born operator can produce an image of the subsurface. This system is equivalent to a cross-correlation operation between two wavefields. The first wavefield is the previously introduced scaled source wavefield $\ddot{\mathbf{P}}_0$. The second wavefield, $\mathbf{q} \in \mathbb{R}^N$, is usually known as the receiver or adjoint wavefield, and it is given by the solution of the system

$$\mathbf{A}^T\mathbf{q} = \mathbf{P}_r^T\mathbf{d}. \tag{1.25}$$

As noted in equation 1.25, the receiver wavefield originates by injecting the recorded reflections with the adjoint sampling operator, $\mathbf{P}_r^T$, and propagating them backwards in time with the adjoint operator of the acoustic wave equation, $\mathbf{A}^T = \left[\mathbf{M}\mathbf{D}_t^2 - \boldsymbol{\nabla}^2\right]^T$, which can be obtained via the adjoint-state method (Plessix, 2006a). The key to this imaging technique is that, ideally, the two wavefields only generate a non-zero image where they meet exactly at the same point in time and space (Claerbout, 1985). In the continuous case, we can write

$$m_{\text{RTM}}(\mathbf{x}) = -\int_T \frac{\partial^2 p_0(\mathbf{x},t)}{\partial t^2} q(\mathbf{x},t)dt, \tag{1.26}$$

making the image generation process explicit through the time and space zero-lag cross-correlation between the two wavefields. As extensively advocated in the literature, the true high-wavenumber reflectivity image can be more accurately estimated using LSRTM for a fixed background model $\mathbf{m}_0$. This linearized inversion algorithm approximates the pseudo-inverse of the Born forward operator. It mitigates issues such as the degradation of amplitude and the presence of low-frequency artifacts commonly associated with the adjoint operator. A detailed explanation of this method will follow in the subsequent chapters.

1.2 Thesis contributions and overview

This thesis contributes to understanding the advantages and limitations of model-based and data-driven deep learning approaches to the seismic imaging problem, with a special focus on Convolutional Neural Networks (CNN). The main contributions of each chapter are as follows:

- In Chapter 2, we explore the application of learned iterative reconstruction methods to LSRTM on synthetic and real data sets. We demonstrate the potential and limitations of supervised single-pass post-processing and cascade-like schemes for CNN-based projection approaches in data-driven LSRTM inversion. We develop a lightweight and practical CNN architecture, therefore omitting the need for over-parameterized networks, considering the physics of seismic migration using the least-squares misfit gradient and the current reflectivity estimate. We also explore how data-driven reflectivity reconstructions can be used as warm-starts for the CGLS algorithm, obtaining an improved reflectivity model in fewer iterations than traditional LSRTM.

- The first part of Chapter 3 proposes a model-space self-supervised learned preconditioner for data-domain iterative LSRTM. The experimental method uses a deep CNN autoencoder to learn the mapping between migrated and re-migrated images in seismic data. This method is proposed to approximate the inverse Hessian, which is then used to re-parameterize the reflectivity model into a low-dimensional space. This approach facilitates a more efficient solution of the linearized waveform inversion, thereby accelerating the convergence of LSRTM and reducing computational costs.

- The remaining of Chapter 3 explores a novel non-iterative self-supervised approach to improve RTM images using deep-learning-based preconditioning in the data space. By employing a CNN that is trained on input-output pairs of observed and demigrated data, the method applies non-linear filters directly to the migrated data. This approach balances image quality with computational efficiency, eliminating the need for the iterative computation typical of LSRTM.

- Optimal RTM-based imaging and inversion algorithms are often contingent upon perfect seismic data acquisition and the premise that the seismic wave propagation conforms strictly to an acoustic wave equation without density variations. These assumptions, however, are only sometimes congruent with the complexities encountered in practical scenarios. Chapter 4 introduces Sparse Full Waveform Least-Squares Reverse Time Migration (Sparse FWLSRTM), an innovative approach to seismic imaging that incorporates sparse regularization to enhance the quality of seismic data imaging. By comparing Sparse FWLSRTM with traditional LSRTM methods, we demonstrate the former's superior capability to delineate geological reflectors with higher precision and reduce imaging artifacts and sidelobes, even when faced with highly decimated data and imprecise velocity models. Moreover, we show how FWLSRTM, with its parameterization based on vectorized reflectivity, can resolve a broader range of seismic events, including refractions, diving waves, and reflections, offering a superior fit for complex data compared to traditional methods without the need for an explicit density model. We outline the theoretical framework for Sparse FWLSRTM, including the

derivation of misfit gradients and the application of sparse regularization techniques. Through various experiments and comparisons, we illustrate the potential of Sparse FWLSRTM as a robust tool for seismic imaging in complex geological settings.

- Chapter 5 devises the application of learned range-nullspace decomposition methods for reflectivity inversion. In the training stage of this supervised framework, access to feasible physical projections onto the range and null spaces of a known forward operator is required. Due to the high computational costs in estimating such projections using Born operators, we explore the application of such networks in a less computationally expensive but still ubiquitous problem in seismic exploration: the recovery of a full-band reflectivity time series from band-limited zero-offset seismic traces. The proposed approach combines classical regularization theory with a learned deep range-nullspace decomposition. Namely, our method extends the learned post-processing approach by learning how to improve an initial reconstruction with estimated missing components from the null space of the forward operator, which, in our case, are the missing frequency components of the reflectivity. At the same time, we incorporate a denoising neural network that acts exclusively on the range space, approximately guaranteeing data consistency. Numerical experiments show that the proposed method naturally enforces a high-resolution prediction consistent with the low-resolution input seismic traces. We also compare the proposed technique with a classical thin-bed reflectivity estimation method on synthetic data and two real data sets.

CHAPTER 2

Least-squares reverse time migration via deep learning-based updating operators [1]

2.1 Introduction

Seismic migration workflows are the cornerstone of subsurface structural imaging. From the family of migration techniques implemented in such workflows, Reverse Time Migration (RTM) (Baysal et al., 1983; Whitmore, 1983) is considered the most effective algorithm for imaging complex geological settings. Its superiority in providing accurate subsurface images stems from the fact that it uses the adjoint of the linearized two-way wave equation, which is a more accurate approximation of the wave propagation phenomena than its asymptotic or one-way counterparts, thus imposing no limitations on the dip angle of reflectors (Etgen et al., 2009). Despite being a sophisticated migration method, RTM images often suffer from backscattering low-frequency noise (Díaz and Sava, 2016), incomplete illumination due to defocusing velocity anomalies (Buur and Kühnel, 2008), and artifacts produced by a coarse source-receiver sampling and corrupted data (Zhang and Sun, 2009), thereby exposing the limitations of using adjoint (rather than inverse) operators to map from seismic data to reflectivity models.

Least-squares migration (LSM) (Lailly and Bednar, 1983; Nemeth et al., 1999) seeks to overcome the limitations of adjoint formulations by posing seismic migration as a linear inversion problem, where a generalized inverse approximates the exact inverse of the modeling operator. Under the assumption of a sufficiently correct background velocity model and adequate

[1] A version of this chapter has been published as a journal article: Torres, K., and M. D. Sacchi, 2022, Least-squares reverse time migration via deep learning-based updating operators, Geophysics, 87, no. 6, S315–S333.

preconditioning of the data, LSM can reduce migration artifacts, remove acquisition marks, compensate for illumination, and increase the resolution of seismic sections (Lambaré et al., 1992; Kühl and Sacchi, 2003; Schuster, 2017). When least-squares migration uses the RTM engine to perform demigration/migration sequences for imaging, we call it least-squares reverse time migration (LSRTM) (Dai et al., 2012; Wang et al., 2017b). LSRTM delivers true amplitude images and sharpens the subsurface reflectors (Dong et al., 2012).

Conventional LSRTM implementations use iterative methods to obtain a reflectivity model that minimizes the l_2-norm misfit, defined in either the data domain, where the input is the recorded data, or the image domain, where the input is the migrated image (Fletcher et al., 2016; Schuster and Liu, 2019). Additionally, due to the ill-posed nature of the LSM problem, a regularization term is frequently incorporated into the misfit to impose a priori information on the model (such as sparseness or other structure constraints) and stabilize the inverted solution (e.g. Clapp, 2005; Wang et al., 2005; Wang and Sacchi, 2007a, 2009; Dutta, 2017; Witte et al., 2017; Li et al., 2020a). Regularized LSRTM methods often produce acceptable results but can be challenging to deploy in practice due to the difficulty of hyperparameter selection and the high computational cost of iteratively evaluating the forward and adjoint operators. Moreover, the selection and design of an appropriate regularization term remain highly nontrivial for complex reflectivity distributions, where a poor choice could compromise the correctness of the inversion. Such limitations open the way to a new generation of methods.

With the recent advent of deep learning (LeCun et al., 2015), machine learning techniques have emerged as powerful alternatives for solving seismic inverse problems with remarkable empirical success. One common approach in the field of velocity model building is to use deep neural networks (DNN) to directly approximate the data-to-model inverse mapping in an end-to-end fashion (e.g. Araya-Polo et al., 2018, 2019; Yang and Ma, 2019; Li et al., 2020b; Liu et al., 2021a; Vantassel et al., 2021). Although this approach circumvents the need for iterative reconstruction, it relies exclusively on vast amounts of sometimes elusive training data to achieve high-quality results and good generalization properties. It also scales poorly for models with a large number of trainable parameters due to memory limitations—an issue analogous to explicit matrix-based formulations of seismic imaging problems (Yao and Jakubowicz, 2016).

For these reasons, alternative strategies aim to combine data-driven applications with traditional inversion processes, often by substituting parts of the original computations or by adopting an unrolled implementation of an iterative algorithm. For instance, a category of deep learning methods bypasses the need for labeled datasets by using partial differential equation kernels to guide the unsupervised training of physics-based neural networks (PBNN), in which the seismic records are regarded as the training data, and the subsur-

face parameters are estimated through Convolutional (Biswas et al., 2019) or Recurrent (Richardson, 2018; Sun et al., 2020) Neural Networks. Alfarraj and AlRegib (2019) and Sun et al. (2021) extend PBNN to a hybrid approach that considers a weighted sum between model and data misfits as the components of the loss function, with the ability to learn from partially labeled data in semi-supervised training. These techniques highlight the benefits of integrating machine-learning operations with known deterministic physical models in the form of forward and backward wave propagation codes.

In recent years, deep learning also has been implemented into LSM workflows. Liu et al. (2020c) show that the iterative solution of the image-domain LSM problem with sparsity constraints resembles a forward pass through a multilayered neural network, establishing analogies between the filters and feature maps of a Convolutional Neural Network (CNN) and the migration Green's functions and reflectivity coefficients. Kaur et al. (2020) approximate the effect of the inverse Hessian on post-stack migrated images by training a Generative Adversarial Network (GAN) using the background velocity model as a conditioner. Most recently, Vamaraju et al. (2021) employs mini-batch gradients and adaptive learning rate optimizers to improve the resolution and reduce the computational burden of conventional pre-stack LSRTM. Further implementations of learning-based techniques in seismic imaging include using a CNN to suppress cross-talk artifacts from dip-angle gathers in elastic RTM images (Lu et al., 2020) and training CNNs as structural (Cheng et al., 2020) or denoising (Liu et al., 2020a) preconditioners to assist in the LSM inversion. For a comprehensive review of deep learning methods in other geophysical applications, we refer the reader to Yu and Ma (2021a), and Adler et al. (2021a).

Exploiting the universal approximation property of CNNs (Hornik et al., 1989) and the fact that they can learn structural features of images from representative examples, we introduce a supervised strategy that interfaces CNN-based machine learning with pre-stack LSRTM, which we have baptized as Deep-LSRTM. This scheme can be regarded as a deep learning extension of the conventional projected gradient-descent method since it substitutes the projection operators with sets of CNNs, learning an update function at each iteration. Our approach incorporates the linearized modeling and migration wave operators into the reconstruction process, evolving in response to the data-misfit gradient and allowing for interpretability in the network design. By detaching the forward and adjoint mappings from the learning step, Deep-LSRTM favorably reduces the networks' parameter complexity and the amount of training data, and also improves robustness and generalizability (Boink et al., 2019; Maier et al., 2019). The efficiency of the method is twofold: once trained, the proposed networks can predict accurate reflectivity models within only a few iterations, and secondly, it implicitly learns the step size and the effect of regularization from the training data. We emphasize that Chen et al. (2021a) have explored a similar idea based on learning generalized projection mappings in the context of blind high-resolution inversion of post-

stack data; however, to the best of our knowledge, this is the first fully iterative application of learned updating projections on seismic pre-stack LSRTM.

In the following sections, we will first describe how the theory of constrained LSRTM connects with Deep-LSRTM. Then we will present the proposed network architecture and the supervised training procedure, where we use 900 reflectivity models derived from folded and heavily faulted pseudo-random velocity distributions as true labels. After training, we evaluate the performance of Deep-LSRTM on three synthetic examples and compare it against a single-step learned reconstruction and a conventional LSRTM solved by the conjugate gradient least-squares algorithm (CGLS) (Hestenes and Stiefel, 1952). We also use field data from the Gulf of Mexico to test how a direct implementation of our method and a modification that includes transfer learning perform on a dataset with characteristics not perfectly represented by the training samples. We observe that our workflow produces high-resolution migration images comparable to traditional LSRTM but drastically reduces the required number of iterations. Finally, we present the conclusions and directions for future work.

2.2 Method

2.2.1 Least-squares reverse time migration

The forward problem of seismic imaging can be represented in a compact matrix notation by

$$\mathbf{d} = \mathbf{L}\mathbf{m}, \tag{2.1}$$

where $\mathbf{d}$ denotes the $M \times 1$ vector of modeled data, $\mathbf{m}$ is the $N \times 1$ vectorized reflectivity or model perturbation, and $\mathbf{L}$ is the $M \times N$ matrix that describes a linearized forward-modeling operator dependant on the acquisition geometry, the source wavelet, and a known background model of the medium (e.g., velocity) (Tarantola, 1984b). In this work, $\mathbf{L}$ defines the forward (or demigration) operator that encapsulates the first Born approximation of the acoustic two-way wave equation for a medium without density variations (Lailly and Bednar, 1983). Theoretically, RTM uses the adjoint of the Born modeling operator to migrate the pre-processed observed seismic data $\mathbf{d}_{\text{obs}}$

$$\mathbf{m}_{\text{mig}} = \mathbf{L}^T\mathbf{d}_{\text{obs}}, \tag{2.2}$$

where $\mathbf{m}_{\text{mig}}$ is the migrated image, an estimate of the true model perturbation (Xu and Sacchi, 2018). Seismic imaging is plagued by noisy data and a non-trivial null space due to bandwidth, illumination, and offset limitations. Thus, LSRTM is formulated as a linear

ill-posed inverse problem aided by regularization. Regularized LSRTM entails finding the reflectivity model that minimizes the mismatch between observed and modeled data in a least-squares sense and is consistent with the available prior knowledge. This amounts to solving the following unconstrained optimization problem

$$\min_{\mathbf{m}} \big\{ J(\mathbf{m}) + \lambda R(\mathbf{m}) = \frac{1}{2} ||\mathbf{L}\mathbf{m} - \mathbf{d}_{\text{obs}}||_2^2 + \lambda R(\mathbf{m}) \big\}, \tag{2.3}$$

where $J(\mathbf{m})$ is the data-misfit function, $||\cdot||_2$ is the l_2-norm that implicitly designates summation over time samples, receivers and sources, $R(\mathbf{m})$ represents any generic regularization/penalty term that encodes knowledge about the structure of the model, and $\lambda > 0$ is the damping parameter that controls the trade-off between the data misfit and the regularization term. Depending on the smoothness and convexity properties of the regularizer, a myriad of specialized optimization methods have been developed to obtain optimal solutions to different variants of equation 2.3. For instance, a common strategy consists in minimizing a non-quadratic norm on some sparsity-promoting representation of the reflectivity model or its gradient, which can be solved with proximal algorithms or the iterative-reweighted least-squares method (Wang and Sacchi, 2007b; Liu et al., 2021b). Although non-quadratic regularization approaches lead to high-resolution images, they still require many iterations, and using large regularisation weights might result in excessive smoothing and underestimated property values. Obtaining an optimal λ is also a cumbersome task and it is generally estimated by a trial-and-error approach. The more rigorous L-curve procedure is not feasible for industrial-scale LSRTM problems as it demands the solution of the inverse problem multiple times (Calvetti et al., 2000).

In the particular case of $R(\mathbf{m}) = \frac{1}{2}||\mathbf{m}||_2^2$ (i.e. assuming zero-order quadratic regularization), equation 2.3 leads to the classical LSRTM with damping. The latter has closed-form solution given by

$$\mathbf{m} = [\mathbf{L}^T\mathbf{L} + \lambda\mathbf{I}]^{-1}\mathbf{L}^T\mathbf{d}_{\text{obs}}, \tag{2.4}$$

which is equivalent to the image-domain LSRTM formulation

$$\mathbf{m} = \mathbf{H}^{-1}\mathbf{m}_{\text{mig}}. \tag{2.5}$$

Due to the prohibitive cost of explicitly inverting and storing the Hessian matrix plus prewhitening given by $\mathbf{H} = \mathbf{L}^{\mathrm{T}}\mathbf{L} + \lambda\mathbf{I}$, the operators $\mathbf{L}$ and $\mathbf{L}^T$ are constructed in implicit form rather than as matrices, and equation 2.4 is solved iteratively or by approximating the Hessian by a manageable size operator (Gao et al., 2020a). Considering a simple gradient descent scheme, we can calculate reflectivity updates by

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \alpha\nabla(J(\mathbf{m}_k) + \lambda\mathbf{m}_k), \tag{2.6}$$

where

$$\begin{aligned}\nabla J(\mathbf{m}_k) &= \mathbf{L}^T(\mathbf{L}\mathbf{m}_k - \mathbf{d}_{\text{obs}}) \\ &= \mathbf{L}^T\mathbf{r}\end{aligned} \tag{2.7}$$

denotes the gradient of the data misfit with respect to the model parameters, $\mathbf{r}$ is the data residual, $\alpha > 0$ is the step length, and k is the iteration number. In practice, we can find stable solutions to the LSRTM problem with the more efficient CGLS algorithm, which allows us to use an analytical step length, weighted l2-norms, and quadratic regularization (Kühl and Sacchi, 2003).

Constrained LSRTM via gradient projections

A different approach to regularized minimization arises when the prior knowledge is introduced in the form of projected constraints (Peters et al., 2019). In such a case, the inverse problem can be rephrased as

$$\min_{\mathbf{m}\in\mathcal{C}} \big\{ J(\mathbf{m}) = \frac{1}{2}||\mathbf{L}\mathbf{m} - \mathbf{d}_{\text{obs}}||_2^2 \big\}, \tag{2.8}$$

where $\mathcal{C} \subset \mathbb{R}^N$ is a closed, non-empty convex set with desired physical constraints. When an efficient method for projecting onto $\mathcal{C}$ is available, we can use the projected gradient descent method (PGD) as a cost-effective algorithm to iteratively solve equation 2.8

$$\mathbf{m}_{k+1} = \mathcal{P}_{\mathcal{C}}\Big(\mathbf{m}_k - \alpha\nabla J(\mathbf{m}_k)\Big), \tag{2.9}$$

with the projection operator formally defined by

$$\mathcal{P}_{\mathcal{C}}(\mathbf{x}) = \arg\min_{\mathbf{m}\in\mathcal{C}} \big\{ \frac{1}{2}||\mathbf{m} - \mathbf{x}||_2^2 \big\}, \tag{2.10}$$

which translates into finding the closest $\mathbf{m} \in \mathcal{C}$ to a given input model update $\mathbf{x}$. In practice, a variety of techniques might be employed as projections. For example, Cheng et al. (2016) use PGD with the Singular Spectrum Analysis filter (SSA) (Oropeza and Sacchi, 2011) as the projection operator to perform pre-stack LSM for blended data, where a deblending constraint is imposed in the shot-index common image domain to separate the coherent multiples from source cross-talk.

Since PGD converges at a linear rate under a constant step size, authors have explored the application of the projected gradient method to seismic waveform inversion (e.g. Becker et al., 2015; Xiang et al., 2016; Peters et al., 2019). In general, convergence properties of PGD are given for convex projections. However, as pointed out by Peters et al. (2019),

one can use non-convex projections and rely on empirical testing to examine convergence properties.

2.2.2 Deep-LSRTM: LSRTM via learned updates

We build the Deep-LSRTM framework upon recent research that suggests that unrolling classical model-based techniques with learned projection networks trained from data may significantly optimize inverse problem formulations (e.g. Gregor and LeCun, 2010; Chang et al., 2017; Meinhardt et al., 2017; Putzky and Welling, 2017; Zhang and Ghanem, 2017). In general, learned projection formulations avoid the explicit computation of regularized constraints. Instead, neural network architectures with K prescribed blocks are constructed, where each block mimics one iteration of a model-based gradient method. Naturally, the PGD scheme presented in the previous section inspires us to replace the projection operator $\mathcal{P}_{\mathcal{C}}(\cdot)$ or any generalized projection that improves the current model update by a neural network $\mathcal{P}_{\Theta}(\cdot)$, parametrized by the set of trainable weights $\Theta = [\theta_0, ..., \theta_{K-1}]$. With this analogy established, we arrive at our Deep-LSRTM formulation, which has appeared in the non-geophysical literature (Adler and Öktem, 2017; Hauptmann et al., 2018). Specifically, we modify the iterative solution of the PGD method (equation 2.9) with CNN updates, designed to encode the prior knowledge represented by the complex distribution of a training data manifold. We choose a CNN-based architecture because the convolution kernels and weight sharing structures allow the learned operators to exploit spatial correlations, naturally enforcing a regularization effect.

Deep-LSRTM iteratively estimates reflectivity updates using the following expression:

$$\mathbf{m}_{k+1} = \mathcal{P}_{\theta_k}(\mathbf{m}_k, \nabla J(\mathbf{m}_k)), \quad k = 0, ..., K-1, \tag{2.11}$$

where each $\mathcal{P}_{\theta_k} \subset \mathcal{P}_{\Theta} : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}^N$ is an updating function trained to map the pair of inputs $(\mathbf{m}_k, \nabla J(\mathbf{m}_k))$ to a single model update $\mathbf{m}_{k+1}$ that has similar characteristics to the training instances. Passing through the multiple set of blocks $\mathcal{P}_{\Theta}$ is analogous to executing the PGD algorithm with a finite number of K iterations. As can be inferred from equation 2.11, we let the updating operators learn how to combine the current model iterate with the gradient of the data-misfit term instead of explicitly computing a step length. The model features are also implicitly learned from the dataset during training rather than imposed as constraints or regularizing terms. This is a differentiating factor compared to traditional LSRTM, where it is often complicated to choose optimal projection or regularization strategies capable of expressing desirable geological and structural elements. Deep-LSRTM also avoids introducing model-dependent tuning parameters.

Compactly, we define each updating operator as N stacked 2D convolutional layers and

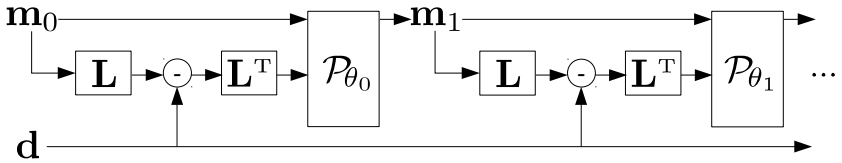

Figure 2.1: Unrolled diagram of the implemented iterative learned network for the first two iterations.

non-linear activation functions (see Appendix A). Additionally, we incorporate concatenation operators, residual connections, and batch normalization (Ioffe and Szegedy, 2015) to increase the expressivity of the network and to help gradient propagation in the training phase.

Figure 2.1 illustrates the first two iterations of the Deep-LSRTM framework. Different from end-to-end implementations, our approach does not need to learn the forward or adjoint operators. This would add complexity in terms of learnable parameters and poses hardware difficulties when working with large images. Instead, the wave operators are computed through the data-misfit gradient $\nabla J(\mathbf{m}_k)$ using equation 2.7 before feeding each updating block. This allows using shallower network architectures, which are less prone to overfitting and require less training data. Moreover, as empirically showed in Adler and Öktem (2017), and Putzky and Welling (2017), interweaving explicitly known operators in learned reconstruction schemes provides more detail to the solution and reduces the training error. The information passed through the data misfit gradient indicates where the seismic image needs improvement to fit the observed data, progressively refining the model at each iteration (Zeng et al., 2017). This means that, within the limitations of operator mismatches, $\nabla J(\mathbf{m})$ partially reconstructs the measurable components, whereas $\mathcal{P}_\Theta$ attempts to recover the null space components (limited aperture and missing frequencies), similar to the regularizer in classical optimization techniques.

In our approach, the inputs contain artifacts and illumination deficiencies. The admissible model update is the one that has reduced artifacts and boosted resolution. Hence, we train the network parameters to make efficient projections by minimizing a loss function between the predictions and the training dataset. Typically, in small scale deep learning inversions, all the parameters of unrolled schemes are trained jointly for all iterations (i.e., end-to-end). This provides an optimal set $\hat{\Theta}$ for the predefined maximum number of K iterations (Monga et al., 2021), provided that the ground-truth solutions to the optimization problem are available. Considering the mean squared error (MSE) loss, this yields the following

optimization problem

$$\begin{aligned}\hat{\Theta} &= \arg\min_{\Theta} \ \frac{1}{\Im}\sum_{i=1}^{\Im} ||\mathbf{m}_K^i - \mathbf{m}_{\text{true}}^i||_2^2, \\ &= \arg\min_{\theta_0,\ldots,\theta_{K-1}} \ \frac{1}{\Im}\sum_{i=1}^{\Im} ||(\mathcal{P}_{\theta_{K-1}} \circ \ldots \circ \mathcal{P}_{\theta_0}(\mathbf{m}_0^i, \nabla J(\mathbf{m}_0^i))) - \mathbf{m}_{\text{true}}^i||_2^2,\end{aligned} \tag{2.12}$$

where $\Im$ is the total number of training instances, and $\mathbf{m}_{\text{true}}^i$ indicates the ground-truth model of the i-th training instance. To solve equation 2.12, stochastic gradient methods commonly used in machine learning, compute an approximation of the MSE loss gradient with respect to the network parameters. For large-scale problems such as Deep-LSRTM, this approach is unfeasible since the loss gradient also requires the computation of $\nabla J(\mathbf{m}_k)$ for each evaluation of the loss function (i.e., multiple calls of $\mathbf{L}$ and $\mathbf{L}^T$ per training instance). To make Deep-LSRTM computationally feasible, we adopt a greedy training approach (Hauptmann et al., 2018), which is grounded in optimization but produces a solution that is only iterate-wise optimal. The greedy training defines a minimization problem for each individual set of parameters θ_k, preventing the gradients from flowing between the updating blocks by optimizing

$$\min_{\theta_k} \ \frac{1}{\Im}\sum_{i=1}^{\Im} ||\mathbf{m}_{k+1}^i - \mathbf{m}_{\text{true}}^i||_2^2 = \min_{\theta_k} \ \frac{1}{\Im}\sum_{i=1}^{\Im} ||\mathcal{P}_{\theta_k}(\mathbf{m}_k^i, \nabla \tilde{J}(\mathbf{m}_k^i)) - \mathbf{m}_{\text{true}}^i||_2^2. \tag{2.13}$$

Compared with equation 2.12, equation 2.13 only requires one call of $\mathbf{L}$ and $\mathbf{L}^T$ per training instance at each iteration k. In this framework, to increase network capacity, each $\mathcal{P}_{\theta_k}$ shares the same architecture as the other updating operators, but each operator has its own set of learned parameters. The updating operators are trained sequentially, and the parameters are fixed after training, meaning that the outputs predicted by an updating block $\mathcal{P}_{\theta_{k-1}}$ from a previous iteration will be part of the dataset used to train $\mathcal{P}_{\theta_k}$ in the next iteration. Finally, to gain substantial improvements in the imaging step, we substitute the raw data-misfit gradient by a preconditioned version (Xu and Sacchi, 2018), given by

$$\nabla \tilde{J}(\mathbf{m}_k) = \mathbf{D}\mathbf{P}_m\mathbf{L}^T\mathbf{P}_d\mathbf{r} \tag{2.14}$$

where $\mathbf{P}_m$ is the illumination compensation operator that includes diagonal weights proportional to the inverse of the source-side wavefield, $\mathbf{P}_d$ is a linear filter that eliminates the diving wave energy from the data residuals, and $\mathbf{D}$ represents the 2D Laplacian filter, which removes low-frequency noise from the gradient.

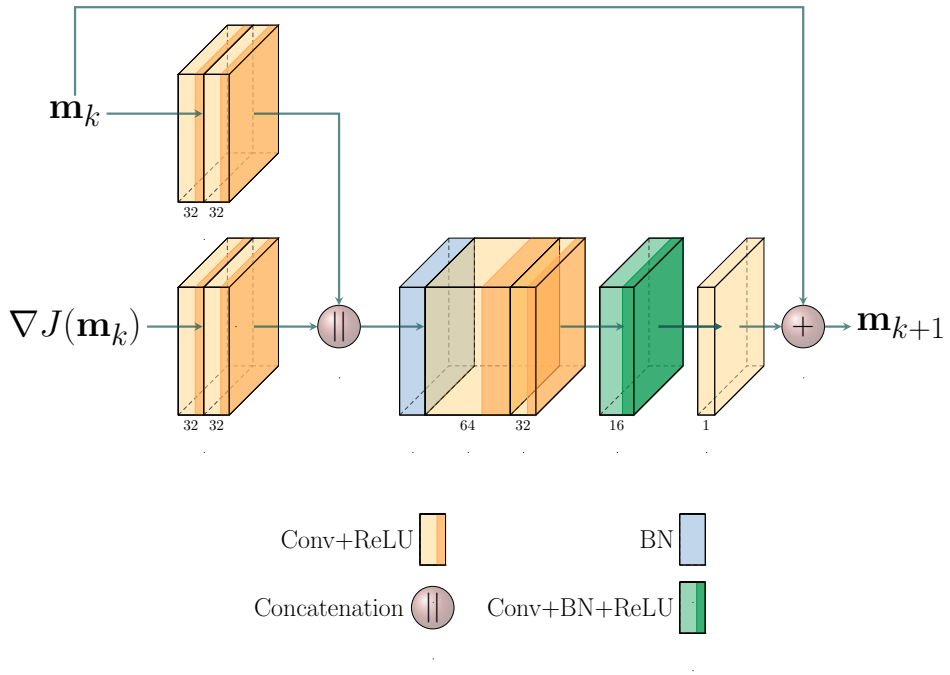

Figure 2.2: The proposed network architecture for each learned updating operator $\mathcal{P}_{\theta_k}$, based on an encoder-decoder scheme. The number of channels is shown below each convolutional layer. ReLU and BN refer to the rectified linear unit function and the Batch Normalization function, respectively.

2.2.3 Network design

We propose to assemble each updating operator $\mathcal{P}_{\theta_k}$ with the network architecture shown in Figure 2.2, following an encoder-decoder sequence. The encoding part uses two consecutive convolutional layers with a ReLU function to extract features from the two input branches containing $\mathbf{m}_k$ and $\nabla J(\mathbf{m}_k)$. These branches do not share intermediary layers until they are merged by a concatenation operator, which fuses the independent feature maps on the channel axis, followed by a Batch Normalization layer (BN). BN along the channel axis is needed to equalize contributions from the merged branches because the feature maps extracted from $\mathbf{m}_k$ have a different scale than those from $\nabla J(\mathbf{m}_k)$. This is crucial for successful training as it avoids the vanishing gradient problem in backward propagation (Ioffe and Szegedy, 2015). The encoder step ends with another convolutional layer that extracts information from the fused feature maps. Subsequently, the decoder part reassembles the feature maps by gradually decreasing the number of channels. The last convolutional layer does not use a ReLU function as the reflectivity updates can be positive or negative. We also add a skip connection between the input layer containing $\mathbf{m}_k$ and the output layer, forcing

the network to learn residual updates (i.e., it learns to compute an update perturbation to the initial reconstruction). Since there is a high structural correlation between inputs and output, learning a residual mapping seems intuitively easier than learning the direct mapping (He et al., 2016).

In more detail, we use convolutional kernels of size 3×3 for all the convolutional layers. For the first iteration, we initialize the parameters following He et al. (2015). For subsequent iterations, while there is no sharing of parameters across blocks, we use the trained weights of the previous iteration to initialize the weights of the next iteration to speed up the training process. All the layers use a stride of size one and are zero-padded such that all feature maps have the same size as the inputs. We obtain an optimal hyperparameter setting using a simple trial and error procedure. Different network components (such as filter size, number of filters, type of activation function, batch size, and learning rate) are changed, and the final network architecture is determined based on the validation dataset performance. We stress that these settings rely purely on empirical experimentation without any robust mathematical proof; thus, other hyperparameters combinations and similar architectures could produce similar results.

2.2.4 Dataset and training procedure

The training samples determine the features that the network learns. In a Bayesian context, the training dataset selection becomes crucial because it defines the prior distribution. Nonetheless, the main limitation of applying supervised deep learning methods in geophysical imaging problems is the lack of databases with realistic and generalized ground-truth labels. Consequently, we prepare a synthetic dataset generating 1200 pseudo-random reflectivity models, acknowledging that the restricted nature of the training instances will affect the quality of the reconstruction. In this work, the reflectivity model is expressed as velocity perturbations in squared slowness units. Therefore, the ground-truth labels are derived from velocity distributions that mimic fractured and folded sedimentary structures ranging from 1500 to 5500 m/s (Figure 2.3). Velocity, folding amplitude, and fault size are set to increase with depth gradually, and all instances have a variable number of layers. Each model has a fixed grid size of 400×200 points with a regular grid spacing of 10 meters. For migration, we calculate the background field of every velocity model using a 2D Gaussian filter with standard deviations from 2 to 6 grid points chosen from a discrete uniform distribution (equivalent to the range of 10 to 50 m in intervals of 10 m). In the following numerical examples, we separate 900 reflectivity models for training, 100 models for validation, and 200 models for testing. The validation set is used to optimize hyperparameters while training and the test models were used to evaluate inference performance.

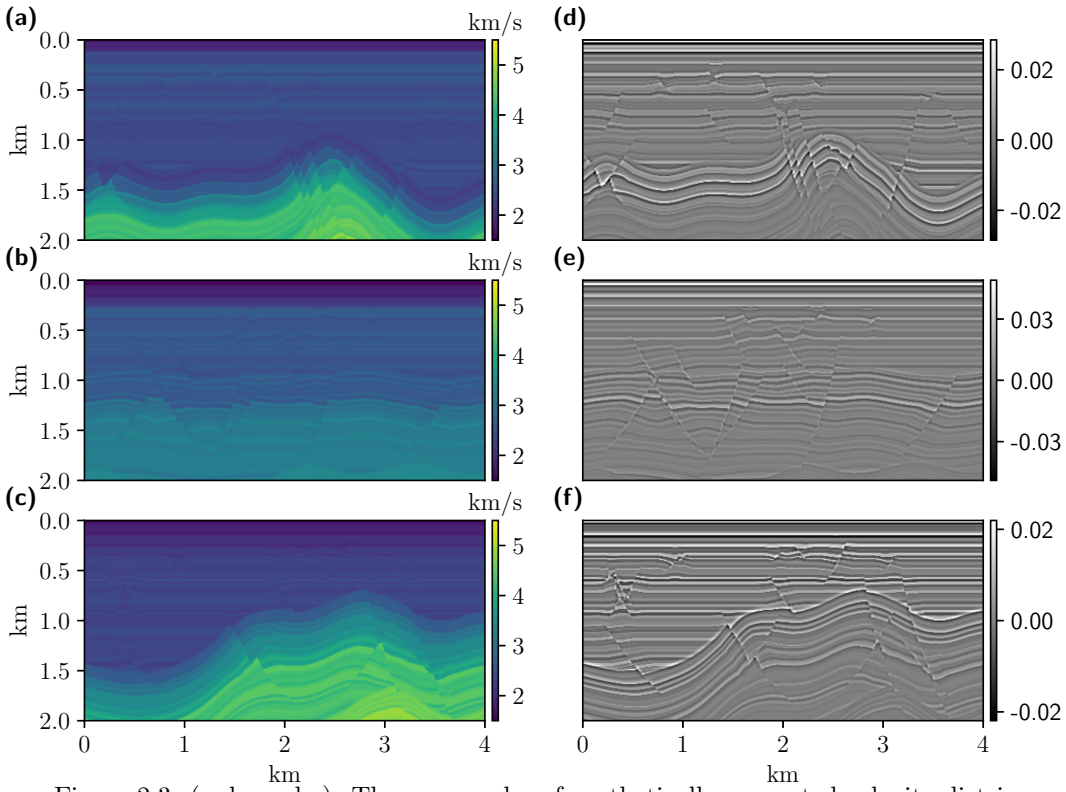

Figure 2.3: (a, b, and c): Three examples of synthetically generated velocity distributions. (d, e, and f): Ground-truth reflectivity samples derived from the velocity models in the first column.

Since the pre-stack LSRTM gradient requires observed seismic data in the shot-gather domain, we generate the shot gathers by simulating an acquisition on every model before starting the training process and store them on disk. Even for such a small dataset, including the data-misfit gradient is the most demanding part of our algorithm during training, and therefore we consider acquisitions with a reduced number of sources. Specifically, the numerical simulations of wave propagation for each sample model involve a fixed-spread geometry with only 15 shots spanning from 100 m to 3900 m in the horizontal direction and 30 m in depth, and 400 collocated receivers evenly placed at 30 m in depth and 10 m spacing. Due to the coarse source distribution, aliasing artifacts are expected in the migrated images. The seismic data are computed using a Time Domain Finite Differences (TDFD) wavefield propagator with a surface absorbing boundary condition to simulate data with removed surface-related multiple. Each shot is recorded for 2.2 s with a time sampling of 1 ms, resulting in 2200 time steps of modeling time. A 20 Hz Ricker wavelet is used as the seismic source signature.

To further decrease the computational burden, we perform the gradient calculation step on a multi-GPU cluster with four NVIDIA GeForce RTX 2080 Ti computing processors to generate gradient images simultaneously. All the models and their corresponding data (i.e., the observed seismic data, the background velocity models, and the seismic modeling parameters) are distributed among the devices using four threads, such that each GPU calculates the data-misfit gradient associated with a different model in parallel. Due to the relatively small size of the models and the number of time steps required to propagate the wavefield to all parts of the domain, we calculate each gradient using the traditional adjoint state method, storing the forward wavefield within the GPU. With this configuration, computing a demigration/migration sequence for 15 sources on each model takes about 25 s. Compared to the cost of multiple forward and backpropagation calls of the gradient step, the optimization of equation 2.13, is negligible.

Finally, we only consider the updating operator of the first Deep-LSRTM iteration for hyperparameter selection, supported by the observation that the first iteration is the one that performs a more severe transformation.

After hyperparameter tuning, we set the maximum number of iterations to $K = 5$, as further increasing the number of blocks does not lead to better reconstructions. We train each updating operator with the Adam optimizer (Kingma and Ba, 2014a) to minimize equation 2.13, using a learning rate of 0.001, a batch size of 2, and 111 epochs, where an *epoch* is defined as one pass through the training data. Training five iterations of Deep-LSRTM takes about 8 hours in the multi-GPU cluster previously mentioned. Furthermore, we do not use data augmentation or other standard training techniques such as dropout or max pool layers. With such configuration, this architecture leads to a total of 341,441

trainable parameters per updating operator.

Algorithm 1 and 2 describe the complete Deep-LSRTM training and inference processes, respectively. Equally to conventional LSRTM algorithms, we assume a zero reflectivity vector ($\mathbf{m}_0 = \mathbf{0}$) as the initial guess.

Algorithm 1 Deep-LSRTM (Training)

Inputs: Dataset $\Im$, initial reflectivity estimates $\mathbf{m}_0^i$, diagonal preconditioners $\mathbf{P}_m^i$, observed data $\mathbf{d}^i$, number of iterations K

Set $k = 0$
while $k < K$ **do**
  $\mathbf{r}^i = (\mathbf{L}\mathbf{m}_k^i - \mathbf{d}^i)$ ▷ Compute residuals for all models
  $\nabla \tilde{J}(\mathbf{m}_k^i) = \mathbf{D}\mathbf{P}_m^i\mathbf{L}^T\mathbf{P}_d\mathbf{r}^i$ ▷ Calculate preconditioned LSRTM gradients for all models
  if $k > 0$ **then**
    Initialize θ_k with θ_{k-1}
  end if
  $\min_{\theta_k} \frac{1}{\Im}\sum_{i=1}^{\Im} ||\mathcal{P}_{\theta_k}(\mathbf{m}_k^i, \nabla\tilde{J}(\mathbf{m}_k^i)) - \mathbf{m}_{\text{true}}^i||_2^2$ ▷ Train $\mathcal{P}_{\theta_k}$ until stopping criteria
  $\mathbf{m}_{k+1}^i = \mathcal{P}_{\theta_k}(\mathbf{m}_k^i, \nabla\tilde{J}(\mathbf{m}_k^i))$ ▷ Update reflectivity for all models
  $k = k + 1$
end while
Output Trained parameters $\Theta = [\theta_0, ..., \theta_{K-1}]$

Algorithm 2 Deep-LSRTM (Inference)

Inputs: Trained parameters Θ, initial reflectivity estimate $\mathbf{m}_0$, diagonal preconditioner $\mathbf{P}_m$, observed data $\mathbf{d}$, number of iterations K

Set $k = 0$
while $k < K$ **do**
  $\mathbf{r} = (\mathbf{L}\mathbf{m}_k - \mathbf{d})$ ▷ Compute residuals
  $\nabla \tilde{J}(\mathbf{m}_k) = \mathbf{D}\mathbf{P}_m\mathbf{L}^T\mathbf{P}_d\mathbf{r}$ ▷ Calculate preconditioned LSRTM gradient
  $\mathbf{m}_{k+1} = \mathcal{P}_{\theta_k}(\mathbf{m}_k, \nabla\tilde{J}(\mathbf{m}_k))$ ▷ Update reflectivity
  $k = k + 1$
end while
Output $\mathbf{m}_K$

Figure 2.4 shows the loss curves of the training and validation datasets versus the number of epochs after five iterations of Deep-LSRTM. It is clear that the error of both datasets reduces drastically in the first epochs of the first iteration and then gradually stagnates until the next iteration takes place. A similar pattern between training and validation losses with no signs of over-fitting indicates good generalization properties of the model. Furthermore,

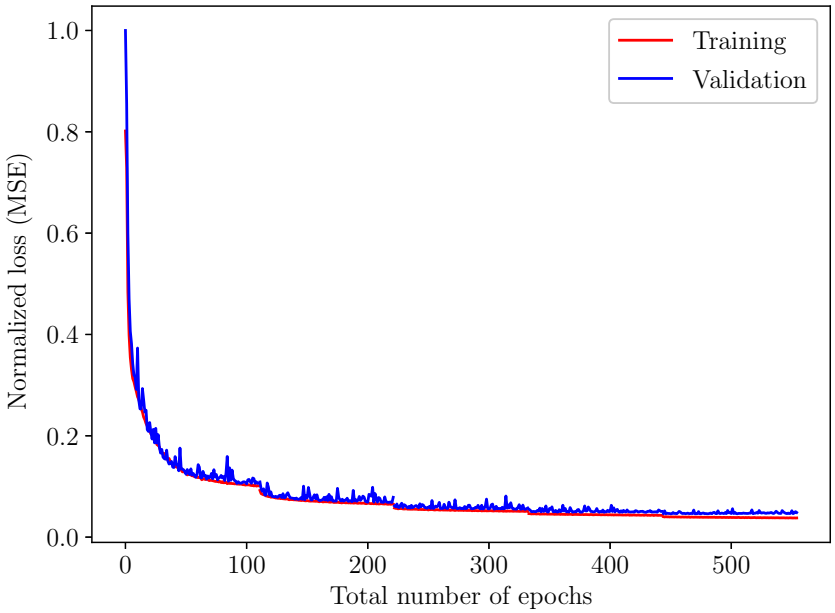

Figure 2.4: The normalized training and validation loss functions versus the number of epochs for the Deep-LSRTM architecture.

the continued decrease in both losses with each iteration verifies the benefits of the iterative scheme.

2.2.5 Transfer learning for real applications

A crucial assumption in supervised deep learning algorithms is that the testing data must be in the same feature space and have the same distribution as the training data (Pan and Yang, 2010). However, when dealing with applications to real data, such condition is hard to ensure if the training dataset is only composed of synthetically generated samples. Hence, a direct application of the trained network usually will not produce satisfactory results because seismic data can have dramatically distinct features than those represented by the synthetic samples. To mitigate this out-of-distribution problem and enable the successful implementation of Deep-LSRTM to real data, we resort to the transfer learning strategy based on model fine-tuning (Park and Sacchi, 2020).

We devise a three-step workflow for a limited set of available measurement data. First, we pre-train the updating operators using Algorithm 1 and the synthetically generated training samples described in the previous section. The objective of the pre-training stage is to build a good prior for the targets of interest. Then, we set apart a reduced number of shots from

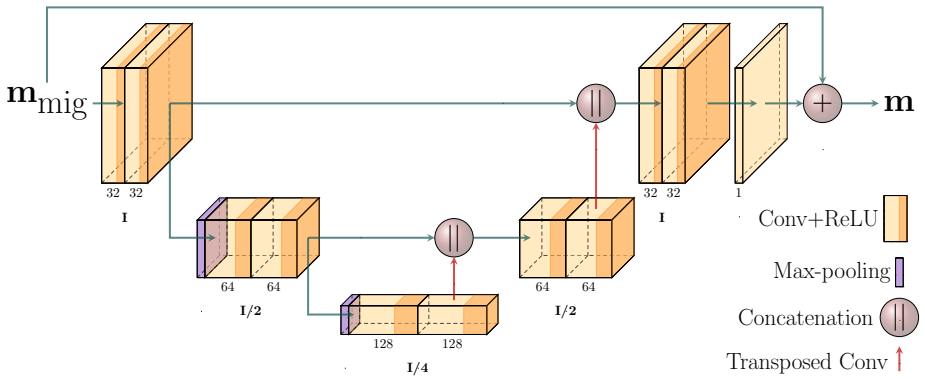

Figure 2.5: Architecture of the residual U-net adopted in this study. The number at the bottom of each convolutional layer indicates the number of channels. **I** refers to the original image size.

the real observed data to obtain ground-truth samples by solving the inverse problem via preconditioned CGLS. We perform fine-tuning by re-training each updating operator with a lower learning rate and fewer epochs, using the pre-trained weights as the initial state. As a result, we have a network model that can be used to predict the target region's reflectivity. The updated parameters are finally tested on a different group of shots to obtain the final image.

2.2.6 Single-step image post-processing via U-net reconstruction

We compare the iterative Deep-LSRTM with a two-step CNN-based post-processing technique, where first the seismic data are migrated to form an initial reconstruction, and then we train a modified version of the U-net (Ronneberger et al., 2015a) architecture to remove artifacts from the RTM images. It represents an efficient non-iterative alternative to Deep-LSRTM because it does not require multiple computations of the data-misfit gradient, making the input-output demands considerably lower, so it is cheaper to train.

This second formulation substitutes equation 2.5 with

$$\mathbf{m} = \Lambda_{\Phi}(\mathbf{m}_{\text{mig}}) \tag{2.15}$$

where $\Lambda_{\Phi} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is an image-to-image transformation represented by the modified U-net network shown in Figure 2.5, and $\mathbf{m}$ is the filtered reflectivity image. We recognize that equation 2.15 is close in spirit to image-domain LSM methods that attempt to approximate the effect of the inverse Hessian in a single iteration (Guitton, 2004).

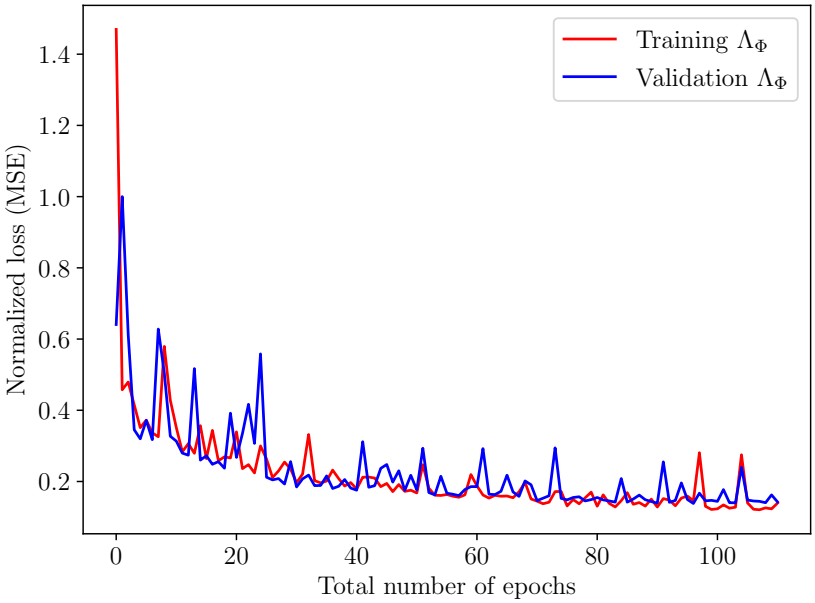

Figure 2.6: The normalized training and validation loss functions versus the number of epochs for the U-net architecture.

