

# Important Notice

This copy may be used only for the purposes of research and private study, and any use of the copy for a purpose other than research or private study may require the authorization of the copyright owner of the work in question. Responsibility regarding questions of copyright that may arise in the use of this copy is assumed by the recipient.

UNIVERSITY OF CALGARY

Application of multiple reflections in seismic imaging

by

Shang Huang

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE  
DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN GEOLOGY AND GEOPHYSICS

CALGARY, ALBERTA

NOVEMBER, 2023

© Shang Huang 2023

# Abstract

Multiple reflections in seismic imaging provide additional information and insights into subsurface structures. However, their practical utilization could be improved due to their complex interaction with the subsurface and interference with primary reflections and noise. This thesis aims to use multiples in seismic imaging in two contexts: classical and machine learning imaging.

For the first context, I attempt to extend aperture illumination in a phase shift plus interpolation (PSPI) migration by adding scattering terms in the phase-shift wavefield propagation operator. This method iteratively adds scattering terms for each reflector as the source and receiver wavefields propagate downward into the subsurface, which helps in efficiently extending the illumination of horizontal reflector edges. In the second context, I consider improving images from reverse time migration (RTM). RTM with multiple reflections (RTMM) can improve illumination but suffers from interferences between different orders of multiples. To overcome this limitation, I proposed a method based on a convolutional neural network (CNN) and U-Nets that approximates the inverse of the Hessian similarly to least squares migration, but with less computational cost. The U-NET is trained to learn patterns representing the relation between the reflectivity obtained through RTMM and the true reflectivity. I further developed this by adding a discrete wavelet transform (DWT) input channel which provides an additional constraint that helps to enhance image resolution.

Finally, as a key application of the above techniques, I consider the problem of time-lapse seismic monitoring which attempts to detect very weak signal differences produced by changes in reservoirs. This technique is affected by changes in the near-surface noise and insufficient illumination to detect the weak changes. I proposed a novel method that leverages stacked long short-term memory (SD-LSTM) and U-Net neural networks to predict and mitigate noise in monitor data. I test the method in a field dataset, DAS VSP data from the CaMI FRS project. The output provides meaningful information and prediction for

CO<sub>2</sub> injection migration within a target area. This result aligns closely with the CaMI FRS project CO<sub>2</sub> injection plan, providing valuable insights for monitoring the CO<sub>2</sub> migration paths for the Basal Belly River Sandstone Formation.

# Preface

This thesis is written in a hybrid style. Chapter 2 is based on a non-submitted body of work, Chapter 3 is based on a paper published in the journal MDPI-Sensors, and chapters 4 and 5 are based on two papers that are currently being prepared as a manuscript for submission to Geophysics and IEEE Transactions on Geoscience and Remote Sensing. The published paper in this thesis with permission of the co-author and MDPI-Sensors journal.

A version of Chapter 3 was published as Huang, S., and Trad, D. O., 2023, Convolutional Neural-Network-Based Reverse-Time Migration with Multiple Reflections: Sensors, 23, no. 8., 4012.

The codes used in Chapter 2 for phase shift plus interpolation were developed by Dr. Robert Ferguson (Ferguson and Margrave, 2005). These codes were adapted and extended by me for the purposes of incorporating scattering terms to generate multiple reflections, and exploring the impact of multiples on reflectivity resolution.

The codes used in Chapter 5 for the data loader and the style of displaying shot records were adapted from Abdullah Alali (Alali et al., 2022) for the purpose of showing selected shot records after using the proposed method.

# Acknowledgements

First, I would like to express my most profound appreciation to my supervisor, Dr. Daniel Trad, for his supervision. His helpful patience, insightful guidance and feedback, and solid expert knowledge encouraged me to enjoy my research journey.

I would like to sincerely thank my committee members, Dr. Kristopher Inanan, Dr. Brandon Karchewski, Dr. Jan Dettmer, Dr. Robert Ferguson and Dr. Mauricio D. Sacchi, for their time and expertise in geophysics.

I am also thankful to Kevin Hall and Kevin Bertram for their support on field data, office devices, preparations for each year's CREWES annual meeting, and settings of presentations and posters for different conventions.

I am also grateful to Dr. Samuel Gray, Dr. Alison Malcolm, and Dr. Ali Fathalian for their helpful discussions. Their suggestions enriched my research.

I would like to thank all of the students and staff members of CREWES who helped me and made the group environment full of happiness. Special thanks to Dr. Huaizhen Chen, Dr. Jian Sun, Dr. Junxiao Li, Dr. Ali Fathalian, Dr. Marcelo Guarido, Dr. Xiaohui Cai, Dr. Yichuan Wang, Xin Fu, Tianze Zhang, Zhan Niu, Qi Hu, Luping Qu, Jinji Li, He Liu, Ziguang Su, Kai Zhuang and Dr. Ron Weir for their helpful discussions and support.

I would also thank my friends, Dr. Paulina Wozniakowska, Minhui Choi, Dr. Wei Wei, Dr. Yiran Wang, Jianqi Zhu, Yixuan Wang, Ziyi Qiu and Rui Wang, for their warmful friendship and support in my research and life.

I would like to thank CREWES industrial sponsors, the Natural Science and Engineering

Research Council of Canada (NSERC), and the China Scholarship Council (CSC), for the financial support that made this thesis possible.

Finally, I would like to thank Carbon Management Canada and the Containment and Monitoring Institute for permission to show the field data in Chapter 5.

*To my parents, Weidong and Yujuan, for their unwavering support and love.*  
*To my husband, Xiangyu, for encouraging and supporting me along this journey.*

# Table of Contents

Abstract	ii
Preface	iv
Acknowledgements	v
Dedication	vii
Table of Contents	x
List of Figures	xvii
List of Tables	xviii
List of Symbols, Abbreviations, and Nomenclature	xix
Epigraph	xxvi
<b>1 Introduction</b>	<b>1</b>
1.1 Seismic primary and multiple reflections . . . . .	1
1.2 Multiple applications in conventional migration . . . . .	2
1.2.1 Incorporating multiple reflections in phase shift plus interpolation mi- gration . . . . .	4
1.3 Multiple reflections in migration combined with machine learning method . .	5
1.4 Thesis overview and objectives . . . . .	8
<b>2 Scattering terms in wavefield propagation for migration</b>	<b>12</b>
2.1 Phase shift plus interpolation migration (PSPI) with scattering terms . . . . .	12
2.1.1 Theory . . . . .	13
2.1.2 Numerical examples . . . . .	17
2.1.3 Considerations . . . . .	22
2.1.4 Conclusions . . . . .	23
<b>3 Convolutional neural network-based reverse time migration with multiple   reflections</b>	<b>24</b>
3.1 Summary . . . . .	24

3.2	Introduction . . . . .	25
3.3	Method . . . . .	27
3.3.1	Four Scenarios in Reverse-Time Migration . . . . .	28
3.3.2	Wavefield Inconsistency . . . . .	30
3.3.3	RTM with Multiples . . . . .	31
3.3.4	Neural Network Plan for Four Workflows . . . . .	40
3.4	Train and test set . . . . .	42
3.5	Results and discussions . . . . .	44
3.5.1	Example 1: Canadian Foothills . . . . .	45
3.5.2	Example 2: Overthrust . . . . .	55
3.5.3	Example 3: SEAM Phase 1 . . . . .	59
3.5.4	Discussion . . . . .	63
3.5.5	Model Performance and Computation Time . . . . .	65
3.6	Conclusions . . . . .	66
<b>4</b>	<b>Discrete wavelet transform application in a CNN-based reverse time mi- gration with multiple reflections</b>	<b>68</b>
4.1	Summary . . . . .	68
4.2	Introduction . . . . .	69
4.3	Discrete Wavelet Transform (DWT) . . . . .	70
4.3.1	DWT in one dimension . . . . .	71
4.3.2	DWT in two dimensions . . . . .	72
4.3.3	Neural network training strategy . . . . .	73
4.4	Numerical examples . . . . .	77
4.4.1	Train and test set . . . . .	77
4.4.2	Using DWT in the network input . . . . .	78
4.4.3	Using DWT and noise in the network input . . . . .	84
4.5	Conclusions . . . . .	86
<b>5</b>	<b>Time-lapse data matching using neural networks with multiple reflections</b>	<b>100</b>
5.1	Summary . . . . .	100
5.2	Introduction . . . . .	101
5.3	Theory . . . . .	103
5.3.1	Time-lapse seismic . . . . .	103
5.3.2	Recurrent neural network (RNN) and LSTM . . . . .	107
5.3.3	Bi-LSTM . . . . .	109
5.3.4	Stacked LSTM and Bi-LSTM . . . . .	111
5.4	Workflow . . . . .	112
5.5	Neural network settings . . . . .	114
5.6	Numerical example . . . . .	117
5.6.1	Horizontal-layered model . . . . .	118
5.6.2	Curve-fault model . . . . .	122
5.6.3	Marmousi model . . . . .	125
5.6.4	Overthrust model . . . . .	130
5.6.5	DAS VSP CaMI FRS CO <sub>2</sub> injection project . . . . .	138

5.7	Discussion . . . . .	145
5.8	Conclusion . . . . .	147
<b>6</b>	<b>Conclusion</b>	<b>149</b>
6.1	Summary . . . . .	149
6.2	Future work . . . . .	151
	<b>Bibliography</b>	<b>153</b>

# List of Figures

1.1	Propagation paths of primary (red line) and the first-order multiple (blue line) reflections. . . . .	3
1.2	Comparison of (a) primary and (b) multiple reflection illumination. . . . .	3
2.1	Workflow for PSPI migration with scattering terms. . . . .	13
2.2	The horizontal-layered example. (a) True velocity model. (b) Smoothed velocity model. (c) Shot records for the horizontal-layered model at 750, 1000, 1250, 1500 and 1750 meters offset below the surface at 15 meters depth. . . . .	19
2.3	The horizontal-layered example. (a) PSPI migration without scattering term; (b) PSPI migration with scattering terms. . . . .	20
2.4	The boxcar model example. (a) True velocity model. (b) Smoothed velocity model. (c) Shot records simulated at 1800, 2800, and 3800 m on the surface. . . . .	20
2.5	The boxcar model example. (a) PSPI migration without scattering term; (b) PSPI migration with scattering terms. . . . .	21
2.6	The boxcar model with a dome. (a) True boxcar-dome velocity model. (b) Smoothed velocity model. (c) Shot records. . . . .	22
2.7	The boxcar model with a dome. (a) PSPI migration without scattering term; (b) PSPI migration with scattering terms. . . . .	23
3.1	(a) RTM of Scenario 1: Smoothed velocity as the input in the RTM process. (b) RTM of Scenario 4: True velocity as the input in the RTM with multiple energy. (c) RTM of Scenario 3: Smooth velocity as the input in the RTM with multiple reflections. (d) Shot illumination of Scenario 1. (e) Shot illumination of Scenario 4. (f) Shot illumination of Scenario 3. . . . .	30
3.2	Pluto example: (a) background reflectivity, (b) true reflectivity, (c) RTM image without multiple energy, (d) RTM image with multiple energy, (e) RTMM with true band-limited reflectivity, and (f) true velocity model. . . . .	33
3.3	Detailed workflow of the modified U-net architecture. Each blue box represents a multi-channel feature. The yellow boxes stand for the concatenated copied feature from the encoder part. The arrows between boxes correspond to the different operations as shown in the right legend. The number of channels is located on top of the box and the image dimensionality is denoted on the left edge. . . . .	34
3.4	Neural network model plan for four scenarios. . . . .	42

3.3	Canadian Foothills model results: <b>(a)</b> reflectivity from the background velocity, <b>(b)</b> true band-limited reflectivity, <b>(c)</b> RTM image without multiple reflections, <b>(d)</b> RTM image with multiple reflections, <b>(e)</b> model R1 result based on workflow 1, <b>(f)</b> model R3 result based on workflow 3, <b>(g)</b> model R4 result based on workflow 4, <b>(h)</b> true Foothills velocity, and <b>(i)</b> LSRTM result after 15 iterations. The boxes indicate areas shown in detail in the next figures.	49
3.4	Amplitude spectrum comparison between models R1, R3, and R4 results for the Canadian Foothills example.	50
3.5	Foothills red box No. 1 results: <b>(a)</b> reflectivity from the background velocity, <b>(b)</b> true windowed band-limited reflectivity, <b>(c)</b> RTM image without multiple energy, <b>(d)</b> RTM image with multiple energy, <b>(e)</b> model R1 result based on workflow 1, <b>(f)</b> model R3 result based on workflow 3, <b>(g)</b> model R4 result based on workflow 4, and <b>(h)</b> true windowed velocity.	52
3.6	Foothills red box No. 2 results: <b>(a)</b> reflectivity from the background velocity, <b>(b)</b> true windowed band-limited reflectivity, <b>(c)</b> RTM image without multiple energy, <b>(d)</b> RTM image with multiple energy, <b>(e)</b> model R1 result based on workflow 1, <b>(f)</b> model R3 result based on workflow 3, <b>(g)</b> model R4 result based on workflow 4, and <b>(h)</b> true windowed velocity.	54
3.7	Crossplots for the Canadian Foothills example: the true band-limited reflectivity against the predicted reflectivity by using models R1 and R3, respectively.	55
3.8	Overthrust model results: <b>(a)</b> reflectivity from the background velocity, <b>(b)</b> true band-limited reflectivity, <b>(c)</b> RTM image without multiple energy, <b>(d)</b> RTM image with multiple energy, <b>(e)</b> model R1 result based on workflow 1, <b>(f)</b> model R3 result based on workflow 3, <b>(g)</b> model R4 result based on workflow 4, and <b>(h)</b> true Overthrust velocity.	57
3.9	Overthrust red box results: <b>(a)</b> reflectivity from the background velocity, <b>(b)</b> true windowed band-limited reflectivity, <b>(c)</b> RTM image without multiple energy, <b>(d)</b> RTM image with multiple energy, <b>(e)</b> model R1 result based on workflow 1, <b>(f)</b> model R3 result based on workflow 3, <b>(g)</b> model R4 result based on workflow 4, and <b>(h)</b> true windowed velocity.	58
3.10	SEAM model results: <b>(a)</b> reflectivity from the background velocity, <b>(b)</b> true band-limited reflectivity, <b>(c)</b> RTM image without multiple energy, <b>(d)</b> RTM image with multiple energy, <b>(e)</b> model R1 result based on workflow 1, <b>(f)</b> model R3 result based on workflow 3, <b>(g)</b> model R4 result based on workflow 4, and <b>(h)</b> true SEAM velocity.	60
3.11	SEAM Phase 1 red box results: <b>(a)</b> reflectivity from the background velocity, <b>(b)</b> true windowed band-limited reflectivity, <b>(c)</b> RTM image without multiple energy, <b>(d)</b> RTM image with multiple energy, <b>(e)</b> model R1 result based on workflow 1, <b>(f)</b> model R3 result based on workflow 3, <b>(g)</b> model R4 result based on workflow 4, and <b>(h)</b> true windowed velocity.	62
3.12	Crossplots for the SEAM Phase 1 example: the true band-limited reflectivity against the predicted reflectivity by using the model R1 and R3, respectively.	63
3.13	<b>(a)</b> Model train loss and <b>(b)</b> validation loss comparison between different neural network models with fifty iterations.	65

4.1	Block diagram of DWT. . . . .	74
4.2	Architecture of DWT applied in the RTMM-CNN. The main structure is a U-Net with more skip connection layers. The input layer includes three channels: background reflectivity, RTMM image, and subband LL after using DWT of RTMM image. The predicted reflection coefficient is the output. . . . .	75
4.3	Neural network model plan and workflows. . . . .	76
4.4	Original image (a) was decomposed by Haar discrete wavelet transform and result (b) LL (c) LH (d) HL (e) HH were obtained by filter banks with interpolation. . . . .	79
4.5	Foothills model results. (a) Reflectivity from the background velocity, (b) true band-limited reflectivity, (c) RTM image without multiple energy, (d) RTM image with multiple energy, (e) model R1 result based on workflow 1, (f) model R3 result based on workflow 3, (g) model R4 result based on workflow 4, and (h) true Foothills velocity. . . . .	82
4.6	Foothills example 1 results. (a) Reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) true label, (e) DWT subband LL without multiple energy, (f) DWT subband LL with multiple energy, (g) model R4 result based on workflow 4, (h) model R1 result based on workflow 1, (i) model R3 result based on workflow 3, and (j) true windowed velocity. . . . .	87
4.7	Foothills example 2 results. (a) Reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) true label, (e) DWT subband LL without multiple energy, (f) DWT subband LL with multiple energy, (g) model R4 result based on workflow 4, (h) model R1 result based on workflow 1, (i) model R3 result based on workflow 3, and (j) true windowed velocity. . . . .	88
4.8	Overthrust model results. (a) Reflectivity from the background velocity, (b) true band-limited reflectivity, (c) RTM image without multiple energy, (d) RTM image with multiple energy, (e) model R1 result based on workflow 1, (f) model R3 result based on workflow 3, (g) model R4 result based on workflow 4, and (h) true Overthrust velocity. . . . .	89
4.9	Overthrust example 1 results. (a) Reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) true label, (e) DWT subband LL without multiple energy, (f) DWT subband LL with multiple energy, (g) model R4 result based on workflow 4, (h) model R1 result based on workflow 1, (i) model R3 result based on workflow 3, and (j) true windowed velocity. . . . .	90
4.10	Overthrust example 2 results. (a) Reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) true label, (e) DWT subband LL without multiple energy, (f) DWT subband LL with multiple energy, (g) model R4 result based on workflow 4, (h) model R1 result based on workflow 1, (i) model R3 result based on workflow 3, and (j) true windowed velocity. . . . .	91
4.11	Cross-plot comparison between results from models R1 and R3 for the Foothills model. . . . .	92

4.12	Cross-plot comparison between results from models R1 and R3 for the Overthrust model. . . . .	92
4.13	Training and validation loss comparison between models R1 and R3. . . . .	93
4.14	Foothills model results after adding noise. (a) Reflectivity from the background velocity, (b) true band-limited reflectivity, (c) RTM image without multiple energy, (d) RTM image with multiple energy, (e) model R1 result based on workflow 1, (f) model R3 result based on workflow 3, (g) model R4 result based on workflow 4, and (h) true Foothills velocity. . . . .	94
4.15	Foothills example 1 results after adding noise. (a) Reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) true label, (e) DWT subband LL without multiple energy, (f) DWT subband LL with multiple energy, (g) model R4 result based on workflow 4, (h) model R1 result based on workflow 1, (i) model R3 result based on workflow 3, and (j) true windowed velocity. . . . .	95
4.16	Foothills example 2 results after adding noise. (a) Reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) true label, (e) DWT subband LL without multiple energy, (f) DWT subband LL with multiple energy, (g) model R4 result based on workflow 4, (h) model R1 result based on workflow 1, (i) model R3 result based on workflow 3, and (j) true windowed velocity. . . . .	96
4.17	Overthrust model results after adding noise. (a) Reflectivity from the background velocity, (b) true band-limited reflectivity, (c) RTM image without multiple energy, (d) RTM image with multiple energy, (e) model R1 result based on workflow 1, (f) model R3 result based on workflow 3, (g) model R4 result based on workflow 4, and (h) true Overthrust velocity. . . . .	97
4.18	Overthrust example 1 results after adding noise. (a) Reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) true label, (e) DWT subband LL without multiple energy, (f) DWT subband LL with multiple energy, (g) model R4 result based on workflow 4, (h) model R1 result based on workflow 1, (i) model R3 result based on workflow 3, and (j) true windowed velocity. . . . .	98
4.19	Overthrust example 2 results after adding noise. (a) Reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) true label, (e) DWT subband LL without multiple energy, (f) DWT subband LL with multiple energy, (g) model R4 result based on workflow 4, (h) model R1 result based on workflow 1, (i) model R3 result based on workflow 3, and (j) true windowed velocity. . . . .	99
5.1	Example of a trace prepared for the neural network training. Primary reflections (red line) from three subsurface reflectors are observed separately at 0.15, 0.4 and 0.6 seconds. The first order of surface multiple reflections from the first and second reflectors are acquired at 0.3 and 0.8 seconds, shown in the dashed line. . . . .	105

5.2	Example of two windowed trace inputs for the neural network training. A shallow window (purple box) contains a primary reflection, and a deep window (blue box) consists of primaries and the first order of surface multiple reflections from the first reflector. . . . .	105
5.3	Prediction of the trace example. Comparisons between the predicted monitor (red line) by (a) shallow window and (b) deep window, and the observed monitor (dashed line). . . . .	106
5.4	Sketch for the recurrent neural network. . . . .	108
5.5	LSTM unit distribution. . . . .	108
5.6	Workflow for Bi-LSTM algorithm. . . . .	111
5.7	Stacked Bi-LSTM. . . . .	112
5.8	Workflow for predicting time-lapse reflectivity change. . . . .	115
5.9	Stacked Bi-LSTM with convolutional layers architecture. . . . .	116
5.10	Velocity model for (a) baseline and (b) monitor systems. . . . .	118
5.11	Horizontal-layered example. Four shots are selected to show data obtained from (a) observed baseline, (b) the observed monitor, (c) the predicted baseline by a shallow window and (d) the predicted baseline by a deep window. . . . .	119
5.12	Horizontal-layered example. Two shots are extracted from Figure 5.11 to show data differences between (a) the observed monitor and baseline, (b) the observed monitor and predicted baseline by a deep window, (c) the observed monitor and predicted monitor using a matching filter, and (d) the target difference excluding near-surface change in the observed monitor data. . . . .	120
5.13	Trace comparison for shots (a) No.3 and (b) No.10. . . . .	122
5.14	Horizontal-layered example's (a) true reflectivity difference. Migration differences are generated by (b) the difference between the observed monitor and baseline, (c) the double-difference method, and (d) U-Net prediction. (e) Target migration difference without near-surface change. . . . .	123
5.15	Velocity model for (a) baseline and (b) monitor systems. . . . .	124
5.16	Curve-fault example. Two selected shots of data differences between (a) observed monitor and baseline, (b) observed monitor and predicted baseline by a deep window, (c) observed monitor and predicted monitor using a matching filter, and (d) the target difference excluding near-surface change in the observed monitor data. . . . .	125
5.17	Trace comparison for shots (a) No.3 and (b) No.10. . . . .	126
5.18	Curve-fault example's (a) true reflectivity difference. Migration differences are generated by (b) the difference between the observed monitor and baseline, (c) double-difference method, and (d) U-Net prediction. (e) Target migration difference without near-surface change. . . . .	127
5.19	Velocity model for (a) baseline and (b) monitor systems. . . . .	128
5.20	Marmousi example. Two selected shots of data differences between (a) observed monitor and baseline, (b) observed monitor and predicted baseline by a deep window, (c) observed monitor and predicted monitor using a matching filter, and (d) the target difference excluding near-surface change in the observed monitor data. . . . .	128
5.21	Trace comparison for shots (a) No.5 and (b) No.10. . . . .	129

5.22	Marmousi example’s (a) true reflectivity difference. Migration differences are generated by (b) the difference between the observed monitor and baseline, (b) double-difference method, and (d) U-Net prediction. (e) Target migration difference without near-surface change. . . . .	130
5.23	Velocity model for Overthrust slice 1 model (a) baseline and (b) monitor systems. . . . .	133
5.24	Velocity model for Overthrust slice 2 model (a) baseline and (b) monitor systems. . . . .	133
5.25	Velocity model for Overthrust slice 3 model (a) baseline and (b) monitor systems. . . . .	134
5.26	Data differences of slice 1 between (a) observed monitor and baseline, (b) observed monitor and predicted baseline using a deep window, (c) observed monitor and predicted monitor using a matching filter, and (d) the target difference excluding near-surface change in the observed monitor data. . . . .	134
5.27	Data differences of slice 2 between (a) observed monitor and baseline, (b) observed monitor and predicted baseline using a deep window, (c) observed monitor and predicted monitor using a matching filter, and (d) the target difference excluding near-surface change in the observed monitor data. . . . .	135
5.28	Data differences of slice 3 between (a) observed monitor and baseline, (b) observed monitor and predicted baseline using a deep window, (c) observed monitor and predicted monitor using a matching filter, and (d) the target difference excluding near-surface change in the observed monitor data. . . . .	136
5.29	Overthrust slice 1 (a) true reflectivity difference. Migration differences are generated by (b) the difference between the observed monitor and baseline, (c) the double-difference method, and (d) U-Net prediction. (e) Target migration difference without near-surface change. . . . .	137
5.30	Overthrust slice 2 (a) true reflectivity difference. Migration differences are generated by (b) the difference between the observed monitor and baseline, (c) the double-difference method, and (d) U-Net prediction. (e) Target migration difference without near-surface change. . . . .	138
5.31	Overthrust slice 3 (a) true reflectivity difference. Migration differences are generated by (b) the difference between the observed monitor and baseline, (c) the double-difference method, and (d) U-Net prediction. (e) Target migration difference without near-surface change. . . . .	139
5.32	Velocity model for Overthrust slice 4 (a) baseline and (b) monitor systems. . . . .	140
5.33	Data differences of slice 4 between (a) observed monitor and baseline, (b) observed monitor and predicted baseline using a deep window, (c) observed monitor and predicted monitor using a matching filter, and (d) the target difference excluding near-surface change in the observed monitor data. . . . .	140
5.34	Overthrust slice 4 (a) true reflectivity difference. Migration differences are generated by (b) the difference between the observed monitor and baseline, (c) the double-difference method, and (d) U-Net prediction. (e) Target migration difference without near-surface change. . . . .	141
5.35	CaMI FRS DAS data acquisition location map in southern Alberta, Canada (Macquet et al., 2022). . . . .	143

5.36	VSP layout map of the survey acquired in March 2021 (Macquet et al., 2022; Wang and Lawton, 2022a). . . . .	144
5.37	Wavefield separation: (a) raw DAS data from straight fibre in OBS2, and isolated upgoing wavefield after (b) median filter and then (c) <i>F-K</i> filter. BBRS reflection is indicated in plot (c) (Wang and Lawton, 2022a). . . . .	144
5.38	DAS VSP stacked data from CaMI FRS. (a) Observed baseline (b) Observed monitor after calibration. (c) Windowed observed baseline. (d) Windowed observed monitor. (e) Windowed observation difference. (f) Windowed predicted CO <sub>2</sub> injection area. . . . .	146
5.39	Trace comparison for DAS VSP data CDP (a) 50 and (b) 70. . . . .	147

# List of Tables

3.1	UNet Architecture - Encoding. . . . .	38
3.2	UNet Architecture - Decoding. . . . .	39
3.3	PSNR (dB) comparison for Foothills example. . . . .	50
3.4	PSNR (dB) comparison for Overthrust example. . . . .	59
3.5	PSNR (dB) comparison for SEAM example. . . . .	61
4.1	PSNR (dB) comparison for Foothills example. . . . .	81
4.2	PSNR (dB) comparison for Overthrust example. . . . .	83
4.3	PSNR (dB) comparison for Foothills example with noise added. . . . .	85
4.4	PSNR (dB) comparison for Overthrust example with noise added. . . . .	85
5.1	PSNR (dB) comparison for synthetic examples. . . . .	129

# List of Symbols, Abbreviations, and Nomenclature

Symbol or abbreviation	Definition
$a$	Coarse approximation of a signal
arg min	Argument of the minimum
$\mathbf{b}_f$	Bias
BBRS	Basal Belly River Sandstone Formation
Bi-LSTM	Bidirectional long short-term memory
BRNN	Bidirectional recurrent neural network
$c$	Detailed information of a signal
$\mathbf{C}_t$	Current cell state
CaMI	The Containment and Monitoring Institute
CDP	Common depth point
CGLS	Conjugate gradient least-squares
CMC	Carbon Management Canada
CNN	Convolutional neural network
$\mathbf{d}$	Observed seismic data
$\mathbf{d}_{calc}$	Predicted shot record data
$\mathbf{d}_{calcba}$	Predicted baseline data by a deep window

$\mathbf{d}_{calcb_s}$	Predicted baseline data by a shallow window
$\mathbf{d}_{obs}$	Observed shot record data
$\mathbf{d}_{obsb}$	Observed baseline data
$\mathbf{d}_{obsm}$	Observed monitor data
$D$	Source wavefield/Downgoing wavefield
DAS	Distributed acoustic sensing
dB	Decibel
$\delta\mathbf{d}(t)$	Residual between monitor and baseline data
$dx$	Horizontal space interval
$dz$	Vertical space interval
$dt$	Time interval
DWT	Discrete wavelet transform
$e$	Mathematical constant
$f$	Frequency
$f_{dom}$	Dominant frequency
$\mathbf{f}_t$	Forget gate
$F-K$	Frequency-wavenumber
$f(\mathbf{R})$	Cauchy function criterion
$f(\mathbf{R}')$	Derivative of Cauchy function criterion
$f(t)$	Function of time
FWM	Full-wavefield migration
$F-X$	Frequency-space
$g$	High pass filter
$\mathbf{g}_t$	Candidate gate
$h$	Low pass filter
$\mathbf{h}_{t-1}$	Hidden layer vector from the previous time step
$\mathbf{h}_t^n$	Network output of the time step $t$ at each unit $n$

$i$	Imaginary unit
$I$	Imaging condition
$\Delta I$	Change in the reflectivity coefficient
$\mathbf{i}_t$	Input gate
$iter$	Number of iterations
$j$	Integer
$k_x$	Horizontal wavenumber
$k_z$	Vertical wavenumber
$\bar{k}_z$	Vertical wavenumber from reference velocities
$\mathbf{L}^T$	Adjoint operator
LI	Linear interpolation
$LL$	Low frequency of a wavelet image coefficient
LSTM	Long short-term memory
LSRTM	Least-squares reverse time migration
LSM	Least-squares migration
$m$	Integer
$\mathbf{m}$	Reflectivity in Chapter 3
$\mathbf{m}^*$	Predicted reflectivity
$\delta\mathbf{m}_{calc}$	Model changes of predicted data
$\mathbf{m}_{mig}$	Migrated reflectivity
$\delta\mathbf{m}_{obs}$	Model changes of observed data
$\mathbf{m}_{pred}$	Output predicted reflectivity
$\mathbf{m}_{pred_1}$	Predicted reflectivity from workflow 1
$\mathbf{m}_{pred_2}$	Predicted reflectivity from workflow 2
$\mathbf{m}_{pred_3}$	Predicted reflectivity from workflow 3
$\mathbf{m}_{pred_4}$	Predicted reflectivity from workflow 4
$\mathbf{m}_{rtm\_scenario1}$	RTM image from scenario 1 in Chapter 3

$\mathbf{m}_{rtm\_scenario2}$	RTM image from scenario 2 in Chapter 3
$\mathbf{m}_{rtm\_scenario3}$	RTMM image from scenario 3 in Chapter 3
$\mathbf{m}_{rtm\_scenario4}$	RTMM image from scenario 4 in Chapter 3
$\mathbf{m}_{rtmm\_DWT}$	DWT subband on RTMM image
$\mathbf{m}_{smooth}$	Smoothed reflectivity
$\delta\mathbf{m}_{time-lapse}$	Time-lapse model changes
$\mathbf{m}_{true}$	True reflectivity
$M$	Multiple reflection
$M_I$	Number of input units
$M'$	Secondary multiple reflections
$M'_B$	Back-propagated receiver wavefields
$M_F$	Forward-extrapolated multiples
$MAX_I$	Maximum possible pixel value
MSE	Mean squared error
$n$	Integer
$\mathbf{n}(t)$	Noise generated from near-surface
$N_H$	Number of hidden units
$N_f$	Number of frequencies
$N_t$	Number of time sampling
$N_x$	Number of horizontal distance points
$N_z$	Number of depth points
NSPS	Non-stationary phase shift
$\mathbf{o}_t$	Output gate
$O$	An objective function
$P$	Primary reflection
$\mathbf{P}^+$	Predicted downgoing wave in the $F$ - $X$ domain
$\mathbf{P}^-$	Predicted upgoing wave in the $F$ - $X$ domain

$\Delta\mathbf{P}^-$	Back-propagated upward data residual
$\vec{P}^+$	Downward extrapolated wavefield
$\vec{P}^-$	Upward extrapolated wavefield
$P_F$	Forward-extrapolated primaries
$\mathbf{P}_{obs}$	Observed data
PSNR	Peak signal-to-noise ratio
PSPI	Phase shift plus interpolation
$R$	Reflection coefficient
$\delta\mathbf{r}(t)$	Subsurface reservoir variation
$\mathbf{R}^U$	Reflectivity of layer's above
$\mathbf{R}^D$	Reflectivity of layer's below
ReLU	Rectified linear unit
RMSE	The root mean squared error
RNN	Recurrent neural network
RTM	Reverse time migration
RTMM	Reverse time migration with multiple reflections
$\delta\vec{S}$	Secondary sources or the scattering terms
$\vec{S}^+(z_0)$	A source on the surface
$\Delta S_D$	Downgoing scattering term
$\Delta S_U$	Upgoing scattering term
SD-LSTM	Stacked long short-term memory
$sgn$	The sign of a real number
$t$	Time
$t_{max}$	Maximum of recorded time
$tanh$	Hyperbolic tangent activation function
$U$	Receiver wavefield/Upgoing wavefield
$v$	Velocity

VSP	Vertical seismic profiling
$w$	Ricker wavelet
$w_{mn}$	The weight of the input $m$ to hidden unit $n$
$w_{n'n}$	The weight of hidden $n$ towards hidden $n'$
<b>W</b>	Extrapolation operator
$\mathbf{W}_f$	Weight matrices
$x$	Horizontal position coordinate
$\mathbf{x}_t$	Current step sequence input
$y_L$	Low-pass subband
$y_H$	High-pass subband
$y_t$	Prediction after using an activation function
$z$	Vertical position coordinate
$\Delta z$	Change in the vertical position coordinate
$Z$	Impedance
$\alpha$	Scaling function translation
$\alpha_t^n$	The activation function of hidden $n$ at time $t$
$\alpha_{v(x)}$	Shift operator
$\beta$	Importance of backward LSTM
$\gamma$	Importance of forward LSTM
$\Gamma^{-1}$	The inverse of the Hessian
$\Gamma_{unet}$	An approximation of the inverse of the Hessian
$\Gamma_{unet\_workflow2}$	The neural network trained on workflow 2
$\Gamma_{unet\_fine\_tuned\_workflow2}$	The pre-trained neural network from workflow 2
$\Gamma_{unet\_workflow4}$	The neural network trained on workflow 4
$\Gamma_{unet\_fine\_tuned\_workflow4}$	The pre-trained neural network from workflow 4
$\epsilon$	Stabilized factor
$\lambda$	Scale parameter

$\omega$	Angular frequency
$\psi(t)$	Mother wavelet
$\phi(t)$	Scaling function
$\rho$	Density
$\sigma$	Logistic sigmoid activation function
$\tau_L$	The length of the low-pass filter
$\tau_H$	The length of the high-pass filter
$\Theta_n$	The activation function of the hidden unit $n$
1-D	One-dimensional
2-D	Two-dimensional
4-D	Four-dimensional
$\infty$	Infinity
$(*)^H$	Complex conjugate
$\ \dots\ _2^2$	Squared norm

# Epigraph

*A prayer without a deed is an arrow without a bowstring; A deed without a prayer is a bowstring without an arrow.*

- Ella Wheeler Wilcox, *Arrow and Bow*

# Chapter 1

## Introduction

### 1.1 Seismic primary and multiple reflections

In seismic exploration, primary waves travel paths with a single reflection at the subsurface interface between the source and receiver, while multiple reflections follow more complicated travel paths with two or more reflections between the source and receiver as they travel underground (Sengbush, 1983; Hu, 2016). Under the situation that only the P-P wave is considered and the converted wave is excluded in this thesis, seismic waves are split into reflected and transmitted waves at each acoustic discontinuity, and as they travel through the subsurface, they generate multiple reflections. Those multiple reflections have multi-reflected and transmitted waves re-vibrated along the travel path. For example, a primary reflection generated from the source on the surface is shown in Figure 1.1 red line, and the first-order multiple simulated from the same source is denoted in the blue line. Even though multiples are considered noise in traditional seismic exploration, they can bring additional reflection information and signals from a complex subsurface structure. A simple sketch for explaining the idea is shown in Figure 1.2, where the coverage length of the multiple reflections is larger than the one for the primaries with the exact shot-receiver coordinates.

Multiple reflections in seismic exploration can be categorized into two types: surface-

related and internal multiples. Free-surface conditions in acoustic wave modeling (Piroux and Lombard, 2001) generate surface-related multiples, typically exhibiting robust reflection coefficients between the air and sea level, and sea bed. On the shot record, they show repeated wave shapes that follow primaries after two-way travel time. In marine data, surface-related multiples can have strong and significant amplitudes, often overwhelming internal multiples. In contrast, internal multiples are generated from reflections reverberated between subsurface interfaces with different rock properties. They are more noticeable in land data because they indicate the presence of layers with high and low-velocity contrasts. Although either surface or internal multiples' amplitudes may be apparent in different cases, the problem with multiple reflections is their interference with primaries, which can obscure vital information about subsurface reflectors in a limited-time recording. Also, multiples have complications if the reflector exhibits roughness. Assumptions and prerequisites for different imaging methods further limit the processing of multiple reflections.

This thesis introduces novel methodologies to address the challenge of achieving high resolution for spatial variability in seismic imaging by using seismic multiple reflections. These methods encompass conventional techniques, as well as innovative machine-learning approaches. The former contains seismic multiple applications in phase shift plus interpolation (PSPI) migration. The latter uses convolutional neural networks and long short-term memory networks on migrated images with multiple reflections added. These methods aim to enhance the resolution and accuracy of seismic imaging results by harnessing the power of multiple reflections. By combining the strengths of both conventional and machine learning-based techniques, this research aims to provide a comprehensive framework for exploiting multiple reflections and achieving a high level of spatial resolution in seismic imaging.

## **1.2 Multiple applications in conventional migration**

For the past three decades, two directions have been worked on multiple reflections in seismic imaging processing (Weglein, 1995). The first direction involves multiple attenuation

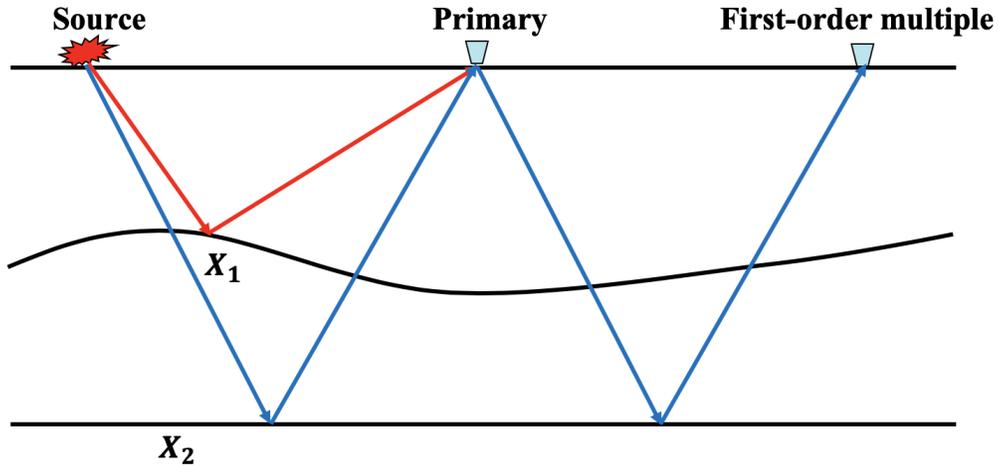


Figure 1.1: Propagation paths of primary (red line) and the first-order multiple (blue line) reflections.

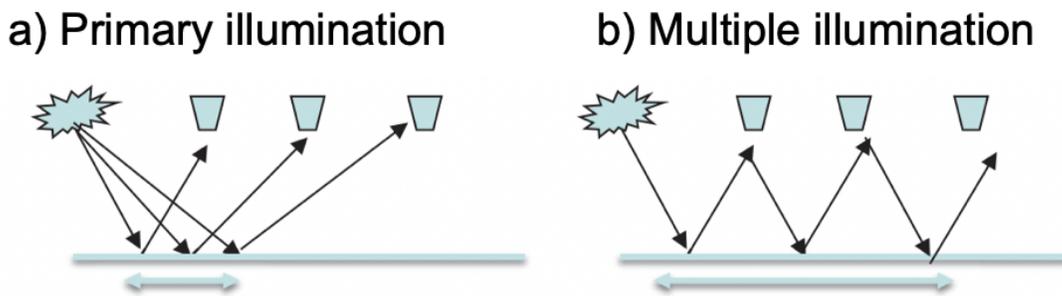


Figure 1.2: Comparison of (a) primary and (b) multiple reflection illumination.

approaches (Wiggins, 1988; Verschuur et al., 1992; Araújo et al., 1994; Weglein, 1995; Weglein and Dragoset, 2005), which are classified into two categories: select and separate multiples from primaries, and predict and subtract multiples. Utilizing multiple reflections, as the other direction, has grown in recent years to reveal multiples or limited orders of multiples that can be used as signals instead of noise in seismic imaging (Berkhout and Verschuur, 1994; Malcolm et al., 2008; Whitmore et al., 2010; Lu et al., 2011; Liu et al., 2011; Zhang and Schuster, 2014; Davydenko and Verschuur, 2017; Lu, 2021). Chapter 2 works in the second direction, to show the application of multiple reflections in seismic imaging, by using a scattering term in the PSPI migration.

### 1.2.1 Incorporating multiple reflections in phase shift plus interpolation migration

Phase shift plus interpolation (PSPI) (Gazdag and Sguazzero, 1984) is an adaptation of phase shift migration (Gazdag, 1978) to accommodate lateral velocity variations. Traditional PSPI migration separately extrapolates source and receiver wavefields downwards through phase shift propagation and interpolation. An imaging condition is applied at each depth to combine those two wavefields to estimate reflection coefficients. While PSPI is a widely used seismic imaging method that offers advantages in terms of computational efficiency and imaging accuracy, the issue of limited illumination aperture can pose limitations in accurately capturing the subsurface structure.

The limited illumination aperture problem occurs when there are areas in the subsurface where the seismic source cannot adequately generate reflections to reach the edges. This results in areas of poor image quality, making it difficult to interpret the subsurface structure accurately. A scattering term can be incorporated into seismic imaging methods to broaden the subsurface illumination (Berkhout and Verschuur, 1994; Berkhout, 2014). The scattering term that generates multiple reflections (Berkhout, 2014; Davydenko and Verschuur, 2017) can be used to reveal additional information about the subsurface.

In Chapter 2, I propose a method that uses PSPI with a scattering term to generate multiples artificially. They do not exist but help extend information by additional reflections. The scattering term mentioned above is determined when calculating the wavefield propagation. It generates reflection and transmission differences at each depth layer, or the secondary wavefields. One benefit is that the input data remains unchanged. There is no need to separate and correlate different orders of multiples during migration. Moreover, the computational cost of this method is relatively small, making it applicable to both dense and coarse acquisition systems. Some synthetic examples explore the benefit of the scattering term to the PSPI.

### 1.3 Multiple reflections in migration combined with machine learning method

In recent years, machine learning has emerged as a popular approach for seismic processing. Neural networks (Caudill, 1987) are particularly useful for their ability to adapt and learn highly non-linear relationships in seismic data (Bergen et al., 2019; Seydoux et al., 2020; Mustafa et al., 2021), making them a valuable tool for improving accuracy and suppressing artifacts in a variety of processing applications. Many machine learning-based methods have been developed and applied to seismic event detection, noise attenuation, trace interpolation, seismic impedance inversion, and multiple eliminations in seismic data processing. These methods utilize various types of neural networks, including generative adversarial networks (GANs) (Alwon, 2018; Kaur et al., 2020c), convolutional neural networks (CNNs) (Siahkoobi et al., 2019b; Liu et al., 2020; Jiang et al., 2021; Gu et al., 2022), and recurrent neural networks (RNNs) (Guo et al., 2019; Birnie and Hansteen, 2022). The applications of these methods in seismic data processing have shown promising results in improving the quality of seismic images, reducing processing time, and facilitating the accurate interpretation of subsurface structures.

Migration is a process that re-locates the seismic events in the true space or time location, using data and a background velocity model, to form a subsurface structure of the reflectivity (Yilmaz, 2001). Because of insufficient sampling and uncertainty in the velocity information, conventional migration methods often yield insufficient imaging illumination and resolution, making the structure interpretation ambiguous and inaccurate. Recently, machine learning techniques have been extensively used in seismic processing to address these limitations. In contrast to conventional seismic migration methods, neural networks depend less on restricting assumptions concerning the underlying model. Moreover, neural networks are data-driven, and adaptive and can learn highly non-linear relations in seismic data processing, making them well-suited for improving accuracy and suppressing artifacts.

This freedom comes at the cost of requiring more data for different geology models, and computational costs for training. Convolutional neural networks, as one category of machine learning approach, have been used in seismic imaging to reconstruct the subsurface structure accurately (Wrona et al., 2018; Lu et al., 2020; Liu et al., 2020; Torres and Sacchi, 2021, 2022). This thesis adopts U-Net (Ronneberger et al., 2015), a convolutional neural network, into the Reverse Time Migration (RTM) (Baysal et al., 1983) image to determine an improved reflectivity prediction. By incorporating machine learning approaches, I seek to overcome the challenges associated with traditional migration methods, such as inaccuracies due to wavefield distortions, limited resolution, and the inability to effectively capture complex subsurface features. The goal is to enhance the accuracy and resolution of subsurface imaging by leveraging the potential of novel techniques that exploit multiple reflections and harness the power of machine learning.

To take advantage of the broad subsurface information from multiple reflections, I use surface-related multiples generated from the free surface boundary condition in the RTM to add more information about subsurface reflection events. Chapter 3 outlines a U-Net-based workflow that uses the energy from multiples to enhance the accuracy of RTM. The crosstalk noise is the interference between two or more signals (Bhasker and Chadha, 2009). By training the network to distinguish between true reflectivity events and crosstalk noise, the neural network method can effectively reduce the impact of the noise on the migration images. Based on this idea, Chapter 4 explores using the discrete wavelet transform (DWT) (Gonzalez and Woods, 2002) subband of migrated images to improve image quality further. By adding low-frequency DWT subband images to the input of the network, the method can introduce physical constraints that help the network extract key features from low-frequency information and tolerate more artifacts in the presence of smooth inputs.

Other than image processing, neural networks are another common use for predicting time series with inputs. That is the way I used them to address the problem of dynamic changes in reservoirs via time-lapse (Lumley, 2001) in Chapter 5. In reservoir monitoring,

time-lapse seismic data is commonly used to detect changes in fluid content in the subsurface at different times over the same area. The machine learning approach can address this task by predicting the difference in seismic data between different time records, which can then be used to derive the reservoir anomaly space through seismic processing. Other researchers also explore machine-learning approaches for time-lapse seismic data analysis. For instance, Hussein et al. (Hussein et al., 2021) propose an unsupervised machine learning method for detecting changes in the subsurface, while Li et al. (Li et al., 2021) use machine learning to monitor CO<sub>2</sub> injection in a reservoir. Jun et al. (Jun and Cho, 2022) investigate the repeatability of time-lapse seismic data and propose a deep learning-based approach to improve the repeatability of seismic images. Alali et al. (Alali et al., 2022) use machine learning to analyze time-lapse seismic data for reservoir characterization. Different from the research above, some innovative steps in this thesis are delineated below.

In Chapter 5, I present a method based on a recurrent neural network, specifically the long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) architecture, to predict time-lapse shot record data from base data with multiple reflections. The predicted difference data can then be used to derive the reservoir anomaly space. To further improve the accuracy and quality of the migration images, I present a U-Net network to enhance the migrated images with the predicted difference data. Subsequently, the proposed methodology is implemented on field data of the vertical seismic profiling (VSP) (Gal’Perin, 1974; Wyatt, 1981; Lines et al., 1984; Gilpatrick and Fouquet, 1989; Sheriff and Geldart, 1995; Lines and Newrick, 2004) acquired through distributed acoustic sensing (DAS) (Mestayer et al., 2011; Daley et al., 2013; Karrenbach et al., 2019; Innanen et al., 2019). The goal is to demonstrate how the predicted CO<sub>2</sub> injection image can be improved using this approach. VSP is a commonly used subsurface characterization and monitoring technique, particularly in seismic exploration. The DAS technique, which uses fibre-optic cables as sensors, offers the advantage of high spatial resolution and continuous monitoring. However, the acquired data are often noisy and contaminated with various types of noise sources, for example,

water layer, soil and seasonal changes. Those changes will influence near-surface velocity distribution and acquisition setting, making it challenging to accurately predict CO<sub>2</sub> injection images. Therefore, the proposed method can provide a viable solution to improve the quality of the predicted image in DAS VSP data. The details of the methodology and the results are presented in Chapter 5.

In a summary of the machine learning application of using multiple reflections in seismic imaging, Chapters 3-5 provide a new way to separate noise and multiples by integrating machine learning algorithms. Leveraging the power of U-Net and stacked LSTM architectures, the thesis demonstrates the efficacy of these network models in forecasting migration images.

## 1.4 Thesis overview and objectives

Multiple reflections are regarded as noise in traditional seismic processing methods. With some additional effort, however, they could play an essential role in providing extended illumination and enhanced resolution. Because of their multiple travel paths, they are more sensitive to velocity variations. In this thesis, I review and develop effective approaches to exploit the benefits of multiples instead of removing them to improve migration images.

The first portion of this thesis aims to investigate adding scattering terms containing multiples in the PSPI migration. Then, I develop a machine learning workflow using neural networks to optimize reverse time migration results. In particular, to distinguish multiple reflections contribution from noise or artifacts in reflectivity prediction, I propose using neural networks to handle the non-linear problem and extract critical features from noisy migration images. Additional input channels containing the reflectivity approximated from the migration background velocity, RTM images, and their discrete wavelet transforms subbands help add physics constraints on training and prediction. This approach is later applied to CO<sub>2</sub> monitoring by using multiples in both monitor and base data. I propose a way to learn data changes from baseline data and use the U-Net to predict subtle changes in reservoir properties with enhanced image quality.

In Chapter 2, I investigate adding scattering terms to the PSPI migration. I present a workflow of PSPI migration with a scattering term added to the wavefield propagation process to provide a broad illumination aperture. Some multiples are additionally generated after adding the scattering term, giving comprehensive information about the subsurface reflectivity.

In Chapter 3, I propose a machine learning workflow that uses a pre-trained convolutional neural network to work on reverse time migration with multiple reflections included. This workflow is designed to extract features from migrated images and background reflectivity models. The network can then predict a sharper reflectivity model compared to the initial migration images. I begin by describing four scenarios that happen in reverse time migration and explain the issue of wavefield inconsistency. To overcome the problem in the RTM with multiple reflection energy, a U-Net with additional skip connections is considered to approximate the effect of the inverse of the Hessian. I use a first network pre-trained with true reflectivities as labels and a second network trained with smooth labels as a way to fine-tune the training parameters by selecting within an acceptable range. An essential contribution of Chapter 3 is that the total wavefield, including primaries, surface multiples and internal multiples, is used in the neural network model that can learn patterns of structured reflection layers and distinguish signals from artifacts or noise. One advantage of this approach is having a lower computational cost than other techniques based on accounting for the extra traveltimes during wave propagation. This is because the whole process does not need to calculate different orders of multiples and their corresponding imaging condition. The trained neural network can be applied to other geology structural models that did not participate during training (inference).

In Chapter 4, I investigate adding another input channel to the network, a discrete wavelet transform of initial migration images. The goal is to add more physical constraints to the training. One advantage of using machine learning workflow is to have a low computational cost, which means the whole process does not need to calculate different orders of multiples or

compute the imaging condition from those orders of multiples. The other benefit is that the trained neural network can be used in other geology structure models. Model generalization works for thin-layered and complex subsurface models. Gaussian noise is also considered at the end of this chapter to test the network duration on smoothing and random noise.

In Chapter 5, I apply machine learning to the problem of time-lapse monitoring with the purpose of using the energy of multiple reflections. I investigate a stacked long short-term memory network and a convolutional neural network to predict the reflectivity change between baseline and monitor data in a reservoir. I start with fundamental knowledge of LSTM that can work with shot records in the geophysical application. Then, I propose a systematic workflow working on each baseline trace and monitoring data above the reservoir anomaly area. The stacked LSTM network estimates updated monitor data from the observed baseline. The CNN is used for predicting the reflectivity of the target area change with suppressed artifacts and enhanced resolution. Then, I discuss the difference and uncertainty of different situations and shapes of reservoir change in this workflow.

In Chapter 6, I summarize the key contributions of this thesis and address some limitations and uncertainties of machine learning applications in migration. I also suggest possible future work directions at the end of the chapter.

In summary, the goal of this thesis, discussed in detail in chapters 2-5, is to address the following ideas:

- Multiples generated by the scattering term and the free surface boundary condition have the advantage of improved subsurface illumination. This makes them valuable in increasing the resolution of migration images. Novel migration techniques introduced in this thesis leverage the power of multiple reflections. The method surpasses the limitations of conventional migration approaches by enhancing migrated image quality.
- Convolutional neural network-based reverse time migration can be trained in data with multiples (input) and ideal migration results (labels) to distinguish signals from artifacts and predict reflection coefficients with improved resolution and accuracy. The proposed

methodology effectively handles wavefield discontinuities, and provides a robust and reliable approach for subsurface imaging.