The U-net can be described as a multi-scale convolutional auto-encoder. The encoder part of the network decomposes and resizes the image through a series of down-sampling (max-pooling) layers to capture more extensive features, akin to wavelet decomposition. The spatial features are then up-sampled via transposed convolutions to reconstruct the final image. Moreover, a series of multilevel skip connections between the encoder and decoder parts helps preserve the different ranges of structures and avoids vanishing gradients (Ronneberger et al., 2015a). Compared to the original U-net architecture, the modified U-net implemented in this study includes an additional skip connection between input and output layers for residual learning and a reduction from five to three scales to work with smaller images and fewer trainable parameters (517,409 in total). Similar to Deep-LSRTM, this architecture is a residual network because of the extrinsic skip connection. For consistency, our U-net is also trained with Adam optimizer maintaining the same hyperparameter configuration as in the previous section. Figure 2.6 shows the loss curves for this method, where we also see a decrease of the loss functions with the number of epochs and training and validation losses with close behavior.

We train the parameters of the U-net architecture by minimizing the MSE loss

$$\min_{\Phi} \frac{1}{\Im} \sum_{i=1}^{\Im} ||\mathbf{m}^i_{\text{mig}} - \mathbf{m}^i_{\text{true}}||_2^2, \tag{2.16}$$

where the training data pairs correspond to RTM images $\mathbf{m}_{\text{mig}}$ and ground-truth reflectivity models $\mathbf{m}_{\text{true}}$.

2.3 Numerical examples

This section reports the results of Deep-LSRTM reconstruction in four different scenarios. The first three scenarios explore a direct implementation of Deep-LSRTM on synthetic models. In the first three scenarios, transfer learning was not used. The fourth scenario entails a field dataset where we use both a direct implementation of our method and the modification that includes the transfer learning component. In our numerical examples, we also consider LSRTM solutions computed by the preconditioned CGLS algorithm and the results obtained by the U-net reconstruction for baseline comparisons. To make fair comparisons between iterative methods, we run the same number of iterations (five) for Deep-LSRTM and CGLS algorithms (unless otherwise stated) and use the same data and model space preconditioners. The terms preconditioned LSRTM and conventional LSRTM are used interchangeably henceforth, referring to the inversion via the preconditioned CGLS method. In order to avoid the inverse crime, all the observed data of the synthetic examples are generated with the solution of the discretized acoustic-wave equation instead of Born acoustic modelling (Tarantola, 1984b). We use Tensorflow (Abadi et al., 2015) for the numerical implementation of the CNN architectures. To quantify the imaging performance of each algorithm, we compute the peak signal-to-noise ratio (PSNR) (Huynh-Thu and Ghanbari, 2008) and the Structural Similarity Index (SSIM) (Wang et al., 2004) (see Appendix A).

2.3.1 Example 1: test dataset

The first scenario examines the reconstruction of the test dataset. The test samples were generated in the same way as the training and validation samples but were not included in the training phase. Hence, this experiment showcases the performance of Deep-LSRTM on models that have the same prior distribution as the training samples. Table 2.1 presents a quantitative comparison of the different reconstruction methods for noiseless data. Even though the U-net approach can produce good quality images, the quantitative measures indicate that Deep-LSRTM delivers more competitive results for this dataset. On average, both learned methods perform better than the preconditioned CGLS algorithm. Figure 2.7 shows the results obtained for one sample of the test dataset (Figure 2.7a). For this model, the migration velocity field was obtained by convolving the true velocity with a 50 m standard deviation 2D Gaussian filter (shown in Figure 2.7b). Figure 2.7c displays the RTM section after illumination compensation and filtering by a Laplacian operator. The

Example 1 - test dataset		
Method	PSNR (db)	SSIM
CGLS	24.81±0.82	0.65±0.089
U-net	31.63±0.74	0.77±0.041
Deep-LSRTM	34.91±0.38	0.87±0.029

Table 2.1: Quantitative measures for the reconstruction of the test dataset. PSNR and SSIM scores are computed in comparison to the ground-truth labels (average results ± standard deviation over 200 different samples).

adjoint image contains a strong acquisition footprint and unresolved areas at deeper parts of the model because of limited aperture. The preconditioned CGLS result (Figure 2.7d) shows slightly increased illumination and frequency content, but its main contribution after five iterations is the partial attenuation of the near-surface artifacts.

In contrast, Deep-LSRTM (Figure 2.7e) considerably improves the amplitude balance, the structural continuity, and the resolution of the image in comparison to the CGLS result for the same number of iterations. Moreover, it satisfactorily learns to remove the acquisition footprint at the top layers. The residual U-net (Figure 2.7f) also delivers a good-quality reconstruction but tends to smear out the reflectors at shadow regions (an example is the anticline structure marked by the blue rectangle in Figure 2.7), and the sharpening of steep faults is limited (indicated by the red rectangle in Figure 2.7). We particularly notice that the deconvolution effect of the post-processing approach degrades in regions where the RTM image has poor illumination, which indicates a strong dependence between the U-net reconstruction and its inputs. On the other hand, the superior deconvolution effect of the iterative learned approach in this example seems to highlight the benefits of residual imaging. By comparing the gradients of the first and last iterations in Figure 2.8, we can observe that the magnitude of the gradient decreases with iterations due to a better match between observed and calculated data, which is a reassuring, albeit expected, result.

Figure 2.9 shows a comparative analysis between the conventional LSRTM and the proposed method by means of the relative model error defined as

$$e(\mathbf{m}_k) = \frac{||\mathbf{m}_k - \mathbf{m}_{\text{true}}||_2}{||\mathbf{m}_{\text{true}}||_2}, \tag{2.17}$$

which is calculated for the inversion results of the sample model in Figure 2.7. Since both algorithms use the zero reflectivity vector as initial guess, they share the same resulting error for the initialization at $k = 0$. The conventional LSRTM converges to a relative model error of 0.36 after 20 iterations. In contrast, Deep-LSRTM achieves a considerably faster convergence to a smaller error value of 0.15 in just 5 iterations.

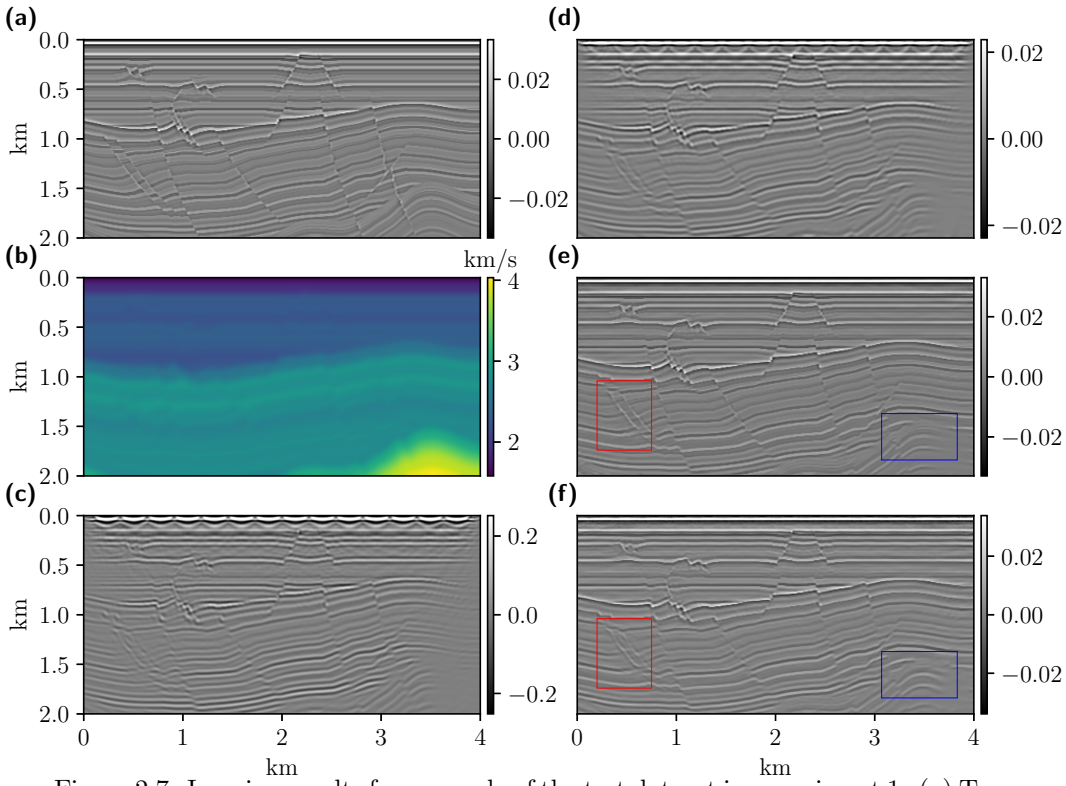

Figure 2.7: Imaging results for a sample of the test dataset in experiment 1. (a) True reflectivity, (b) migration velocity model, (c) RTM image, (d) CGLS and (e) Deep-LSRTM methods after five iterations, respectively, and (f) the U-net reconstruction. The red and blue rectangles exemplifies reconstruction at poorly illuminated step faults and anticline structures, respectively.

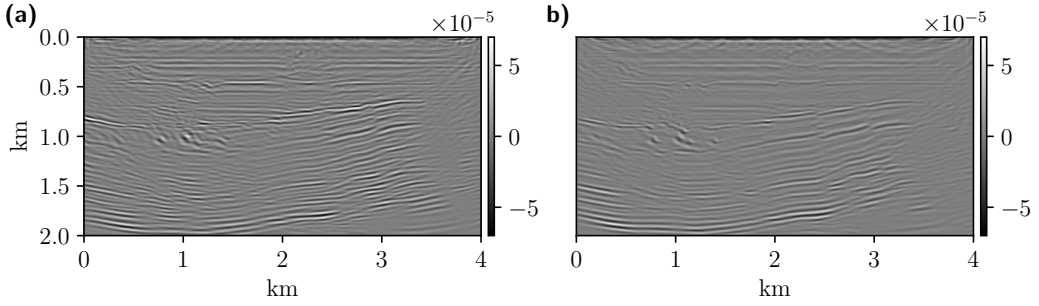

Figure 2.8: Deep-LSRTM gradients of the first (a) and last (b) iterations for the sample of the test dataset in experiment 1, shown in Figure 2.7a.

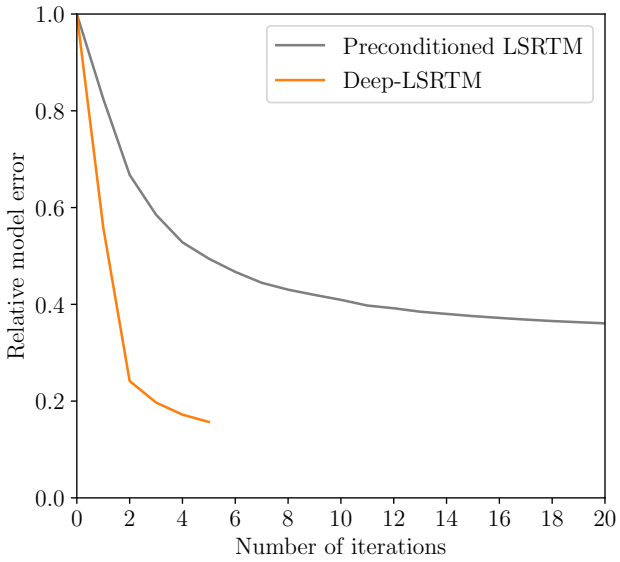

Figure 2.9: Convergence curve of the relative model error for comparative analysis between the preconditioned LSRTM and Deep-LSRTM.

2.3.2 Example 2: cropped Marmousi

The following scenario comprises a more complex setting extracted from the central part of the Marmousi model (Martin et al., 2006a). The purpose of this test is to explore the generalization capacity of Deep-LSRTM to environments whose distributions differ from the ones used at training time. Similar to the first experiment, Figure 2.10a shows the true reflectivity model, and Figure 2.10b displays the background velocity, computed with a 50 m standard deviation 2D Gaussian filter. Figure 2.10c shows the RTM image, which possesses unfocused reflectors at the deeper part of the model and strong source-sampling aliasing artifacts that are uncorrelated with the geology, manifested as spurious tails. The least-squares method improves the content of high-frequency features, so the preconditioned CGLS image shows higher resolution with deblurring capability, especially at the shallow layers. While the learned approaches offer increased resolution in deeper parts of the model, it can be noticed that, once again, the focusing properties of the U-net reconstruction (Figure 2.10) are limited by the RTM result. In this case, the U-net also introduces jittered high-frequency noise at the top layers. We attribute this behavior to the high structural complexity of the Marmousi model in the near-surface region compared to the less complex training samples. In this regard, the iterative learned approach achieves better generalization properties showing enhanced continuity of the reflectors and not introducing high-frequency artifacts at the top layers. Our qualitative assessment is supported by the PSNR and SSIM scores,

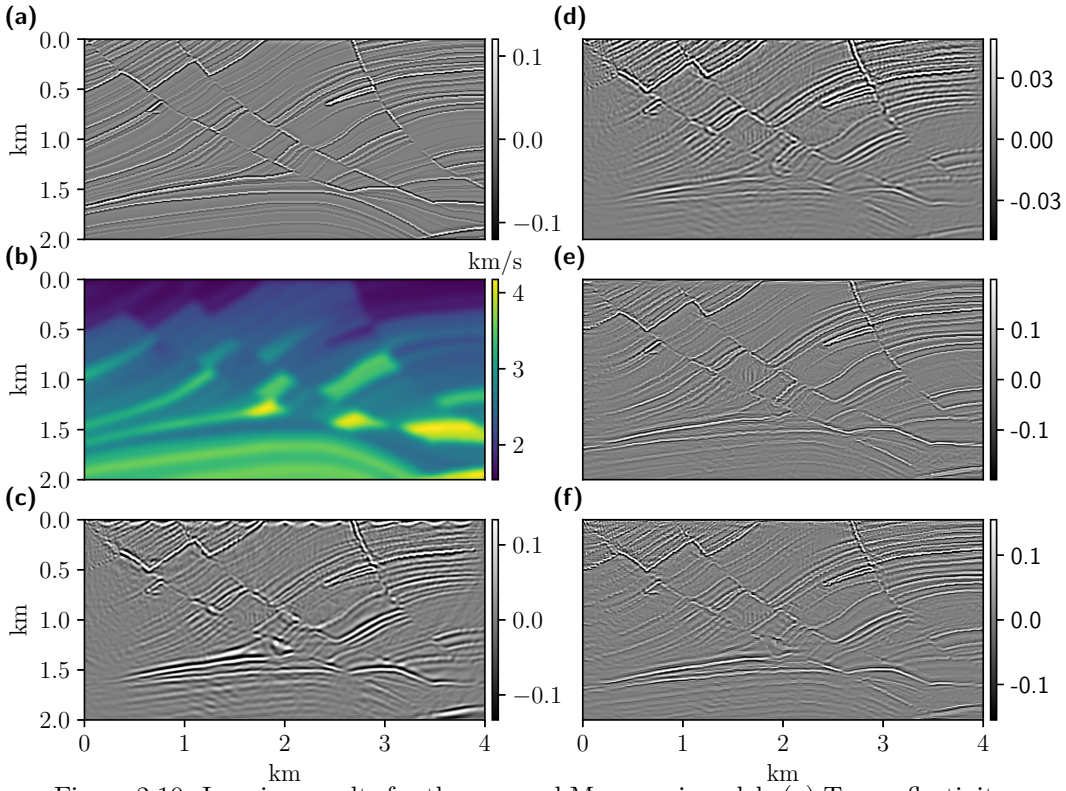

Figure 2.10: Imaging results for the cropped Marmousi model. (a) True reflectivity, (b) migration velocity model, (c) RTM image, (d) CGLS and (e) Deep-LSRTM methods after five iterations, respectively, and (f) the U-net reconstruction.

presented in Table 2.2, where Deep-LSRTM attains superior performance.

Learned reconstructions as warm-starts for CGLS

Exploiting the adaptability of the preconditioned CGLS algorithm, we illustrate the possibility of efficiently improving the results attained by the learned methods based on warm-start reconstruction. Warm-start enables the preconditioned CGLS method to initialize the recursive updates with an initial reflectivity guess $\mathbf{m}_0 \neq \mathbf{0}$ provided by the user. Using the learned reconstruction results as feasible solutions to warm-start conventional LSRTM can help obtain an optimal solution in a reduced number of iterations. Figure 2.11a and Figure 2.11b display the results of five CGLS iterations (considering again a coarse source acquisition of only 15 shots) employing the Deep-LSRTM reconstruction and the U-net reconstruction as warm-start models, respectively. In general, the results show enhancement in the recovery of model features not entirely reconstructed by the networks, especially at shallower

Example 2 - cropped Marmousi		
Method	PSNR (db)	SSIM
CGLS	27.46	0.47
U-net	28.37	0.53
Deep-LSRTM	29.87	0.65
Warm-started CGLS		
$\text{CGLS}_{\text{U-net}}$	29.96	0.58
$\text{CGLS}_{\text{Deep-LSRTM}}$	30.37	0.69

Table 2.2: Quantitative measures for the reconstruction of a cropped version of the Marmousi model. PSNR and SSIM scores are computed in comparison to the ground-truth label.

regions, which yields higher PSNR and SSIM scores in both cases (see Table 2.2). Figure 2.11b also exhibits a reduction of the spurious high-frequency artifacts present in the initial U-net result. The convergence plot of the least-squares data misfit shown in Figure 2.12 verifies that warm-starting the CGLS algorithm with deep learning results produces faster convergence than conventional LSRTM, with the inversion started with the Deep-LSRTM result decreasing slightly faster when compared with its U-net counterpart.

All of the tested methods cannot completely remove the remaining migration artifacts in the image, specifically in the central part of the model. We anticipate that increasing the number of sources can mitigate such artifacts at the expense of longer turnaround times.

2.3.3 Sensitivity to background model errors

Conventional LSRTM is highly sensitive to wrong background models by limiting its capacity to flatten and focus events at regions with significant velocity errors. In light of the lack of robustness of DNN-based inversion methods to slight variations in the reconstruction process (Antun et al., 2020), we test the proposed learning approaches against background models calculated with different degrees of spatial smoothing. The test is performed on the same setup as the previous section using the Marmousi model. To further explore the influence of the velocity in the inference process, we also consider a variant of the Deep-LSRTM architecture that makes direct use of the background velocity field. Although the migration velocity is already embedded in the forward and adjoint operators, this modification is designed to help recover missing information in the model space by explicitly introducing the background velocity as a third complementary branch. In other words, before the evaluation step, we train the modified Deep-LSRTM architecture to take inputs of the form $(\mathbf{m}_k^i, \nabla\tilde{J}(\mathbf{m}_k^i), \mathbf{v}_0^i)$, where $\mathbf{v}_0^i$ is the fixed migration velocity model of the i-th training sample. Explicitly using the migration velocity model as an extra input, yields the modified

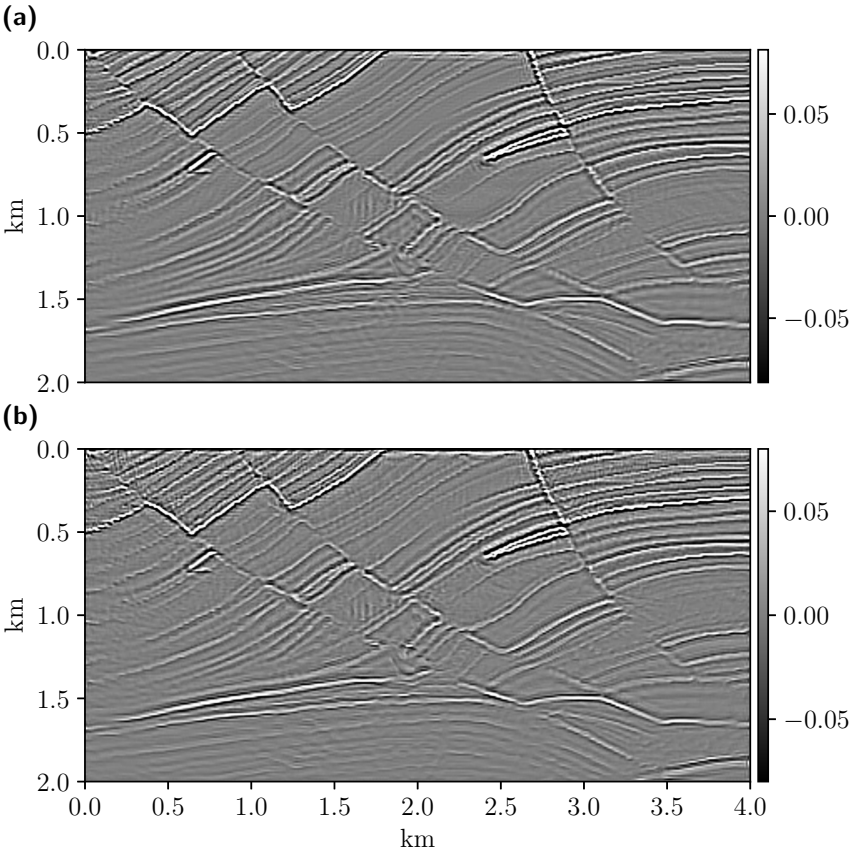

Figure 2.11: Results after five iterations of the warm-started CGLS algorithm using (a) the fifth iteration of the Deep-LSRTM method and (b) the U-net reconstruction, as initial guesses, respectively.

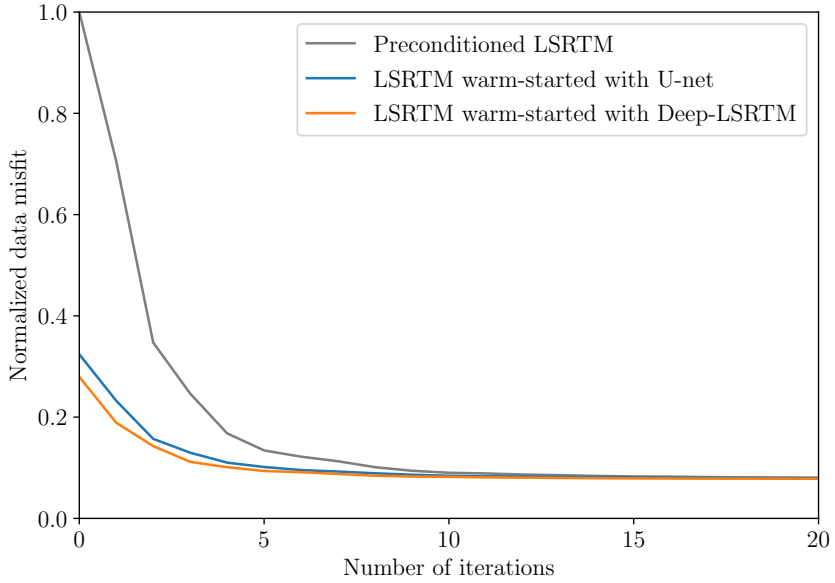

Figure 2.12: Convergence curves for the preconditioned LSRTM, LSRTM warm-started with the prediction of the U-net post processing approach, and LSRTM warm-started with the prediction of the Deep-LSRTM result. The curves are normalized with respect to the initial value of the preconditioned LSRTM data misfit.

optimization problem

$$\min_{\theta_k} \frac{1}{\Im} \sum_{i=1}^{\Im} ||\mathcal{P}_{\theta_k}(\mathbf{m}_k^i, \nabla \tilde{J}(\mathbf{m}_k^i), \mathbf{v}_0^i) - \mathbf{m}_{\text{true}}^i||_2^2. \tag{2.18}$$

Figure 2.13 shows the PSNR metric for all the methods considering a background velocity model smoothed with a 2D Gaussian filter of standard deviation varying from 50 to 110 m in intervals of 10 m. We observe a general degradation of the reconstruction quality as smoothing increases for all the techniques under consideration. It can be noticed that providing a correct velocity model as direct input in the modified Deep-LSRTM architecture helps recover reflectors that are undetectable by the adjoint operator. This result is displayed in Figure 2.14a. However, for background velocities with high levels of smoothing not seen during training, both the post-processing U-net and the original Deep-LSRTM architecture are more robust than the three-branch Deep-LSRTM version, which deteriorates much quicker. Figure 2.14b shows the artifact-prone prediction of the modified Deep-LSRTM with a smoother velocity model.

Lastly, we test the reconstruction of our method using a migration velocity model with 5% faster velocity everywhere. Figures 2.15a, 2.15b, 2.15c, and 2.15d show the results of

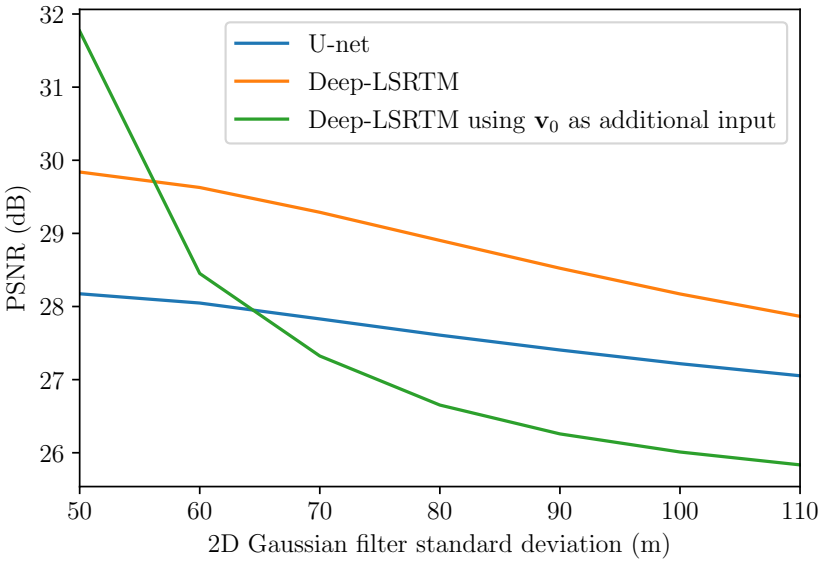

Figure 2.13: Comparison of reconstruction quality for the learned approaches on the central part of the Marmousi model using migration velocity models with different levels of smoothing.

the RTM, 20 iterations of conventional LSRTM, the U-net method, and five iterations of Deep-LSRTM, respectively. While all the methods fail to correctly position the reflection events due to significant inaccuracies on the velocity model, it is apparent that Deep-LSRTM provides a result with better spatial resolution than the traditional LSRTM, which produces a blurred and incoherent image, and a superior capability to filtering spurious artifacts compared to the learned post-processing technique.

2.3.4 Sensitivity to random noise

We examine the robustness of the learned approaches concerning additional noise in the seismic data. Accordingly, we synthetically contaminate the observed data with different levels of normally distributed random noise. Figure 2.16 displays the evolution of PSNR scores of the reconstructions for different levels of signal-to-noise ratio (SNR). Interestingly, the residual U-net proves to be more robust for the addition of noise variations, whereas Deep-LSRTM degrades faster under decreasing signal-to-noise (S/N) ratios. We attribute this behavior to the higher number of parameters and multi-scale architecture of the U-net, which has been proven efficient in removing and detecting artifacts in images.

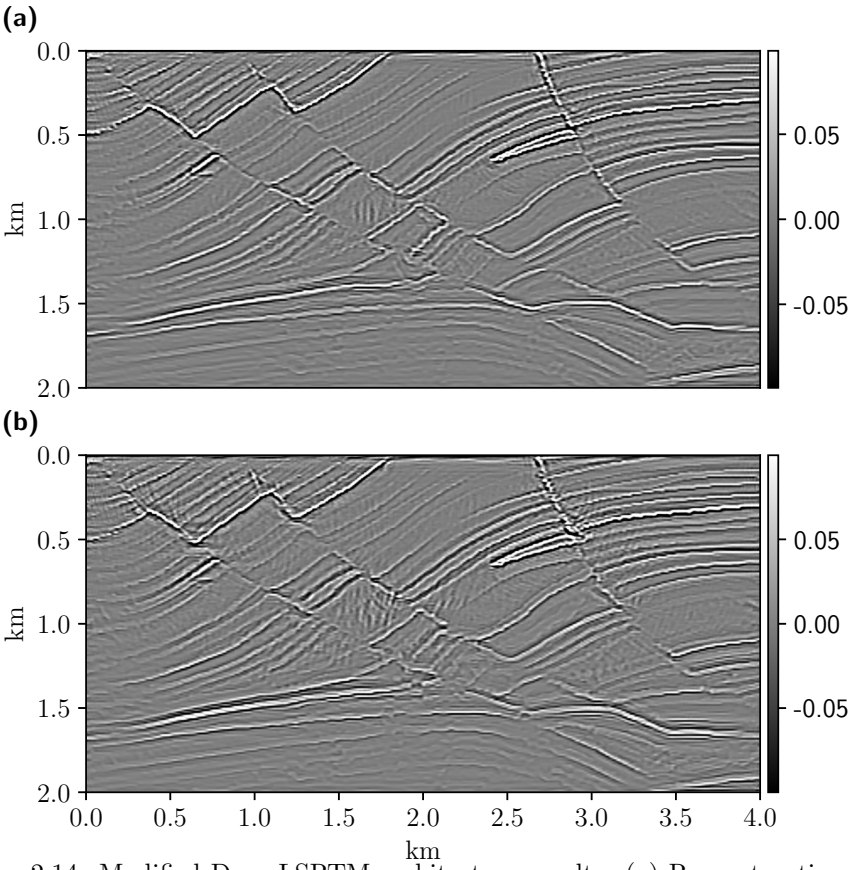

Figure 2.14: Modified Deep-LSRTM architecture results: (a) Reconstruction with a 50 m deviation 2D Gaussian filter. (b) Reconstruction with a 110 m deviation 2D Gaussian filter.

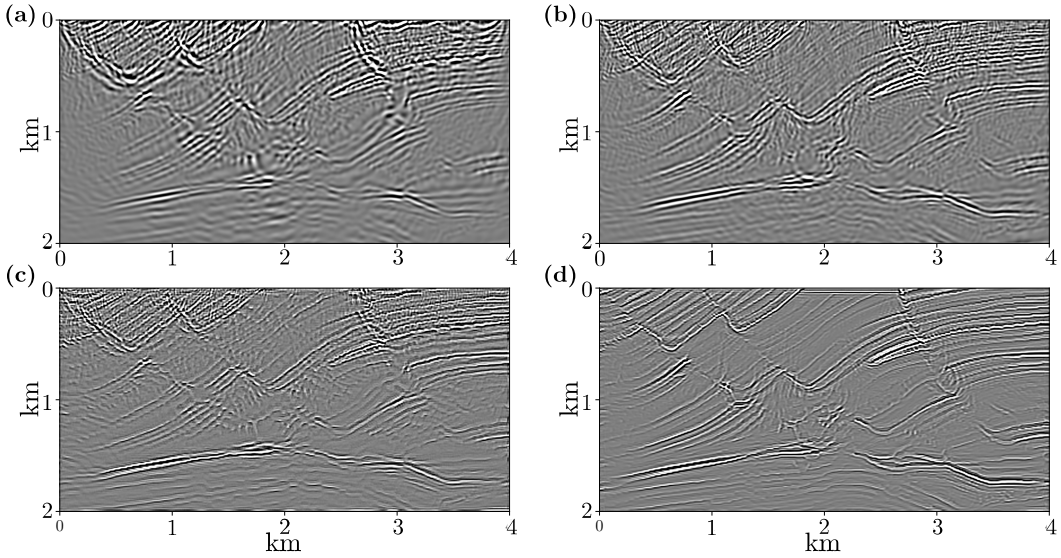

Figure 2.15: The migration images for an inaccurate migration velocity model with 5% faster velocity everywhere: (a) the RTM, (b) 20 iterations of preconditioned LSRTM, (c) the U-net reconstruction, and (d) the Deep-LSRTM method.

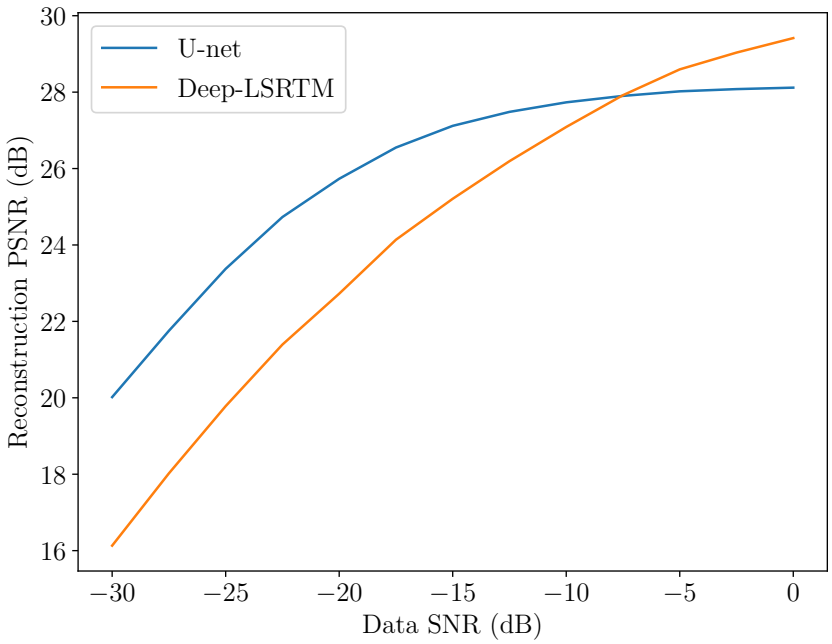

Figure 2.16: Robustness study of the learned methods against noisy data reconstruction. The horizontal axis indicates the level of signal-to-noise ratio (SNR).

2.3.5 Sensitivity to dominant frequency and grid spacing

To further investigate the generalization ability of the proposed workflow in terms of other relevant parameters, we experimentally quantify the prediction performance regarding variations in the dominant frequency of the modelling Ricker wavelet and the grid spacing. We perform these tests on the Marmousi example. The results are presented in Table 2.3. We see a general degradation of the prediction results for values that are different from the baseline (i.e. $f_{\text{dom}} = 20Hz$, h=10 m) since all the experiments from the training dataset used a fixed value for the dominant frequency and fixed grid spacing. We can notice less degradation in the prediction performance for values closer to the baseline configuration. The phenomenon of training generalization still requires further research since it may be influenced by a variety of other factors not considered in this study (e.g. the subsurface complexity of the training models and the number of training samples).

	$f_{\text{dom}} = 10$ Hz	$f_{\text{dom}} = 15$ Hz	$f_{\text{dom}} = 20$ Hz
$h = 10$ m	26.75	28.47	**29.87**
$h = 15$ m	28.75	28.91	26.61
$h = 20$ m	29.52	26.62	26.59

Table 2.3: PSNR (dB) scores of Deep-LSRTM for different configurations of dominant frequency f_{dom} and grid spacing h tested in the Marmousi example. The filled cells designate the values of f_{dom} and h used for the training stage.

2.3.6 Example 3: Sigsbee2a model

For our third example, we use the resampled Sigsbee2a model (Paffenholz et al., 2002), a challenging salt model with 12 km in horizontal distance and 4 km in depth with a grid interval of 10 m (Figure 2.17a). We simulate an off-end acquisition geometry where the positions of the receivers move with the positions of the sources from left to right at zero depth. The data set consists of 100 observed shot records with 5 s recording time and 183 evenly spaced hydrophones. The maximum and minimum offsets are 4874 m and 100 m, respectively. The wavelet frequency and time sampling are kept the same as the training data.

The high-velocity contrast and irregular shape of the intruding salt body challenge the generalization of our trained networks, which have not seen complex examples with salt during training. Figure 2.17 shows the inversion results for the Sigsbee2a example. For comparison, we include the results after 20 iterations of the preconditioned LSRTM (Figure 2.17b), and the U-net post-processing approach (Figure 2.17c). The result after five iterations of our

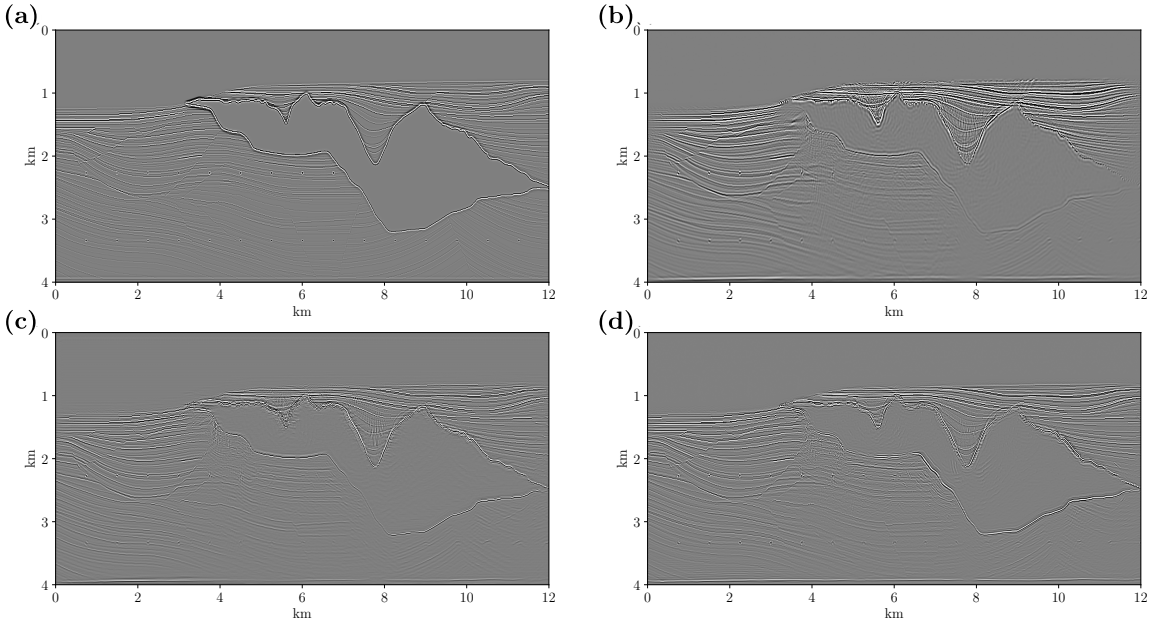

Figure 2.17: Imaging results for the Sigsbee2a model. (a) True reflectivity, (b) LSRTM image, (c) U-net image, (d) Deep-LSRTM image.

iterative network is displayed in Figure 2.17d. The predicted Deep-LSRTM image produced good results, particularly on the faulted sedimentary region at the leftmost part of the model. We believe this is because the samples used for training are closer to representing the full space solution on this specific region of the model by correctly describing all features of interest. We also observe improved reconstruction performance at the shallow layers and at the top of the salt region, where we have the best illumination in the model compared with the other two methods.

To show how Deep-LSRTM copes with severe illumination issues in sub-salt areas, Figure 2.18 presents a close-up view of a shadow zone region (indicated by the dashed red box) near the top left flank of the salt body. Structures in this shadow zone produce events with very little or no illumination, and they cannot be modeled using Born modeling and restored using LSRTM (Figure 2.18b). In contrast, the regularization performance of our method is remarkable, showing better continuity than the LSRTM inversion after 20 iterations, from which we can identify the reflectors under the salt overburden. In addition, there is better focusing on the two point diffractors located beneath the salt (indicated by yellow arrows).

To further address illumination loss challenges, we present a close-up view of a deeper sub-salt region in Figure 2.19. While none of the methods achieves a perfect amplitude balancing

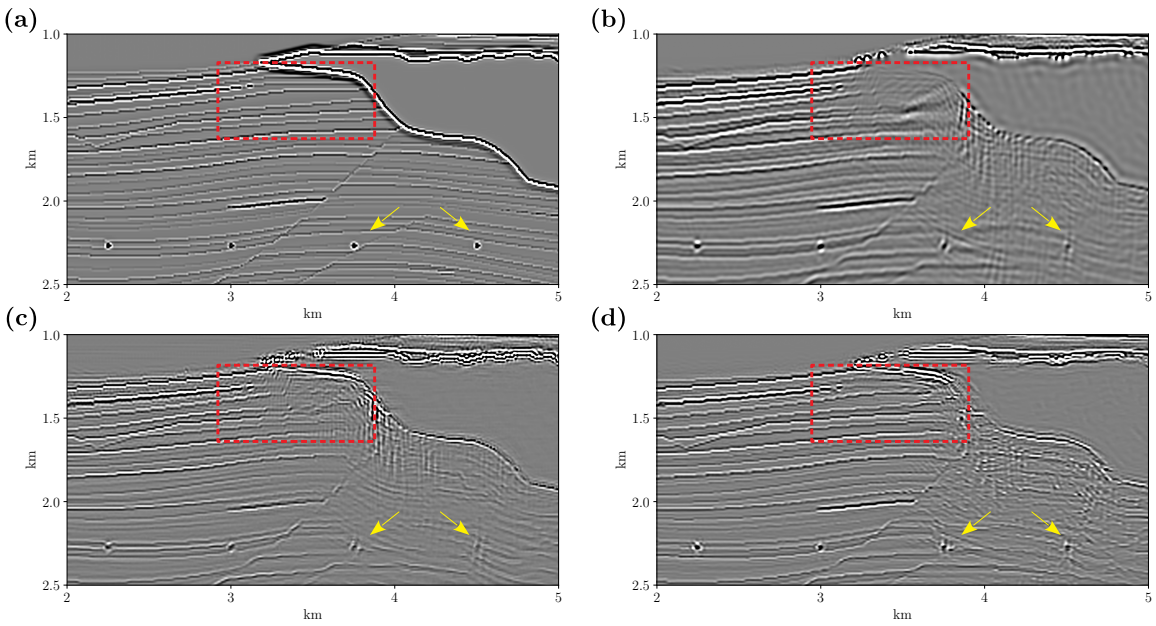

Figure 2.18: A close-up comparison of the reconstruction methods for the Sigsbee2a example. (a) True reflectivity. (b) conventional LSRTM image after 20 iterations. (c) U-net image. (d) Deep-LSRTM image. The red dashed box indicates a shadow zone region near the top left flank of the salt body. The yellow arrows indicate two point diffractors under the salt body.

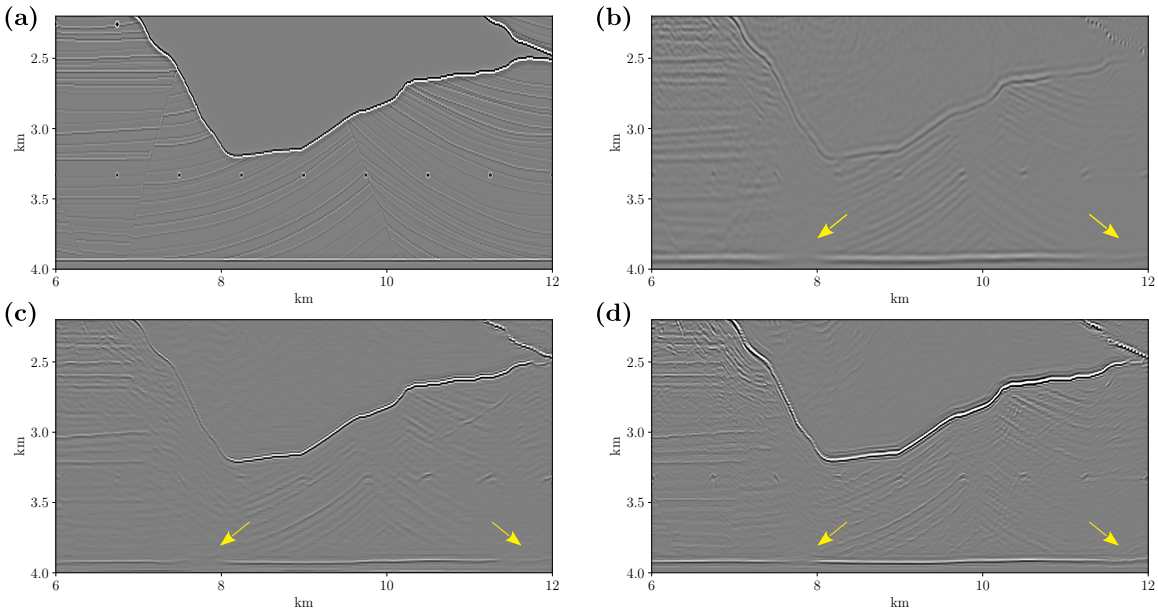

Figure 2.19: Close-up view of the reconstructions for the bottom right region of the Sigsbee2a model. (a) True reflectivity. (b) conventional LSRTM image after 20 iterations. (c) U-net image. (d) Deep-LSRTM image. The yellow arrows indicate regions where it is challenging to image the bottom reflector due to illumination limitations.

of the bottom reflector near the sub-salt areas (indicated by yellow arrows), we observe higher spatial resolution and sharper events in the predictions of the learned approaches compared to those of the LSRTM image.

The Sigsbee2a example exhibits three essential points about the proposed method. First, Deep-LSRTM works in the image domain rather than the data domain, and therefore it is robust against changes in the acquisition geometry. Second, since the architecture of the projection operators is fully convolutional, our technique can also handle inputs of different sizes than the ones used in training. Moreover, input image size has no bearing on the number of trainable parameters; they are determined by the number of hidden layers and feature maps. Finally, although our framework is trained on a simulated dataset containing only simple structures with shallow depths, it remains applicable to more complex data featuring salt bodies and regions with severe illumination loss. We attribute this behavior to the iterative nature of the algorithm, which leads to more accurate inversion results because it incorporates a feedback mechanism that promotes consistency with the physics of the problem. We stress that our method may not perform well for all sorts of data. However, this experiment demonstrates that the reconstructions may still be acceptable for some testing data that differs from the training set.

2.3.7 Example 4: Gulf of Mexico data set

The last example comprises the implementation of our method on a 2D Gulf of Mexico (GOM) marine line, known as the Mississippi Canyon data set (Dragoset, 1999). Such a data set is useful to test the generalization capability of our learned iterative approach on true Earth models with inevitably inaccurate migration velocities and hugely different structures than those learned during training. The velocity model used for migration is shown in Figure 2.20a (Guitton et al., 2012), containing a shallow salt body in a deep water environment. The model is discretized in a regular grid with 2908×350 points, corresponding to a distance of 29.08 km and a depth of 3.5 km with 10 m of grid spacing. The acquisition setup is off-end, with 809 sources and 180 receivers per shot moving from left to right at the surface. Receiver and shot spacing is 26.67 m. The farthest offset is 4875 m, and the nearest offset is 100 m. The seismic data underwent surface-related multiples elimination prior to imaging. To decrease the computational cost, we selected only 179 evenly separated shots of the original 809 shots for migration, 133 m apart. The shot decimation also introduces truncation artifacts, which can be used to assess how successful the proposed method is at reducing acquisition artifacts (Rocha et al., 2018; Guitton, 2012). Figure 2.20b shows the RTM result. The strong velocity contrast around the top of the salt produces strong amplitude, low-frequency artifacts in this model region.

Furthermore, sub-salt events suffer from uneven illumination, visible migration artifacts, and limited resolution. Figure 2.20c shows the LSRTM result after 20 iterations of preconditioned CGLS, which provides a noticeable improvement in the image, displaying more bandwidth content, reduced artifacts, and better-resolved reflectors, especially near and beneath the salt body. Figure 2.20d shows a direct implementation of Deep-LSRTM, which produces an artificial enhancement of the amplitude in depth. However, the output synthesized by the learned iterative approach still displays patterns seen in the training set, clearly indicating the risk of performing learned reconstructions on data that have undergone a significant distributional shift. We use the previously discussed transfer learning scheme to improve this result further. The reference model used as the ground-truth image is extracted using a different group of 60 shots from the same dataset with source spacing of 400 m and performing 20 iterations of the preconditioned CGLS. Then, we retrain the weights of each updating operator with only 20 additional epochs and a reduced learning rate of 1e-5. The result of Deep-LSRTM after transfer learning is shown in Figure2.20e. Transfer learning has a significant impact on the predictions of the learned iterative approach. Interestingly, the outcome of the updated networks has better preservation and continuity of reflectors at the top of the salt and produces a cleaner image with fewer low frequency content than the conventional LSRTM result. This might be because the original weights were trained with clean labels, so Deep-LSRTM already possesses a filtering effect by learning how to remove

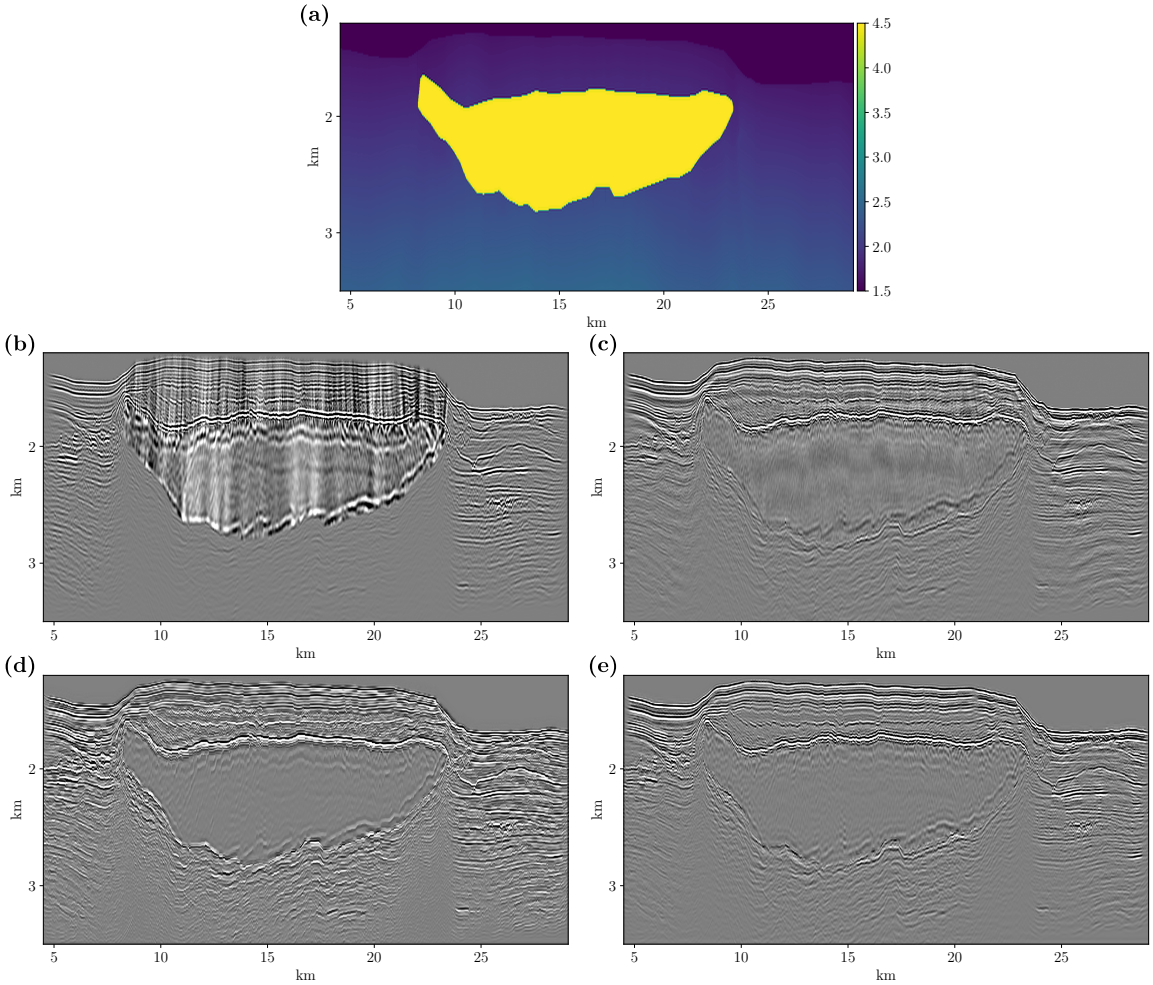

Figure 2.20: Gulf of Mexico field data example. (a) Migration velocity field. (b) RTM result. (c) LSRTM result. (d) Deep-LSRTM without transfer learning. (e) Deep-LSRTM after transfer learning.

migration artifacts efficiently (Yosinski et al., 2014). Deep-LSRTM is also more efficient in removing unwanted artifacts product of significant velocity errors as illustrated in Example 2.

Since there is no label for the real data case to validate the results, we show the wavenumber spectra for the results of LSRTM, Deep-LSRTM before transfer learning, and Deep-LSRTM after transfer learning in Figures 2.21a, 2.21b, and 2.21c, respectively, where it can be notice that the retrained approach produces a wavenumber that is closer to the LSRTM result. In addition, Figure 2.22 shows the demigrated shot gathers for a source located in the central part of the model at 15.6 km. While the adjoint method cannot match the data due to amplitude and phase mismatches, all the inversion results yield a better correlation with the observed recorded data. Figure 2.23 shows this behavior in more detail by comparing three contiguous near-offset traces extracted from the shot mentioned above. Comparing the demigrated shots obtained using the inverted results (Figures 2.22c, 2.22d, and 2.22e), we notice that while the change is not dramatic for near offset traces, the waveform fit has been improved in the mid-to-long offsets after the use of transfer learning.

2.4 Conclusions

Based on recent advances in deep learning, we have built an LSRTM framework that leverages the universal approximation capabilities of CNNs to predict reflectivity updates by mimicking a projected gradient descent algorithm. Tests on synthetic data show that, despite using a reasonably small training set, the iterative Deep-LSRTM approach yields superior results than conventional LSRTM baselines for the same number of iterations. Once trained, the computational cost per iteration of the learned projection method is similar to one iteration of the CGLS algorithm, but it requires fewer iterations to deliver high-resolution results. When evaluated on the testing data, Deep-LSRTM also outperforms a two-step residual U-net post-migration application according to the PSNR and SSIM scores, highlighting the value of including the forward and adjoint wave operators in the inference process. Satisfactory results over the central part of the Marmousi model and the Sigsbee2a model (out-of-distribution samples) confirm that the Deep-LSRTM network is not severely influenced by model over-fitting. On the other hand, the modified residual U-net method stood as an efficient alternative, filtering RTM images to obtain enhanced reflectivity approximations. However, this comes at the trade-off of being more susceptible to over-fitting to the training data. For the Marmousi test, both learned methods proved to be stable to different degrees of smoothing of the migration velocity model, while the multi-scale architecture of the U-net provides more robust performance for seismic data with higher levels of Gaussian noise. Additional iterations of the warm-started CGLS algorithm using either

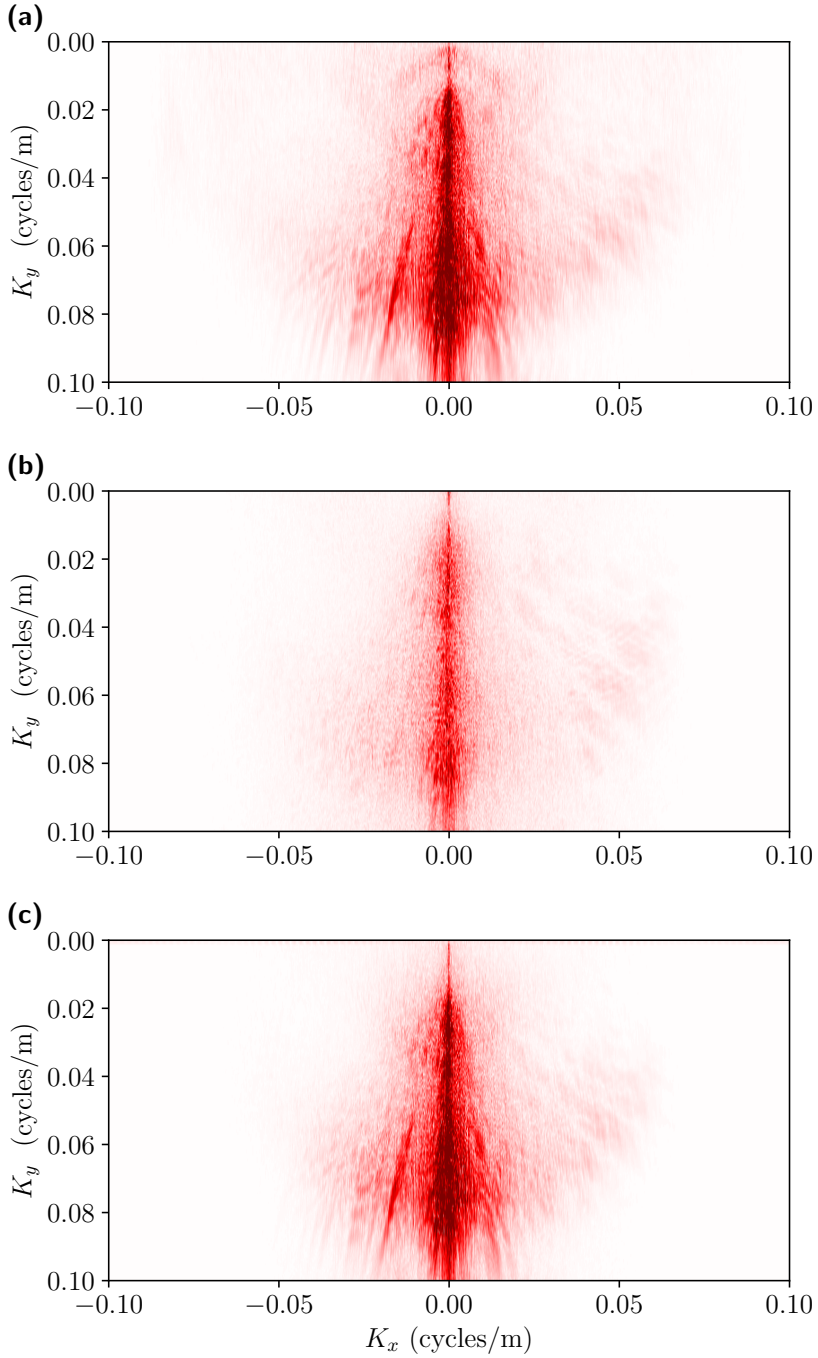

Figure 2.21: 2D wavenumber spectrum of (a) LSRTM, (b) Deep-LSRTM before transfer learning, (c) Deep-LSRTM after transfer learning. All three plots use the same color scale.

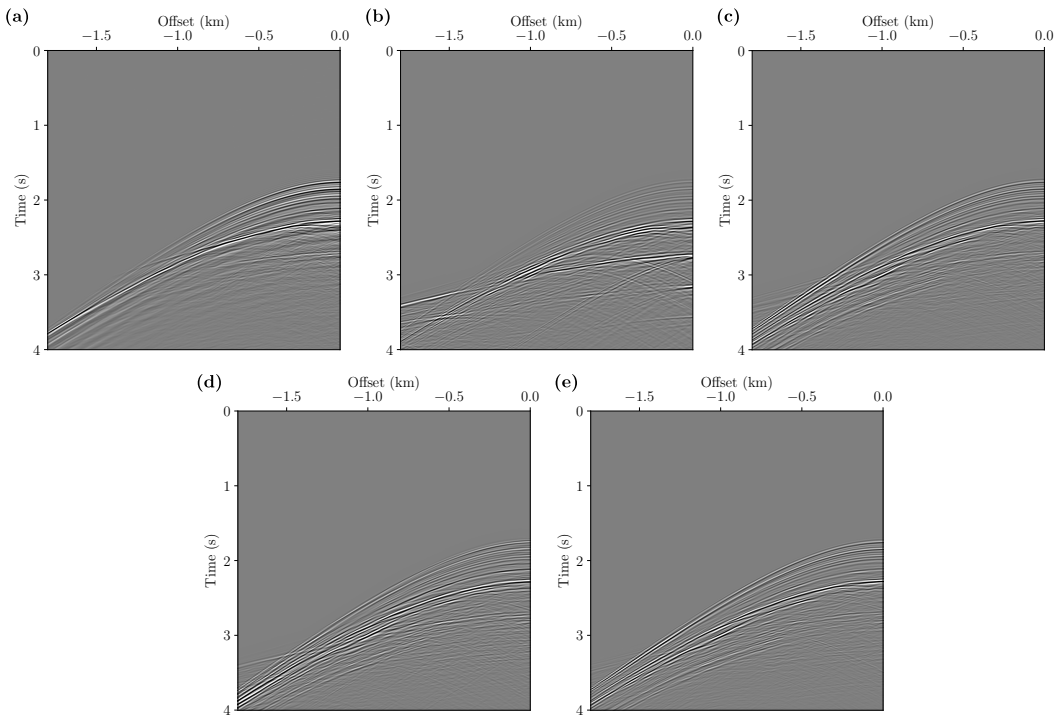

Figure 2.22: The shot gather of a source located at x=15.6 km for (a) the observed data set, (b) the demigrated data using the RTM image, (c) the demigrated data using the LSRTM image, (d) the demigrated data using Deep-LSRTM without transfer learning, and (e) the demigrated data using Deep-LSRTM after transfer learning.

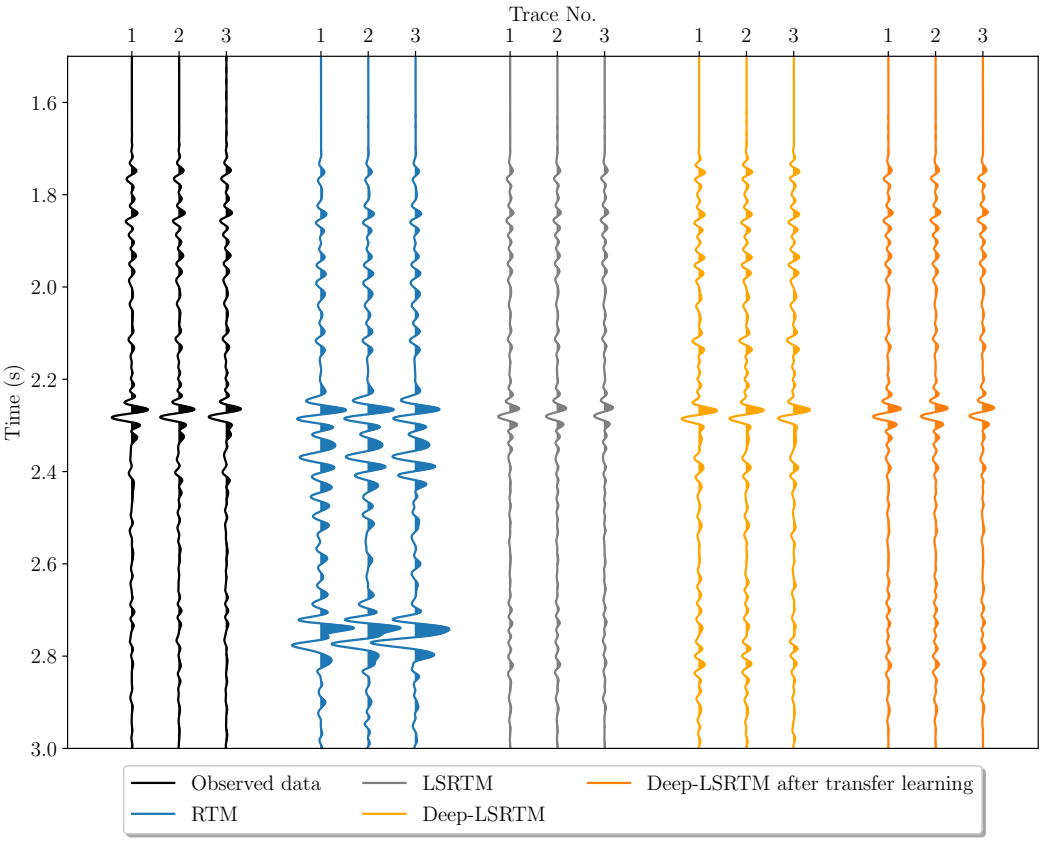

Figure 2.23: Amplitude comparison of three contiguous near-offset traces extracted from the shot gathers displayed in Figure 2.22.

of the results given by the learned approaches as initial guesses led to higher PSNR and SSIM scores. A direct implementation of Deep-LSRTM on the Mississippi Canyon data set shows that the technique is sensitive to the different characteristics of the seismic data. Transfer learning helps to adapt the trained weights to a new distribution at a fraction of conventional LSRTM using only a reduced portion of the observed data. However, future research is needed to find the optimal number of shots for transfer learning and analyze how it affects the final prediction. It is significant for us also to study more efficient ways of adapting our method to real data sets. This will be the core of our future investigation.

The proposed approach is not limited to the 2D acoustic RTM engine and can potentially be extended to other migration techniques and 3D models. Future research will explore the performance of Deep-LSRTM in 3D large-scale acquisitions. Moreover, we expect that incorporating more information into the networks using, for example, extended domain imaging conditions can help improve reconstruction performance in scenarios with wrong velocity models. Identifying uncertainty in DNN solutions to inverse problems is still in its early stage. Still, studies that consider uncertainty quantification of the proposed network architecture are also further required.

CHAPTER 3

Seismic imaging through neural network-based data- and model-space preconditioning [1]

3.1 Introduction

The ultimate goal of seismic imaging is to retrieve the underground reflectivity model from seismic reflection data measured at the surface. To this end, a linearized waveform inversion problem with an accurate migration velocity enables the least-squares migration (LSM) inversion, capable of extracting valuable information from pre-stack data to provide true amplitude and high-resolution reflectivity images (Lailly and Bednar, 1983; Nemeth et al., 1999). LSM is cast as a linear inverse problem where the Hessian (normal operator), related to the seismic resolution and subsurface illumination, is implicitly inverted iteratively (Whitmore, 1983). Since the Hessian operator encodes information about the velocity model, the acquisition geometry, and the band-limited data, inverting it results in deblurred images and reduced artifacts generated by complex overburden, limited acquisition geometry, and band-limited wavefields (Wang et al., 2017b). In the acoustic case, the linearization in LSM entails keeping the background velocity model fixed over iterations while estimating a high-wavenumber perturbation (i.e., reflectivity) that honours the best fit of the single scattered wavefield in a least-squares sense (Tarantola, 1984b). This is commonly referred to as a data-domain implementation of LSM, which attempts to minimize the difference between observed and modeled data so that the inverted reflectivity closely resembles the recorded seismic data when demigrated.

[1] A version of this chapter has been published as a conference proceeding: A deep-learning inverse Hessian preconditioning for iterative least-squares migration. In 84th EAGE Annual Conference & Exhibition (Vol. 2023, No. 1, pp. 1-5)

As with many other large-scale geophysical problems, LSM is commonly solved through gradient optimization. Regardless of what solver is chosen to update the solution (e.g. a conjugate gradient or quasi-Newton method), LSM generally requires multiple iterations to achieve an acceptable result. Slow convergence is expected due to limitations in the forward modelling operator, noise in the data and a non-trivial null space, which render LSM a very ill-posed and ill-conditioned problem. Despite often exploiting regularization or preconditioning strategies to make the problem more stable and better conditioned, LSM inevitably incurs substantial computational costs because each iteration demands the implementation of intensive adjoint (pre-stack migration) and forward (demigration) operators (Kühl and Sacchi, 2003; Clapp, 2005; Wang et al., 2005; Wang and Sacchi, 2007a; Dutta, 2017; Witte et al., 2017; Li et al., 2020a). Furthermore, when the reverse time migration (RTM) technique is used as the migration engine to exploit the potential of the two-way wave equation, the cost of iterative data inversion (referred to as least-squares reverse time migration, LSRTM) can severely hinder large-scale 3D applications (Dai et al., 2012).

Conversely, LSM can be adapted to the image domain, explicitly formulated as an image deblurring problem (Fletcher et al., 2016; Wang et al., 2017b). Image-domain LSM inverts for a reflectivity model to fit the adjoint migrated section. It can reduce the computational cost of the inversion because the input is a 2D/3D migration image, which has significantly smaller dimensions than the 3D/5D pre-stack input in the data domain. However, gradient-based iterative inversion is still needed to harness the full potential of LSM, and, theoretically, the data domain approach is more accurate than the image-domain formulation in solving the inverse Hessian (Schuster, 2017; Wang et al., 2017a). Data-domain LSM also can handle cases in which image-domain Hessian representations, such as point-spread functions, are not locally supported (e.g., simultaneous source migration and migration of multiples) (Xu et al., 2022a). Regardless of the implementation domain, the primary disadvantage of iterative LSM stems from the extensive computations required during each iteration. Therefore, implicitly inverting the Hessian can be impractical, especially for 3D data sets, as tens of iterations are typically required to gain substantial improvements.

3.1.1 Previous work

Multiple formulations have been proposed to alleviate the computational aspect of LSM in both data and image domains. As an explicit computation of the Hessian is prohibitively expensive outside of target-oriented inversions (Valenciano et al., 2006; Tang, 2009), a common goal of these techniques is to bypass the storage of many Green's functions by finding inexpensive low-rank approximations of the Hessian or its inverse. For example, to account for uneven illumination, practitioners frequently scale the migrated image with approximations of the inverse Hessian diagonal (Shin et al., 2001; Rickett, 2003; Guitton et al., 2007; Sacchi

et al., 2007; Liu et al., 2013). To increase the resolution, it is still necessary to consider the off-diagonal terms, so the diagonal Hessian estimate can instead be used as a preconditioner for data domain LSM to double the convergence rate of the conjugate gradient method (Luo and Schuster, 1991; Chen and Sacchi, 2017a). Similarly, Aoki and Schuster (2009) propose to accelerate LSM with deblurring filters within regularization and preconditioning schemes. Alternatively, a single-iteration technique known as migration deconvolution (Hu et al., 2001; Yu et al., 2006) assumes a 1D layered media to approximate a horizontally invariant inverse Hessian where off-diagonal elements can be cheaply incorporated. Another non-iterative approach based on a bank of matching filters approximates the inverse Hessian locally with a non-stationary convolution operator (Guitton, 2004; Greer et al., 2018). The matching filters are helpful for amplitude and kinematic corrections of migrated images and can be regarded as a low-rank approximation of the inverse Hessian (Guitton, 2004). Nonetheless, this technique offers little improvement in resolution because, even though it is designed to match lower-resolution images to higher-resolution images, it cannot effectively recover higher wave numbers through convolution. In other words, the matching filters do not fill the null space of the Hessian. Later, Guitton (2017) suggests using the matching filtering technique as a preconditioning operator to speed up the convergence of iterative 3D LSM. Wang et al. (2016) extend this idea to image-domain LSM by estimating the matching filters between migrated and re-migrated sections in the curvelet domain, while Herrmann et al. (2009) and Wang et al. (2017a) use a similar curvelet-based approach to data-domain LSM. Similarly, Khalil et al. (2016) propose a data-domain preconditioning technique, which only requires one demigration and two migration operations. Instead of working directly in the model space, this method computes non-stationary filters by matching demigrated and observed data, effectively approximating an inverse to the data-domain Hessian (Guo et al., 2022; Operto et al., 2023). Liu and Peter (2018) and Liu et al. (2019) explore later variations of this idea by applying Wiener and Gabor deconvolution approaches, respectively. Other methods involve creating Hessian approximations using operators of a manageable size by combining Kronecker products (Gao et al., 2020a), a sequence of space and frequency weights (Tangkijwanichakul and Fomel, 2021), and localized block-wise matrices (Jiang and Zhang, 2019).

Recent years have witnessed a surge of interest in machine learning methods to aid geophysical inversion, encompassing applications ranging from seismic denoising to full-waveform inversion (Yu and Ma, 2021b; Adler et al., 2021b). Considering the data-intensive nature of the LSM problem, researchers have developed deep-learning techniques, mostly based on convolutional neural networks (CNN), to enhance the resolution of migrated images. A CNN is a type of neural network composed of a sequence of convolutional operators (layers), which enhances its ability to express complex functions as the number of layers and neurons in each layer increases (Goodfellow et al., 2016).

For instance, Chen et al. (2020) and Liu et al. (2020b) design a supervised denoising filter trained to remove migration artifacts. Kaur et al. (2020) use generative adversarial networks (GAN) (a specific type of neural networks typically comprising two CNNs, a generator and a discriminator) in a conditional framework to simulate the inverse Hessian from migrated images and synthetic reflectivity labels. To accelerate and regularize iterative LSRTM, Torres and Sacchi (2022a) substitute the projection operator of a gradient-descent formulation by pre-trained deep CNNs. Kumar et al. (2022) introduced a deep learning variant of the matching filtering method, replacing the filter coefficients with a non-linear mapping between migrated and remigrated images, parameterized by a CNN. Moreover, inspired by the success of stochastic solvers for training neural networks on large datasets, mini-batch optimization has been applied to LSRTM using both first (Vamaraju et al., 2021) and second (Farias et al., 2023) order algorithms. This approach uses subsets of shot gathers and is more computationally efficient than traditional (full-batch) LSRTM, based on initial studies using noiseless synthetic data. The references provided are not exhaustive given the vast amount of literature on these subjects. We have selected a representative group of citations that interested readers can use as a starting point for further exploration.

3.1.2 Deep-learning-based preconditioning

This chapter introduces two novel deep-learning-based preconditioning strategies for seismic imaging.

The first strategy acts on the model (image) space and consists of two distinct building blocks. The first component follows Kumar et al. (2022) and estimates the effect of the inverse Hessian by training a CNN from pairs of migrated and re-migrated images. However, instead of a fully convolutional autoencoder, we introduce a deep autoencoder with a 1D lower-rank representation to benefit from its dimensionality reduction capabilities. We build the training dataset from small overlapping patches of migrated (high-fidelity data) and re-migrated (low-fidelity data) images, randomly cropped in the shot-index domain. The technique only requires paired training samples obtained from the available seismic data and the action of the physical operators, circumventing the need for a representative dataset of ground-truth reflectivity labels. Once we get the approximation of the inverse Hessian, the second component solves LSRTM directly on a lower-dimensional space by integrating the trained decoder in the optimization problem through a (non-linear) change of variables. Specifically, the decoder connects a high-dimensional model space, where forward simulations occur, to a lower-dimensional (latent) space using a series of non-linearities and convolution operations. Since the decoder learns to synthesize model realizations from low-dimensional representations of the discrepancy between high-fidelity and low-fidelity images, it has a preconditioning effect that can be used to enhance model features related to images

with reduced artifacts and more illumination balance. Combining the deep-learning inverse Hessian with LSRTM potentially overcomes the resolution limitations of single-iteration matching filtering. Also, given the reduced number of inferred parameters and the fast generation of enhanced model realizations, the deep-learning parameterization significantly improves the LSRTM inversion performance.

The second strategy is a single-step adaptation of the learned model-space preconditioning scheme described above. However, it operates in the data space rather than in the image space, and only requires a single demigration and an extra migration. It is mainly designed to enhance RTM images by training a CNN to estimage non-linear matching filters on input-output pairs of observed and demigrated data. After training, the CNN filters can be directly applied to the observed data before the migration process to enhance the adjoint results. The method's effectiveness is demonstrated through tests conducted on the Marmousi data set, showing that the CNN data preconditioning approach improves amplitude balance and focusing around under-illuminated regions, and reduces artifacts compared to the original RTM section.

3.2 Learned model-domain preconditioning

In this section, we review the mathematical formulation of LSM and establish the foundations for computing the action of the inverse Hessian based on the deep convolutional autoencoder. Then, we show examples of estimating the reflectivity using iterative non-linear inversion with and without the model reduction preconditioning scheme. We rely on empirical testing to examine the convergence properties of the method. We also include a discussion of limitations and prospects for further research.

3.2.1 Theory

We aim to estimate a reflectivity model vector $\mathbf{m} \in \mathbb{R}^n$ from the single scattered seismic data $\mathbf{d} \in \mathbb{R}^m$. Assuming a suitable discretization, a linear relation between the model and data spaces can be established employing the forward Born modelling operator $\mathbf{L} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, such that

$$\mathbf{d} = \mathbf{L}\mathbf{m}. \tag{3.1}$$

With an accurate background velocity model, a migrated image $\mathbf{m}_{\text{mig}}$ gives us an initial approximation of the subsurface reflectivity regarding reliable structural information. It is computed by applying the adjoint of the Born modelling operator $\mathbf{L}^T : \mathbb{R}^m \rightarrow \mathbb{R}^n$ to the

recorded data $\mathbf{d}_{\text{obs}}$ (Claerbout, 1992):

$$\mathbf{m}_{\text{mig}} = \mathbf{L}^T \mathbf{d}_{\text{obs}}. \tag{3.2}$$

$\mathbf{L}^T$ and $\mathbf{L}$ are compact representations of two computer subroutines (i.e., we never explicitly construct such matrices): a standard RTM with cross-correlation imaging condition and demigration modelling, respectively. We carefully design our migration and demigration engines to form an exact adjoint pair that passes the dot-product test with numerical precision (Claerbout, 1992). However, as migration does not correspond to inversion, i.e., $\mathbf{L}^T\mathbf{L} \neq \mathbf{I}$, the migrated image is blurred, artifact-prone and has uneven amplitudes (Gray, 1997). To find an approximate inverse to the Born modelling operator, LSM minimizes the quadratic objective function

$$J(\mathbf{m}) = \frac{1}{2}||\mathbf{L}\mathbf{m} - \mathbf{d}_{\text{obs}}||_2^2, \tag{3.3}$$

where $||\cdot||_2$ indicates the l_2 norm. Equation 3.3 has the formal least-squares solution

$$\mathbf{L}^T\mathbf{L}\mathbf{m} = \mathbf{L}^T\mathbf{d}_{\text{obs}}, \tag{3.4}$$

where $\mathbf{H} = \mathbf{L}^T\mathbf{L}$ is the square and symmetric Hessian matrix with elements corresponding to second-order derivatives of $J(\mathbf{m})$ with respect to $\mathbf{m}$. Although $\mathbf{H}$ possesses zero or small singular values corresponding to shadow zones, we assume it to be invertible; otherwise, a regularization term needs to be incorporated into equation 3.3. Alternatively, the image domain formulation of LSM rewrites equation 3.4 in terms of the Hessian and the migrated image

$$\mathbf{H}\mathbf{m} = \mathbf{m}_{\text{mig}}, \tag{3.5}$$

and seeks a true image $\mathbf{m} = \mathbf{H}^{-1}\mathbf{m}_{\text{mig}}$ by minimizing the model-space objective function

$$J(\mathbf{m}) = \frac{1}{2}||\mathbf{H}\mathbf{m} - \mathbf{m}_{\text{mig}}||_2^2. \tag{3.6}$$

It is too expensive to directly compute the inverse Hessian $\mathbf{H}^{-1}$. Instead, a gradient-based procedure can be used to find an iterative solution

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \alpha\Delta\mathbf{m}_k, \tag{3.7}$$

where k denotes the current iteration, α is the step-length, and $\Delta\mathbf{m}_k$ is the update direction. In the simplest case, $\Delta\mathbf{m}_k$ is equal to the cost-function gradient given by

$$\nabla J(\mathbf{m}_k) = \frac{\partial J(\mathbf{m}_k)}{\partial \mathbf{m}_k} \tag{3.8}$$

$$= \mathbf{L}^T(\mathbf{L}\mathbf{m}_k - \mathbf{d}_{\text{obs}}) \tag{3.9}$$

$$= \mathbf{L}^T\mathbf{L}\mathbf{m}_k - \mathbf{m}_{\text{mig}}. \tag{3.10}$$

To accelerate convergence, we can roughly mimic a Gauss-Newton method (Schuster, 2017; Fichtner, 2010)

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \alpha(\mathbf{L}^T\mathbf{L})^{-1}\nabla J(\mathbf{m}_k), \tag{3.11}$$

with a preconditioner $\mathbf{P}$, and compute iterative reflectivity updates

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \alpha\mathbf{P}\nabla J(\mathbf{m}_k), \tag{3.12}$$

such that $\mathbf{P} \approx (\mathbf{L}^T\mathbf{L})^{-1}$ is a preconditioner operator that approximates the effect of the inverse Hessian and acts on the model domain.

Approximating the inverse Hessian with matching filters

As noticed by Guitton (2004), we can reproduce the effect of the Hessian (equation 3.5) by a demigration/migration sequence

$$\mathbf{L}^T\mathbf{L}\mathbf{m}_1 = \mathbf{m}_2, \tag{3.13}$$

with $\mathbf{m}_1 = \mathbf{L}^T\mathbf{d}_{\text{obs}}$. As $\mathbf{m}_1$ and $\mathbf{m}_2$ are known, a crude approximation of the inverse Hessian can be found by defining a non-stationary convolutional operator $\mathbf{P} \approx \mathbf{H}^{-1}$ and minimizing the cost function

$$E(\mathbf{P}) = ||\mathbf{m}_1 - \mathbf{P}\mathbf{m}_2||_2^2 + \lambda R(\mathbf{P}), \tag{3.14}$$

where $R(\mathbf{P})$ is a regularization term added for stability, and λ is the trade-off parameter. After solving for $\mathbf{P}$, we obtain an improved image through a single-step filtering,

$$\hat{\mathbf{m}} = \mathbf{P}\mathbf{m}_1, \tag{3.15}$$

or by incorporating $\mathbf{P}$ into the iterative scheme described in equation 3.12 to precondition the gradient at each iteration.

Alternatively, we propose parameterizing the sought preconditioning operator by the weights of a deep convolutional autoencoder (CAE), a variant of the basic autoencoder model that

incorporates convolutional layers into both its encoder and decoder components (Goodfellow et al., 2016).

Approximating the inverse Hessian with a deep autoencoder

A deep autoencoder is a non-linear dimensionality reduction method inspired by the linear principal component analysis (PCA) technique (Kramer, 1991). Along with other generative models such as GANs and Variational Autoencoders (VAEs), it can provide a low-dimensional representation of the model space that can be directly exploited as priors in various inverse problems (Gao et al., 2019, 2020b; Chen and Schuster, 2020; Asim et al., 2020; Hyun et al., 2021; Ravasi, 2021; Liu et al., 2022; Levy et al., 2022). A CAE comprises the deterministic pair (E_ϕ, D_θ) of CNNs, where $E_\phi : \mathbb{R}^n \to \mathbb{R}^h$ denotes the encoder network parameterized by weights ϕ, $D_\theta : \mathbb{R}^h \to \mathbb{R}^n$ corresponds to the decoder network parameterized by weights θ, and h is the latent space dimension. For $n > h$, E_ϕ is trained to encode samples $\mathbf{x} \in \mathbb{R}^n$ in the lower-dimensional space $\mathbb{R}^h$ such that D_θ can reconstruct an estimated sample $\hat{\mathbf{x}}$ from its latent representation $\mathbf{z} \in \mathbb{R}^h$ through a reverse mapping. In this case, the latent representation vector of fixed dimensions $\mathbf{z} = E_\phi(\mathbf{x})$ is an informational bottleneck, which induces the CAE to capture the most important features of the input sample. For example, we can express the encoder part of a simple CAE, in which E_ϕ comprises only two hidden layers: one convolutional and one fully connected. In this case, the latent representation output is

$$\mathbf{z} = \sigma\big(\mathbf{W}^2(\sigma(\mathbf{W}^1 * \mathbf{x} + \mathbf{b}^1)) + \mathbf{b}^2\big), \tag{3.16}$$

with σ an element-to-element non-linear function (e.g. ReLU, LeakyReLU, Tanh), $\mathbf{b}^l$ denoting the bias vectors, and $\mathbf{W}^l$ representing the filters and weights of the convolutional ($l = 1$) and fully connected ($l = 2$) layers, respectively. For this example, $\phi = [\mathbf{W}^1, \mathbf{b}^1, \mathbf{W}^2, \mathbf{b}^2]$. Analogously, a symmetric decoder maps from latent to model space in a transposed manner with weights $\theta = [\tilde{\mathbf{W}}^1, \tilde{\mathbf{b}}^1, \tilde{\mathbf{W}}^2, \tilde{\mathbf{b}}^2]$.

Generally, CAEs are trained with unsupervised algorithms so that the output samples approximate the inputs based on the latent representation. A practical choice to adjust the CAE weights in the training stage, given a training dataset with N samples, is to minimize the loss function

$$E(\phi, \theta) = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{x}^i - D_\theta(E_\phi(\mathbf{x}^i))||_2^2. \tag{3.17}$$

Restricting $h < n$ serves as a form of regularization, preventing the CAE from becoming an identity mapping. One could also add regularization terms acting on the network weights or the latent variable to avoid overfitting and enforce structure (e.g. sparsity) on the lower-

dimensional space (Obmann et al., 2020).

The unsupervised training strategy in the equation 3.17 assumes access to N input samples $\{\mathbf{x}^i\}_{i=1}^N$ from a representative dataset. Therefore, there is still implicitly assumed that ground-truth images are available for training. Thus, the achievable performance of deep autoencoder methods is inherently bounded by the amount and quality of the available ground-truth data. Collecting high-quality and high-resolution reflectivity images from a vastly heterogeneous subsurface is an elusive task in seismic imaging. Furthermore, seismic imaging is computationally intensive, and we cannot realistically rely on a bank of thousands of LSM-generated images. This is also different from other geophysical problems, where high-fidelity representations can be inferred from re-organizing the available data in another domain or where the uncorrupted signal can be easily accessed (e.g. seismic deblending or interpolation) (Xu et al., 2022b). We partially overcome this issue by relying on the migration and re-migration approach. Based on equation 3.13, we rewrite the inverse Hessian approximation problem as

$$\mathbf{m}_1 = D_\theta(E_\phi(\mathbf{m}_2)), \tag{3.18}$$

and establish a supervised training strategy given by

$$E(\phi,\theta) = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{m}_1^i - D_\theta(E_\phi(\mathbf{m}_2^i))||_2^2 + \beta||\theta||_2^2 + \lambda||\phi||_2^2. \tag{3.19}$$

where $\mu, \lambda > 0$. Similar to the linear matching filtering procedure (notice the correspondence between equations 3.14 and 3.19), this training strategy enforces the CAE to learn a filtering task as a function of the migrated and re-migrated paired samples. To have access to multiple realizations of $\mathbf{m}_1$ and $\mathbf{m}_2$ within the same acquisition setup, we build our training dataset with random crops of overlapping patches from the volume of unstacked images $\tilde{\mathbf{m}}_1$ and re-migrated images $\tilde{\mathbf{m}}_2$. Such volumes are obtained using the source extended imaging condition (Huang et al., 2016). More specifically, our bank of training high-fidelity and low-fidelity samples is defined as

$$\tilde{\mathbf{m}}_1 = \begin{pmatrix} \mathbf{m}_1^1 \\ \mathbf{m}_1^2 \\ \vdots \\ \mathbf{m}_1^{Ns} \end{pmatrix}; \qquad \tilde{\mathbf{m}}_2 = \begin{pmatrix} \mathbf{m}_2^1 \\ \mathbf{m}_2^2 \\ \vdots \\ \mathbf{m}_2^{Ns} \end{pmatrix}, \tag{3.20}$$

where the super-indexes $1, 2, ..., Ns$ denote the corresponding shot number.

LSRTM in the latent space

After the training stage, we use the decoder as a non-linear synthesis operator to solve data-domain LSRTM in the latent space. This amounts to formulating the new problem as:

$$J(\mathbf{z}) = \frac{1}{2}||\mathbf{L}D_\theta(\mathbf{z}) - \mathbf{d}_{\text{obs}}||_2^2 \tag{3.21}$$

$$\hat{\mathbf{z}} = \text{argmin}_{\mathbf{z}}\ J(\mathbf{z}) \tag{3.22}$$

$$\hat{\mathbf{m}} = D_\theta(\hat{\mathbf{z}}), \tag{3.23}$$

where, $\hat{\mathbf{m}}$ is the final inverted reflectivity. Since it is often impractical to encode the entire reflectivity model into a single latent vector due to computational constraints and the risk of information loss, we employ a patching operator that allows for a more distributed and detailed encoding process. Thus, in practice, we split the sought reflectivity model in N_p overlapping patches of pre-defined size, $\hat{\mathbf{z}} = \{\hat{\mathbf{z}}_i\}_{i=1}^{N_p}$. For simplicity, before performing the forward modelling on the full image, we attach as the last layer of the decoder an unpatching operator that assembles individually decoded patches $\{D_\theta(\mathbf{z}_i)\}_{i=1}^{N_p}$ back together using weighing functions (Claerbout and Fomel, 2008). Notice that while the result is produced by a CNN (as shown in equation 3.23), this approach does not involve training during the data inversion stage, which is a crucial aspect of other popular deep learning frameworks like Deep Image Prior (DIP) (Ulyanov et al., 2018). DIP only uses corrupted data in the reconstruction process and purely relies on the CNN architecture to act as prior. In DIP, the unsupervised training stage with a single training sample becomes the mechanism to solve the inverse problem, which might demand many iterations. In principle, our approach eliminates the need for a specific CNN architecture to exploit correlations in the data to learn their inner structure, which results in a more efficient solution.

However, since we did not train the decoder weights using ground-truth reflectivity labels, the set of non-linear basis functions they represent cannot entirely depict the true reflectivity model (i.e. the true solution is not in the decoder range) (Bora et al., 2017). In addition, LSRTM is a linear inverse problem and benefits from linear solvers such as the CGLS algorithm (Hestenes and Stiefel, 1952) to monotonically minimize the original cost function (Hansen, 2010b). By applying a non-linear transformation, we are inherently changing the topography of the cost function, and we may be at risk of ending up in one of the multiple local minimums. For these reasons, solving equation 3.21 might yield imperfect results with bounded reconstruction quality.

To overcome the first issue, future research should explore the incorporation of regularization terms to relax the constraint on the range of the decoder and allow for slight deviations. Less explicit approaches can also be considered to extend the range of the original decoder, for example, by refining intermediate layers of the network or tweaking the decoder model

in response to the observed data. For now, we employ the L-BFGS solver (Nocedal, 1980), which provides fast convergence and additional information about the local curvature of the new cost function to steer its search direction along the variable space. We rely on the decoder as a preconditioning operator to produce high-fidelity images relatively similar to those from the target distribution. Despite these limitations, we still expect to obtain valuable results comparable to some extent to least-squares iterative inversion and use the enhancing effect learned by the decoder to gain illumination and artifact attenuation at a reduced computational cost while maintaining high resolution. Finally, to find a solution for the problem depicted in equation 3.21, optimization solvers such as L-BFGS require the gradient of the new cost function with respect to the latent space variable. To solve this challenge, we incorporate the forward and adjoint wave-equation operators (migration and demigration subroutines) using the automatic differentiation functionality of *Pytorch* (Paszke et al., 2019). Based on the chain rule, the gradient can be computed as

$$\nabla J(\mathbf{z}) = \frac{\partial \mathbf{m}_k}{\partial \mathbf{z}} \frac{\partial J}{\partial \mathbf{m}_k}, \tag{3.24}$$

where the last term on the right-hand side is calculated using equation 3.9 via the adjoint-state method and then fed into equation 3.24, which is seamlessly computed with *Pytorch*'s back-propagation algorithm (Paszke et al., 2017).

The second issue, related to the local minimum, might be mitigated by correctly choosing an accurate initial estimate. In the following example, we set $\mathbf{z}_0 = E_\phi(\mathbf{m}_{\text{mig}})$ as it produces stable results. Furthermore, $\mathbf{m}_{\text{mig}}$ is available from the training stage (through a stacking operator acting on $\tilde{\mathbf{m}}_1$), so initializing the inversion with the encoded adjoint image does not incur in additional computational costs.

3.2.2 Numerical experiment

In this section, we present a proof-of-concept example through a 2D synthetic test, where we aim to find the reflectivity model of a relatively simple layered medium with slightly dipping reflectors, shown in Figure 3.1b. We obtain the background velocity model (not shown) for forward simulation and migration by smoothing the true velocity model with a moving box average. The experiment simulates a fixed spread acquisition of 65 sources and 128 receivers. The source and receiver spacings are 24 m and 12 m, respectively. The first source and receiver are at $x = 0$ m, and all sources and receivers are at $z = 0$ m depth. The spatial grid interval is 12 m in x and z. The seismic source is a Ricker wavelet of 20 Hz (dominant frequency), and the recording time is 1.8 seconds sampled at 1 ms. In this example, the observed data $\mathbf{d}_{\text{obs}}$ is noise-free, and our migration velocity is relatively accurate.

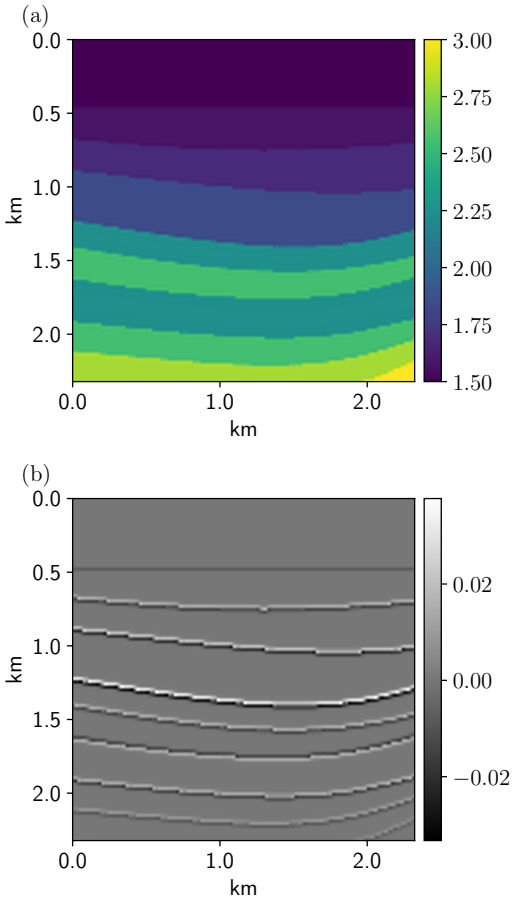

Figure 3.1: (a) The velocity model in km/s. (b) The squared slowness perturbation in s^2/km^2 representing the true reflectivity model.

Network training and evaluation

We define a fixed patch size of 64×64 for the training stage with a stride of 8×64 grid points. To build the dataset, we individually migrate and re-migrate all the shot-gathers. The total number of patches is 2304. Figures 3.2a and 3.2b show several randomly chosen patches from the bank of labels $\tilde{\mathbf{m}}_1$ and inputs $\tilde{\mathbf{m}}_2$, respectively. As expected, the patches of re-migrated images differ significantly from the $\tilde{\mathbf{m}}_1$ patches, presenting much higher amplitude imbalance and more migration-related artifacts.

To monitor the network's performance while training, we split the samples of the dataset as 90% for training and 10% for validation. Even though the method will not be used on another data set, the validation dataset helps to monitor the generalization performance on all parts of the model and different shot positions. To avoid manual annotation, as in

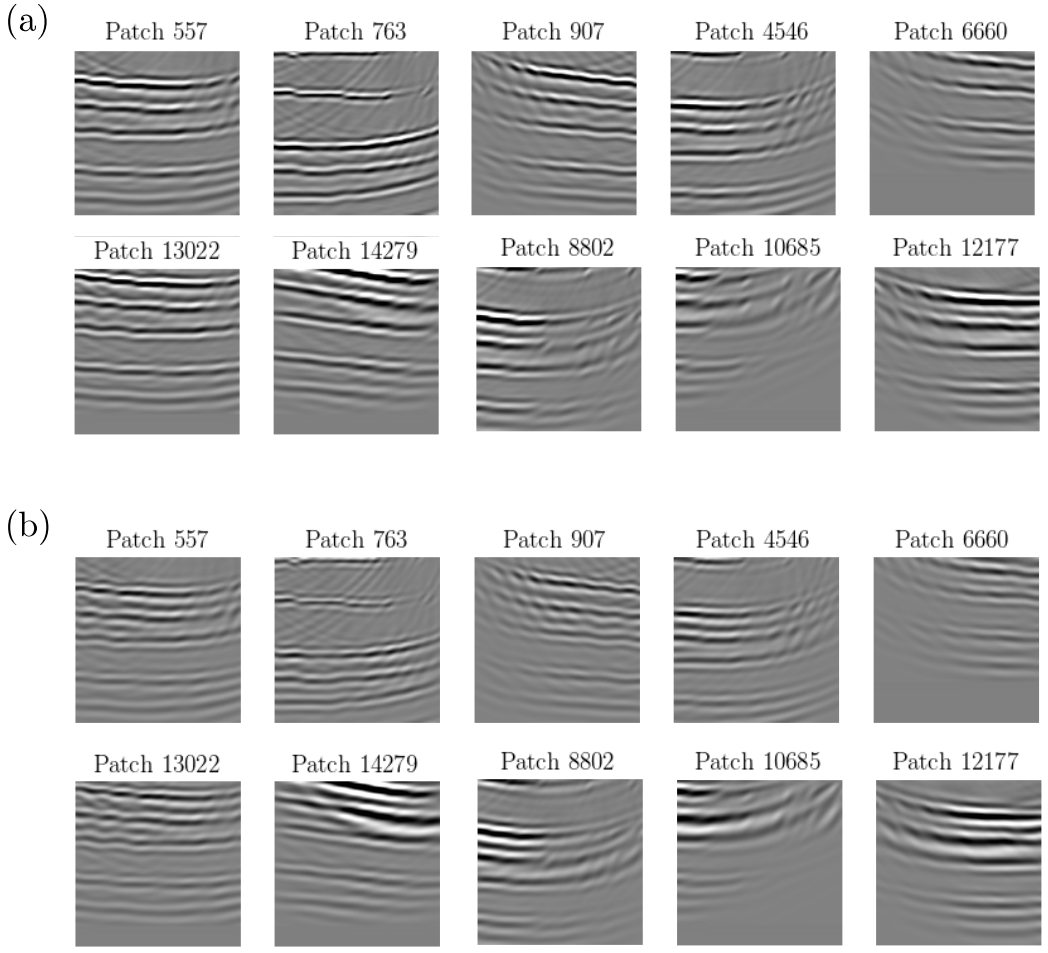

Figure 3.2: Training dataset. Random selection of paired samples: (a) Labels from $\tilde{\mathbf{m}}_1$ (high-fidelity samples). (b) Their corresponding input patches from $\tilde{\mathbf{m}}_2$ (low-fidelity samples). All the images are plotted using the same amplitude range.

self-supervised frameworks, the only pre-processing step we apply to the training dataset is a standardization technique where we subtract the mean and divide it by the standard deviation of the inputs and the labels separately. The standardization step is important to avoid sub-optimal results, given that the magnitude of the amplitudes in migrated and re-migrated images differs greatly, ranging from -10^n to 10^n. As a result, the data has a very large variance, and the training loss's convergence might be compromised. Furthermore, the activation functions of the network tend to be non-linear only around zero. Therefore, standardization helps the network learn non-linear patterns while avoiding undertrained parameters by letting data have zero mean and unit variance. The normalization is applied

globally (i.e., considering all the unpatched and individually migrated images) to avoid the risk of blowing up areas of low reflectivity responses and make them as important as those of high response. This linear transformation is given by

$$\tilde{\mathbf{m}}_{\text{std}} = \frac{\tilde{\mathbf{m}} - \mu}{\delta}, \tag{3.25}$$

where $\mathbf{m}_{\text{std}}$ stands for the standardized data, $\tilde{\mathbf{m}}$ represents either the dataset of inputs or labels, and μ and δ stand for the data mean and variance, respectively. Once normalized, we only use a random horizontal flip as a data augmentation technique during training to artificially increase the size of the dataset.