- The discrete wavelet transform of migrated images can be used to create a useful additional input channel for the process discussed in the previous chapter.
- A pre-trained neural network provides a tool to enforce parameters inside reasonable ranges to fine-tune networks, make them stable, and increase their generalization power for applicability to other models.
- Time series prediction tools by machine learning can help time-lapse monitoring. By training in areas without reservoir changes, they can learn the patterns of near-surface data changes of the monitor and baseline surveys. These patterns can be used to suppress artifacts in the later migration process. Additionally, surface-related multiples in data can help provide more information about deep subsurface structures because of their long travel paths. The proposed method works on field DAS VSP data with high reliability and accuracy. This result demonstrates its potential for practical implementation in real-world carbon capture scenarios.

# Chapter 2

## Scattering terms in wavefield propagation for migration

### 2.1 Phase shift plus interpolation migration (PSPI) with scattering terms

The phase shift plus interpolation (PSPI) (Gazdag and Sguazzero, 1984) algorithm is useful for solving the scalar wave equation. One common issue with PSPI migration is that, like other migrations, it suffers from aperture limitations (Alsdorf, 1997; Schleicher et al., 1997; Sun, 2000; Schulte, 2012). In this chapter, I propose an approach that solves the limited illumination in migration by adding scattering terms in the phase shift wavefield. This change does not require modifying the shot records, but iteratively adds the scattering term for each reflector when stepping the source and received wavefields downward into the subsurface. Numerical examples, tested in simulated horizontal layers and boxcar shape structures, provide compelling evidence of the effectiveness of these terms in alleviating the aperture limitation issue, particularly in the context of horizontal events. By incorporating these terms, the reflector edge illumination is significantly extended, resulting in improved

imaging of subsurface structures.

### 2.1.1 Theory

In this subsection, the algorithm of PSPI migration with scattering terms is explained, along with a basic framework shown in Figure 2.1.

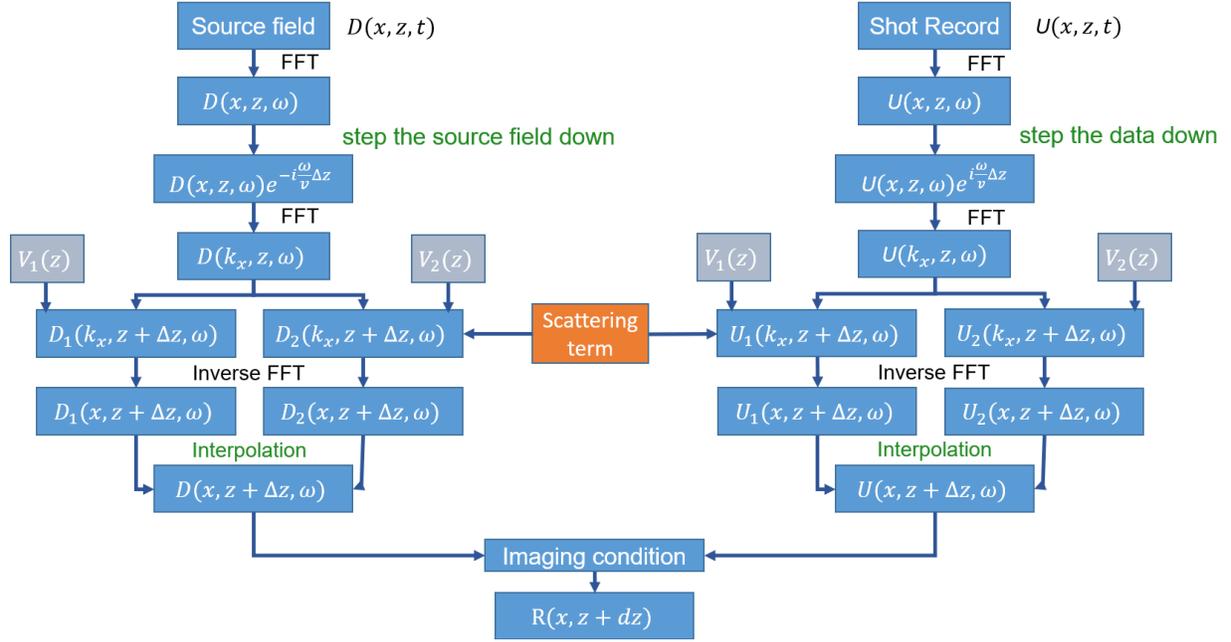


Figure 2.1: Workflow for PSPI migration with scattering terms.

### PSPI migration

Traditional PSPI migration separately extrapolates source and receiver wavefields downwards through phase shift propagation and interpolation (Gazdag and Sguazzero, 1984). A source wavefield is generated at the seismic source and propagates in the subsurface before any discontinuity interactions. A receiver wavefield originates at discontinuities and travels to the receivers. At each depth, an imaging condition, which indicates two wavefields coincide at discontinuities, estimates reflection coefficients (Berkhout, 1982; Claerbout, 1984; Sava and Vlad, 2011). Inspired by scattering terms used in the full wavefield migration (Davydenko

and Verschuur, 2017), in this proposed approach, whose workflow is shown in Figure 2.1, a scattering term is added in the phase shift extrapolation for upward and downward directions, to generate additional reflection and transmission. For example, in order to obtain the receiver wavefield at each depth layer, the receiver wavefield needs to propagate downwards. A novel approach is employed whereby, in addition to utilizing the acquired shot data, the source wavefield computed from the preceding depth level is incorporated. This integration of the source wavefield helps determine the receiver wavefields' current depth as they are stepped downwards. Then, the imaging condition shown in Figure 2.1 is applied using the total of original wavefields and the secondary scatters mentioned above. Next, I show a detailed description of this workflow.

The first step of this method is determining the downward extrapolation of the source wavefield  $D(x, z, t)$  and the receiver wavefield  $U(x, z, t)$  (that is, a shot record). After an initial Fourier transform over temporal space, the source and receiver wavefields at depth  $z$  are denoted as  $D(x, z, \omega)$  and  $U(x, z, \omega)$  respectively, where  $\omega$  denotes frequency.

Using the nonstationary phase shift method and (Margrave, 1998; Margrave and Ferguson, 1999; Ferguson and Margrave, 2005), the source and receiver wavefields can be stepped down by a one-way wavefield extrapolation propagator through laterally varying velocity  $v(x)$ :

$$D(k_x, z + \Delta z, \omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} D(x, z, \omega) \alpha_{v(x)}(k_x, x, \omega) e^{-ik_x x} dx, \quad (2.1)$$

$$U(k_x, z + \Delta z, \omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} U(x, z, \omega) \alpha_{v(x)}(k_x, x, \omega) e^{-ik_x x} dx. \quad (2.2)$$

According to Margrave and Ferguson (1999), the shift operator  $\alpha_{v(x)}(k_x, x, \omega)$  is defined as

$$\alpha_{v(x)}(k_x, x, \omega) = \begin{cases} e^{i\Delta z k_z(x)}, & |k_x| \leq \frac{\omega}{v(x)} \\ e^{-|\Delta z k_z(x)|}, & |k_x| > \frac{\omega}{v(x)} \end{cases}, k_z(x) = \sqrt{\frac{\omega^2}{v(x)^2} - k_x^2}, \quad (2.3)$$

where  $v(x)$  means the medium velocity. The operator ensures that an exponential decay

rapidly attenuates evanescent energy, which is generated from seismic waves reflected from an interface at a post-critical angle.  $k_x$  and  $k_z$  represent horizontal and vertical wavenumber respectively. Then, an inverse Fourier transform moves the extrapolated wavefields from the wavenumber to the spatial domain. When the actual velocity equals the reference velocity, the desired wavefield extrapolation is assumed to be equivalent to a reference wavefield. The source wavefield  $D(x, z + \Delta z, \omega)$  and receiver wavefield  $U(x, z + \Delta z, \omega)$  can be given:

$$D(x, z + \Delta z, \omega) = \int_{-\infty}^{\infty} D(k_x, z + \Delta z, \omega) e^{ik_x x} dk_x, \quad (2.4)$$

$$U(x, z + \Delta z, \omega) = \int_{-\infty}^{\infty} U(k_x, z + \Delta z, \omega) e^{ik_x x} dk_x. \quad (2.5)$$

### The role of scattering terms

Figure 2.1 shows that scattering terms are added to the phase shift propagation. The PSPI wavefield propagation chooses a set of reference velocities, including minimal and maximal extremes of medium velocity  $v(x)$  and sampling the fluctuations. Then, the desired extrapolation wavefields are approximated from the set of reference wavefields based on reference velocities (Margrave and Ferguson, 1999), and scattering terms ( $\Delta S_D$  and  $\Delta S_U$ ), can be determined by:

$$D_{v(x)+S_D}(x, z + \Delta z, \omega) = D_{v(x)}(x, z + \Delta z, \omega) + \Delta S_D, \quad (2.6)$$

$$U_{v(x)+S_U}(x, z + \Delta z, \omega) = U_{v(x)}(x, z + \Delta z, \omega) + \Delta S_U. \quad (2.7)$$

The phase shift velocity  $v(x)$  in downgoing  $D_{v(x)}$  and upgoing  $U_{v(x)}$  waves are approximated from a small set of reference velocities  $(v_j, v_{j+1})$  (Bagaini et al., 1995), where  $j$  is an integer and starting from 1. The maximum limit depends on the number of reference velocities chosen at that depth level. A linear interpolation is applied on the phase and amplitude of  $D_{v(x)+S_D}$  and  $U_{v(x)+S_U}$ , respectively, to determine the PSPI propagation output at each

depth layer.

Downward  $\Delta S_D$  and upward  $\Delta S_U$  scattering terms shown in equations 2.6 and 2.7 do not influence or change the raw input wavefields, but are generated as the secondary wavefields. They provide wavefield differential transmission (Berkhout, 2014; Davydenko and Verschuur, 2017), which can generate internal multiples that help to recover subsurface illumination. In this chapter, an underlying assumption is the utilization of acoustic media for wavefield propagation and migration under an angle-independent case. Next, reference velocities from the previous depth layer are stored for computing the scattering terms  $\Delta S_D$  and  $\Delta S_U$ :

$$\Delta S_D = \mathcal{F}^{-1} \left( R(k_x, z + \Delta z) D(k_x, z + \Delta z, \omega) - R(k_x, z + \Delta z) U(k_x, z + \Delta z, \omega) \right), \quad (2.8)$$

$$\Delta S_U = \Delta S_D, \quad (2.9)$$

where  $\mathcal{F}^{-1}$  means inverse Fourier transform on wavenumber domain. In the parentheses, the terms  $R(k_x, z + \Delta z) D(k_x, z + \Delta z, \omega)$  and  $R(k_x, z + \Delta z) U(k_x, z + \Delta z, \omega)$  denote multiplication. They show the reflection effect when downgoing and upgoing wavefields reach depth layer  $z + \Delta z$ . The reflectivity coefficient  $R(k_x, z + \Delta z)$  is given by

$$R(k_x, z + \Delta z) = \frac{(\omega \rho(x, z + \Delta z)) / k_z(\omega, z + \Delta z) - (\omega \rho(x, z)) / k_z(\omega, z)}{(\omega \rho(x, z + \Delta z)) / k_z(\omega, z + \Delta z) + (\omega \rho(x, z)) / k_z(\omega, z)}, \quad (2.10)$$

where  $\omega$  denotes the temporal frequency and  $\rho$  represents layer density. The vertical wavenumber  $k_z(\omega, z)$  and  $k_z(\omega, z + \Delta z)$  are determined from previous and current layer reference velocities, respectively.

## Imaging condition

Based on Claerbout (1971), a 1D deconvolution imaging condition is

$$I(z) = \frac{1}{2\pi} \int_{\omega} \frac{U(z, \omega)}{D(z, \omega)} d\omega, \quad (2.11)$$

where  $z$  indicates current depth layer,  $\omega$  means frequency. Time domain deconvolution corresponds to division in the frequency domain. So, the deconvolution imaging condition, after integrating over all frequencies, determines the zero-lag reflection coefficient where the interfaces are located. The deconvolution imaging condition provides improved resolution by reducing crosstalk and allowing the downgoing  $D(z, \omega)$  and upgoing  $U(z, \omega)$  wavefields to match kinematically at subsurface discontinuities (Claerbout, 1971; Valenciano and Biondi, 2003; Poole et al., 2010). For the 2D situation, the deconvolution imaging condition (Valenciano and Biondi, 2003) is defined with a stabilized reflectivity estimate:

$$I(x, z) = \sum_{\omega} \frac{D_{v(x)}(x, z, \omega)^* U_{v(x)}(x, z, \omega)}{D_{v(x)}(x, z, \omega)^* D_{v(x)}(x, z, \omega) + \epsilon} \frac{\Delta\omega}{2\pi}, \quad (2.12)$$

where  $\epsilon$  is a stabilized factor. It is added to avoid the case of the denominator being zero. In this way, the final reflectivity can be estimated by summing over the temporal frequencies within a discrete set of frequency values (Poole et al., 2010).

In the next subsection, some synthetic examples show that scattering terms used in the PSPI method can enlarge the subsurface illumination and improve the imaging result resolution.

### 2.1.2 Numerical examples

In this subsection, the assumption is made that the wave propagation occurs in an acoustic medium. This acoustic nature of the medium ensures that the upward and downward scattering terms within the wavefield are equivalent. Furthermore, a smooth velocity model is considered for the migration process, assuming gradual variations in velocity without significant heterogeneities or abrupt changes. However, it should be noted that the accuracy and range of the input velocity model can significantly impact the generation of secondary wavefields associated with the scattering terms. So, an assumption is that the smoothed velocity model is known in the synthetic example. The choice of smoother is limited due

to the dependency of input velocity model. After conducting various trials and tests, a Gaussian smoother with a half-width of 31 cells is determined to be used. In the following section, I will show three synthetic examples to demonstrate the significance of including the scattering terms and their impact on the accuracy and resolution of the PSPI.

### **Horizontal-layered model**

The first example is a horizontal-layered model, whose size is 351x511 points with 5 meters spatial interval, and there are three layers in total with velocities of 2000, 2800 and 3500 m/s, respectively. The time sampling rate is 1 ms, and the recorded time is 2.047 s. Five shots are simulated at around 15 meters depth to mimic the situation of a sparse acquisition. The number of receivers is the same as that of horizontal points.

A sharp velocity model (Figure 2.2a) is used for generating shot records (Figure 2.2c). Then, a smoothed background velocity (Figure 2.2b) is input to PSPI migration to generate the migrated images. Compared with the PSPI migration result without scattering terms in Figure 2.3a, adding scattering terms (Figure 2.3b) can help indicate the locations of the reflectors. The oscillation terms generated manually rebuild the reflector edges with widened information and increased amplitudes. Because of a strong dependency on the smoothed velocity input shown in equation 2.10 to calculate wavenumber frequencies, the migration result after adding the scattering term predicts a relatively smooth reflectivity coefficient.

### **Boxcar model**

In this section, a horizontal-layered model combined with a boxcar shape anomaly in the middle depth is used to test the proposed method in the presence of lateral velocity changes. The model has 301 x 511 gridpoints with a 5 m spatial interval. Figures 2.4a and b give the true and smoothed input velocity models. Five shots with 50 m intervals are located on the surface starting from 500 m. A fourth-order finite difference method is used for forward modelling shown in Figure 2.4c.

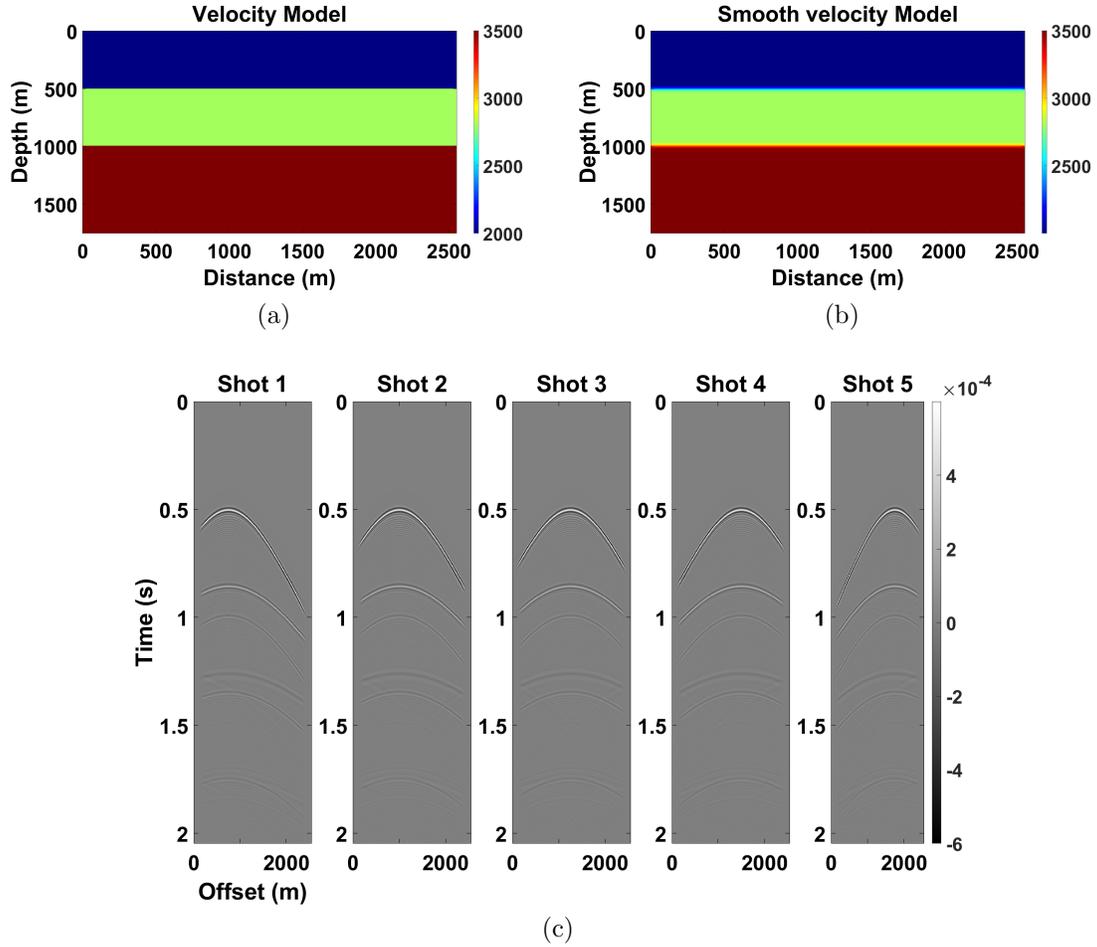


Figure 2.2: The horizontal-layered example. (a) True velocity model. (b) Smoothed velocity model. (c) Shot records for the horizontal-layered model at 750, 1000, 1250, 1500 and 1750 meters offset below the surface at 15 meters depth.

Figure 2.5a gives the PSPI migration result without scattering term. The first reflector at 500 meters depth has been recovered with high amplitudes, but the boxcar anomaly and the deep horizontal event cannot be predicted accurately. On the other hand, Figure 2.5b, which uses scattering terms, can give an accurate migrated location for the subsurface structure. For example, the boxcar anomaly prediction between depth 1000 and 1500 meters has larger amplitudes on the upper side boundary than Figure 2.5a. Two horizontal layers can be reconstructed with higher resolution and extended boundary information by adding the scattering terms than not using them.

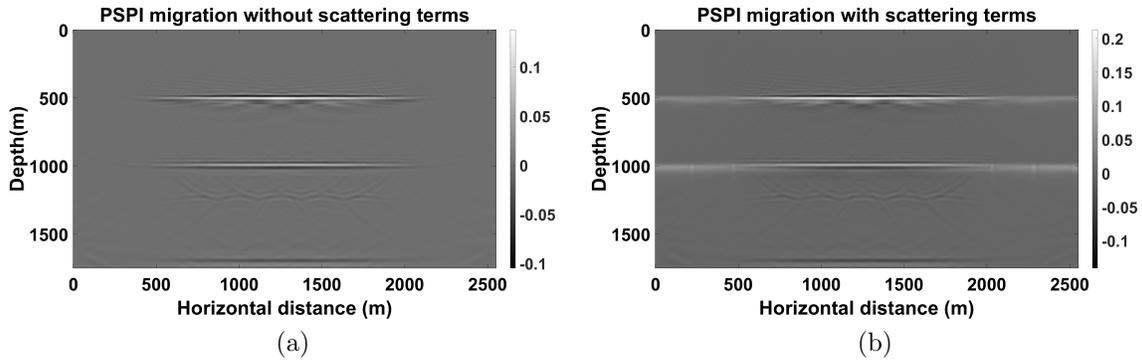


Figure 2.3: The horizontal-layered example. (a) PSPI migration without scattering term; (b) PSPI migration with scattering terms.

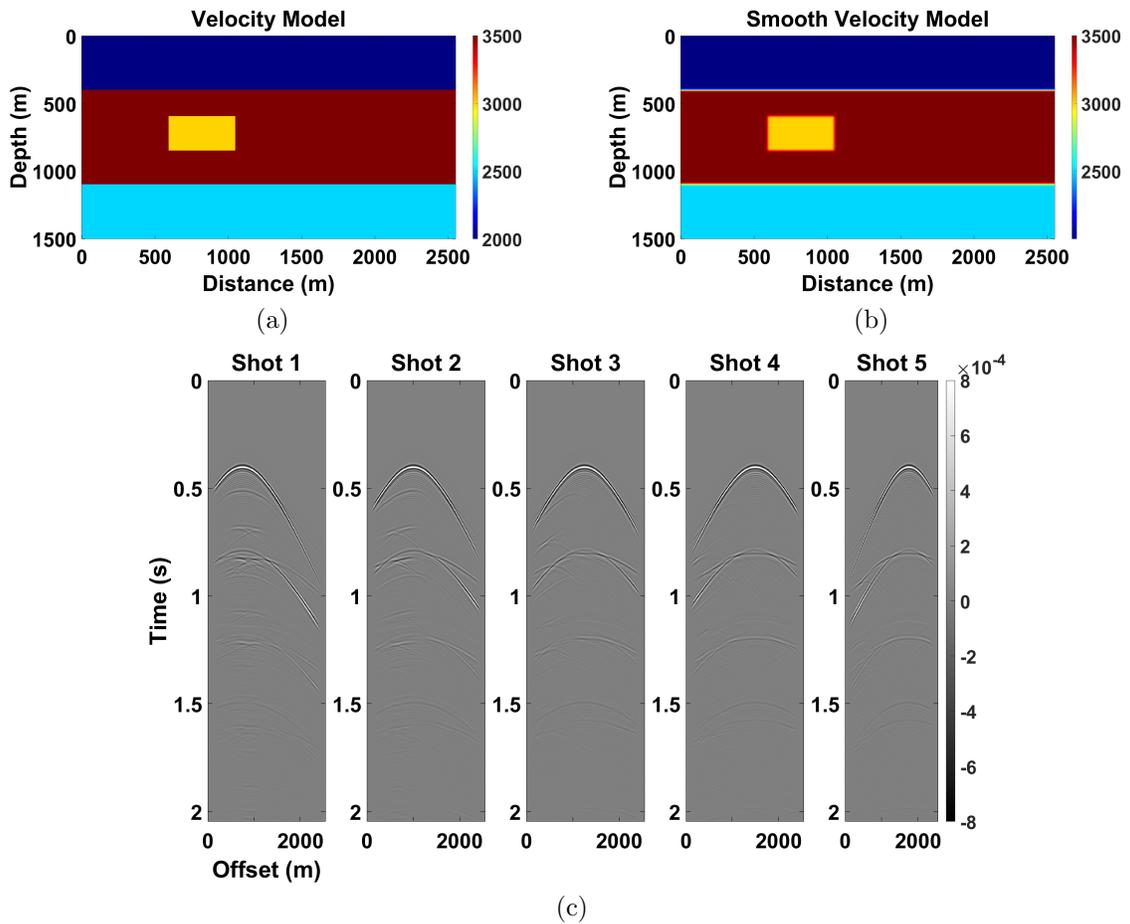


Figure 2.4: The boxcar model example. (a) True velocity model. (b) Smoothed velocity model. (c) Shot records simulated at 1800, 2800, and 3800 m on the surface.

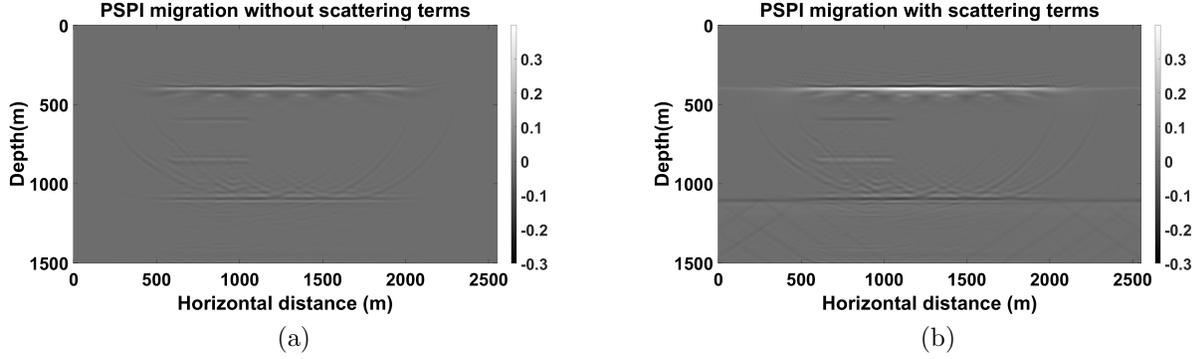


Figure 2.5: The boxcar model example. (a) PSPI migration without scattering term; (b) PSPI migration with scattering terms.

### Boxcar shape with a dome model

The next numerical example turns to a more complicated case that adds a curvature shape anomaly with  $276 \times 476$  points. The setting of the rest of the model parameters, including horizontal distance, depth, time and space intervals, and the number of shots, are the same as those in the previous example. Figures 2.6a and b show the true and smoothed velocity models, and Figure 2.6c demonstrates the shot records obtained from a horizontal distance at 550, 950, 1350, 1750 and 2150 m separately.

For the boxcar shape structure, the results show a similar result as the previous example. The scattering terms (Figure 2.7b) help PSPI migration to locate where the boxcar anomaly is since it provides the top and bottom boundary information, precisely increasing amplitudes of the upper side reflector. As for the right curvature anomaly, Figure 2.7a, without scatters, cannot recover the details of the structure shape and only gives rough information for the top and bottom boundaries at 500 m and 875 m depth. However, PSPI migration with scattering terms shown in Figure 2.7 can migrate the anomaly's top boundary to a correct location. Similarly, the reflectivity of the layer near the dome structure is predicted with increased amplitudes. For the deep horizontal event, PSPI migration with the scattering term can predict a higher amplitude of the reflector than Figure 2.7a, but with more artifacts below the event.

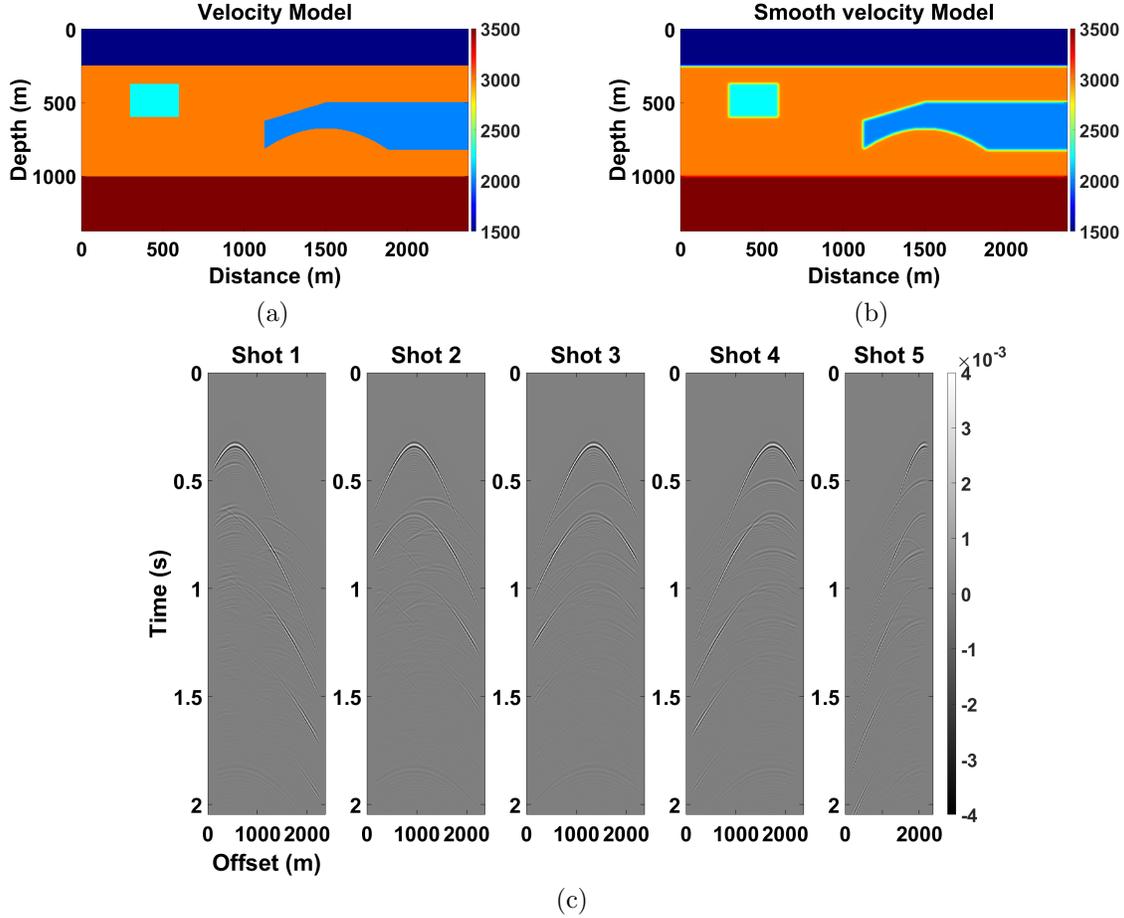


Figure 2.6: The boxcar model with a dome. (a) True boxcar-dome velocity model. (b) Smoothed velocity model. (c) Shot records.

### 2.1.3 Considerations

#### Dependency on the migration velocity

The proposed method improves PSPI migration on structures with lateral velocity variations by using scattering terms to extend the subsurface information. Still, this method depends on the accuracy of velocity information. If the velocity has a large bias, for example, the scattering terms calculated based on the reflectivity coefficient variation will lead to the wrong location.

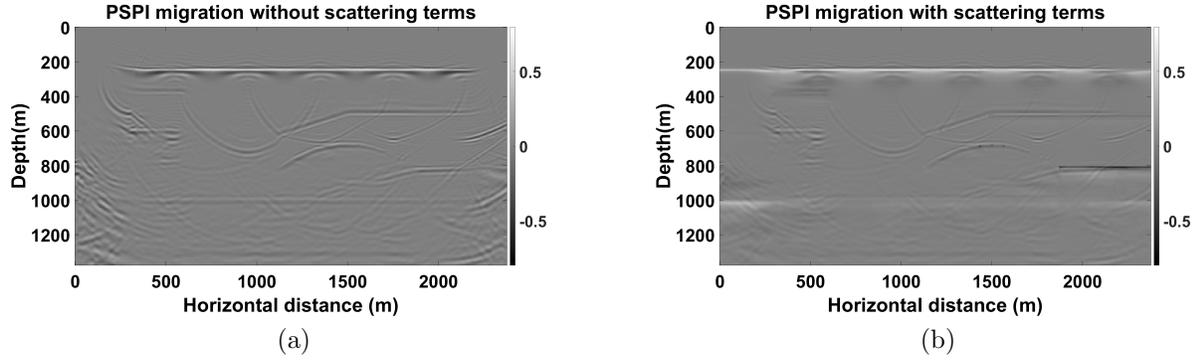


Figure 2.7: The boxcar model with a dome. (a) PSPI migration without scattering term; (b) PSPI migration with scattering terms.

### 2.1.4 Conclusions

When given a sparse and coarse shot acquisition, PSPI migration with scattering terms can extend the aperture information and provide reflectors with higher resolution and amplitudes than without scattering terms. However, even though PSPI migration with scattering term migrates the reflector location accurately at the deeper structure, the incoherent noise is below the structure due to wavefield inferences. To overcome this challenge, future research can focus on developing innovative approaches for noise removal.

# Chapter 3

## Convolutional neural network-based reverse time migration with multiple reflections

In the last chapter, I discussed the use of PSPI migration with scattering terms to generate and migrate multiples. The limitations include the difficulty of generating and migrating those reflections adequately without artifacts. High computational cost also limits the quality of migration results. So, in this chapter, based on the traditional migration approach, I propose a method that helps to resolve reflection events with improved accuracy and less computation time.

### 3.1 Summary

Reverse-time migration (RTM) has the advantage that it can handle steep dipping structures and offer high-resolution images of the complex subsurface. Nevertheless, there are some limitations to the chosen initial model, aperture illumination and computation efficiency. RTM has a strong dependency on the initial velocity model. The RTM result image will

perform poorly if the input background velocity model is inaccurate. One solution is to apply least-squares reverse-time migration (LSRTM), which updates the reflectivity and suppresses artifacts through iterations. However, the output resolution still depends heavily on the input and accuracy of the velocity model, even more than for standard RTM. For the aperture limitation, RTM with multiple reflections (RTMM) is instrumental in improving the illumination but will generate crosstalks because of the interference between different orders of multiples. We proposed a method based on a convolutional neural network (CNN) (O’Shea and Nash, 2015) that behaves like a filter applying the inverse of the Hessian. This approach can learn patterns representing the relation between the reflectivity obtained through RTMM and the true reflectivity obtained from velocity models through a residual U-Net with an identity mapping. Once trained, this neural network can be used to enhance the quality of RTMM images. Numerical experiments show that RTMM-CNN can recover structures and thin layers with higher resolution and improved accuracy, compared with the RTM-CNN method. Additionally, the proposed method demonstrates the generalization ability across diverse geology models, including complex thin layers, salt bodies, folds, and faults. Moreover, The computational cost of the method is lower compared with LSRTM.

## 3.2 Introduction

Reverse-time migration (RTM) (Baysal et al., 1983; Whitmore, 1983; McMechan, 1983; Levin, 1984) can handle steep geologic structure flanks and lateral velocity variations. However, it suffers from coherent or incoherent artifacts in diving and back-scattered waves (Larner et al., 1983; Yoon et al., 2004), as well as low resolution and illumination for deep structure when given insufficient source-receiver offsets. A solution to suppress the artifact issue is applying least-squares reverse-time migration (LSRTM) (Dong et al., 2012), which uses RTM as the forward modeling and inverse engine to minimize amplitude differences between observed data and predicted data and updates the reflectivity iteratively. Extensive research on least-squares imaging such as compressive sensing (Herrmann and Li, 2012),

uncertainty quantification (Herrmann et al., 2019), sparsity constraints (Wu et al., 2021), curvelet-domain sparse constraint (Wu et al., 2022a), and multiplicative Cauchy constraint (Yao et al., 2022) help to improve imaging.

Although LSRTM improves illumination with respect to RTM, it still has a limited aperture problem, since it uses only primary reflections. Multiple migration used in imaging (Tu and Herrmann, 2015) and the RTM (RTMM) (Liu et al., 2011; Li et al., 2017; Wang et al., 2017; Zhang et al., 2020) can help to broaden the subsurface illumination and refine the accuracy and resolution. Another way to enhance imaging quality is through deep learning (Goodfellow et al., 2016). Many researchers have worked on this approach and addressed accuracy improvement and artifact suppression in seismic processing. For example, ground roll attenuation (Jia et al., 2019; Kaur et al., 2020a), seismic inversion applications (Richardson, 2018; Sun et al., 2020, 2021; Vantassel et al., 2021), transfer learning applications in modeling and imaging (Siahkoohi et al., 2019a), and generative neural networks in inverse problems (Siahkoohi, 2022; Liu et al., 2020) propose to use the multilayered convolutional neural network (CNN) (LeCun et al., 1998; Krizhevsky et al., 2012; LeCun et al., 2015) as the solution to the problem of sparse least-squares migration (LSM) to suppress coherent and incoherent noise in migration results. For deep learning in RTM and least-squares RTM, Wu et al. (2018) propose LSRTM with the adaptive moment estimation in the frequency domain; Kaur et al. (2020b) apply a generative adversarial network on RTM images with a velocity attribute conditioner to estimate the inverse of the Hessian and match with least-squares migrated images; Lu et al. (2020) also uses CNNs on dip-angle domain elastic reverse-time migration to improve image quality; Vamaraju et al. (2021) introduces the idea on minibatch LSRTM, and Torres and Sacchi (2021, 2022) use blocks of residual CNN on LSRTM with a preconditioned conjugate gradient least-squares algorithm (CGLS) to enhance image resolution; Zhang et al. (2022) uses deep learning for accelerating prestack correlative LSRTM. These methods mitigate artifacts and foster resolution by training a machine-learning network.

Exploiting the two facts that multiple reflections can enhance the imaging bandwidth and convolutional neural network (CNN) can learn the lithologic structure from different feature maps. We propose a CNN-based RTM with the multiple reflections energy method (RTMM-CNN). In this approach, we use a U-Net (Ronneberger et al., 2015) acting as a filter to learn the reflection boundaries from the RTMM results, and we also make the filter learn the mapping of multiple energy. We use the U-Net-based RTM image as the baseline model without adding multiple energy (RTM-CNN). Models have two components: preconditioning and fine-tuning. The preconditioning constrains the parameter range in the fine-tuned models, improving the image quality. Results show that the proposed method can obtain reflectivity prediction with extended illumination, refined structural boundaries, high accuracy, and enhanced resolution.

### 3.3 Method

Before the theory part, some basic geophysical variables need to be explained. Seismic waves, generated from the simulation of sources on the surface, are extrapolated downwards into the subsurface. When they encounter some discontinuities of physical properties, reflection and transmission waves are generated on the interfaces.

These discontinuities are called seismic impedance discontinuities. They are the product of subsurface velocity and density and are components of the reflectivity equation to generate wavefield perturbations.

$$Z = \rho v, \tag{3.1}$$

where  $\rho$  is the density of rock and  $v$  denotes the velocity of that rock.

The goal of seismic imaging is to estimate, for each point of the subsurface, a parameter called reflection coefficient, or reflectivity. This parameter indicates the ratio of waves reflected and transmitted through a discontinuity media, compared with incident waves. For an acoustic normal incident wave situation (Kearey and Brooks, 1991), the expression of the

reflection coefficient is

$$R = \frac{Z_2 - Z_1}{Z_2 + Z_1}, \quad (3.2)$$

where  $Z_1$  and  $Z_2$  mean the impedance of the first and second medium, respectively. Thus, the target of seismic migration and imaging is to obtain the approximation of reflectivity and interpret subsurface structures by using collected wavefields.

### 3.3.1 Four Scenarios in Reverse-Time Migration

Reverse-time migration involves two wavefields. The source wavefield produces the wave propagation from shots, and the receiver wavefield reproduces the time-reverse wave propagation from receivers (which is acquired data). Since the cross-correlation of these two wavefields generates the desired reflectivity, it is essential to make them physically consistent. The shot wavefield will be limited mainly by our knowledge of velocities. Since velocities are usually not known at a level of detail to create internal reflections, we usually deal with what is called the Born wavefield (no internal reflections). This wavefield may or may not contain surface multiples depending on how we perform modeling (for example, using or not using an absorbing boundary condition (Sochacki et al., 1987; Higdon, 1991) on the surface). For the receiver wavefield, however, there is a limitation not only on the velocities (the same as for the source wavefield) but also on what the data contain (were internal or external multiples attenuated before migration?). There are several possibilities for the relationship between these wavefields. For this paper, we can separate the following four scenarios, although only scenarios 1, 3, and 4 are critical for the following discussions:

- Scenario 1: Smooth or background velocity input to RTM and absorbing boundary conditions on the surface. Multiples were attenuated from data before migration. This scenario is the typical case in real data applications. The forward wavefield will be Born modeling (no internal multiples)
- Scenario 2: True velocity input to RTM and data without multiples. This scenario

is not practical, because true velocities are not known at that level of detail. There is an inconsistency between forward wavefields (which will have all multiples) and reverse wavefields (which will have some attenuated multiples and some generated during backward propagation). We leave this scenario out of the discussion.

- Scenario 3: Smooth velocity input to RTM and data with multiples. This scenario is our goal, because it is close to reality (only the background model is known), and data will contain multiples (unless attenuated explicitly). These multiples will provide additional illumination for RTM. Nonetheless, there is an inconsistency between wavefields that has to be addressed.
- Scenario 4: True velocity input to RTM and data with multiples. As for this scenario, it is also hard to perform in reality, but it has an ideal result that combines high-frequency bandwidth and multiple reflections, which makes migrated results with high resolution and insight for subsurface structure interpretation. We try to build a neural network on the basis of scenario 3 and make the result close to scenario 4.

From Figure 3.1a and d, we can see that RTM in scenario 1 can adequately recover the structure of a relatively simple thrust model. As expected, the illumination is stronger on the shallow part of the thrust structure than on the deeper one. Significant shallow-depth illumination is the typical RTM result expected in practice. Although there are some difficulties to achieve in practical conditions, the RTM of scenario 4 (Figure 3.1c and f) is an ideal result. The thrust structure is estimated with high resolution and accuracy, and the illumination is very good for shallow and deep reflectors. The RTM of scenario 3 (Figure 3.1b and e) seems somewhat better than in scenario 1 but worse than in scenario 4.

The illumination is not inadequate for the deeper geological structure. However, the more complicated the model, the more deterioration of the image appears, as the crosstalk will introduce more artifacts. If the thrust model had many reflectors in between, the crosstalk would be significantly visible, and the image would be poor. The situation above is easy

to obtain in practical applications if the multiples are not removed from the data before migration.

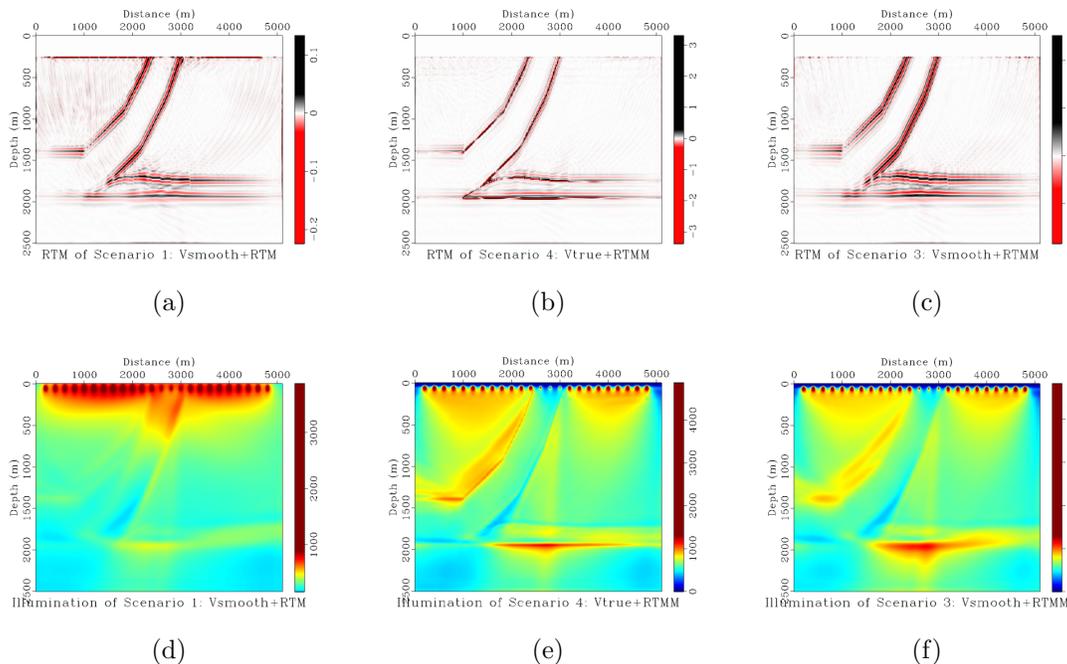


Figure 3.1: (a) RTM of Scenario 1: Smoothed velocity as the input in the RTM process. (b) RTM of Scenario 4: True velocity as the input in the RTM with multiple energy. (c) RTM of Scenario 3: Smooth velocity as the input in the RTM with multiple reflections. (d) Shot illumination of Scenario 1. (e) Shot illumination of Scenario 4. (f) Shot illumination of Scenario 3.

### 3.3.2 Wavefield Inconsistency

As mentioned above, an RTM with a smooth velocity and data with multiples (scenario 3) will have inconsistency between wavefields. The forward wavefield will consist of traveltimes from primaries, surface multiples, and internal multiples during the imaging process. However, the reverse wavefield will contain primaries and attenuated multiples due to the smooth background velocity model, which does not generate internal reflections. Since the velocity model lacks the information to unravel the traveltimes for multiples properly, the receiver wavefield will incorrectly cross-correlate with the source wavefield, and crosstalk noise will

occur (Liu et al., 2011; Dong et al., 2012).

To mitigate the traveltime mismatch, previous work, for example, (Schuster (2002, 2003)) and Jiang et al. (2007), proposed using modified Green’s functions to migrate multiples. The method above constructs the Kirchhoff imaging condition for multiples by combining traveltime picked from a shifted source wavelet and obtained data, which is based on Fermat’s principle. It assumes that the traveltimes of a lower-order event can be picked (or windowed in the prestack data) in order to image a higher-order event. Another essential factor is that the multiples’ energy must be sufficient for picking.

To alleviate the wavefield inconsistency and traveltime mismatch in the reverse-time migration result, our proposed method lets the neural network learn the traveltime mismatch and correct the inconsistency iteratively during training. Certainly, this is not a trivial assumption. Like other deep learning applications, it is impossible to provide a justification or proof that this would be the case, and experimentation is critical. The benefit is that we can take advantage of all the receiver’s information, including primaries, multiples, or crosstalks (if they help improve reflection coefficient information). Thus, the extended illumination brought from multiple reflections can help us improve the subsurface image result. A smoothed reflectivity model from background velocity and corresponding RTM outcome with multiple energy are considered the input channels of this neural network. In other words, we want to try to obtain high-resolution results of scenario 4 from scenario 3.

### 3.3.3 RTM with Multiples

The workflow of reverse-time migration is given by a forward- and reversed-time propagation of source and receiver wavefields, respectively, followed by an imaging condition. As multiple reflections are considered in the RTM process, free surface boundaries need to add to the top of velocity models.

Similar to Liu et al. (2011), using primaries  $P(x, z, t)$ , we apply the total observations ( $P(x, z, t) + M(x, z, t)$ ) as the virtual source to generate multiples  $M'(x, z, t)$ , where

$M(x, z, t)$  represents internal multiples generated from the first forward modeling. Then, multiples  $M'(x, z, t)$  have the total wavefields, including primaries and surface and internal multiples. During the imaging process, those virtual-source wavefields will be forward-extrapolated into the subsurface as  $P_F(x, z, t) + M_F(x, z, t)$ . The newly acquired data  $M'(x, z, t)$  will be back-propagated into the subsurface and considered as our reversed-time receiver wavefields  $M'_B(x, z, t)$ .

A zero-lag cross-correlation imaging condition based on the virtual-source and receiver wavefields Liu et al. (2011) can be applied:

$$I(x, z) = \sum_{t=1}^{t_{max}} (P_F(x, z, t) + M_F(x, z, t)) * M'_B(x, z, t). \quad (3.3)$$

Even though the imaging condition generates crosstalks, the trained neural network can learn patterns and features from the relationship between the migrated result and the true reflectivity model. In that case, the neural network can exploit the benefits of multiple energy and mitigate artifacts in the image.

Figure 3.2 shows the Pluto zero-offset migration images with respect to a background reflectivity by applying RTM (scenario 1, Figure 3.2c), RTMM (scenario 3, Figure 3.2d), and RTMM with true band-limited reflectivity (scenario 4, Figure 3.2e), respectively. The migrated image using multiple energy (Figure 3.2d) can help extend horizontal-layer illumination, recover deep thin-layer structures and provide lateral continuity structural information. It is beneficial for a neural network to identify reflection events using multiple reflections in the migration. In the next section, we see that reflectivity predictions can be improved by using migration velocity models as constraints introduced into the network in a secondary channel.

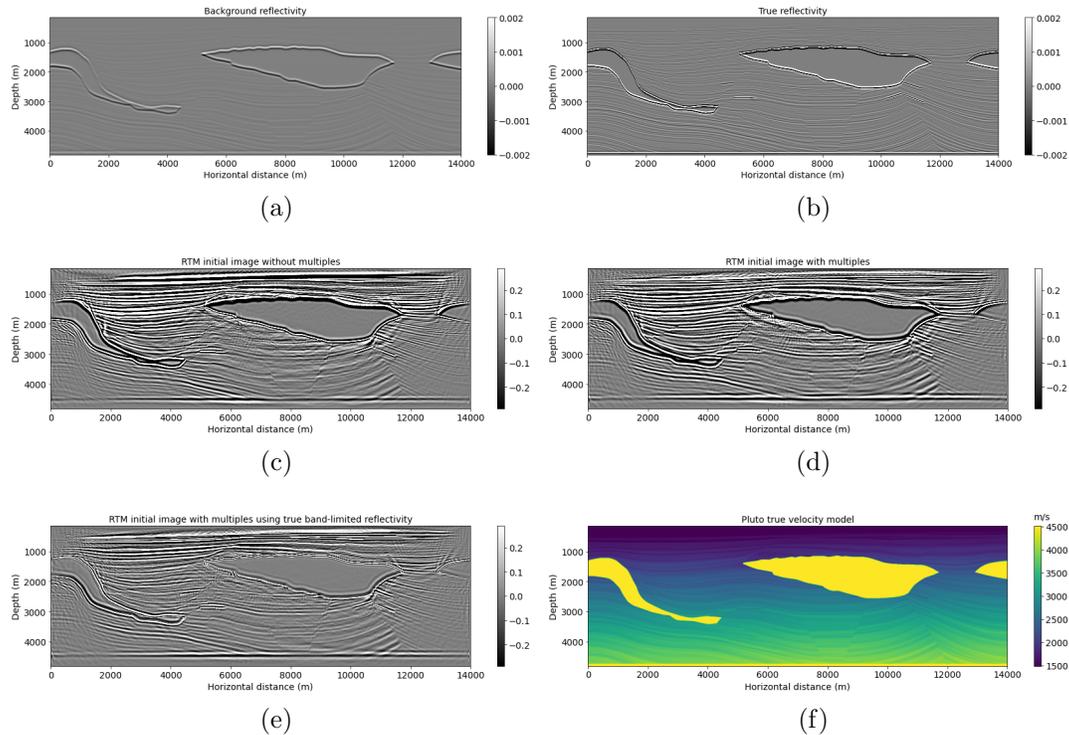


Figure 3.2: Pluto example: (a) background reflectivity, (b) true reflectivity, (c) RTM image without multiple energy, (d) RTM image with multiple energy, (e) RTMM with true band-limited reflectivity, and (f) true velocity model.

## A U-Net-based RTM with multiples

Given a particular type of input in deep learning, we train a multilayer network to predict the desired outcome. A series of weights are calculated to map the inputs to the desired output during training. Although, essentially, this is just a geometrical mapping, the transformation contains both linear (the weights) and non-linear elements (the activation functions), which, added to a large number of weights, have the potential to take into account many complex effects. By choosing the types of inputs and the desired outputs (known as labels), we can make the network learn any particular mapping we need. There are many limitations on what this type of geometrical transformation can learn, but more importantly, there are limitations on the generality of this mapping. In general, it is difficult to say whether a particular mapping will succeed, and we often have to rely on numerical experiments to

achieve a conclusion. In this paper, we propose to use a U-Net (Figure 3.3) to map the outcomes of a reverse-time migration obtained from data with multiples and a migration velocity model (smooth) to a well-resolved image (RTMM-CNN). The reflectivity obtained from the migration velocity is incorporated as a physical constraint to provide low-frequency information to the network.