For the deep autoencoder architecture, we employ the ResNet CAE model provided in Ravasi (2021), which includes multiple residual blocks as the backbone of the network, each composed of two 2D convolutional layers (a sequence of multiple 2D convolutional filters, a batch normalization operator, and a leaky ReLU non-linear activation) and a skip connection over the two layers. To slightly improve performance, we concatenate two consecutive residual blocks on each level, producing a deeper architecture but retaining the same complexity level (He et al., 2016). We also change the number of 2D filter coefficients from 3×3 to 5×5. Generally, a larger convolution kernel means a more robust learning capacity for capturing larger-scale patterns in the data since it expands the receptive field of the convolution kernel (i.e., the region's size in the input space that each convolution operation acts upon). For the following example, we obtain at least a moderately increased performance with these modifications regarding training stability and prediction quality. The total trainable filter coefficients (network parameters) are approximately 15 million. The input shape is hard coded to the dataset dimensionality (patch size), and the dimension of the latent space is fixed to $h = 300$. We use the Adam method (Kingma and Ba, 2014a) to optimize the network parameters by minimizing equation 3.19 with 50 epochs, batch size of 256, a learning rate of $1e^{-3}$, and $\beta, \lambda = 1e^{-5}$. According to our experience, training with these hyper-parameters yields acceptable initial results.

Figure 3.3 shows the evolution of the training and validation losses (normalized by their maximum values). As expected, the validation loss is larger than the training loss for all epochs. However, we observe no significant divergence between the two curves, implying a stable and convergent training stage without significant overfitting. The increase of the average signal-to-noise ratio shown in Figure 3.4 for both training and validation datasets further supports this analysis. Figure 3.5 displays eight input-output-label triples from the validation dataset. We notice how the predicted patches from shot-index image gathers have more suppressed migration artifacts and improved amplitude balance while retaining significant reflection events.

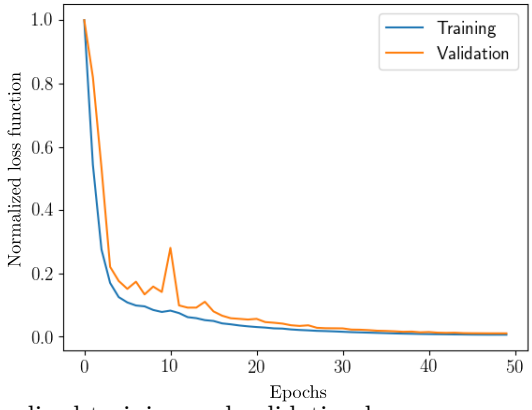

Figure 3.3: Normalized training and validation losses versus number of epochs.

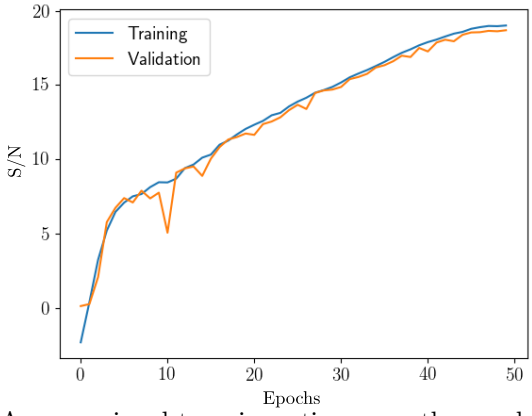

Figure 3.4: Average signal-to-noise ratio versus the number of epochs.

Solving the inverse problem

Applying the patching technique with the same configuration as in the training stage, we split the sought reflectivity model into 18 patches. This corresponds to a dimensionality reduction factor of approximately 3× compared with the original model dimensions.

First, we only invert the central shot located at $x = 774$ m. Figure 3.6c shows the results of the inversion after ten iterations of the proposed method, along with the initial RTM section (Figure 3.6a). We also run the inversion without the decoder synthesizer for comparison (Figure 3.6b). These results show that most migration artifacts have been effectively reduced in the decoder-based inversion, and the image still presents resolution improvements compared to the adjoint result. As a quasi-Newton method, the L-BFGS solver approximates the inverse Hessian with a combination of gradients from previous iterations. However, comparing Figures 3.6b and 3.6c proves that the imaging enhancement we obtain relies

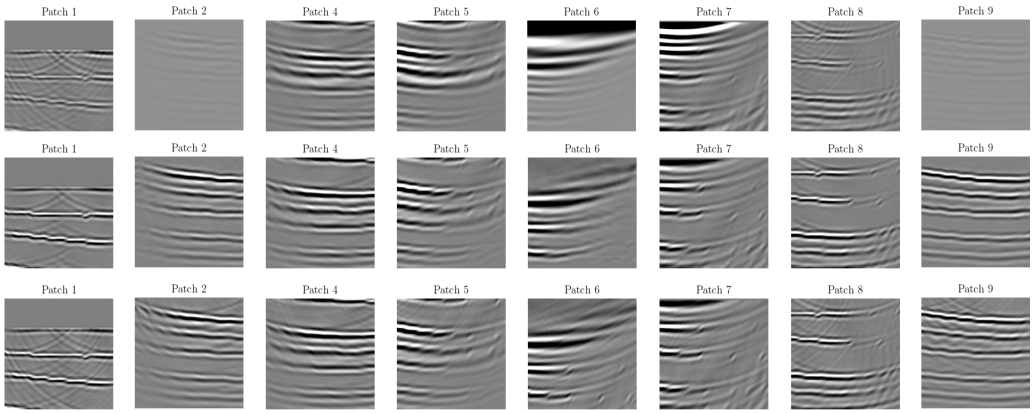

Figure 3.5: Random patches from the validation dataset: low-fidelity (top), network predictions (middle), high-fidelity labels (bottom). All the images are plotted using the same amplitude range.

mainly on the action of the decoder. Although there are some remnant artifacts produced by the inversion of a single shot-index gather, the difference between the L-BFGS and the L-BFGS with decoder results is obvious. Figure 3.7 shows the normalized objective function versus the number of iterations. Although both inversions converge to a similar value, the decoded inversion achieves convergences in fewer iterations, translating into improved computational performance. However, unlike the single L-BFGS inversion, the decoded one cannot completely reduce the misfit. Using the Born modelling operator, we can map the inverted images to the data domain and assess the reconstruction quality and consistency by comparing the predicted data with the observed data. Figure 3.8 shows the data domain comparison. As expected, the data discrepancy between the decoded inversion and the true reflectivity is not entirely zero. We hypothesize that the reconstruction is roughly restricted to the decoder range and the associated representation error due to the lower-dimensional parameterization. Increasing the latent space dimension to match the original model space results in a lower data misfit (Figure 3.8e). Convergence is achieved with the same reduced number of iterations but at the expense of increased memory requirements. A smaller data misfit given a larger latent space is reasonable since the problem turns less overdetermined as h increases.

Despite training on shot-indexed migrated images, we also observe good generalization performing the inversion on a stacked section by applying iterative LSRTM to multiple shot gathers so that only one reflectivity model explains all of the data. This amounts to solving

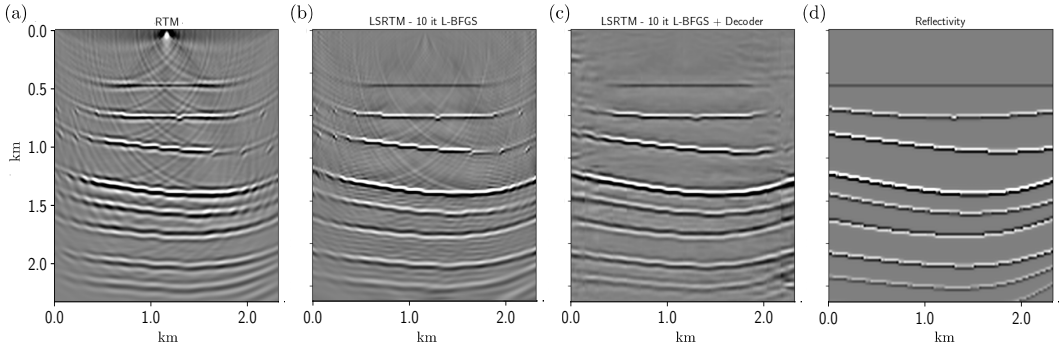

Figure 3.6: Image space results for the inversion of the central shot. (a) RTM image. (b) L-BFGS result. (c) Proposed method. (d) True reflectivity.

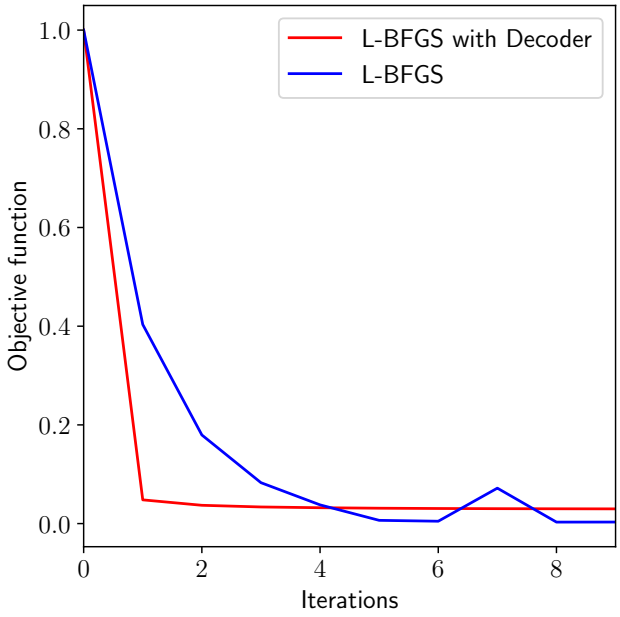

Figure 3.7: Convergence of the L-BFGS algorithm (with and without the decoder synthesizer) versus the number of iterations for the central shot inversion.

the non-linear overdetermined system of equations

$$\begin{pmatrix} \mathbf{d}_{\text{obs}}^{1} \\ \mathbf{d}_{\text{obs}}^{2} \\ \vdots \\ \mathbf{d}_{\text{obs}}^{Ns} \end{pmatrix} = \begin{pmatrix} \mathbf{L}^{1} \\ \mathbf{L}^{2} \\ \vdots \\ \mathbf{L}^{Ns} \end{pmatrix} D_{\theta}(\mathbf{z}). \tag{3.26}$$

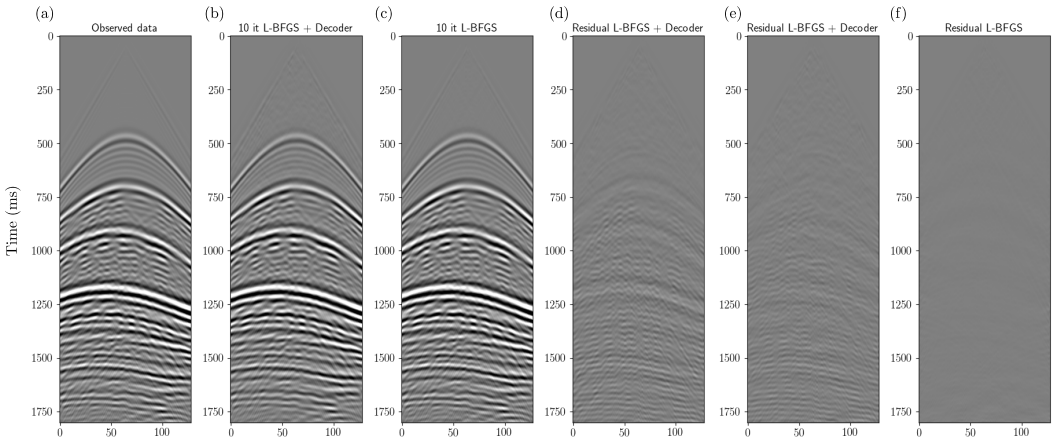

Figure 3.8: Data space results for the inversion of the central shot. The horizontal axis displays the receiver number. (a) Observed data. (b) Demigration on the L-BFGS. (c) Demigration on the proposed method result. (d) Data residual for the proposed method (b). (d) Data residual for a result with a larger latent space. (e) Data residual for single L-BFGS. All images are shown with the same amplitude range.

To show the method's potential in dealing with sparse acquisitions, we limit the multiple shot inversion to only $N_s = 10$ shots evenly spaced at the surface. Figure 3.9 shows the result of the stacked inversion. Naturally, stacking is effective in suppressing interferences observed in single-shot inversions. We still notice a substantial improvement in the decoded inversion over the traditional LSRTM result.

3.2.3 Discussion and conclusions

We developed a fast imaging framework that relies on deep learning and data domain inversion to recover an improved subsurface reflectivity model. A preliminary numerical example demonstrates the potential of the method. Compared to other supervised deep-learning techniques that need paired samples of ground-truth labels and initial reconstruction models, the proposed method does not use the former. The training stage does not need complicated pre-processing and requires minimal user interaction. It is also relatively cheap since we are simulating the effects of the Hessian operator at the cost of only one migration/re-migration sequence, equivalent to one iteration of conjugate gradients. Moreover, the deep synthesizer operator used in the inversion stage is similar in spirit to preconditioning schemes in linear inverse problems, in which the preconditioner promotes stable features in the model, improving the eigenvalue distribution of the forward operator and increasing the rate of convergence.

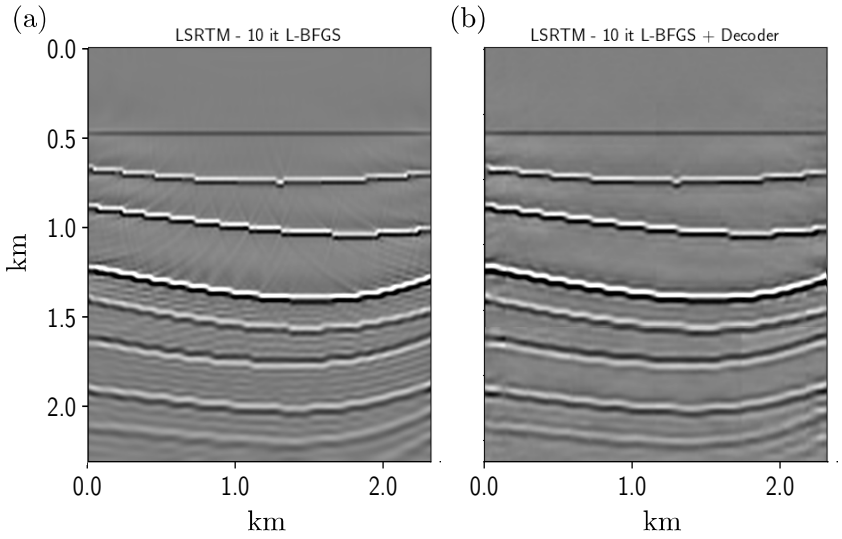

Figure 3.9: Image space results for the inversion of 10 shots. (a) Single L-BFGS. (b) Proposed method.

A tangible reduction in computational costs supports the enticing perspective of using deep-learning frames to improve the quality of images. It might be essential in making more expensive approaches, such as extended-domain LSRTM, numerically feasible. A minor variant of the method that can be the focus of future research consists of approximating the Hessian instead of its inverse, which renders a better-conditioned problem, and then solving the LSM problem in the image domain.

Despite the computational advantages of the method, certain aspects can be further enhanced. One limitation of the technique is that it currently constraints the solution to those roughly in the decoder range. One future direction is to study the effect of regularization terms to overcome this issue. For example, seismic data are assumed to have low-rank characteristics in transformed domains, so exploring sparse inversion in the latent space might be possible. Alternatively, once the optimal latent code is found, we could refine the solution by fine-tuning the decoder weights and solving for a new latent variable in an unsupervised fashion using the observed data. Another limitation arising from computational constraints is the adopted patching strategy. In this method, once each patch has been individually reconstructed, they are reassembled, applying suitable weights to the areas where the patches overlap, to create the final result. Although this approach provides a partial solution to reduce the computational burden, it is not optimal and demands the setting of additional hyper-parameters, such as the number of patches and the degree of overlap.

Benefiting from the non-linear representation of the CAE, we can also incorporate more variable as the inputs of the encoding layer For example, we could incorporate the migra-

tion velocity model and/or the stacked migrated section. Although such an approach would not strictly approximate the inverse Hessian anymore, it could provide meaningful information for broadband reflectivity inversion. Finally, the decoder's non-convexity breaks the optimization's theoretical guarantees that linear reflectivity inversion such as conventional LSRTM offers. Finding robust strategies to avoid unsatisfactory results related to defective local minimum is still necessary.

3.3 Learned data-domain preconditioning

Similar to the computation of matching filters in the model space, Khalil et al. (2016) propose a data-domain preconditioning technique that only requires one demigration and two migration operations. Instead of working directly in the image space, this method computes non-stationary filters by matching demigrated and observed data, effectively approximating an inverse to the data-domain Hessian matrix. Liu and Peter (2018) and Liu et al. (2019) explore later variations of this idea by applying Wiener and Gabor deconvolution approaches, respectively.

Building upon the data-domain preconditioning method and the recent advent of deep learning algorithms, this section presents a single-step imaging framework that utilizes a CNN to compute non-linear, non-stationary matching filters in the data domain. Specifically, we employ the symmetrical U-net architecture (Ronneberger et al., 2015b) as a data-domain preconditioner, training it with overlapping patches of observed and demigrated shot gather data. Following the training process, the filtering procedure can be effortlessly applied to the observations with minimal computational burden compared to seismic modelling and migration. Subsequently, the filtered data is remigrated to return to the image domain, resulting in an RTM outcome that exhibits reduced artifacts. Moreover, by applying preconditioning in the data domain, we remove the requirement for a representative training dataset containing model-space reflectivity labels.

3.3.1 Theory

Assuming an invertible Hessian matrix, $\mathbf{H}$, the closed-form solution of LSRTM is given by

$$\begin{aligned} \mathbf{m}_{ls} &= [\mathbf{L}^T\mathbf{L}]^{-1}\mathbf{L}^T\mathbf{d}_{\text{obs}} \\ &= \mathbf{H}^{-1}\mathbf{m}_{\text{mig}}, \end{aligned} \tag{3.27}$$

where $\mathbf{m}_{ls}$ denotes the least-squares solution, and $\mathbf{m}_{\text{mig}}$ represents the RTM image as the result of applying the adjoint of the Jacobian operator, $\mathbf{L}^T$, to the vectorized observed data,

$\mathbf{d}_{\text{obs}}$.

Equation 3.27 suggests that the pseudo-inverse yields an equivalent solution to the normal equations in cases of overdetermined inverse problems. Nevertheless, the seismic inversion system described in equation 3.1 exhibits both overdetermination, stemming from the abundance of seismic measurements, and underdetermination, arising from the limited receiver coverage of the subsurface and the band-limited nature of the seismic source. The minimum norm solution gives one particular solution in the context of underdetermined inverse problems

$$\mathbf{m}_{ls} = \mathbf{L}^T[\mathbf{L}\mathbf{L}^T]^{-1}\mathbf{d}_{\text{obs}}, \tag{3.28}$$

where we now assume that the matrix $[\mathbf{L}\mathbf{L}^T]$ is invertible. Thus, an alternative to least-squares imaging can be obtained by defining a data-domain preconditioning operator $\mathbf{P} \approx [\mathbf{L}\mathbf{L}^T]^{-1}$, such that

$$\mathbf{m}_{ls} \approx \mathbf{L}^T\mathbf{P}\mathbf{d}_{\text{obs}}. \tag{3.29}$$

A practical way of computing $\mathbf{P}$ is by exploiting the correlation between migration and demigration,

$$\begin{aligned} \mathbf{d}_{\text{mig}} &= \mathbf{L}\mathbf{m}_{\text{mig}} \\ \mathbf{d}_{\text{mig}} &= \mathbf{L}\mathbf{L}^T\mathbf{d}_{\text{obs}} \\ [\mathbf{L}\mathbf{L}^T]^{-1}\mathbf{d}_{\text{mig}} &= \mathbf{d}_{\text{obs}} \\ \mathbf{P}\mathbf{d}_{\text{mig}} &\approx \mathbf{d}_{\text{obs}}, \end{aligned} \tag{3.30}$$

where $\mathbf{d}_{\text{mig}}$ indicates demigrated data. Since $\mathbf{d}_{\text{mig}}$ and $\mathbf{d}_{\text{obs}}$ are known, the elements of $\mathbf{P}$ can be computed, for example, with a non-stationary matching filtering technique minimizing

$$E(\mathbf{P}) = ||\mathbf{P}\mathbf{d}_{\text{mig}} - \mathbf{d}_{\text{obs}}||_2^2 + \lambda R(\mathbf{P}), \tag{3.31}$$

where $\mathbf{P}$ is defined as a multidimensional convolutional operator along all spatial axes, and R indicates a suitable regularization term.

CNN-based data-domain preconditioning

We propose parameterizing the preconditioning operator with a CNN, $\mathbf{P}_{\theta_d}$, with θ_d representing the trained CNN weights acting on the data domain. Through this parameterization, we aim to leverage the non-linear representation capabilities of CNNs to enhance deblurring and achieve superior amplitude balancing in RTM images in complex geological scenarios with poor illumination, like subsalt regions and steeper reflectors. This choice is motivated

by the growing evidence suggesting that CNNs can outperform traditional fixed, linear bases like Fourier and Curvelet transform, which are currently considered state-of-the-art. It also expands on the methodology presented in the previous section, as a data-space analogous of the model-domain deep-learning-based preconditioning for iterative LSRTM.

Thus, we implement a neural network architecture that focuses on extracting complex features from the high-dimensional data space to learn a non-linear surrogate of the inverse matrix $[\mathbf{L}\mathbf{L}^T]^{-1}$ without the need for intermediate processing steps or diagonal approximations (Liu and Peter, 2018).

Akin to the previous section, after an initial migration/demigration sequence, the CNN-based preconditioner is trained by minimizing the loss

$$E(\theta_d) = \frac{1}{N_s}\sum_{i=1}^{N_s} ||\mathbf{P}_{\theta_d}(\mathbf{d}^i_{\text{mig}}) - \mathbf{d}^i_{\text{obs}}||_2^2 + \lambda||\theta_d||_2^2, \tag{3.32}$$

over a training dataset of N_s paired samples of observed and demigrated data $\{(\mathbf{d}^i_{\text{mig}}, \mathbf{d}^i_{\text{obs}})\}_{i=1}^{N_s}$. In this approach, we treat the observed data samples as the labels and the initial demigrated data as the inputs to the network. Finally, we get an enhanced RTM image by first applying the trained network on the observed reflections and then migrating this CNN-preconditioned data,

$$\begin{aligned} \mathbf{m}_{CNN} &= \mathbf{L}^T\mathbf{P}_{\theta_d}(\mathbf{d}_{\text{obs}}) \\ &\approx \mathbf{L}^T[\mathbf{L}\mathbf{L}^T]^{-1}\mathbf{d} \\ &\approx \mathbf{L}^T\mathbf{L}^{T^{-1}}\mathbf{L}^{-1}\mathbf{L}\mathbf{m} \\ &\approx \mathbf{m}_{ls}. \end{aligned} \tag{3.33}$$

Similar to the previously mentioned data-domain preconditioning methods derived from classic signal processing techniques, the overall cost of our approach amounts to approximately two RTM operations and one demigration. Compared to the cost of migration/demigration of the whole data, the CNN training and inference overhead is negligible.

3.3.2 Neural network architecture and training details

In this study, we chose the U-net architecture for its ability to preserve high-level details. Previously, the U-net has been generally applied as a segmentation precondition in imaging problems (Huang and Huang, 2021), as a nullspace projection surrogate in post-stack reflectivity inversion (Torres and Sacchi, 2023), and for the detection of geological faults (Wu et al., 2019).

As can be noticed from Figure 3.10, its architecture consists of an encoding path and a decoding path, where both input and output are images. This setup captures local and global features, enabling the model to learn complex mappings between input and output. The U-Net incorporates several operations. Each step in the contracting path involves two 3×3 convolutional layers followed by the rectified linear unit (ReLU) activation function. Downscaling is achieved through 2×2 max pooling with a stride of 2. Conversely, the expansive path (right side) utilizes 2×2 upsampling with the same stride, accompanied by two convolutional layers that reduce the number of feature channels. The U-Net also includes skip connections that link the left and right paths to preserve spatial information lost during max pooling. These connections help preserve high-resolution information and facilitate the recovery of fine details in the output.

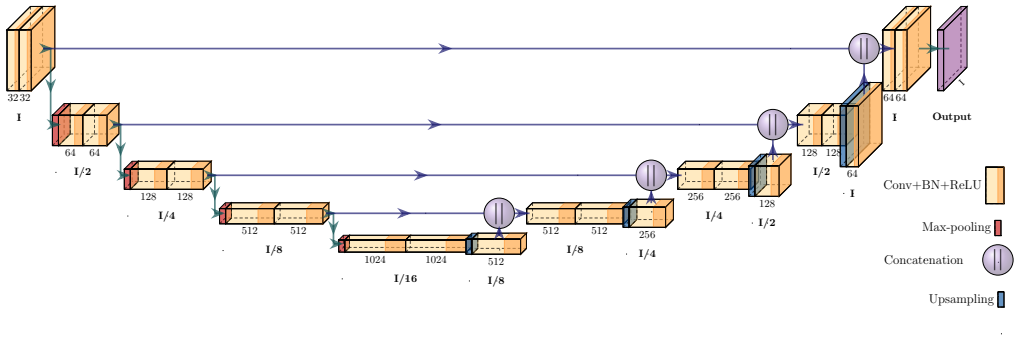

Figure 3.10: The U-net architecture. The number at the bottom of each convolutional layer indicates the number of channels. **I** refers to the original image size.

To train the U-net, we utilize overlapping patches of demigrated and observed data, randomly shuffled before being fed into the network. Ten realizations from the paired training samples are shown in Figure 3.11 and 3.12. When using RTM images for demigration, the energy magnitude of demigrated shot gathers exhibits significant variation compared to the observed data. This magnitude can range from -10^n to 10^n depending on the number of stacking shots and the source energy. We employ the standardization technique explained in the previous section to address this data variance and accelerate the convergence of the training loss function. This technique ensures that the training data is transformed to have a mean of zero and a variance of one. Furthermore, we introduce a random horizontal flipping operation to enhance the training process and increase the diversity of training samples.

As the trained U-net will be applied to the observed data in the final step of our workflow (equation 3.33), we divide the original dataset into 90% training samples and the remaining 10% as validation samples. This division allows for the training of the U-net using most of the data while retaining a separate portion for evaluating the model's performance.

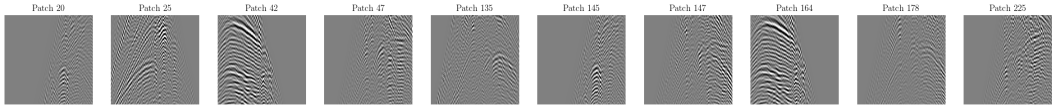

Figure 3.11: 10 random realizations from the patched demigrated shot gathers used as input training data.

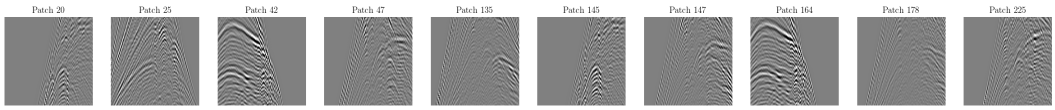

Figure 3.12: 10 random realizations from the patched observed shot gathers used as output training labels.

3.3.3 Numerical examples

We test our approach on the 2D acoustic Marmousi dataset (Versteeg, 1994). The recorded data are derived from the true velocity model depicted in Figure 3.13. The migration velocity presented in Figure 3.14 is obtained by smoothing the true velocity using a 2D Gaussian smoother. Figure 3.19 shows the true reflectivity model. In this benchmark scenario, 103 shots are evenly initiated at intervals of 50 m, employing a 30 Hz Ricker wavelet as the source function. The observed data are obtained using a fixed-spreading acquisition setup with 512 receivers at 10 m intervals for each shot-gather. The recording duration covers 3 seconds, with direct waves being eliminated.

Equation 3.32 is minimized using Adam with 50 epochs, weight decay damping term $\lambda = 10^{-4}$ and a learning rate of 10^{-3}.

We initially present the RTM result for a single shot, migrated at x=1km, as depicted in Figure 3.15. Figure 3.16 illustrates the outcome of migrating the observations after being filtered by the CNN. Upon comparing these two images, we observe that the preconditioned result exhibits improved illumination at deeper reflectors and a noticeable reduction in low-frequency backscattering noise. Subsequently, we showcase the complete RTM section by stacking all individually migrated shot gathers, both without data preconditioning (Figure 3.17) and with data preconditioning (Figure 3.18) using the CNN. Notably, the amplitude balance is substantially enhanced, and there is a distinct reduction in migration artifacts around the top-right steeper reflectors (highlighted by the top red arrow), as well as a greater focusing power on the deeper reflectors below the bottom high-velocity intrusions (indicated by the bottom red arrow). Additionally, our method effectively mitigates the sparse migration footprints present in the shallower part of the model.

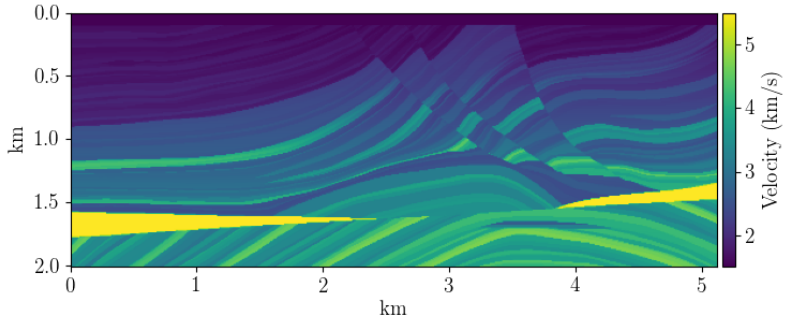

Figure 3.13: Marmousi velocity model.

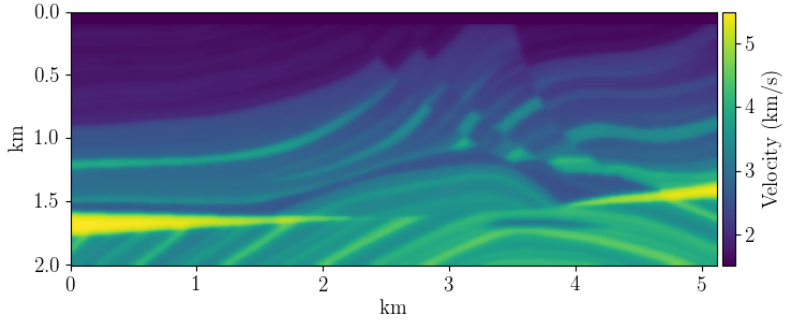

Figure 3.14: Migration velocity model.

3.3.4 Discussion and conclusions

Identifying optimal preconditioner operators can be a non-trivial task. It requires expertise and extensive experimentation. This work proposed an alternative data-domain preconditioning based on deep-learning filtering. We showed that our approach could deliver enhanced migrated sections with improved focusing capability and amplitude balance around under-illuminated regions. The training stage does not need further pre-processing and requires minimal user interaction. It is also relatively cheap: regarding wave equation operations, the method requires only a single extra migration compared to one LSRTM iteration.

As reported in existing literature (Nichols, 1997), the adjoint operator, which is the final operator employed in equation 3.33, cannot generate model components within the null space. Therefore, it should be acknowledged that the proposed data-domain technique might not be particularly successful in aiding the retrieval of a reflectivity model with higher wavenumber content due to the assumptions it is based on. In terms of iterative inversion, future work will focus on combining the proposed data-domain (left) preconditioner with image-space (right) preconditioning to accelerate the convergence of iterative LSRTM towards high-resolution

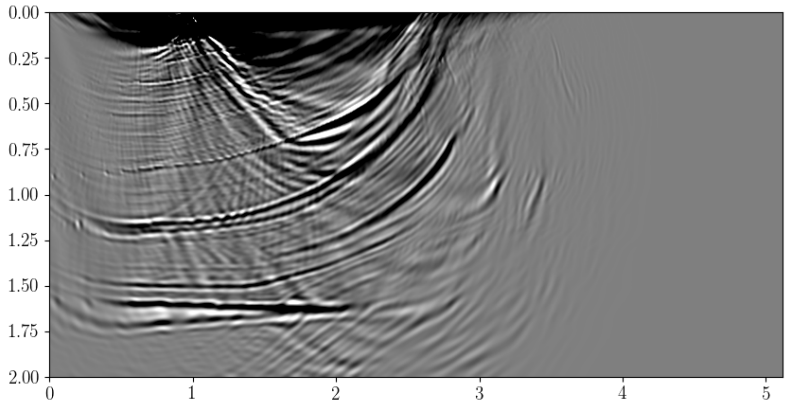

Figure 3.15: Single-shot RTM.

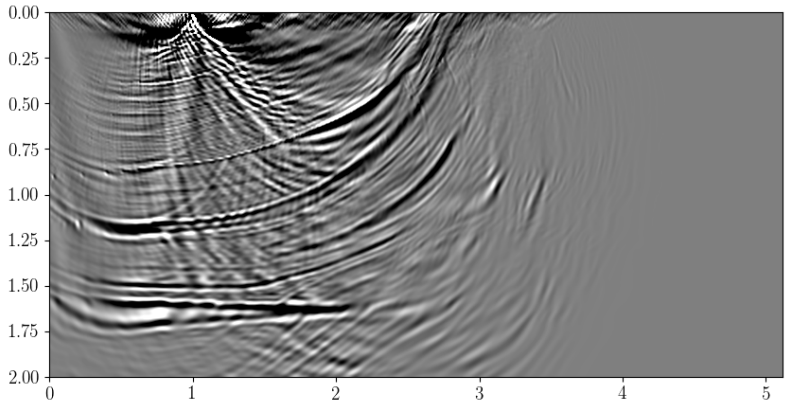

Figure 3.16: Single-shot RTM with CNN-based preconditioning.

reflectivity models.

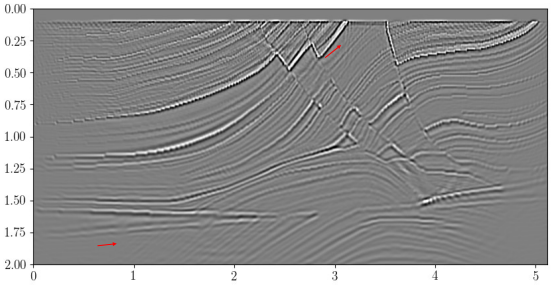

Figure 3.17: Stacked RTM section.

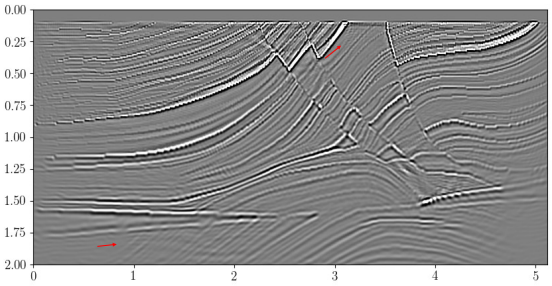

Figure 3.18: Stacked RTM section with CNN-based preconditioning.

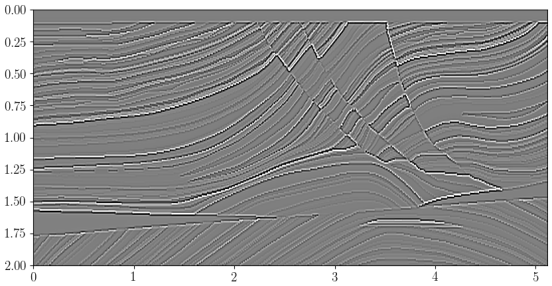

Figure 3.19: True reflectivity.

CHAPTER 4

Sparse vector reflectivity inversion with full-wavefield LSRTM

4.1 Introduction

Least-squares reverse time migration (LSRTM) offers distinct advantages in subsurface imaging, such as compensating for wavefield propagation effects and limited resolution typically seen with conventional migration. This makes it a preferred method to update and improve the reflectivity model in complex geological settings. However, it faces significant drawbacks, primarily due to its dependence on the first-order scattering approximation (Born modeling) for generating predicted data during the forward propagation of the linearized inversion. Thus, it is unsuited to model large time shifts between the background wavefield and the scattered wavefield. This limitation confines LSRTM to deal with only primary waves with near- to mid-reflection angles (Korsmo et al., 2022), implying the need for the challenging elimination of other wave modes in the observed data, despite their potential to contain significant information. For example, LSRTM considers prismatic waves as noise, thus failing to correctly depict steeply dipping structures. Therefore, a pre-processing step is essential in conventional LSRTM to prevent crosstalk in the imaging results, especially when dealing with complex seismic data that includes multiple reflections (Wong et al., 2015). Furthermore, due to the ill-posed and ill-conditioned nature (i.e., solutions are not unique and/or unstable), LSRTM results often contain migration artifacts and unwanted sidelobes around reflectors. These issues primarily arise due to the limited acquisition aperture, the restricted bandwidth of noisy seismic data, and inaccuracies in the modeling kernel (Zeng et al., 2014). On top of that, many different factors, such as sub-optimal acquisition design or rapidly changing velocity profiles, can play a role in creating shadow zones in

migrated images or, broadly speaking, in images produced through least-squares migration. These shadow regions, associated with small and null singular values of the Hessian, reside in the null space of the forward operator and exacerbate the problem of non-uniqueness in the solutions.

To alleviate the side effects of conventional LSRTM, we perform a non-linear least-squares inversion combining the full-wavefield vector reflectivity modeling engine (Whitmore et al., 2020) with a sparse constraint via the hyperbolic penalty function (HPF) (Claerbout, 2014). The full-wavefield modeling parameterizes the variable density acoustic wave equation regarding velocity and vector reflectivity, eliminating the need to construct a density model. It generates seismic data with all waveforms, including multiples and transmission events, directly from the reflectivity without assuming they are linearly related. This enables reflectivity inversion with a more comprehensive and accurate match between observed and calculated data while bypassing the first-order Born approximation and the data pre-processing step needed in classic LSRTM. In addition, imposing the sparsity constraint as model-styling regularization helps to stabilize the solution and mitigate artifacts, thus enhancing the vertical resolution of the estimated seismic images.

Other non-linear LSRTM schemes can also refine the velocity model during the iterative migration process (Yao and Jakubowicz, 2012; Korsmo et al., 2022). Nonetheless, similar to Davydenko and Verschuur (2017) and Wu et al. (2024), we assume that a background velocity has been previously estimated and is not updated through iterations, reducing the non-linearity of the problem. In other words, we solely focus on inverting the vector reflectivity model. When contrasted with least-squares inversion lacking regularization, a 2D numerical experiment indicates that applying the HPF combined with the full-wavefield vector reflectivity engine promotes sparsity in the images while displaying fewer artifacts and more clearly delineated reflectors.

The outline of this chapter is as follows. First, we review the forward operator of full-wavefield vector reflectivity modeling in a re-parameterized variable-density acoustic earth. We also derive its adjoint wave equation and present the corresponding stencils for forward and backward simulations in the time domain. These pair of forward and adjoint wavefields characterize what we have termed as the full-wavefield LSRTM framework, which is essentially a local optimization problem. Thus, we elaborate on its misfit function and gradient formulation. Then, we add sparsity constraints to the inverse problem of retrieving the horizontal and vertical vector reflectivity components from the recorded seismic data at the surface. Additionally, we detail the process of deriving a relative density model from the obtained vector reflectivity, which is a secondary result of our analysis. Finally, we demonstrate the effectiveness of the inversion with and without sparsity constraints on the Marmousi2 model (Martin et al., 2006b).

4.2 Full-wavefield LSRTM

4.2.1 Forward modeling

In its continuous form, the 2D full-wavefield vector reflectivity acoustic wave equation used in this study as the forward modeling engine is given by (Whitmore et al., 2020)

$$\left[\frac{1}{v_p(x,z)^2}\frac{\partial^2}{\partial t^2} - \left(\nabla^2 + \frac{1}{v_p(x,z)}\nabla v_p(x,z)\cdot\nabla - 2\vec{r}(x,z)\cdot\nabla\right)\right] u_s(x,z,t) = f_s(t), \quad (4.1)$$

where

$$\vec{r} = \begin{bmatrix} r_x \\ r_z \end{bmatrix} = \frac{1}{2}\frac{\nabla z}{z} = \frac{1}{2}\begin{bmatrix} \frac{\partial}{\partial x}\ln(\mathrm{z}) \\ \frac{\partial}{\partial z}\ln(\mathrm{z}) \end{bmatrix} \quad (4.2)$$

denotes the vector reflectivity with horizontal and vertical components, r_x and r_z, respectively, $\mathrm{z} = \rho v_p$ represents the acoustic impedance, v_p is the velocity model, ρ the density, f_s is the scaled seismic source at the shot coordinates (x_s, z_s), and $u_s(x,z,t)$ is the pressure wavefield associated to that source (hence the subscript), dependent on time and space variables. It is assumed that the pressure wavefield has zero initial conditions, i.e., the wavefield does not have any energy before zero time:

$$u_s(x,z,0) = \frac{\partial u_s(x,z,0)}{\partial t} = 0. \quad (4.3)$$

Whitmore et al. (2020) show that equation 4.1 is simply a re-parameterization of the acoustic wave equation with variable density (equation 1.1) that avoids the explicit dependence on the density model by defining the vector reflectivity as the normalized rate of impedance change in each spatial direction. This reconfiguration of the wave equation has powerful applications to multiparameter waveform inversion (Yang et al., 2021), and may also be useful for adaptive multiple removal. Also, an alternative application of this modeling engine include Fletcher et al. (2023), suggesting a method to diminish reflections from a specific boundary by averaging the wavefields from two independent forward simulations.

To simulate the forward modeling of the wavefield on the subsurface, we incorporate an extra dampening term, $\eta(x,z)\frac{\partial u(t,x,z)}{\partial t}$, to mimic an infinite domain (Sochacki et al., 1987). The factor η is set to 0 within the boundaries of the physical domain and progressively increases from the interior to the exterior within the damping layer. Depending on the width of the damping layer and the frequency band, η may exhibit either a linear or exponential increase. At the boundary of the artificial domain, we assume the solution fulfills the radiation boundary conditions

$$u_s(x,z,t)\big|_{(x,z)\to\infty} \to 0, \quad \nabla u_s(x,z,t)\big|_{(x,z)\to\infty} \to 0, \quad (4.4)$$

indicating that the wavefield is progressively damped as we move away from the source towards the computational domain's edge. Omitting variable dependence on space dimensions to simplify the notation, the expression for the vector reflectivity modeling with the absorbing boundary term is

$$\left[\frac{1}{v_p^2}\frac{\partial^2}{\partial t^2} - \left(\nabla^2 + \frac{\nabla v_p}{v_p}\cdot\nabla - 2\vec{\mathbf{r}}\cdot\nabla\right) + \eta\frac{\partial}{\partial t}\right] u_s(t) = f_s(t). \tag{4.5}$$

Using a second-order central-difference approximation for the time derivative, we can construct the stencil necessary to compute the full solution of the acoustic wavefield over time. Within the boundaries of the physical domain (i.e., ignoring the absorbing boundary region), the stencil is given by

$$\begin{aligned} u_s(t+dt) = {} & dt^2 v_p\left(\frac{\partial u_s(t)}{\partial x}\frac{\partial v_p}{\partial x} + \frac{\partial u_s(t)}{\partial y}\frac{\partial v_p}{\partial y}\right) - dt^2 v_p^2\left(2r_x\frac{\partial u_s(t)}{\partial x} + 2r_y\frac{\partial u_s(t)}{\partial y} - \nabla^2 u_s(t)\right) \\ & + 2u_s(t) - u_s(t-dt) + dt^2 v_p^2 f_s(t), \end{aligned} \tag{4.6}$$

where dt represents the time-stepping interval, and $\frac{\partial}{\partial x}$, $\frac{\partial}{\partial z}$, and ∇^2 denote the first-order spatial derivatives in horizontal and vertical dimensions and the Laplacian operator, respectively, which can be computed with high-order finite difference approximations. Discretizing and solving the vector reflectivity wave equation by applying this explicit Euler scheme for all time steps results in the linear system,

$$\mathbf{F}(\vec{\mathbf{r}})\mathbf{u}_s = \mathbf{f}_s, \tag{4.7}$$

where $\mathbf{u}_s \in \mathbb{R}^{N_u}$ signifies the vector containing the discrete wavefield's solution, $\mathbf{f}_s \in \mathbb{R}^{N_u}$ denotes the source term, and $\mathbf{F}(\vec{\mathbf{r}}) \in \mathbb{R}^{N_u \times N_u}$ encapsulates the matrix form of the discrete wave equation, which non-linearly depends on the known vector reflectivity parameter $\vec{\mathbf{r}} \in \mathbb{R}^{2\times N_n}$. Here, $N_u = N_n \times N_t$, where $N_n = N_x \times N_z$, and N_i designates the number of points along the i spatial axis of the numerical grid. Likewise, N_t represents the number of time samples. The time-marching structure of the stencil indicates that the matrix $\mathbf{F}(\vec{\mathbf{r}})$ is lower triangular, depending on the assumed time and spatial boundary conditions. In practice, however, we never explicitly form or invert this matrix but rather perform forward substitution which is equivalent to solving the problem sequentially, row by row, starting from the top. Thus, the forward stencil is equivalent to implement the matrix-vector product describing the propagation of the acoustic source

$$\mathbf{u}_s = \mathbf{F}(\vec{\mathbf{r}})^{-1}\mathbf{f}_s, \tag{4.8}$$

which represents a linear operator with respect to the source term. Finally, the vector symbolizing the projection of the propagated wavefield onto the receiver coordinates is given by

$$\mathbf{d} = \mathbf{P}_d \mathbf{u}_s, \tag{4.9}$$

where $\mathbf{d} \in \mathbb{R}^{N_m}$, and $\mathbf{P}_d \in \mathbb{R}^{N_m \times N_u}$, with $N_m = N_r \times N_t$ and N_r denoting the number of receivers, is the sampling-to-receivers operator, a discretization of the delta function. We assume a stationary-recording acquisition ($\mathbf{P}_d$ does not depend on s) for the sake of compact notation but the method can be applied to other acquisition geometries.

Figure 4.1 compares the snapshots of the forward-propagated wavefield at time=2.5 seconds between the full-wavefield vector reflectivity modeling and Born modeling on the BP synthetic model. Similarly, Figure 4.2 compares a shot gather obtained via full-wavefield vector reflectivity modeling with a shot gather computed using Born modeling. As can be noticed in both images, the full-wavefield vector reflectivity resolves a broader range of seismic events, including refractions, diving waves, and reflections beyond the Born approximation. We also show the data recorded using the original variable density acoustic wave equation for comparison.