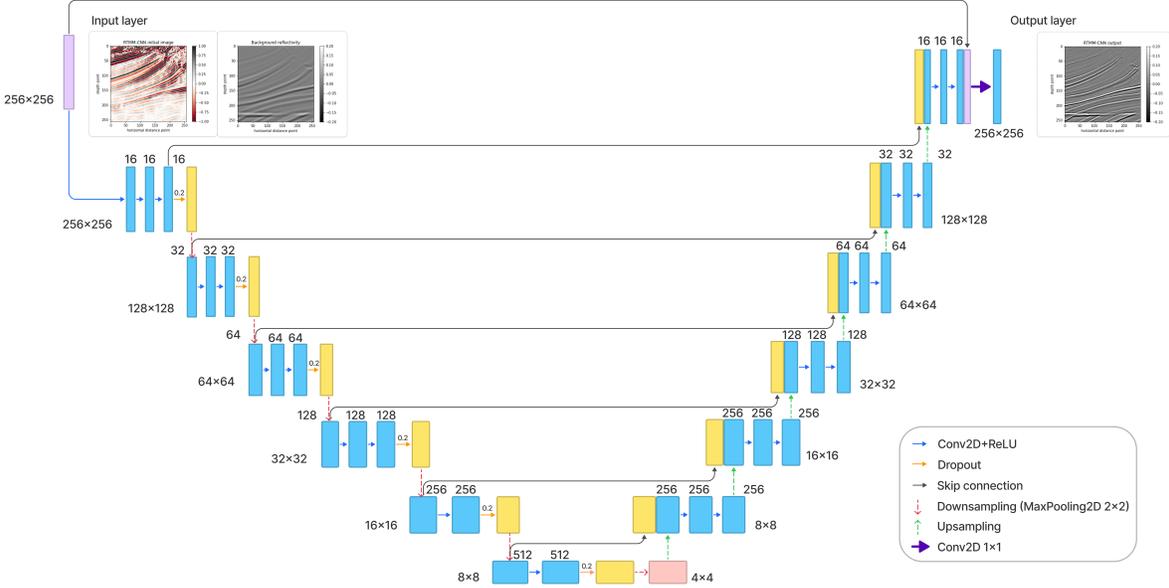


Figure 3.3: Detailed workflow of the modified U-net architecture. Each blue box represents a multi-channel feature. The yellow boxes stand for the concatenated copied feature from the encoder part. The arrows between boxes correspond to the different operations as shown in the right legend. The number of channels is located on top of the box and the image dimensionality is denoted on the left edge.

U-Net (Ronneberger et al., 2015) is an encoder-decoder approach commonly used for image segmentation (pixel classification), but we utilize it here for a regression problem. In this paper, we develop a U-Net (Figure 3.3) with additional multilayer convolutional blocks and skip connections to learn from residuals and patterns in the data. Convolutional blocks are used for capturing detailed input features. For example, they help distinguish signals and noise from images with multiple reflection information. The network downsamples the input data into small sizes for the encoder part. It reduces its dimensionality to learn key features of

different reflectors from RTMM images, smoothed initial reflectivity, and accurate reflectivity labels. Then, these subsurface key features are upsampled to the original dimensions by transposed convolutions. Additional skip connections work as identity mapping because the signal could be directly propagated from one unit to any other unit (He et al., 2016). These identity shortcut connections help to smooth key feature propagation and strengthen the training result with weak constraints. For the output layer, a linear activation function is used for obtaining positive and negative prediction values, which obey the nature of reflectivity amplitude.

The U-Net provides a mechanism to design a prediction filter from our training data (the RTM images and additional support channels) to the labels (the simulated reflectivity obtained from true velocities). This U-Net operator acts similarly to an image domain LSRTM, but the inverse of the Hessian is calculated not from inverse filtering but by training. In comparison, LSRTM in the image domain yields a high-definition image by removing the effect of the Hessian from the migrated image. The calculation of the inverse of the Hessian filter can be calculated with different methods (Nemeth et al., 1999; Hu et al., 2001; Guitton, 2004; Yu et al., 2006; Aoki and Schuster, 2009; Guo and Wang, 2020), but for simplicity, we can summarize as follows:

$$\mathbf{m}^* = \arg \min_m \left\{ \frac{1}{2} \|\mathbf{\Gamma} \mathbf{m} - \mathbf{m}_{mig}\|_2^2 \right\}. \quad (3.4)$$

A formal solution to Equation (3.4) is

$$\mathbf{m}^* = \mathbf{\Gamma}^{-1} \mathbf{m}_{mig} = \mathbf{\Gamma}^{-1} (\mathbf{L}^T \mathbf{d}), \quad (3.5)$$

$$\mathbf{\Gamma} = \mathbf{L}^T \mathbf{L}, \quad (3.6)$$

where  $\mathbf{\Gamma}^{-1}$  is the inverse Hessian,  $\mathbf{L}^T$  is the adjoint operator, and  $\mathbf{d}$  represents the observed seismic data. From Equation (3.5), we can find that prediction  $\mathbf{m}^*$  is generated from the

deblurring process of the first migrated image  $\mathbf{m}_{mig}$  by the inverse of the Hessian matrix  $\mathbf{\Gamma}$ .

Similarly, a neural network, U-Net in our case, can be used as an approximate inverse Hessian (Kaur et al., 2020b; Torres and Sacchi, 2021) to determine the imaging result. The benefit is that there is no need to compute the expensive inverse Hessian operator. The Hessian contains the effects of limited acquisition aperture, uneven illumination, and band-limited source wavelets. These effects compromise the goal of obtaining a true-amplitude and high-resolution reflectivity (Guo and Wang, 2020). The feed-forward procedure in our proposed method for a multilayer CNN is  $\mathbf{\Gamma}_{UNET}$ , and the solution can be determined as follows, depending on the different scenarios:

$$\text{Workflow 1 : } \mathbf{m}_{pred_1} = \mathbf{\Gamma}_{UNET\_fine\_tuned\_workflow2}(\mathbf{m}_{rtm\_scenario1}, \mathbf{m}_{smooth}), \quad (3.7)$$

$$\text{Workflow 2 : } \mathbf{m}_{pred_2} = \mathbf{\Gamma}_{UNET\_workflow2}(\mathbf{m}_{rtm\_scenario2}, \mathbf{m}_{true}), \quad (3.8)$$

$$\text{Workflow 3 : } \mathbf{m}_{pred_3} = \mathbf{\Gamma}_{UNET\_fine\_tuned\_workflow4}(\mathbf{m}_{rtmm\_scenario3}, \mathbf{m}_{smooth}), \quad (3.9)$$

$$\text{Workflow 4 : } \mathbf{m}_{pred_4} = \mathbf{\Gamma}_{UNET\_workflow4}(\mathbf{m}_{rtmm\_scenario4}, \mathbf{m}_{true}), \quad (3.10)$$

where  $\mathbf{m}_{rtm}$  means RTM image,  $\mathbf{m}_{rtmm}$  is the RTMM image, and  $\mathbf{m}_{pred}$  represents the output reflectivity coefficient prediction. The subscripts after  $\mathbf{m}_{rtm}$  and  $\mathbf{m}_{rtmm}$  correspond to the different scenarios mentioned before. For instance,  $\mathbf{m}_{rtm\_scenario1}$  represents the RTM image from scenario 1. The neural networks used in the workflows are set individually depending on the preconditioning or final imaging demand. For example, the neural network model generated from workflow 2 ( $\mathbf{\Gamma}_{UNET\_workflow2}$ ) is used for preconditioning, because its inputs are the true reflectivity and the RTM image from scenario 2. Then, this saved model is treated as a pre-trained model for workflow 1. After fine-tuning, workflow 1 will obtain a final training model  $\mathbf{\Gamma}_{UNET\_fine\_tuned\_workflow2}$ . A detailed explanation for decisions about preconditioning and final imaging models is delineated in a later section. As for  $\mathbf{m}_{true}$  and  $\mathbf{m}_{smooth}$ , the former denotes the true band-limited reflectivity, which is used as our labels during training;

the latter is an initial reflectivity calculated from the background velocity used for migration. This reflectivity contains only low-frequency information, similar to what migration normally uses, but uses it as an additional input channel, providing a supplementary constraint for the network. The workflows correspond to scenarios 1 to 4, mentioned previously.

For a detailed U-Net architecture (Tables 3.1 and 3.2), there are 45 layers for encoding and 44 layers for decoding, respectively. In the contracting path, each convolutional block has three convolutional layers for the first four blocks. The last two blocks contain two convolutional layers. After each block, the maxpooling layer, with a size of 2 by 2 cells, halves the image’s size and increases the neural network’s depth. Table 3.1 indicates that, at the end of the encoding part, the image size is reduced from 256x256x2 to 4x4x512 by using the convolutional blocks and maxpooling layers. The number of channels increases from 2 to 512. On the other side, in the expansive path, extracted features are upscald by transposed convolutional layers and back to the image’s original size. Following transposed convolution, the resized image is concatenated with an image from the contracting path sharing the same size. Skip connection combines previous image information and makes a stable and accurate prediction. Before outputting the prediction, another skip connection layer is added to obtain a precise result.

### Mean Squared Error (MSE)

As estimating the reflectivity coefficient from a seismic migration profile with an initial reflectivity model is a regression problem, a mean squared error (MSE) loss is used to evaluate the model performance and calculate the gradient:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\mathbf{m}_{pred}^i - \mathbf{m}_{true}^i)^2, \quad (3.11)$$

where  $n$  is the total number of samples,  $\mathbf{m}_{pred}$  is derived from the workflows above (Equations (3.7)–(3.10)), and  $\mathbf{m}_{true}$  denotes the true reflectivity model.

Layer number	Type	Size	Output
1	Input		256x256x2
2	Conv2D	16 filters	256x256x16
3	Batch Normalization		256x256x16
4	Conv2D	16 filters	256x256x16
5	Batch Normalization		256x256x16
6	Conv2D	16 filters	256x256x16
7	Batch Normalization		256x256x16
8	Dropout	20%	256x256x16
9	MaxPooling2D	2x2	128x128x16
10	Conv2D	32 filters	128x128x32
11	Batch Normalization		128x128x32
12	Conv2D	32 filters	128x128x32
13	Batch Normalization		128x128x32
14	Conv2D	32 filters	128x128x32
15	Batch Normalization		128x128x32
16	Dropout	20%	128x128x32
17	MaxPooling2D	2x2	64x64x32
18	Conv2D	64 filters	64x64x64
19	Batch Normalization		64x64x64
20	Conv2D	64 filters	64x64x64
21	Batch Normalization		64x64x64
22	Conv2D	64 filters	64x64x64
23	Batch Normalization		64x64x64
24	Dropout	20%	64x64x64
25	MaxPooling2D	2x2	32x32x64
26	Conv2D	128 filters	32x32x128
27	Batch Normalization		32x32x128
28	Conv2D	128 filters	32x32x128
29	Batch Normalization		32x32x128
30	Conv2D	128 filters	32x32x128
31	Batch Normalization		32x32x128
32	Dropout	20%	32x32x128
33	MaxPooling2D	2x2	16x16x128
34	Conv2D	256 filters	16x16x256
35	Batch Normalization		16x16x256
36	Conv2D	256 filters	16x16x256
37	Batch Normalization		16x16x256
38	Dropout	20%	16x16x256
39	MaxPooling2D	2x2	8x8x256
40	Conv2D	512 filters	8x8x512
41	Batch Normalization		8x8x512
42	Conv2D	512 filters	8x8x512
43	Batch Normalization		8x8x512
44	Dropout	20%	8x8x512
45	MaxPooling2D	2x2	4x4x512

Table 3.1: UNet Architecture - Encoding.

Layer number	Type	Size	Output
1	Conv2D Transpose	256 filters	8x8x256
2	Batch Normalization		8x8x256
3	Concatenate		8x8x768
4	Conv2D Transpose	256 filters	8x8x256
5	Batch Normalization		8x8x256
6	Conv2D Transpose	256 filters	8x8x256
7	Batch Normalization		8x8x256
8	Conv2D Transpose	256 filters	16x16x256
9	Batch Normalization		16x16x256
10	Concatenate		16x16x512
11	Conv2D Transpose	256 filters	16x16x256
12	Batch Normalization		16x16x256
13	Conv2D Transpose	256 filters	16x16x256
14	Batch Normalization		16x16x256
15	Conv2D Transpose	128 filters	32x32x128
16	Batch Normalization		32x32x128
17	Concatenate		32x32x256
18	Conv2D Transpose	128 filters	32x32x128
19	Batch Normalization		32x32x128
20	Conv2D Transpose	128 filters	32x32x128
21	Batch Normalization		32x32x128
22	Conv2D Transpose	64 filters	64x64x64
23	Batch Normalization		64x64x64
24	Concatenate		64x64x128
25	Conv2D Transpose	64 filters	64x64x64
26	Batch Normalization		64x64x64
27	Conv2D Transpose	64 filters	64x64x64
28	Batch Normalization		64x64x64
29	Conv2D Transpose	32 filters	128x128x32
30	Batch Normalization		128x128x32
31	Concatenate		128x128x64
32	Conv2D Transpose	32 filters	128x128x32
33	Batch Normalization		128x128x32
34	Conv2D Transpose	32 filters	128x128x32
35	Batch Normalization		128x128x32
36	Conv2D Transpose	16 filters	256x256x16
37	Batch Normalization		256x256x16
38	Concatenate		256x256x32
39	Conv2D Transpose	16 filters	256x256x16
40	Batch Normalization		256x256x16
41	Conv2D Transpose	16 filters	256x256x16
42	Batch Normalization		256x256x16
43	Concatenate		256x256x16
44	Conv2D	1 filter	256x256x1

Table 3.2: UNet Architecture - Decoding.

## Peak Signal-to-noise Ratio (PSNR)

A peak signal-to-noise ratio (PSNR) is used to evaluate the model performance:

$$\text{PSNR} = 20 \log_{10} \left( \frac{MAX_I}{\sqrt{MSE}} \right), \quad (3.12)$$

where  $MAX_I$  denotes the maximum possible pixel value of the image, and  $MSE$  is the mean squared error based on the Equation (3.11).

### 3.3.4 Neural Network Plan for Four Workflows

In this section, we introduce in detail four workflows that are defined in the previous section. To train the neural network and make it learn patterns from both accurate and smoothed inputs, workflows 1 and 3 in Figure 3.4 are fine-tuned based on the neural networks obtained from workflows 2 and 4. The networks trained by workflows 2 and 4 act as initialization and regularization constraints. The pretraining process can act as a regularizer (Erhan et al., 2010) to introduce a helpful prior and implicitly minimize the appropriate parameters' range for the next steps of fine-tuning training. For high-level abstraction learning in a deep architecture, the regularizer imposes some constraints on the parameters to direct the minima where the cost function seeks. As in workflows 2 and 4, true reflectivity is used as one input channel. This helps to reduce the neural network parameter space and provides fine-tuned neural networks in workflows 1 and 3 with an initial model to train on. Even though the input channel changes to a smoothed background reflectivity, the neural network will learn the critical reflector information. Furthermore, the pre-trained models prevent networks 1 and 3 from creating new reflectors or artifacts not present in their inputs. Because the migrations from sharp velocities used in workflows 2 and 4 are impossible in practice, these networks cannot be used directly during inference. Instead, they help to initialize and constrain the other networks for optimization. A detailed description is shown in Figure 3.4, where reflectivity and RTM/RTMM images are the input for training the neural network. This

neural network plan is similar to the idea of ensemble learning (Hansen and Salamon, 1990; Krogh and Vedelsby, 1994; Zhou, 2021), which combines several learning algorithms to solve the same problem for obtaining a better prediction. This paper’s workflows share the same training neural network structure but with different training inputs. The difference is that workflows 2 and 4 are first trained using the true band-limited reflectivity and corresponding RTM/RTMM images obtained from scenarios 2 and 4, respectively. Then, workflows 1 and 3 use the pre-trained models R2 and R4 to fine-tune the neural network given on a smoothed input, whose RTM and RTMM images are generated from scenarios 1 and 3 and produce the updated models R1 and R3. Model R1, meaning the network trained from workflow 1, is then used to predict our baseline model, which is a result that we can easily obtain but want to improve.

Note that the smoothed input now is the reflectivity calculated from the background smoothed velocity. The reason for smoothing the input is that the neural network tolerates slight incorrect velocity errors more. After learning patterns from smoothed inputs, the neural network can distinguish reflectivity events from crosstalk or artifacts. This process can mitigate unexpected noise from migrated images and result in high accuracy and resolution. The comparison between different model outputs and performance is illustrated in detail in the numerical result section. We expect that, in general, models R1 and R3 will produce better images than R2 and R4. Furthermore, we expect that R3 can take advantage of multiple reflections’ wide illumination and information to predict an improved reflectivity.

Although scenario 4’s RTMM image is fed into workflow 4 to train a preconditioned model R4, we want to let the fine-tuned model R3 learn and predict an accurate image close to scenario 4’s output. To clarify the whole process, let us recall the definitions of scenario 4, workflow 4 and model 4: scenario 4 uses true velocity to obtain the RTM with surface multiples; workflow 4 uses the true reflectivity and output, as we would have in scenario 4. In this workflow, a U-Net trains the input with two channels: true reflectivity and RTMM images from scenario 4. The model R4 is then stored from workflow 4 and will be used

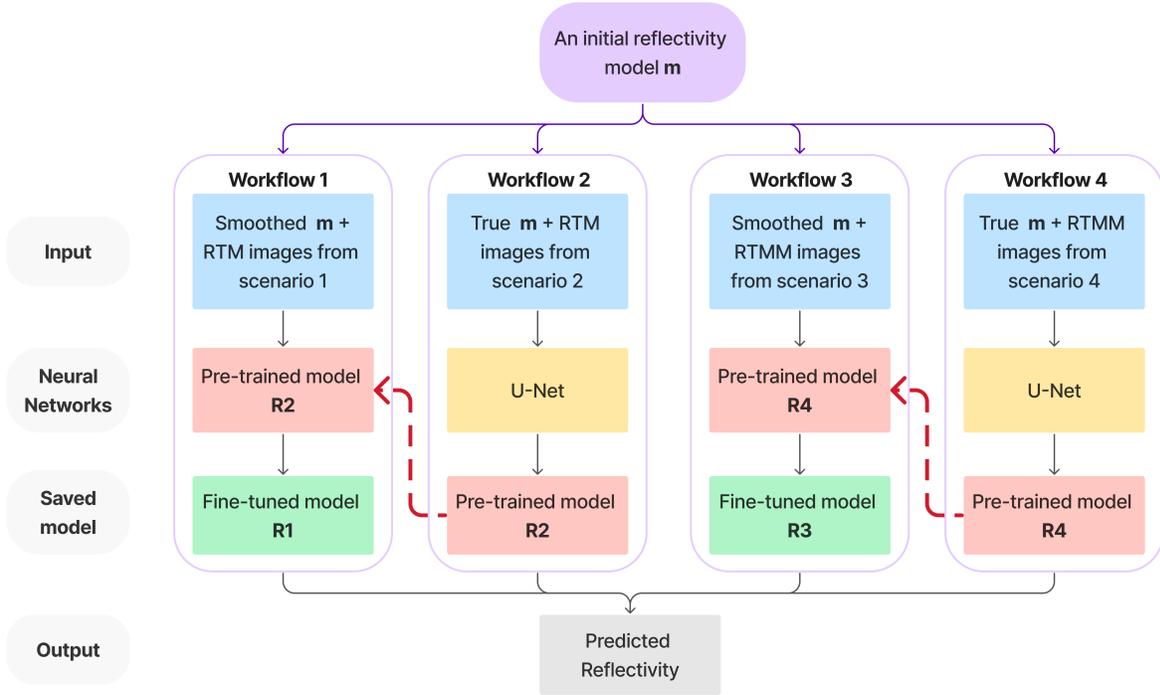


Figure 3.4: Neural network model plan for four scenarios.

as a pre-trained model for workflow 3. After training, the model R3 will be a fine-tuned neural network obtained from workflow 3, which uses smooth reflectivity and RTMM images as input. The initial model for R3 was R4 (from workflow 4); therefore, there will be an improvement from model R4.

### 3.4 Train and test set

We chose a series of common velocity models for training: Sigsbee2b, Amoco, Pluto, BP2004, Marmousi I and II, and others we built arbitrarily. We generated synthetic data, migrated all these velocity models, and used their RTM/RTMM images as training data. We calculated their reflectivities from their true velocities as training labels and made them band-limited by convolving with a time-domain 25 Hz Ricker wavelet. The shots and receivers were located just below the surface with 160 and 16 m spacing, using a cell size of 8 m. The sources

are Ricker wavelets with a 20 Hz dominant frequency. The total record time was variable and longer for the deeper salt models, with a maximum of 7.2 s for the Sigsbee2b case. For the modeling and migration, we used a fourth-order finite-difference method with a 15 Hz dominant frequency, implemented with CUDA for GPUs (Trad, 2022).

We employed data augmentation techniques such as image resizing and smoothing to increase the dataset instead of image rotation and flipping. Although very common in computer vision, these last two techniques are inappropriate in seismic imaging, because physical and geological principles constrain geophysical images. The vertical direction represents depth, while the horizontal direction denotes offset. Rotating or flipping the images would violate the fundamental principle that migration results are obtained from seismic wave propagation. We defined shots on the surface and simulated them to generate seismic waves that were then extrapolated into the subsurface. These waves were reflected and transmitted by subsurface structures, and the receivers on the surface generated shot records. The final step is to use these shot records to migrate reflections to their correct positions and create subsurface images. Therefore, we refrained from using rotation or flipping as a data augmentation method. Furthermore, no new data points were created in the input.

Our baseline model R1 is trained on workflow 1, corresponding to scenario 1, using smoothed reflectivity calculated from the background velocity and RTM images without multiples as the input channels. On the other hand, the proposed model R3 uses RTM images with multiple energy as one of the inputs. Before training, the RTM and RTMM images of scenarios 2 and 4 are divided randomly into 2700 spatial windows with  $256 \times 256$  grid points. For example, the Pluto model has  $601 \times 1750$  points. Suppose the random sequence numbers for horizontal distance and depth are 100 and 50, respectively. In that case, a chosen window should be located in the original model with offset numbers 100-356 and 50-306 points in depth, because the window size is 256 points in a square shape. Working in windows is not only practical for handling large images but also introduces a regularization effect since, if the predictions are correct, they should contain the same information where

they overlap. If the predictions are not similar, the summation of predictions from different windows will reduce the resolution.

As for scenarios 1 and 3, RTM/RTMM images are separated into 2500 subwindows. The train and validation set ratio is 0.8:0.2. We chose these numbers because a large training data size can help generalization. The maximum number of iterations for each training model is limited to 200 using the Adam optimizer with a batch size of 64. The learning rate is reduced during iterations to avoid the solution falling into local minima.

Then, we test our neural networks on three examples: the Canadian Foothills, a 2D slice of the Overthrust, and the SEAM Phase 1 geology models. These examples were not used during the training to test generalization; that is, how the neural network performs on new data. In the next section, we show a detailed comparison between neural networks in different scenarios and situations.

### 3.5 Results and discussions

This section tests predictions for the Canadian Foothills, Overthrust, and SEAM examples by independently working through workflows 1, 3, and 4. These examples test the neural networks' capability for generalization. The spatial interval for each example is 8 m with an 0.8 ms time sampling rate. As discussed previously, we can use the prediction from model R4, the network trained from workflow 4, as a reference and a regularization network. This result corresponds to a neural network trained using true band-limited reflectivity and RTMM images. However, the results could be better, because the inputs for inference are migrations with wavefield inconsistency (data with multiples migrated with smooth velocity). The model R1 from workflow 1 is our baseline model trained on a smoothed input without multiple energy.

On the other hand, although workflow 3 is similar to workflow 1, utilizing a smoothed reflectivity input, model R3 has RTMM images as input instead of RTM images. After learning patterns from multiple reflections' energy and smoothed inputs, model R3 can

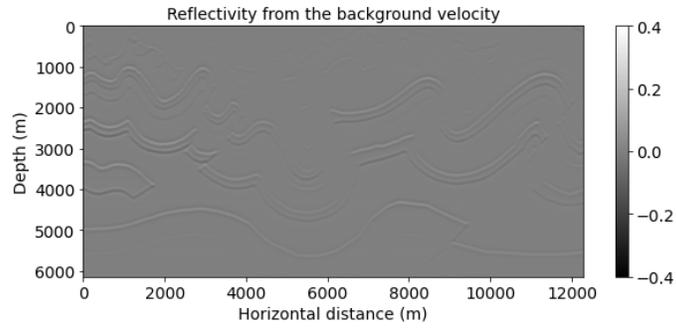
distinguish reflectivity events from crosstalk or artifacts. This process can mitigate migrated noise and improve resolution. Detailed numerical analysis and comparison is shown in the next part.

### 3.5.1 Example 1: Canadian Foothills

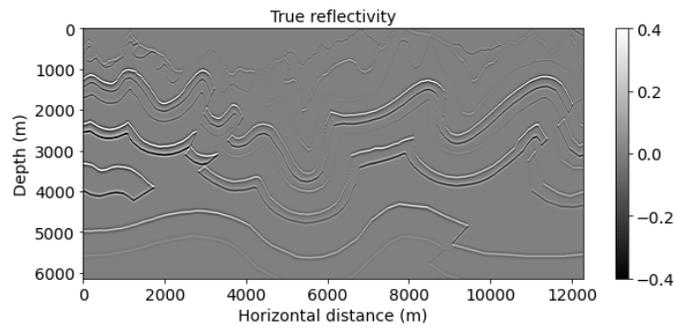
The Canadian Foothills example initially has  $1000 \times 1600$  points, but we chose  $768 \times 1536$  gridpoints for the neural network prediction. We simulated 78 shots and 795 receivers at the near-surface, with 160 and 16 m spacing separately. Figure 3.3 shows the results of the neural network models R1, R3, and R4 predictions on workflows 1, 3, and 4, correspondingly. A smoothed reflectivity generated from the background velocity (Figure 3.3a) is the first input channel for models R1, R3, and R4. Note that the smooth reflectivity input (Figure 3.3a) is in its original value, which does not have high frequencies due to smoothing, but the amplitude will be scaled during testing. RTM image (Figure 3.3c) is set as the second input for model R1; on the other hand, RTM image with multiples (Figure 3.3d) is used as the second input channel for models R3 and R4. The migration of the Canadian Foothills model using multiples with smooth velocity (Figure 3.3d) has increased illumination and artifacts due to the wavefield inconsistency described earlier. The result for workflow 3 (model R3 (Figure 3.3f)) tries to correct for these inconsistencies and shows somewhat better lateral event continuity with artifact reduction in comparison with model R1 prediction (Figure 3.3e), which did not use multiples. For example, the shallow curvatures in the middle horizontal distance can be seen clearly in Figure 3.3f, with higher resolution and less noise (Table 3.3) in comparison with Figure 3.3e. Additionally, compared with the model R4 result (Figure 3.3g), which is set as our reference, model R3 (Figure 3.3f) can also give a more accurate prediction of geological structures with improved resolution, which is closer to the true band-limited reflectivity (Figure 3.3b) calculated directly from the velocity model.

The example was also tested with LSRTM, and after 15 iterations, the resulting image is displayed in Figure 3.3i. Compared with RTM and RTMM results, the LSRTM image

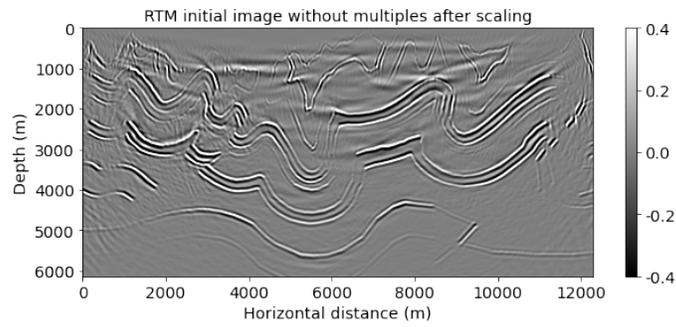
provides additional high-resolution information about the reflectors, particularly the side curvature boundary between 10,000 and 12,000 m at a depth of approximately 2000 m. However, the computational cost of LSRTM is at least twice that of RTMM for one iteration, and it requires several hours to complete 15 iterations, even when using high-performance computing by OpenMPI. This time is longer than required for RTMM calculation and neural network training. On the other hand, the model R3 result, which is a fast approximation of the LSRTM output, can recover most of the reflectors with noise suppression. Therefore, this result confirms that the proposed method achieves enhanced efficiency in reflectivity calculation compared with the LSRTM method.



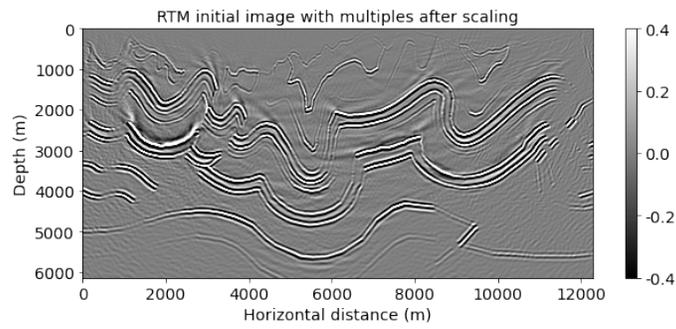
(a)



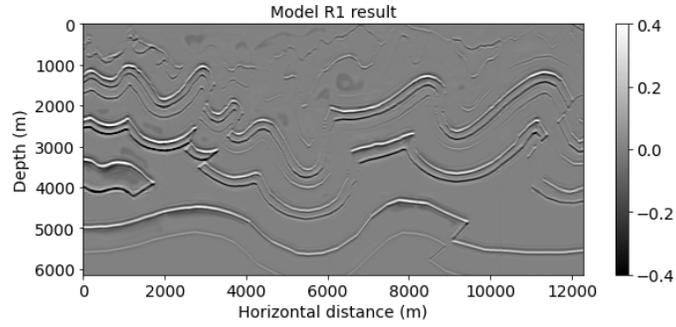
(b)



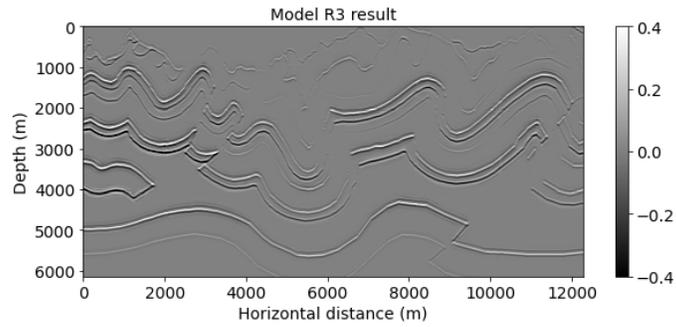
(c)



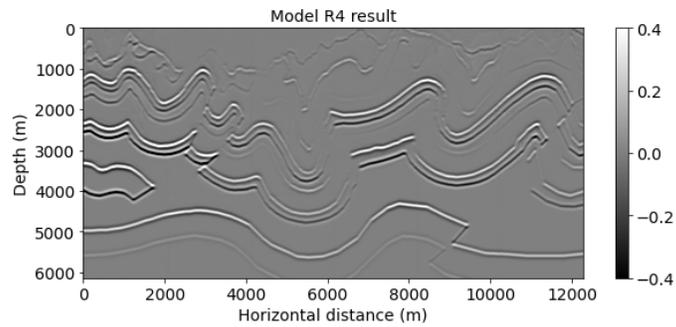
(d)



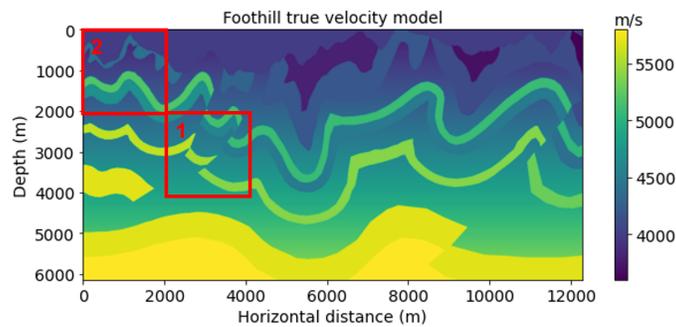
(e)



(f)



(g)



(h)

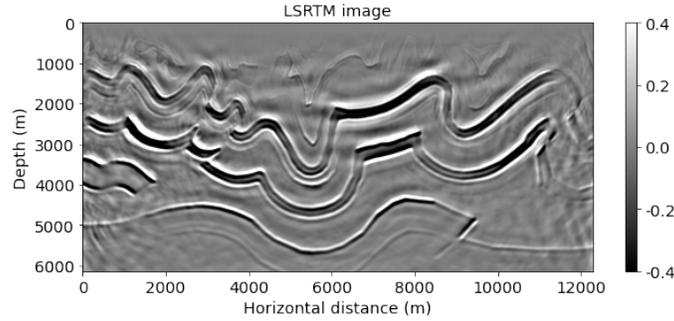


Figure 3.3: Canadian Foothills model results: (a) reflectivity from the background velocity, (b) true band-limited reflectivity, (c) RTM image without multiple reflections, (d) RTM image with multiple reflections, (e) model R1 result based on workflow 1, (f) model R3 result based on workflow 3, (g) model R4 result based on workflow 4, (h) true Foothills velocity, and (i) LSRTM result after 15 iterations. The boxes indicate areas shown in detail in the next figures.

In Figure 3.4, we compare the average amplitude spectrum of the results of the different networks. The reflectivity obtained from the smooth velocity model and used as an input channel (long dashed line) has lost low and high frequencies as expected, since it comes from a background velocity. Even though model R1 (point dashed line) can aid in recovering low frequencies between 0.002 and around  $0.002 \text{ m}^{-1}$ , model R3 (solid line) predicts a broader frequency band and higher values after about  $0.008 \text{ m}^{-1}$ . This observation indicates that model R3 takes advantage of true band-limited reflectivity on the low-frequency band, which is learned from the pre-trained network R4 and the multiple energy from RTMM images on high frequencies. Thus, model R3 can predict and rebuild more information on low and high frequencies, promoting output resolution and accuracy.

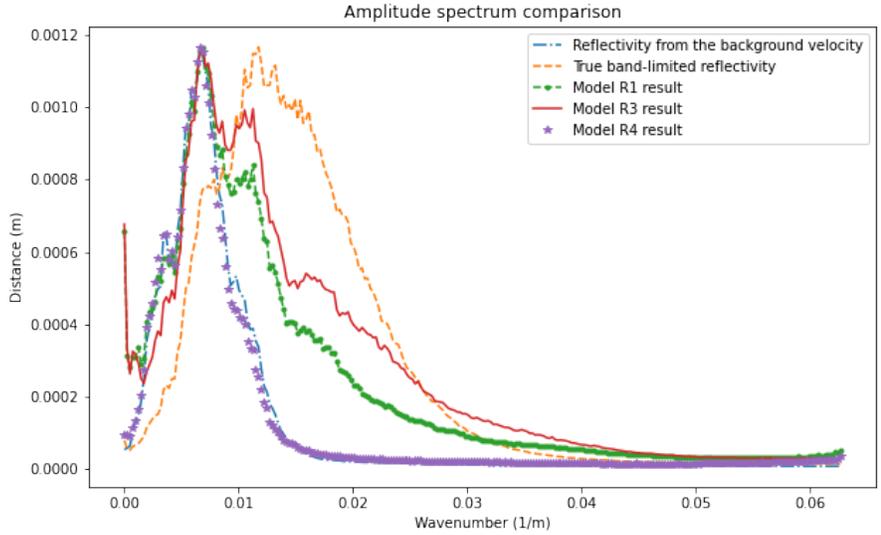


Figure 3.4: Amplitude spectrum comparison between models R1, R3, and R4 results for the Canadian Foothills example.

When comparing the peak signal-to-noise ratio shown in Table 3.3, the model R3 application has the highest value of the three results, which means it has the most confidence in predictions compared with other outputs. Most of the structural information of the true band-limited reflectivity image is preserved.

Table 3: PSNR (dB) comparison for Foothills example			
Prediction	Model R1	<b>Model R3</b>	Model R4
Total Foothills	24.84	<b>25.80</b>	20.59
Example 1	20.88	<b>22.58</b>	16.40
Example 2	19.16	<b>20.18</b>	17.03

Table 3.3: PSNR (dB) comparison for Foothills example.

Figure 3.5 contains the same information as Figure 3.3 but zoomed into a window at middle depth (red block No. 1 in Figure 3.3h). A normalization scaling is applied to make

the smooth reflectivity input more visible. The migrated image with multiple energy (Figure 3.5d) shows more illumination than the regular (no-multiples) RTM image (Figure 3.5c). For instance, the small fault on the top right can be migrated with higher illumination in the RTMM image (Figure 3.5d). The prediction from model R3 (Figure 3.5f) shows fewer artifacts and enhanced resolution and accuracy, whereas the prediction from model R1 (Figure 3.5e) shows artifacts at around 2800 m in depth. The forecast from model R4 (Figure 3.5g) is similar to the smooth reflectivity input. It shows no improvements as expected, since the network from workflow four was pre-trained using accurate inputs and cannot handle smoothing inputs.

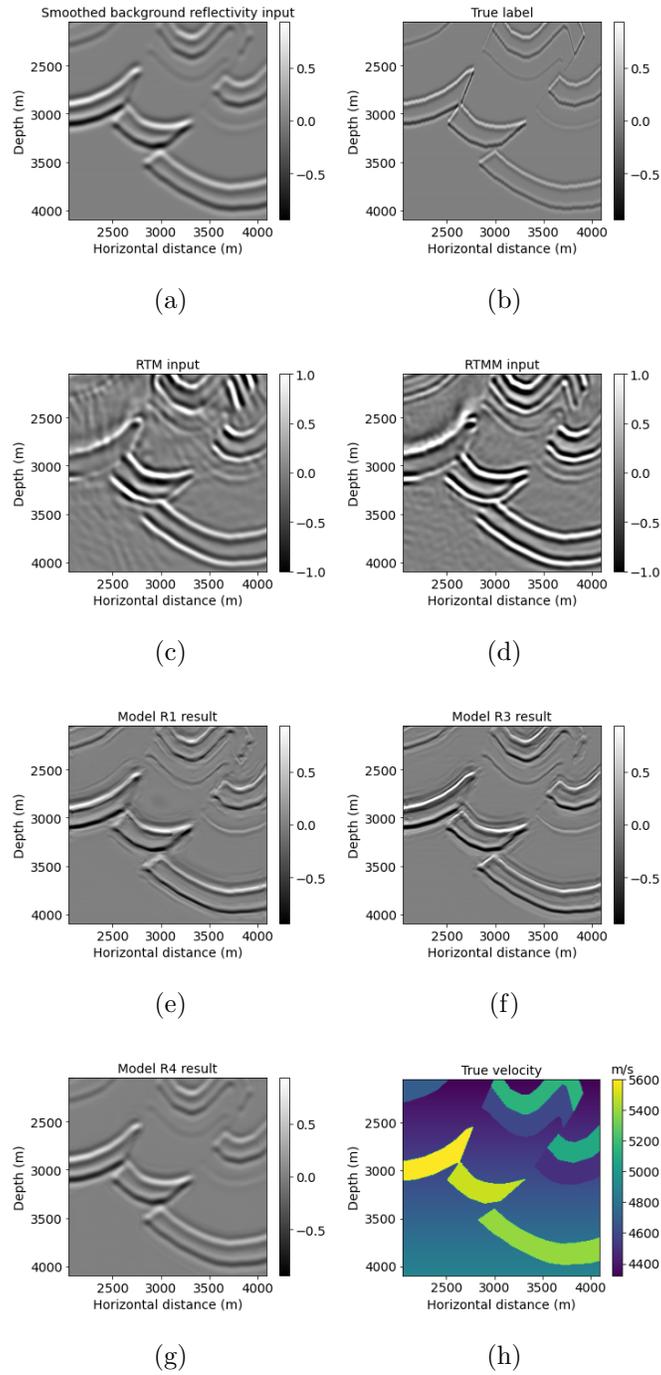


Figure 3.5: Foothills red box No. 1 results: (a) reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) RTM image with multiple energy, (e) model R1 result based on workflow 1, (f) model R3 result based on workflow 3, (g) model R4 result based on workflow 4, and (h) true windowed velocity.

Figure 3.6 shows a different (shallower) window (red block No. 2 in Figure 3.3h). The multiples used in the RTMM image (Figure 3.6d) predict workflow 3 (Figure 3.6f) better than the predictions from workflows 1 and 4 (Figure 3.6e,g). For example, the depth structure between 0 and 1000 m can be predicted with larger amplitude and accuracy in Figure 3.6f. Moreover, the model R3 result can have better lateral event continuity than Figure 3.6e,g. The PSNR value of model R3 in Table 3.3 is 20.18 dB, which is also the highest among other models, yielding that it can recover more reflectivity events at accurate locations.

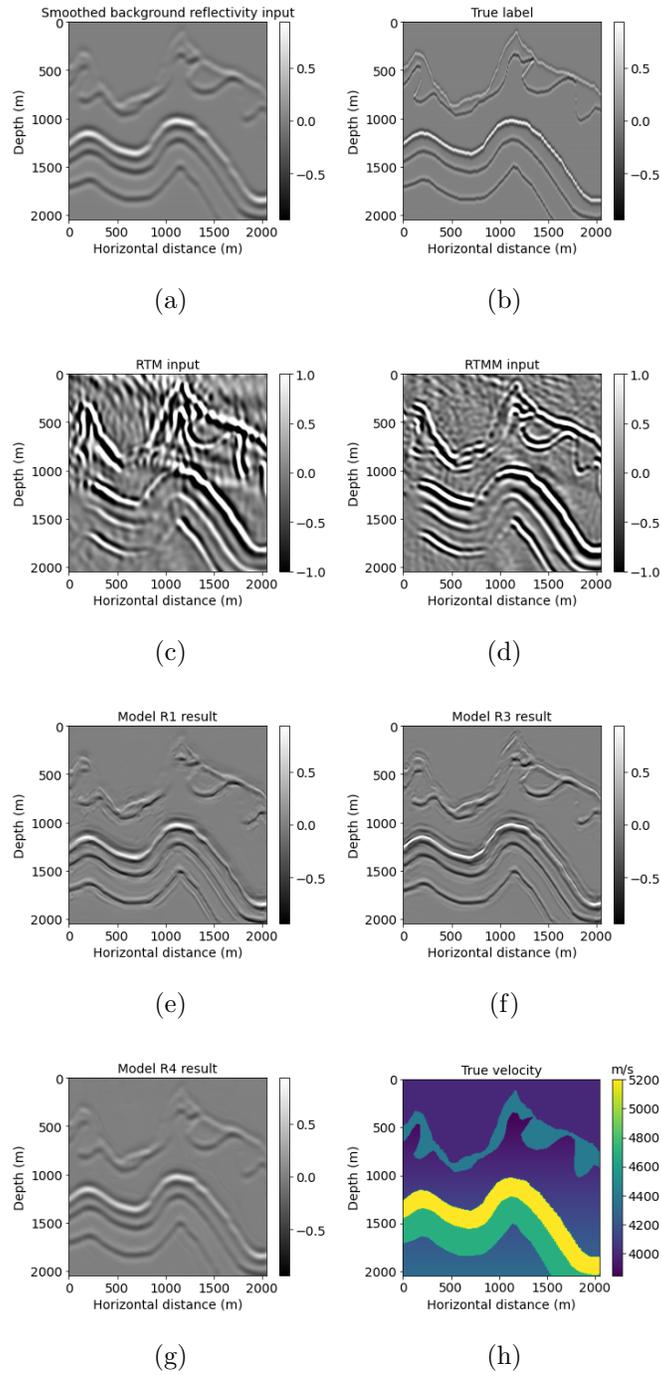


Figure 3.6: Foothills red box No. 2 results: (a) reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) RTM image with multiple energy, (e) model R1 result based on workflow 1, (f) model R3 result based on workflow 3, (g) model R4 result based on workflow 4, and (h) true windowed velocity.

Figure 3.7 denotes the crossplots between the two traces ( $x = 2400$  and  $8000$  m) from the true band-limited reflectivity of the Foothills example and the ones predicted using model R1 (diamond scattered points) and model R3 (round scattered points) separately. Both traces indicate that model R3, using multiple reflections, can predict a higher correlation with the true band-limited reflectivity values than model R1. Since the relation between the prediction and true label should be linear, the slope of model R3 results (round scattered points in Figure 3.7) is closer to one compared with model R1 (diamond scattered points).

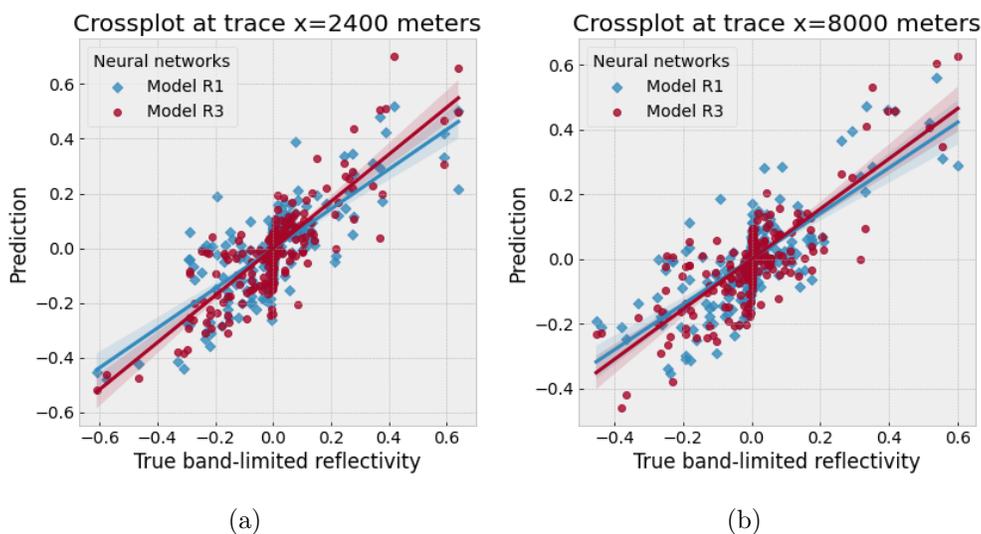


Figure 3.7: Crossplots for the Canadian Foothills example: the true band-limited reflectivity against the predicted reflectivity by using models R1 and R3, respectively.

### 3.5.2 Example 2: Overthrust

A 2D subset of a synthetic 3D Overthrust model is the second numerical example, representing a more complicated geological structure with thin layers. The example size is  $818 \times 1602$  points, with 79 shots simulated at the near-surface. For further neural network prediction purposes, we extract  $768 \times 1536$  points from the original example. Figure 3.8 represents the result for this Overthrust example. The smooth reflectivity is shown in Figure 3.8a. We can observe that the RTMM image (Figure 3.8d) yields extended subsurface illumination

compared with the RTM image (Figure 3.8c). Correspondingly, model R3 prediction using smooth input and multiple energy in Figure 3.8f still provides better-augmented resolution and accuracy than model R1 and R4 results (Figure 3.8e,g). The PSNR value of model R3 results in Table 3.4 is 24.61 dB, whereas model R1 and R4 results are 24.09 dB and 20.61 dB, respectively.

For the windowed example (red box in Figure 3.8h), Figure 3.9f displays enhanced information for the reverse fault from model R3 prediction in comparison with models R1 and R4 (Figure 3.9e,g). For example, the top layer above the small Overthrust at around 2500 m depth can be seen clearly with fined resolution in Figure 3.9f, whereas Figure 3.9e,g gives blurred and smoothed predictions. Furthermore, model R3 can recover the lateral variations with significant amplitude for the thin-layer structures below the reverse fault. The PSNR of the model R3 result is the highest value, 20.11 dB, shown in Table 3.4.

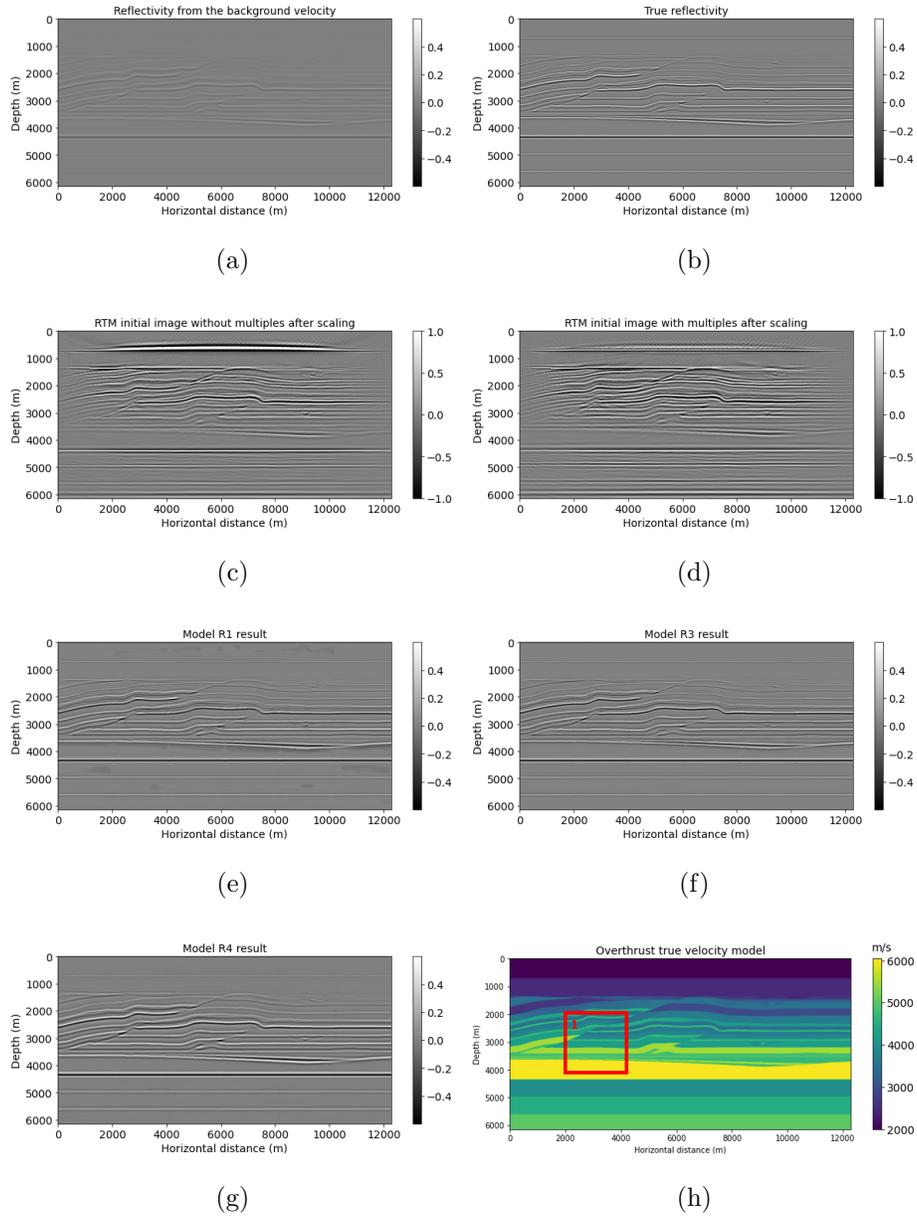


Figure 3.8: Overthrust model results: (a) reflectivity from the background velocity, (b) true band-limited reflectivity, (c) RTM image without multiple energy, (d) RTM image with multiple energy, (e) model R1 result based on workflow 1, (f) model R3 result based on workflow 3, (g) model R4 result based on workflow 4, and (h) true Overthrust velocity.

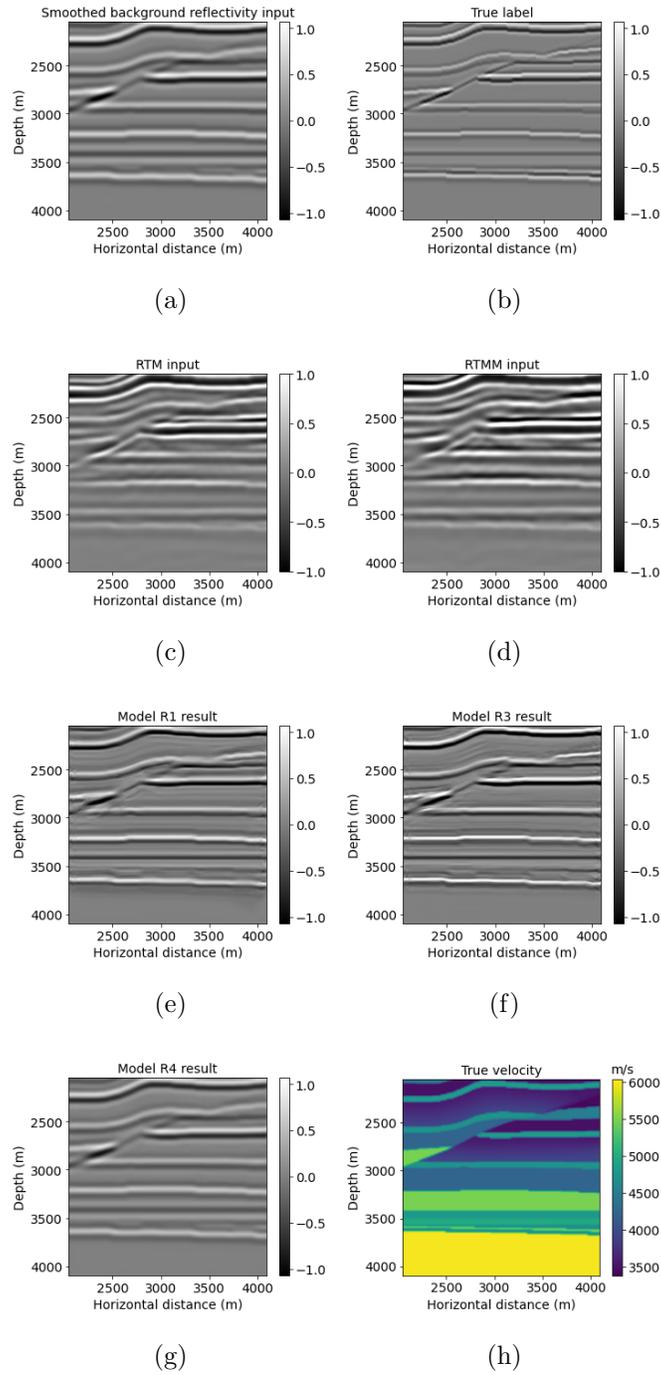


Figure 3.9: Overthrust red box results: (a) reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) RTM image with multiple energy, (e) model R1 result based on workflow 1, (f) model R3 result based on workflow 3, (g) model R4 result based on workflow 4, and (h) true windowed velocity.