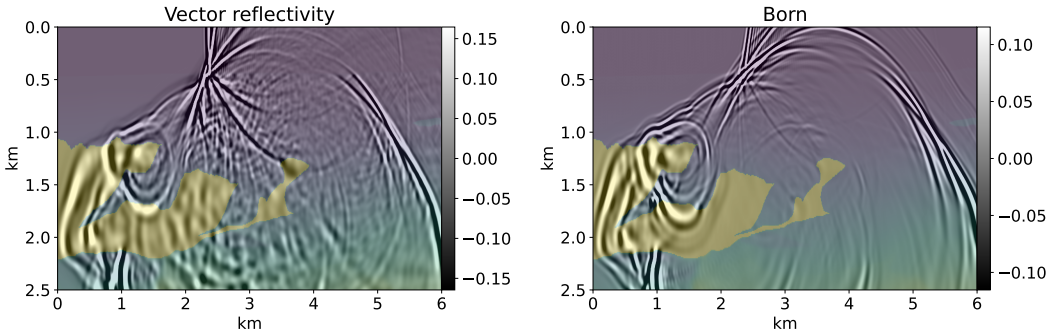

Figure 4.1: Forward propagation experiment on the BP synthetic model. The seismic source is located at $x = 6$ km and $z = 0.02$ km. Left: Acoustic wavefield at $t = 2.5$ seconds modeled with the full-wavefield vector reflectivity engine. Right: Born modeling for the same time snapshot. The velocity model containing a salt body is shown on the background for reference.

4.2.2 Adjoint modeling and gradient

In this section we provide the derivation of the FWLSRTM gradient with respect to the vector reflectivity parameter using the time-domain adjoint-state method following Plessix (2006b) and Fichtner (2010). Within the framework of optimization problems constrained by large systems of partial differential equations, this mathematical tool allows us to calculate

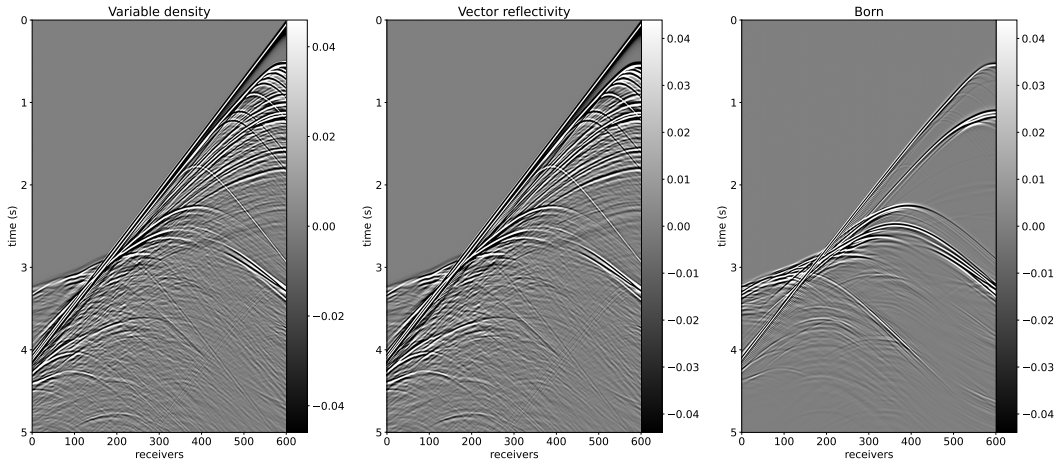

Figure 4.2: Forward modeling comparison on the BP synthetic model. Left: Recorded traces using the variable density acoustic wave equation. Middle: Recorded traces using the full-wavefield acoustic vector reflectivity modeling. Right: Born modeling for the same experiment. LSRTM uses the Born approximation focusing on first-order scattering in forward modeling, which limits its use to near-to-mid reflection angles and neglects other wave modes such as diving and prismatic waves.

the gradient in an efficient way. This is particularly pertinent given the impracticality of directly calculating and storing the Jacobian matrix, even with modern computing hardware.

To find a solution to the FWLSRTM problem, I use gradient-descent-based algorithms such as the L-BFGS method (Liu and Nocedal, 1989), which derives quasi-Newton updates as a function of the misfit value and its gradient. The computation of the latter also entails the derivation and implementation of the adjoint vector reflectivity wave equation. Consequently, we introduce a robust and precise framework to resolve the optimization problem. This approach stands in contrast to prior methodologies that relied on the heuristic impedance kernel combined with the inverse scattering imaging condition to estimate vector reflectivity updates (Yang et al., 2021).

To begin, we define the FWLSRTM objective function for a single shot data (thereby dropping the source-dependent index from my previous notation), $E(\vec{r}) \in \mathbb{R}$, by assigning the squared l_2-norm to a residual function, $e(\vec{r})$, which quantifies the error between the predicted and observed data, $d(\psi_g; \vec{r})$ and $d_{\text{obs}}(\psi_g)$, respectively,

$$E(\vec{r}) = \frac{1}{2} \int_{\Psi} e(\vec{r})^2 \, d\psi_g = \frac{1}{2} \langle e(\vec{r}), e(\vec{r}) \rangle, \tag{4.10}$$

where the residual function

$$e(\vec{r}) = d(\psi_g; \vec{r}) - d_{\text{obs}}(\psi_g) = \int_{\Psi} \left[u(\psi; \vec{r}) - u_{\text{obs}}(\psi)\right] \delta(\psi - \psi_g)\, d\psi, \tag{4.11}$$

is expressed as an integral over the entire physical domain, $\Psi = (x, z, t)$. The predicted and observed pressure fields, denoted by u and u_{obs}, propagate across the entire space-time domain. However, they are sampled exclusively at the spatial and temporal coordinates of the receivers, expressed as ψ_g, by the Dirac delta function δ. It is important to note that, for the sake of notational brevity, this work is conducted within a 2D (x and z spatial coordinates) plus time domain. However, the methodology described herein can be readily adapted to a 3D plus time domain with just an additional integration over the y-dimension.

Using the chain rule for differentiation, the gradient of the objective function can be expressed as

$$\nabla_{\vec{r}} E = \langle \nabla_{\vec{r}} e, e \rangle = \langle \nabla_u e \nabla_{\vec{r}} u, e \rangle. \tag{4.12}$$

The inability to compute the gradient arises from the term $\nabla_{\vec{r}} u$ containing the Fréchet derivatives, due to the vast size of the model parameter space and the state variable u. To circumvent the direct computation of this term, we can first express the forward operator of the vector reflectivity wave equation as

$$F(u(\vec{r}), \vec{r}) = \left[\frac{1}{v_p^2}\frac{\partial^2}{\partial t^2} - \nabla^2 - \frac{\nabla v_p}{v_p} \cdot \nabla + 2(\vec{r} \cdot \nabla)\right] u - f = 0, \tag{4.13}$$

and its derivatives

$$\begin{aligned} \nabla_u F \nabla_{\vec{r}} u + \nabla_{\vec{r}} F &= 0, \\ \left[\frac{1}{v_p^2}\frac{\partial^2}{\partial t^2} - \nabla^2 - \frac{\nabla v_p}{v_p} \cdot \nabla + 2(\vec{r} \cdot \nabla)\right] \nabla_{\vec{r}} u + 2\nabla u &= 0. \end{aligned} \tag{4.14}$$

We then introduce the adjoint state variable, λ, also known as the Lagrange multiplier, compute its product with Equation 4.14,

$$\langle \lambda, \nabla_u F \nabla_{\vec{r}} u + \nabla_{\vec{r}} F \rangle = \langle \lambda, \nabla_u F \nabla_{\vec{r}} u \rangle + \langle \lambda, \nabla_{\vec{r}} F \rangle = 0, \tag{4.15}$$

and add this null expression to the gradient equation as follows:

$$\nabla_{\vec{r}} E = \langle \nabla_u e \nabla_{\vec{r}} u, e \rangle + \langle \lambda, \nabla_u F \nabla_{\vec{r}} u \rangle + \langle \lambda, \nabla_{\vec{r}} F \rangle. \tag{4.16}$$

By implementing the definition of adjoint operators,

$$\langle Aw_2, w_1 \rangle = \langle w_2, A^* w_1 \rangle \tag{4.17}$$

(in which w_1 and w_2 are test functions from the domains of the arbitraty operators A and A^*, respectively), and establishing the adjoint-state variable in a convenient manner, we can isolate and eliminate $\nabla_{\vec{r}} u$ from the first two terms on the right-hand side of Equation 4.16. This gives

$$\begin{aligned} \nabla_{\vec{r}} E &= \langle \nabla_{\vec{r}} u, \nabla_u e^* e \rangle + \langle \nabla_u F^* \lambda, \nabla_{\vec{r}} u \rangle + \langle \lambda, \nabla_{\vec{r}} F \rangle. \\ &= \langle \nabla_{\vec{r}} u, \nabla_u e^* e + \nabla_u F^* \lambda \rangle + \langle \lambda, \nabla_{\vec{r}} F \rangle \\ &= \langle \lambda, \nabla_{\vec{r}} F \rangle, \end{aligned} \tag{4.18}$$

when λ satisfies the adjoint-state equation

$$\nabla_u F^* \lambda = -\nabla_u e^* e. \tag{4.19}$$

The last line in Equation 4.18 tells us that the gradient of the FWLSRTM objective function for a single shot experiment can be efficiently obtained by solving

$$\nabla_{\vec{r}} E = \begin{bmatrix} \nabla_{r_x} E \\ \nabla_{r_z} E \end{bmatrix} = 2 \int_0^T \lambda \nabla u dt, \tag{4.20}$$

which represents the accumulation of a vector field, ∇u, weighted by a scalar field, λ, over the specified time period. In this scenario, the FWLSRTM gradient provided by the adjoint-state method requires solving only two PDEs, as opposed to solving a set of PDEs that matches the total number of elements in the model parameter, which would be required to construct the Jacobian.

The last ingredient in our recipe consists of finding an explicit expression for the adjoint-state variable. Following Equation 4.19, we first derive the term for the right-hand side adjoint source, $-\nabla_u e^* e$. Per the definition of the residual function (Equation 4.11), we notice that

$$\nabla_u e = \int_\Psi \delta(\psi - \psi_g) d\psi, \tag{4.21}$$

and

$$\begin{aligned}\langle \nabla_u e w_2, w_1 \rangle &= \int_\Psi \Big(\int_\Psi w_2(\psi)\delta(\psi - \psi_g)d\psi \Big) w_1(\psi_g)d\psi \\ &= \int_\Psi w_2(\psi_g)\Big(\int_\Psi w_1(\psi)\delta(\psi - \psi_g)d\psi \Big) d\psi \\ &= \int_\Psi w_2(\psi_g) w_1(\psi_g) d\psi \\ &= \langle w_2, \nabla_u e w_1 \rangle,\end{aligned} \tag{4.22}$$

which indicates that the operator $\nabla_u e$ is self-adjoint, i.e., $\nabla_u e = \nabla_u e^*$. Therefore, the adjoint source term is equal to the negative of the residuals.

Similarly, to obtain the term $\nabla_u F^*$, we resort to the adjoint definition given by Equation 4.17. Namely,

$$\begin{aligned}\langle w_2, \nabla_u F w_1 \rangle &= \int_0^T \int_{z_0}^{z_1} \int_{x_0}^{x_1} w_2 \left[\frac{1}{v_p^2}\frac{\partial^2}{\partial t^2} - \nabla^2 - \frac{\nabla v_p}{v_p}\cdot\nabla + 2(\vec{r}\cdot\nabla) \right] w_1 \, dxdzdt \\ &= \int_0^T \int_{z_0}^{z_1} \int_{x_0}^{x_1} \frac{w_2}{v_p^2}\frac{\partial^2 w_1}{\partial t^2} - w_2\nabla^2 w_1 - w_2\frac{\nabla v_p}{v_p}\cdot\nabla w_1 + 2w_2(\vec{r}\cdot\nabla w_1)\, dxdzdt.\end{aligned} \tag{4.23}$$

To deal with the first term on the right-hand side containing temporal derivatives, we integrate by parts twice and impose zero initial conditions to w_1 such that $w_1(x,z,0) = \frac{\partial w_1(x,z,0)}{\partial t} = 0$. By applying integration by parts to this terms, we move the differentiation from one factor in the product to the other. Moreover, we prescribe zero final conditions on w_2, i.e., $w_2(x,z,T) = \frac{\partial w_2(x,z,T)}{\partial t} = 0$. Thus, the expression can be reduced to

$$\begin{aligned}\int_0^T \frac{w_2}{v_p^2}\frac{\partial^2 w_1}{\partial t^2}dt &= \frac{1}{v_p^2}\left\{ \left[w_2\frac{\partial w_1}{\partial t} \right]_0^T - \int_0^T \frac{\partial w_2}{\partial t}\frac{\partial w_1}{\partial t}dt \right\} \\ &= \frac{1}{v_p^2}\left\{ \left[w_2\frac{\partial w_1}{\partial t} \right]_0^T - \left[\frac{\partial w_2}{\partial t} w_1 \right]_0^T + \int_0^T \frac{\partial^2 w_2}{\partial t^2} w_1 dt \right\}, \\ &= \int_0^T \frac{w_1}{v_p^2}\frac{\partial^2 w_2}{\partial t^2}dt.\end{aligned} \tag{4.24}$$

The remaining terms contain spatial derivatives and can be treated in a similar way assuming that both w_1 and w_2 vanish at the boundaries x_0, x_1, z_0, and z_1(homogeneous boundary

conditions). Consequently, Equation 4.23 can be reformulated as

$$\begin{aligned}\langle w_2, \nabla_u F w_1 \rangle &= \int_0^T \int_{z_0}^{z_1} \int_{x_0}^{x_1} w_2 \left[\frac{1}{v_p^2} \frac{\partial^2}{\partial t^2} - \nabla^2 - \frac{\nabla v_p}{v_p} \cdot \nabla + 2(\vec{r} \cdot \nabla) \right] w_1 \, dx dz dt \\ &= \int_0^T \int_{z_0}^{z_1} \int_{x_0}^{x_1} w_1 \left[\frac{1}{v_p^2} \frac{\partial^2}{\partial t^2} - \nabla^2 + \nabla \cdot \frac{\nabla v_p}{v_p} - \nabla \cdot 2\vec{r} \right] w_2 \, dx dz dt \\ &= \langle \nabla_u F^* w_2, w_1 \rangle,\end{aligned} \tag{4.25}$$

which gives us the term $\nabla_u F^*$. We rewrite equation 4.19 to get the final expression for the adjoint-state vector reflectivity wave equation:

$$\begin{aligned}&\left[\frac{1}{v_p^2} \frac{\partial^2}{\partial t^2} - \left(\nabla^2 - \nabla \cdot \frac{\nabla v_p}{v_p} + \nabla \cdot 2\vec{r} \right) \right] \lambda = -e, \\ &\frac{1}{v_p^2} \frac{\partial^2 \lambda}{\partial t^2} - \nabla^2 \lambda + \nabla \cdot \left(\lambda \frac{\nabla v_p}{v_p} \right) - \nabla \cdot \left(2\lambda \vec{r} \right) = -e\end{aligned} \tag{4.26}$$

with final conditions

$$\lambda(t = T) = 0, \quad \frac{\partial \lambda(t = T)}{\partial t} = 0. \tag{4.27}$$

To avoid the computation of nested spatial derivatives at each timestep, we can use the product rule on the third term from the left-hand side in equation 4.26, and rewrite the adjoint equation as

$$\frac{1}{v_p^2} \frac{\partial^2 \lambda}{\partial t^2} - \nabla^2 \lambda + \lambda \nabla^2 \log v_p + \nabla \log v_p \cdot \nabla \lambda - \nabla \cdot \left(2\lambda \vec{r} \right) = -e, \tag{4.28}$$

where the terms $\nabla^2 \log v_p$ and $\nabla \log v_p$ can be precomputed and stored in memory.

Given the final conditions, the corresponding expression for the update of the adjoint wavefield follows a form akin to the stencil presented in Equation 4.6, but solving the time-stepping scheme backwards in time ($t = T \rightarrow 0$):

$$\begin{aligned}\lambda(t - dt) = dt^2 v_p^2 \Bigg(&- \Big(\frac{\partial^2 \log(v_p)}{\partial x^2} + \frac{\partial^2 \log(v_p)}{\partial y^2} \Big) \lambda(t) + 2 \Big(\frac{\partial \big(r_x \lambda(t) \big)}{\partial x} + \frac{\partial \big(r_y \lambda(t) \big)}{\partial y} \Big) \\ &- \Big(\frac{\partial \lambda(t)}{\partial x} \frac{\partial \log(v_p)}{\partial x} + \frac{\partial \lambda(t)}{\partial y} \frac{\partial \log(v_p)}{\partial y} \Big) + \frac{\partial^2 \lambda(t)}{\partial x^2} + \frac{\partial^2 \lambda(t)}{\partial y^2} \Bigg) \\ &+ 2\lambda(t) - \lambda(t + dt) + dt^2 v_p^2 f_{adj}(t)\end{aligned} \tag{4.29}$$

where $f_{adj} = -e$ implies the time-reversed residuals acting as sources. Following the same notation as the previous section, the discretization of Equation 4.29 for all the time steps can be represented by the upper triangular matrix $\mathbf{F}(\vec{\mathbf{r}})^T$. The propagation for the adjoint

wavefield is then given by the solution of the linear system

$$\mathbf{F}(\vec{\mathbf{r}})^T \lambda = \mathbf{f}_{adj}, \tag{4.30}$$

which is computed by simulating wave propagation starting from the last time step using the data residuals as sources to update the previous adjoint wavefield, $\lambda(t-dt)$, rather than the next wavefield. Similar to the forward problem, this matrix does not need to be explictly built, making it efficient for large-scale problems.

The gradient for each shot (Equation 4.20) is therefore given by the product between two wavefields propagating in opposite time directions: the forward wavefield, u, simulated on a known background velocity, and the retro-propagated wavefield, λ, which is generated by injecting the data residuals from the position of the receivers as the adjoint seismic source, backwards in time. A naive computational approach to estimating the gradient entails storing one of these two wavefields in the entire domain and then accessing its time slices to perform the product with the other wavefield on-the-fly while stacking through the time dimension. However, more advanced wavefield reconstruction schemes can also be implemented to ease the computational load, especially in 3D, where we deal with 4D wavefields (e.g., Symes, 2007; Nguyen and McMechan, 2015).

Once the misfit gradient has been computed, we can obtain vector reflectivity updates via iterative gradient-based minimization

$$\vec{r}^{\,k+1} = \vec{r}^{\,k} + \gamma^k p^k (\nabla E(\vec{r}^{\,k})), \tag{4.31}$$

where the superscript k indicates the current iteration, γ is the step length computed via a line-search algorithm, and p is the gradient search direction, which we compute with a quasi-Newton algorithm to precondition the gradient for an accelerated convergence. The iterative optimization algorithm runs until reaching the maximum number of iterations, which serves as the stopping criterion. Figure 4.3 presents a simplified FWLSRTM workflow while a more complete multisource FWLSRTM algorithm is presented in algorithm 1.

4.2.3 Gradient verification

This section focuses on two specific tests: the dot-product test and the gradient test. As mentioned in the previous section, the solution to the inverse problem relies heavily on the accurate estimation of the misfit gradient, which depends on both the forward and the adjoint wavefields. Therefore, it is important first to confirm the exactness of the derivation of Equation 4.26 and its discretization as the true numerical adjoint wave equation of the full-wavefield vector reflectivity modeling engine. To do this, we employ the dot-product

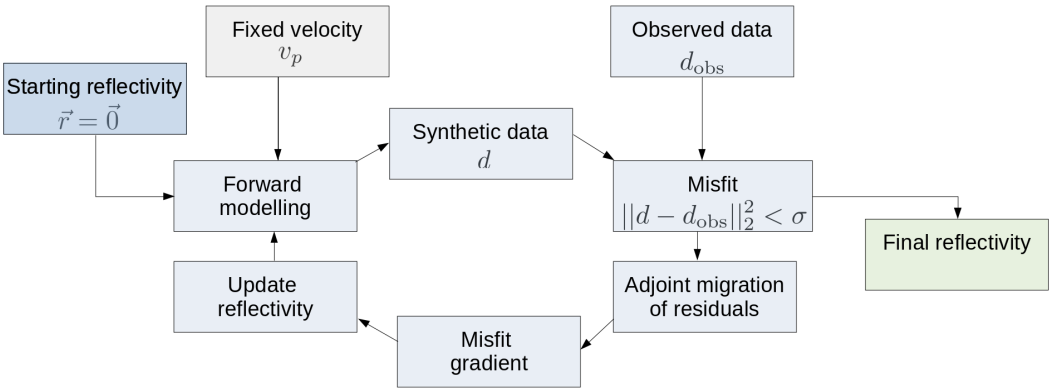

Figure 4.3: Simplified FWLSRTM workflow.

Algorithm 3 Multisource FWLSRTM

Set initial $\vec{r} = 0$
for $k = 1$ to Max. iterations **do**
  for $s = 1$ to N_s **do** ▷ Loop over sources
    Compute $u_s(x, z, t; \vec{r}^k)$ ▷ 1 PDE solve
    Compute residuals $e_s(\vec{r}^k) = d_s(\psi_g; \vec{r}) - d_{\text{obs}_s}(\psi_g)$
    Compute $\lambda_s(x, z, t)$ ▷ 1 PDE solve
    Compute $(\nabla_{\vec{r}} E)_s = 2 \int_T \lambda_s \nabla u_s dt$
  end for
  $\nabla_{\vec{r}} E = \sum_{s=1}^{n_s} (\nabla_{\vec{r}} E)_s$ ▷ Stack gradient over sources
  Compute p^k ▷ Gradient-based techniques
  Compute γ^k ▷ Line search. Requires PDE solves
  $\vec{r}^{k+1} = \vec{r}^k + \gamma^k p^k$
end for

test (Claerbout, 2014) given by the inner product

$$\frac{< F\vec{a}, b > - < \vec{a}, F^T b >}{< F\vec{a}, b >} = 0, \tag{4.32}$$

where $\vec{\mathbf{a}} \in \mathbb{R}^{2 \times N_n}$ and $\mathbf{b} \in \mathbb{R}^{N_m}$ are random vectors spanned in the model and data spaces, respectively. To perform the dot-product test, we use the experiment shown in Figure 4.4, where the long vector $\vec{\mathbf{a}}$ in column form is selected as a single flat reflector model, given by the concatenation of the two vector reflectivity x and z components (Figure 4.4c and 4.4d)

$$\vec{\mathbf{a}} = \begin{bmatrix} \mathbf{a}_x \\ \mathbf{a}_z \end{bmatrix}. \tag{4.33}$$

Similarly, Figure 4.5 shows the model $\mathbf{b}$ represented as the data residuals between the "observed" and "synthetic" forward responses for a source located at the middle of the

model's surface. We use a zero vector reflectivity to compute the latter. The horizontal and vertical components of the gradient using 400 receivers, also deployed at the model's surface, are presented in Figure 4.6. Our numerical results indicate that the discrete adjoint is accurate, passing the dot product test up to (single) numerical precision.

For comparison, we use the exact same configuration to compute the gradient response for the standard LSRTM problem using the adjoint of the Born modeling operator to migrate the data residuals. An important observation is the lack of low-wavenumber energy in both components of the misfit gradient related to the vector reflectivity engine (Figures 4.6a and 4.6b) compared to the image displaying the traditional LSRTM gradient (Figure 4.6c). Furthermore, in the vertical component of the vector reflectivity gradient, the "rabbit-ear" wavepaths associated with the source- and receiver-side wavefields have opposite phases. In seismic acquisitions with a high density of shots and receivers, these opposing wave-paths cancel out, rendering the low-frequency "rabbit-ear" effect negligible on the stacked vertical component of the vector reflectivity. Conversely, the "rabbit-ear" wave-paths have the same phase in the horizontal component of the vector reflectivity gradient, and the migration's "smile" exhibits varying positive and negative values across each side of the shot. Consequently, a multi-shot experiment results in a stacked horizontal component of vector reflectivity with smaller amplitudes and higher low-frequency artifacts. We show this in section 1.5 in a more realistic experiment.

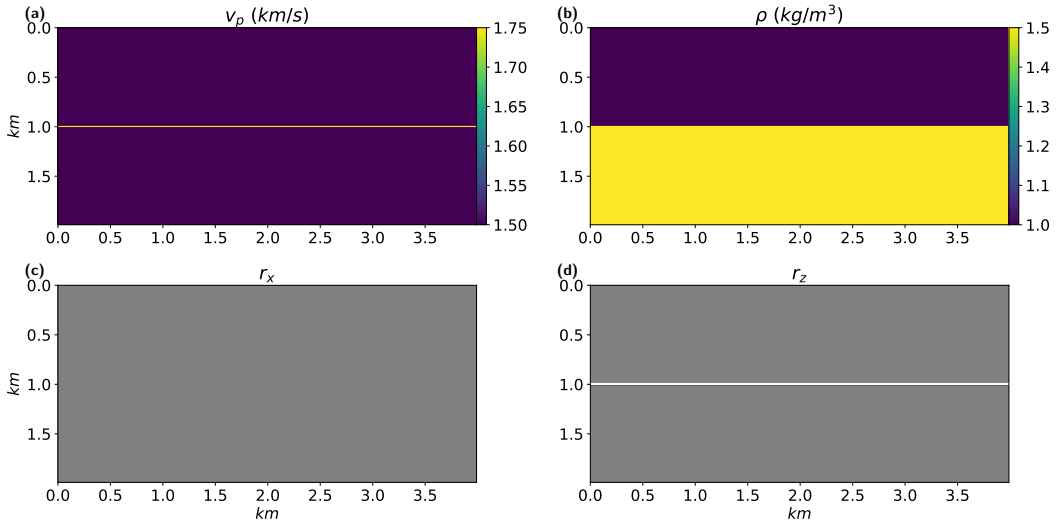

Figure 4.4: Flat reflector experiment to calculate the misfit gradient response of a single shot located at the middle of the model's surface and registered with 400 receivers, also at the surface of the model. (a) velocity model, (b) density model, (c) horizontal and (d) vertical components of the vector reflectivity model.

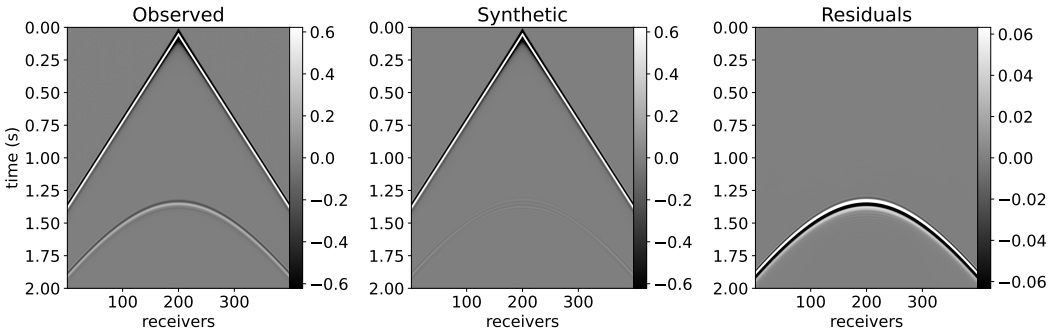

Figure 4.5: Left: "observed" data generated with a variable density acoustic wave equation using the models from Figures 4.4a and 4.4b. Middle: "synthetic" or "calculated" data generated with the full-wavefield vector reflectivity forward modeling using the true velocity model (Figure 4.4a), but a zero vector reflectivity model. Right: the data residuals.

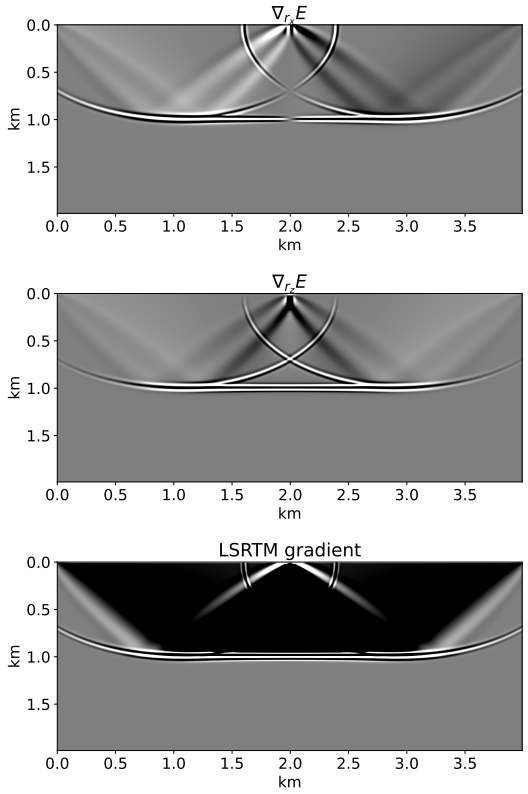

Figure 4.6: Single-shot gradient components for the flat reflector experiment. Top: horizontal component of the vector reflectivity gradient. Middle: vertical component of the vector reflectivity gradient. Bottom: standard LSRTM gradient.

Secondly, we test for the accuracy of the misfit gradient. To do this straightforwardly, we calculate individual elements of the two gradient components using a finite difference central approximation. Specifically, we approximate the derivative of the objective function with respect to the horizontal and vertical components of the vector reflectivity by varying the model parameters and observing the change in the objective function. We test this approximation for a few different points in the computational domain following

$$\frac{\partial E}{\partial r_{x_i}} \approx \frac{E(r_{x_i} + \Delta r_{x_i}, r_{z_i}) - E(r_{x_i} - \Delta r_{x_i}, r_{z_i})}{2\Delta r_{x_i}}, \tag{4.34}$$

$$\frac{\partial E}{\partial r_{z_i}} \approx \frac{E(r_{x_i}, r_{z_i} + \Delta r_{z_i}) - E(r_{x_i}, r_{z_i} - \Delta r_{z_i})}{2\Delta r_{z_i}}, \tag{4.35}$$

where x_i and z_i denote discrete points in the computational grid for each gradient component, and Δr_{x_i} and Δr_{z_i} are small perturbations. Varying the value of these model perturbations between 10^{-6} and 10^{-1}, we observe numerical errors that are below 10^{-2} when compared with the adjoint-state-derived gradient.

4.3 Sparse regularization

Regularization strategies are essential for controlling stability and ambiguity in the solution, particularly when faced with insufficient seismic data. Furthermore, even with sufficient data, maintaining a regularization term is crucial to account for the noise inherent in real seismograms and avoid overfitting, which could result in inaccurate results. Therefore, we modify the original problem described in Equation 4.10 by incorporating the HPF as the regularization term, which is differentiable and thus facilitates the use of straightforward optimization methods based on gradient descent techniques. This allows us to retrieve a sparse model of the vector reflectivity, characterized by a long-tail prior, which proves beneficial when the model components are expected to exhibit spiky characteristics. The regularized cost-functional for the sparsity promoting full-wavefield LSRTM takes the additive form

$$\min_{\vec{r}} J(\vec{r}) = E(\vec{r}) + \mu R_\epsilon(\vec{r}), \tag{4.36}$$

with $\mu > 0$ as the weight balancing the importance of sparseness of the vector reflectivity and data fitting, and

$$R_\epsilon(\vec{r}) = \sqrt{r_x^2 + r_z^2 + \epsilon^2} - \epsilon, \tag{4.37}$$

denotes the HPF term, where ϵ is the l_1/l_2 norm threshold.

The gradient search direction then becomes

$$\nabla J(\vec{r}) = \nabla E(\vec{r}) + \mu \nabla R_\epsilon(\vec{r}), \tag{4.38}$$

with

$$\nabla R_\epsilon(\vec{r}) = \begin{bmatrix} \frac{r_x}{\sqrt{r_x^2 + r_z^2 + \epsilon^2}} \\ \frac{r_z}{\sqrt{r_x^2 + r_z^2 + \epsilon^2}} \end{bmatrix}. \tag{4.39}$$

4.4 Deriving density models from the vector reflectivity

In the oil and gas industry, both density and velocity play an important role in seismic interpretation steps. Therefore, once we obtain the final vector reflectivity, a relative density model can be calculated as a byproduct by solving the normal equations

$$\text{argmin}_m \left(\left\| \frac{1}{2} \begin{bmatrix} D_x \\ D_z \end{bmatrix} m - \begin{bmatrix} r_x \\ r_z \end{bmatrix} \right\|_2^2 + \alpha \|m\|_2^2 \right) \tag{4.40}$$

with a linear solver such as CGLS. In Equation 4.40, D_x and D_y are the first-order derivative operators in the horizontal and vertical dimensions, and α is a damping parameter. The final density model is given by $\rho = \frac{\exp(m)}{v_p}$. This extra step after the inversion extends the applicability of our sparsity-promoting results.

4.5 Numerical experiments

We illustrate the effectiveness of our FWLSRTM workflow in two numerical experiments. The first experiment entails a high-density box model with a homogeneous velocity. The second one is the Marmousi2 model (Martin et al., 2006b). In these experiments, we generate the observed data with a 20 Hz Ricker wavelet using the variable-density acoustic wave equation to alleviate the inverse crime. The data contains multiples, direct arrivals, and diving waves. The results are compared to traditional (i.e., linear/Born-modeling-based) LSRTM, which requires a mute to be applied to the data, removing direct, diving, and far-offset events. All reflectivity images are plotted on the same scale for a fair comparison. In all the examples, we set 30 iterations as the stopping criteria for the optimization algorithm.

4.5.1 A high-density box model

We first demonstrate the effectiveness of FWLSRTM in resolving reflectors with large dip angles using a simple high-density box model, as shown in Figure 4.7a. The migration velocity is a homogeneous background velocity model (not shown). Figures 4.7c and 4.7d show the horizontal and vertical components of the true vector reflectivity parameter, respectively. The model size is 101×101 with a grid spacing of 10 m in the x and z directions. We generate 31 shots with a 33 m shot interval, each with 101 receivers with a 10 m receiver interval. The record length is 1 s, with a sampling interval of 2 ms.

Figure 4.7b shows the traditional LSRTM result, which is deficient in recovering the horizontal component of the reflectivity at limited observation apertures. On the other hand, FWLSRTM can resolve all reflectors as shown in Figures 4.7e and 4.7f, demonstrating its ability to fully use more recorded wavefield information, such as prismatic waves and internal multiples. We find lower resolution and decreased amplitude balance in the recovered vertical reflectors from the FWLSRTM-inverted vector reflectivity horizontal component compared to the retrieved vertical reflectivity component. This is because, with limited offset, the primary reflections can mostly illuminate the horizontal reflectors, and the multiples illuminating the vertical reflectors provide lower wavenumber information about the model compared to primaries (Huang and Schuster, 2014). We also note a phase discrepancy in the reflectors produced by LSRTM relative to those from FWLSRTM, potentially attributable to the fact that LSRTM inverts for velocity perturbations. Although the results of this simple example could be enhanced with regularization, we do not apply sparsity constraints; rather, we focus on showcasing the potential of FWLSRTM in recovering steeply dipping structures by using a more comprehensive physics model than traditional LSRTM.

4.5.2 Application to the Marmousi2 model

The FWLSRTM method is now tested on the more complex Marmousi example. The survey comprises 25 shots and 667 receivers evenly distributed at the surface. Figures 4.8a and 4.8b show the true velocity and density models. Figure 4.8c shows the smooth velocity model used as the background migration field. Figures 4.9a and 4.9b show the horizontal and vertical components of the true vector reflectivity model. The initial guess for the inversions is set to a zero vector reflectivity, and μ was selected through trial and error and kept fixed throughout iterations.

We obtain the final vector reflectivity components for the seismic data inversion without (Figures 4.9c and 4.9d) and with (Figures 4.9e and 4.9f) HPF regularization after 30 iterations of L-BFGS. Inversion results with this dataset show that the iterative migrations

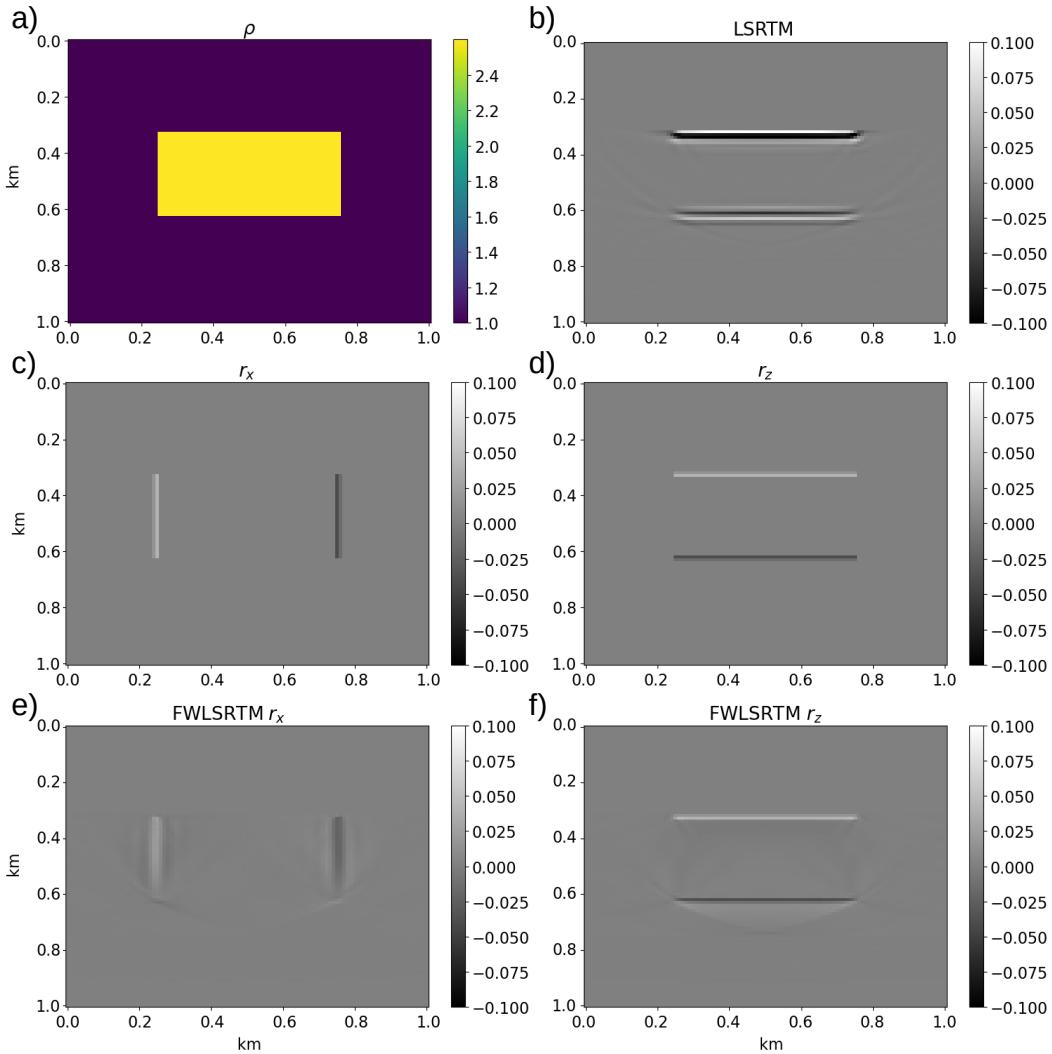

Figure 4.7: High-density box model with homogeneous velocity experiment: a) density model, b) LSRTM, c) true horizontal vector reflectivity component, d) true vertical vector reflectivity component, e) FWLSRTM horizontal vector reflectivity component, f) FWLSRTM vertical vector reflectivity component.

with HPF regularization are sparse and present fewer migration artifacts, less contaminating sidelobes, and better signal content than their non-regularized counterpart. Moreover, all the least-squares results have correct amplitudes and behave as an approximate inversion since forward modelling reproduces the data approximately, as shown in Figure 4.10 for a shot positioned at the surface at x=3.33 km. No frequency domain transformations or spatial filtering techniques such as high-pass filters, Laplacian filters, or any other form of image enhancement were employed to produce these final results. Figure 4.11 compares the

inverted results, with and without sparse regularization, and the true vertical component of the vector reflectivity for a trace in depth extracted at the middle of the model. We observe that in both cases, the inverted results match well with the corresponding defined reflectivity model. The inverted result with HPF regularization has more high-wavenumber components and better agrees with the true reflectivity.

For comparison, we also conducted 30 iterations of traditional LSRTM (as shown in Figure 4.8d), in which we applied a mute to the data to remove direct, diving and far-offset events. Despite this, the final image still exhibits substantial cross-talk attributable to parameter leakage. This is because standard LSRTM only inverts for velocity perturbations using an imprecise modeling engine. On the other hand, with the FWLSRTM, reflectivity changes caused by density variations are not erroneously mapped as velocity updates. We recognize that other studies have suggested multi-parameter LSRTM to deal with the cross-talk between model parameters, including elastic (Chen and Sacchi, 2017b) and variable-density (Farshad and Chauris, 2021) reformulations. However, they still rely on and are restricted by the assumptions inherent to Born modeling, and require substantially greater computational resources.

Contrasting the results of FWLSRTM with traditional LSRTM, which heavily relies on an accurate velocity model due to the Born approximation, also reveals insights into how the vector reflectivity parameterization acts as a model-extension approach. It inherently makes the problem less over-determined by increasing the number of unknowns in the system. This comparison suggests that the FWLSRTM method is likely more resilient to the challenges of inaccurate velocity models. This observation becomes increasingly evident in the deeper sections of our results. FWLSRTM also significantly reduces backscattering noise compared to traditional LSRTM, enhancing seismic imaging.

Figure 4.12 shows the imaging results for the same experiment but changing the migration velocity model to a highly smoothed velocity, as shown in Figure 4.12a. The results indicate that FWLSRTM still exhibits fewer artifacts and provides more accurate images than traditional LSRTM. Sparse FWLSRTM further reduces the high-frequency artifacts. With an inaccurate model, the seismic events are not expected to indicate the true reflection position. However, the better focused events after migration can give a hint and help the interpretation to better identify the true events and coherent noise. In practical scenarios, LSRTM tends to be more susceptible to migration velocity errors compared to conventional RTM because it introduces incoherent stacking in each iteration, leading to blurring effects. To manage these unknown velocity discrepancies, pre-stack strategies are typically used in LSRTM, adding an extra dimension through a shot index, ray-parameter, time-lag, or subsurface offset to mitigate the issue. Future research should investigate the adaptation of these strategies to FWLSRTM to further refine the accuracy of imaging with respect to

velocity model errors.

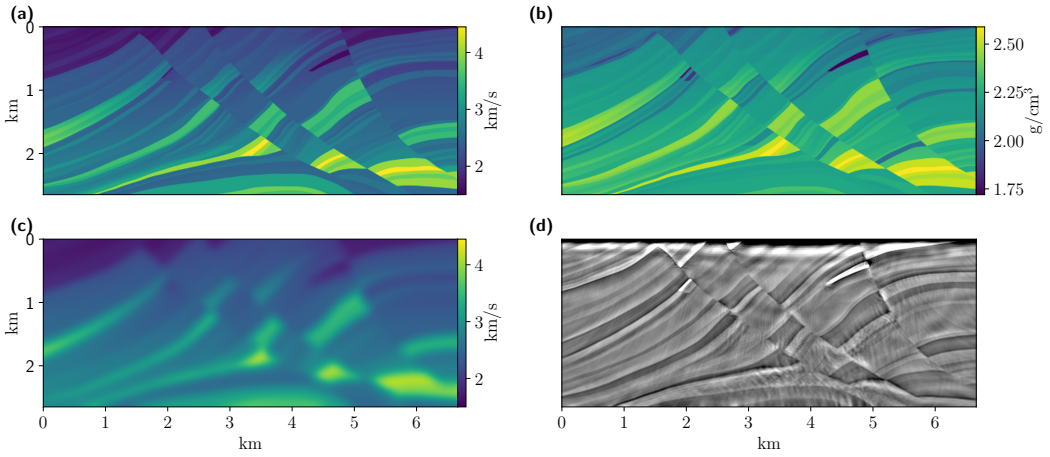

Figure 4.8: The 2D Marmousi2 model: (a) true velocity model, (b) true density model, (c) background migration velocity field, (d) traditional LSRTM inversion.

4.6 Conclusions

Classic LSRTM is solved in a linear inversion framework, which can be restrictive in handling data from complex geological settings where non-linearity might be significant. We successfully obtained enhanced vector reflectivity models by re-formulating LSRTM as a non-linear full-wavefield least-squares inversion in combination with sparse regularization. In our numerical experiment, applying this method to the 2D Marmousi2 model demonstrated its clear superiority. The parameter μ, which controls the regularization strength, requires tuning. Our results indicate that sparse full-wavefield LSRTM reduced artifacts and sidelobes more effectively and delineated reflectors more precisely than traditional methods, even in the presence of highly decimated data. Moreover, the ability of our approach to adapt to imprecise velocity models highlights its potential as a robust tool in seismic imaging, especially in complex geological settings. Our tests imply that the nonlinear FWLSRTM workflow may produce more accurate updates in combination with velocity updates in FWI, leading to a better convergence.

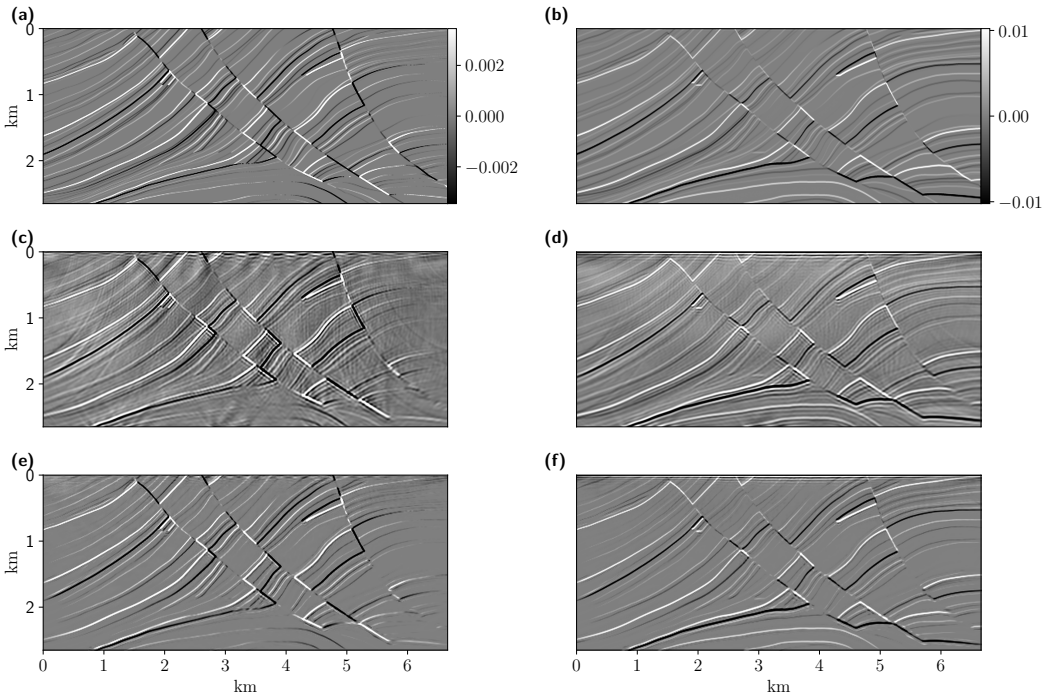

Figure 4.9: Full-wavefield LSRTM inversion of the 2D Marmousi2 model: (a), (c) and (e) show the horizontal components for the true vector reflectivity and the inversions without and with HPF regularization, respectively. Similarly, (b), (d) and (f) show the true and inverted vertical components. All the results are displayed using consistent amplitude scales, as demonstrated in Figures (a) and (b), thereby accurately representing true amplitude reflectivity imaging.

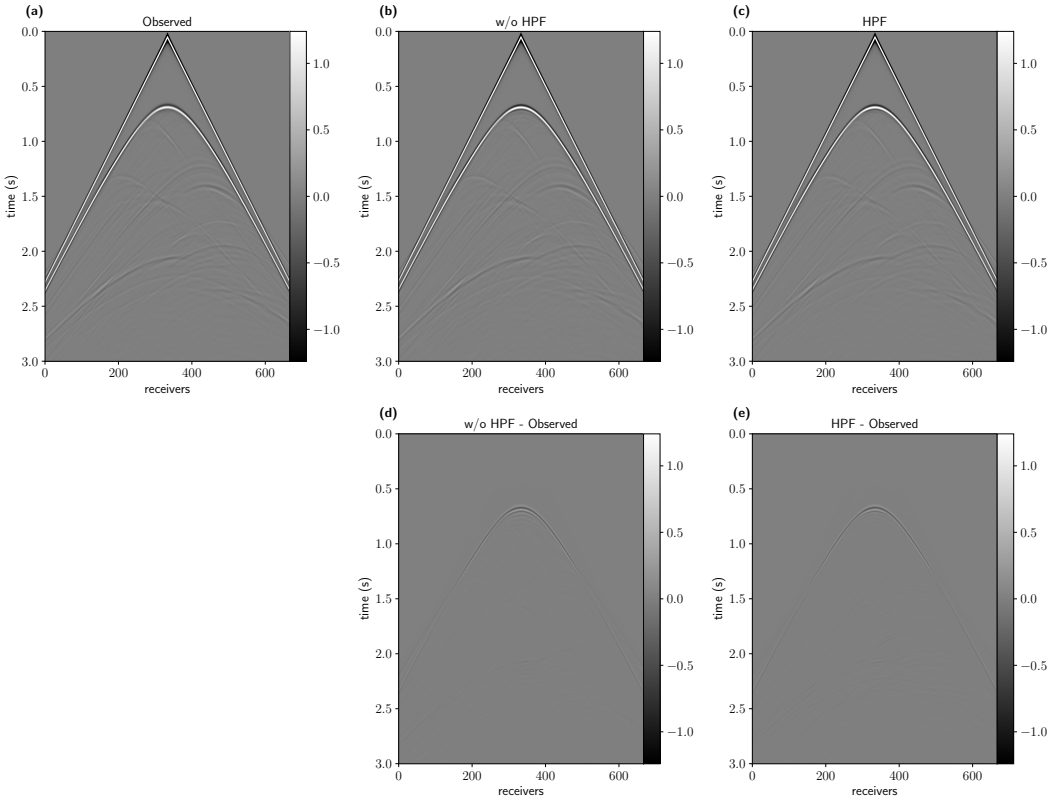

Figure 4.10: Comparison of traces in the data domain for the Marmousi2 inversion experiment for a shot positioned at the surface at x=3.33 km. (a) Observed data using the true velocity and density models. (b) and (c) show the forward modelling shot gather obtained with the vector reflectivity engine using the inverted results without and with sparse regularization, respectively. (d) and (e) show the data residuals for each result ((a) - (b) and (a) - (c)).

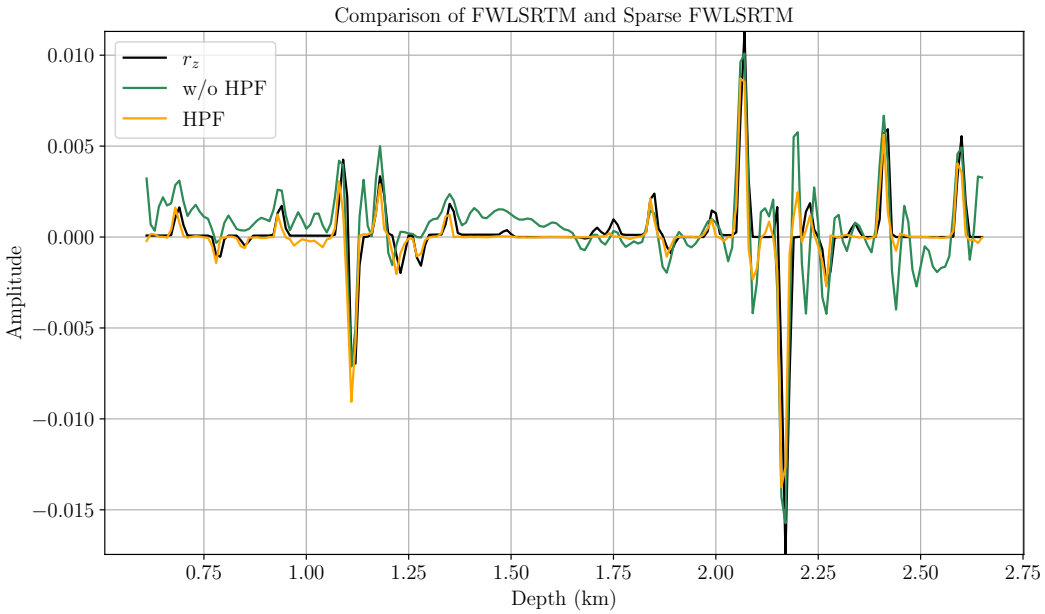

Figure 4.11: Comparison of traces from the vertical component of the inverted results, with (orange) and without (green) sparse regularization, and the true reflectivity model (black) at location $x = 3.33$ km.

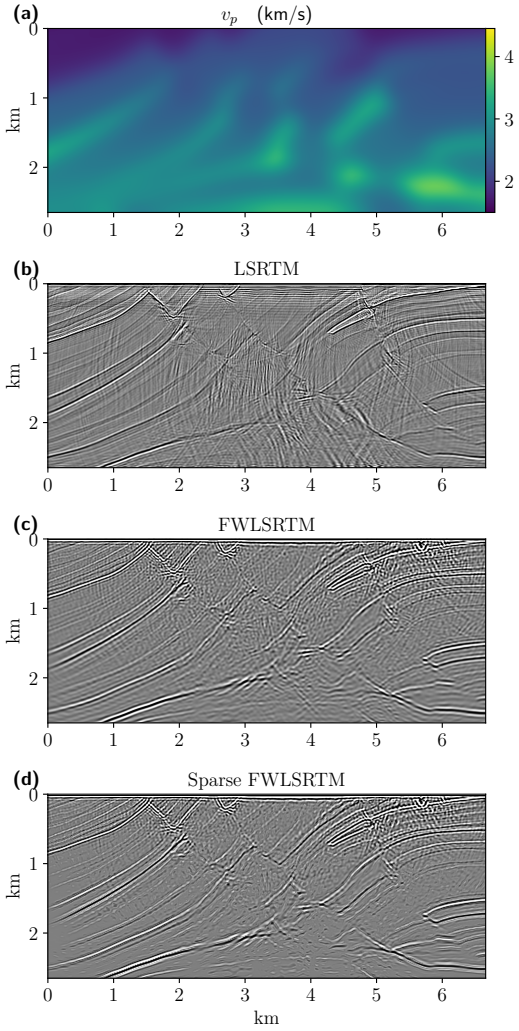

Figure 4.12: Full-wavefield LSRTM inversion of the 2D Marmousi2 model using a highly smoothed velocity model: (a) velocity model, (b) standard LSRTM result, (c) vertical component of full-wavefield LSRTM, (d) vertical component of full-wavefield LSRTM with sparse regularization. A Laplacian filter was applied to the images to remove low-frequency artifacts.

CHAPTER 5

Learned reconstruction with a nullspace-range decomposition[1]

5.1 Introduction

Enhancing the resolution of band-limited seismic data via effective deconvolution techniques has always been a recurring goal in exploration seismology. Superior bandwidth content translates into potentially resolving reflectors hidden under the tuning thickness and ultimately provides more accurate structural and stratigraphic information from high-resolution images (Chopra et al., 2006, 2009). After the reflection data undergoes a sequence of processing steps (Levin, 1989), a common assumption is that the zero-offset seismic trace can be modelled as a linear system. The linear system entails the convolution of a band-limited source wavelet (i.e., the blurring kernel with most of its energy concentrated within some pass-band) with the earth's impulse response, which is typically conceived as a broadband reflectivity time series representing layers of constant material parameters (Robinson and Treitel, 1980). The deconvolution process attempting the frequency enhancement then considers the computation of an approximate solution for the discretized problem

$$\mathbf{d}_\epsilon = \mathbf{L}\mathbf{m} + \epsilon, \tag{5.1}$$

where $\mathbf{m} \in \mathbb{R}^n$ is the (vectorized) reflectivity time series, $\mathbf{d} \in \mathbb{R}^m$ contains the seismic traces, $\epsilon \in \mathbb{R}^m$ denotes an unknown data error represented as additive noise, and the linear forward operator $\mathbf{L} : \mathbb{R}^n \to \mathbb{R}^m$ contains the stationary seismic wavelet properly arranged

[1] A version of this chapter has been published as a journal article: Torres, K. and Sacchi, M. D., 2023, Deep decomposition learning for reflectivity inversion, Geophysical Prospecting, 71(6), pp.963-982.

into a Toeplitz matrix

$$\mathrm{L}_{ij} = \begin{cases} w_{i-j+1} & i \geq j \\ 0 & i < j \end{cases}. \tag{5.2}$$

This enables convolution via simple matrix-times-vector multiplication, with w_{i-j+1} as the $i-j+1$th sampling point of the wavelet. For this purpose, this chapter adopts a classic deconvolution framework, in which the noise is white and Gaussian, and the wavelet is assumed time-invariant and known or at least well approximated as a preliminary step (Ulrych et al., 1995). While a more realistic model entails an unknown and non-stationary propagating wavelet, this simplification serves as a good approximation. We point out, however, that multiple efforts have also considered time-variant and blind deconvolution frameworks that simultaneously estimate the wavelet and the reflectivity in a non-linear fashion (e.g. Kaaresen and Taxt, 1998; Kazemi and Sacchi, 2013; Gholami and Sacchi, 2013; Chen et al., 2023).

Even when the wavelet is available a priori, reflectivity inversion is an underdetermined problem due to the presence of a non-trivial kernel or null space, resulting from the lack of low and high frequencies of the seismic wavelet. Rigorously, the missing model components of $\mathbf{m}$ lie in the spectrum gaps describing the null space of the forward operator. Given the existence of non-uniqueness, many reflectivity series fit the acquired data equally well, despite having radically different features. Only some of these solutions can accurately characterize the true earth's impulse response. Additionally, the ill-posedness of the problem makes the reconstruction process more vulnerable to noise, especially in the high and low bands where the noise contribution is stronger. Thus, a direct inversion of the Toeplitz matrix $\mathbf{m} = \mathbf{L}^{-1}\mathbf{d}_\epsilon$ is impossible, and appropriate priors upon the reflectivity must be promoted to reduce the null space ambiguity and determine a unique and credible approximation of $\mathbf{m}$.

Early attempts to perform reflectivity inversion via deconvolution are based on classic inverse-filtering theory assuming a white random reflectivity sequence with Gaussian prior distribution and a minimum-phase seismic wavelet (Berkhout, 1977; Robinson and Treitel, 1980; Scales and Smith, 1994; Yilmaz, 2001). Such restrictions render a stable and causal wavelet inverse filter that can be applied to the data to retrieve a reflectivity estimate. Nevertheless, gaussianity yields band-limited results with broadened peaks and side-lobe artifacts that preclude closely spaced reflectors from being sharply resolved. To overcome the shortcomings that typify conventional least-squares deconvolution, many methods have been proposed for high-resolution, sparse-spike or thin-bed reflectivity inversion, including techniques that use minimum entropy (Wiggins, 1978; Sacchi et al., 1994) and l_p $(0 \leq p \geq 1)$ sparsity constraints (Taylor et al., 1979; Levy and Fullagar, 1981; Oldenburg et al., 1983; Debeye and Van Riel, 1990; Sacchi, 1997; Chopra et al., 2006; Zhang and Castagna, 2011; Gholami and Sacchi, 2012).

Particularly, sparse priors are only valid through mathematically enforcing that the reflectivity series consists of a few isolated spikes comprising a plane homogeneous layered model. Under this simplified construction, the seismic trace constitutes a finite superposition of seismic wavelets. This hypothesis is suitable when only a few strong reflections dominate the seismogram. In reality, however, well-log data presents a stochastic pattern that is far more complex (Walden, 1985; Tenorio, 2001). Furthermore, when sub-optimally applied as a trace-by-trace process, sparse deconvolution algorithms do not consider the inherent continuity in the spatial dimensions, and we require more complicated multichannel techniques to avoid harming the signal (Idier and Goussard, 1993; Kaaresen and Taxt, 1998; Gholami and Sacchi, 2013), which still may face challenges for seismic data with complex structures.

Additionally, sparse inversion algorithms suppress random noise to some extent when Gaussian statistics are assumed for the data error, but they often have significant sensitivity to outliers that negatively impacts the results (Debeye and Van Riel, 1990). Ultimately, practical applications of high-resolution deconvolution still face several challenges related to regularization and hyper-parameter selection, intensive demand for human-computer interaction, and the high computational cost of iterative reconstruction in large 3D seismic volumes. Despite providing state-of-the-art solutions, limitations from utilizing handcrafted priors inspire the development of alternative data-driven and learning-based seismic data processing methods.

In recent years, various geophysical problems have explored applications of supervised deep learning techniques using Convolutional Neural Networks (CNN) (LeCun et al., 2015), for they have powerful representation learning properties and the potential to process extensive seismic surveys with minimal human intervention (Yu and Ma, 2021c). In the supervised regime, one trains a neural network as a universal approximator to recover model parameters from observed data with many high-quality pre-labelled solutions from a representative (training) dataset. Deep learning applications on seismic inversion include end-to-end approaches (Araya-Polo et al., 2018; Mandelli et al., 2019; Chai et al., 2021a; Wu et al., 2021a), in which the neural networks directly learn an inverse data-to-model mapping. Such networks bypass the use of explicit physics operators but rely on a vast amount of training samples to learn the underlying physics of the problem. To reduce the dependency on training data, learned iterative schemes (Torres and Sacchi, 2022b) incorporate physics into the learning process and replace various components of unrolled iterative reconstruction algorithms with neural network computations. Alternatively, to avoid iterations, the learned post-processing method first maps the measurements to the model space through a known physics operator (either the pseudo-inverse $\mathbf{L}^{\dagger} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ or an approximation to it) and then trains a neural network to learn a model perturbation that potentially improves this initial reconstruction (Kaur et al., 2020; Zhang et al., 2021). Even though these techniques have demonstrated remarkable empirical success, many supervised approaches still lack a

data consistency constraint to enforce that the predicted model matches the acquired data, a necessary condition for a reliable solution to the inverse problem.

Consequently, such deep learning schemes do not lead to convergent regularization strategies. Hence, the results might look realistic, but there is no way to assess their accuracy. Unsupervised approaches (Dhara and Sen, 2022; Chen et al., 2022; Kong et al., 2022) address this issue by design, but they often amplify the expensive iterative nature of traditional methods when the trainable weights are not correctly initialized.

Schwab et al. (2019) introduced regularization via null space networks as an alternative image domain restoration method to account for data consistency. By computing a projection onto the null space of the forward operator after the last weight layer of a residual architecture, it is possible to train a neural network to learn the missing components of the initial reconstruction. The null space projection ensures that the output estimates are consistent with the observed input data. As a generalization to null space networks for noisy data, Chen and Davies (2020) introduced the concept of deep decomposition learning, which attaches a complementary network to act as a denoiser on the range of the pseudo-inverse. Similarly, Schwab et al. (2020) allow the null space networks to act on the orthogonal complement of the kernel by being dependent on the regularization technique that produces the initial reconstruction and demonstrate the convergence properties of these algorithms. Based on these ideas, we investigate the extension of data-consistent null space learning on the inversion of reflectivity. Specifically, we extend the deep decomposition approach by using the truncated singular value decomposition (TSVD) as an initial regularized reconstruction for approximating the low-frequency components of the model. As a second step, we trained two neural networks to recover the missing parts of the model and the "inverted" noise, respectively.

This chapter is an extension of Torres and Sacchi (2022c). In the next section, we reintroduce the concept of null space networks for the convolutional model for normal incidence seismograms. To our knowledge, regularization via null space networks has not been reported in the literature in the context of full band reflectivity inversion. As a proof of concept, we empirically demonstrate the behaviour of the proposed method on a single-channel toy dataset in which the optimal solution is known and two 2D field datasets. These examples show that when the wavelet is known, the proposed deconvolution approach may reconstruct a suitable high-resolution reflectivity model from band-limited noisy data. The performance, we believe, is comparable to the state-of-the-art techniques in this category. The chapter ends with a discussion of limitations and prospects for further development of this reflectivity inversion method.

5.2 Method

5.2.1 Deconvolution

Many classic deconvolution methods find approximate deterministic solutions to equation 5.1 by computing the reflectivity model $\mathbf{m}^\star$ that minimizes a composite objective function

$$\mathbf{m}^\star = \arg\min_{\mathbf{m}} \big\{ J(\mathbf{m}) + \lambda R(\mathbf{m}) \big\}, \tag{5.3}$$

where the first term is a convex data-fidelity term, usually defined as the least-squares data error $J(\mathbf{m}) = ||\mathbf{L}\mathbf{m} - \mathbf{d}_\epsilon||_2^2$, $R(\mathbf{m})$ represents a regularization function, and $\lambda > 0$ denotes the regularization parameter. For example, Wiener or Tikhonov regularized deconvolution can be derived assuming a Gaussian prior for the reflectivity $R(\mathbf{m}) = ||\mathbf{m}||_2^2$, which has an analytical solution

$$\mathbf{m}_\lambda^\star = [\mathbf{L}^T\mathbf{L} + \lambda\mathbf{I}]^{-1}\mathbf{L}^T\mathbf{d}_\epsilon, \tag{5.4}$$

where $\mathbf{L}^T$ is the transpose of $\mathbf{L}$ (a cross-correlation operator) and $\mathbf{I} \in \mathbb{R}^n$ is the identity operator. Analogously, when the problem allows an explicit singular value decomposition (SVD) of $\mathbf{L}$, the truncated SVD (TSVD) pseudo-inverse $\mathbf{L}_k^\dagger$ can be used as an intuitive brute-force low-rank approximation to obtain a filtered solution

$$\mathbf{m}_k^\star = \mathbf{L}_k^\dagger \mathbf{d}_\epsilon \tag{5.5}$$

$$= \mathbf{V}_k \mathbf{S}_k^{-1} \mathbf{U}_k^T \mathbf{d}_\epsilon, \tag{5.6}$$

with $\mathbf{U}_k \in \mathbb{R}^{m\times k}$ and $\mathbf{V}_k \in \mathbb{R}^{n\times k}$ as the top k rows of the unitary matrices $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} \in \mathbb{R}^{n\times n}$, respectively, and $\mathbf{S}_k \in \mathbb{R}^{k\times k}$ as a diagonal matrix containing the largest k singular values of $\mathbf{L}$. TSVD and Tikhonov regularizations provide similar solutions under certain conditions (Hansen, 2010a). In both cases, the regularization parameters λ and k attempt to reduce the influence of noise by suppressing the less reliable signal components. However, it is well-known that none of these techniques are suitable when the solution is discontinuous. Deconvolution via Tikhonov regularization, for instance, tends to smear the solution by enforcing that the reflectivity coefficients concentrate close to zero, considering $R(\mathbf{m}) = ||\mathbf{m}||_2^2 = ||\mathbf{m} - \mathbf{m}_{\text{prior}}||_2^2$ with $\mathbf{m}_{\text{prior}} = \mathbf{0}$ as a vector of all zeros. The TSVD pseudo-inverse truncates the spectrum of $\mathbf{L}$ and only provides minimum norm solutions since no elements in the null space of $\mathbf{L}$ are added to the solution.

On the other hand, the layered earth model visualizes the unknown reflectivity series as a

train of isolated spikes represented by a set of delta functions

$$m_n = \sum_{s=0}^{S-1} a_s \delta_{n-\Gamma_s} \tag{5.7}$$

where S is the number of nonzero spikes of amplitude a_s, and $\Gamma \subset \{0, ..., N-1\}$. Thus, enforcing a sparse property to the reconstructed reflectivity through the long-tailed l_1-norm while retaining the l_2-norm on the data-fidelity term yields efficient sparse-spike deconvolution algorithms that seek the solution to the convex problem

$$\min_{\mathbf{m}} \{ \frac{1}{2} ||\mathbf{Lm} - \mathbf{d}_{\text{obs}}||_2^2 + \mu ||\mathbf{m}||_1 \}, \tag{5.8}$$

which allows for resolving closely spaced reflectors. Nonetheless, controlling the sparsity level of the reconstructed reflectivity is challenging. In particular, because of the constant shrinkage parameter μ, l_1-regularization results in a biased estimate by over-penalizing large-valued reflectivity coefficients. Alternatively, we propose to solve the reflectivity inversion problem given in equation 5.1 with a deep decomposition technique based on null space network regularization that constructs a learnable prior from training data and possesses data consistency and denoising properties.

5.2.2 Deep decomposition learning for reflectivity inversion

The outset of the deep decomposition learning as a regularization technique is the active-null space decomposition of the signal, i.e., we decompose the domain of the forward operator into two sub-spaces: the measurement space and the null space. Accordingly, we might think of any reflectivity model $\mathbf{m}$ in the domain of $\mathbf{L}$ as being made up of two unique orthogonal vectors,

$$\mathbf{m} = \mathbf{m}_R + \mathbf{m}_N = P_R(\mathbf{m}) + P_N(\mathbf{m}), \tag{5.9}$$

such that $\mathbf{m}_R$ lies in the range of the pseudo-inverse $\mathbf{L}^\dagger$ which is also the active space solution, and $\mathbf{m}_N$ lies in the null space. By definition, these two components satisfy, respectively,

$$\mathbf{m}_R = \mathbf{L}^\dagger \mathbf{d}_\epsilon \tag{5.10}$$

$$= \mathbf{L}^\dagger \mathbf{Lm} + \mathbf{L}^\dagger \epsilon, \tag{5.11}$$

and

$$\mathbf{Lm}_N = 0. \tag{5.12}$$

Based on this fragmentation of the model, we can define the ideal reconstruction as

$$\mathbf{m}^* = \mathbf{L}^\dagger \mathbf{d}_\epsilon - \mathbf{L}^\dagger \epsilon + \mathbf{m}_N. \tag{5.13}$$

In other words, the solution is expressed in terms of a unique minimum norm least-squares solution ($\mathbf{L}^\dagger \mathbf{d}_\epsilon$) minus the "inverted" noise plus the null space vector.

As denoted in equation 5.9, the model components can be obtained from two orthogonal projections, P_R and P_N, defined as

$$P_R = \mathbf{L}^\dagger \mathbf{L}, \tag{5.14}$$

and

$$P_N = \mathbf{I} - \mathbf{L}^\dagger \mathbf{L}. \tag{5.15}$$

Using a physics-engaged approach promoted by the application of the above-mentioned orthogonal projections, deep decomposition learning attempts to solve equation 5.13 with a trained estimator $\Lambda : \mathbb{R}^m \to \mathbb{R}^n$ defined as

$$\Lambda(\mathbf{d}_\epsilon; \theta_1, \theta_2) = \mathbf{L}^\dagger \mathbf{d}_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger \mathbf{d}_\epsilon + P_N \circ \mathbf{N}_{\theta_2} \circ (\mathbf{L}^\dagger \mathbf{d}_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger \mathbf{d}_\epsilon), \tag{5.16}$$

where $\mathbf{F}_{\theta_1}$ and $\mathbf{N}_{\theta_2}$ are two trainable neural networks. Compared to equation 5.13, it is clear that the second term in the right-hand side of equation 5.16 tries to estimate the negative "inverted" noise by projecting the output of the network $\mathbf{F}_{\theta_1}$ onto the range of the pseudo-inverse. Likewise, the third term in equation 5.16 tries to estimate the null space component from the denoised input $\mathbf{L}^\dagger \mathbf{d}_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger \mathbf{d}_\epsilon \approx \mathbf{L}^\dagger \mathbf{d}_\epsilon - \mathbf{L}^\dagger \epsilon$. When Λ lacks the explicit denoising element ($P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger \mathbf{d}_\epsilon = 0$), the estimator turns into a standard null space regularization network of the form

$$\begin{aligned} \Lambda(\mathbf{d}_\epsilon; \theta) &= \mathbf{L}^\dagger \mathbf{d}_\epsilon + P_N \circ \mathbf{N}_\theta \circ \mathbf{L}^\dagger \mathbf{d}_\epsilon \\ &= (\mathbf{I} + P_N \circ \mathbf{N}_\theta)(\mathbf{L}^\dagger \mathbf{d}_\epsilon), \end{aligned} \tag{5.17}$$

where the data consistency property, $\mathbf{L}\Lambda(\mathbf{d}_\epsilon; \theta) = \mathbf{d}_\epsilon$, is exactly preserved.

For noisy seismic data, the results obtained using the naive solution $\mathbf{L}^\dagger \mathbf{d}_\epsilon$ are non-interpretable. Therefore, in this work, we use TSVD for computing a regularized approximation $\mathbf{L}_k^\dagger$ to the pseudo-inverse, where the regularization parameter is given by the number of non-truncated singular values $k > 0$. One must keep in mind that ill-posed problems in the discrete setting are not the same as numerically rank-deficient problems, which are characterized by a noticeable separation between large and small singular values. Therefore, setting a threshold

for truncation in discrete ill-posed problems is expected to be challenging due to the lack of a clear gap. When the true model is unavailable, or the noise level is unknown, one may refer to data-dependent strategies to approximate the optimal truncation, such as the *optimal singular value hard thresholding* (Gavish and Donoho, 2014).

The solution obtained by TSVD is the classical regularized least-square solution that will yield a reflectivity of spectral properties similar to the original data and no significant gain in bandwidth. TSVD produces a stable solution that prevents small singular values of $\mathbf{L}$ from amplifying noise by only recovering signal components corresponding to sufficiently large singular values. The absent components (frequencies in the null space of the operator) have an unreliable model-to-data mapping and, therefore, will be recovered in the learning stage. In other words, because of the impossibility of precisely differentiating the signal from noise in the active space, we partially recover some reflections with TSVD filtering and then predict the residual noise component with a denoising network. Next, we attempt to increase the high-frequency content of the reflectivity with the estimation of the null space element, which limits the non-uniqueness of the problem. Instead of explicitly incorporating a sparse regularizer in an iterative least-squares inversion, the spike representation of the reflectivity is embedded in the manifold of representative solutions in the training labels. The null space network will learn it during training.

Finally, the overall procedure aims to jointly seek the weights that minimize

$$E(\theta_1, \theta_2) = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{m}^i - \Lambda(\mathbf{d}_\epsilon^i; k, \theta_1, \theta_2)||_2^2 + \lambda_1 \sum_{i=1}^{N} ||\mathbf{L}\mathbf{F}_{\theta_1}(\mathbf{L}_k^\dagger \mathbf{d}_\epsilon^i) - \epsilon^i||_2^2 + \lambda_2 ||\theta_2||_2^2, \quad (5.18)$$

where the first term enforces "benignant" inductive bias by carrying out supervised training on a synthetic dataset $\mathcal{D} = \{(\mathbf{m}^i, \mathbf{d}_\epsilon^i, \epsilon^i)\}_{i=1}^N$ using the Mean Squared Error (MSE) loss, the second term prevents the denoising component from breaking the data consistency property. The third term provides the null space estimator with robustness to small perturbations via weight regularization (Schwab et al., 2019).

We define $\mathbf{F}_{\theta_1}$ as a denoising convolutional neural network (DnCNN) architecture (Zhang et al., 2017) (Figure 5.1) designed to predict the residual noise component from a corrupted input with the operations in the hidden layers. We set $\mathbf{N}_{\theta_2}$ as an encoder-decoder neural network, shown in Figure 5.2, inspired by the U-net architecture (Ronneberger et al., 2015a). This type of architecture is a two-stage convolutional neural network commonly used in image segmentation and reconstruction from insufficient data due to its straightforward design, high representation power, and fast convergence in training. The encoder portion of $\mathbf{N}_{\theta_2}$ implements convolutions and downsampling operations to extract high-level features from the inputs. The decoder then performs convolutional and upsampling operations on the retrieved features to produce the required outputs. Skip connections feed

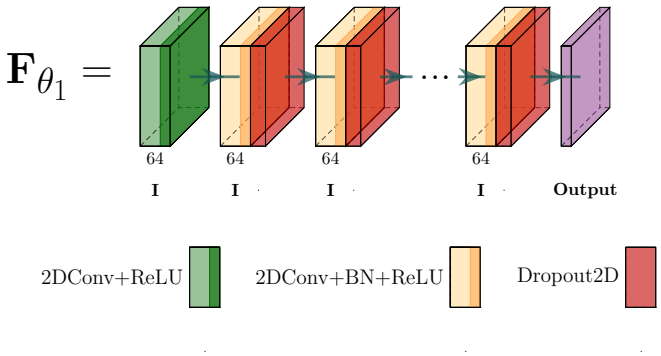

Figure 5.1: Convolutional neural network architecture of the denoiser $\mathbf{F}_{\theta_1}$ with corresponding depth of 17. $\mathbf{I}$ refers to the original image size. The number at the bottom of each convolutional layer indicates the number of channels. The convolutional kernels are of size 3×3. ReLU and BN stand for *rectified linear unit* and *batch normalization*, respectively.

the extracted features from the encoder sections to their respective decoder sections to recover the information lost in the downsampling process, enhancing the model's ability to produce high-quality outputs with the desired size at a reduced computational cost. This architecture enables the model to learn complex representations and patterns by providing a broad field of view of signal features through downsampling operations. However, this capability also increases the risk of overfitting, especially when training data is limited. To address this, we incorporate dropout layers as suggested by (Tompson et al., 2015), which are essential for improving prediction performance while mitigating the risk of overfitting.

A strength of the null space network component is that the estimator $\Lambda(\mathbf{d}_\epsilon; k, \theta_1, \theta_2)$ only adds missing information without introducing inconsistencies with the acquired data, even when applied to models very different from the training data, which in principle can improve the reconstruction quality if compared to non-data-consistent approaches. Algorithm 1 summarizes the proposed scheme.

As noticed, computing the trained estimator only requires one SVD of the forward operator before training, so the technique is practically convenient. The following section presents evidence that regularization via deep decomposition can be applied to field seismic data sets and produce plausible results in the cases exhibited here.

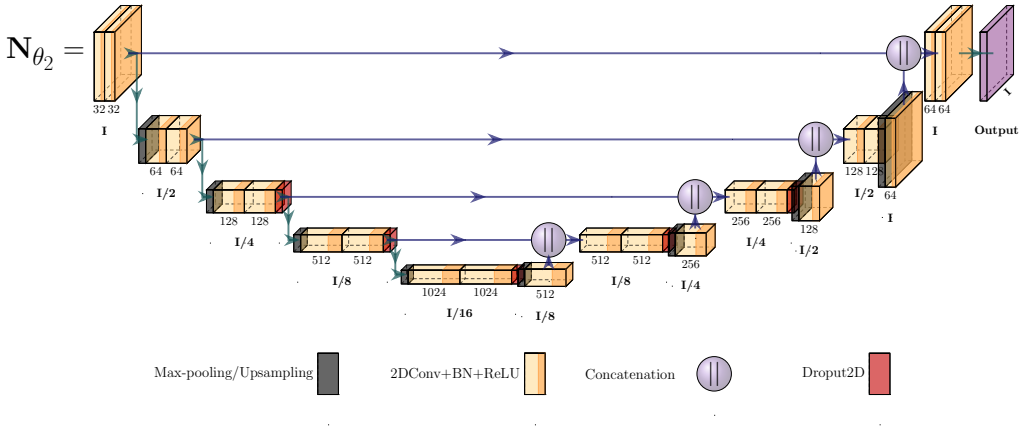

Figure 5.2: U-net based architecture of the null space neural network $\mathbf{N}_{\theta_2}$. $\mathbf{I}$ refers to the original image size. The number at the bottom of each convolutional layer indicates the number of channels. The convolutional kernels are of size 3×3. Max-pooling has a kernel size of 2×2.

5.3 Numerical examples

We evaluate the proposed learned regularization method on a synthetic example and two real data sets. Setting the trade-off parameters to $\lambda_1 = 10e^{-6}$ and $\lambda_2 = 10e^{-8}$ yields stable solutions for the three experiments considered in this section. Stochastic gradient descent with the adaptive moment estimation scheme (Kingma and Ba, 2014b) minimizes equation 5.18, running 400 epochs as the stopping criteria with a learning rate of 0.001.

We incorporate different realizations of unstructured Gaussian noise into the training data samples using a fixed and predefined signal-to-noise ratio SNR=20. Namely, to each i-th training realization, we add a noise vector $\epsilon^i = \alpha^i \mathbf{n}^i$ to the clean signal $\mathbf{d}^i$ such that $\mathbf{d}^i_\epsilon = \mathbf{d}^i + \epsilon^i$. For this, we adopt the definition

$$\begin{aligned} \text{SNR} &= \frac{\text{Power of the clean signal}}{\text{Power of the additive noise}} \\ &= \frac{||\mathbf{d}^i||_2^2}{{\alpha^i}^2 ||\mathbf{n}^i||_2^2} \end{aligned}$$

where α^i is a scalar used to yield the desired SNR and $\mathbf{n}^i$ is sampled from a standard normal distribution.

Algorithm 4 Deep decomposition estimator $\Lambda(\mathbf{d}_\epsilon; k, \theta_1, \theta_2)$

Requires:

- Forward operator $\mathbf{L}$ (constructed with an estimated wavelet as per equation 5.2)
- Projection operators P_R, P_N
- Training dataset $\mathcal{D} = \{(\mathbf{m}^i, \mathbf{d}^i_\epsilon, \epsilon^i)\}_{i=1}^N$

Parameters:

$(k,\ \lambda_1,\ \lambda_2) > 0$, stopping criteria

SVD of $\mathbf{L} \rightarrow \mathbf{L}^\dagger_k$

Training stage:

  repeat for all labels until stopping criteria

    $\Lambda(\mathbf{d}^i_\epsilon; k, \theta_1, \theta_2) = \mathbf{L}^\dagger_k \mathbf{d}^i_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger_k \mathbf{d}^i_\epsilon + P_N \circ \mathbf{N}_{\theta_2} \circ (\mathbf{L}^\dagger_k \mathbf{d}^i_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger_k \mathbf{d}^i_\epsilon)$

    Minimize $\frac{1}{N}\sum_{i=1}^N ||\mathbf{m}^i - \Lambda(\mathbf{d}^i_\epsilon; k, \theta_1, \theta_2)||_2^2 + \lambda_1 \sum_{i=1}^N ||\mathbf{L}\mathbf{F}_{\theta_1}(\mathbf{L}^\dagger_k \mathbf{d}^i_\epsilon) - \epsilon^i||_2^2 + \lambda_2 ||\theta_2||_2^2$

  Output Trained parameters θ_1, θ_2

Inference stage:

  Input: Observed data $\mathbf{d}_\epsilon$

  Output $\mathbf{m}^\star_\Lambda = \mathbf{L}^\dagger_k \mathbf{d}_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger_k \mathbf{d}_\epsilon + P_N \circ \mathbf{N}_{\theta_2} \circ (\mathbf{L}^\dagger_k \mathbf{d}_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger_k \mathbf{d}_\epsilon)$

5.3.1 Synthetic example: single-channel deconvolution

Relying on the convolutional model of the seismic trace, we attempt to recover a known full-band reflectivity sequence from a zero-offset seismic trace. To train the estimator, we generate 5000 1D random reflectivity samples and obtain the corresponding data by convolving a 60 Hz Ricker wavelet and adding the observational noise. The maximum amplitude of the wavelet is unity. The deconvolution process uses the exact wavelet and signal-to-noise ratio.

Figure 5.3 shows the results for a test reflectivity model different from all training samples but generated using the same random procedure. For a quantitative evaluation of the results, we calculate the reconstruction accuracy

$$\text{Accuracy (dB)} = 10 \log_{10} \frac{||\mathbf{m}||_2^2}{||\mathbf{m} - \mathbf{m}^*||_2^2}, \tag{5.19}$$

where $\mathbf{m}$ and $\mathbf{m}^*$ are the true and inverted reflectivity models, respectively. Figure 5.3b shows the initial TSVD solution, where some of the most prominent reflections are partially recovered, but a residual band-limited component masks the non-resolvable events. Moreover, oscillations are visible in the result, courtesy of the truncated expansion. For comparison, Figure 5.3c displays a sparse-spike deconvolution obtained as a solution of equation 5.8 via 100 iterations of the fast iterative shrinkage-thresholding algorithm (FISTA) (Beck and Teboulle, 2009) with regularization parameter $\mu = 0.5$. Even though the inversion with sparsity promotion enables a full-band solution, the noisy data impedes successfully retrieving the correct amplitude and positioning of some events, and the solution presents spurious

spikes. Figure 5.3d shows the result obtained with deep decomposition regularization, where we conclude that, both visually and quantitatively, the learned estimator inversion produces higher quality results and fewer spurious events compared to the two previous techniques. Finally, Figures 5.4a and 5.4b show the amplitude spectrum of the true and the estimated result and the solution's range and null space components, respectively. Both figures show that deep decomposition regularization approximately recovers the true missing frequency components of the original signal.

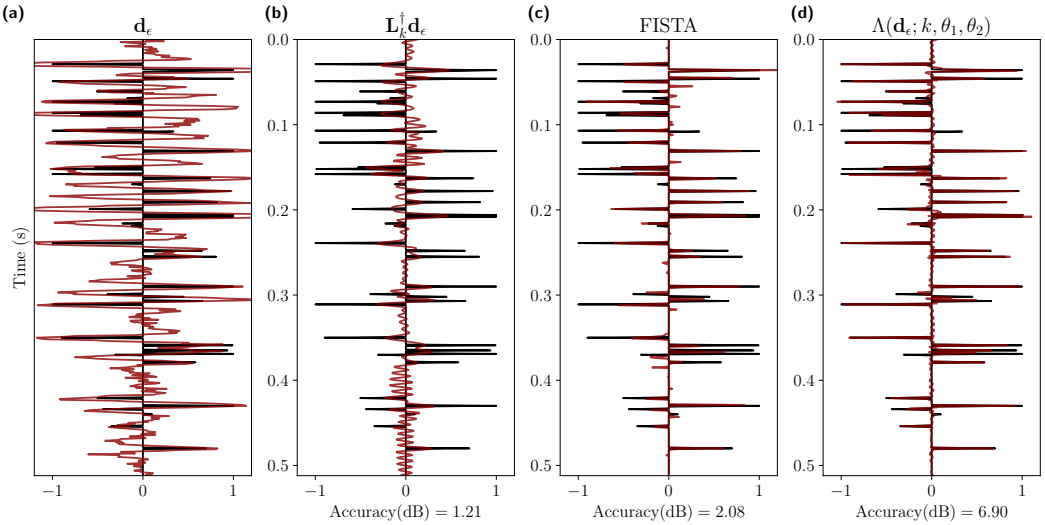

Figure 5.3: Single-channel deconvolution example. True reflectivity $\mathbf{m}$ (black) superposed with (a) the noisy trace $\mathbf{d}_\epsilon$, (b) the TSVD result, (c) the FISTA result, and (d) the deep null space regularization.

The following section will test the proposed method on real post-stack seismic data. Despite the fact that the post-stack process typically yields a high SNR, the proposed deep estimator has been designed to function in the presence of noise. We test how the method reacts with noisier datasets by injecting different values of SNR into the observed data and recording the reconstruction accuracy using equation 19. The prediction results are shown in Table 1. We notice only a slight deterioration in accuracy with SNR values different from the one used in training. The deterioration is stronger for lower SNR values.

Albeit synthetically, this example demonstrates that the proposed algorithm provides a broad-band, sparse and spiky reflectivity solution that, under controlled conditions, is more faithfully resolved than the filtered TSVD and spike-sparse deconvolution.

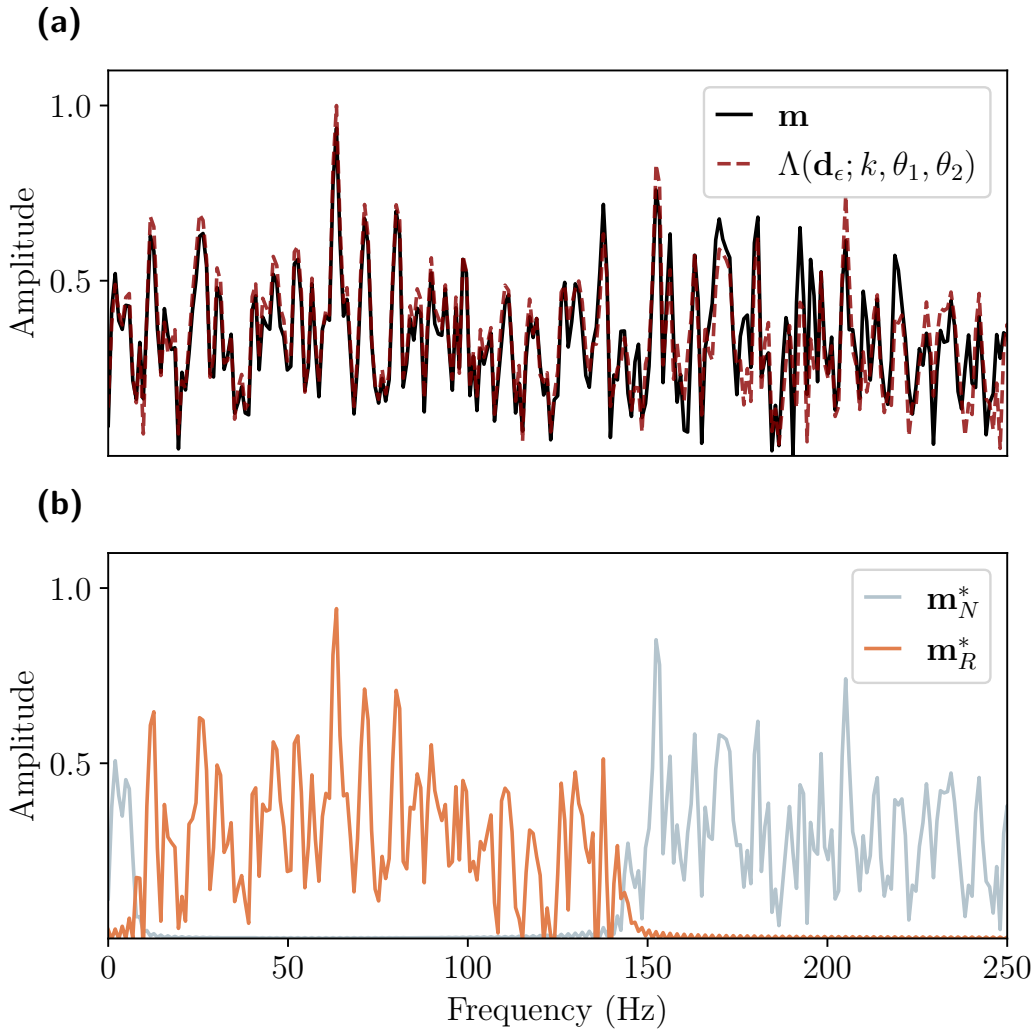

Figure 5.4: Comparison of (a) the amplitude spectrum of the true reflectivity model and the deep null space regularization solution, and (b) the amplitude spectrum of the individual range (orange) and null space (gray) retrieved components.

5.3.2 Field data examples

We also adopted the proposed estimator to predict full-band reflectivity models from two real 2D datasets. In a 1D trace-by-trace deconvolution technique, thin layers can be vertically subtle, and the desired spatial coherence might be easily lost (Wu et al., 2021a). Instead of using 1D models and noisy traces, the networks were trained with synthetically generated 2D reflectivity models and their corresponding noisy data slices to improve lateral continuity and robustness to noise. We follow Torres and Sacchi (2022b) to generate

SNR	Accuracy (dB)
24	6.82
22	6.86
20	6.90
18	6.79
16	6.43
14	5.92
12	5.86
10	5.55

Table 5.1: Testing the sensitivity to random noise on the single-channel deconvolution example using different values of SNR. The filled cells designate the SNR value used for the training stage.

synthetic reflectivity models (as shown in Figure 5.5) considering the spatial correlation imposed by depositional processes while mimicking fractured, faulted and folded sedimentary structures that can take arbitrary shapes and orientations. The training samples reflect that physical, nonrandom processes produce the earth's geology, and subsurface layers can have many arbitrary orientations, including sharp discontinuities.

In the following tests, the training stage was performed on a workstation with one NVIDIA GeForce RTX 2080 Ti graphics processing unit (GPU). The total training time for running 400 epochs on the 2D datasets with 5000 training samples of dimensions 144×496 took approximately 15 minutes and 8.6 GB of VRAM using a batch size of 4.

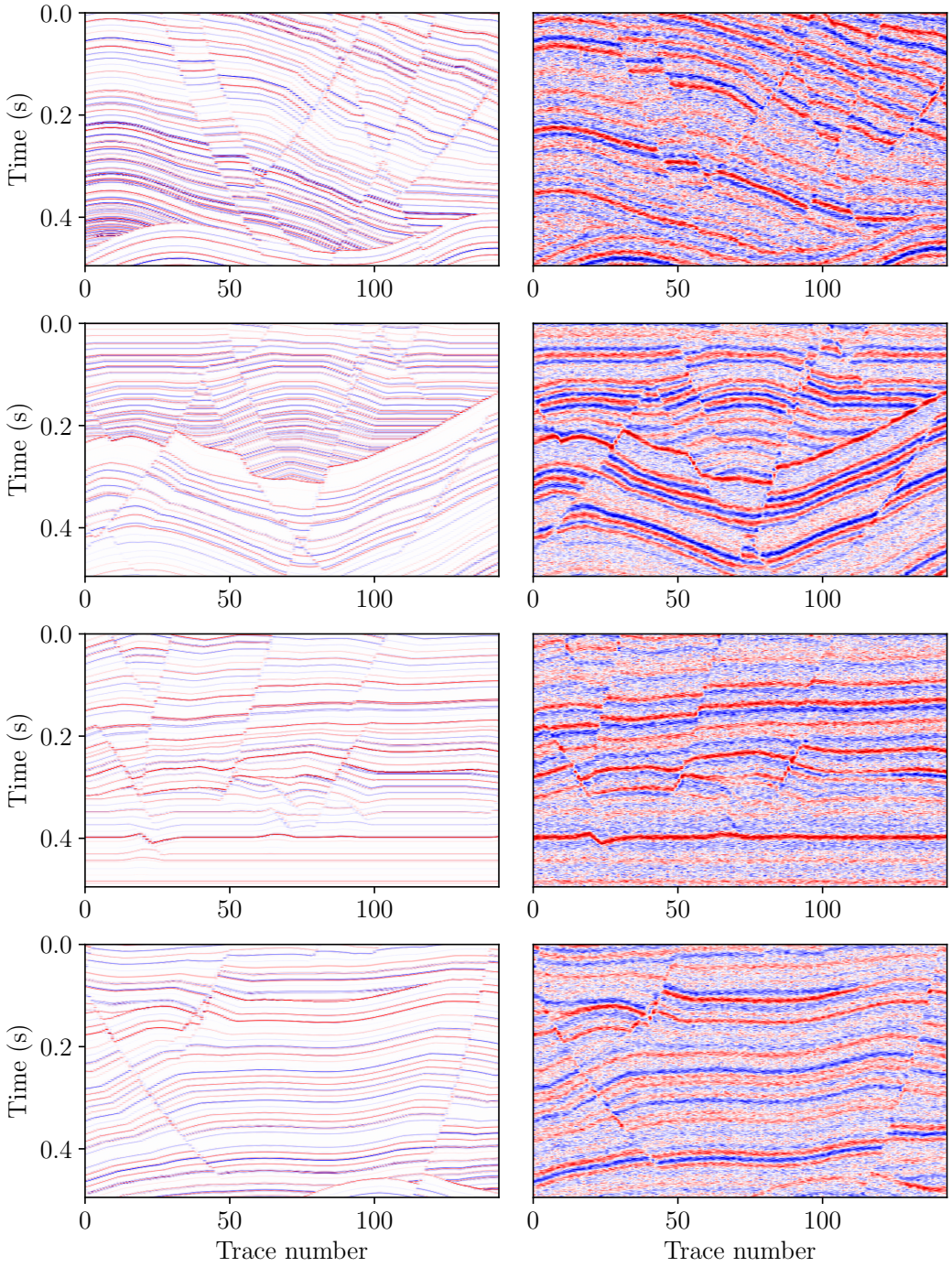

Figure 5.5: Multiple realizations of reflectivity models (left column) and noisy data (right column) used for training the estimator to perform 2D thin-bed reflectivity inversion.

Despite dealing with different geological settings, both experiments used the same reflectivity labels in the training data to test the generalization properties of the proposed estimator. We notice high-resolution results even when there is a significant shift in the data distribution.