Table 4: PSNR (dB) comparison for Overthrust example			
Prediction	Model R1	<b>Model R3</b>	Model R4
Total Overthrust	24.09	<b>24.61</b>	20.61
Example 1	19.06	<b>20.11</b>	15.99

Table 3.4: PSNR (dB) comparison for Overthrust example.

### 3.5.3 Example 3: SEAM Phase 1

The two examples above show our proposed neural network R3 can handle complex subsurface structures with faults and thin layers. We use another example to show model R3’s generalization ability on a more complicated model with thinner layers, folds, and a salt body. Figure 3.10 shows the SEAM Phase 1 velocity model. Note that SEAM Phase 1 has not been used in training or validating neural network workflow. We extracted a part of the original model with  $801 \times 1301$  points. There are 35 shots and 250 receivers separated by 240 and 40 m, with a 15 Hz dominant Ricker wavelet. We chose sparse source-receiver coordination, because insufficient obtained data are normal in a real case. We want to explore the power of multiple reflection energy and neural network applied in this project to see if the pre-trained neural network can classify useful multiple reflections from noisy data. The total time recording length is 7.2 s, with a 0.8 ms sampling rate. Since we resized this model to  $768 \times 1024$  points to be fed into our pre-trained neural network, the maximum offset is changed to 8.192 km. A smoothed background velocity is input to the reverse-time migration to avoid accurate information leakage.

Similar to previous examples, reverse-time migration after using multiple reflections (Figure 3.10d) gives a more accurate top layer structure than without using multiples (Figure 3.10c). Combined with PSNR comparison (Table 3.5), the model R3 prediction shown in Figure 3.10f improved reflectivity resolution and precision. Its PSNR is 26.32, which is larger than the model R1 prediction of 24.58.

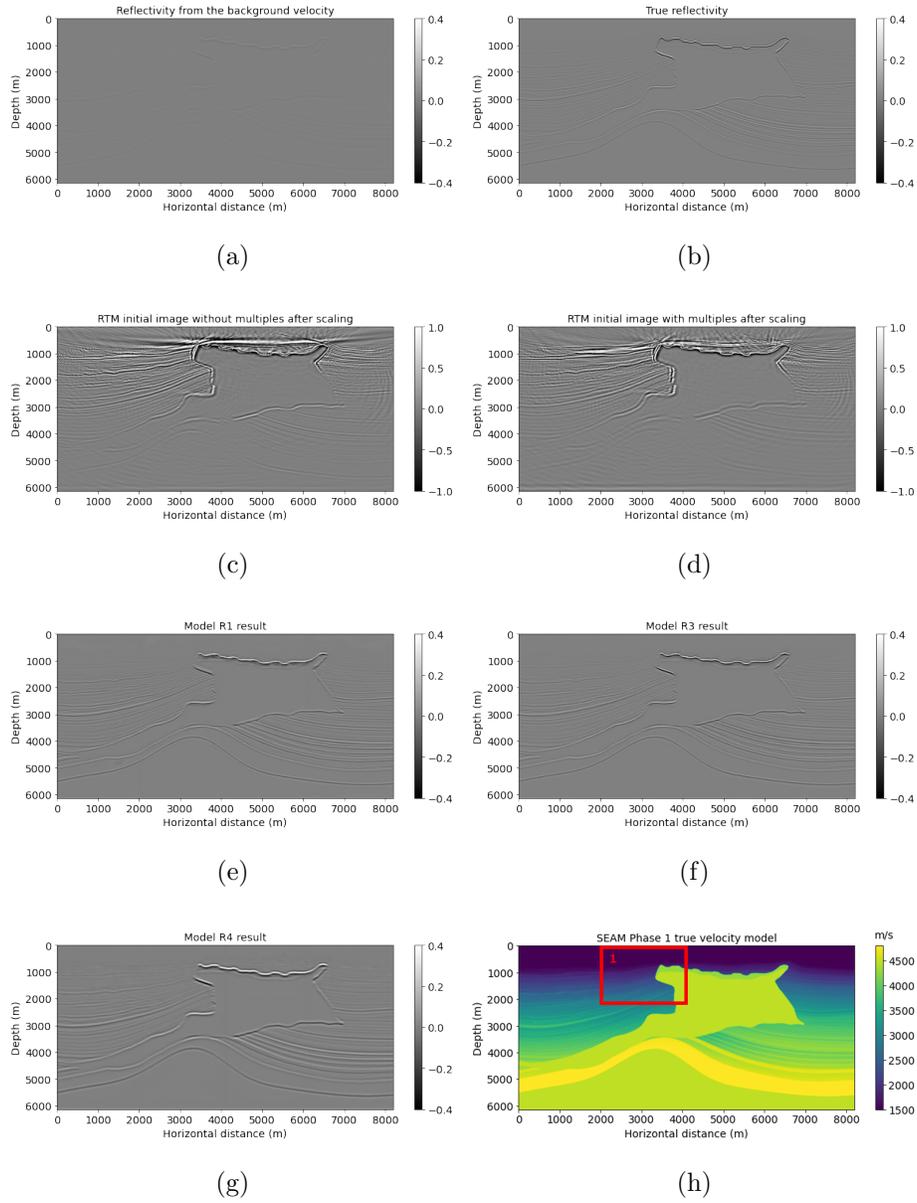


Figure 3.10: SEAM model results: (a) reflectivity from the background velocity, (b) true band-limited reflectivity, (c) RTM image without multiple energy, (d) RTM image with multiple energy, (e) model R1 result based on workflow 1, (f) model R3 result based on workflow 3, (g) model R4 result based on workflow 4, and (h) true SEAM velocity.

We list a windowed example in Figure 3.11 for a detailed comparison. Model R3 result (Figure 3.11f) can indicate clearer events compared with model R1 prediction (Figure 3.11e). For example, in Figure 3.11e, model R1 result has a more blurred top layer with artifacts

near the top above 800 m depth, whereas the prediction generated by model R3 at that area is clear with enhanced quality. For deeper events, model R3 also provides high resolution for dipping reflectors below 1500 m. Accordingly, the PSNR of this windowed example obtained by model R3 is 22.88, which is higher than the model R1 result.

Table 5: PSNR (dB) comparison for SEAM example.			
<b>Prediction</b>	<b>Model R1</b>	<b>Model R3</b>	<b>Model R4</b>
Total SEAM	24.58	<b>26.32</b>	23.75
Example 1	21.52	<b>22.88</b>	20.14

Table 3.5: PSNR (dB) comparison for SEAM example.

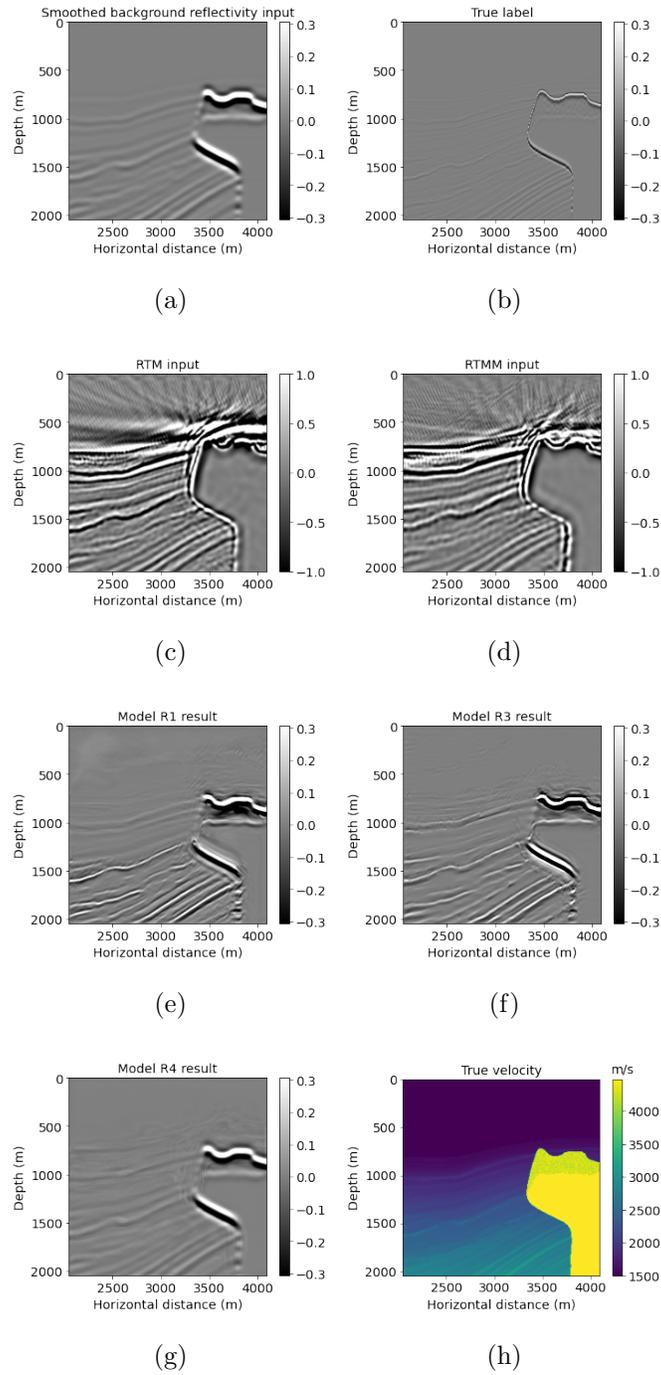


Figure 3.11: SEAM Phase 1 red box results: (a) reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) RTM image with multiple energy, (e) model R1 result based on workflow 1, (f) model R3 result based on workflow 3, (g) model R4 result based on workflow 4, and (h) true windowed velocity.

Furthermore, the crossplot for this example, shown in Figure 3.12, proves that model R3 prediction (round scatter points) gives a higher correlation with the true band-limited reflectivity compared with model R1 prediction (diamond scatter points).

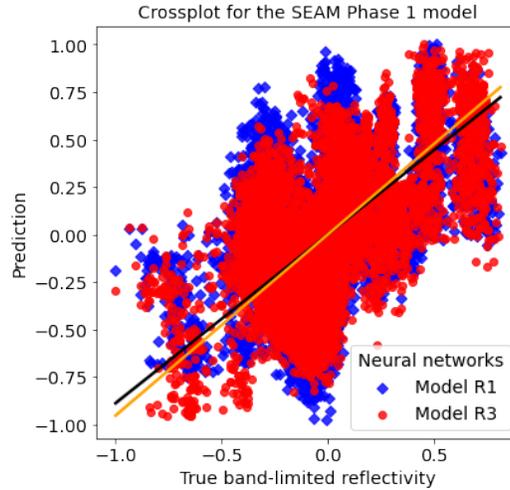


Figure 3.12: Crossplots for the SEAM Phase 1 example: the true band-limited reflectivity against the predicted reflectivity by using the model R1 and R3, respectively.

### 3.5.4 Discussion

***Input channels selection:*** We relied mostly on intuition and experimentation for deciding the input channels. For example, we compared results with a background velocity instead of a smooth reflectivity for the second input channel, but the results deteriorated. Although both results were alike, we observed more artifacts when using the background velocity, in particular at the boundaries between windows. This observation suggests that the network learns to calculate the directional derivative from the velocity in the second case, which accentuates the footprint caused by window overlapping. This example shows that often we can understand what the network does simply by experimenting with different inputs.

***True labels/output selection:*** The currently proposed method uses a band-limited reflectivity as the true label for training, generated from the training velocity models by converting to time, convolving with a wavelet, and converting back to depth. This is one

of many possible choices. For example, we also tried to use the full bandwidth reflectivity with the expectation of extending the frequency band from the inputs when making the inference. The average spectrum for the output was undoubtedly more expansive than a band-limited label, but the results show many artifacts. We can explain this result by the network learning to perform deconvolution, which is sensitive to noise levels larger than the signal as we move away from the dominant frequencies. Once again, we can understand the network by experimenting with different inputs and outputs and conclude that the same constraints as classical processing limit the network. It calculates the operators by extracting the information from the data instead of hard-coded rules.

***U-Net architecture selection:*** For the U-Net architecture, we tried a shallower network as well, with five blocks of convolutional layers on the encoder and decoder parts, instead of the six blocks shown in Figure 3.3. The results were blurry for thin layers and small structures and not as good as the U-Net results in this paper. This indicates that small details extracted by the sixth block, shown at the bottom in Figure 3.3, were important for training and prediction. For the input size, we chose to use  $256 \times 256$  points with two channels. We also tried other sizes, larger and smaller and square and rectangular, but the chosen size seemed to be optimal for this problem. For the encoder part, the filter shape changes from 16 to 512, and the kernel size decreases from 11 to 1 as the filter shape increases. There is no stride included in the convolutional layers. The padding is set to “same”, preserving the input size for each layer. The activation functions for all the convolutional layers were set to Rectified Linear Unit (ReLU). To initialize the layer weights, we use the “He” initializer (He et al., 2015), which draws samples from a truncated normal distribution centered on zero. For the downsampling, we use maxpooling with a  $2 \times 2$  size. A batch normalization layer was applied after each activation function. Additionally, we combined “drop out” with batch normalization in each block to reduce generalization errors.

***Mean squared error (MSE) and mean absolute error (MAE):*** Mean squared error (MSE) penalizes larger prediction errors compared with mean absolute error (MAE),

because significant errors are emphasized and have a relatively greater effect on the value of the performance metric. After testing with MSE and MAE, respectively, we found MAE results can be lower than MSE in workflows 2 and 4, whereas they perform worse when applied in workflows 1 and 3. The result indicates the MAE can handle well with accurate input, but it has limits on smooth input.

### 3.5.5 Model Performance and Computation Time

Figure 3.13 illustrates the model loss comparison between RTM-CNN and RTMM-CNN with fifty iterations when the starting learning rate is 0.001. Due to the true band-limited reflectivity input, model R4 (point dashed line) provides the lowest loss value. The model R3 (solid line) can converge to a smaller loss value, e.g., 0.0005, than the baseline model R1 (dashed line) for a smoothed input. For the validation loss shown in Figure 3.13, model R4 (point dashed line) gives the lowest loss value after 50 iterations, as it is used as a regularizer. Model R3 (solid line) can still converge to a smaller value than model R1 (dashed line), meaning model R3 works better in the validation set.

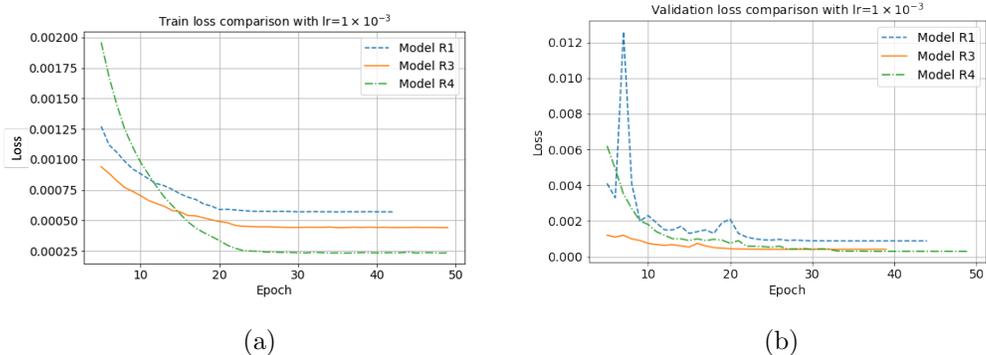


Figure 3.13: (a) Model train loss and (b) validation loss comparison between different neural network models with fifty iterations.

When training the network, each iteration takes around 53 s. We set the number of iterations to a large number (200), but training usually stops at around 50 iterations, which

stops converging. Each neural network will take approximately 2650 s of runtime with an NVIDIA K80/T4 16GB GPU and 25.46GB RAM. The migration process is performed on an NVIDIA GeForce RTX 2080 Ti with 64 GB RAM. For the data preparation, each shot takes around one second for forward modeling and three seconds for imaging. Compared with the proposed method in this paper, LSRTM requires one forward modeling and one migration per iteration. The runtime for LSRTM can extend to several hours when the number of iterations is large. By contrast, the proposed method works directly with a single RTM calculation. The computational cost is reduced, because the cost of inference in neural networks is very small. Although the training can be computationally expensive, more than a regular LSRTM, this is performed only once. These synthetic examples were all under a 2D situation, so future work should consider an extension of this work to a 3D situation.

### 3.6 Conclusions

The proposed RTMM-CNN method, which incorporates multiple reflections in illumination, is capable of improving the quality of reflectivity obtained from migration, particularly when applied to a smooth initial model. The trained neural network takes advantage of multiple reflections and a reflectivity input from the background velocity model. The former enhances subsurface structure illumination, while the latter allows the neural network to accommodate for velocity errors. The network, trained with multiple reflections and a true velocity model, is a preconditioner that restricts the range of potential parameters due to its supplementary information. Once a smoothed reflectivity is fed into the pre-trained model, a new fine-tuned model can be obtained by further training to tolerate additional biases caused by preconditioning. The U-Net operator approximates the inverse of the Hessian, suppressing image artifacts and enhancing the resolution of reflectors. This paper represents an initial step towards using multiple reflections for subsurface imaging with U-Net in realistic scenarios with limited velocity information. Even though the neural network model exhibits robust generalization capabilities across diverse synthetic geology models, it has limitations on real

data when the velocity is unknown or inaccurate. It is expected that the effectiveness of the neural network will be further improved with the emergence of even smoother migration velocity models or advanced methods, for example, transfer learning or other network architectures.

# Chapter 4

## Discrete wavelet transform application in a CNN-based reverse time migration with multiple reflections

### 4.1 Summary

In seismic imaging, achieving high-resolution and accurate images poses several challenges. These challenges include limited aperture illumination of subsurface structures, inaccuracies and smoothness in the initial models, and constraints on computational resources. To address these limitations, deep learning techniques have emerged as a promising approach for enhancing image resolution and accuracy. In this chapter, a discrete wavelet transform (DWT) is deployed together with a U-Net for enhancing migrated images by using energy from multiples. The neural network approximates the inverse of the Hessian to obtain high-quality reflectivity prediction. The DWT adds physical constraints to the network inputs to

help distinguish signals from noise. Two synthetic examples are illustrated in this chapter. Results show that the subbands of the DWT help the network learn patterns from smoothed inputs, extract critical features from data, and enhance image resolution. Multiple reflections generated from a free surface condition provide valuable information for subsurface structure expanding prediction illumination. Furthermore, this method can handle thin layer structures with improved peak signal-to-noise ratio and resolution, because the higher the PSNR is, the better reconstructed image that matches the original image.

## 4.2 Introduction

Seismic imaging is a process for estimating rock parameters from seismic data (Schuster, 2020). Migration defines different methods of seismic imaging to detect and gain information from a subsurface structure. They generate reflectivity maps locating seismic events and collapsing energy to correct places (Gray et al., 2001). Compared with traditional depth migration methods, reverse time migration (RTM) (Baysal et al., 1983; Whitmore, 1983; McMechan, 1983; Levin, 1984) can handle lateral velocity variations, such as steep geologic structures. As an advanced migration approach, least-squares reverse time migration (LSRTM) (Dong et al., 2012) can update the reflectivity iteratively with improved accuracy. However, RTM and LSRTM sometimes produce poor image resolution because of limited illumination. One resource to expand the illumination aperture is using, in addition to the primary reflections, the multiple reflections (Liu et al., 2011; Li et al., 2017; Wang et al., 2017; Zhang et al., 2020). Although RTM with multiple energy (RTMM) can help refine image accuracy and resolution with the help of multiple reflections, as discussed in the previous chapter, this method is influenced by wave interference between multiple reflections, primary reflections and noise.

In recent years, deep learning has proven to be a helpful approach to enhancing seismic image resolution and the quality of interpretation. It can learn features from the non-linear relationship between seismic data and rock parameters. Many researchers have used deep

learning applications in RTM or LSRTM (Wu et al., 2018; Kaur et al., 2020b; Vamaraju et al., 2021; Torres and Sacchi, 2021, 2022; Zhang et al., 2022). These methods mitigate sampling artifacts by training a machine-learning network to filter them out and therefore, they can be used to increase the seismic resolution. These networks have a flexible algorithmic structure that can be used to enhance the recovery of information from seismic data. Mentioning the image recovery, discrete wavelet transforms combined with neural networks have been used as a tool for feature extraction (Ghazali et al., 2007), de-noising (Wang et al., 2010; Suraj et al., 2014), super-resolution in deep learning (Wu et al., 2022b), and seismic data reconstruction in geophysics (Liu et al., 2022).

The process of obtaining reflectivity prediction from seismic imaging results is similar to image recovery. Thus, to mitigate the problem of poor image quality because of limited aperture, a novel method is proposed in this chapter, applying a discrete wavelet transform in migrated images. Then, the transformed image is set as another input channel in the neural network proposed in Chapter 3. RTM with multiple reflections can expand subsurface illumination and improve image accuracy. Discrete wavelet transforms in deep learning, adding physical constraints, can learn features from migrated data and enhance image resolution.

### **4.3 Discrete Wavelet Transform (DWT)**

A discrete wavelet transform (DWT) is a wavelet transform that decomposes a signal into a set of wavelet basic functions of different frequencies. The DWT can perform multi-resolution signal analysis, which captures both frequency and time location information (Acharya and Ray, 2005). The wavelet transform decomposition for a digital signal includes lower-frequency and higher-frequency subbands. Lower-frequency subbands have finer frequency resolution and coarser time resolution compared to the higher-frequency subbands.

### 4.3.1 DWT in one dimension

Based on Mallat (1989) and Acharya and Ray (2005), the idea of multi-resolution analysis is to approximate a function  $f(t)$  at different levels of resolution. There are two functions in the multi-resolution analysis: the mother wavelet  $\psi(t)$  and the scaling function  $\phi(t)$ . Wavelet functions are dilated, translated and scaled versions of a common mother wavelet. The dilated and translated versions of the scaling function are given by:

$$\phi_{m,n}(t) = 2^{-m/2}\phi(2^{-m}t - n), \quad (4.1)$$

where  $m$  and  $n$  are integers. For a fixed  $m$ , the set of scaling functions  $\phi_{m,n}(t)$  are orthonormal. By the linear combinations of the scaling function and its translations, a set of functions can be generated

$$f(t) = \sum_n \alpha_n \phi_{m,n}(t). \quad (4.2)$$

The multi-resolution analysis decomposes a signal into two parts: approximation of the original signal from finer to coarser resolution and the detailed information lost due to the approximation.

$$f_m(t) = \sum_n a_{m+1,n} \phi_{m+1,n} + \sum_n c_{m+1,n} \psi_{m+1,n}, \quad (4.3)$$

where  $f_m(t)$  denotes the value of the input function  $f(t)$  at resolution  $2^m$ ,  $c_{m+1,n}$  is the detail information, and  $a_{m+1,n}$  is the coarser approximation of the signal at resolution  $2^{m+1}$ . The functions  $\phi_{m+1,n}$  and  $\psi_{m+1,n}$ , are the dilation and wavelet basis functions (orthonormal).

The decomposition of signals using the discrete wavelet transform can be expressed using finite impulse response (FIR) (Meddins, 2000) filters. The wavelet coefficients for the signal  $f(t)$  then can be defined by

$$\begin{cases} c_{m,n}(f) = \sum_k g_{2n-k} a_{m-1,k}(f), \\ a_{m,n}(f) = \sum_k h_{2n-k} a_{m-1,k}(f), \end{cases} \quad (4.4)$$

where  $g$  and  $h$  are the high-pass and low-pass filters. Since the synthesis filters  $h$  and  $g$  have been derived from the orthonormal basis functions  $\phi$  and  $\psi$ , these filters give exact reconstruction

$$a_{m-1,i}(f) = \sum_n h_{2n-i} a_{m,n}(f) + \sum_n g_{2n-i} c_{m,n}(f). \quad (4.5)$$

Given the input discrete signal  $x(n)$ , it will be filtered in parallel by both a low-pass filter ( $h$ ) and a high-pass filter ( $g$ ) at each transform level. The two output streams are then subsampled by simply dropping the alternate output samples in each stream to produce the low-pass subband  $y_L$  and high-pass subband  $y_H$ .

$$\begin{cases} y_L(n) = \sum_{i=0}^{\tau_L-1} h(i)x(2n-i), \\ y_H(n) = \sum_{i=0}^{\tau_H-1} g(i)x(2n-i), \end{cases} \quad (4.6)$$

where  $\tau_L$ , and  $\tau_H$  are the lengths of the low-pass ( $h$ ) and high-pass ( $g$ ) filters respectively.

### 4.3.2 DWT in two dimensions

A simple approach for the implementation of the DWT in two dimensions (2-D) (Mallat, 1989; Rioul and Vetterli, 1991; Xiong et al., 1998; Acharya and Ray, 2005) is to perform a standard one-dimensional (1-D) DWT row-wise to produce an intermediate result and then perform the same 1D DWT column-wise on this intermediate result to produce the final result. The two-dimensional scaling functions can be expressed as separable functions, which are the product of two one-dimensional scaling functions such as  $\phi_2(x, y) = \phi_1(x)\phi_1(y)$ , which is the same for the wavelet function  $\psi(x, y)$  as well.

If an image has  $M$  rows and  $N$  columns, applying the one-dimensional transform in each row, produces two subbands in each row with a size of  $M \times \frac{N}{2}$ . Then, applying a 1-D DWT column-wise on the subbands (intermediate result), produces the subbands LL, LH, HL, and HH with a size of  $\frac{M}{2} \times \frac{N}{2}$ , respectively. LH, HL and HH contain the input signal's high-frequency information around discontinuities (edges in an image). LL is a coarser version

of the original input signal and provides an input to the next decomposition level. The reconstruction is performed by operating in the reverse direction: first on columns, then on rows. Thus, separable 2-D DWT has three wavelet functions ( $m$  and  $n$  are coordinates of the input image):

$$\psi^1(m, n) = \phi(m)\psi(n) \quad \textit{LH wavelet}, \quad (4.7)$$

$$\psi^2(m, n) = \psi(m)\phi(n) \quad \textit{HL wavelet}, \quad (4.8)$$

$$\psi^3(m, n) = \psi(m)\psi(n) \quad \textit{HH wavelet}, \quad (4.9)$$

and one scale function:

$$\phi^2(m, n) = \phi(m)\phi(n) \quad \textit{Approximation LL}. \quad (4.10)$$

Figure 4.1 shows a block diagram of a 2-D DWT. An  $K$  level decomposition can be performed, resulting in  $3K + 1$  different frequency bands: LL is low frequency or approximation coefficients, and the wavelet image coefficients LH, HL, and HH which correspond, respectively, to vertical high frequencies (horizontal edges), horizontal high frequencies (vertical edges), and high frequencies in both directions (corners).

### 4.3.3 Neural network training strategy

In this project, a U-Net architecture (Ronneberger et al., 2015) with additional skip connection layers is proposed to learn patterns from migrated images and discrete wavelet transform (DWT) filtered images. The basic U-Net architecture is derived from previous work (Huang and Trad, 2023) mentioned in Chapter 3. An additional channel, the LL subband, is added to the neural network input. The reason for applying the LL subband is to help extract features from low-frequency information in addition to other input channels. This is intended to complement the information provided by other input channels and improve the accuracy of the model's feature extraction.

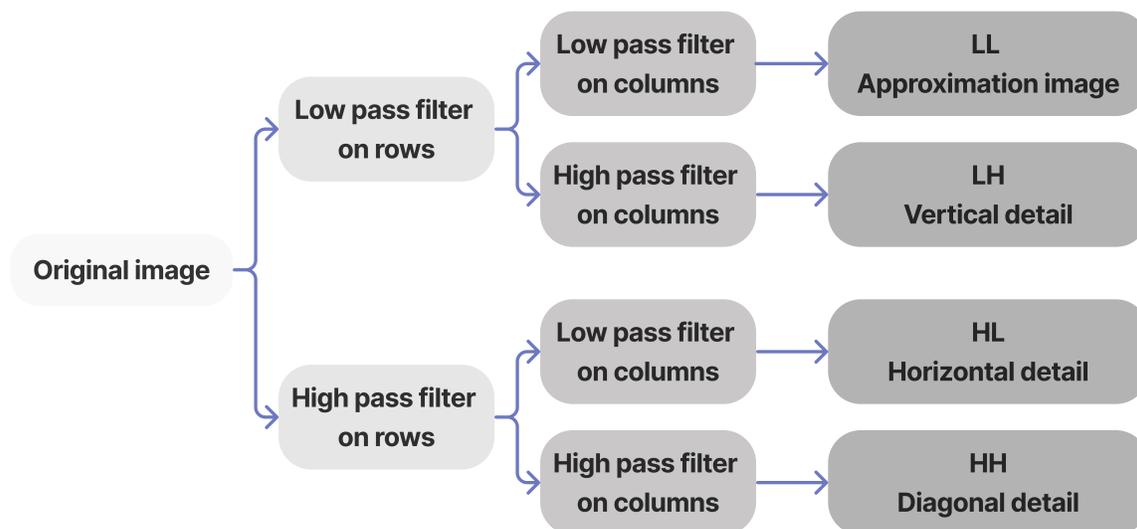


Figure 4.1: Block diagram of DWT.

Figure 4.2 illustrates the DWT-applied U-Net architecture. The encoder part of the network has three input channels: smooth background reflectivity, RTM images with multiple reflections, and corresponding LL subband images. The network then downsamples the original images into smaller sizes to learn key features from residuals and patterns in the data. The smoothed reflectivity and LL subband images add physical constraints of subsurface structures to the neural network. Subsurface structure key features are then upsampled to the original dimensions by transpose convolutions. Additional skip connections are utilized to work as identity mapping (He et al., 2016) and help to strengthen the training result with weak constraints.

The network operator acts as an approximation of the inverse of the Hessian (Kaur et al., 2020b; Torres and Sacchi, 2022) to filter migrated data into a predicted reflectivity model, but with more robust and physical data constraints in the input channel. The solution can be determined as follows:

$$\mathbf{m}_{pred} = \Gamma_{unet}(\mathbf{m}_{rtmm}, \mathbf{m}_{smooth}, \mathbf{m}_{rtmm_{DWT}}), \quad (4.11)$$

where  $\mathbf{m}_{rtmm}$  is the RTMM initial image,  $\mathbf{m}_{smooth}$  denotes the smooth background reflectivity model,  $\mathbf{m}_{rtmm_{DWT}}$  means DWT subband on RTMM image, and  $\mathbf{m}_{pred}$  represents the output reflection coefficient prediction.

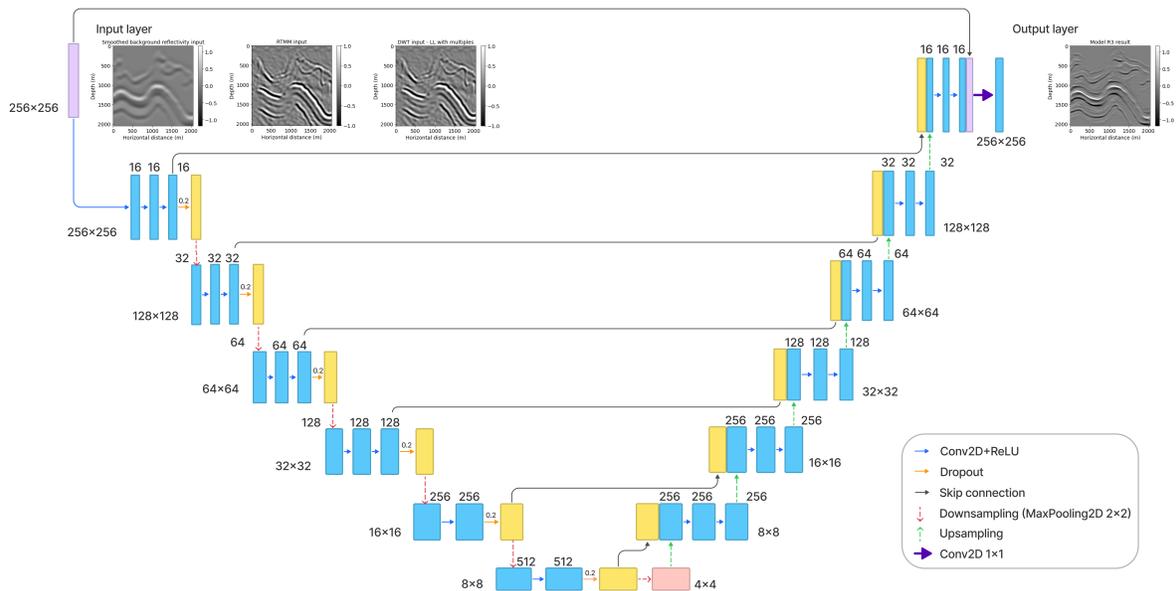


Figure 4.2: Architecture of DWT applied in the RTMM-CNN. The main structure is a U-Net with more skip connection layers. The input layer includes three channels: background reflectivity, RTMM image, and subband LL after using DWT of RTMM image. The predicted reflection coefficient is the output.

A mean squared error (MSE) loss function is used to penalize prediction errors. A peak signal-to-noise ratio (PSNR) is used to evaluate the model performance. Figure 4.3 illustrates the relation between the different network models and workflows. Similar to Chapter 3, models R2 and R4 are treated as pre-trained models using as inputs their true reflectivity, their migration images, and their corresponding DWT subband LL. The only difference between R2 and R4 is that model R2 uses RTM images with primaries only as input, whereas model R4 utilizes multiple reflections in the migration image as an input channel. The pre-conditioned models R2 and R4 can minimize the appropriate parameters range for the next steps of fine-tuning training. Then, models R1 and R3 are fine-tuned

based on R2 and R4's parameters, respectively. At this step, the input reflectivity of models R1 and R3 are smoothed instead of the true one. Additionally, RTMM images are fed into model R3, and RTM images are input to model R1 for fine-tuning separately.

In other words, model R4 primarily captures the true patterns of reflectivity from the input data. It produces smooth predictions when provided with inputs that are smooth or blurry. Hence, I selected the results obtained from model R4 as the reference for comparison. On the other hand, model R1 serves as the baseline network, while model R3, which considers multiple reflections, is expected to exhibit the best performance among the other models. In the upcoming section, I will provide a detailed description of the results obtained from models R1, R3, and R4, focusing on the influence of the DWT subband and multiple reflections.

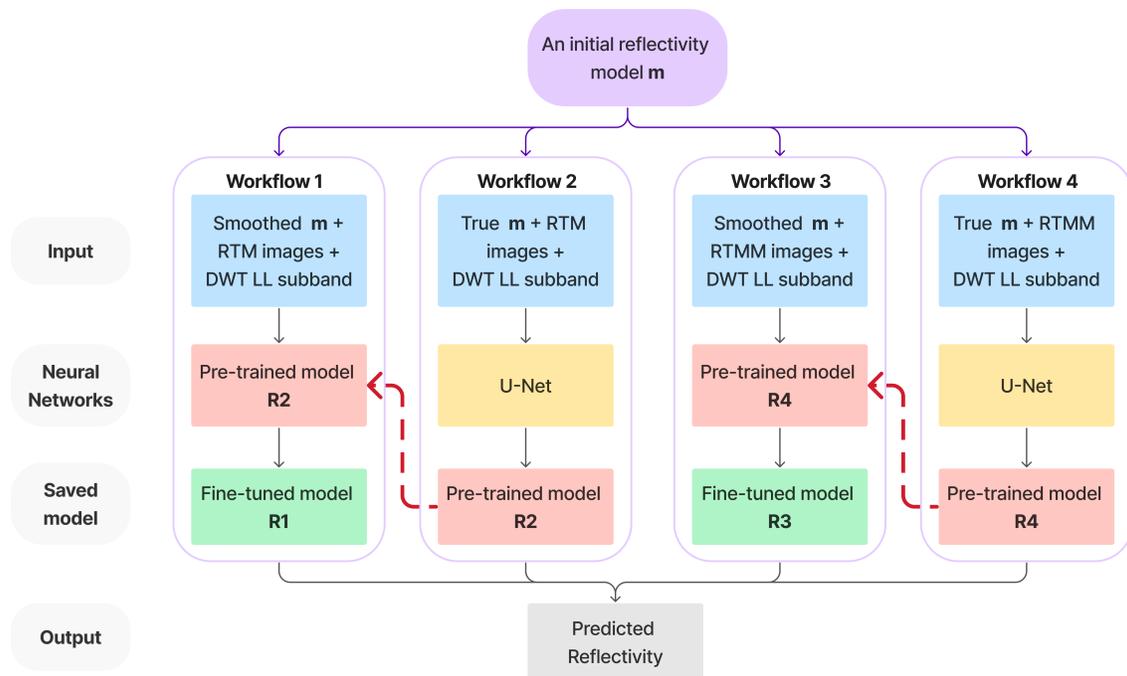


Figure 4.3: Neural network model plan and workflows.

## 4.4 Numerical examples

In this section, several numerical examples will be presented to demonstrate the performance of this approach in terms of refined reflectivity predictions, emphasizing the achieved high resolution and enhanced accuracy. Through these examples, I hope to provide a clear understanding of the advantages of this approach. The results will be showcased in a manner that emphasizes the strengths of the method and outlines the subsequent steps for further improvement.

### 4.4.1 Train and test set

For the train and test set used in this approach, I have used a variety of different velocity models, such as Sigsbee2b, Amoco, Agbami, Pluto, BP2004, Marmousi and others arbitrarily defined. Each model has a different grid size, but 8 meters is chosen as the spatial interval for each grid point. The source uses a 20 Hz dominant frequency of a Ricker wavelet. The total record time is 7.2 seconds with 0.8 milliseconds of temporal sampling. The shot and receiver intervals are 80 and 16 meters, respectively. A fourth-order finite difference method is used for the forward modelling. Velocity models are applied for the generation of shot records. Subsequently, the migration results, obtained by applying RTM or RTMM with these velocities and shot records, are utilized as one of the input channels.

Model R1 in Figure 4.3 is set as the baseline model without using multiple reflections. Before training, the whole RTM and RTMM images, and the corresponding DWT subband were partly chosen and divided randomly into 1900 different spatial windows with 256x256 points each. The train and test set ratio is the same as that in the previous chapter, 0.8:0.2. It should be noted that the validation set only includes a subset of the pixels from the migration images. Additionally, it is important to mention that all the inputs and outputs included in this chapter have been normalized for scaling purposes. By utilizing this normalization technique, the outputs obtained from the neural network model are consistent

to compare across different experiments. The performance of the model can then be evaluated based on metrics such as MSE and PSNR.

#### 4.4.2 Using DWT in the network input

For the neural network architecture in this chapter, I selected the initial migration images (RTM images in model R1, RTMM images in model R3), and the smoothed background reflectivity as the first two input channels. Additionally, the approximation image LL was considered as the third input channel since it inherits major patterns from the initial migration images and contains some low-frequency information because of the low pass filter. The subband LL can help extract information from the low-frequency range, while subbands HL and HH may not provide the same level of information. For example, Figure 4.4 shows different subbands after applying discrete wavelet transform on the Pluto geology model. Compared with HL and HH, subbands LL and LH can maintain much information on geologic structures. However, the subband LH (e.g. Figure 4.4c) has many discontinuities that may mislead the neural networks. Furthermore, the low-frequency subband LL can help avoid neural networks' strong dependency on other input channels, such as background reflectivity, or RTMM images.

This proposed technique has been evaluated on two different geology models: the Foothills and the Overthrust models. Those two models mentioned have not undergone training or testing within the neural network framework. They are resized to a grid size of  $768 \times 1536$ . This resizing is necessary to match the input channel size of the network, which is set at  $256 \times 256$ . In order to obtain a subband LL image with the same size as the input migration images, a bicubic interpolation with a factor of 2 was applied to the LL image during the process of windowing the inputs. Bicubic interpolation is a method of image resizing that preserves the image's fine details and sharpness, and prevents the generation of artifacts that could distort the image. By using this method, the LL subband image can be accurately resized to match the size of the input migration images, which is essential for the neural

network to learn from all available input channels.

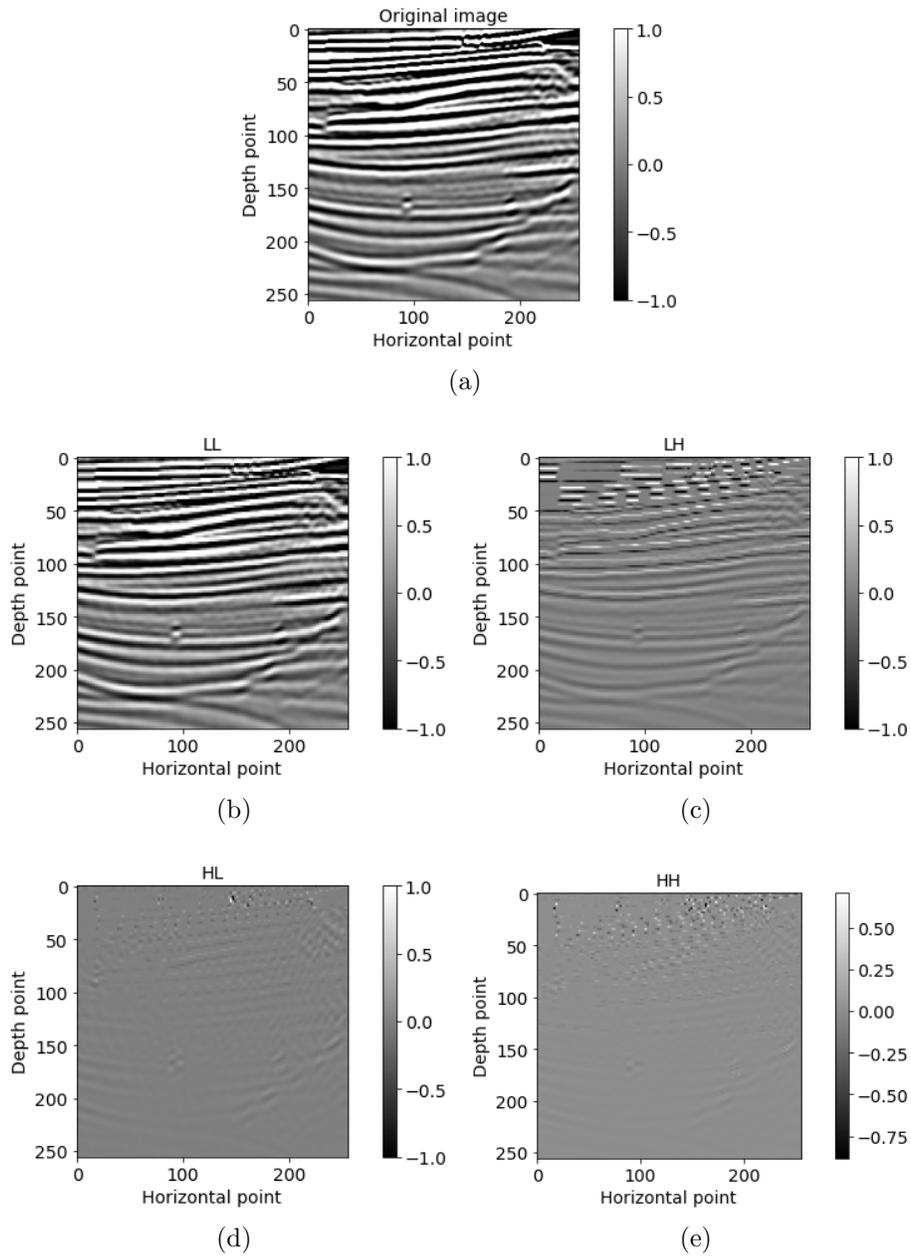


Figure 4.4: Original image (a) was decomposed by Haar discrete wavelet transform and result (b) LL (c) LH (d) HL (e) HH were obtained by filter banks with interpolation.

## Foothills

The results of the proposed approach applied to the Foothills geology model are presented in this subsection. Figure 4.5 shows the total results of the Foothills model. Compared with the RTM image (Figure 4.5c), the RTMM image (Figure 4.5d) contains more reflections from deep subsurface structures below 4000 meters depth. Even though model R1 output has significant amplitudes for shallow structure, the model R3 result (Figure 4.5f), which includes multiple energy channels and the LL subband, has improved resolution and accurate reflector prediction, for shallow and deep structures, compared to those of R1 (Figure 4.5e) or R4 (Figure 4.5g). Model R4 is used to fine-tune the optimized preconditioned parameters for R3. Table 4.1 shows that the PSNR value of the model R3 prediction for the Foothills model is the highest, at 23.26 dB, compared to the predictions obtained by models R1 and R4. These results demonstrate the effectiveness of the proposed technique in improving the resolution and accuracy of reflector predictions.

Details of the results are shown in Figure 4.6 and 4.7. Two of three input channels are listed on the first row, including smooth background reflectivity, RTM image and RTMM image. In this chapter, the background reflectivity is smoothed with a Gaussian smoother with a radius of 9 points on the horizontal and depth axes. The second row shows the true label, DWT input subband LL, and LL with multiples, arranged from left to right. LL without multiples is the input of model R1, and LL with multiples is fed into R3. As the subband LL is extracted from the initial migration images, it preserves the low-frequency geological features.

The predictions generated from the model R4 in both examples (Figure 4.6g and 4.7g) provide a pre-conditioning effect that is similar to the input smoothed reflectivity, but cannot give a refined resolution. Model R1 results (Figure 4.6h and 4.7h) have a lower resolution than the true label and fail to capture the fine structures. In contrast, model R3, which contains multiple reflections and DWT subband LL inputs, can generate enhanced predictions (Figure 4.6i and 4.7i) and suppress artifacts properly. For example, lateral velocity variation and

Table 1: PSNR (dB) comparison for Foothills example			
Prediction	Model R1	<b>Model R3</b>	Model R4
Total Foothills	22.11	<b>23.26</b>	18.09
Example 1	16.80	<b>17.23</b>	13.78
Example 2	17.90	<b>19.28</b>	13.98

Table 4.1: PSNR (dB) comparison for Foothills example.

fault details above 1000 meters depth are recovered properly in model R3 prediction. Folds between 1000 and 2000 meters depth recovered with improved amplitude and resolution in model R3 output. These observations prove that model R3, fine-tuned from model R4, keeps the power to extract key features from noisy data, and also is adapted to some smoothed input. Additionally, compared with model R1 predictions, which also use an LL input channel, R3 gives better resolution and illumination on structures and faults because of having useful multiple reflections. The PSNR values in table 4.1 also prove R3 has the highest value and performance than other neural network models.

### Overthrust

In the case of the Overthrust example, figures 4.8-4.10 indicate total and windowed predicted results by models R1, R3 and R4. Similar to the previous example result, The results show that the use of RTM with multiple reflections in Figure 4.8d provides clearer subsurface layers when compared to Figure 4.8c, which does not contain surface multiples. This is particularly evident in the layers above 1500 meters and below 4000 meters, which have more significant amplitudes in Figure 4.8d, obtained by multiple reflections.

It is worth noting that the PSNR value of the model R3 result is 18.84 dB, which is the highest value compared with the values obtained by model R1, which is 17.20 dB, and model R4, which is 15.16 dB. This indicates that the proposed technique of using the DWT subband LL as an input channel along with the smoothed background reflectivity and initial migration images can enhance the accuracy and resolution of the predicted reflectivity in

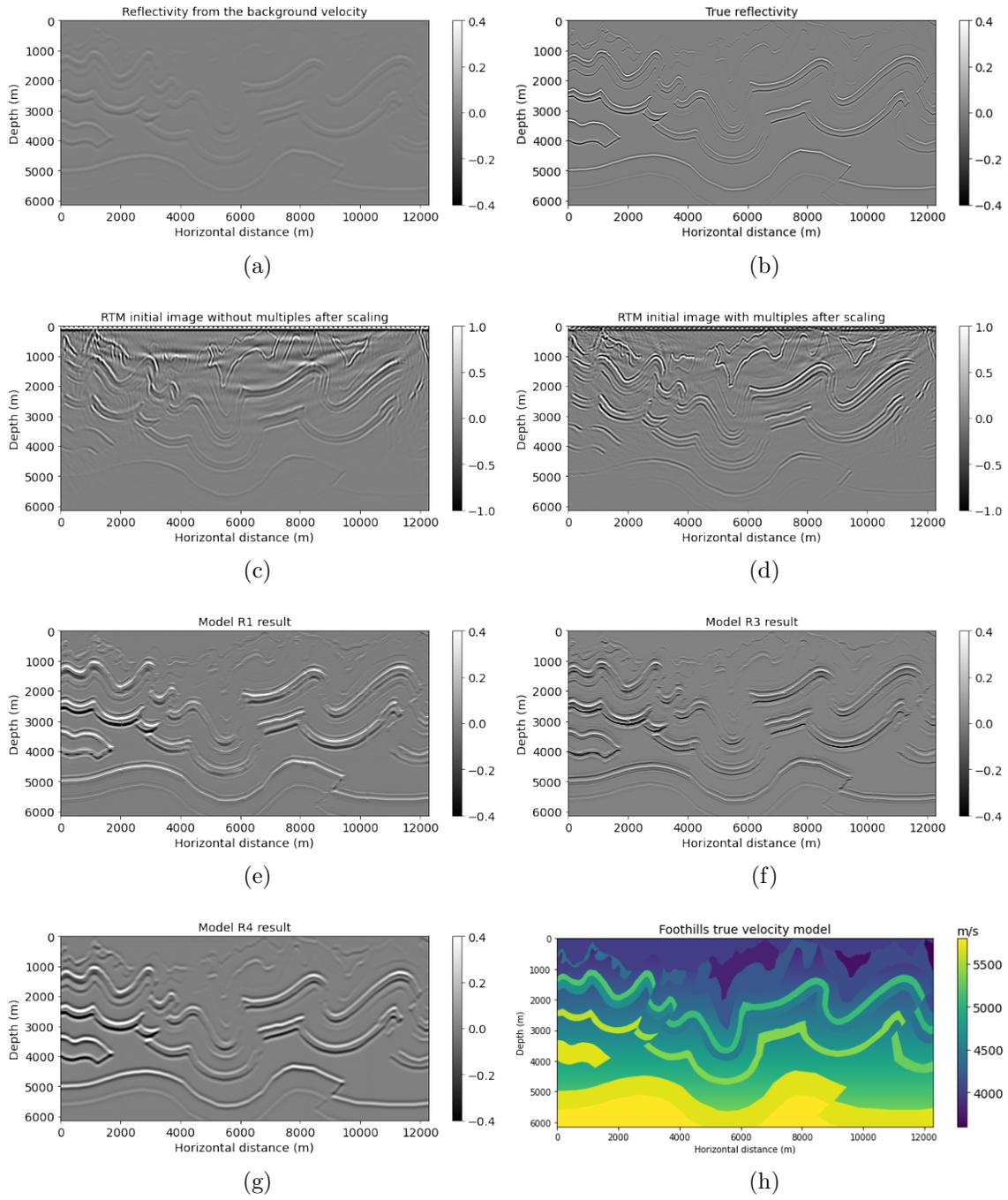


Figure 4.5: Foothills model results. (a) Reflectivity from the background velocity, (b) true band-limited reflectivity, (c) RTM image without multiple energy, (d) RTM image with multiple energy, (e) model R1 result based on workflow 1, (f) model R3 result based on workflow 3, (g) model R4 result based on workflow 4, and (h) true Foothills velocity.

Prediction	Model R1	<b>Model R3</b>	Model R4
Total Overthrust	17.21	<b>18.82</b>	15.06
Example 1	13.18	<b>13.23</b>	9.84
Example 2	12.67	<b>12.96</b>	9.84

Table 4.2: PSNR (dB) comparison for Overthrust example.

the Overthrust model as well.

The windowed results for the Overthrust example are shown in figures 4.9 and 4.10, after combining the DWT subband LL, model R3 with multiples can predict reflectivity with high resolution and accuracy. For example, Model R3 successfully recovers a small fault situated between 2500 and 3000 meters depth (Figure 4.9i), while the same location is predicted as a curvature in model R1’s output (Figure 4.9h). Also, model R3 predicts events with improved resolution compared with smoothed reflectivity input. Precisely, the horizontal layers below 3500 meters in depth. The subtle layers are blurry in the model R1 result, whereas they are distinct in the model R3 result.

The other example shows that the thin layer structure in Figure 4.10 above 2700 meters can be predicted with better accuracy by model R3 (Figure 4.10f), and the amplitude near the small thrust fault at the bottom has clear differences to indicate. It is important for accurate interpretation of subsurface features.

In figures 4.11 and 4.12, I show cross-plots for the Foothills and Overthrust models. Red scatter points with an orange line represent model R3 prediction, and blue diamond points with a black line indicate model R1 output. Even though the difference is slight, the slope of model R3 predictions is closer to 1 than model R1, which means the model R3 predictions correlate with the true band-limited reflectivity more than model R1 predictions.