Alberta foothills seismic data

The first dataset entails a 2D seismic section from Alberta, Canada, in the foothills of the Rocky Mountains (Chopra et al., 2009). It consists of 300 traces digitized in increments of 1 ms. Figure 5.6 shows the estimated wavelet and frequency spectrum. Figure 5.7

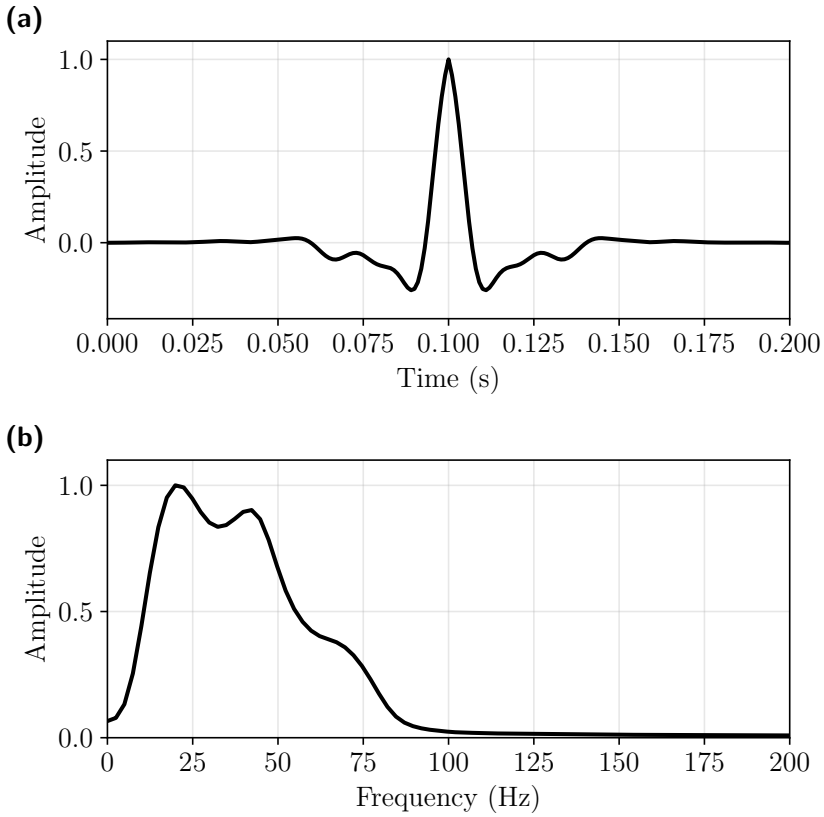

Figure 5.6: (a) Extracted source wavelet from the Alberta foothills seismic data. (b) Amplitude spectrum.

portrays the results, where for completeness, we also compare the proposed method to the output of a thin-bed spectral inversion (TSI) (the thin-bed reflectivity inversion method used here is commercially available as ThinMan) (Chopra et al., 2006) (Figure 5.7c) and the solution obtained with a learned post-processing residual U-net architecture (ResUnet) (Figure 5.7d) given by $\mathbf{m}^{\star}_{\text{ResUnet}} = (\mathbf{I} + \mathbf{U}_{\theta})(\mathbf{L}_k^{\dagger}\mathbf{d}_{\epsilon})$, in which $\mathbf{U}_{\theta}$ has the same design as

$\mathbf{N}_{\theta_2}$. For consistency, the ResUnet was trained by minimizing the MSE loss $\sum_i^N ||\mathbf{m}_i - \left(\mathbf{L}_k^\dagger \mathbf{d}_\epsilon^i - \mathbf{U}_\theta(\mathbf{L}_k^\dagger \mathbf{d}_\epsilon^i)\right)||_2^2$ using the same training data, number of epochs and learning rate as the proposed estimator.

No additional information (e.g., sonic logs) is available in the studied area, which limits the validation of the results. However, the deep decomposition output compares favourably with the conventional TSI solution, which provides confidence that the proposed technique has revealed substantial reflectivity information hidden in the original seismic section. Moreover, the predicted deconvolution output of the deep estimator shows more continuous reflectors and fewer noise artifacts than TSI due to an improvement in the lateral continuity and the explicit denoising component. At first glance, the deep decomposition prediction performs similarly to the ResUnet solution. Nonetheless, the residual data panel shown in Figure 5.7e presents almost no signal for the proposed solution. In other words, the re-convolution of the deep decomposition result with the wavelet is almost identical to the input data. It indicates the proposed deconvolution's ability to better preserve the overall solution's fidelity to the measured data. On the other hand, a moderate amount of residual error is visible for both ResUnet and TSI re-simulated outputs (Figures 5.7f and 5.7g). The degree of approximation shows clearly in the comparison of a representative middle trace in terms of the Pearson correlation coefficient (Figure 5.8)

The deep decomposition estimator offers versatility from the image perspective as we can decompose the prediction of the proposed deconvolution operator in the three panels shown in Figure 5.9 for visual inspection, according to equation 5.13. Figure 5.9a displays the minimum norm solution alone, anticipated to fail due to its lack of resolution. The inverted noise panel (see Figure 5.9b) mainly presents uncorrelated and band-limited noise. The predicted null space component (Figure 5.9c) exhibits the thin-beds and high-frequency features.

Lastly, Figure 5.10 compares the average power spectral density of the inverted reflectivities. We see that the three deconvolution methods yield an increase in the spectral content for both low and high frequencies compared to the input data, significantly improving the resolution.

Penobscot seismic data: a blind well test

We further test the deconvolution estimator in a third example comprising a 2D section of the Penobscot 3D survey (crossline number 1155), an offshore dataset recorded at a 4ms sampling rate. Figure 5.11 shows the statistically extracted wavelet and frequency spectrum. Figure 5.12 shows the predicted output and input data. In general, the deconvolved solution is superior if we keep in mind the quality of the input seismic data in terms of extra reflection

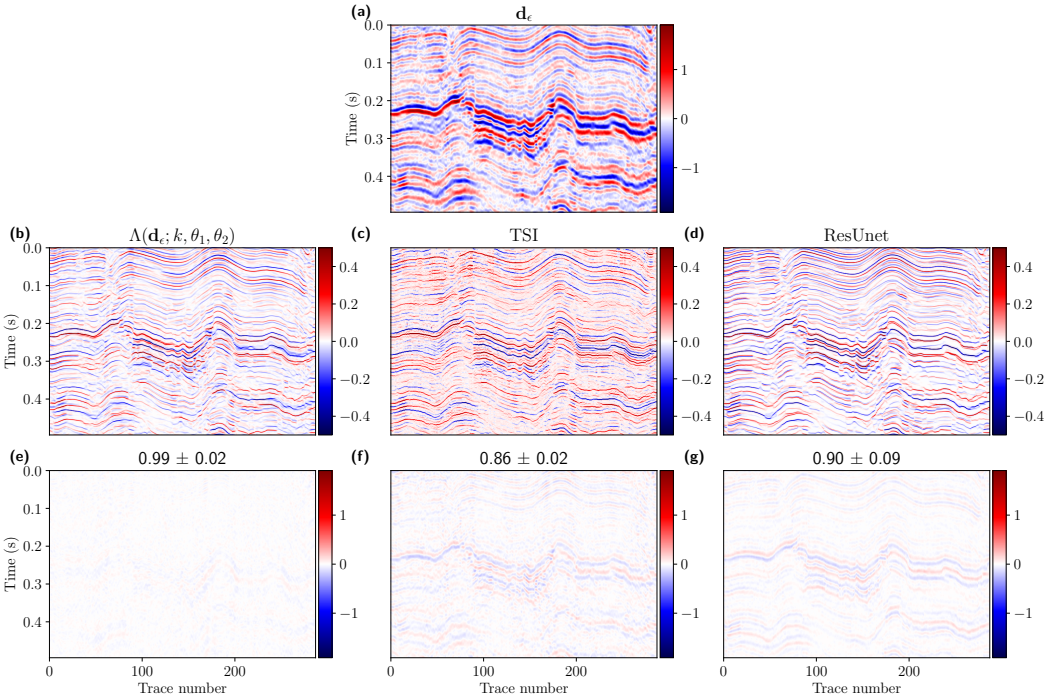

Figure 5.7: Alberta foothills results: (a) field data example portraying the input band-limited data $\mathbf{d}_\epsilon$, (b) full-band reflectivity inverted by the proposed method, (c) classical thin-bed reflectivity inversion via the TSI algorithm, (d) ResUnet result. Panels (e), (f) and (g) show the data residual $\mathbf{d}_\epsilon - \mathbf{L}\mathbf{m}^\star$, taking $\mathbf{m}^\star$ as the results in (a), (b) and (c), respectively, and the Pearson correlation coefficient (mean ± standard deviation) of the traces with respect to the input data on top.

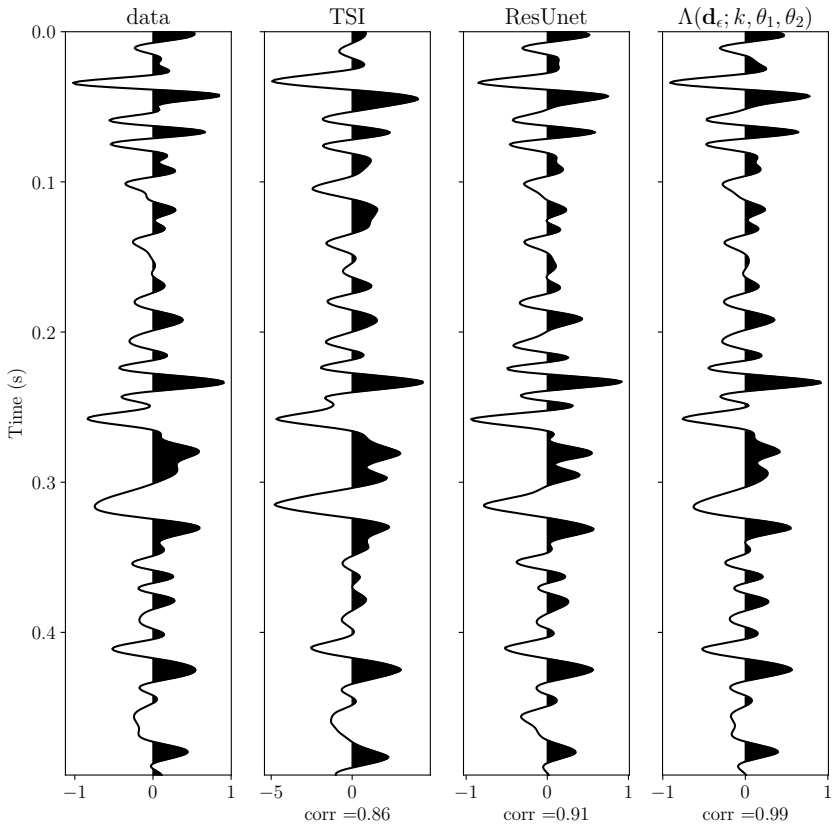

Figure 5.8: Representative trace from the Alberta foothills dataset showing the Pearson correlation coefficient (bottom) between the re-convolved results and the input data.

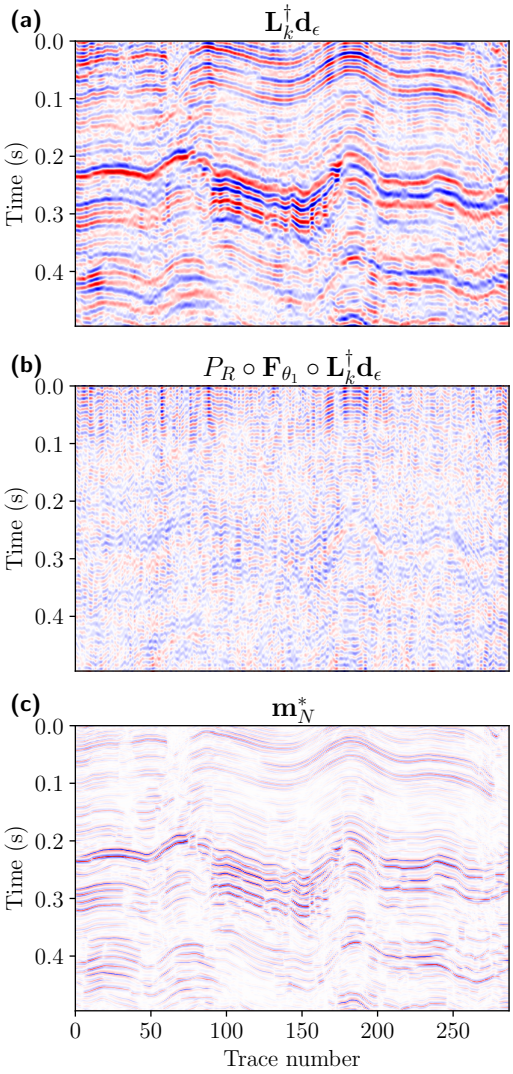

Figure 5.9: Individual components of the deep decomposition solution for the Alberta foothills dataset: (a) the band-limited solution obtained via TSVD ($\mathbf{L}_k^\dagger \mathbf{d}_\epsilon$), (b) residual noise leaking into the solution through the pseudo-inverse ($P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger \mathbf{d}_\epsilon$), and (c) the predicted null space component $P_N \circ \mathbf{N}_{\theta_2} \circ (\mathbf{L}^\dagger \mathbf{d}_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger \mathbf{d}_\epsilon)$.

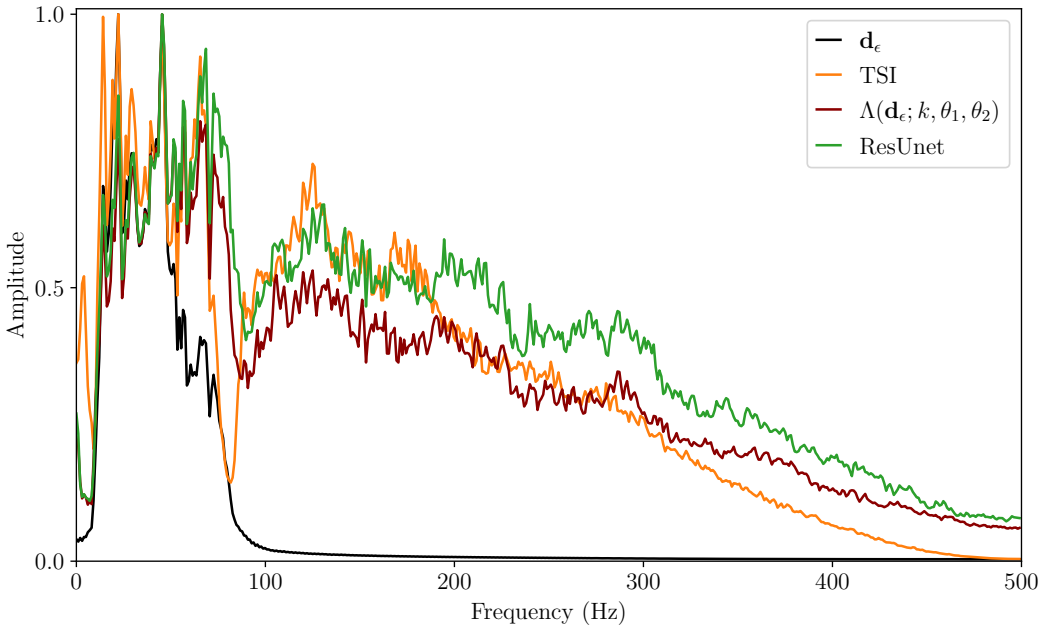

Figure 5.10: Normalized average power spectral density for the Aberta foothills data example and three deconvolution techniques for real data thin-bed reflectivity estimation: deep decomposition learning solution $\Lambda(\mathbf{d}_\epsilon; k, \theta_1, \theta_2)$, TSI solution (Chopra et al., 2006), and ResUnet.

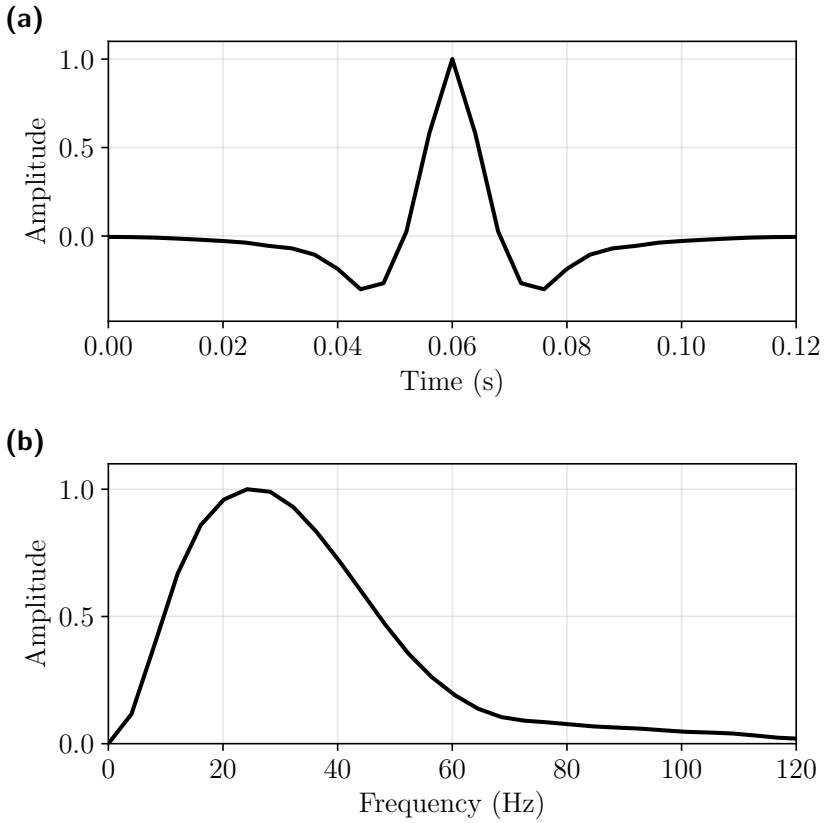

Figure 5.11: (a) Extracted source wavelet from the Penobscot seismic data. (b) Amplitude spectrum.

cycles and the fault detail. While there is an evident loss of detail in the data's deeper part (approximately after 2 seconds), the amplitudes and lateral coherence of the reflectors are well preserved throughout the deconvolved traces. Additionally, we notice the data's relative reliability since the spectrum's overall shape (see Figure 5.13) within the seismic main frequency band is approximately preserved and enhanced.

The studied area also contains one well log to compare the reflectivity profiles. We first calculate the impedance and reflectivity curves from the P-wave sonic and density logs. Then, the reflectivity is converted to the time domain using the P-wave log data. The final synthetic trace is obtained by convolving the well-log-derived reflectivity with the estimated wavelet. We observe a good correlation between the seismic sections (input data and the deconvolved result) and the synthetic seismogram, especially in key horizons. For more detail, Figure 5.14 shows a close-up view of the well-derived synthetic trace and the real data and predicted traces in the exact location of the well-log data.

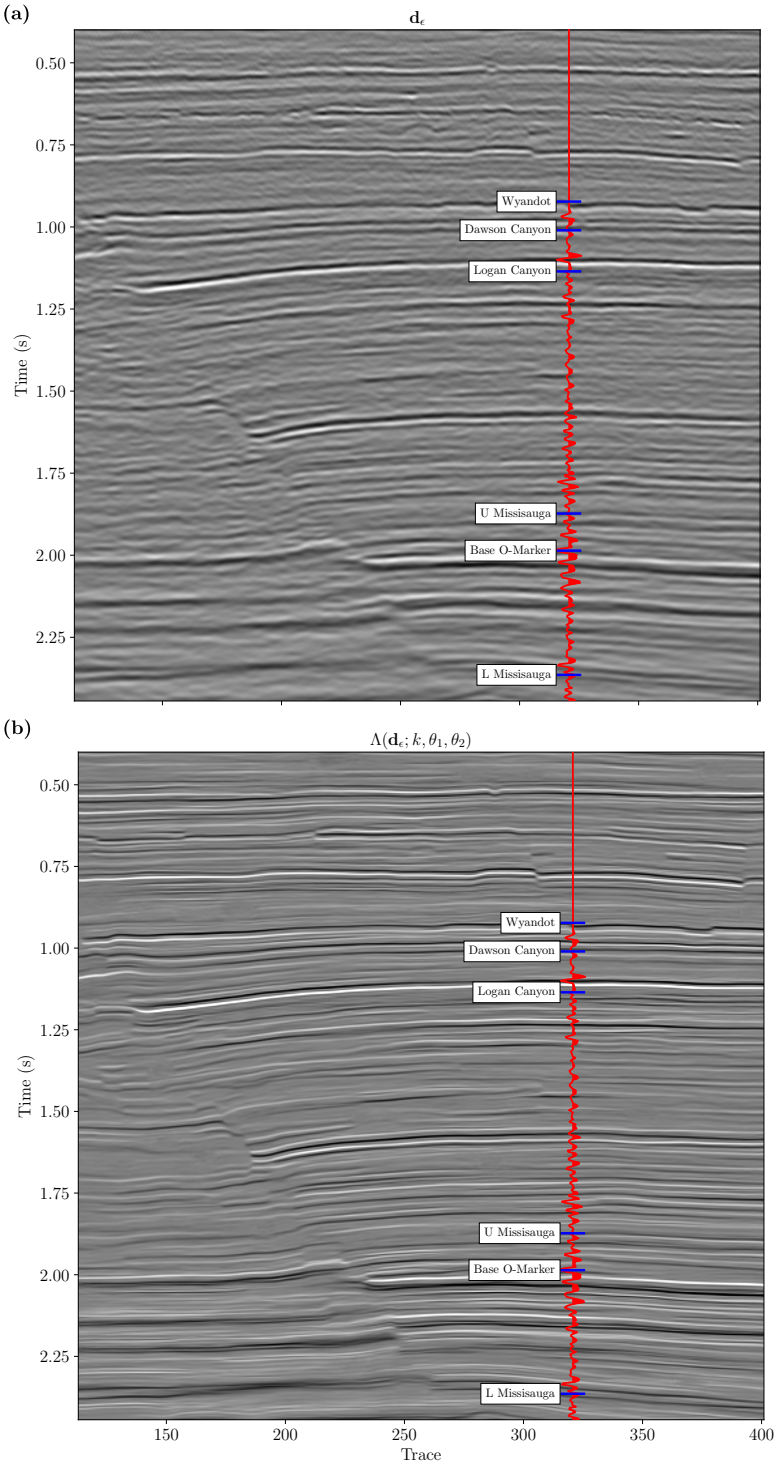

Figure 5.12: (a) Field data example from the Penobscot dataset. (b) Full-band reflectivity inverted by the proposed method.

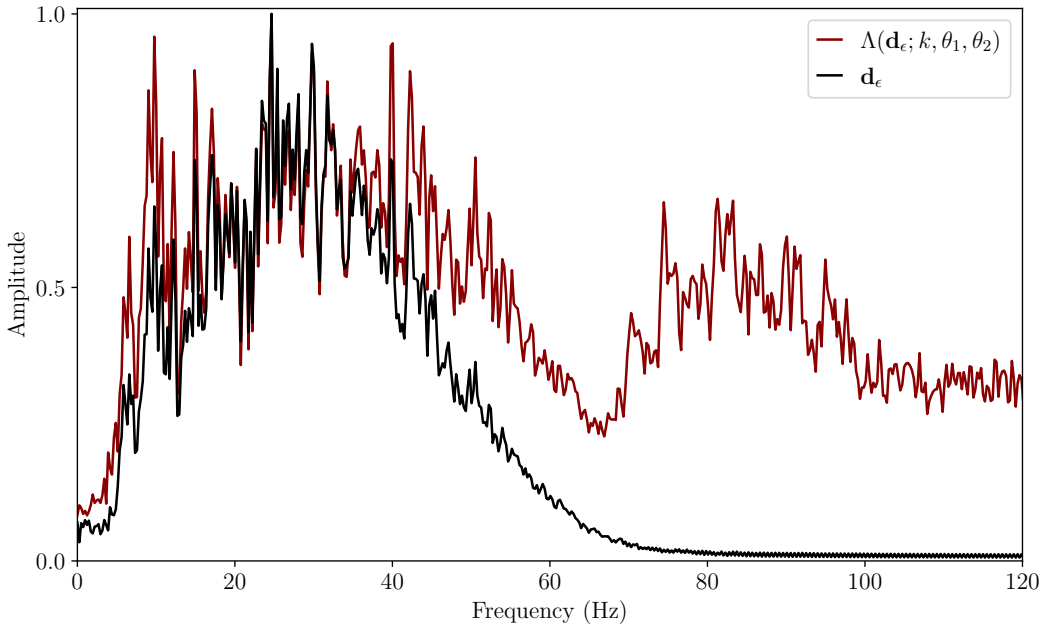

Figure 5.13: Normalized average power spectral density for the Penobscot data example and the decomposition learning solution $\Lambda(\mathbf{d}_\epsilon; k, \theta_1, \theta_2)$.

Furthermore, we validate the inverted results in Figure 5.15 by comparing the predicted reflectivity convolved with a known bandpass wavelet with the synthetic seismogram. In general, the predicted reflectivity recovers the most significant features. We observe an appropriate degree of agreement with the well-derived reflectivity despite not applying any stretch or squeeze of the well-log in this process. A point worth considering is that a perfect well-tie is impossible because we used a simplistic physics model that neglects more complex phenomena such as a time and space-variant source wavelet and a dispersive media (White and Hu, 1998).

Like the previous example, Figure 5.16 displays the individual elements of the deconvolved prediction in Figure 5.12b. Once again, the TSVD output (Figure 5.16a) presents the inevitable band-limited deconvolution and a general loss in resolution with depth. Figure 5.16b exhibits artifacts in the inverted noise panel, and Figure 5.16c presents thin-bed reflectors as the predicted null space components.

Uncertainty quantification

In the previous examples, the deconvolution problems were solved deterministically and did not provide any uncertainty on the obtained solution. Uncertainty estimates, however, may

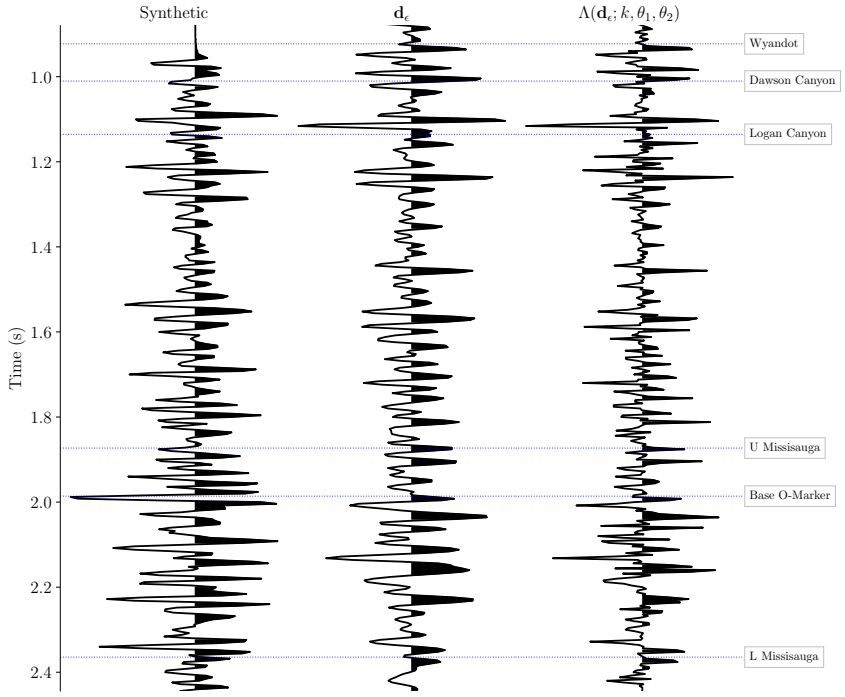

Figure 5.14: Close-up view of the Penobscot dataset. Results show the well-derived synthetic trace (left), the input measurements (middle) and the predicted result (right) for the trace where the well log data is located. Interpreted horizons are shown for reference.

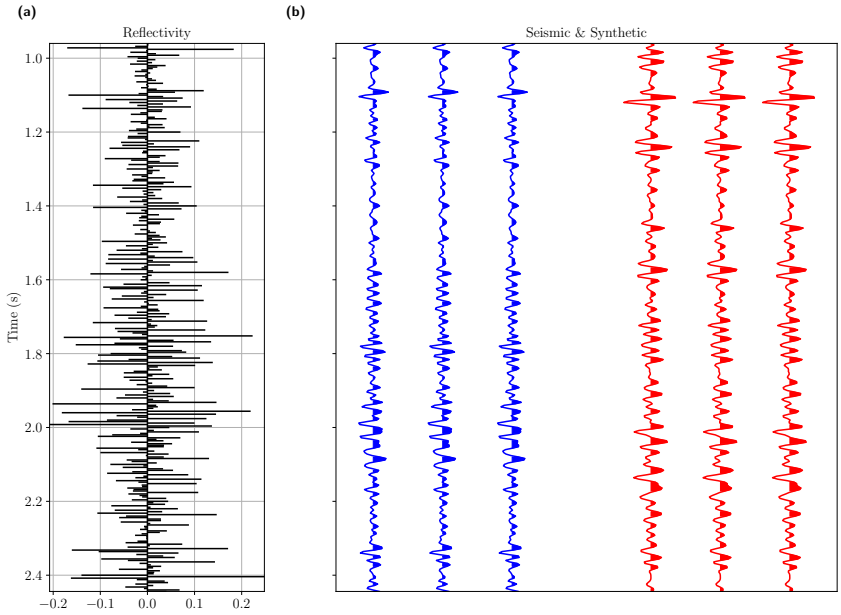

Figure 5.15: Validation panel for the Penobscot dataset. (a) well-log-derived reflectivity in the time domain. (b) Simulated data from the well-log-derived reflectivity in (a) (in blue) and simulated data from the predicted reflectivity (in red). The simulations result from convolving each reflectivity profile with a 60 Hz Ricker wavelet.

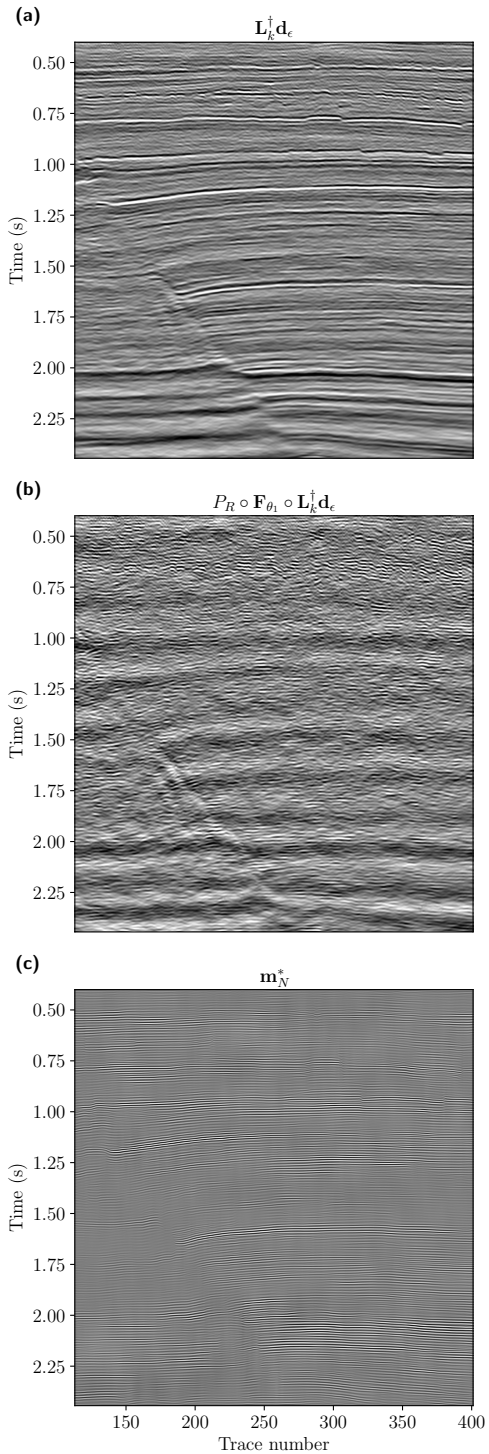

Figure 5.16: Individual components of the deep decomposition solution for the Penobscot dataset: (a) the band-limited solution obtained via TSVD ($\mathbf{L}_k^\dagger \mathbf{d}_\epsilon$), (b) residual noise leaking into the solution through the pseudo-inverse ($P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger \mathbf{d}_\epsilon$), and (c) the predicted null space component $P_N \circ \mathbf{N}_{\theta_2} \circ (\mathbf{L}^\dagger \mathbf{d}_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger \mathbf{d}_\epsilon)$.

give valuable insight into reflectivity estimation by revealing the reliability of the predictions, which might impact subsequent decision-making. In this section, we focus specifically on epistemic uncertainty quantification, which refers to the uncertainty associated with estimating the trainable parameters of the deep decomposition estimator conditioned on the training data in our neural network setting. Namely, several parameter configurations $\Theta = [\theta_1, \theta_2]$ may explain the data for a particular training dataset, rendering different predictions on the input seismic at inference time. Many methods have been proposed to encode the uncertainty of neural networks over the model parameters (Abdar et al., 2021), including Bayesian neural networks (BNNs) adopting variational inference to approximate Bayesian statistics. BNNs usually double the number of parameters per layer compared to their non-Bayesian counterparts since each trainable weight is replaced by a trainable Gaussian weight distribution parametrized by its standard deviation and mean. The high parameter dimensionality of BNNs leads to slow convergence in training which might hurt performance.

As an efficient alternative that does not increase the number of parameters and requires minimal modification of the original estimator without retraining, we implement Monte Carlo Dropout (MC dropout). Unlike regular dropout regularization that only applies dropout layers at training time, MC dropout activates dropout during test time to sample from the posterior parameter distribution. Gal and Ghahramani (2016) show the connection between dropout and approximate inference in a Gaussian process and propose MC dropout to approximate the exact posterior inference relying on Bernoulli distributed weights.

Concretely, MC dropout utilizes layers $\mathbf{W}_i^{\text{do}} \in \Theta$ defined as

$$\mathbf{W}_i^{\text{do}} = \mathbf{W}_i \text{diag}(\mathbf{z}_i), \tag{5.20}$$

where $\mathbf{W}_i$ is the weight matrix for each i-th convolutional layer before dropout is applied, $\text{diag}(\mathbf{z}_i)$ is the randomly generated dropout mask, and $\mathbf{z}_i \sim \text{Bernoulli}(p_i)$ are the randomly activated coefficients with dropout probability p_i, which can be fixed or set as a learnable parameter. We then perform Monte Carlo integration to approximate an empirical unbiased estimator

$$\hat{\mathbb{E}}(\mathbf{m}) \approx \frac{1}{T} \sum_{t=1}^{T} \Lambda(\mathbf{d}_\epsilon; k, \hat{\Theta}_t) \tag{5.21}$$

$$\approx \frac{1}{T} \sum_{t=1}^{T} \Lambda(\mathbf{d}_\epsilon; k, \hat{\theta}_{1_t}, \hat{\theta}_{2_t}), \tag{5.22}$$

such that $\Lambda(\mathbf{d}_\epsilon; k, \hat{\theta}_{1_t}, \hat{\theta}_{2_t})$ are realizations of independent draws of random dropout masks for the same estimator Λ. In practice, equation 5.22 is equivalent to running T stochastic

forward passes through the trained estimator, all with the same input, and computing the mean of the dropout realizations (Gal and Ghahramani, 2016). Finally, we use the pixel-wise variance to indicate instabilities in the reconstruction for a particular input, which contains information about the uncertainty of the model parameters

$$\hat{\sigma}^2(\mathbf{m}) = \frac{1}{T}\sum_{t=1}^{T}(\Lambda(\mathbf{d}_\epsilon; k, \hat{\theta}_{1_t}, \hat{\theta}_{2_t}) - \hat{\mathbb{E}}(\mathbf{m})) \odot (\Lambda(\mathbf{d}_\epsilon; k, \hat{\theta}_{1_t}, \hat{\theta}_{2_t}) - \hat{\mathbb{E}}(\mathbf{m})), \tag{5.23}$$

where $\odot$ represents element-wise multiplication. This strategy is similar in spirit to the one proposed by Velis (2008) for quantifying the uncertainty of stochastic inversion in sparse deconvolution via fast simulated annealing.

In this example, we use the Penobscot dataset from the previous section and set $p_i = 20\%$ and $T = 200$. Figures 5.17a and 5.17b show the averaged solution and the pixel-wise standard deviation of the inferred reflectivities, respectively. To reduce the bias towards strong reflectors in the estimated uncertainty, Figure 5.17c shows the normalized standard deviation by dividing the point-wise standard deviation estimate by the envelope of the mean (Siahkoohi et al., 2022a). Since this real dataset is an out-of-distribution sample, we notice relatively high model uncertainty in all model regions. The estimator can also express an increased uncertainty in the deeper parts of the studied area, which corresponds to partially imaged reflectors and less resolvable features in the input data due to amplitude loss. Ultimately, this result allows us to flag unreliably reconstructed reflectors and quantify seismic structural uncertainty. In addition, due to the data consistency property of the deep decomposition estimator, multiple realizations from the deep null space network can be used to stochastically explore the null space of the forward operator and obtain a family of solutions that honour the data. However, the uncertainty maps will be partially biased since the approximated posterior still depends on the training dataset used to extract the prior information to solve the inverse problem and other hyper-parameters such as the dropout rate. Changing these variables could significantly modify the calculated uncertainty.

5.4 Discussion and future work

Some points deserve special consideration when using the proposed technique to deconvolve noisy seismograms. First, since we adopt the convolutional model for the seismogram, it is required that initial processing recovers amplitudes and removes multiples as well as possible. Surface-consistent deconvolution should also be implemented, allowing us to assume an invariant wavelet across traces.

Due to the necessity of constructing the forward operator from the test data, deconvolution via deep decomposition can be classified as an acquisition-restricted learning method. In

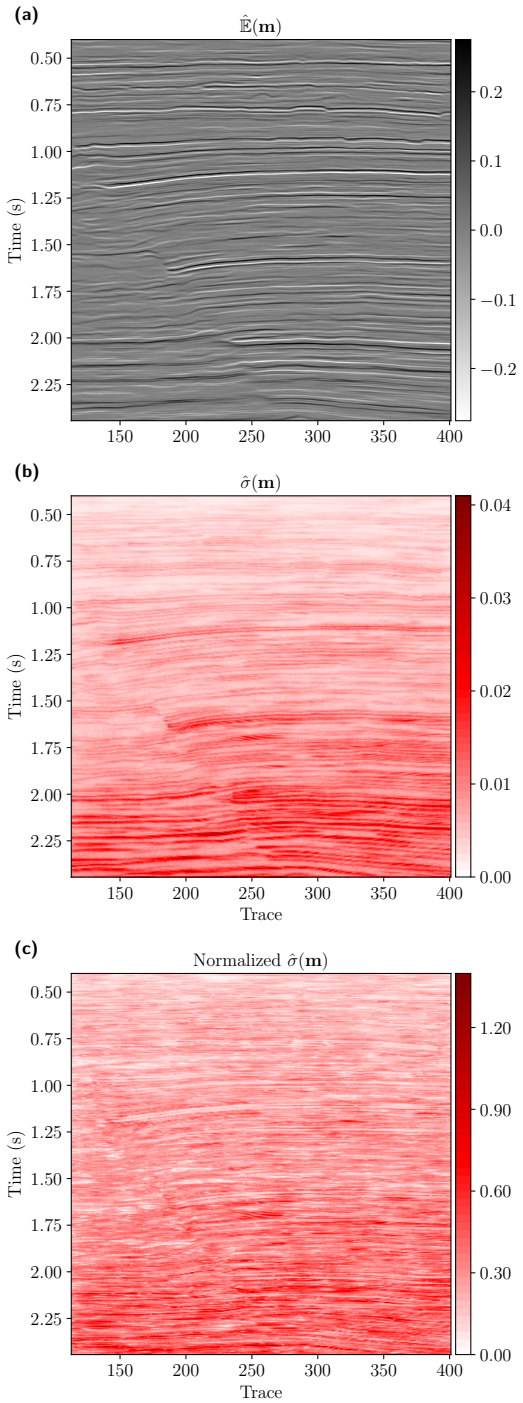

Figure 5.17: (a) Estimated mean after MC dropout running 200 forward passes through the deep estimator for the Penobscot field data. (b) Point-wise standard deviation among samples. (c) Normalized point-wise standard deviation by the estimated mean.

other words, because of the dependency on the forward operator and its null space components, the deep estimator needs to be trained every time the signature of the estimated source wavelet changes. This does not represent a disadvantage in practice since SVD is only calculated once before training, which renders the algorithm's training phase relatively cheap, depending on the number of training samples. Transfer learning could potentially ease the computational burden of the training stage, but the success of such an approach needs further investigation and escapes the scope of this chapter. Likewise, the direct computation of the projection operators is intractable for large-scale problems. Iterative solvers such as the conjugate gradient method can help calculate approximations to these projections, for example, by solving the linear system

$$P_N(\mathbf{x}) \approx \arg\min_{\mathbf{m}} ||\mathbf{m} - \mathbf{x}||_2^2 \quad \text{s.t.} \quad \mathbf{L}\mathbf{m} = 0, \tag{5.24}$$

with $\mathbf{x}$ as an initial guess model. Another option is to train a neural network to approximate the projection operators as proposed in Kuo et al. (2022).

l_p norms can be easily incorporated into the proposed training loss to investigate the use of explicit sparsity-promoting norms in combination with the deep decomposition deconvolution operator to measure the resolving power in sparse reflectivity inversion. While we only consider white Gaussian noise, significant non-Gaussian noise may appear as low-rank components, affecting the proposed method's performance. In this case, the denoising element can be adapted to give $\mathbf{F}_{\theta_1}$ a broader range to annihilate non-Gaussian and correlated noise.

Post-processing regularization via null space filtering is not new to the geophysics community. Deal and Nolet (1996) proposed a similar method, coined null space shuttles, in which desired features are imposed on an initial inverted solution by applying non-linear filters in the null space to preserve the observed data. Variations of this method include successful applications to tomography (de Wit et al., 2012; Osypov et al., 2013) and full waveform inversion (Keating and Innanen, 2021). The most significant differences between the method of Deal and Nolet (1996) and the proposed approach relate to the treatment of the filter design. Compared to the handcrafted filtering in the null space shuttles method, we make no explicit a priori assumptions about the properties of the solution. Instead, we use only reliable data in combination with the additional knowledge learned by the network from the training samples. Thus, deconvolution via deep decomposition can be considered a data-driven version of the null space shuttle approach.

In the proposed method, our constraints are based on knowledge of the physical process embedded in the forward operator, so we dispense the use of wells to prevent adding bias at an early stage (Li et al., 2022), although having a single well control point might be helpful for wavelet extraction. In the second example, we use well-log data for quality control but

not in the inversion process. Once high-resolution seismic is obtained using the suggested approach, conventional inversion employing well-log data can be performed. Downton et al. (2020) show that using a small number of well-logs and a physics-based data augmentation technique can create multiple realizations of area-specific data labels. Future research should include additional well-log data information in the estimator training process to guide the reconstruction of the missing frequencies.

For the single-channel deconvolution example, the proposed method can accurately recover the low-frequency components of the reflectivity belonging to the null space (see amplitude spectrum shown in Figures 4a and 4b). In this case, the forward operator is accurate, and the noise level is known. We attribute the more substantial boost of high frequencies to the fact that these conditions are not completely met in the two real data scenarios, which hinders the reconstruction of a perfectly flattened spectrum. A low-frequency component leakage occurs on the estimated negative inverted noise panels (Figures 9b and 16b), which are subtracted to approximate the ideal reconstructions. However, in the presented real-case scenarios, the lower part of the reconstructed reflectivity spectrum still improves, allowing the technique to produce valid thin bed reflectivity results. More testing should be done to boost the inverted low-frequency components by fine-tuning the regularization parameter from the TSVD solution or the noise suppression to avoid the low-frequency leakage into the subtracted component. Also, an extension of the method could contemplate accepting an initial low-frequency model as input to boost the deep estimator prediction on this part of the spectra. This approach has been exploited in other deep learning-based seismic inversion techniques such as Kaur et al. (2020), Downton et al. (2020) and Wu et al. (2021a), but exceeds the scope of this chapter. Initial low-frequency models can be constructed from smoothed well log information, smoothed velocity fields, prior geologic information or more complicated bandwidth extension frameworks (e.g. Gholami and Sacchi, 2013; Lesage et al., 2015; Bianchin et al., 2019).

Recently, different authors have proposed deep-learning-based reflectivity and impedance inversion techniques. Chai et al. (2021b) propose an end-to-end sparse-spike inversion algorithm based on 2D and 3D convolutional neural networks (CNN), showing that multi-trace filters outperform results obtained by single-channel methods. Unlike our physics-guided approach, they only extract the wavelet from the signal to generate the input-output training pairs but do not integrate the physics into the training stage. Similarly, Wu et al. (2021b) introduce an impedance inversion technique using a 2D CNN with a weakly supervised training loss that requires multiple well-logs from the same field data. While this approach allows explicitly incorporating prior knowledge extracted from well-log data and an initial low-frequency model to guide the prediction, it is too dependent on the number and distribution of wells to build the training dataset. Moreover, this approach also suffers the limitations of fully learned end-to-end approaches due to the lack of a physics-

driven component. On the other hand, Chen et al. (2022) implement a regularized deep image prior (DIP) Ulyanov et al. (2018) framework for inverting a high-resolution model by parametrizing the reflectivity with the trainable weights of a neural network. Since DIP is an unsupervised technique, it enforces data consistency by design while minimizing the data misfit. DIP only uses corrupted data in the reconstruction process and purely relies on the CNN architecture to act as prior. In Chen et al. (2022), the training stage and the optimization problem are the same, which might demand many iterations. Our approach eliminates iterations and the need for a specific CNN architecture to exploit correlations in the data to learn their inner structure, which results in a more efficient solution. Finally, Chen et al. (2021b) and Chen et al. (2021c) propose supervised and semi-supervised blind methods, respectively, in which they simultaneously invert the seismic wavelet and a high-resolution post-stack model. In these physics-guided iterative approaches, the prior is implicitly learned by trainable proximal operators that mimic an unrolled non-linear solver instead of explicitly exploring the null space of the forward operator. Mainly, Chen et al. (2021c) designed a semi-supervised data-consistent loss incorporating well log information as a regularization term, which can be adapted to extend our method by reducing the dependence on synthetic labels and will be investigated in a future report. Future work will test the proposed estimator on additional seismic inverse problems and compare it with other regularization methods. Also, a thorough numerical comparison between our method and other deep learning techniques will be conducted in upcoming research.

One characteristic of the proposed deconvolution method is that it is possible to estimate the prediction uncertainty of the inversion by performing several forward passes using different weight seeds via MC dropout. While easy to train and compute, MC dropout is a simplistic approximation of Bayesian variational inference. The quality of the uncertainty estimation depends on the appropriateness of the dropout as a Bayesian approximation for the given model and problem. In this regard, future work should explore reflectivity inversion via deep decomposition integrating uncertainty quantification approaches that may be more efficient and robust to out-of-distribution and noisy data. For example, recent research (Siahkoohi et al., 2022b) advocates for invertible neural networks leveraging (conditional) normalizing flows to parameterize surrogate conditional distributions and enable efficient variational inference. As in all uncertainty quantification techniques, accuracy and computational efficiency have a trade-off. Explaining this trade-off for the presented reflectivity inversion problem is a question of ongoing research. Future work should also adapt the proposed framework to tolerate wavelet phase inaccuracies, extensions to 3D and a forward modelling operator considering a more complex physics framework.

5.5 Conclusions

In this work, we investigated regularizing networks via the deep decomposition approach to deal with the always-desirable goal of estimating full-band reflectivity from band-limited seismic data. We illustrate how learned null space regularization adds reasonable estimates from the null space, improving classic regularization solutions. Additionally, we combined the deep decomposition learning method with TSVD, which helps produce clean inputs to train the null space network efficiently. As a result, we produce an approximate high-resolution deconvolution operator for a specific and predetermined forward operator. Despite this limitation, the enhancement in resolution caused by the null space network formulation may be key to achieving widespread adoption of deep learning in full-band reflectivity estimation. We demonstrate the method's effectiveness in multiple datasets and obtained reflectivity sections with no spurious artifacts and good lateral continuity. The inversion of 2D field datasets illustrates that the algorithm works in the presence of noise and is practical. An important research direction is identifying ways of efficiently integrating deep null space regularization with more significant problems where the direct computation of the orthogonal projections is prohibitively expensive.

CHAPTER 6

Conclusions

6.1 Summary

The main challenge in solving seismic imaging problems is that, without any prior assumptions, recovering an accurate representation of the true reflectivity model from the observed reflection data is usually unsatisfactory—even when the forward model is perfectly known. This thesis explored how different handcrafted and learned regularization and preconditioning methodologies can aid and improve model-based inversion schemes.

Two common issues of LSRTM are the many iterations required to produce substantial subsurface imaging improvements and the difficulty of choosing adequate regularization strategies with optimal hyper-parameters. In Chapter 2, we investigated how supervised learning can mitigate these shortcomings by solving the LSRTM problem through an iterative deep-learning framework inspired by the projected gradient descent algorithm. In particular, we develop an image-to-image approach interlacing the gradient steps at each iteration with blocks of residual CNNs that capture the prior information in the training phase. By including the least-squares data-misfit gradient in the learning process, we force the solution to comply with the observed seismic data. At the same time, the CNN projections implicitly account for the regularization effects that lead to high-resolution reflectivity updates. After training with 900 randomly generated instances, our network ensemble can estimate accurate reflectivity distributions in only a few iterations. To demonstrate the effectiveness and generalization properties of the method, we consider three synthetic cases: a folded and faulted model, the Marmousi model, and the Sigsbee2a model. We empirically find that it is possible to obtain an improved reflectivity model for out-of-distribution instances by using the learned reconstructions as warm starts for the conjugate gradient

algorithm and bridging the gap between the learned and conventional LSRTM schemes. Finally, we apply the proposed network with transfer learning on a 2D towed-streamer Gulf of Mexico field data set, producing high-resolution images comparable to traditional LSRTM but drastically reducing the required number of iterations.