Figure 4.13 illustrates the network train and validation loss comparison between RTM-CNN, model R1, and RTMM-CNN, model R3, with seventy iterations when the starting learning rate is 0.001. Model R3 (red line) can converge to a smaller loss value, e.g. 0.0004,

than the baseline model R1 (blue dashed line) for a smoothed input, which means model R3 works better in both train and validation sets.

### 4.4.3 Using DWT and noise in the network input

To assess the neural network’s generalization ability and make the synthetic data more realistic, Gaussian noise with a variance of 0.2 was added to the migration images (RTM/RTMM). Noise is always present in real observations. It is worth noting that the neural networks are not trained with Gaussian noise data. Examples with Gaussian noise added will be input to the networks directly without further training or fine-tuning. The introduction of noise may affect the quality of the data, but it can also lead to more robust and accurate results due to the neural network’s ability to learn and extract meaningful features from noisy data. Therefore, this experiment aims to evaluate the impact of noise on the neural network’s performance and to determine if the models can generalize well to new and more challenging data conditions.

Figures 4.14-4.16 give the comparison of the Foothills example, and figures 4.17-4.19 show observations from the Overthrust example. After adding Gaussian noise, both examples hardly distinguish reflection events from noise. Precisely, in Figure 4.14, horizontal events at deep depth are blurred and covered by noise, and only shallow structures are distinguishable. Under this situation, model R3 can still perform good predictions on reflectivity recovery with high resolution. For example, in Figure 4.15, the DWT subband LL inputs (figures 4.15e and f) are smoother than the migration inputs (figures 4.15b and c) due to approximation. Model R3 prediction (Figure 4.15i) has greatly-resolved reflection events. Small curvatures between 500 and 1000 meters in depth can be predicted properly.

In the other windowed example depicted in Figure 4.16, the subsurface structure predicted by model R3 (Figure 4.16i) exhibits more enhanced features, particularly the two small faults on the top, which are more distinguishable when compared to the model R1 result (Figure 4.16h). Moreover, the layers can be located in their correct positions with high resolution.

Prediction	Model R1	<b>Model R3</b>	Model R4
Total Foothills	22.17	<b>23.11</b>	18.13
Example 1	16.84	<b>16.97</b>	13.85
Example 2	17.81	<b>19.04</b>	14.03

Table 4.3: PSNR (dB) comparison for Foothills example with noise added.

Prediction	Model R1	<b>Model R3</b>	Model R4
Total Overthrust	17.20	<b>18.84</b>	15.16
Example 1	13.18	<b>13.25</b>	10.34
Example 2	12.72	<b>12.91</b>	10.34

Table 4.4: PSNR (dB) comparison for Overthrust example with noise added.

Notably, the PSNR of model R3 in this windowed example also attains the highest value, 19.04 dB, outperforming other outputs.

In the Overthrust example, since migration with multiples can provide useful information about reflection events, RTMM input (For example, Figure 4.19c) has higher signal amplitude than RTM input (Figure 4.19b). Even though the LL subband provides low-frequency information, the model R3 prediction can still extract main events from highly-resolved migrated images. Similarly, for the rest of the windowed examples, the predictions by model R3 of Overthrust are consistent with the noise-free results, which have clearer predictions with boosted resolution and accuracy than the model R1 predictions. These results demonstrate the efficacy of the proposed neural network in accurately predicting the subsurface structure and its ability to generalize to realistic data with Gaussian noise. The ability of model R3 to capture subtle structural features, coupled with its superior performance metrics, highlights its potential in subsurface imaging.

## 4.5 Conclusions

Reflectivity prediction by RTMM-CNN can be enhanced by using a DWT subband channel. With the help of the pre-trained models, this extra channel can aid in fine-tuning the network to extract important features from the low-frequency information in the LL channel. The energy from multiple reflections provides additional subsurface illumination, helping the neural network distinguish signals from noise. The neural network operator approximates the inverse of the Hessian, which can suppress image artifacts and improve the reflectivity resolution. The next step is to let the model learn how to predict a geology model with rapid lateral velocity change and test it in the field data. In future research, an extension such as Curvelet transform could be considered to effectively represent images across different scales and orientations.

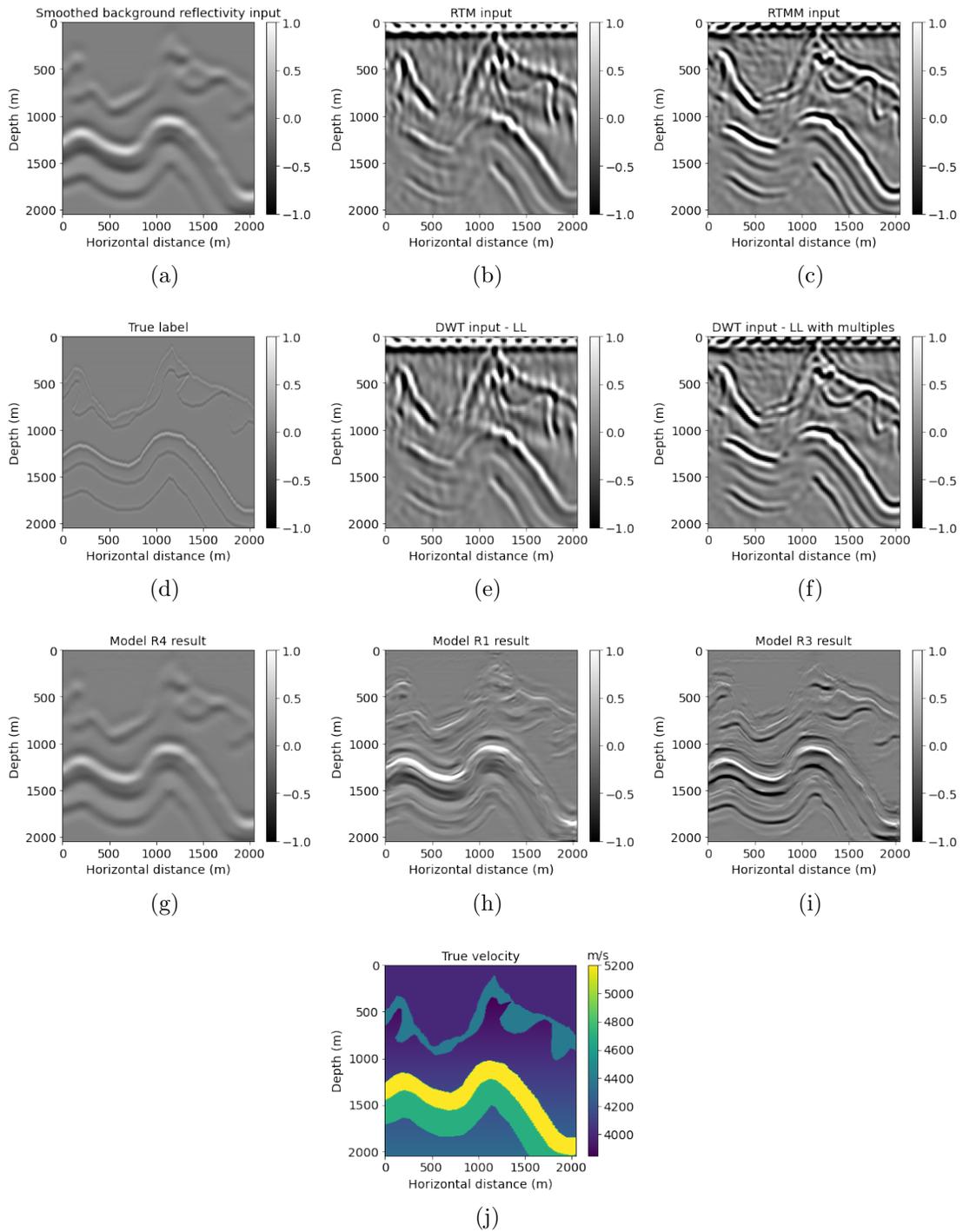


Figure 4.6: Foothills example 1 results. (a) Reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) true label, (e) DWT subband LL without multiple energy, (f) DWT subband LL with multiple energy, (g) model R4 result based on workflow 4, (h) model R1 result based on workflow 1, (i) model R3 result based on workflow 3, and (j) true windowed velocity.

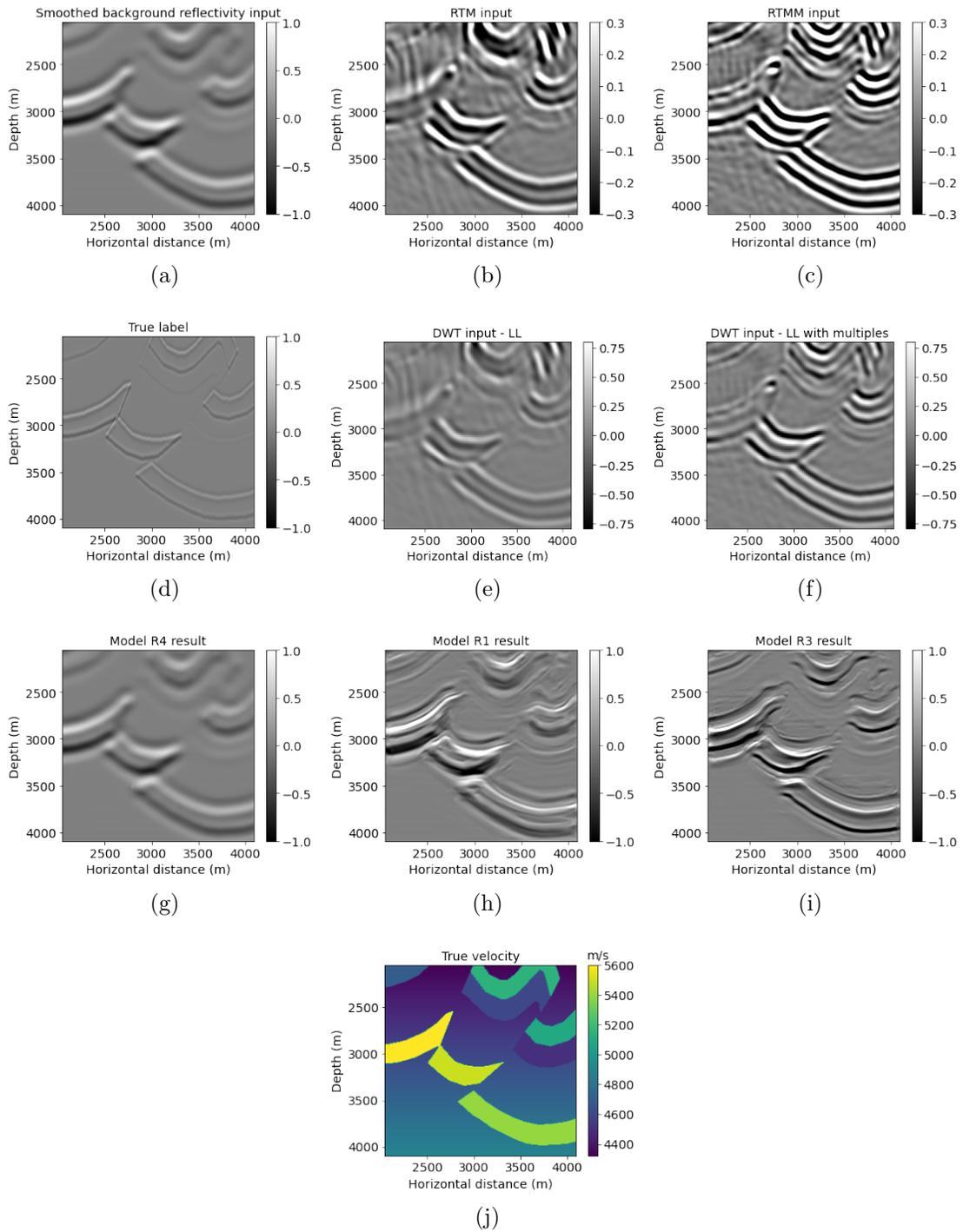


Figure 4.7: Foothills example 2 results. (a) Reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) true label, (e) DWT subband LL without multiple energy, (f) DWT subband LL with multiple energy, (g) model R4 result based on workflow 4, (h) model R1 result based on workflow 1, (i) model R3 result based on workflow 3, and (j) true windowed velocity.

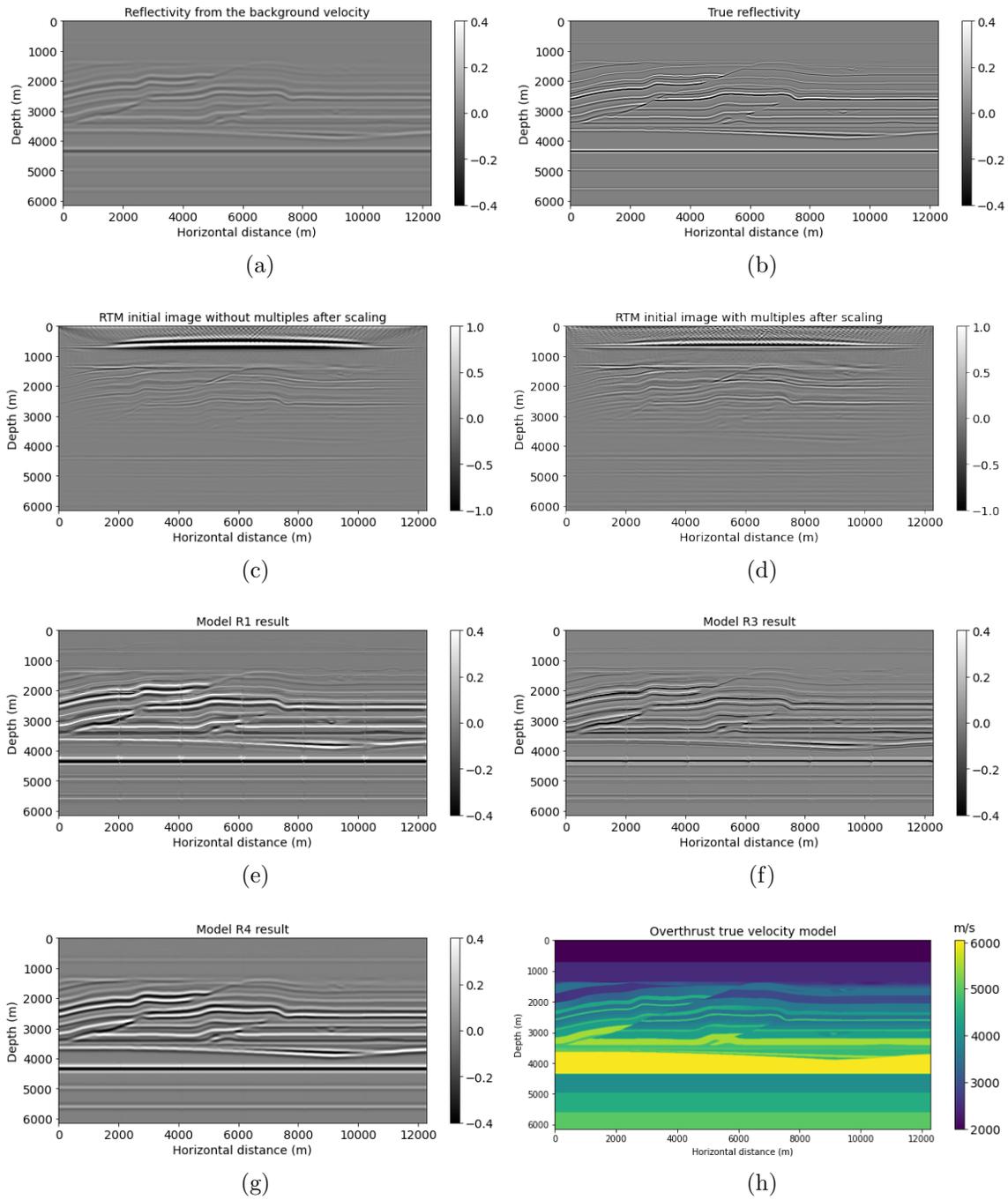


Figure 4.8: Overthrust model results. (a) Reflectivity from the background velocity, (b) true band-limited reflectivity, (c) RTM image without multiple energy, (d) RTM image with multiple energy, (e) model R1 result based on workflow 1, (f) model R3 result based on workflow 3, (g) model R4 result based on workflow 4, and (h) true Overthrust velocity.

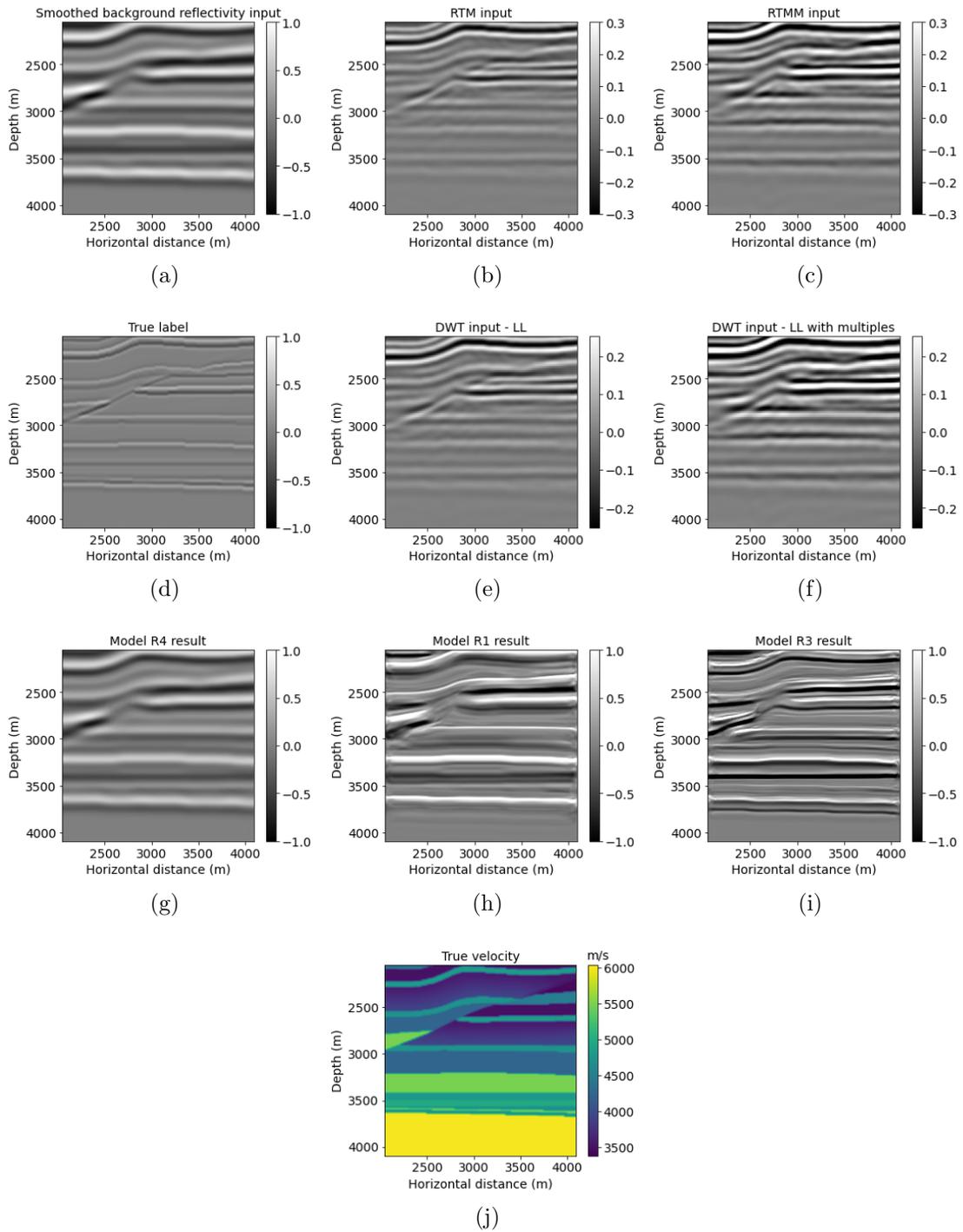


Figure 4.9: Overthrust example 1 results. (a) Reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) true label, (e) DWT subband LL without multiple energy, (f) DWT subband LL with multiple energy, (g) model R4 result based on workflow 4, (h) model R1 result based on workflow 1, (i) model R3 result based on workflow 3, and (j) true windowed velocity.

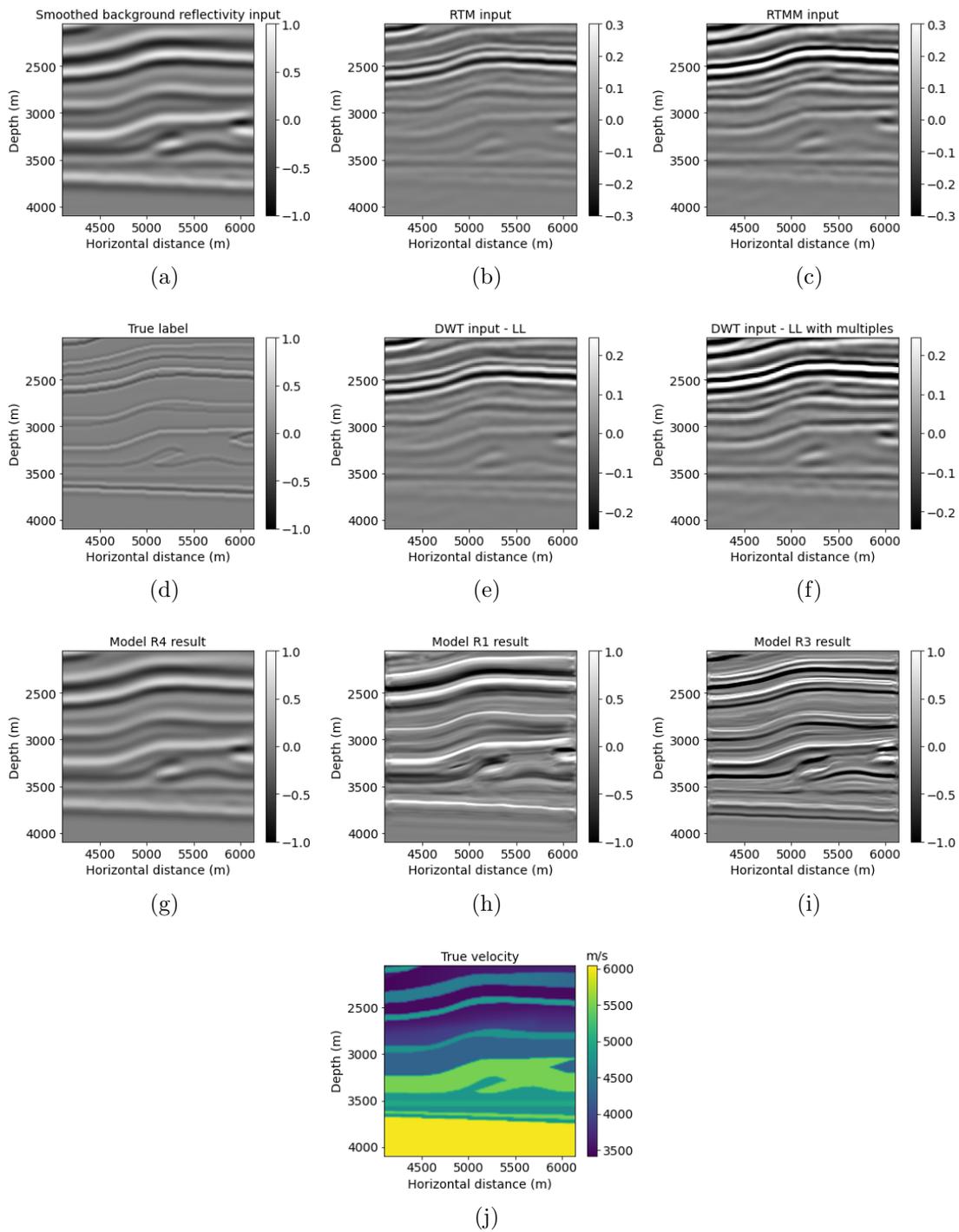


Figure 4.10: Overthrust example 2 results. (a) Reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) true label, (e) DWT subband LL without multiple energy, (f) DWT subband LL with multiple energy, (g) model R4 result based on workflow 4, (h) model R1 result based on workflow 1, (i) model R3 result based on workflow 3, and (j) true windowed velocity.

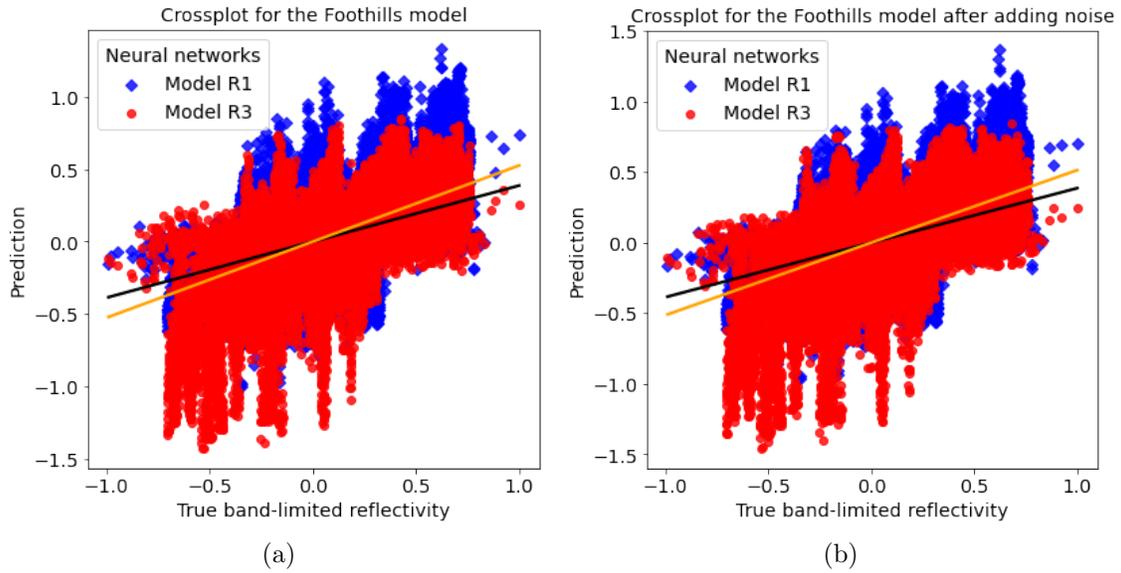


Figure 4.11: Cross-plot comparison between results from models R1 and R3 for the Foothills model.

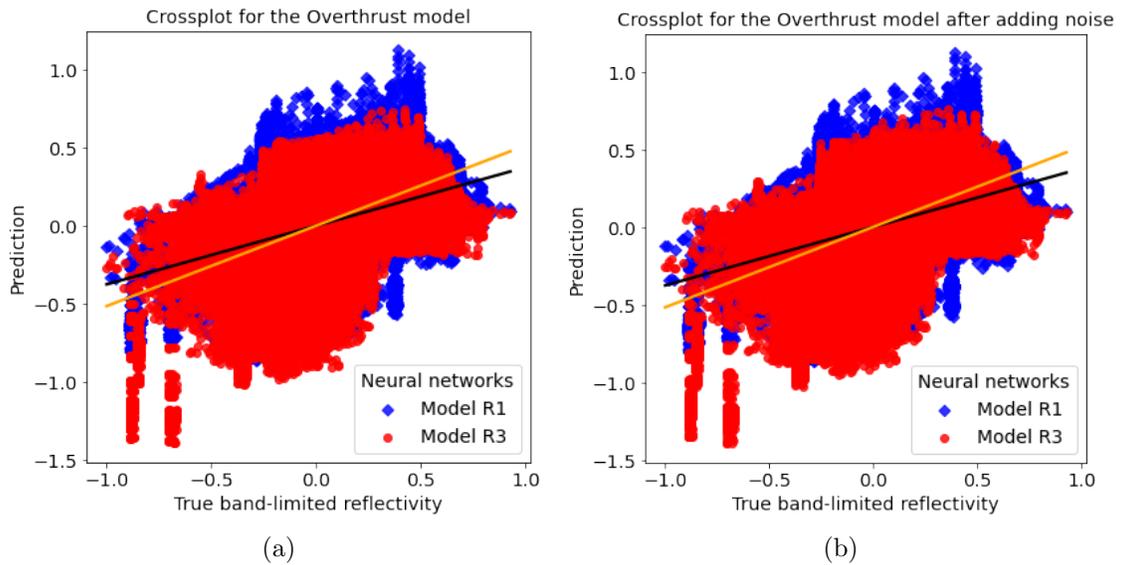


Figure 4.12: Cross-plot comparison between results from models R1 and R3 for the Overthrust model.

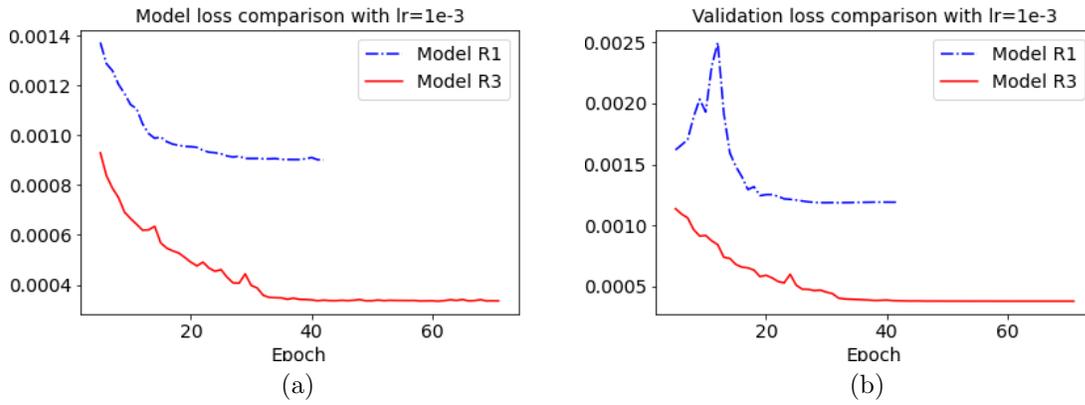


Figure 4.13: Training and validation loss comparison between models R1 and R3.

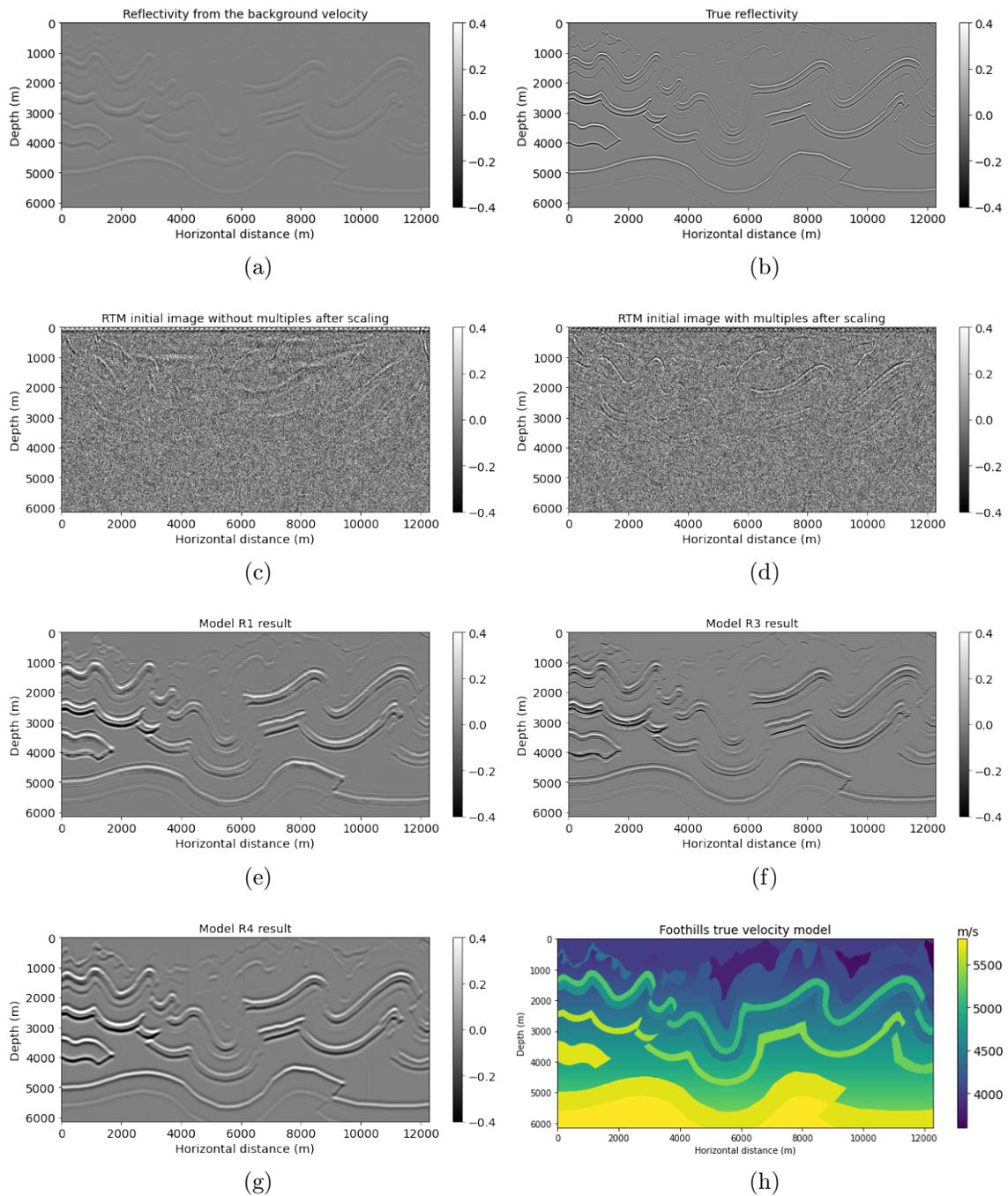


Figure 4.14: Foothills model results after adding noise. (a) Reflectivity from the background velocity, (b) true band-limited reflectivity, (c) RTM image without multiple energy, (d) RTM image with multiple energy, (e) model R1 result based on workflow 1, (f) model R3 result based on workflow 3, (g) model R4 result based on workflow 4, and (h) true Foothills velocity.

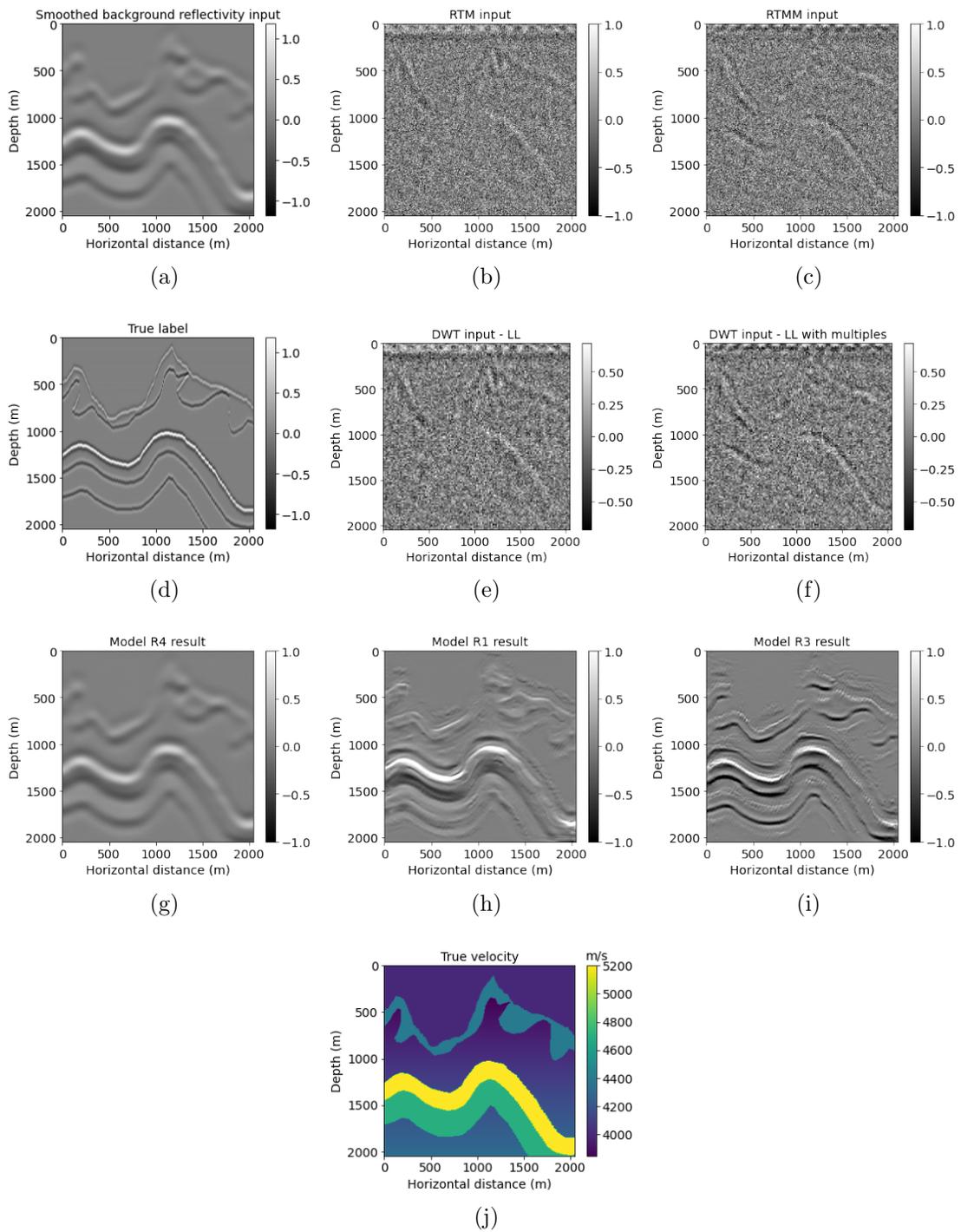


Figure 4.15: Foothills example 1 results after adding noise. (a) Reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) true label, (e) DWT subband LL without multiple energy, (f) DWT subband LL with multiple energy, (g) model R4 result based on workflow 4, (h) model R1 result based on workflow 1, (i) model R3 result based on workflow 3, and (j) true windowed velocity.

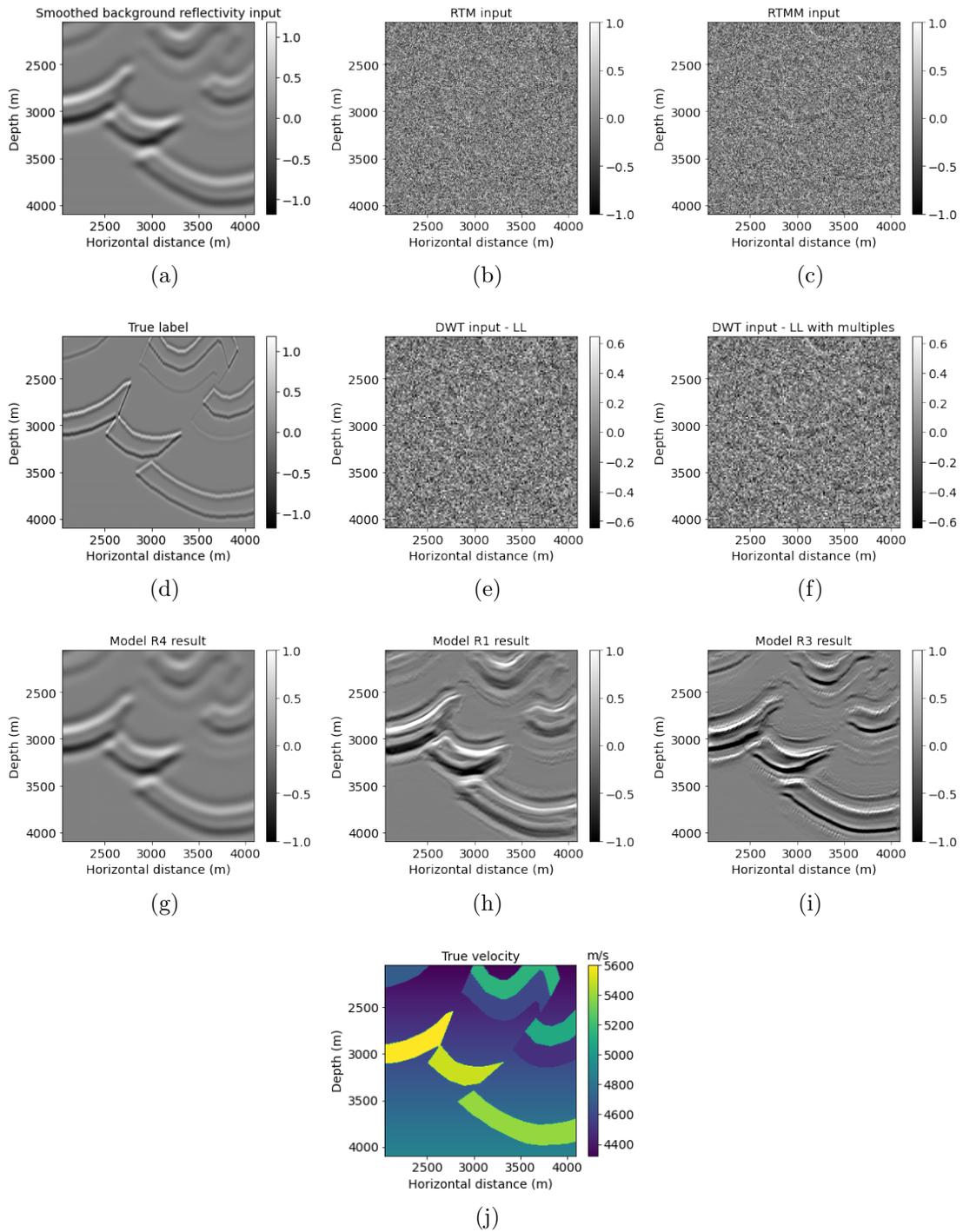


Figure 4.16: Foothills example 2 results after adding noise. (a) Reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) true label, (e) DWT subband LL without multiple energy, (f) DWT subband LL with multiple energy, (g) model R4 result based on workflow 4, (h) model R1 result based on workflow 1, (i) model R3 result based on workflow 3, and (j) true windowed velocity.

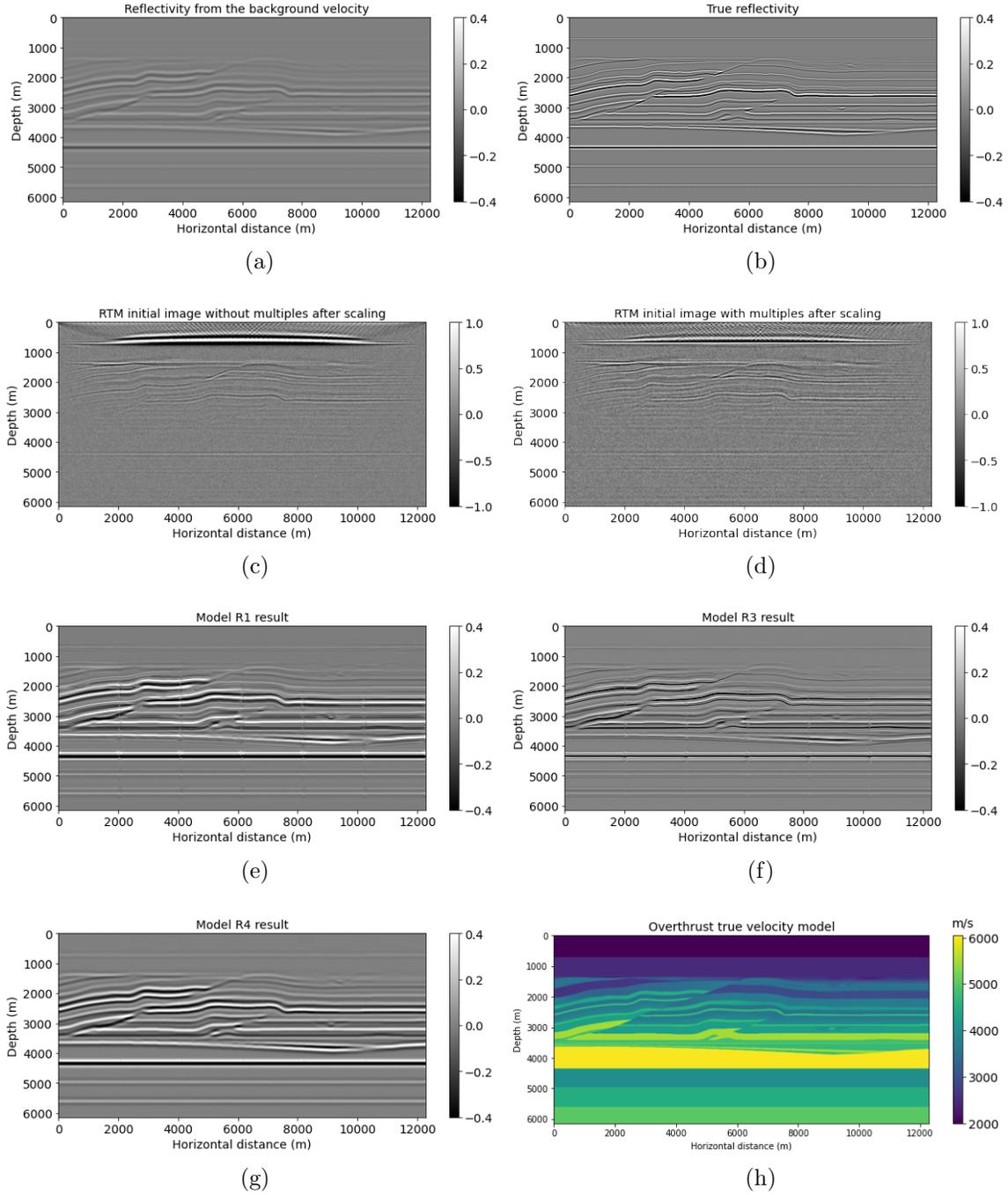


Figure 4.17: Overthrust model results after adding noise. (a) Reflectivity from the background velocity, (b) true band-limited reflectivity, (c) RTM image without multiple energy, (d) RTM image with multiple energy, (e) model R1 result based on workflow 1, (f) model R3 result based on workflow 3, (g) model R4 result based on workflow 4, and (h) true Overthrust velocity.

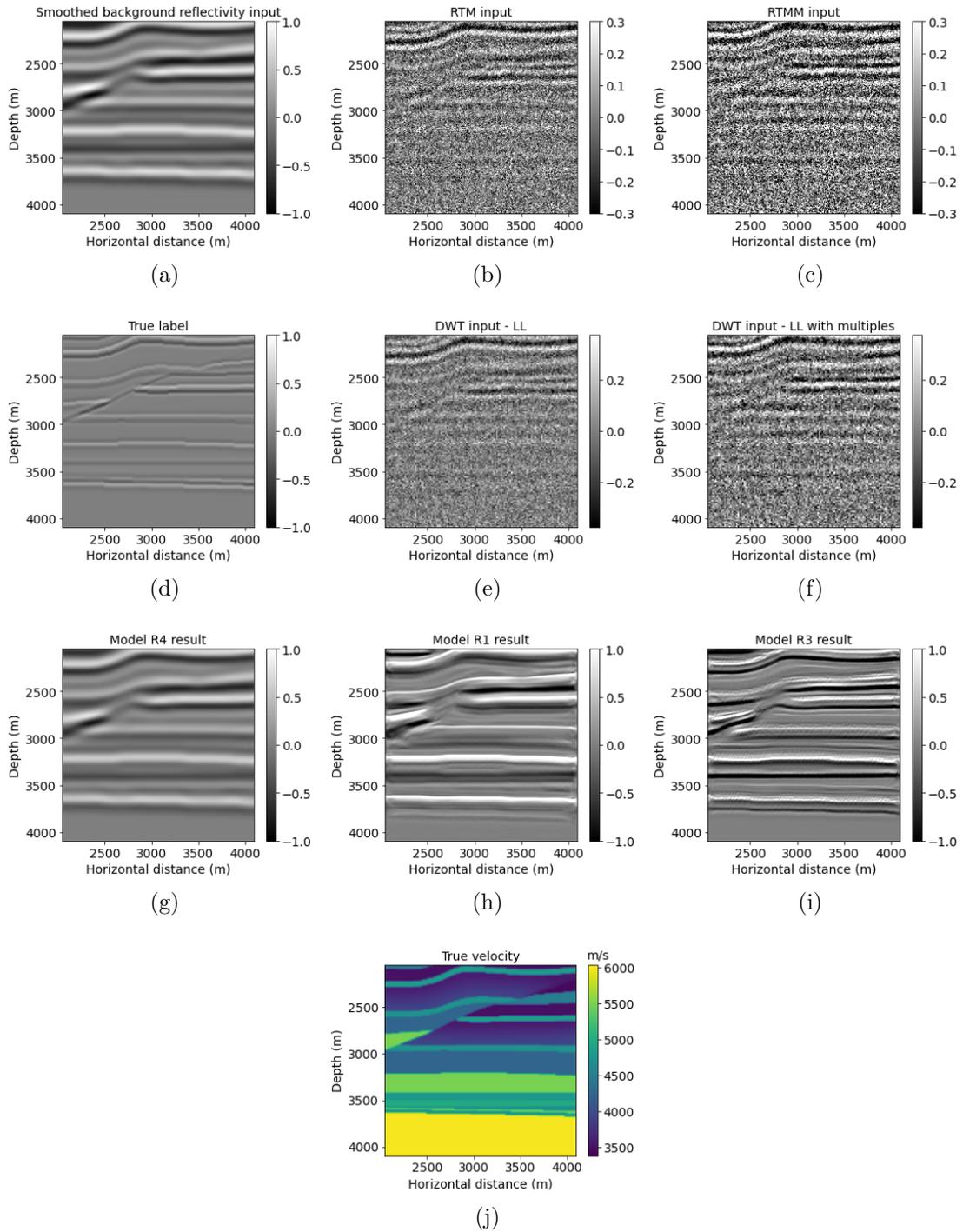


Figure 4.18: Overthrust example 1 results after adding noise. (a) Reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) true label, (e) DWT subband LL without multiple energy, (f) DWT subband LL with multiple energy, (g) model R4 result based on workflow 4, (h) model R1 result based on workflow 1, (i) model R3 result based on workflow 3, and (j) true windowed velocity.

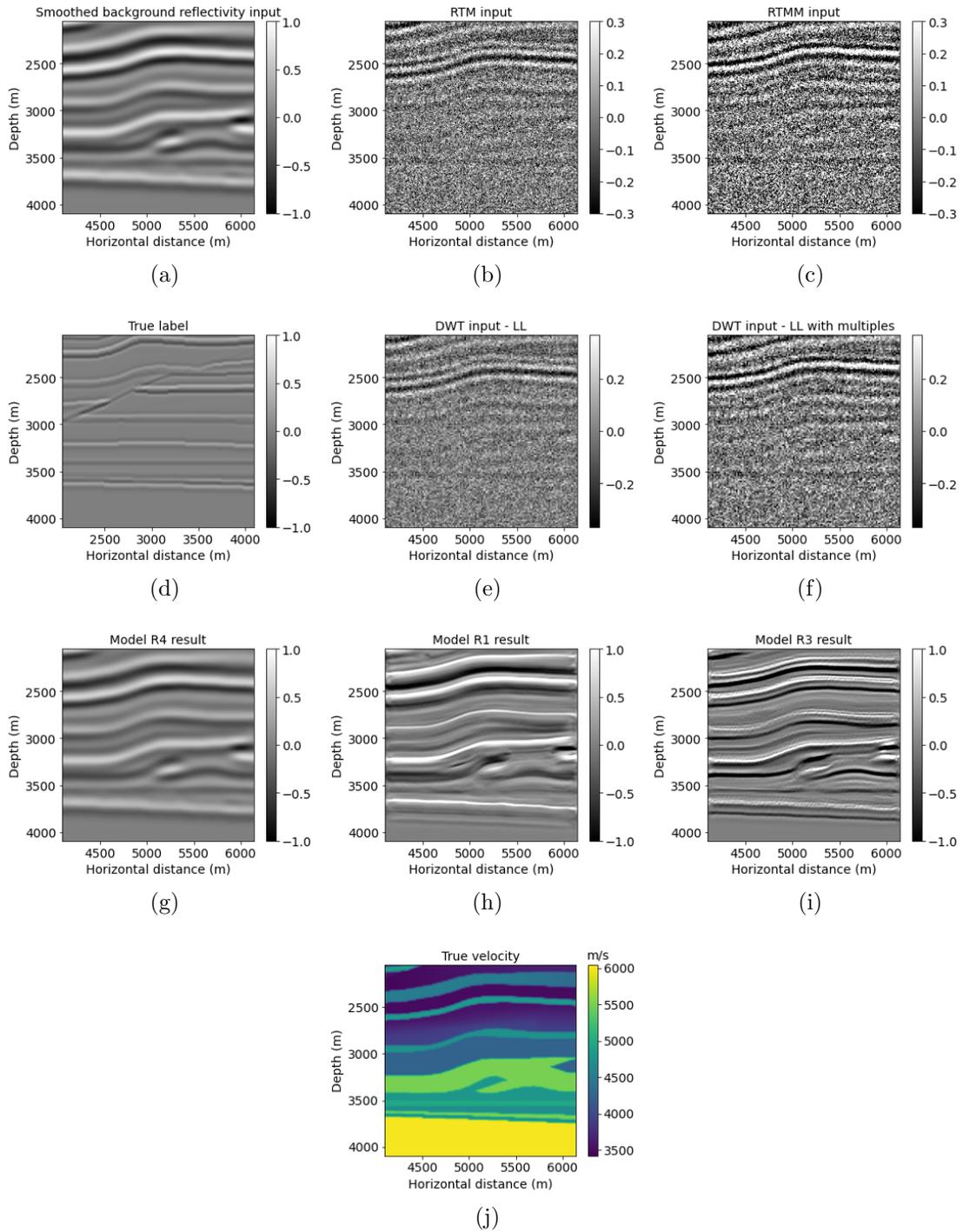


Figure 4.19: Overthrust example 2 results after adding noise. (a) Reflectivity from the background velocity, (b) true windowed band-limited reflectivity, (c) RTM image without multiple energy, (d) true label, (e) DWT subband LL without multiple energy, (f) DWT subband LL with multiple energy, (g) model R4 result based on workflow 4, (h) model R1 result based on workflow 1, (i) model R3 result based on workflow 3, and (j) true windowed velocity.

# Chapter 5

## Time-lapse data matching using neural networks with multiple reflections

### 5.1 Summary

Time-lapse seismic is a complex problem in reservoir monitoring because near-surface noise, poor subsurface illumination, and the inherent weak amplitude of reservoir changes affect the quality of the interpretation. Because of its importance, geophysicists have tried many approaches to solve these problems. In this chapter, I apply deep learning methods to address these challenges. Firstly, a stacked long short-term memory (SD-LSTM) neural network adapts the near-surface baseline data to the near-surface monitor data. This assumes that differences from the near-surface are not due to changes in the reservoir but differences in the seismic experiment (acquisition and processing). A U-Net is then followed to work on differences between monitor and baseline images to suppress noise on a large scale. Furthermore, the energy from surface multiples is added during migration and shot record generation in

the forward modelling step to increase subsurface illumination. A double-difference method is applied to the predicted and observed data to give a final difference. The results show that the SD-LSTM can anticipate and mitigate noise in the monitor data. The final difference between baseline and monitor models has suppressed significant noise after combining SD-LSTM, U-Net and surface multiples. The proposed method is also tested in a field dataset, DAS VSP data from the CaMI FRS project, with extended bidirectional SD-LSTM and convolutional neural networks (CNN). The output provides meaningful information and prediction for CO<sub>2</sub> injection migration within a target area, which matches the CO<sub>2</sub> injection plan.

## 5.2 Introduction

Time-lapse seismic monitoring, which acquires seismic data at time intervals over the same site to obtain reservoir variations, has contributed to detecting subsurface physical properties and reservoir behaviour in recent years (Wang, 1997; Koster et al., 2000; Pennington, 2000; Lumley, 2001; Arts et al., 2003; Isaac and Lawton, 2006, 2014; Chadwick et al., 2010; Wang and Morozov, 2020; Henley and Lawton, 2021). Particularly, time-lapse becomes essential in carbon capture and storage reservoirs to monitor the movement and possible leaks of CO<sub>2</sub>, which is necessary for the safety and the success of the process and tracking oil and gas displacement effects during long-term change. In another context, time-lapse is a crucial technique for enhanced oil recovery projects to improve fluid injection efficiency.

Time-lapse seismic monitoring projects face many challenges. Seismic imaging of time-lapse changes is difficult because of their weak amplitudes, often covered by noise differences due to near-surface complexities. Variations in the acquisition system and geometry create complex wavefield differences that do not cancel by subtraction. The weak effect of reservoir changes on seismic amplitudes is overwhelmed by false anomalies. Poor subsurface illumination accentuates these problems since 4D seismic captures only parts of the reservoir changes. Some geophysical approaches (Rickett and Lumley, 2001; Ayeni and Biondi,

2010; Zhang et al., 2013; Bergmann et al., 2014; Wapenaar and Van Ijsseldijk, 2021; Fu and Innanen, 2022) try to solve the challenges above. In addition to geophysical methods, deep neural networks have become useful tools to deal with these problems (Yuan et al., 2020; Zhong et al., 2020; Li et al., 2021; Hussein et al., 2021; Alali et al., 2022; Li and Alkhalifah, 2022). Long short-term memory (LSTM) and bidirectional long short-term memory (Bi-LSTM) (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005) are two typical recurrent neural networks that have been applied in many geophysical problems to learn non-linear relationships. For example, seismic data reconstruction (Yoon et al., 2020), missing well log estimation (Pham and Wu, 2019), elastic properties and litho-fluid facies estimation (Aleardi, 2022), seismic impedance inversion and parameter estimation (Calderón-Macías et al., 2000; Moya and Irikura, 2010; Alfarraj and AlRegib, 2019; Das et al., 2019; Guo et al., 2019; Roy et al., 2020). The Bi-LSTM can work with long and dense temporal traces as the LSTM does, but it can also learn from long-term forward and backward temporal dependencies from historical data. It works for complex situations, including noise overlapping with data. Thus, these networks are suitable to be used in time-lapse seismic data.

In this chapter, I propose a data-driven method to predict and mitigate non-reservoir-related changes between baseline and monitor data, using stacked LSTM and CNN-Bi-LSTM with surface multiple reflections. The networks predict non-reservoir data variations of the monitor data from the baseline data by using as training input the data acquired above the reservoir. Then, a double difference method is applied to the migrated data to generate model residuals between predictions and observations. The time-lapse model residuals are fed into a U-Net for image denoising. The U-Net, in this chapter, acts as a filter and mask to enhance image resolution and mitigate noise on a large scale. Reservoir changes become more visible after attenuating the migration artifacts. Additionally, I take advantage of multiple reflections generated from free surface boundary conditions to broaden subsurface illumination.

## 5.3 Theory

### 5.3.1 Time-lapse seismic

In time-lapse seismic, baseline data is set as the reference. After injecting CO<sub>2</sub> or other fluids into a target area, monitor data collected after a long period of time is used to indicate the seismic attribute variations. Based on Alali et al. (2022), the residual between monitor data  $\mathbf{d}_{obsm}$  and baseline data  $\mathbf{d}_{obsb}$  is

$$\delta\mathbf{d}(t) = \mathbf{d}_{obsm}(t) - \mathbf{d}_{obsb}(t) = \mathbf{n}(t) + \delta\mathbf{r}(t), \quad (5.1)$$

where  $t$  means time,  $\mathbf{n}(t)$  denotes noise generated from near-surface change and non-repeatable data due to the change of acquisition systems, which should be eliminated. The data difference term,  $\delta\mathbf{r}(t)$  denotes subsurface reservoir variations at the target layer. The aim is to determine reservoir changes from the signal, which is mixed with noise and migration artifacts. This is difficult to achieve with classical linear algorithms since artifacts and noise can have complex non-linear origins, for example, in the near-surface. Neural networks can learn non-linear patterns that characterize the noise and artifacts by using the monitor and baseline data above the targeted anomaly as training data. This network can then be used for the whole survey to predict and mitigate the noise. The input traces are baseline data, and true labels are monitor data above the reservoir change area.

Different from other research, in this chapter, two windows are constructed to predict two sets of baseline data. The first window, located at a time record of a shallow depth, matches near-surface change from baseline to estimated baseline data, given the corresponding section of the observed monitor data as labels. Even though it is called “estimated baseline data”, it should predict some near-surface noise. The other window, positioned at a time record to a greater depth, is deployed to predict another baseline data from observed baseline data. However, it is essential to note that the two sets of predicted baseline data refer precisely to the areas above the reservoir changes. The aforementioned terms individually indicate the

predictions obtained from the shallow and deep windows, which are subsequently used in a double difference method. After subtraction, the near-surface noise should be reduced, and reservoir variations can be obtained.

Figures 5.1 to 5.3 present a simplified example illustrating the rationale behind utilizing shallow and deep windows in this chapter. These figures show a trace used in the workflow and the corresponding estimated results. In Figure 5.1, the trace used in neural network training is shown in the dashed line with the first order of surface multiples added at 0.3 and 0.8 seconds. The primary reflections depicted in this figure compare different time records for primaries and multiples, aiding in determining the shallow and deep windows in the subsequent step.

Figure 5.2 provides a trace comparison between the observed baseline (red line) and monitor (dashed line) with two windows. One assumption is that the observed monitor has a 0.03-second time shift lag from baseline due to the near-surface change. Additionally, its amplitude decreases to 0.9 times that of the baseline data. A reservoir change is identified at the 1-second mark in the time records.

Two windows are defined based on the primary and surface multiple distributions shown in Figure 5.1. Firstly, a shallow window with one primary reflection is established, ranging from 0 to 0.2 seconds. Then, a deep window is defined from 0 to 0.45 seconds, involving two primary reflections and one surface multiple reflection. Both windows are positioned above the reservoir change area, which is located at 1 second. Traces selected from shallow and deep windows are separately fed into a neural network for training and testing.

The results of shallow window prediction, deep window prediction and data differences are presented in Figure 5.3a, b and c, respectively. For ease of comparison, Figure 5.3c demonstrates that, before the 0.2 seconds time record, the predicted monitor obtained from the shallow window (dashed line) can give fewer artifacts than the deep window (red line).

On the other hand, after 0.2 seconds, the deep window demonstrates its capability to leverage information from surface multiples and additional primaries, resulting in more ac-

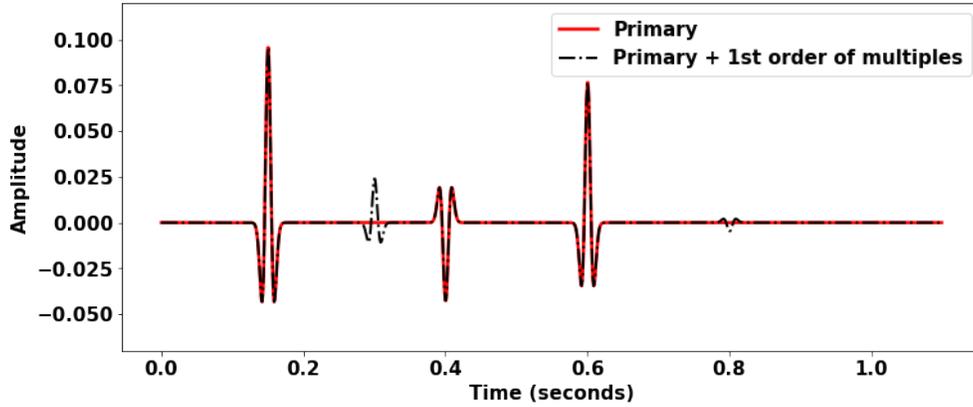


Figure 5.1: Example of a trace prepared for the neural network training. Primary reflections (red line) from three subsurface reflectors are observed separately at 0.15, 0.4 and 0.6 seconds. The first order of surface multiple reflections from the first and second reflectors are acquired at 0.3 and 0.8 seconds, shown in the dashed line.

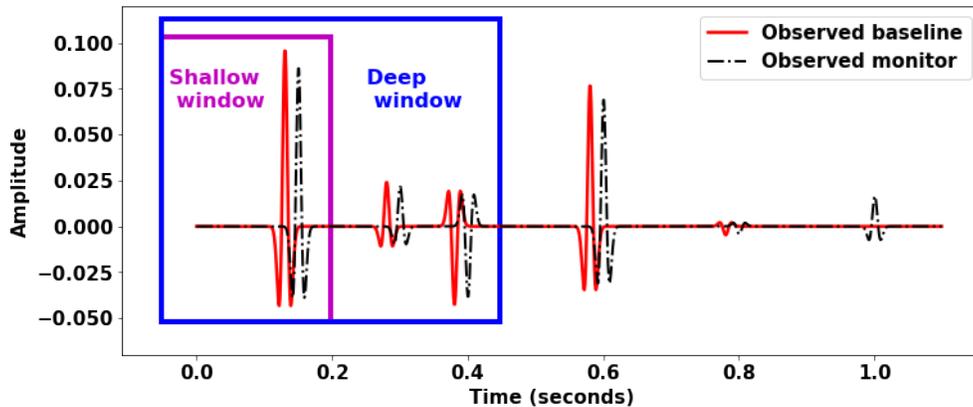
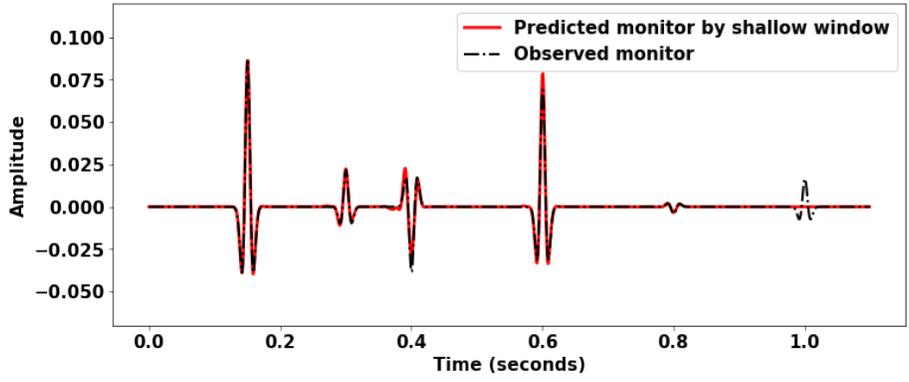


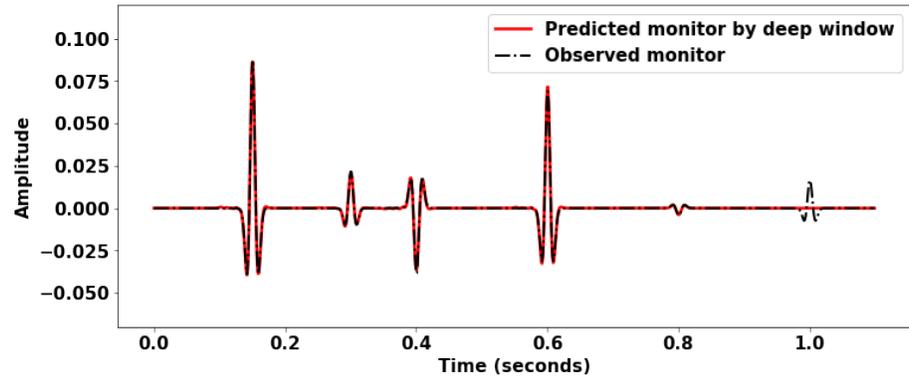
Figure 5.2: Example of two windowed trace inputs for the neural network training. A shallow window (purple box) contains a primary reflection, and a deep window (blue box) consists of primaries and the first order of surface multiple reflections from the first reflector.

curate trace prediction than the shallow window. From this observation, the shallow window prediction effectively handles and mitigates the impact of near-surface change, and the deep window prediction can recover the amplitude of the observed monitor data from the baseline. This rationale underlies the selection of two windows for predicting the two baseline data sets. A detailed algorithm design will be shown in the workflow section.

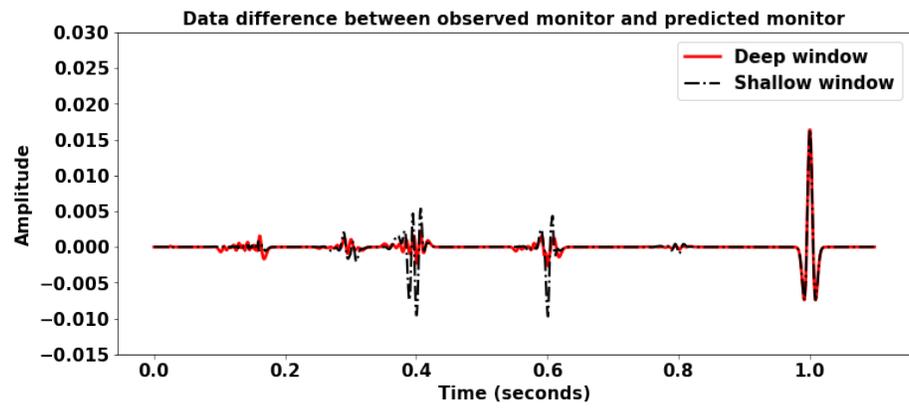
In the following subsections, I introduce some fundamental knowledge about LSTM,



(a)



(b)



(c)

Figure 5.3: Prediction of the trace example. Comparisons between the predicted monitor (red line) by (a) shallow window and (b) deep window, and the observed monitor (dashed line).

Bi-LSTM and stacked neural networks.

### 5.3.2 Recurrent neural network (RNN) and LSTM

Next, I will set the deep learning background used in this chapter. I will start from a simple LSTM framework, and then expand to a Bi-LSTM and stacked neural networks. A recurrent neural network (RNN) is an artificial neural network that uses sequential or time series data. RNN (Jordan, 1986; Rumelhart et al., 1985) is derived from a feed-forward neural network where the connections between nodes do not form a cycle and deliver information in one direction (Figure 5.4). On the other hand, RNN has an internal self-looped deep-learning architecture. It allows previous output to affect subsequent input and output. In other words, the current input will learn from and depend on the past sequence output. After obtaining the current output, it will be sent back into the recurrent network. The benefit of RNN is that it can process variable-length input sequences, for example, time sequences. While training an RNN model, vanishing or exploding gradient issues might occur. If the gradient is too small or large, it tends to grow or vanish when it is passed back through many time steps.

Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) was designed with special memory cells to store temporal information. It can remember values over arbitrary time intervals with gate structures shown in Figure 5.5. Also, it can avoid the vanishing gradient problem usually occurring in RNN. The gradient in LSTM contains the gate's vector of activations, allowing the network to control the gradient values better and avoid getting too small or large. This structure will enable LSTM to remember long-range features better than conventional recurrent neural networks. This capability makes LSTMs good candidates to work with seismic traces since usually, we are interested in capturing long-term dependencies along the time direction introduced by physical features across the ray paths.

Within each LSTM cell (shown in Figure 5.5), there are four gates in total:  $\mathbf{f}_t$ ,  $\mathbf{i}_t$ ,  $\mathbf{g}_t$  and  $\mathbf{o}_t$ , which are the *forget gate*, *input gate*, *candidate gate* and *output gate* cell activation vectors, respectively. They regulate the flow of information in and out of the LSTM cell.

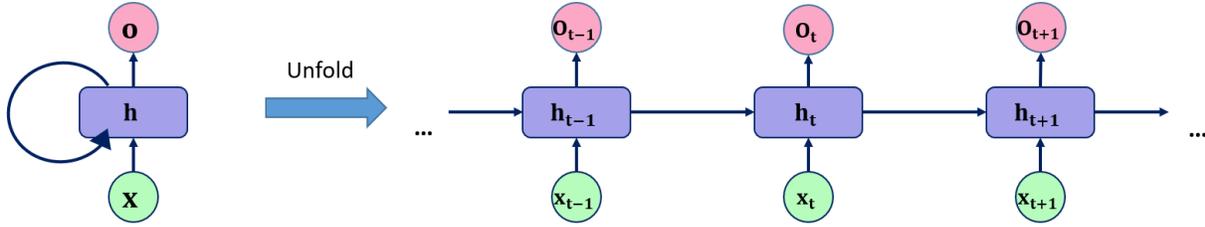


Figure 5.4: Sketch for the recurrent neural network.

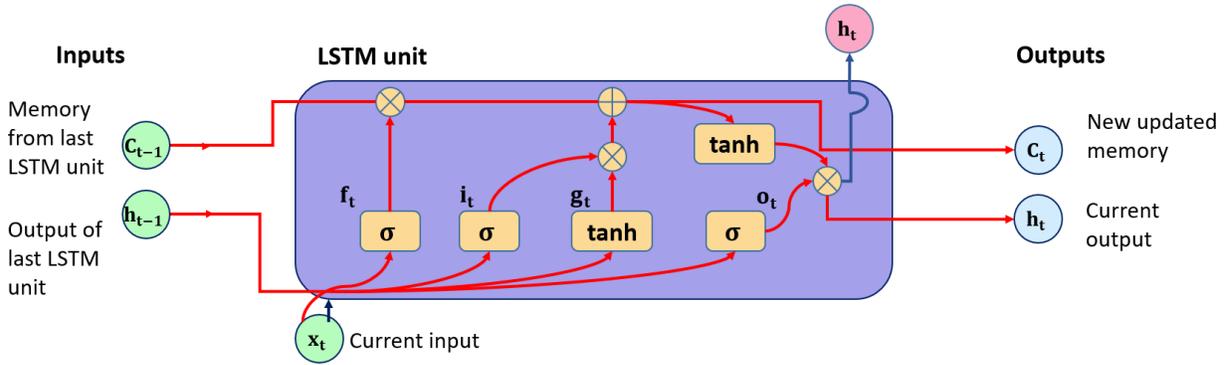


Figure 5.5: LSTM unit distribution.

They have the same size as the hidden vector  $\mathbf{h}_t$ . Next, each gate vector will be illustrated in detail. The forget gate  $\mathbf{f}_t$  is determined by

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f), \quad (5.2)$$

where  $\mathbf{h}_{t-1}$  is the hidden layer vector from the previous time step, and  $\mathbf{x}_t$  means the current step input vector.  $\sigma$ ,  $\mathbf{W}_f$  and  $\mathbf{b}_f$  represent the logistic sigmoid activation function, weight matrices and bias for the forget gate.

Then, the input gate  $\mathbf{i}_t$  follows a similar behaviour as the forget gate but with different weight and bias:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i). \quad (5.3)$$

Another candidate gate  $\mathbf{g}_t$  can be determined with a *tanh* activation layer:

$$\mathbf{g}_t = \tanh(\mathbf{W}_g[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_g). \quad (5.4)$$

The last gate in the LSTM unit is output gate  $\mathbf{o}_t$ , and it can be obtained by:

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o). \quad (5.5)$$

After having all the gates, the next step is to calculate and determine a new updated memory  $\mathbf{C}_t$  and current output  $\mathbf{h}_t$ . The former needs memory from the last LSTM unit  $\mathbf{C}_{t-1}$  combined with forget gate  $\mathbf{f}_t$  and the product of input gate  $\mathbf{i}_t$  and candidate gate  $\mathbf{g}_t$ . Then, the current cell state  $\mathbf{C}_t$  can be calculated by

$$\mathbf{C}_t = \mathbf{i}_t \mathbf{g}_t + \mathbf{f}_t \mathbf{C}_{t-1}. \quad (5.6)$$

As for the output of the LSTM cell  $\mathbf{h}_t$ , it is made of the product of output gate  $\mathbf{o}_t$  and current cell state  $\mathbf{C}_t$  after applying *tanh* activation layer:

$$\mathbf{h}_t = \mathbf{o}_t \tanh(\mathbf{C}_t). \quad (5.7)$$

### 5.3.3 Bi-LSTM

Unlike conventional RNNs, bidirectional RNNs (BRNNs) can deal with sequential data in both directions, forward and backward, by using two separate hidden layers (Schuster and Paliwal, 1997; Graves et al., 2013b). Based on BRNNs and LSTM, the bidirectional long short-term memory (Bi-LSTM) network (Graves and Schmidhuber, 2005) was developed to capture long sequences in reverse and forward directions. One hidden layer processes the input sequence in the forward direction. The other hidden layer handles the input in the reverse direction. It includes doing backward passes for output neurons, forward and

backward states, and updating error functions and weights. Thus, the output of the current time step is obtained from both layers' hidden vectors.

The forward function of Bi-LSTM with inputs of  $M$  units and  $N$  hidden units is shown below:

$$\mathbf{h}_t^n = \sum_{m=1}^{M_I} x_t^m w_{mn} + \sum_{n'=1, t>0}^{N_H} \alpha_{t-1}^{n'} w_{n'n}, \quad (5.8)$$

$$\alpha_t^n = \Theta_n(h_t^n), \quad (5.9)$$

where  $\mathbf{h}_t^n$  is the output of the current time step at each unit  $n$ , and  $x_t$  denotes the sequence input.  $w_{mn}$  represents the weight of the input  $m$  to hidden unit  $n$ , and  $w_{n'n}$  means the weight of hidden unit  $n$  towards hidden unit  $n'$ .  $\Theta_n$  means the activation function of the hidden unit  $n$ . The activation function of hidden unit  $n$  at time step  $t$  is given by  $\alpha_t^n$ .

The backward calculation is

$$\frac{\delta O}{\delta w_{nk}} = \sum_{t=1}^T \frac{\delta O}{\delta h_t^n} \alpha_t^n, \quad (5.10)$$

$$\frac{\delta O}{\delta \alpha_t^n} = \Theta'_n(h_t^n) \left( \sum_{k=1}^K \frac{\delta O}{\delta h_t^n} w_{nk} + \sum_{n'=1, t>0}^H \frac{\delta O}{\delta h_{t+1}^{n'}} w_{nn'} \right), \quad (5.11)$$

where  $O$  denotes an objective function with unit  $K$  output.

For the bidirectional LSTM, we need to consider the forward and backward flows in two separate layers (Du et al., 2020). The final output can be obtained by

$$h_t = \gamma h_t^f + \beta h_t^b, \quad (5.12)$$

$$y_t = \sigma(h_t), \quad (5.13)$$

where  $h_t^f$  is the forward LSTM layer output which takes time sequences from  $x_1$  to  $x_T$ ,  $h_t^b$  denotes the backward LSTM layer output which takes the reverse time sequences from

$x_T$  to  $x_1$ .  $\gamma$  and  $\beta$  represent the importance of forward LSTM and backward LSTM, and satisfy  $\gamma + \beta = 1$ .  $h_t$  is the sum of two LSTM outputs, and  $y_t$  is the prediction after using an activation function. In this chapter, the prediction at this stage involves separately estimating the baseline and monitor data from two distinct windows. For the output dense layer, a linear activation function is selected, considering that seismic traces can contain negative values.

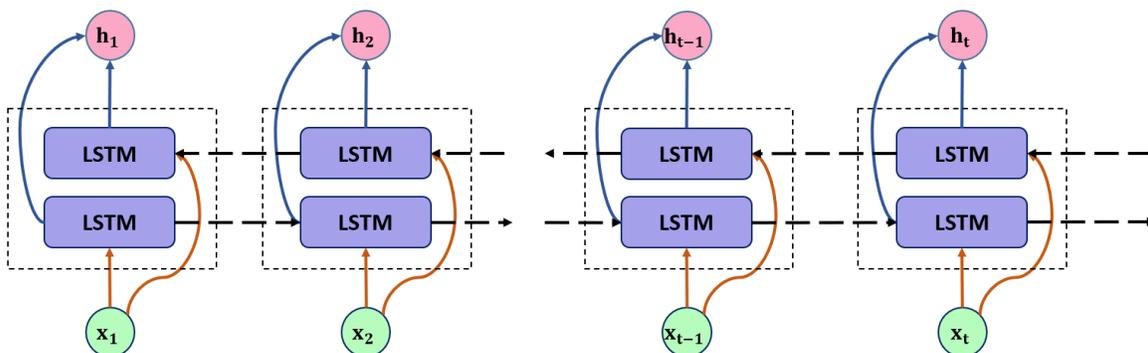


Figure 5.6: Workflow for Bi-LSTM algorithm.

### 5.3.4 Stacked LSTM and Bi-LSTM

Based on the research from Graves et al. (2013a), and Cui et al. (2018), a deep LSTM can be generated by stacking several LSTM hidden layers on top of each other. This scheme also works for the Bi-LSTM networks. The output of one Bi-LSTM hidden layer will be fed into the subsequent Bi-LSTM hidden layer as the input. Figure 5.7 shows the stacked Bi-LSTM mechanism. The input of every hidden layer should consist of both the forward and backward layers at the level below. Stacked Bi-LSTM can detect and build up an effectively high level of sequential data representations.

The subsequent section will present a detailed description of the proposed method's workflow and parameter configuration. This will provide a comprehensive understanding of the methodology. Additionally, the parameter settings, such as learning rate, batch size, hidden layer size and the number of epochs, will be discussed to ensure reproducibility and

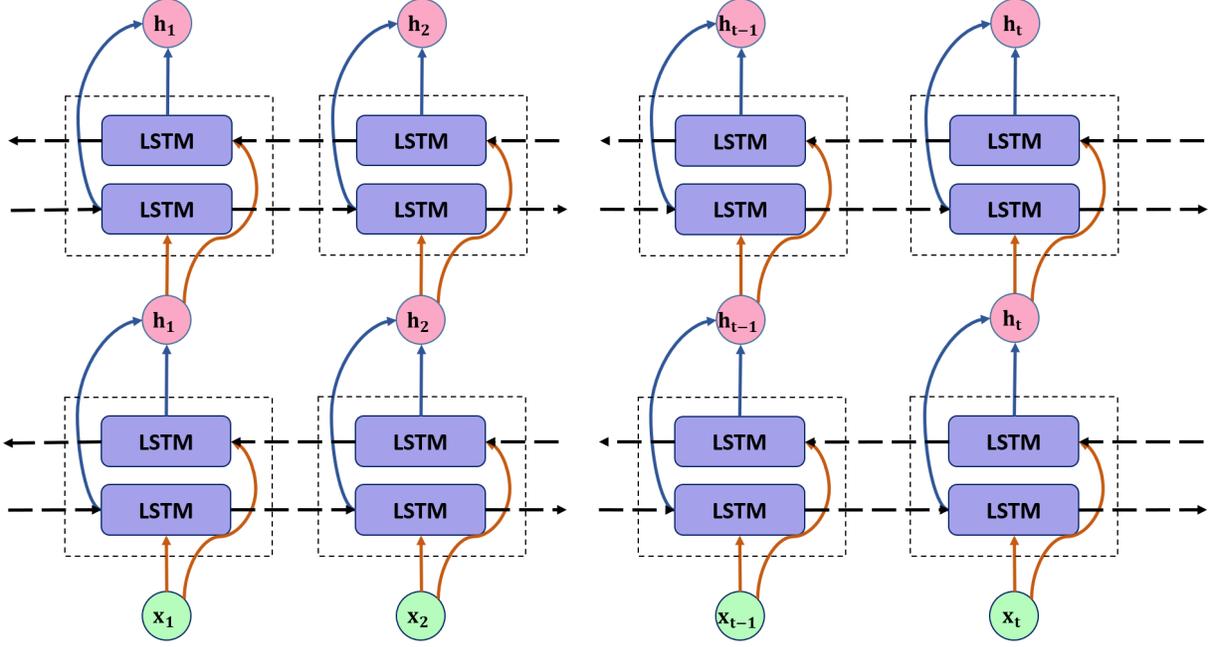


Figure 5.7: Stacked Bi-LSTM.

optimize the method's performance.

## 5.4 Workflow

The workflow presented in this chapter, depicted in Figure 5.8, consists of four distinct steps that delineate implementing the proposed method. In the first step, I extend the idea of Alali et al. (2022), training stacked LSTM neural networks using different trace parts of observed baseline data  $\mathbf{d}_{obsb}$  by windowing: the shallow part of traces is used for predicting baseline data  $\mathbf{d}_{calcb_s}$ , and the deep part of traces above the reservoir change is trained to determine another predicted baseline data  $\mathbf{d}_{calcb_d}$  but with multiples included. During the training, different parts of baseline data are the input and the corresponding parts of monitor data  $\mathbf{d}_{obsm}$  are set as labels. The reason for training the network with data above the reservoir is that time-lapse requires minimizing the differences in acquisition and other near-surface variations while keeping the differences produced by fluid changes in the reservoir. Training on the shallow part can compensate only for variations in near-surface

velocity. Without these corrections, the time shift in the deeper part will contain a mix of both effects. In summary, neural networks shift the predicted baseline close to the observed monitor data without affecting the anomalies sought to detect. To compare the effectiveness of the proposed method against the traditional approach, I apply a matching filter with a filter size of 21 temporal steps to the baseline shot records. An analysis of the results is provided in the subsequent sections.

After obtaining the two sets of predicted baseline data, the next step is calculating the migration difference by an elastic reverse time migration with surface-related multiples added. Then, A double-difference method, calculated from the difference between two residuals of observed and predicted migration images, is applied to determine the time-lapse reservoir changes. The double-difference concept originated from tomography and inversion (Watanabe et al., 2004; Asnaashari et al., 2015; Zhou et al., 2010). To address the limitations of the direct data difference approach ( $\mathbf{d}_{obsm} - \mathbf{d}_{obsb}$ ), I propose an extension of the data difference formulation with two additional sets of predicted baseline data. This modification is necessary as the direct data difference method yields unsatisfactory results. To improve the accuracy of the subtraction process and mitigate the influence of near-surface noise, I introduce a more complex technique. So, the difference between the two sets of observation and prediction data is:

$$\delta \mathbf{d} = (\mathbf{d}_{obsm} - \mathbf{d}_{obsb}) - (\mathbf{d}_{calcb_s} - \mathbf{d}_{calcb_d}), \quad (5.14)$$

where  $\mathbf{d}_{obsm}$  and  $\mathbf{d}_{obsb}$  denote observed monitor and baseline data, respectively, and  $\mathbf{d}_{calcb_s}$  and  $\mathbf{d}_{calcb_d}$  represent predicted data by using shallow and deep windows separately. The predicted baseline using a shallow window is subtracted from the direct data difference to reduce the near-surface noise. Additionally, to account for the contribution of multiple reflections, the predicted baseline obtained using a deep window is added to the resulting difference. This addition helps to incorporate the valuable information provided by multiple

reflections. Then, the time-lapse model changes  $\delta\mathbf{m}_{time-lapse}$  can be derived:

$$\delta\mathbf{m}_{time-lapse} = \delta\mathbf{m}_{obs} - \delta\mathbf{m}_{calc}, \quad (5.15)$$

where  $\delta\mathbf{m}_{obs}$  and  $\delta\mathbf{m}_{calc}$  are calculated from the first and second parentheses in equation 5.14, respectively.

After applying reverse time migration and double-difference on migrated images, the final step is using a U-Net (Ronneberger et al., 2015) to denoise predicted reflectivity model residuals. This U-Net architecture was first mentioned in Chapter 3 with additional skip connection layers. This chapter uses it to learn patterns from time-lapse reflectivity differences and mitigate artifacts in the migrated images. In this chapter, the input channel contains the differences in reflectivity between observed and predicted migrated images. The additional skip connection layer works as an identity mapping, to transfer information and patterns from the input. It is important to note that, for the real field data example with DAS, stacked data is utilized instead of migration images. This decision is made due to the absence of information regarding the imaging velocity. Therefore, stacked data is employed as a suitable alternative for the purpose of the analysis. Additionally, I use stacked Bi-LSTM with convolutional neural network blocks in the DAS example because bidirectional neural networks can handle better the complexity of DAS data for the near-surface part.

## 5.5 Neural network settings

The main assumption in this chapter is that monitor and baseline acquisition systems are the same but with near-surface velocity changes in the monitor data. Five geology models are used for training and testing, including a horizontal-layered model, a curve-fault model, the Marmousi model, the Overthrust model, and a field DAS example. A slight reservoir anomaly is located at the deep part of each model to generate weak amplitudes of data difference. The train and test rates are set to 0.8: 0.2.

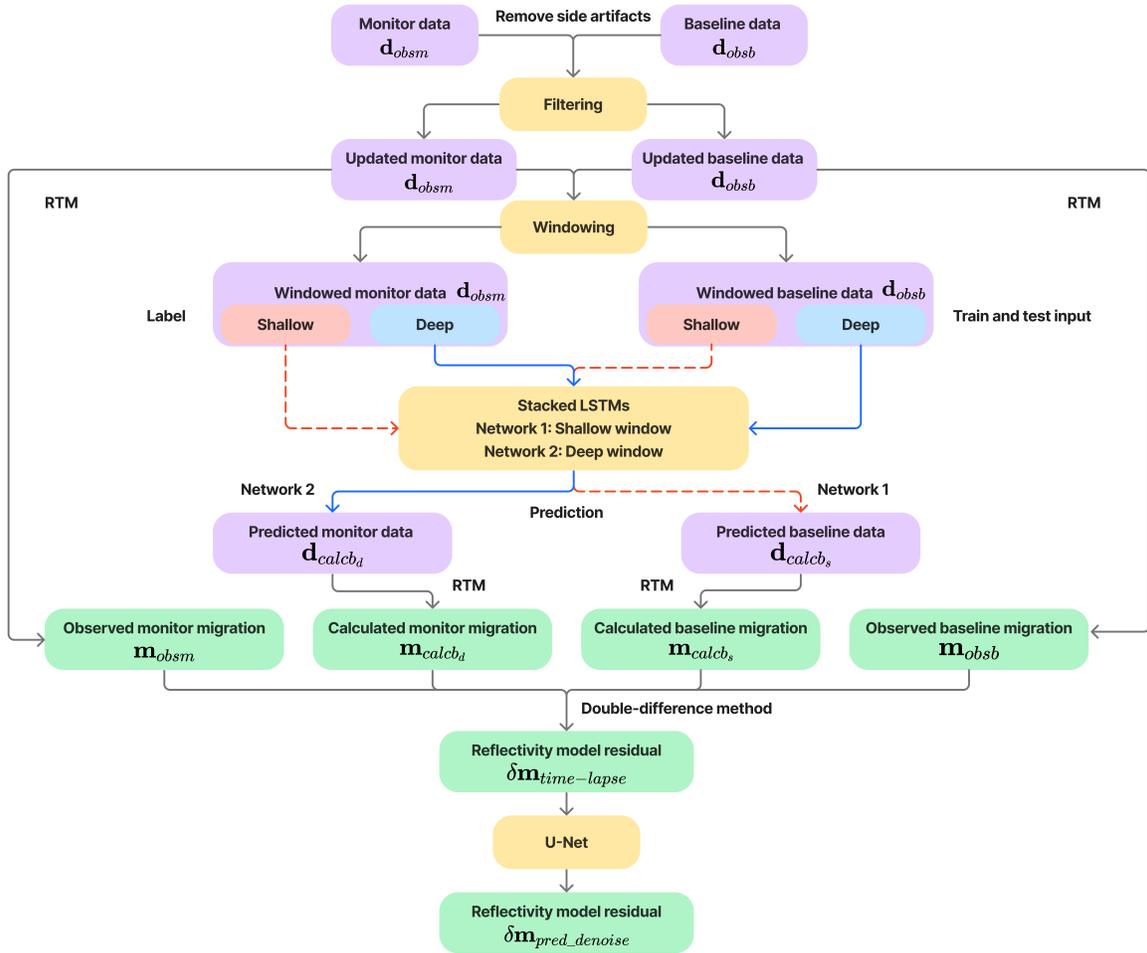


Figure 5.8: Workflow for predicting time-lapse reflectivity change.

A stacked LSTM neural network is used to predict data anomaly above reservoir changes for the first four synthetic examples. The hyperparameters for the first stacked LSTM are listed as follows. The batch size is 128, the number of epochs is 50, the learning rate starts from 0.01, and the hidden layer size is 100. Two layers of LSTM are applied with a 0.1% dropout. The final output activation function is a *tanh* function.

As for the DAS VSP data, a complex CNN-Bi-LSTM is applied to handle noisy data for extracting and learning key features of signals from noise. The 1D CNN-Bi-LSTM has a more profound architecture compared with the two-layer stacked LSTM mentioned above. Figure 5.9 shows the neural network architecture's hyperparameters in each step box. It combines

two 1D CNN blocks with stacked Bi-LSTM layers to suppress significant noise through training. Convolutional layers can extract amplitude information from the seismic traces and their output is fed into the stacked Bi-LSTM. After that, an LSTM layer with long units is set to strengthen the training process and provide stabilization. ReLu activation functions are used in the ConvNet and LSTM. Lastly, A dense layer with a linear activation function outputs the final prediction. The learning rate also starts from 0.01 with a “patience” hyperparameter of 15 iterations. The number of epochs is 100, with a batch size of 512.

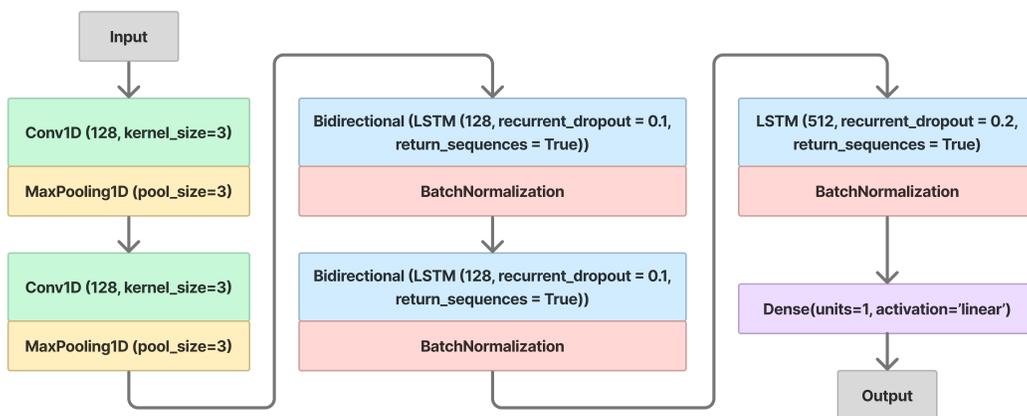


Figure 5.9: Stacked Bi-LSTM with convolutional layers architecture.

In the synthetic examples in this chapter, I also use a U-Net to suppress the noise in the predicted model differences. The total number of input windows varies depending on the model’s size, which is extracted partly from the predicted reflectivity residual. The true (known) reflectivity difference is set as the label. The train and test rates are also set to 0.8: 0.2. After trials with different window sizes, empirically, I decided to use  $64 \times 64$  points, which seems a proper size to capture features in the reservoir anomaly. Learning rates for this U-Net start from 0.001 with “patience” of 20 iterations. There are 200 epochs in total, with a batch size of 64. The root-mean-squared error (RMSE) and peak-signal-to-noise ratio (PSNR), introduced in Chapter 3, are applied to measure network performance. Similarly to the mean square error, the root mean square error (RMSE) penalizes large errors when

given a large-scale data set and evaluates the quality of predictions. It is the square root of the average of the squared differences between the estimated and the actual value of the variable/feature.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\mathbf{d}_{calc}^i - \mathbf{d}_{obs}^i)^2}, \quad (5.16)$$

where  $n$  is the total number of samples,  $\mathbf{d}_{calc}$  are the predicted shot records, and  $\mathbf{d}_{obs}$  represents the observed data.

## 5.6 Numerical example

In this section, I show five examples to illustrate the generalization of the proposed workflow for different subsurface structures with weak reservoir anomalies. The first four are synthetic models with the same spatial interval, time sampling rate and total record time. The models have near-surface velocity changes from the baseline data to the monitoring data, which could come, in a real case, from changes in the acquisition geometry (not in this case) or changes in the near-surface conditions, like for example, in the water table level or freezing conditions. This setting is designed to test the attenuation of near-surface differences that account for most non-reservoir-related changes between baseline and monitor surveys. The proposed stacked LSTM can learn minor changes and time shifts from the perturbations and predict baseline data close to monitor data at shallow depth. The fourth example shows the versatility of the proposed method by showcasing its ability to generalize to similar models or situations using pre-trained features. This highlights the robustness and adaptability of the approach, as it can effectively estimate CO<sub>2</sub> migration paths for data collected from other parts of the same survey. For the last example, a field DAS dataset, the monitor data has a denser acquisition system than the baseline data. Based on the data above, the proposed method using CNN-Bi-LSTM neural network gives reliable predictions on the shape of the CO<sub>2</sub> injection area.

### 5.6.1 Horizontal-layered model

The horizontal-layered model contains several flat layers with a thrust anomaly at around 1800 meters. The total size of the model is  $281 \times 251$  gridpoints with 10 meters spatial interval (Figure 5.10). I simulated 50 shots with 126 receivers each at 10 meters in depth. The recorded time is 3 seconds with a 0.0015 seconds time sampling rate. At the uppermost portion of the monitor velocity model, two layers are introduced to simulate near-surface variations, with velocity values ranging from 1480 m/s to 1500 m/s, whereas, at the same location, the baseline model has a velocity of 1500 m/s. This near-surface modification in the monitor survey simulates the weather layer and soil change on the near-surface.

In Figure 5.11, I show the first four shots for the observed baseline (Figure 5.11a), the observed monitor (Figure 5.11b), the predicted baseline by a shallow window (Figure 5.11c) and the predicted baseline by a deep window (Figure 5.11d), respectively. According to the workflow in the previous section, a shallow part of observed baseline traces  $d_{calcb_s}$ , set before 0.975 seconds, is used to predict the calculated baseline. The other calculated baseline  $d_{calcb_a}$  is predicted from a deeper part of observed baseline traces, between 0.525 and 1.425 seconds. Both the shallow or deep parts of the traces are above the reservoir anomaly, which occurs after 1.6 seconds in Figure 5.11.

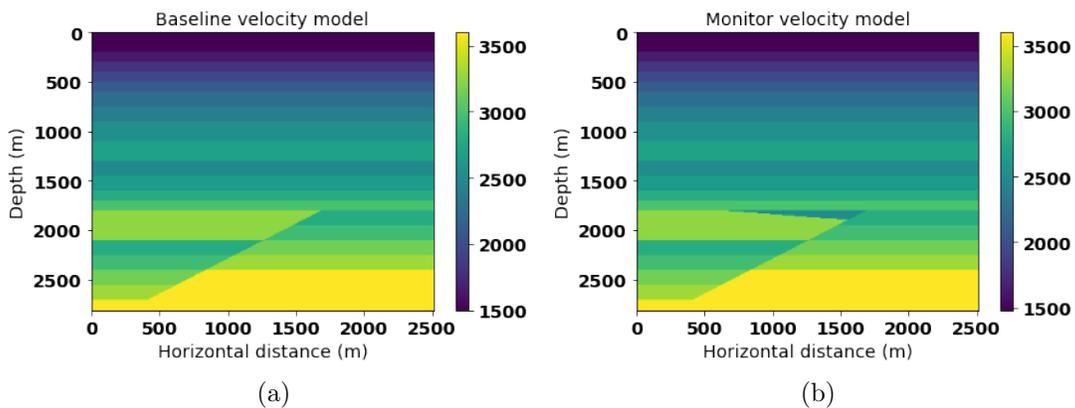
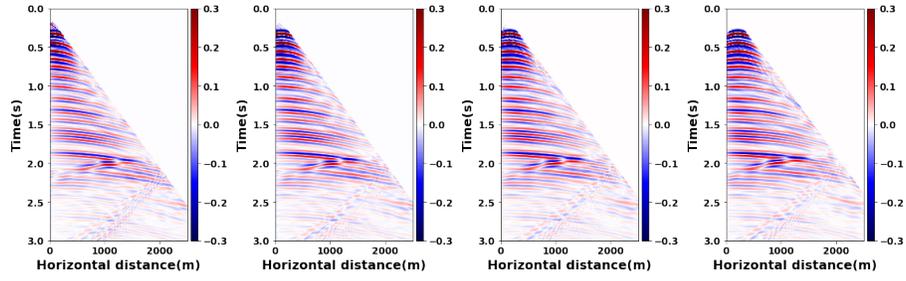
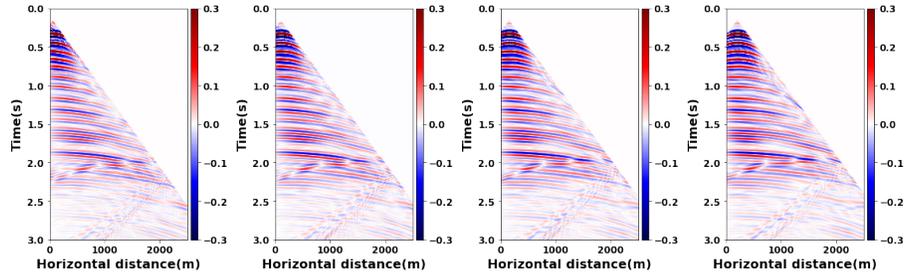


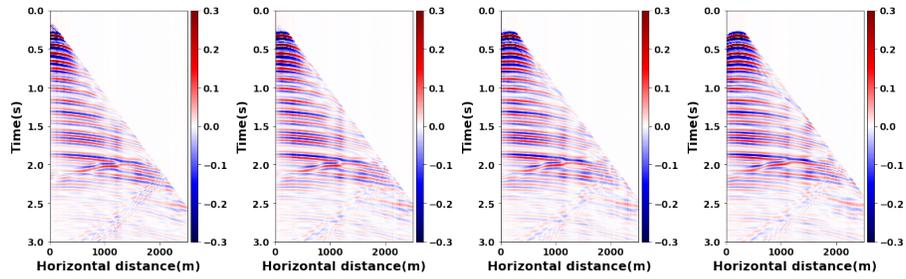
Figure 5.10: Velocity model for (a) baseline and (b) monitor systems.



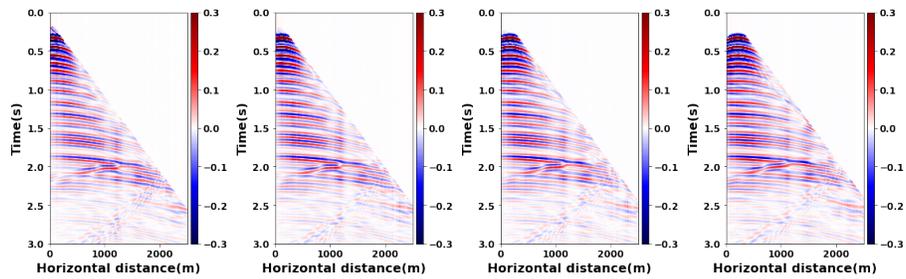
(a)



(b)



(c)



(d)

Figure 5.11: Horizontal-layered example. Four shots are selected to show data obtained from (a) observed baseline, (b) the observed monitor, (c) the predicted baseline by a shallow window and (d) the predicted baseline by a deep window.

Figure 5.12 shows data differences between two selected shots from Figure 5.11. The data differences between the observed monitor and predicted baseline by a deep window (Figure

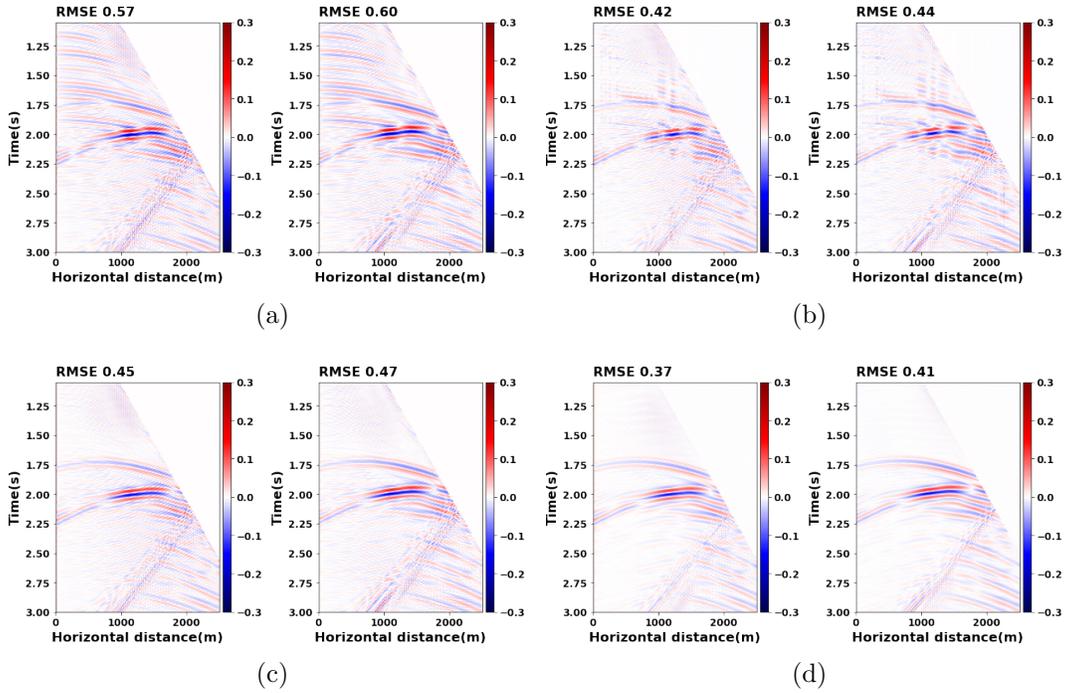


Figure 5.12: Horizontal-layered example. Two shots are extracted from Figure 5.11 to show data differences between (a) the observed monitor and baseline, (b) the observed monitor and predicted baseline by a deep window, (c) the observed monitor and predicted monitor using a matching filter, and (d) the target difference excluding near-surface change in the observed monitor data.

5.12b), have significant amplitudes after 1.6 seconds due to changes in the velocity model simulating a CO<sub>2</sub> injection. Compared to the data difference between the observed monitor and baseline (Figure 5.12a), predicted shot records can suppress time shifts and noise in the shallow section in Figure 5.12b. The RMSE values for Figure 5.12b are smaller than the data difference between acquired data sets (Figure 5.12a). This observation illustrates that predicted shot records can mitigate time shifts in the monitor data and highlight the reflections from the reservoir anomaly. Figure 5.12c shows the result after processing with the matching filter. Its RMSE values are slightly larger than those of the stacked LSTM approach. For comparison, Figure 5.12d illustrates the target difference without the near-surface changes in the acquired monitor data. Clear reflection differences can be seen between 1.6 and 2.25 seconds. Upon observation, it was found that the continuity of reflection

differences is more prominent than the proposed method. However, the proposed stacked LSTM can reduce some noise between the acquired baseline and monitor data. At the same time, it also tends to distort the signal of reflection differences slightly. Despite this limitation, the stacked LSTM model is still preferred as it improves the overall quality of the predicted data.