Identifying optimal preconditioners in seismic inversion is not straightforward; it demands expertise and extensive experimentation. Chapter 3 presents two approaches that combine deep-learning-based preconditioning techniques in data and model spaces to accelerate iterative pre-stack LSRTM and improve non-iterative pre-stack imaging, respectively. The first method employs a deep autoencoder formulation that learns a mapping between shot-indexed migrated and re-migrated images, enabling it to estimate an approximation of the inverse Hessian. This process facilitates the re-parameterization of the reflectivity model into a low-dimensional space through the autoencoder's dimensionality reduction capabilities. It allows linearized waveform inversion to be solved more efficiently. As a result, the convergence of LSRTM is expedited, significantly reducing artifacts and achieving plausible high-resolution imaging in fewer iterations. These benefits are demonstrated in trials with synthetic data where sparse shots are inverted. The second method relies on a non-iterative approach to improve RTM images by training a CNN on input-output pairs of observed and demigrated data, computing nonlinear filters that precondition the migrated data, enhancing the focusing capability and amplitude balance, particularly in under-illuminated regions. Our method's effectiveness was confirmed through tests conducted on the Marmousi data set, showing improvements in amplitude balance and reduced artifacts compared to traditional RTM outputs. Both preconditioning frameworks sidestep the need for paired samples of ground-truth labels and initial reconstruction models, which are commonly required in other supervised deep-learning techniques. The training stage avoids complicated pre-processing and demands minimal user interaction, enhancing user-friendliness and cost-effectiveness. We simulate the effects of the Hessian operator at the cost of only one migration/re-migration sequence, equivalent to one iteration of conjugate gradients.

Incorporating overturned waves and multiples in seismic imaging is one of the most plausible means by which imaging results might be improved, particularly in regions of complex subsurface structures such as salt bodies. Chapter 4 presents a study on FWLSRTM, an advanced subsurface imaging technique capable of inverting wave modes beyond single scattering, enhanced by sparse regularization. We discuss the limitations of traditional LSRTM, which relies on first-order scattering approximations and is hindered by insufficient data processing, limited physical operators and noise. The new FWLSRTM method overcomes these challenges by utilizing a nonlinear inversion approach that integrates full-wavefield vector reflectivity modelling with a sparse regularization term, improving data fidelity and reducing artifacts. This chapter outlines the theoretical foundations, including the mathematical

modelling and equations used in the new method. It also provides detailed results from numerical experiments that compare the new technique against traditional methods using different models like high-density box and Marmousi2 models. The results demonstrate that sparse FWLSRTM significantly enhances the resolution and clarity of subsurface imaging, better delineate geological layers, and is more resilient to the inaccuracies in velocity models commonly encountered in complex geological settings. The study highlights the advantages of the nonlinear and sparse regularization approaches in FWLSRTM, a robust tool for seismic imaging that can handle the complexities of real-world geological data more effectively than traditional methods.

Chapter 5 explored seismic broadband deconvolution with regularization using the deep decomposition technique to derive full-band reflectivity from band-limited seismic data. We demonstrate how incorporating learned null space regularization provides reasonable estimates from the null space, thereby enhancing traditional regularization methods. Furthermore, we integrated the deep decomposition learning method with TSVD, facilitating the generation of clean inputs for efficiently training the null space network. Consequently, we developed an approximate high-resolution deconvolution operator tailored to a specific and predefined forward operator constructed with the seismic wavelet. Although this presents a relative limitation in that the high-resolution deconvolution operator needs to be trained for each seismic survey, the improvement in resolution achieved through the null space network approach could be crucial for the broader acceptance of deep learning in full-band reflectivity estimation. We validated the method's efficacy across various datasets, achieving reflectivity sections free of spurious artifacts and characterized by strong lateral continuity. The successful inversion of 2D field datasets confirms that the algorithm remains effective amidst noise, proving its practicality. A vital avenue for future research is exploring efficient ways to integrate deep null space regularization into larger-scale problems where direct computation of orthogonal projections is impractically costly, such as in LSRTM, FWLSRTM and FWI.

6.2 Future research

As widely stressed in all geophysical literature, seismic inverse problems are ill-posed, meaning that an infinite number of models can match the observed data. Throughout this thesis, we have predominantly utilized deterministic methods to address seismic imaging problems due to computational restrictions, except for employing a surrogate Bayesian approach in the latter part of Chapter 5 to evaluate model uncertainty. This approach highlights a pathway toward integrating stochastic methods, which might be further explored to enhance model accuracy and reliability through uncertainty quantification. We acknowledge

the necessity of exploring more complex, statistically robust techniques within rigorous statistical inference frameworks to quantify uncertainty effectively. This is particularly vital as methods evolve and computational feasibility improves, allowing for more sophisticated Bayesian approaches to be integrated into mainstream applications of wave-equation-based seismic imaging techniques.

Despite the advancements presented in Chapter 3 of the thesis, we identify several challenges that warrant discussion. For example, a current limitation is the highly nonlinear mapping from latent space to reflectivity model space, which complicates gradient-based optimization due to prevalent local minima issues. As an alternative, we suggest global optimization and stochastic sampling of the latent space, which might offer more robust solutions, albeit at a higher computational cost. The potential application of generative neural networks for sampling prior models through variational inference methods presents a promising area for future research. Moreover, the dimension of the latent space in both data and model space preconditioning techniques is selected arbitrarily and might not accurately reflect the actual dimension of the posterior. Currently, we depend on heuristic approaches instead of systematic methods to estimate the data manifold's latent dimension.

Besides effectively accounting for uncertainty in the model space, seismic inversion might also benefit from multiparameter approaches. Exploring subsurface structures using P and S wave data in inversion algorithms promises significant advancements. By leveraging these wave modes' unique propagation characteristics and sensitivities, particularly the distinct response of shear waves to anisotropy, we could achieve a more detailed and comprehensive understanding of subsurface conditions. Multiparameter inversion in LSRTM mitigates the limitations posed by poor illumination from the single P wave type inversion and enhances the overall resolution and accuracy of the seismic imaging. Extensions of the presented techniques to multiparameter inversion are an exciting path for future research.

In practice, the precision of velocity models plays a critical role in LSRTM-like inversion. Furthermore, an accurate velocity model's role is also evident, especially in detailed lithology assessment techniques such as AVO analysis for reservoir characterization. We underscore the importance of ensuring the accuracy of the background model, as errors can significantly affect the outcomes of linear waveform inversion techniques and subsequent geological interpretations. Thus, when the velocity model requires refinement, the next logical step following linearized waveform inversion is to pursue joint or simultaneous inversion of both the velocity and reflectivity models. Therefore, the techniques presented in the thesis should be explored and adapted to scenarios where velocity is also unknown. This adaptation would enable a more dynamic and comprehensive approach to seismic imaging, allowing for simultaneous refinement and adjustment of the reflectivity and velocity models based on the seismic data.

Bibliography

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, 2015, TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abdar, M., F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, 2021, A review of uncertainty quantification in deep learning: Techniques, applications and challenges: Information Fusion, **76**, 243–297.

Adler, A., M. Araya-Polo, and T. Poggio, 2021a, Deep learning for seismic inverse problems: Toward the acceleration of geophysical analysis workflows: IEEE Signal Processing Magazine, **38**, 89–119.

——— 2021b, Deep learning for seismic inverse problems: Toward the acceleration of geophysical analysis workflows: IEEE Signal Processing Magazine, **38**, 89–119.

Adler, J. and O. Öktem, 2017, Solving ill-posed inverse problems using iterative deep neural networks: Inverse Problems, **33**, 124007.

Aki, K. and P. G. Richards, 2002, Quantitative seismology.

Alfarraj, M. and G. AlRegib, 2019, Semi-supervised learning for acoustic impedance inversion, *in* SEG Technical Program Expanded Abstracts 2019, 2298–2302. Society of Exploration Geophysicists.

Anagaw, A. Y. and M. D. Sacchi, 2012, Edge-preserving seismic imaging using the total variation method: Journal of Geophysics and Engineering, **9**, 138–146.

Antun, V., F. Renna, C. Poon, B. Adcock, and A. C. Hansen, 2020, On instabilities of deep learning in image reconstruction and the potential costs of ai: Proceedings of the National Academy of Sciences, **117**, 30088–30095.

Aoki, N. and G. T. Schuster, 2009, Fast least-squares migration with a deblurring filter:

Geophysics, **74**, WCA83–WCA93.

Araya-Polo, M., S. Farris, and M. Florez, 2019, Deep learning-driven velocity model building workflow: The Leading Edge, **38**, 872a1–872a9.

Araya-Polo, M., J. Jennings, A. Adler, and T. Dahlke, 2018, Deep-learning tomography: The Leading Edge, **37**, 58–66.

Asim, M., M. Daniels, O. Leong, A. Ahmed, and P. Hand, 2020, Invertible generative models for inverse problems: mitigating representation error and dataset bias: International Conference on Machine Learning, 399–409.

Baysal, E., D. D. Kosloff, and J. W. Sherwood, 1983, Reverse time migration: Geophysics, **48**, 1514–1524.

Beck, A. and M. Teboulle, 2009, A fast iterative shrinkage-thresholding algorithm for linear inverse problems: SIAM Journal on Imaging Sciences, **2**, 183–202.

Becker, S., L. Horesh, A. Aravkin, and S. Zhuk, 2015, General optimization framework for robust and regularized 3d full waveform inversion: Presented at the 77th Annual International Conference and Exhibition.

Berkhout, A. J., 1977, Least-squares inverse filtering and wavelet deconvolution: Geophysics, **42**, 1369–1383.

Bianchin, L., E. Forte, and M. Pipan, 2019, Acoustic impedance estimation from combined harmonic reconstruction and interval velocity: Geophysics, **84**, R385–R400.

Biswas, R., M. K. Sen, V. Das, and T. Mukerji, 2019, Prestack and poststack inversion using a physics-guided convolutional neural network: Interpretation, **7**, SE161–SE174.

Boink, Y. E., C. Brune, and S. Manohar, 2019, Robustness of a partially learned photoacoustic reconstruction algorithm: Photons Plus Ultrasound: Imaging and Sensing 2019, 88 – 94, SPIE.

Bora, A., A. Jalal, E. Price, and A. G. Dimakis, 2017, Compressed sensing using generative models: International Conference on Machine Learning, 537–546.

Buur, J. and T. Kühnel, 2008, Salt interpretation enabled by reverse-time migration: Geophysics, **73**, VE211–VE216.

Calvetti, D., S. Morigi, L. Reichel, and F. Sgallari, 2000, Tikhonov regularization and the l-curve for large discrete ill-posed problems: Journal of Computational and Applied Mathematics, **123**, 423 – 446. Numerical Analysis 2000. Vol. III: Linear Algebra.

Chai, X., G. Tang, K. Lin, Z. Yan, H. Gu, R. Peng, X. Sun, and W. Cao, 2021a, Deep learning for multitrace sparse-spike deconvolution: Geophysics, **86**, V207–V218.

——— 2021b, Deep learning for multitrace sparse-spike deconvolutiondl for multitrace sparse deconvolution: Geophysics, **86**, V207–V218.

Chang, J. R., C.-L. Li, B. Póczos, B. Vijaya Kumar, and A. C. Sankaranarayanan, 2017, One network to solve them all — solving linear inverse problems using deep projection models: 2017 IEEE International Conference on Computer Vision (ICCV), 5889–5898.

Chen, D. and M. E. Davies, 2020, Deep decomposition learning for inverse imaging problems: Presented at the Proceedings of the European Conference on Computer Vision (ECCV).

Chen, H., J. Gao, X. Jiang, Z. Gao, and W. Zhang, 2021a, Optimization-inspired deep learning high-resolution inversion for seismic data: Geophysics, **86**, R265–R276.

——— 2021b, Optimization-inspired deep learning high-resolution inversion for seismic data: Geophysics, **86**, R265–R276.

Chen, H., J. Gao, W. Zhang, and P. Yang, 2021c, Seismic acoustic impedance inversion via optimization-inspired semisupervised deep learning: IEEE Transactions on Geoscience and Remote Sensing, **60**, 1–11.

Chen, H., M. Sacchi, H. H. Lari, J. Gao, and X. Jiang, 2023, Nonstationary seismic reflectivity inversion based on prior-engaged semi-supervised deep learning method: Geophysics, **0**, 1–72.

Chen, H., M. D. Sacchi, and J. Gao, 2022, Seismic reflectivity inversion via a regularized deep image prior: Second International Meeting for Applied Geoscience & Energy, 2586–2590.

Chen, K. and M. D. Sacchi, 2017a, Elastic least-squares reverse time migration via linearized elastic full-waveform inversion with pseudo-hessian preconditioning: Geophysics, **82**, S341–S358.

——— 2017b, Elastic least-squares reverse time migration via linearized elastic full-waveform inversion with pseudo-hessian preconditioning: Geophysics, **82**, S341–S358.

Chen, Y., Y. Huang, and L. Huang, 2020, Suppressing migration image artifacts using a support vector machine method: Geophysics, **85**, S255–S268.

Chen, Y. and G. T. Schuster, 2020, Seismic inversion by newtonian machine learning: Geophysics, **85**, WA185–WA200.

Cheng, C., Y. He, B. Wang, and Y. Huang, 2020, Structure enhanced least-squares migration by deep learning based structural preconditioning, *in* SEG Technical Program Expanded Abstracts 2020, 2908–2912. Society of Exploration Geophysicists.

Cheng, J., N. Kazemi, and M. Sacchi, 2016, Least-squares migration via a gradient projection method - application to seismic data deblending: Presented at the 78th EAGE Conference and Exhibition 2016.

Chopra, S., J. Castagna, and O. Portniaguine, 2006, Seismic resolution and thin-bed reflectivity inversion: CSEG recorder, **31**, 19–25.

Chopra, S., J. Castagna, and Y. Xu, 2009, Thin-bed reflectivity inversion and some applications: First Break, **27**, 55–62.

Claerbout, J., 2014, Geophysical image estimation by example: Lulu. com.

Claerbout, J. F., 1985, Imaging the Earth's Interior: Blackwell Scientific Publications.

——— 1992, Earth soundings analysis: Processing versus inversion: Blackwell scientific publications london.

Claerbout, J. F. and S. Fomel, 2008, Image estimation by example: geophysical soundings image construction: multidimensional autoregression: Citeseer.

Clapp, M. L., 2005, Imaging under salt: Illumination compensation by regularized inversion: PhD thesis, Citeseer.

Dai, W., P. Fowler, and G. T. Schuster, 2012, Multi-source least-squares reverse time migration: Geophysical Prospecting, **60**, 681–695.

Davydenko, M. and D. Verschuur, 2017, Full-wavefield migration: using surface and internal multiples in imaging: Geophysical Prospecting, **65**, 7–21.

de Wit, R. W. L., J. Trampert, and R. D. van der Hilst, 2012, Toward quantifying uncertainty in travel time tomography using the null-space shuttle: Journal of Geophysical Research: Solid Earth, **117**.

Deal, M. M. and G. Nolet, 1996, Nullspace shuttles: Geophysical Journal International, **124**, 372–380.

Debeye, H. W. J. and P. Van Riel, 1990, Lp-norm deconvolution: Geophysical Prospecting, **38**, 381–403.

Dhara, A. and M. K. Sen, 2022, Physics-guided deep autoencoder to overcome the need for a starting model in full-waveform inversion: The Leading Edge, **41**, 375–381.

Dong, S., J. Cai, M. Guo, S. Suh, Z. Zhang, B. Wang, and e. Z. Li, 2012, Least-squares reverse time migration: Towards true amplitude imaging and improving the resolution, *in* SEG Technical Program Expanded Abstracts 2012, 1–5. Society of Exploration Geophysicists.

Downton, J. E., O. Collet, D. P. Hampson, and T. Colwell, 2020, Theory-guided data science-based reservoir prediction of a North Sea oil field: The Leading Edge, **39**, 742–750.

Dragoset, B., 1999, A practical approach to surface multiple attenuation: The Leading Edge, **18**, 104–108.

Dutta, G., 2017, Sparse least-squares reverse time migration using seislets: Journal of Applied Geophysics, **136**, 142 – 155.

Díaz, E. and P. Sava, 2016, Understanding the reverse time migration backscattering: noise or signal?: Geophysical Prospecting, **64**, 581–594.

Etgen, J., S. H. Gray, and Y. Zhang, 2009, An overview of depth imaging in exploration geophysics: Geophysics, **74**, WCA5–WCA17.

Farias, F., A. A. Souza, and R. C. Pestana, 2023, Applying a stochastic quasi-newton optimizer to least-squares reverse time migration: Computers & Geosciences, **171**, 105292.

Farshad, M. and H. Chauris, 2021, Accelerating the multi-parameter least-squares reverse time migration using an appropriate preconditioner: Computational Geosciences, **25**, 2071–2092.

Fichtner, A., 2010, Full seismic waveform modelling and inversion: Springer Science &

Business Media.

Fletcher, R., J. Hobro, and J. Rickett, 2023, Suppressing reflections with vector reflectivity acoustic modelling: 84th EAGE Annual Conference & Exhibition, 1–5.

Fletcher, R. P., D. Nichols, R. Bloor, and R. T. Coates, 2016, Least-squares migration — data domain versus image domain using point spread functions: The Leading Edge, **35**, 157–162.

Gal, Y. and Z. Ghahramani, 2016, Dropout as a bayesian approximation: Representing model uncertainty in deep learning: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, 1050–1059, JMLR.org.

Gao, W., G. Matharu, and M. D. Sacchi, 2020a, Fast least-squares reverse time migration via a superposition of kronecker products: Geophysics, **85**, S115–S134.

Gao, Z., C. Li, N. Liu, Z. Pan, J. Gao, and Z. Xu, 2020b, Large-dimensional seismic inversion using global optimization with autoencoder-based model dimensionality reduction: IEEE Transactions on Geoscience and Remote Sensing, **59**, 1718–1732.

Gao, Z., Z. Pan, J. Gao, and Z. Xu, 2019, Building long-wavelength velocity for salt structure using stochastic full waveform inversion with deep autoencoder based model reduction: Presented at the SEG International Exposition and Annual Meeting.

Gavish, M. and D. L. Donoho, 2014, The optimal hard threshold for singular values is $4/\sqrt{3}$: IEEE Transactions on Information Theory, **60**, 5040–5053.

Gholami, A. and M. D. Sacchi, 2012, A fast and automatic sparse deconvolution in the presence of outliers: IEEE Transactions on Geoscience and Remote Sensing, **50**, 4105–4116.

——— 2013, Fast 3D blind seismic deconvolution via constrained total variation and GCV: SIAM Journal on Imaging Sciences, **6**, 2350–2369.

Goodfellow, I., Y. Bengio, and A. Courville, 2016, Deep learning: MIT Press.

Gray, S. H., 1997, True-amplitude seismic migration: A comparison of three approaches: Geophysics, **62**, 929–936.

Greer, S., Z. Xue, and S. Fomel, 2018, Improving migration resolution by approximating the least-squares hessian using nonstationary amplitude and frequency matching, 4261–4265.

Gregor, K. and Y. LeCun, 2010, Learning fast approximations of sparse coding: ICML 2010 - Proceedings, 27th International Conference on Machine Learning, 399–406. 27th International Conference on Machine Learning, ICML 2010 ; Conference date: 21-06-2010 Through 25-06-2010.

Guitton, A., 2004, Amplitude and kinematic corrections of migrated images for nonunitary imaging operators: Geophysics, **69**, 1017–1024.

——— 2012, Blocky regularization schemes for Full-Waveform Inversion: Geophysical Prospecting, **60**, 870–884.

——— 2017, Fast 3d least-squares rtm by preconditioning with nonstationary matching fil-

ters: SEG Technical Program Expanded Abstracts 2017, 4395–4399.

Guitton, A., G. Ayeni, and E. Díaz, 2012, Constrained full-waveform inversion by model reparameterization: Geophysics, **77**, R117–R127.

Guitton, A., A. Valenciano, D. Bevc, and J. Claerbout, 2007, Smoothing imaging condition for shot-profile migration: Geophysics, **72**, S149–S154.

Guo, G., O. Stephane, G. Ali, and S. Hossein, 2022, Approximate data-domain hessian in extended-source time-domain full waveform inversion using matching filter and conjugate gradient method: 83rd EAGE Annual Conference & Exhibition, 1–5.

Hanke, M. and P. C. Hansen, 1993, Regularization methods for large-scale problems: Surv. Math. Ind, **3**, 253–315.

Hansen, P. C., 2010a, Discrete inverse problems: Society for Industrial and Applied Mathematics.

——— 2010b, Discrete inverse problems: insight and algorithms: SIAM.

Hauptmann, A., F. Lucka, M. Betcke, N. Huynh, J. Adler, B. Cox, P. Beard, S. Ourselin, and S. Arridge, 2018, Model-based learning for accelerated, limited-view 3-d photoacoustic tomography: IEEE Transactions on Medical Imaging, **37**, 1382–1393.

He, K., X. Zhang, S. Ren, and J. Sun, 2015, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), 1026–1034, IEEE Computer Society.

——— 2016, Deep residual learning for image recognition: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778.

——— 2016, Deep residual learning for image recognition: Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778.

Herrmann, F. J., C. R. Brown, Y. A. Erlangga, and P. P. Moghaddam, 2009, Curvelet-based migration preconditioning and scaling: Geophysics, **74**, A41–A46.

Hestenes, M. R. and E. Stiefel, 1952, Methods of conjugate gradients for solving linear systems: J Res NIST, **49**, 409–436.

Hornik, K., M. Stinchcombe, and H. White, 1989, Multilayer feedforward networks are universal approximators: Neural Networks, **2**, 359–366.

Hu, J., G. T. Schuster, and P. A. Valasek, 2001, Poststack migration deconvolution: Geophysics, **66**, 939–952.

Huang, Y. and J. Huang, 2021, Optimized plane-wave least-squares reverse-time migration via a convolutional network, *in* First International Meeting for Applied Geoscience & Energy, 1350–1354. SEG.

Huang, Y., R. Nammour, and W. Symes, 2016, Flexibly preconditioned extended least-squares migration in shot-record domain: Geophysics, **81**, S299–S315.

Huang, Y. and G. T. Schuster, 2014, Resolution limits for wave equation imaging: Journal of Applied Geophysics, **107**, 137–148.

Huynh-Thu, Q. and M. Ghanbari, 2008, Scope of validity of psnr in image/video quality assessment: Electronics Letters, **44**, 800 – 801.

Hyun, C. M., S. H. Baek, M. Lee, S. M. Lee, and J. K. Seo, 2021, Deep learning-based solvability of underdetermined inverse problems in medical imaging: Medical Image Analysis, **69**, 101967.

Idier, J. and Y. Goussard, 1993, Multichannel seismic deconvolution: IEEE Transactions on Geoscience and Remote Sensing, **31**, 961–979.

Ioffe, S. and C. Szegedy, 2015, Batch normalization: Accelerating deep network training by reducing internal covariate shift: Proceedings of the 32nd International Conference on Machine Learning, 448–456, PMLR.

Jiang, B. and J. Zhang, 2019, Least-squares migration with a blockwise hessian matrix: A prestack time-migration approach: Geophysics, **84**, R625–R640.

Kaaresen, K. F. and T. Taxt, 1998, Multichannel blind deconvolution of seismic signals: Geophysics, **63**, 2093–2107.

Kaur, H., N. Pham, and S. Fomel, 2020, Improving the resolution of migrated images by approximating the inverse hessian using deep learning: Geophysics, **85**, WA173–WA183.

Kazemi, N. and M. D. Sacchi, 2013, Sparse multichannel blind deconvolution: Geophysics, **79**, V143–V152.

Keating, S. D. and K. A. Innanen, 2021, Null-space shuttles for targeted uncertainty analysis in full-waveform inversion: Geophysics, **86**, R63–R76.

Khalil, A., H. Hoeber, G. Roberts, and F. Perrone, 2016, An alternative to least-squares imaging using data-domain matching filters, *in* 2016 SEG International Exposition and Annual Meeting. SEG.

Kingma, D. P. and J. Ba, 2014a, Adam: A method for stochastic optimization. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

——— 2014b, Adam: A method for stochastic optimization. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Kong, F., F. Picetti, V. Lipari, P. Bestagini, X. Tang, and S. Tubaro, 2022, Deep prior-based unsupervised reconstruction of irregularly sampled seismic data: IEEE Geoscience and Remote Sensing Letters, **19**, 1–5.

Korsmo, Ø., Y. Yang, N. Chemingui, A. Pankov, and A. Castiello, 2022, Nonlinear ls-rtm based on the seismic scale separation and full-wavefield multiparameter fwi: Second International Meeting for Applied Geoscience & Energy, 977–981.

Kramer, M. A., 1991, Nonlinear principal component analysis using autoassociative neural networks: AIChE journal, **37**, 233–243.

Kühl, H. and M. D. Sacchi, 2003, Least-squares wave-equation migration for avp/ava inver-

sion: Geophysics, **68**, 262–273.

Kumar, A., G. Hampson, T. Rayment, and T. Burgess, 2022, A deep learning inverse hessian for least-squares migration: **2022**, 1–5.

Kuo, J., J. Granstedt, U. Villa, and M. A. Anastasio, 2022, Computing a projection operator onto the null space of a linear imaging operator: tutorial: Journal of the Optical Society of America A, **39**, 470.

Lailly, P. and J. Bednar, 1983, The seismic inverse problem as a sequence of before stack migrations: Conference on inverse scattering: theory and application, 206–220.

Lambaré, G., J. Virieux, R. Madariaga, and S. Jin, 1992, Iterative asymptotic inversion in the acoustic approximation: Geophysics, **57**, 1138–1154.

LeCun, Y., Y. Bengio, and G. Hinton, 2015, Deep learning: Nature, **521**, 436–444.

Lesage, A.-C., J. Yao, F. Hussain, and D. J. Kouri, 2015, Low-frequency reflection-data augmentation by an inpainting method: 1d acoustic media: Geophysics, **80**, R139–R153.

Levin, S. A., 1989, Surface-consistent deconvolution: Geophysics, **54**, 1123–1133.

Levy, S. and P. K. Fullagar, 1981, Reconstruction of a sparse spike train from a portion of its spectrum and application to high-resolution deconvolution: Geophysics, **46**, 1235–1243.

Levy, S., J. Hunziker, E. Laloy, J. Irving, and N. Linde, 2022, Using deep generative neural networks to account for model errors in markov chain monte carlo inversion: Geophysical Journal International, **228**, 1098–1118.

Li, C., Q. Zhu, and B. Roy, 2022, Seismic super resolution method for enhancing stratigraphic interpretation: Second International Meeting for Applied Geoscience & Energy, 165–169, Society of Exploration Geophysicists and American Association of Petroleum Geologists.

Li, F., J. Gao, Z. Gao, X. Jiang, and W. Sun, 2020a, Least-squares reverse time migration with sparse regularization in the 2d wavelet domain: Geophysics, **85**, S313–S325.

Li, S., B. Liu, Y. Ren, Y. Chen, S. Yang, Y. Wang, and P. Jiang, 2020b, Deep-Learning Inversion of Seismic Data: IEEE Transactions on Geoscience and Remote Sensing, **58**, 2135–2149.

Liu, B., S. Yang, Y. Ren, X. Xu, P. Jiang, and Y. Chen, 2021a, Deep-learning seismic full-waveform inversion for realistic structural models: Geophysics, **86**, R31–R44.

Liu, D. C. and J. Nocedal, 1989, On the limited memory bfgs method for large scale optimization: Mathematical programming, **45**, 503–528.

Liu, M., D. Grana, and L. P. de Figueiredo, 2022, Uncertainty quantification in stochastic inversion with dimensionality reduction using variational autoencoder: Geophysics, **87**, M43–M58.

Liu, Q., Y. Lu, H. Sun, and H. Zhang, 2019, Single-step data-domain least-squares reverse-time migration using gabor deconvolution for subsalt imaging: IEEE Geoscience and Remote Sensing Letters, **17**, 13–16.

Liu, Q. and D. Peter, 2018, One-step data-domain least-squares reverse time migrationlsrtm: Geophysics, **83**, R361–R368.

Liu, X., Y. Chen, and L. Huang, 2020a, Least-squares reverse-time migration with a machine-learning-based denoising preconditioner, *in* SEG Technical Program Expanded Abstracts 2020, 2993–2997. Society of Exploration Geophysicists.

——— 2020b, Least-squares reverse-time migration with a machine-learning-based denoising preconditioner: Presented at the SEG International Exposition and Annual Meeting.

Liu, Y., Y. Du, and Y. Luo, 2021b, Sparsity-promoting least-squares interferometric migration for high-resolution passive source location: Geophysics, **86**, KS1–KS9.

Liu, Y., W. W. Symes, and Z. Li, 2013, Multisource least-squares extended reverse time migration with preconditioning guided gradient method, *in* SEG Technical Program Expanded Abstracts 2013, 3709–3715. Society of Exploration Geophysicists.

Liu, Z., Y. Chen, and G. Schuster, 2020c, Deep convolutional neural network and sparse least-squares migration: Geophysics, **85**, WA241–WA253.

Lu, Y., H. Sun, X. Wang, Q. Liu, and H. Zhang, 2020, Improving the image quality of elastic reverse-time migration in the dip-angle domain using deep learning: Geophysics, **85**, S269–S283.

Luo, Y. and G. T. Schuster, 1991, Wave equation inversion of skeletalized geophysical data: Geophysical Journal International, **105**, 289–294.

Maier, A. K., C. Syben, B. Stimpel, T. Würfl, M. Hoffmann, F. Schebesch, W. Fu, L. Mill, L. Kling, and S. Christiansen, 2019, Learning with known operators reduces maximum error bounds: Nature Machine Intelligence, **1**, 373–380.

Mandelli, S., F. Borra, V. Lipari, P. Bestagini, A. Sarti, and S. Tubaro, 2019, Seismic data interpolation through convolutional autoencoder: 2018 SEG International Exposition and Annual Meeting, SEG 2018, 4101–4105.

Martin, G. S., R. Wiley, and K. J. Marfurt, 2006a, Marmousi2: An elastic upgrade for Marmousi: The Leading Edge, **25**, 156–166.

——— 2006b, Marmousi2: An elastic upgrade for marmousi: The leading edge, **25**, 156–166.

Meinhardt, T., M. Möller, C. Hazirbas, and D. Cremers, 2017, Learning proximal operators: Using denoising networks for regularizing inverse imaging problems: CoRR, **abs/1704.03488**.

Monga, V., Y. Li, and Y. C. Eldar, 2021, Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing: IEEE Signal Processing Magazine, **38**, 18–44.

Nemeth, T., C. Wu, and G. T. Schuster, 1999, Least-squares migration of incomplete reflection data: Geophysics, **64**, 208–221.

Nguyen, B. D. and G. A. McMechan, 2015, Five ways to avoid storing source wavefield snapshots in 2d elastic prestack reverse time migration: Geophysics, **80**, S1–S18.

Nichols, D., 1997, A simple example of a null space and how to modify it: Stanford Explo-

ration Project Report, **82**, 185–192.

Nocedal, J., 1980, Updating quasi-newton matrices with limited storage: Mathematics of computation, **35**, 773–782.

Obmann, D., J. Schwab, and M. Haltmeier, 2020, Deep synthesis network for regularizing inverse problems: Inverse Problems, **37**, 015005.

Oldenburg, D. W., T. Scheuer, and S. Levy, 1983, Recovery of the acoustic impedance from reflection seismograms.: Geophysics, **48**, 1318–1337.

Operto, S., A. Gholami, H. Aghamiry, G. Guo, S. Beller, K. Aghazade, F. Mamfoumbi, L. Combe, and A. Ribodetti, 2023, Extending the search space of full-waveform inversion beyond the single-scattering born approximation: A tutorial review: Geophysics, **88**, R671–R702.

Oropeza, V. and M. Sacchi, 2011, Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis: Geophysics, **76**, V25–V32.

Osypov, K., Y. Yang, A. Fournier, N. Ivanova, R. Bachrach, C. E. Yarman, Y. You, D. Nichols, and M. Woodward, 2013, Model-uncertainty quantification in seismic tomography: Method and applications: Geophysical Prospecting, **61**, 1114–1134.

Paffenholz, J., B. McLain, J. Zaske, and P. J. Keliher, 2002, Subsalt multiple attenuation and imaging: Observations from the Sigsbee2B synthetic dataset: SEG Technical Program Expanded Abstracts 2002, 2122–2125, Society of Exploration Geophysicists.

Pan, S. J. and Q. Yang, 2010, A survey on transfer learning: IEEE Transactions on Knowledge and Data Engineering, **22**, 1345–1359.

Park, M. J. and M. D. Sacchi, 2020, Automatic velocity analysis using convolutional neural network and transfer learning: Geophysics, **85**, V33–V43.

Paszke, A., S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, 2017, Automatic differentiation in pytorch.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, 2019, Pytorch: An imperative style, high-performance deep learning library, *in* Advances in Neural Information Processing Systems 32, 8024–8035. Curran Associates, Inc.

Peters, B., B. R. Smithyman, and F. J. Herrmann, 2019, Projection methods and applications for seismic nonlinear inverse problems with multiple constraints: Geophysics, **84**, R251–R269.

Plessix, R.-E., 2006a, A review of the adjoint-state method for computing the gradient of a functional with geophysical applications: Geophysical Journal International, **167**, 495–503.

——— 2006b, A review of the adjoint-state method for computing the gradient of a functional with geophysical applications: Geophysical Journal International, **167**, 495–503.

Putzky, P. and M. Welling, 2017, Recurrent inference machines for solving inverse problems: CoRR, **abs/1706.04008**.

Ravasi, M., 2021, Seismic wavefield processing with deep preconditioners: First International Meeting for Applied Geoscience & Energy Expanded Abstracts, 2859–2863.

Richardson, A., 2018, Seismic full-waveform inversion using deep learning tools and techniques.

Rickett, J. E., 2003, Illumination-based normalization for wave-equation depth migration: Geophysics, **68**, 1371–1379.

Robinson, E. and S. Treitel, 1980, Geophysical signal analysis. Prentice-Hall Foundations of Earth Science Series: Prentice-Hall.

Rocha, D., P. Sava, and A. Guitton, 2018, 3D acoustic least-squares reverse time migration using the energy norm: Geophysics, **83**, S261–S270.

Ronneberger, O., P. Fischer, and T. Brox, 2015a, U-net: Convolutional networks for biomedical image segmentation: CoRR, **abs/1505.04597**.

——— 2015b, U-net: Convolutional networks for biomedical image segmentation, *in* Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, 234–241. MIC.

Saad, Y., 2011, Numerical methods for large eigenvalue problems: revised edition: SIAM.

Sacchi, M. D., 1997, Reweighting strategies in seismic deconvolution: Geophysical Journal International, **129**, 651–656.

Sacchi, M. D., D. R. Velis, and A. H. Cominguez, 1994, Minimum entropy deconvolution with frequency-domain constraints: Geophysics, **59**, 938–945.

Sacchi, M. D., J. Wang, and H. Kuehl, 2007, Estimation of the diagonal of the migration blurring kernel through a stochastic approximation, 2437–2441.

Sava, P. and B. Biondi, 2004, Wave-equation migration velocity analysis. i. theory: Geophysical prospecting, **52**, 593–606.

Scales, J. and M. Smith, 1994, Introductory geophysical inverse theory. Center for Wave Phenomena: Samizdat Press.

Schuster, G., 2017, Seismic inversion: SEG.

Schuster, G. and Z. Liu, 2019, Least squares migration: Current and future directions: Presented at the 81st EAGE Conference and Exhibition 2019.

Schwab, J., S. Antholzer, and M. Haltmeier, 2019, Deep null space learning for inverse problems: Convergence analysis and rates: Inverse Problems, **35**.

——— 2020, Big in Japan: Regularizing Networks for Solving Inverse Problems: Journal of Mathematical Imaging and Vision, **62**, 445–455.

Shin, C., K. Yoon, K. J. Marfurt, K. Park, D. Yang, H. Y. Lim, S. Chung, and S. Shin, 2001, Efficient calculation of a partial-derivative wavefield using reciprocity for seismic

imaging and inversion: Geophysics, **66**, 1856–1863.

Siahkoohi, A., G. Rizzuti, and F. J. Herrmann, 2022a, Deep bayesian inference for seismic imaging with tasks: Geophysics, **87**, S281–S302.

Siahkoohi, A., G. Rizzuti, R. Orozco, and F. J. Herrmann, 2022b, Reliable amortized variational inference with physics-based latent distribution correction.

Sochacki, J., R. Kubichek, J. George, W. R. Fletcher, and S. Smithson, 1987, Absorbing boundary conditions and surface waves: Geophysics, **52**, 60–71.

Sun, J., K. A. Innanen, and C. Huang, 2021, Physics-guided deep learning for seismic inversion with hybrid training and uncertainty analysis: Geophysics, **86**, R303–R317.

Sun, J., Z. Niu, K. A. Innanen, J. Li, and D. O. Trad, 2020, A theory-guided deep-learning formulation and optimization of seismic waveform inversion: Geophysics, **85**, R87–R99.

Symes, W. W., 2007, Reverse time migration with optimal checkpointing: Geophysics, **72**, SM213–SM221.

Tang, Y., 2009, Target-oriented wave-equation least-squares migration/inversion with phase-encoded hessian: Geophysics, **74**, WCA95–WCA107.

Tangkijwanichakul, T. and S. Fomel, 2021, Inverse hessian estimation in least-squares migration using chains of operators: 82nd EAGE Annual Conference & Exhibition, EAGE, Extended Abstracts, **2021**, 1–5.

Tarantola, A., 1984a, Inversion of seismic reflection data in the acoustic approximation: Geophysics, **49**, 1259–1266.

——— 1984b, Linearized inversion of seismic reflection data: Geophysical Prospecting, **32**, 998–1015.

Taylor, H. L., S. C. Banks, and J. F. McCoy, 1979, Deconvolution with the l1 norm: Geophysics, **44**, 39–52.

Tenorio, L., 2001, Modeling non-gaussian reflectivities: Generalizing wiener-levinson deconvolution: Geophysics, **66**, 1913–1920.

Tompson, J., R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, 2015, Efficient object localization using convolutional networks: CVPR, 648–656, IEEE Computer Society.

Torres, K. and M. Sacchi, 2022a, Least-squares reverse time migration via deep learning-based updating operators: Geophysics, **87**, S315–S333.

——— 2022b, Least-squares reverse time migration via deep learning-based updating operators: Geophysics, **87**, S315–S333.

Torres, K. and M. D. Sacchi, 2022c, Deep Null Space Regularization for Seismic Inverse Problems: 83rd EAGE Annual Conference & Exhibition, 1–5, European Association of Geoscientists & Engineers.

——— 2023, Deep decomposition learning for reflectivity inversion: Geophysical Prospecting, **00**, 00–00.

Ulrych, T. J., D. R. Velis, and M. D. Sacchi, 1995, Wavelet estimation revisited: The

Leading Edge, **14**, 1139–1143.

Ulyanov, D., A. Vedaldi, and V. Lempitsky, 2018, Deep image prior: Proceedings of the IEEE conference on computer vision and pattern recognition, 9446–9454.

Valenciano, A. A., B. Biondi, and A. Guitton, 2006, Target-oriented wave-equation inversion: Geophysics, **71**, A35–A38.

Vamaraju, J., J. Vila, M. Araya-Polo, D. Datta, M. Sidahmed, and M. K. Sen, 2021, Minibatch least-squares reverse time migration in a deep-learning framework: Geophysics, **86**, S125–S142.

Vantassel, J. P., K. Kumar, and B. R. Cox, 2021, Using convolutional neural networks to develop starting models for 2d full waveform inversion: arXiv, 2104.01626.

Velis, D. R., 2008, Stochastic sparse-spike deconvolution: Geophysics, **73**.

Versteeg, R., 1994, The marmousi experience: Velocity model determination on a synthetic complex data set: The Leading Edge, **13**, 927–936.

Walden, A. T., 1985, Non-gaussian reflectivity, entropy, and deconvolution: Geophysics, **50**, 2862–2888.

Wang, J., H. Kuehl, and M. D. Sacchi, 2005, High-resolution wave-equation ava imaging: Algorithm and tests with a data set from the western canadian sedimentary basin: Geophysics, **70**, S91–S99.

Wang, J. and M. D. Sacchi, 2007a, High-resolution wave-equation amplitude-variation-with-ray-parameter (avp) imaging with sparseness constraints: Geophysics, **72**, S11–S18.

——— 2007b, High-resolution wave-equation amplitude-variation-with-ray-parameter (avp) imaging with sparseness constraints: Geophysics, **72**, S11–S18.

——— 2009, Structure constrained least-squares migration: SEG Technical Program Expanded Abstracts 2009, 2763–2767.

Wang, M., S. Huang, and P. Wang, 2017a, Improved iterative least-squares migration using curvelet-domain hessian filters: SEG Technical Program Expanded Abstracts 2017, 4555–4560.

Wang, P., A. Gomes, Z. Zhang, and M. Wang, 2016, Least-squares rtm: Reality and possibilities for subsalt imaging, *in* SEG Technical Program Expanded Abstracts 2016, 4204–4209. Society of Exploration Geophysicists.

Wang, P., S. Huang, and M. Wang, 2017b, Improved subsalt images with least-squares reverse time migration: Interpretation, **5**, SN25–SN32.

Wang, Z., A. Bovik, H. Sheikh, and E. Simoncelli, 2004, Image quality assessment: from error visibility to structural similarity: IEEE Transactions on Image Processing, **13**, 600–612.

White, R. E. and T. Hu, 1998, How accurate can a well tie be?: The Leading Edge, **17**, 1065–1071.

Whitmore, N., J. Ramos-Martinez, Y. Yang, and A. Valenciano, 2020, Full wavefield mod-

eling with vector reflectivity: EAGE 2020 Annual Conference & Exhibition Online, 1–5.

Whitmore, N. D., 1983, Iterative depth migration by backward time propagation, *in* SEG Technical Program Expanded Abstracts 1983, 382–385. Society of Exploration Geophysicists.

Wiggins, R. A., 1978, Minimum entropy deconvolution: Geoexploration, **16**, 21–35.

Witte, P., M. Yang, and F. Herrmann, 2017, Sparsity-promoting least-squares migration with the linearized inverse scattering imaging condition: Presented at the 79th EAGE Conference and Exhibition 2017.

Wong, M., B. Biondi, and S. Ronen, 2015, Imaging with primaries and free-surface multiples by joint least-squares reverse time migration: Geophysics, **80**.

Woodward, M. J., 1992, Wave-equation tomography: Geophysics, **57**, 15–26.

Wu, X., L. Liang, Y. Shi, and S. Fomel, 2019, Faultseg3d: Using synthetic data sets to train an end-to-end convolutional neural network for 3d seismic fault segmentation: Geophysics, **84**, IM35–IM45.

Wu, X., S. Yan, Z. Bi, S. Zhang, and H. Si, 2021a, Deep learning for multi-dimensional seismic impedance inversion: Geophysics, **86**.

——— 2021b, Deep learning for multidimensional seismic impedance inversion: Geophysics, **86**, R735–R745.

Wu, Z., Y. Liu, and J. Yang, 2024, Nonlinear least-squares reverse time migration of prismatic waves for delineating steeply dipping structures: Geophysics, **89**, S129–S143.

Xiang, K., X. Chen, H. Chen, and Y. Chen, 2016, Least-squares reverse time migration for blended data with a local low-rank constraint: Presented at the 78th EAGE Conference and Exhibition 2016.

Xu, L. and M. D. Sacchi, 2018, Preconditioned acoustic least-squares two-way wave-equation migration with exact adjoint operator: Geophysics, **83**, S1–S13.

Xu, P., B. Feng, H. Wang, and S. Liu, 2022a, Noniterative least-squares reverse time migration based on an efficient asymptotic hessian/point-spread function estimation: Geophysics, **87**, S169–S184.

Xu, W., V. Lipari, P. Bestagini, M. Ravasi, W. Chen, and S. Tubaro, 2022b, Intelligent seismic deblending through deep preconditioner: IEEE Geoscience and Remote Sensing Letters, **19**, 1–5.

Yang, F. and J. Ma, 2019, Deep-learning inversion: A next-generation seismic velocity model building method: Geophysics, **84**, R583–R599.

Yang, Y., J. Ramos-Martinez, D. Whitmore, G. Huang, and N. Chemingui, 2021, Simultaneous velocity and reflectivity inversion: Fwi+ lsrtm: 82nd EAGE Annual Conference & Exhibition, 1–5.

Yao, G. and H. Jakubowicz, 2012, Non-linear least-squares reverse-time migration: SEG International Exposition and Annual Meeting, SEG–2012.

——— 2016, Least-squares reverse-time migration in a matrix-based formulation: Geophysical Prospecting, **64**, 611–621.

Yilmaz, O., 2001, Seismic data analysis, volume **10** *of* Investigations in Geophysics: Society Of Exploration Geophysicists, 2 ed edition.

Yosinski, J., J. Clune, Y. Bengio, and H. Lipson, 2014, How transferable are features in deep neural networks?: Advances in neural information processing systems, **27**.

Yu, J., J. Hu, G. T. Schuster, and R. Estill, 2006, Prestack migration deconvolution: Geophysics, **71**, S53–S62.

Yu, S. and J. Ma, 2021a, Deep learning for geophysics: Current and future trends: Earth and Space Science Open Archive, 73.

——— 2021b, Deep learning for geophysics: Current and future trends: Reviews of Geophysics, **59**, e2021RG000742. e2021RG000742 2021RG000742.

——— 2021c, Deep Learning for Geophysics: Current and Future Trends: Reviews of Geophysics, **59**.

Zeng, C., S. Dong, and B. Wang, 2014, Least-squares reverse time migration: Inversion-based imaging toward true reflectivity: The Leading Edge, **33**, 962–968.

——— 2017, A guide to least-squares reverse time migration for subsalt imaging: Challenges and solutions: Interpretation, **5**, SN1–SN11.

Zhang, J. and B. Ghanem, 2017, Ista-net: Iterative shrinkage-thresholding algorithm inspired deep network for image compressive sensing: CoRR, **abs/1706.07929**.

Zhang, K., W. Zuo, Y. Chen, D. Meng, and L. Zhang, 2017, Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising: IEEE Transactions on Image Processing, **26**, 3142–3155.

Zhang, R. and J. Castagna, 2011, Seismic sparse-layer reflectivity inversion using basis pursuit decomposition: Geophysics, **76**, R147–R158.

Zhang, W., J. Gao, T. Yang, X. Jiang, and W. Sun, 2021, Least-squares reverse time migration using convolutional neural networks: Geophysics, **86**.

Zhang, Y. and J. Sun, 2009, Practical issues of reverse time migration: True amplitude gathers, noise removal and harmonic-source encoding: Beijing International Geophysical Conference and Exposition 2009: Beijing 2009 International Geophysical Conference and Exposition, Beijing, China, 24–27 April 2009, 204–204.

APPENDIX A

Parametrization of the projection operators and figures of merit from chapter 2

Conceptually, each updating operator block $\mathcal{P}_{\theta_k}$, corresponding to the k iteration of the Deep-LSRTM framework, can be characterized as an N-stack of 2D convolutional layers with trainable weights and non-linear activation functions of the form:

$$\mathcal{P}_{\theta_k} = \big((\phi_N \circ W_{w_N,b_N}) \circ ... \circ (\phi_1 \circ W_{w_1,b_1})\big)_k, \quad n = 1, ..., N, \tag{A.1}$$

where ϕ is the rectifier linear unit function

$$\text{ReLU}(\mathbf{m}) = \max(0, \mathbf{m}), \tag{A.2}$$

which is applied to each convolutional layer for easier and faster training, and

$$W^q_{w_n,b_n} = \Big(b^q_n + \sum_{p \in P} w^{q,p}_n * g_p \Big), \quad q \in Q, \tag{A.3}$$

where P and Q denote the number of channels for the input data g and the output feature maps W_{w_n,b_n} of the nth convolutional layer, respectively, b^q_n is a bias term, and $w^{q,p}_n$ is the two-dimensional convolutional kernel (or filter). We provide two types of measurements to assess the performances of different techniques. The PSNR function (Huynh-Thu and Ghanbari, 2008) computes the peak signal-to-noise ratio between the reconstructed image $\mathbf{m}$ and the ground-truth image $\mathbf{m}_{\text{true}}$ and is calculated by

$$\text{PSNR}(\mathbf{m}, \mathbf{m}_{\text{true}}) = 20 \log_{10} \Big(\frac{\max(\mathbf{m}_{\text{true}})}{\sqrt{\text{MSE}}}\Big), \tag{A.4}$$

where MSE is the mean squared error and $\max(\cdot)$ is the maximum possible intensity value. The structural similarity index measure (SSIM) (Wang et al., 2004) is used to indicate the perceived similarity in the reconstructions. It is locally computed between two windows x and y of equal size $N \times N$, which move pixel-by-pixel over the entire reconstructed and ground-truth images. The SSIM is given by the following expression:

$$\text{SSIM(x,y)} = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{A.5}$$

where μ_x and σ_x indicate the average and variance of x, respectively; μ_y and σ_y denote the average and variance of y, respectively; σ_{xy} represent the covariance of x and y, and c_1 and c_2 are two constants to avoid instability.