For further illustration, two traces are extracted from shots 3 and 10, shown in Figure 5.13. The predicted baseline by a shallow window (blue line) gives more details for shallow time series than the predicted baseline using a deep window (green line). For example, the predicted baseline is closer to observed monitor data (dashed red line) between 1.4 and 1.5 seconds in Figure 5.13a. On the other hand, because the predicted baseline by a deep window is trained with a deeper time series above the reservoir anomaly, it should contain some information about time shifts and amplitude patterns from the deep section of monitor data. For instance, the deep-window predicted baseline could forecast time shifts close to observation monitor data above the reservoir change between 1.5 and 1.6 seconds.

Migration results are shown in Figure 5.14. Observed and predicted data are migrated with a smoothed baseline velocity model to avoid information leakage. Figure 5.14e is designated as the target outcome to be attained by the prediction, as it is computed without considering the presence of near-surface changes. Compared with the observed difference migration result (Figure 5.14b), double difference migration (Figure 5.14c) has improved amplitude for reservoir change at 1800 m depth. This migrated image is then put into the U-Net, and the prediction is shown in Figure 5.14d. Based on the fundamental shape of the reservoir anomaly given in the double difference migration, the forecast after U-Net has recovered most of the anomaly with suppressed artifacts from other reflection events. Even though some noise is generated from windowing, the image has enhanced accuracy and resolution. The PSNR of the U-Net result is 21.95 dB, which indicates better quality than that of the observed difference, which has a PSNR of 19.06 dB. Additionally, the double-difference approach has a PSNR of 19.19 dB, demonstrating that the U-Net is more effective in reduc-

ing noise and improving image quality. The result shows a profound improvement, but it depends on the quality of input channels. The prediction could be biased and inaccurate if the resolution and amplitude of subsurface structures in the input are poor.

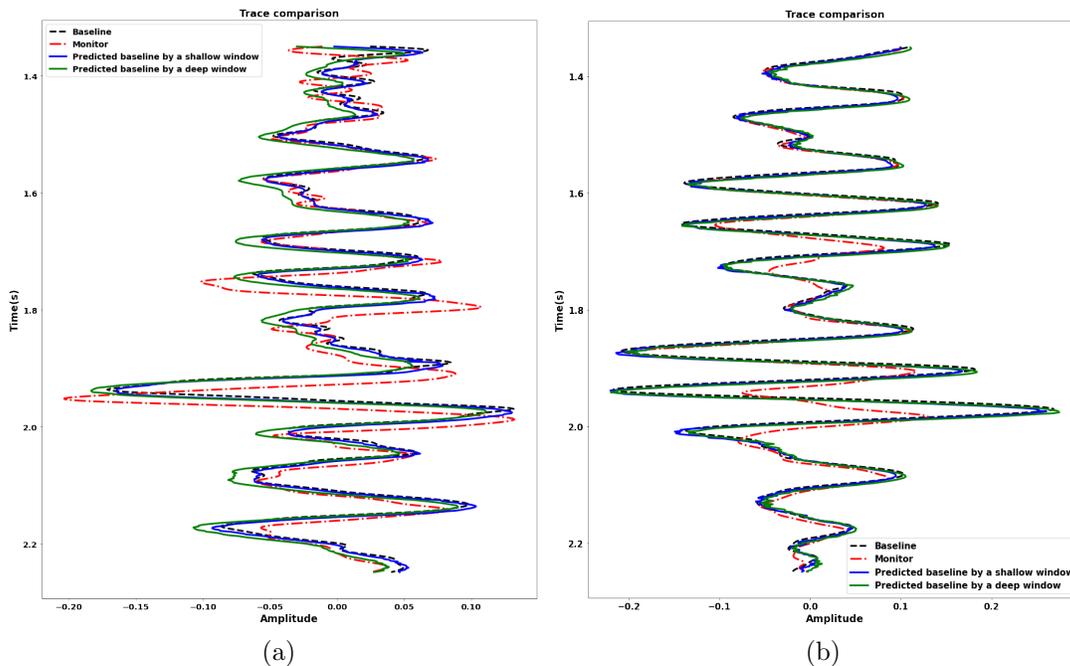


Figure 5.13: Trace comparison for shots (a) No.3 and (b) No.10.

## 5.6.2 Curve-fault model

Inspired by velocity models of an open source FWI project (Deng et al., 2021), I combined two synthetic models into one complete curve-fault model (Figure 5.15) with  $279 \times 290$  gridpoints. A curved near-surface anomaly was added in the monitor velocity, whose velocity varies from 1450 to 1550 m/s. Fifty-seven shots are simulated with 145 receivers located at 10 meters depth. The observed monitor data has curved near-surface variations at around 70 meters depth. Changes in velocity, simulating a CO<sub>2</sub> injection, are designed between depths of 1700 and 2000 meters with 1700 m/s velocity. Training parts are extracted from traces above two seconds where the reservoir change is located. A shallow window is chosen between 0.225-1.275 seconds, and a deep window ranges from 0.450 to 1.500 seconds, where the reflections

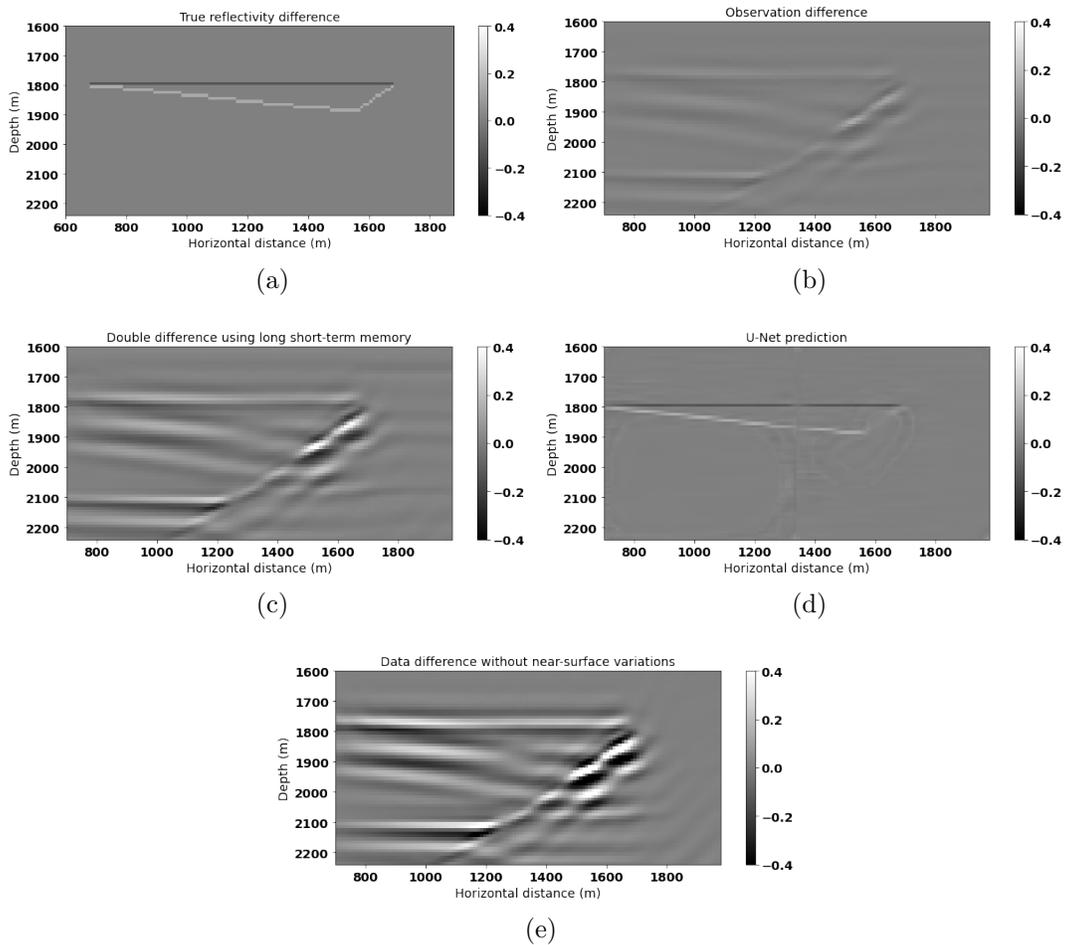


Figure 5.14: Horizontal-layered example’s (a) true reflectivity difference. Migration differences are generated by (b) the difference between the observed monitor and baseline, (c) the double-difference method, and (d) U-Net prediction. (e) Target migration difference without near-surface change.

of the reservoir change occur after 1.75 seconds.

In Figure 5.16, the differences between monitor and baseline data for two shots are compared. Figure 5.16b shows the differences between observed and predicted baseline data. This difference has fewer artifacts at the top of the shot records but shows more significant reflections from the CO<sub>2</sub> injection area. The RMSE value for the first shot in Figure 5.16b is 0.78, which is smaller than the result obtained from the classical approach with a matching filter, which is 2.19. Additionally, Figure 5.16b result is closer to the target difference, which excludes near-surface changes, as shown in Figure 5.16d. This indicates that the proposed

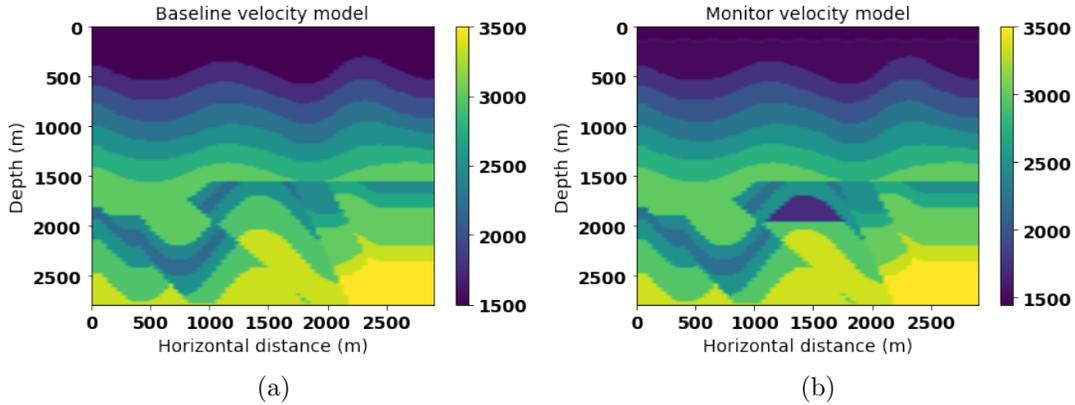


Figure 5.15: Velocity model for (a) baseline and (b) monitor systems.

method can effectively mitigate noise between the acquired baseline and monitor data while preserving significant reflection differences. The results suggest that the proposed method is superior to the matching filter in terms of accuracy and reliability in this example.

A trace comparison in Figure 5.17 shows a similar result as the previous example in that the shallow-window predicted baseline (blue) and deep-window predicted baseline (green) provide different details in transferring the observed baseline to the observed monitor above the reservoir change location. For example, the predicted baseline provides appropriate amplitudes that are close to the observed monitor above 1.4 seconds. After 1.4 seconds, the deep-window predicted baseline (green), learned with additional time-shift patterns from the observed monitor data, has a precise prediction of amplitudes and slight phase shifts above the reservoir change area.

After obtaining the calculated baseline shots by two windows, the double difference migration (Figure 5.18c) has larger amplitudes for the anomaly structure compared with the observed data difference migration (Figure 5.18b). The result is also close to the ideal target migration difference shown in Figure 5.18e, which is the result from velocity models without the near-surface difference. Accordingly, after using U-Net, the reflectivity difference prediction (Figure 5.18d) removes other irrelevant reflection events but keeps the dome structure that generates velocity change with enhanced resolution. The U-Net prediction yields the

highest PSNR value of 22.71 dB, outperforming both the observed difference (20.68 dB) and the double-difference approach (21.03 dB). The double-difference approach's PSNR is higher than the observed difference. This observation indicates that the double-difference method helps improve the prediction of reservoir anomalies.

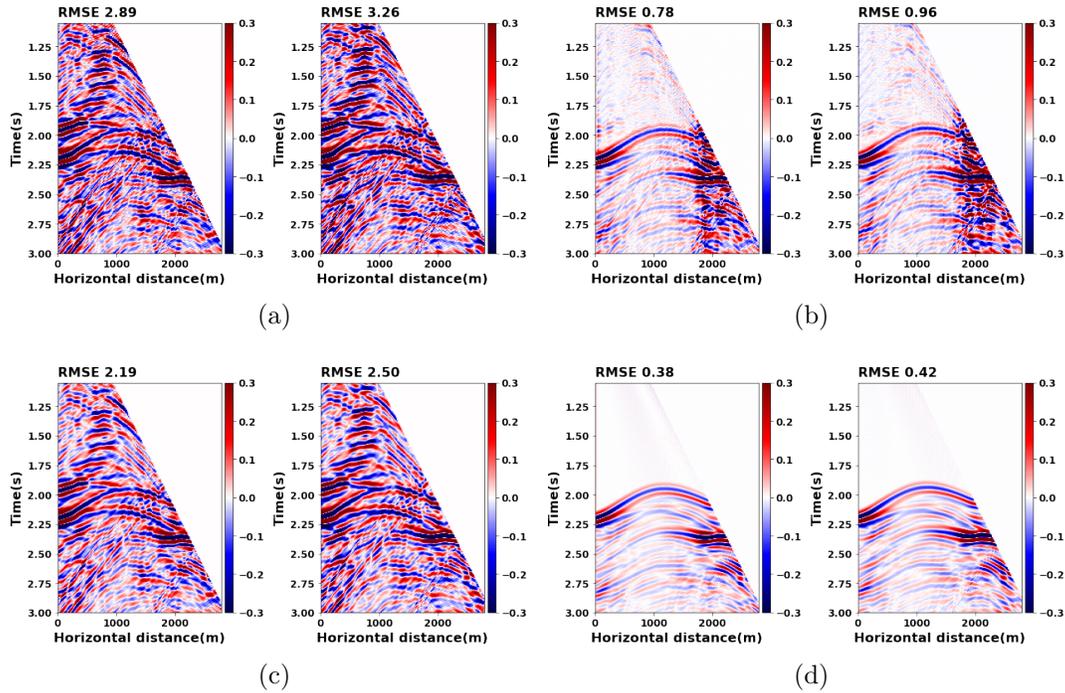


Figure 5.16: Curve-fault example. Two selected shots of data differences between (a) observed monitor and baseline, (b) observed monitor and predicted baseline by a deep window, (c) observed monitor and predicted monitor using a matching filter, and (d) the target difference excluding near-surface change in the observed monitor data.

### 5.6.3 Marmousi model

In this example, a part of the Marmousi model (Versteeg, 1994) is extracted and modified by adding several horizontal layers on the top (Figure 5.19). Now the model size changes to  $201 \times 361$  with 72 shots and 181 receivers, respectively. Three thin layers are located near the surface in the monitor velocity model, with velocity varying from 1400 to 1500 m/s to simulate near-surface changes. A velocity change, simulating  $\text{CO}_2$  injection, is introduced at

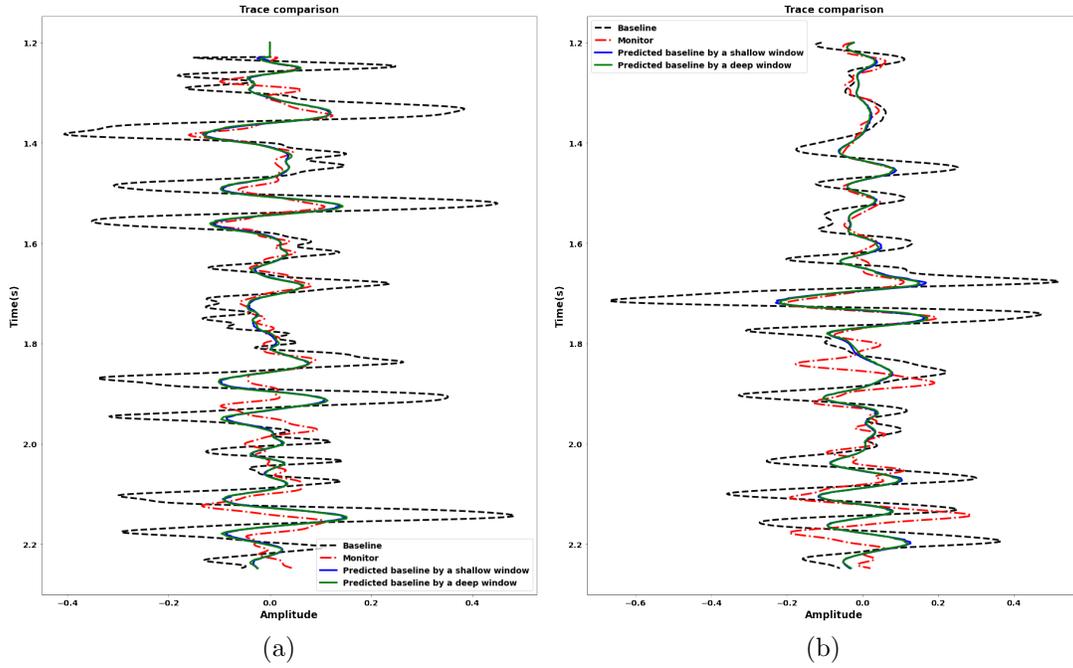


Figure 5.17: Trace comparison for shots (a) No.3 and (b) No.10.

a depth of 1500 m with 1700 m/s. This would be a good location for CO<sub>2</sub> injection because the model has a caprock on the top and a fault on the side to cut off the CO<sub>2</sub> migration path. The shallow window is defined within the time range of 0.075-1.125 seconds, while the deep window spans from 0.075 to 1.425 seconds. The reservoir change reflections occur after 2 seconds.

Since the pretended CO<sub>2</sub> injection area is located deep in the structure compared with the previous two examples, the data difference appears later in the shot record, mainly after 2 seconds. The predicted baseline data (Figure 5.20b) has fewer artifacts above 1.75 seconds compared with the matching filter result (Figure 5.20c). This observation is consistent with the target difference illustrated in Figure 5.20d. Thus, in this example, the proposed method again provides a better approach to predicting the reflections from the CO<sub>2</sub> injection area than the matching filter. The predicted shot record data by the proposed method shows fewer artifacts on the top of the time record, and the subtraction gives more significant reflections from the CO<sub>2</sub> injection area.

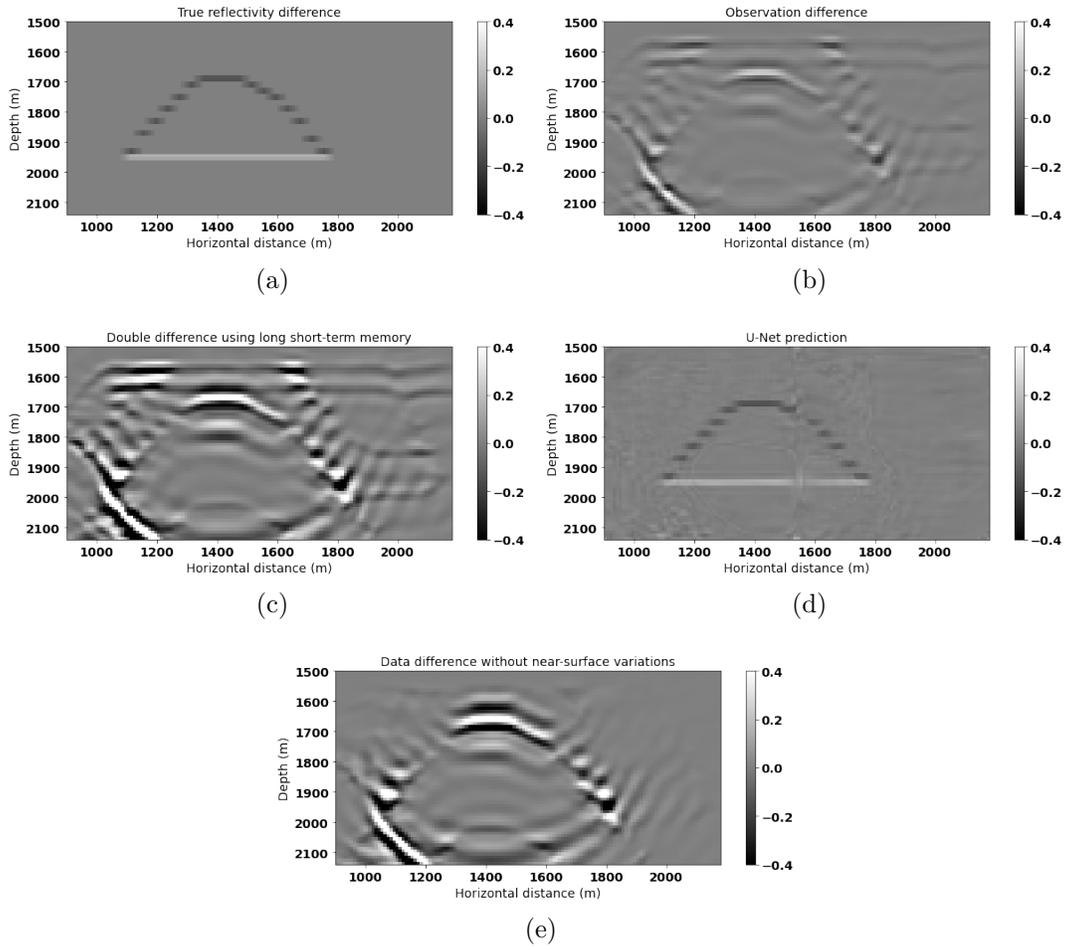


Figure 5.18: Curve-fault example’s (a) true reflectivity difference. Migration differences are generated by (b) the difference between the observed monitor and baseline, (c) double-difference method, and (d) U-Net prediction. (e) Target migration difference without near-surface change.

To further investigate the performance of the method, a trace comparison between the predicted and observed data is presented in Figure 5.21. The predicted baseline by a deep window (green line) predicts amplitude and phase shifts closer to the observed monitor (dashed red line) than the predicted baseline by a shallow window (blue line) above 2 seconds. After 2 seconds, except for reflections of the reservoir change, the shallow-window predicted baseline produced higher accuracy in estimating time shifts. Conversely, the deep-window predicted baseline offers more precise amplitude estimations.

Figure 5.22d presents the U-Net prediction of the CO<sub>2</sub> injection area. The prediction

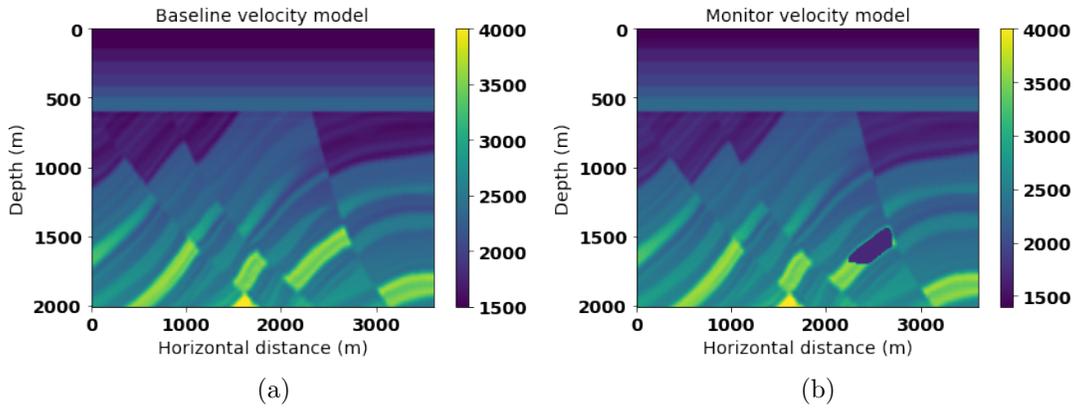


Figure 5.19: Velocity model for (a) baseline and (b) monitor systems.

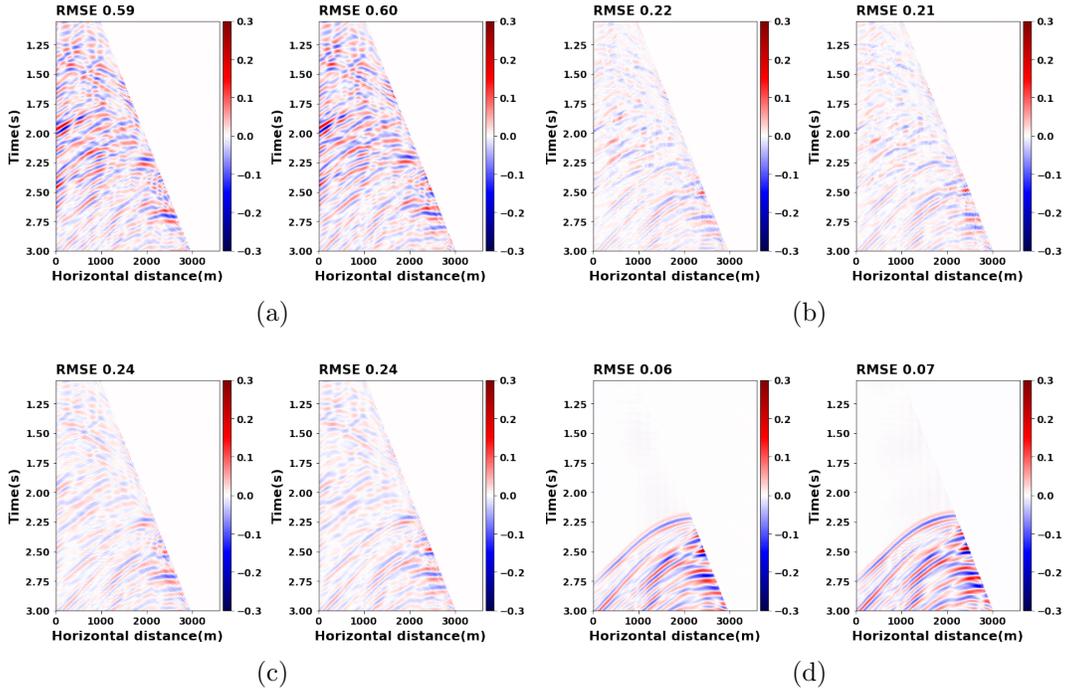


Figure 5.20: Marmousi example. Two selected shots of data differences between (a) observed monitor and baseline, (b) observed monitor and predicted baseline by a deep window, (c) observed monitor and predicted monitor using a matching filter, and (d) the target difference excluding near-surface change in the observed monitor data.

reveals a clear change in the reservoir with the high resolution based on the double-difference migrated image shown in Figure 5.22c. The amplitude of the top boundary of the reservoir change is close to the expected target (Figure 5.22e), which does not account for near-surface

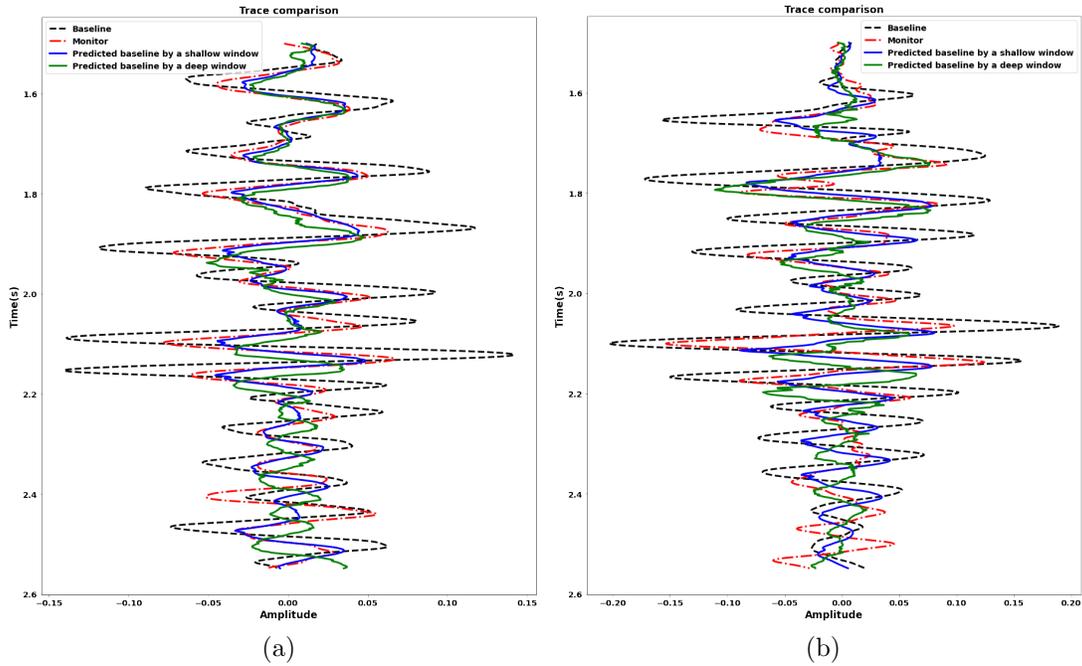


Figure 5.21: Trace comparison for shots (a) No.5 and (b) No.10.

changes. Although some noise appears above the reservoir, which could be attributed to either the U-Net learning from the noise or irrelevant reflection events, the amplitude of the predicted CO<sub>2</sub> injection area is more significant than the noise. In addition, the PSNR of the U-Net prediction is 33.37 dB, which is higher than the observed difference. In these results, the proposed double difference produces enhanced amplitudes for reflecting reservoir anomalies. Additionally, this U-Net-based method can effectively extract and predict CO<sub>2</sub> injection anomalies with high precision and accuracy, even in complex geological environments.

Migration difference	Observed	Double difference	U-Net
Horizontal-layered model	19.06	19.19	<b>21.95</b>
Curve-Fault model	20.68	21.03	<b>22.71</b>
Marmousi model	17.78	17.20	<b>33.37</b>

Table 5.1: PSNR (dB) comparison for synthetic examples.

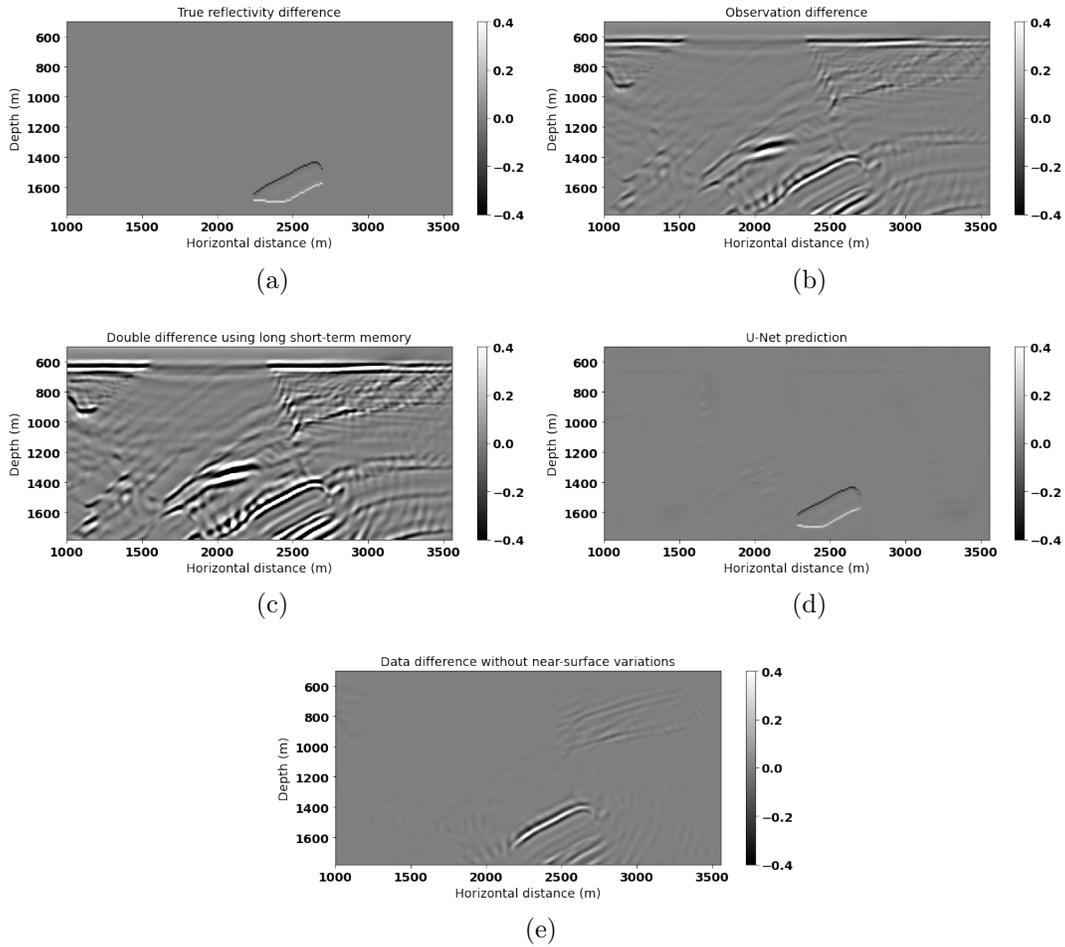


Figure 5.22: Marmousi example’s (a) true reflectivity difference. Migration differences are generated by (b) the difference between the observed monitor and baseline, (b) double-difference method, and (d) U-Net prediction. (e) Target migration difference without near-surface change.

### 5.6.4 Overthrust model

In this section, a simulation of  $\text{CO}_2$  injection migration was conducted on four slices of the Overthrust model (Aminadeh, 1997). A  $\text{CO}_2$  injection site was simulated by creating velocity changes at the top of a possible trap at around 1400 meters depth, where the velocity is 1500 m/s. As we choose different slices of the 3D model, we can see the velocity anomaly taking different shapes. The objective of the tests in this subsection could be interpreted in two different ways. One is to pretend these changes in the velocity anomalies correspond to

changes in the size of the injection as time passes. In other words, I use a 3D model as if it were a 2D model with historical data, and use the network to predict updated reservoir migration areas. The other interpretation is that we can have detailed information about the size of the anomaly in one section (slice) of the model, and use that information to predict the changes on a different sector (slice). Either interpretation is approximately consistent with a test where the first three slices are utilized for training a neural network, while the last slice is used for testing the pre-trained neural network in inference.

The Overthrust geology model contains various structures, including the combination of thin layers, folds and faults. I selected a trap around 1400 meters in depth with continuous top layers overhead to simulate the CO<sub>2</sub> injection zone. The volume and shape of this zone are customized among different slices using sine and cosine functions. To add further complexity to the simulation, I included several thin horizontal layers on top, crossing curved layers to mimic seabeds. Under the design and description above, the modified example had  $241 \times 351$  gridpoints in total, with a 10-meter spatial interval. There were 88 shots and 176 receivers located at 10 meters in depth, with 2000 time steps. The recorded time sampling rate was 0.0015 seconds. Four slices in this subsection are assumed to represent reservoir variation over time.

The primary aim was to assess the capability of the trained networks to generalize and predict the CO<sub>2</sub> injection migration in the unseen slice accurately. By training the neural networks on historical data (slices 1-3 in this case), they can be utilized to identify and forecast the potential migration of CO<sub>2</sub> injection over time. The neural networks can be deployed as a tool to provide insights and forecasts on how the CO<sub>2</sub> injection reservoir may change and evolve in the near future. The neural network's ability to recognize patterns and make predictions based on past data can be a valuable asset for anticipating future changes in the CO<sub>2</sub> injection migration pattern. As such, it can aid in the development of more effective strategies for CO<sub>2</sub> injection management and reduce the potential negative impacts and costs on the environment.

### Slices 1-3

The volume of the pretended CO<sub>2</sub> injection area expands from slices 1 to 3. Figures 5.23-5.25 show the variation of the anomaly shape over time. The reservoir anomaly in the simulation starts as a small top within a trap and gradually expands to a larger volume with over 200 meters of thickness as time progresses. The shallow window is selected within the time range of 0-0.975 seconds, while the deep window extends from 0.450 to 1.350 seconds. It should be noted that the reflections of the reservoir change are observed after 1.5 seconds.

To make the neural network work on slices and predict data within a reasonable range, I propose a workflow with two separate training tasks: a) stacked LSTM for predicting shot records, and b) U-Net for forecasting migration images. However, in this example, the pre-trained stacked LSTM is used for the next sliced model to make enhancements until all the slices are processed. The first step is to let a neural network with stacked LSTM architecture to learn general information about the Overthrust model from slice 1. The input is 88 shot records of baseline data above the CO<sub>2</sub> injection area, and the corresponding output is two-set of predicted baseline data depending on the demand. This pre-trained network is then saved and used in slice 2 for fine-tuning. Next, the fine-tuned network is applied to slice 3 for further adjustment. The final saved network is treated as the one used for testing on slice 4. Similarly, the U-Net follows the same logic but outputs the estimation of the reflectivity difference. This example aims to assess the generalization ability of the proposed method to adapt as the subsurface structure changes either by time or spatial variations.

The following figures in this subsection present the results of the neural network approach using the stacked LSTM to recover reflections from the reservoir in each of the slices. Figure 5.26-5.28 display the data differences of exemplified shot records. By predicting two sets of baseline shot records by shallow and deep windows, the neural network method can mitigate noise above 1.75 seconds and improve the amplitudes of the reflections. The performance of the neural network method is compared with the traditional matching filter approach, and the results show that the RMSE has dropped by an average of 43% with the neural network

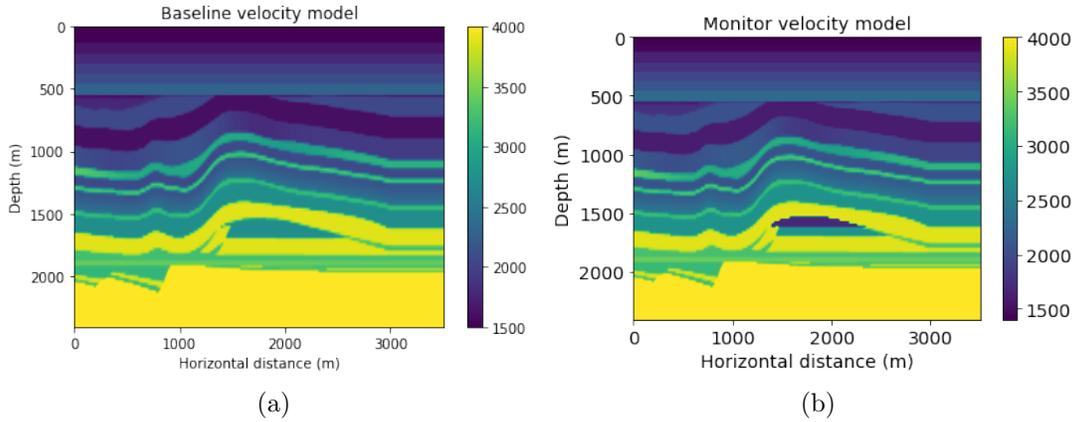


Figure 5.23: Velocity model for Overthrust slice 1 model (a) baseline and (b) monitor systems.

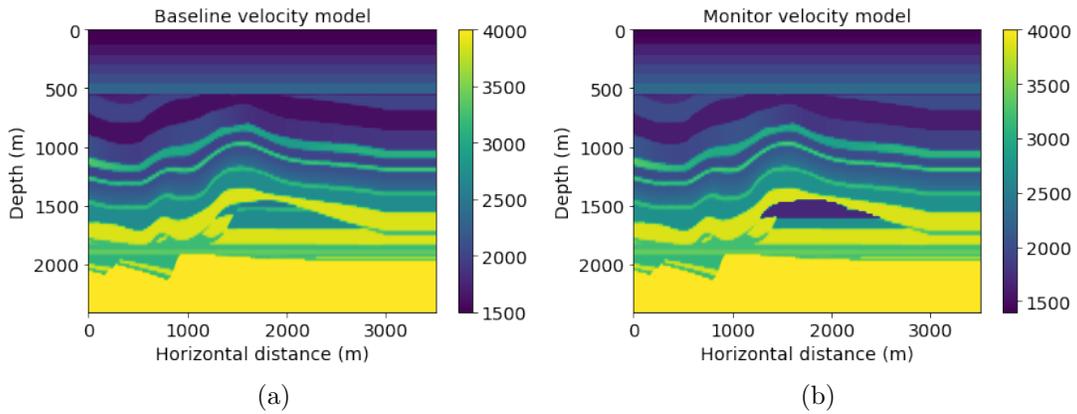


Figure 5.24: Velocity model for Overthrust slice 2 model (a) baseline and (b) monitor systems.

approach. This observation indicates that compared to using a matching filter, the neural network approach using LSTM can capture information and predict long time series with fewer artifacts.

The estimations of reflectivity differences are depicted in figures 5.29-5.31. Following the same observation shown before, the double-difference method works better in recovering the reflectivity differences caused by CO<sub>2</sub> injection, leading to increased amplitudes compared to the observation migration difference. Moreover, the output images of slices 1-3 predicted

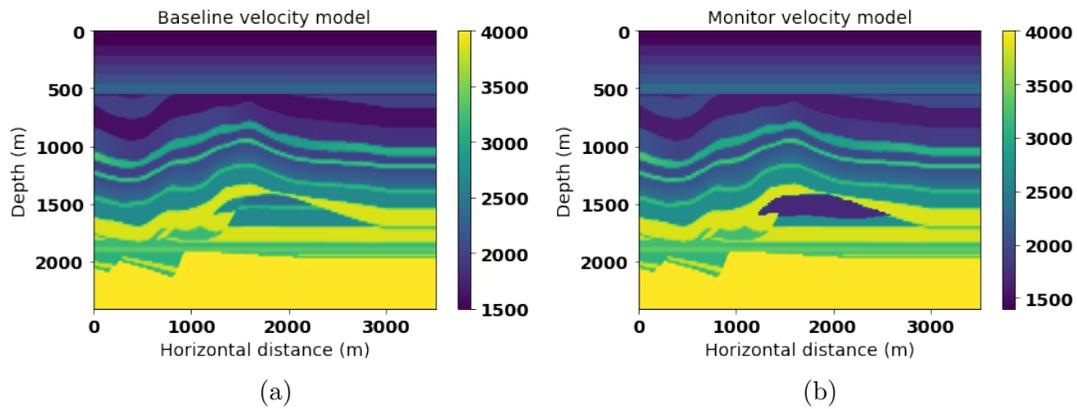


Figure 5.25: Velocity model for Overthrust slice 3 model (a) baseline and (b) monitor systems.

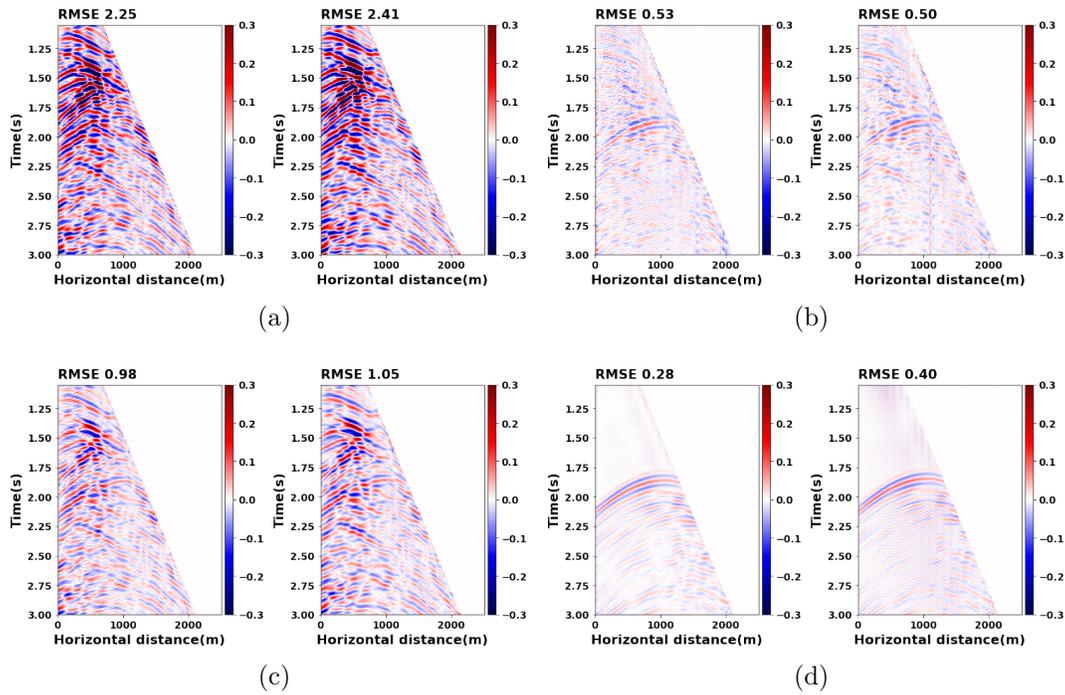


Figure 5.26: Data differences of slice 1 between (a) observed monitor and baseline, (b) observed monitor and predicted baseline using a deep window, (c) observed monitor and predicted monitor using a matching filter, and (d) the target difference excluding near-surface change in the observed monitor data.

by the U-Net display cleaner output images with boosted resolution. It proves that fine-tuned neural networks are capable of handling different reservoir volumes and enhancing the

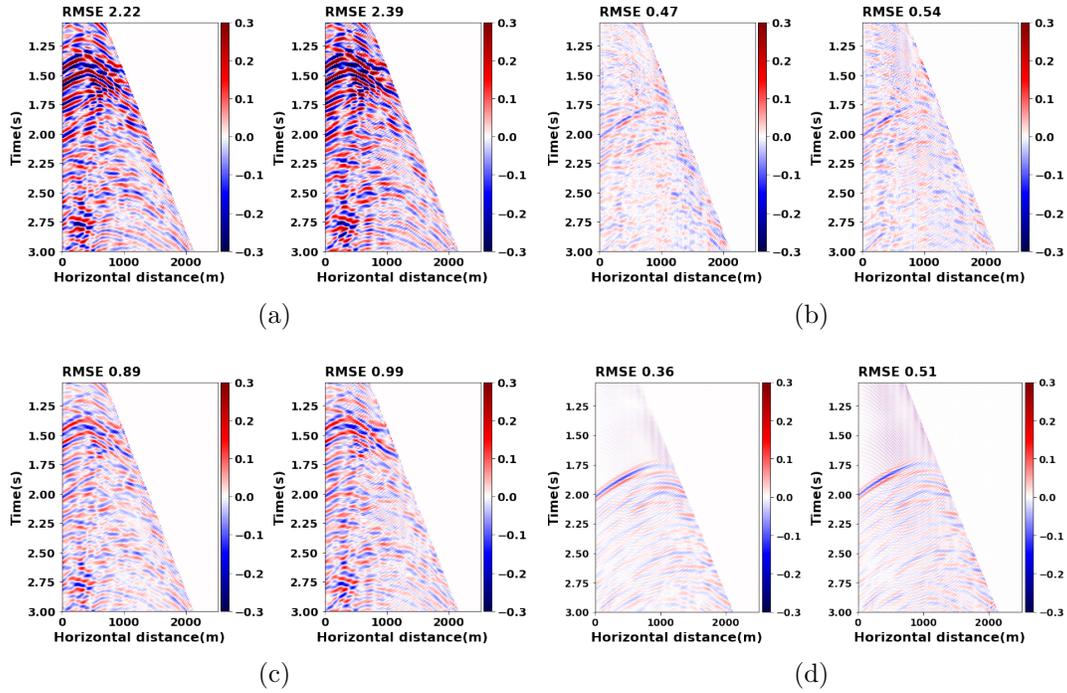


Figure 5.27: Data differences of slice 2 between (a) observed monitor and baseline, (b) observed monitor and predicted baseline using a deep window, (c) observed monitor and predicted monitor using a matching filter, and (d) the target difference excluding near-surface change in the observed monitor data.

quality of the output images.

#### Slice 4

The fine-tuned stacked LSTM and U-Net networks discussed in the previous subsection are evaluated for their generalization ability in this example. The CO<sub>2</sub> injection area with a curvature at the bottom, as shown in Figure 5.32, adds some complexity and challenges to the fine-tuned networks since shapes with large curvatures were not present in the previous slices. The training slices had slightly curved or horizontal layers. After fine-tuning the stacked LSTM network, the forecast baseline by a deep window is displayed in Figure 5.33b. It exhibits significant reflections from the CO<sub>2</sub> injection area at around 1.75 seconds, with less noise compared to the prediction after applying a matching filter (Figure 5.33c). This observation demonstrates that the fine-tuned stacked LSTM network has a good generalization

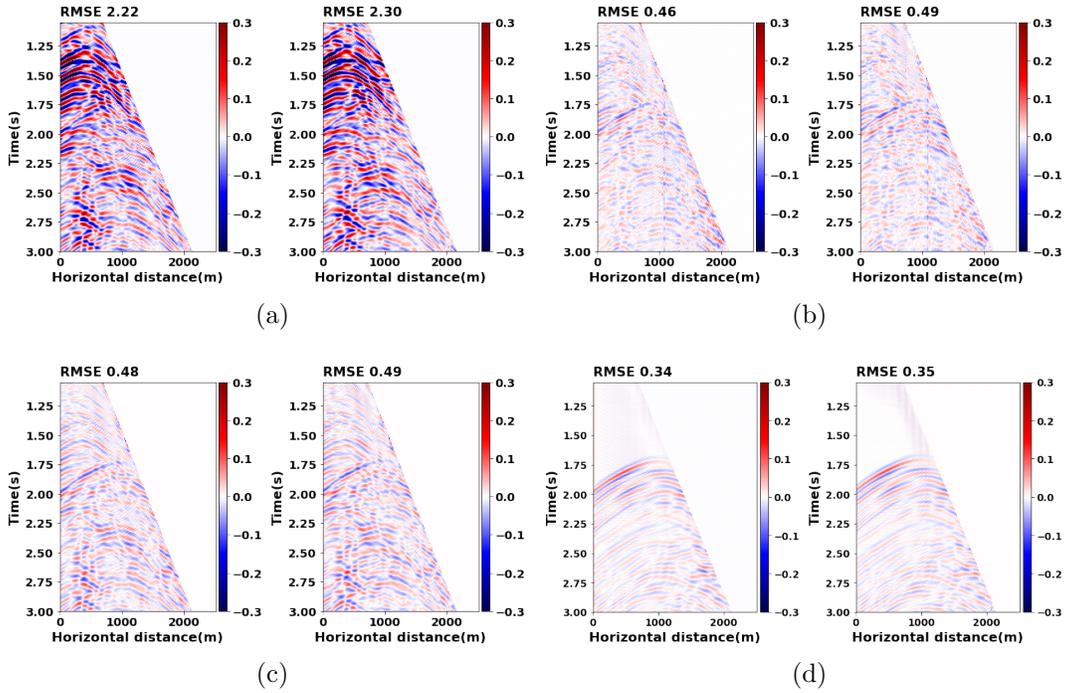


Figure 5.28: Data differences of slice 3 between (a) observed monitor and baseline, (b) observed monitor and predicted baseline using a deep window, (c) observed monitor and predicted monitor using a matching filter, and (d) the target difference excluding near-surface change in the observed monitor data.

ability, and it can predict accurately in scenarios with different shapes and structures.

The outcome obtained from the double-difference method (Figure 5.34c) exhibits increased amplitude in the reservoir change area, which closely aligns with the target expectation shown in Figure 5.34e, in contrast to the observed difference (Figure 5.34b). The U-Net predictions of the reflectivity difference are shown in Figure 5.34d, demonstrating that the fine-tuned U-Net can handle the curved shape of the CO<sub>2</sub> injection area and provide a clean and enhanced image with refined resolution. These results prove that the proposed fine-tuned neural networks can cope with reservoir volumes with different shapes and structures, providing high-quality predictions with enhanced image resolution. This observation signifies the potential of neural networks to improve the accuracy of reservoir monitoring and management in CO<sub>2</sub> injection applications.

This test is done under the assumption that the weathering layer variation is slight for

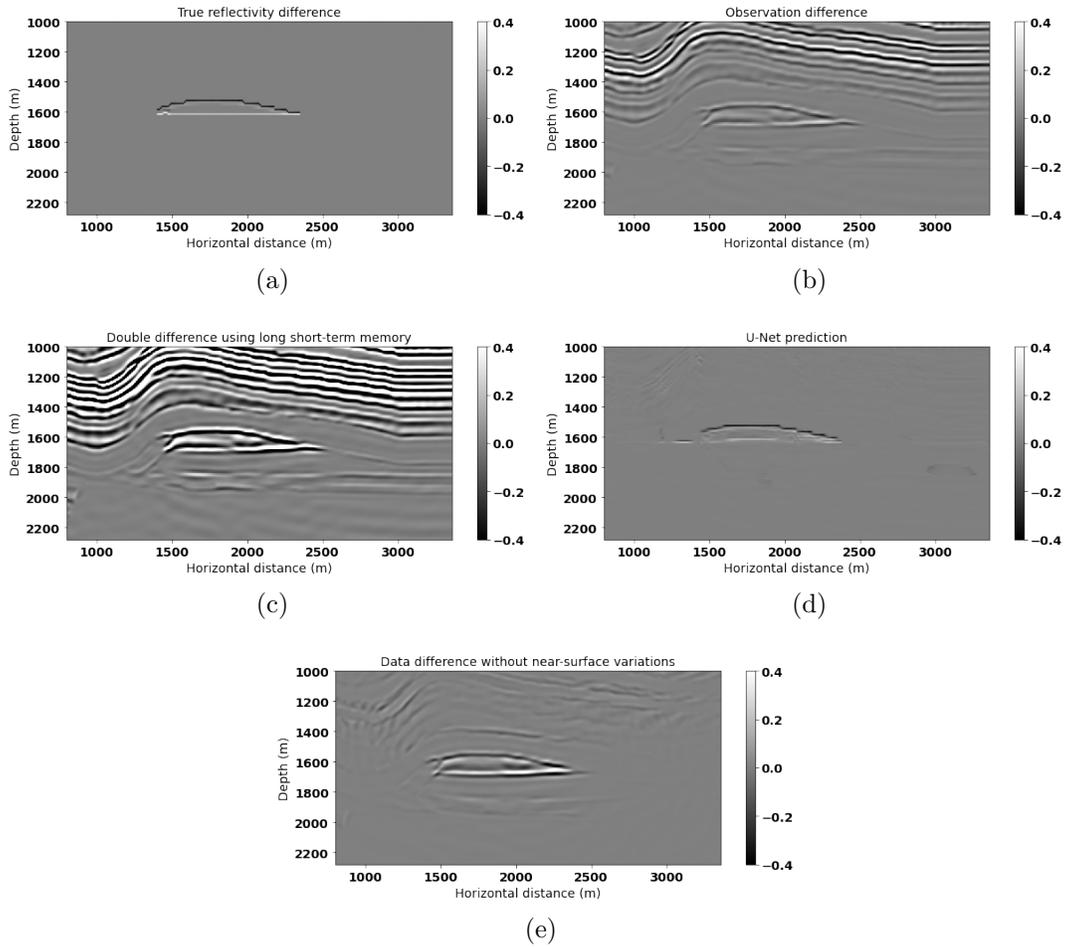


Figure 5.29: Overthrust slice 1 (a) true reflectivity difference. Migration differences are generated by (b) the difference between the observed monitor and baseline, (c) the double-difference method, and (d) U-Net prediction. (e) Target migration difference without near-surface change.

baseline and monitoring observations. Another assumption is that the noise level is relatively small compared with field data. It is important to note that these assumptions were made for the purpose of this experiment, and in future work, it is necessary to adjust the workflow to work with real field datasets. This can be done by incorporating techniques to handle the variability in weather layer and noise levels commonly observed in field data.

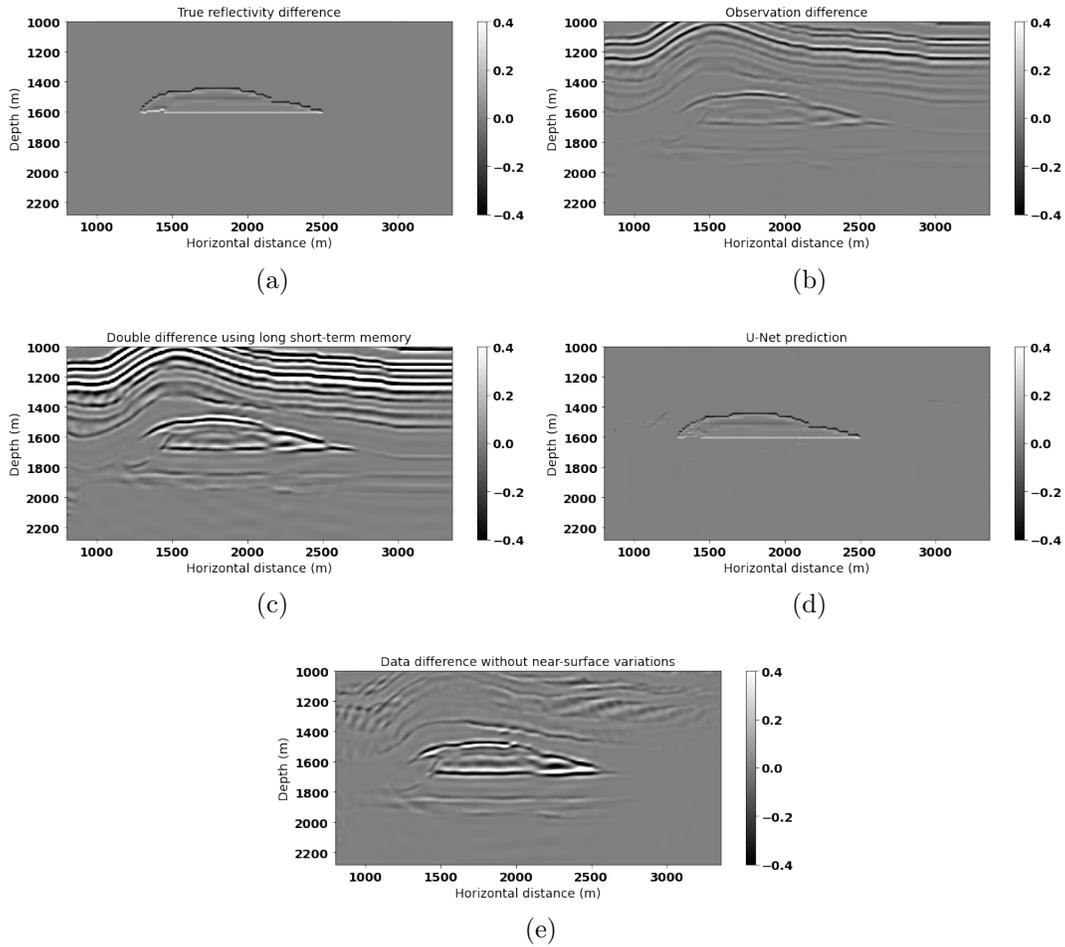


Figure 5.30: Overthrust slice 2 (a) true reflectivity difference. Migration differences are generated by (b) the difference between the observed monitor and baseline, (c) the double-difference method, and (d) U-Net prediction. (e) Target migration difference without near-surface change.

## 5.6.5 DAS VSP CaMI FRS CO<sub>2</sub> injection project

### DAS VSP data and the geology of CaMI FRS project

In this section, I will apply the proposed method to data from a Rayleigh scattering-based distributed acoustic sensing (DAS) system. DAS data are a continuous, real-time measurement using fibre optic cables to measure axial strain. The optical fibre is the sensing element. This system can detect and cover large distances, for example, an entire length of the well, and offer an opportunity to deploy some acquisition geometries for which conventional geo-

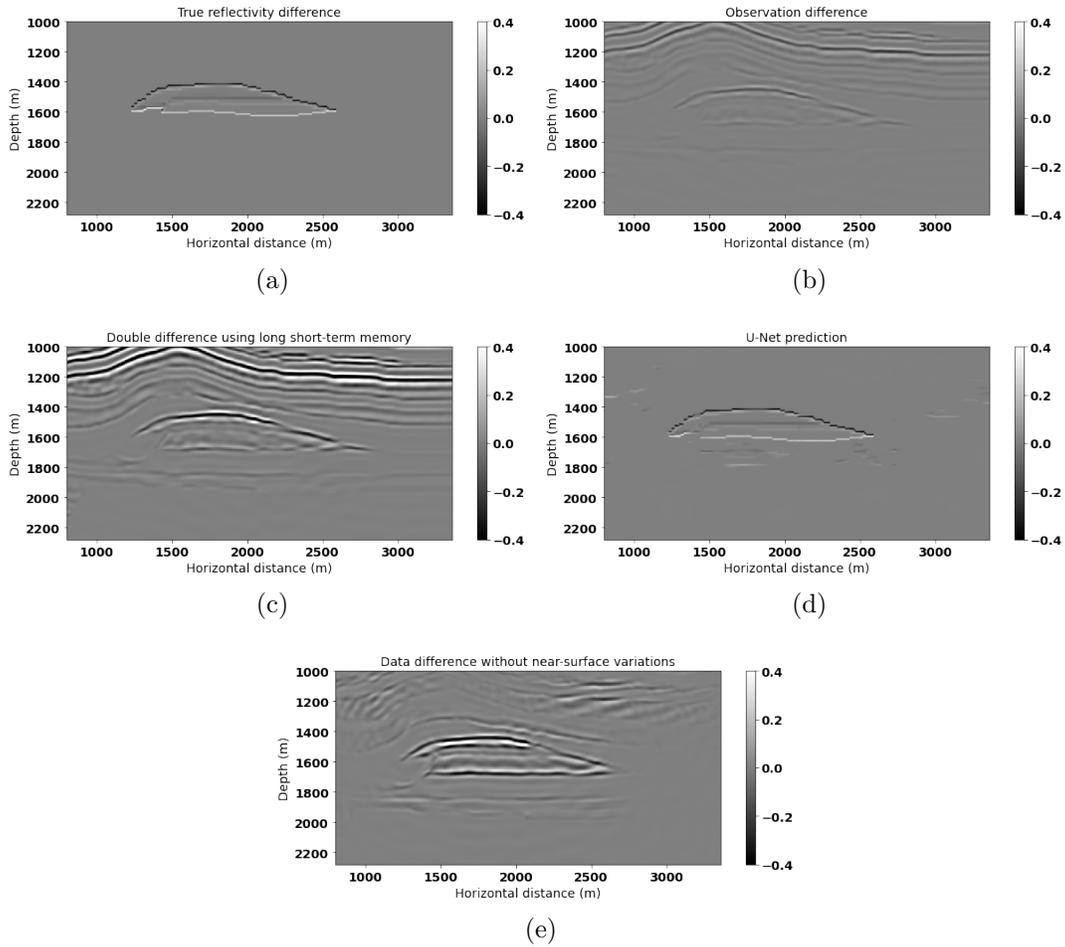


Figure 5.31: Overthrust slice 3 (a) true reflectivity difference. Migration differences are generated by (b) the difference between the observed monitor and baseline, (c) the double-difference method, and (d) U-Net prediction. (e) Target migration difference without near-surface change.

phone sensors cannot be used. The benefit of using the DAS system in time-lapse monitoring is that measurements can be recorded along the fibre with a very dense spatial sampling. Combining DAS and vertical seismic profiling (VSP) using densely sampled accelerometers, can monitor the growth of reservoir variations. This measurement is cost-effective and provides high resolution and easy repeatability near the wellbore. The deployment of DAS fibre cable in vertical wells is optimal for P-wave VSP surveys to monitor reservoir changes because DAS system fibre is mostly sensitive to longitudinal strain (Pevzner et al., 2022; Eaid et al., 2022; Wang and Lawton, 2022b).

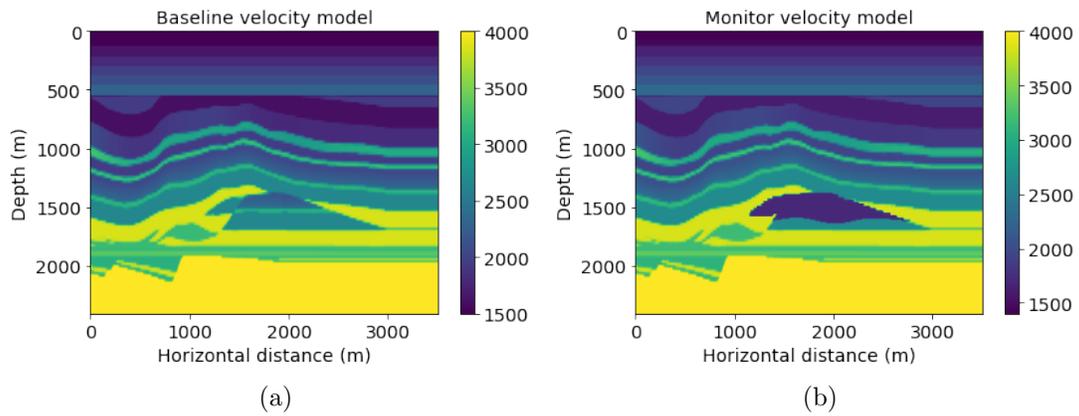


Figure 5.32: Velocity model for Overthrust slice 4 (a) baseline and (b) monitor systems.

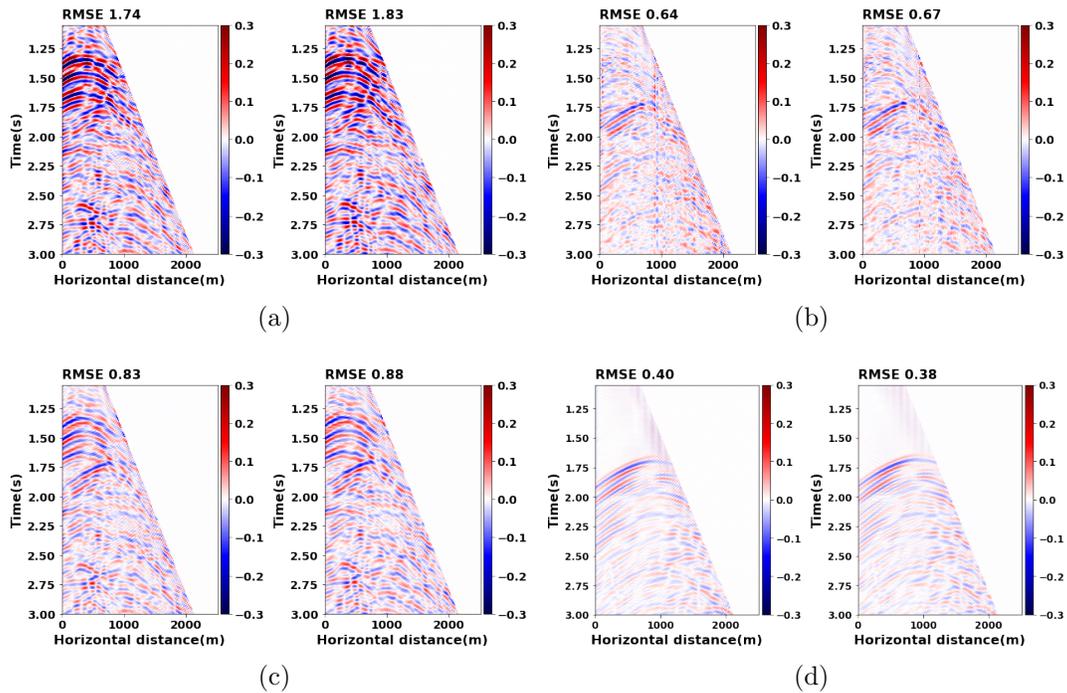


Figure 5.33: Data differences of slice 4 between (a) observed monitor and baseline, (b) observed monitor and predicted baseline using a deep window, (c) observed monitor and predicted monitor using a matching filter, and (d) the target difference excluding near-surface change in the observed monitor data.

The data for this experiment comes from the Containment and Monitoring Institute (CaMI) of Carbon Management Canada (CMC). CaMI has conducted a few DAS VSP

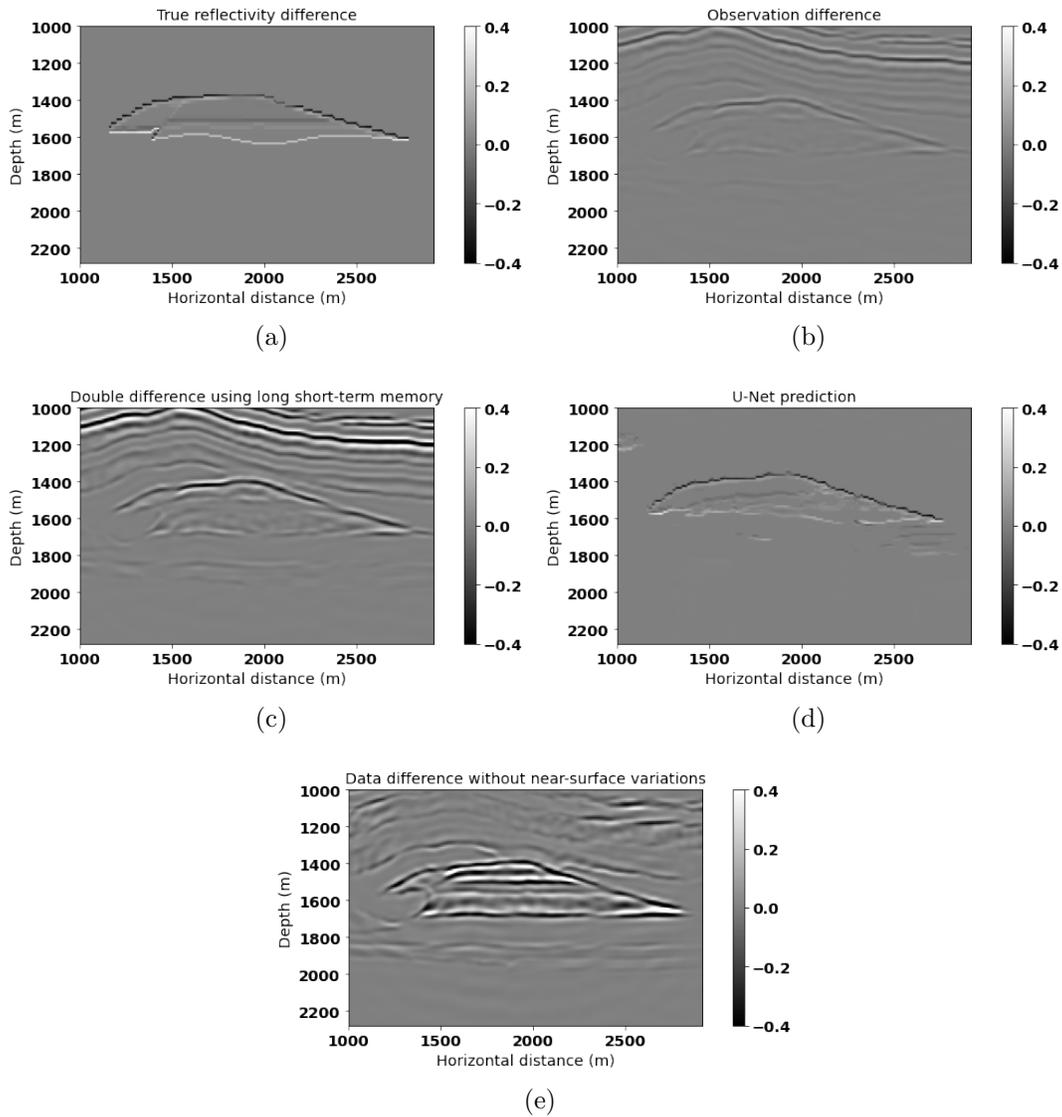


Figure 5.34: Overthrust slice 4 (a) true reflectivity difference. Migration differences are generated by (b) the difference between the observed monitor and baseline, (c) the double-difference method, and (d) U-Net prediction. (e) Target migration difference without near-surface change.

surveys for the present study at the Field Research Station (FRS), in collaboration with the University of Calgary, in Newell County, Alberta, Canada (Figure 5.35). Time-lapse DAS VSP is one of the key technologies at the FRS (Wang and Lawton, 2022b). There is one CO<sub>2</sub> injection well and two deep observation wells (OBS1 and OBS2) with a maximum depth of 348 m and 334 m separately at the FRS shown in Figure 5.36. A southwest–northeast

(SW–NE) oriented horizontal trench is set close to OBS2, where DAS fibre cables with a continuous loop are located. Figure 5.36 also shows the shots location of the one-shot walkaround and three walkaway lines survey. The baseline and monitor data is extracted from Line 13. The fibre cables are from the top to the maximum depth of 348 m in OBS1 and 334 m in OBS2. The seismic acquisition system repeatability for Line 13 is high because of the DAS fibre cables permanent installation and the same source type (Kolkman-Quinn and Lawton, 2022; Wang and Lawton, 2022b).

A small volume of CO<sub>2</sub> (several tens of tonnes per year) is injected into the shallow subsurface, the Upper Cretaceous Basal Belly River Sandstone Formation (BBRS), at a depth of approximately 300 m, to simulate an unplanned CO<sub>2</sub> migration in a deeper and larger scale (Isaac and Lawton, 2016; Lawton et al., 2019). The goal of that project seeks for the technologies to track the growth of CO<sub>2</sub> plume injection and reduce the risk of potential leakage (Macquet et al., 2022). This subsection will develop a robust neural network to locate reservoir variations and can be used for further time-lapse monitoring of CO<sub>2</sub> injection migration.

The first CO<sub>2</sub> gas plume injection was done in May 2017. A small volume of CO<sub>2</sub> of up to 20 tonnes per year has been injected into the target Basal Belly River Sandstone (BBRS) formation with a 7 m thickness. This target formation has fine- to medium-grained sandstone, composed mainly of poorly to well-sorted, angular to subangular quartz grain. The caprock is the Upper Cretaceous Foremost Formation sealing for CO<sub>2</sub> storage. It is a 152-m thick clayey sandstone with interbedded coal layers (Macquet et al., 2022). Then, the monitor data was acquired in March 2021.

The DAS data were processed following a sequence similar to VSP processing. According to Wang and Lawton (2022b), downgoing and upgoing wavefields from straight fibre in OBS2 were separated through a median filter followed by a frequency-wavenumber ( $F$ - $K$ ) filter (Figure 5.37). In this part, I apply a stacked CNN-Bi-LSTM network and the double-difference method to the processed stacked VSP data to improve CO<sub>2</sub> detection in the

injection area.

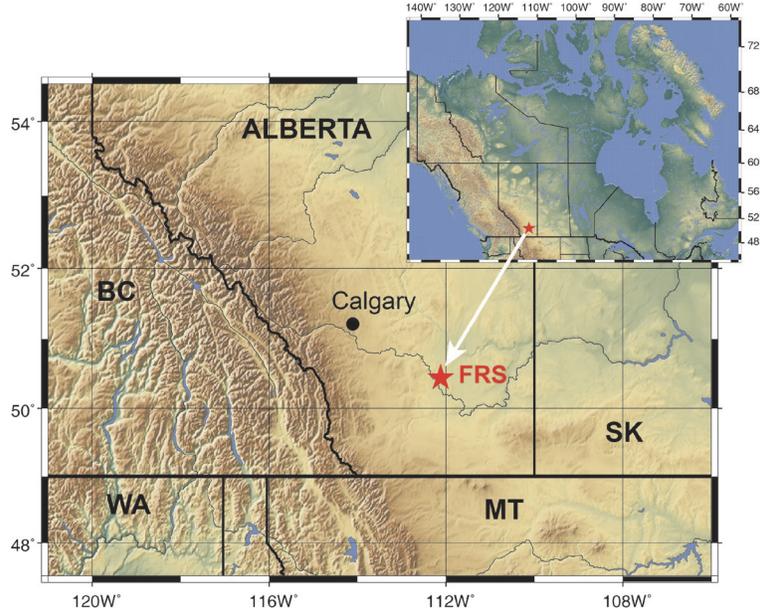


Figure 5.35: CaMI FRS DAS data acquisition location map in southern Alberta, Canada (Macquet et al., 2022).

## Result

The target formation for the CaMI FRS CO<sub>2</sub> injection project is the Basal Belly River Sandstone (BBRS), which is located at around 260 ms in DAS VSP stacked data (Figure 5.38b). By applying the proposed method, significant improvements in the amplitude of the BBRS CO<sub>2</sub> injection zone in the DAS VSP stacks can be observed in Figure 5.38f. Within the BBRS injection area, reductions in the stacked data are noticeable, which can be attributed to increased CO<sub>2</sub> saturation leading to a decrease in seismic data amplitude.

This time-lapse response prediction aligns closely with the locations of CaMI's injection well (solid line) and southwestern observation well (dashed line). The predicted CO<sub>2</sub> migration area indicates that the reservoir has migrated along the target formation over the past four years, remaining within a reasonable range. Notably, the concentration zone of CO<sub>2</sub> is concentrated near the observation well, specifically between CDP 50 and 60, at around 260

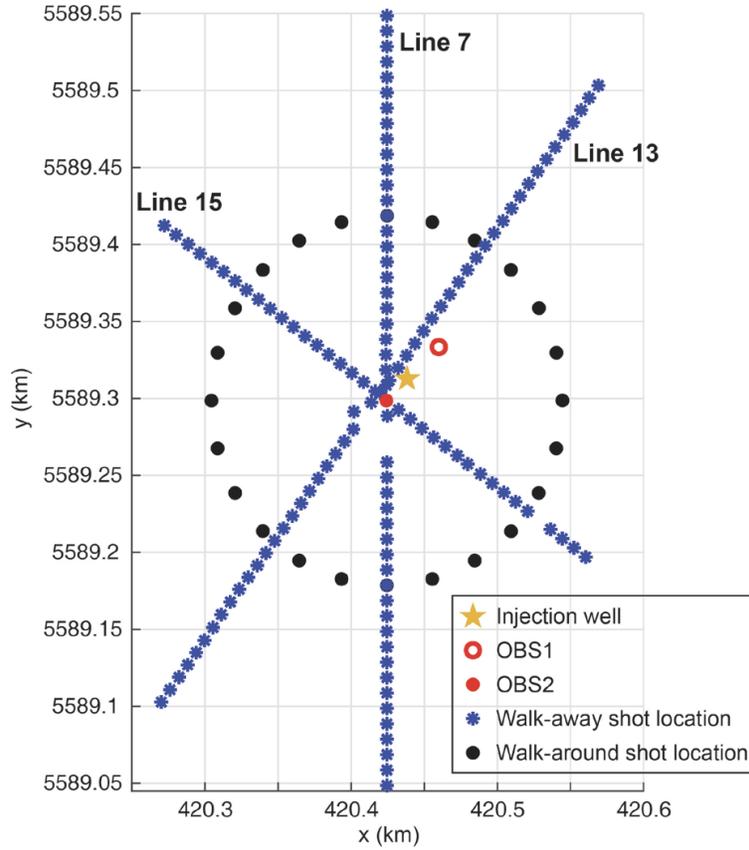


Figure 5.36: VSP layout map of the survey acquired in March 2021 (Macquet et al., 2022; Wang and Lawton, 2022a).

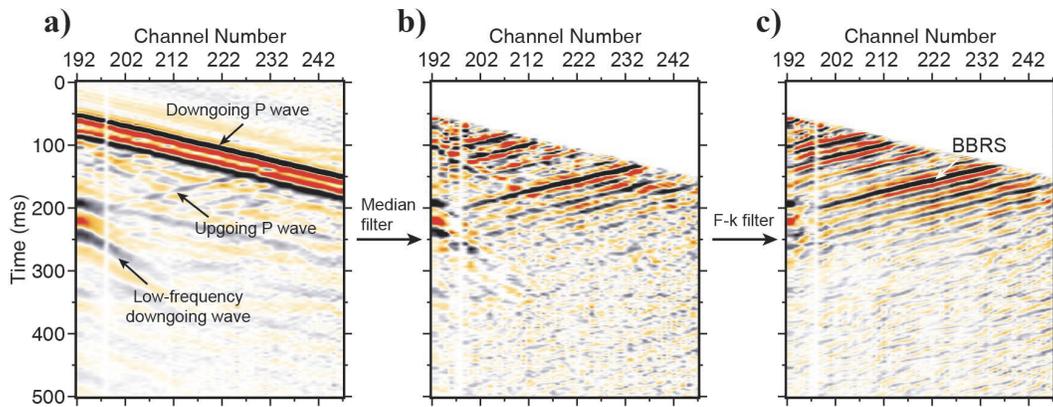


Figure 5.37: Wavefield separation: (a) raw DAS data from straight fibre in OBS2, and isolated upgoing wavefield after (b) median filter and then (c)  $F$ - $K$  filter. BBRS reflection is indicated in plot (c) (Wang and Lawton, 2022a).

ms.

For a comprehensive comparison, we present trace comparisons at DAS VSP CDP 50 (Figure 5.39a) and CDP 70 (Figure 5.39a) to illustrate the effectiveness of the proposed method. The blue line represents the predicted data difference, while the dashed line corresponds to the observed difference.

In Figure 5.39, a decrease in the amplitude of the stacked data is observed after the injection of CO<sub>2</sub> at a depth of 260 meters. This reduction in amplitude aligns with the expected behaviour, as the density decreases in the area where CO<sub>2</sub> is concentrated. The predicted data difference can capture this change. Additionally, apart from the CO<sub>2</sub> injection zone, a decrease in the predicted data difference is observed at around 220 and 230 meters. This indicates that the proposed method effectively mitigates the noise between baseline and monitor data, emphasizing the reservoir anomaly.

This method serves as a valuable tool for tracking future CO<sub>2</sub> migration, ensuring that any potential leakage to other formations can be detected and prevented effectively.

## 5.7 Discussion

In addition to using clean traces in synthetic examples, I also examined the performance of the model when different levels of noise were added to the input traces. To be precise, I tested two levels of noise: 6% and 18%.

When 6% noise was added to the baseline data, the predicted results showed minor differences compared to using the clean traces, although the training accuracy was slightly higher. On the other hand, adding 18% noise resulted in poorer performance on the monitor data prediction. During training, the training and validation loss showed fluctuations, indicating unstable training. Although the training loss reached low values at some points, the overall instability was a concern.

These findings suggest that while the proposed method is capable of handling some levels of noise in the input data, higher levels of noise may adversely affect its performance. In

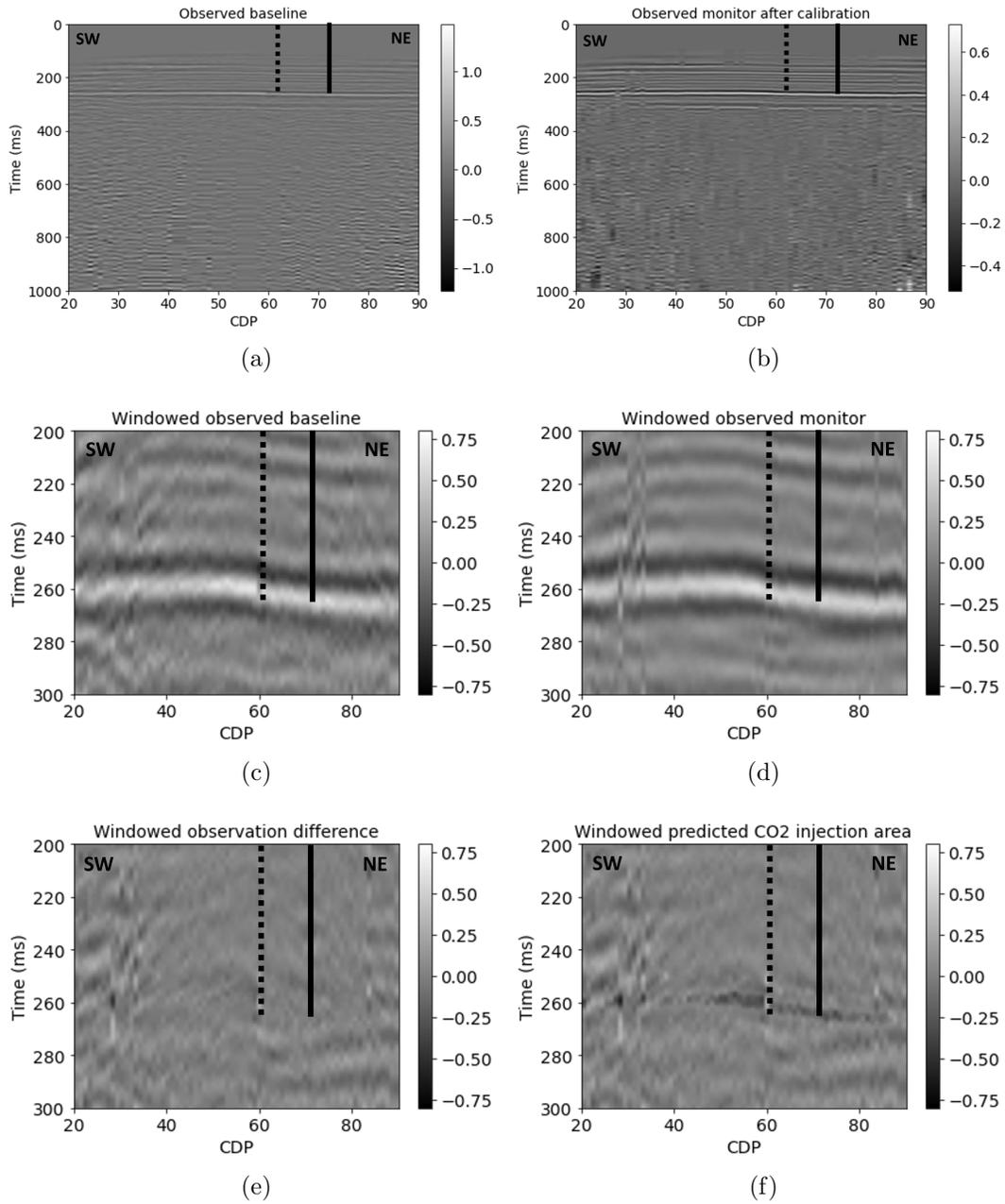


Figure 5.38: DAS VSP stacked data from CaMI FRS. (a) Observed baseline (b) Observed monitor after calibration. (c) Windowed observed baseline. (d) Windowed observed monitor. (e) Windowed observation difference. (f) Windowed predicted CO<sub>2</sub> injection area.

future work, it would be important to investigate ways to improve the model's robustness to noise.

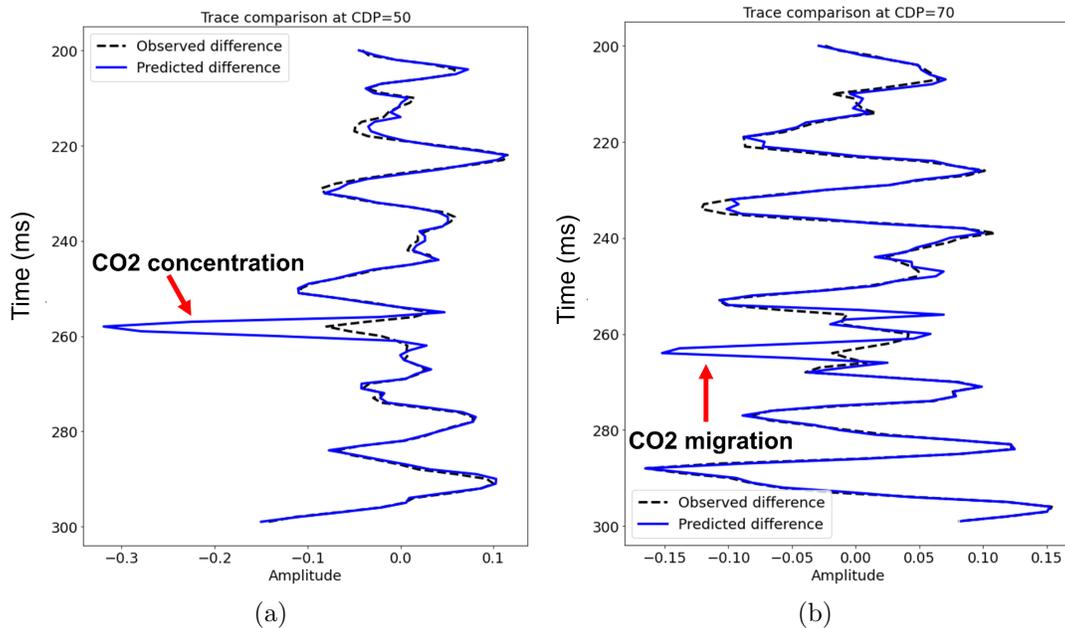


Figure 5.39: Trace comparison for DAS VSP data CDP (a) 50 and (b) 70.

## 5.8 Conclusion

In this chapter, I proposed a novel approach that utilizes stacked LSTM, CNN-Bi-LSTM, and U-Net neural networks, and the double-difference method to learn changes due to the near-surface and shallow depth signal information, including multiple reflections, above the reservoir changes. By using shallow and deep windows, the method can be used in time-lapse monitoring to predict unwanted noise patterns and/or data distortions in the baseline data. The noise is then subtracted from observed monitor data using the double-difference method. Numerical examples show that this method can improve amplitudes of the reservoir change with artifact suppression, ultimately mitigating noise and improving the migration image. Additionally, fine-tuned stacked LSTM and U-Net can be generalized to geology structures with similar patterns in the pre-training process. This implies that the pre-trained models could be reused for reservoirs with similar characteristics, where true velocity changes are unknown making it impossible to re-train the network. Moreover, this method also works for real DAS data that redirect the possible area for CO<sub>2</sub> injection migration.

Future work can focus on extending the method to a broader range of complex geology models for both training and testing and exploring the transferability of pre-trained models to different geological structures and evaluating their performance. By testing this neural network architecture and workflow in various scenarios, the model can be further generalized to other subsurface imaging situations. Additionally, exploring the use of alternative neural network architectures and incorporating additional sources of information, such as well data, can provide further improvements to this approach.

# Chapter 6

## Conclusion

### 6.1 Summary

Multiple reflections are considered critical and expected to play an essential role in seismic imaging, as they can provide valuable information about subsurface structures. Using multiple energy can significantly expand the subsurface illumination and improve the quality of the resulting reflectivity images. This thesis presents three main approaches for utilizing multiples in seismic imaging.

The approach described in Chapter 2 involves PSPI migration with scattering terms. It can extend the aperture information and provide reflectors with higher resolution and amplitudes than without scattering terms by generating internal reflection perturbations, even when given sparse and coarse shot acquisition data. The migration result can provide high-resolution estimates of reflectivity. Furthermore, this approach helps to alleviate the aperture limitation problem commonly encountered in migration techniques.

Chapter 3 presents the RTMM-CNN method, which introduces multiple reflections in a U-Net to enhance the quality of reflectivity obtained from migration, mainly when applied to a smooth initial model. The trained neural network takes advantage of multiple reflections and a reflectivity input from the background velocity model, enhancing subsurface structure

illumination and allowing the neural network to accommodate velocity errors. Before fine-tuning the network, a neural network trained with multiple reflections and an accurate velocity model serves as a preconditioner that limits the range of potential parameters due to the additional information it incorporates. After feeding a smoothed reflectivity into the pre-trained model, a new fine-tuned model can be obtained by further training the neural network to account for the potential biases caused by preconditioning. The U-Net operator approximates the inverse of the Hessian, which helps suppress image artifacts and enhance the resolution of reflectors. The RTMM-CNN examples show generalization abilities across a range of geological models, indicating its potential for use in various subsurface imaging scenarios when given smooth inputs—for example, thin layers, faults and folds, and salt bodies with lateral velocity variations.

Chapter 4 presents an extended version of the RTMM-CNN method proposed in Chapter 3, called RTMM-CNN with a DWT subband channel. This update provides an improved prediction of the reflectivity coefficient. Adding a DWT subband LL and a pre-trained model enables the fine-tuned model to extract key features from low-frequency information and tolerate more artifacts from smooth input. Using multiple reflection energy provides additional subsurface illumination, assisting the neural network in distinguishing signals from noise.

Chapter 5 introduces a new method that utilizes stacked long short-term memory and U-Net neural networks, in conjunction with the double-difference process, to learn near-surface noise and shallow depth signal information, including multiple reflections, present above reservoir changes like the ones present in CO<sub>2</sub> injection areas. By predicting two sets of baseline data by shallow and deep windows assigned using trained neural networks, the model can recover data anomalies of the reservoir change while mimicking noise patterns. The double-difference method is then used to subtract the noise from the observed monitor data, improving the amplitudes of the reservoir change, with artifact suppression. This method was tested to be effective in numerical and real DAS data, redirecting the possible

area for CO<sub>2</sub> injection migration. The predicted reflectivity model has improved quality and accuracy. In addition, it is possible to apply the fine-tuned stacked LSTM and U-Net models to geological structures with comparable patterns during the pre-training phase. This indicates that the pre-trained models can be utilized again for reservoirs with similar features, thereby decreasing the expense and duration required for training new models.

## 6.2 Future work

In the method of PSPI migration with scattering terms, although this approach can migrate the reflector location at the deeper structure with increased amplitudes, multiple reflections can create incoherent noise below the reflector structure due to wavefield inferences. To further enhance the accuracy of the migration method, it is essential to address the issue of incoherent noise that appears below the structure due to wavefield inferences. Future work can focus on developing new approaches to remove this noise and improve the accuracy of the migration method.

In Chapter 3, the RTMM-CNN approach presents a promising way to utilize multiple reflections in subsurface imaging with U-Net, especially in situations with limited velocity information. However, there is still scope for improvement in future work. One possible direction is to investigate the effectiveness of the proposed method in more complex geological environments with real field datasets. Another direction is to explore further the use of more advanced deep learning techniques, such as attention mechanisms, to enhance the resolution and fidelity of the reconstructed reflectivity. Additionally, it would be beneficial to investigate the impact of different acquisition parameters, such as the shot and receiver intervals and acquisition systems, on the performance of the proposed method. Finally, as migration velocity models continue to evolve, it is expected that the effectiveness of the neural network will be further improved by incorporating even smoother models.

Future work in Chapter 4 can focus on extending the method to predict geology models with rapid lateral velocity changes, and the method's performance can be evaluated on real-

world field data. Additional efforts can be directed towards exploring the incorporation of more DWT subbands, including high-frequency information, to improve the method's accuracy further.

In the context of time-lapse seismic monitoring, neural networks' generalization ability is limited. The performance of a trained neural network may vary significantly when exposed to different real-world data, so the model's ability is restricted to offer comprehensive insights and predictions. To address this challenge, further research can develop techniques, such as adding domain knowledge in the networks and expanding neural networks to a broad range of complex geological models for training, testing and transfer learning purposes. This would allow the neural network architecture and workflow to be tested and refined for different scenarios, leading to a more generalized model for subsurface imaging applications. Additionally, exploring the use of alternative neural network architectures and integrating supplementary information, such as well-log data, can potentially enhance the performance of this approach.

# Bibliography

- Acharya, T., and A. K. Ray, 2005, Image processing: principles and applications, chapter 5: John Wiley & Sons.
- Alali, A., V. Kazei, B. Sun, and T. Alkhalifah, 2022, Time-lapse data matching using a recurrent neural network approach: *Geophysics*, **87**, 1–83.
- Alardi, M., 2022, Elastic properties and litho-fluid facies estimation from pre-stack seismic data through bidirectional long short-term memory networks: *Geophysical Prospecting*, **70**, 558–577.
- Alfarraj, M., and G. AlRegib, 2019, Semisupervised sequence modeling for elastic impedance inversion: *Interpretation*, **7**, SE237–SE249.
- Alsdorf, D., 1997, A simple method for migrating narrow aperture, noisy seismic reflection data and application to project indepth (international deep profiling of tibet and the himalaya) deep seismic profiles: *Journal of Geophysical Research: Solid Earth*, **102**, 17807–17811.
- Alwon, S., 2018, Generative adversarial networks in seismic data processing: Presented at the 2018 SEG International Exposition and Annual Meeting, OnePetro.
- Aminadeh, F., 1997, Seg/eaga 3-d salt and overthrust models: *SEG/EAGE 3-D Modeling Series* No. 1, SEG.
- Aoki, N., and G. T. Schuster, 2009, Fast least-squares migration with a deblurring filter: *Geophysics*, **74**, WCA83–WCA93.
- Araújo, F. V., A. B. Weglein, P. M. Carvalho, and R. H. Stolt, 1994, Inverse scattering

- series for multiple attenuation: An example with surface and internal multiples, *in* SEG Technical Program Expanded Abstracts 1994: Society of Exploration Geophysicists, 1039–1041.
- Arts, R., O. Eiken, A. Chadwick, P. Zweigel, L. Van der Meer, and B. Zinszner, 2003, Monitoring of co2 injected at sleipner using time lapse seismic data: Greenhouse Gas Control Technologies-6th International Conference, Elsevier, 347–352.
- Asnaashari, A., R. Brossier, S. Garambois, F. Audebert, P. Thore, and J. Virieux, 2015, Time-lapse seismic imaging using regularized full-waveform inversion with a prior model: which strategy?: *Geophysical prospecting*, **63**, 78–98.
- Ayeni, G., and B. Biondi, 2010, Target-oriented joint least-squares migration/inversion of time-lapse seismic data sets: *Geophysics*, **75**, R61–R73.
- Bagaini, C., E. Bonomi, and E. Pieroni, 1995, Data parallel implementation of 3-d pspi: *SEG Technical Program Expanded Abstracts*.
- Baysal, E., D. D. Kosloff, and J. W. Sherwood, 1983, Reverse time migration: *Geophysics*, **48**, 1514–1524.
- Bergen, K. J., P. A. Johnson, M. V. de Hoop, and G. C. Beroza, 2019, Machine learning for data-driven discovery in solid earth geoscience: *Science*, **363**, eaau0323.
- Bergmann, P., A. Kashubin, M. Ivandic, S. Lüth, and C. Juhlin, 2014, Time-lapse difference static correction using prestack crosscorrelations: 4d seismic image enhancement case from ketzin: *Geophysics*, **79**, B243–B252.
- Berkhout, A., 1982, Imaging of acoustic energy by wave field extrapolation.
- , 2014, An outlook on the future of seismic imaging, part ii: Full-wavefield migration: *Geophysical Prospecting*, **62**, 931–949.
- Berkhout, A., and D. J. Verschuur, 1994, Multiple technology: Part 2, migration of multiple reflections, *in* SEG Technical Program Expanded Abstracts 1994: Society of Exploration Geophysicists, 1497–1500.
- Bhasker, J., and R. Chadha, 2009, Crosstalk and noise, *in* Static timing analysis for nanome-

- ter designs: A practical approach: Springer, 147–177.
- Birnie, C., and F. Hansteen, 2022, Bidirectional recurrent neural networks for seismic event detection: *Geophysics*, **87**, KS97–KS111.
- Calderón-Macías, C., M. K. Sen, and P. L. Stoffa, 2000, Artificial neural networks for parameter estimation in geophysics [link]: *Geophysical prospecting*, **48**, 21–47.
- Caudill, M., 1987, Neural networks primer, part I: *AI Expert*, **2**, 46–52.
- Chadwick, A., G. Williams, N. Delepine, V. Clochard, K. Labat, S. Sturton, M.-L. Buddensiek, M. Dillen, M. Nickel, A. L. Lima, et al., 2010, Quantitative analysis of time-lapse seismic monitoring data at the sleipner co 2 storage operation: *The Leading Edge*, **29**, 170–177.
- Claerbout, J. F., 1971, Toward a unified theory of reflector mapping: *Geophysics*, **36**, 467–481.
- , 1984, *Imaging the earth’s interior*: Stanford University Stanford.
- Cui, Z., R. Ke, Z. Pu, and Y. Wang, 2018, Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction: arXiv preprint arXiv:1801.02143.
- Daley, T. M., B. M. Freifeld, J. Ajo-Franklin, S. Dou, R. Pevzner, V. Shulakova, S. Kashikar, D. E. Miller, J. Goetz, J. Henniges, et al., 2013, Field testing of fiber-optic distributed acoustic sensing (das) for subsurface seismic monitoring: *The Leading Edge*, **32**, 699–706.
- Das, V., A. Pollack, U. Wollner, and T. Mukerji, 2019, Convolutional neural network for seismic impedance inversion: *Geophysics*, **84**, R869–R880.
- Davydenko, M., and D. Verschuur, 2017, Full-wavefield migration: using surface and internal multiples in imaging: *Geophysical Prospecting*, **65**, 7–21.
- Deng, C., S. Feng, H. Wang, X. Zhang, P. Jin, Y. Feng, Q. Zeng, Y. Chen, and Y. Lin, 2021, OpenFWI: large-scale multi-structural benchmark datasets for seismic full waveform inversion: arXiv preprint arXiv:2111.02926.
- 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: Society of Exploration Geophysicists, 1–5.
- Du, G., Z. Wang, B. Gao, S. Mumtaz, K. M. Abualnaja, and C. Du, 2020, A convolution bidirectional long short-term memory neural network for driver emotion recognition: IEEE Transactions on Intelligent Transportation Systems.
- Eaid, M., S. Keating, and K. Innanen, 2022, Combined elastic fwi of accelerometer and das vsp data from a co2 sequestration test site in newell county, alberta: Presented at the SEG/AAPG International Meeting for Applied Geoscience & Energy, OnePetro.
- Erhan, D., A. Courville, Y. Bengio, and P. Vincent, 2010, Why does unsupervised pre-training help deep learning?: Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, 201–208.
- Ferguson, R. J., and G. F. Margrave, 2005, Planned seismic imaging using explicit one-way operators: Geophysics, **70**, S101–S109.
- Fu, X., and K. A. Innanen, 2022, Time-lapse seismic imaging using shot gathers with non-repeatable source wavelets: Geophysics, **88**, 1–108.
- Gal’Perin, E., 1974, Vertical seismic profiling: Seg special publication no. 12: Society of Exploration Geophysicists, Tulsa.
- Gazdag, J., 1978, Wave equation migration with the phase-shift method: Geophysics, **43**, 1342–1351.
- Gazdag, J., and P. Sguazzero, 1984, Migration of seismic data by phase shift plus interpolation: Geophysics, **49**, 124–131.
- Ghazali, K. H., M. F. Mansor, M. M. Mustafa, and A. Hussain, 2007, Feature extraction technique using discrete wavelet transform for image classification: 2007 5th Student Conference on Research and Development, IEEE, 1–4.
- Gilpatrick, R., and D. Fouquet, 1989, A user’s guide to conventional vsp acquisition: The

- Leading Edge, **8**, 34–39.
- Gonzalez, R. C., and R. E. Woods, 2002, Digital image processing. upper saddle river: J.: Prentice Hall.
- Goodfellow, I., Y. Bengio, and A. Courville, 2016, Deep learning: MIT press.
- Graves, A., N. Jaitly, and A.-r. Mohamed, 2013a, Hybrid speech recognition with deep bidirectional lstm: 2013 IEEE workshop on automatic speech recognition and understanding, IEEE, 273–278.
- Graves, A., A.-r. Mohamed, and G. Hinton, 2013b, Speech recognition with deep recurrent neural networks: 2013 IEEE international conference on acoustics, speech and signal processing, Ieee, 6645–6649.
- Graves, A., and J. Schmidhuber, 2005, Framewise phoneme classification with bidirectional lstm and other neural network architectures: Neural networks, **18**, 602–610.
- Gray, S. H., J. Etgen, J. Dellinger, and D. Whitmore, 2001, Seismic migration problems and solutions: Geophysics, **66**, 1622–1640.
- Gu, Z., L. Tao, H. Ren, R.-S. Wu, and J. Geng, 2022, Internal multiple elimination with an inverse-scattering theory guided deep neural network: Presented at the SEG/AAPG International Meeting for Applied Geoscience & Energy, OnePetro.
- Guitton, A., 2004, Amplitude and kinematic corrections of migrated images for nonunitary imaging operators: Geophysics, **69**, 1017–1024.
- Guo, R., J. Zhang, D. Liu, Y. Zhang, and D. Zhang, 2019, Application of bi-directional long short-term memory recurrent neural network for seismic impedance inversion: 81st EAGE Conference and Exhibition 2019, European Association of Geoscientists & Engineers, 1–5.
- Guo, S., and H. Wang, 2020, Image domain least-squares migration with a hessian matrix estimated by non-stationary matching filters: Journal of Geophysics and Engineering, **17**, 148–159.
- Hansen, L. K., and P. Salamon, 1990, Neural network ensembles: IEEE transactions on pattern analysis and machine intelligence, **12**, 993–1001.

- He, K., X. Zhang, S. Ren, and J. Sun, 2015, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification: Proceedings of the IEEE international conference on computer vision, 1026–1034.
- , 2016, Identity mappings in deep residual networks: European conference on computer vision, Springer, 630–645.
- Henley, D. C., and D. C. Lawton, 2021, Time-lapse detection using raypath interferometry: Geophysics, **86**, Q27–Q36.
- Herrmann, F. J., and X. Li, 2012, Efficient least-squares imaging with sparsity promotion and compressive sensing: Geophysical prospecting, **60**, 696–712.
- Herrmann, F. J., A. Siahkoohi, and G. Rizzuti, 2019, Learned imaging with constraints and uncertainty quantification: arXiv preprint arXiv:1909.06473.
- Higdon, R. L., 1991, Absorbing boundary conditions for elastic waves: Geophysics, **56**, 231–241.
- Hochreiter, S., and J. Schmidhuber, 1997, Long short-term memory: Neural computation, **9**, 1735–1780.
- Hu, J., G. T. Schuster, and P. A. Valasek, 2001, Poststack migration deconvolution: Geophysics, **66**, 939–952.
- Hu, W., 2016, An improved immersed boundary finite-difference method for seismic wave propagation modeling with arbitrary surface topography: Geophysics, **81**.
- Huang, S., and D. Trad, 2023, Convolutional neural-network-based reverse-time migration with multiple reflections: Sensors, **23**, 4012.
- Hussein, M., R. R. Stewart, D. Sacrey, D. H. Johnston, and J. Wu, 2021, Unsupervised machine learning for time-lapse seismic studies and reservoir monitoring: Interpretation, **9**, T791–T807.
- Innanen, K. A., D. Lawton, K. Hall, K. L. Bertram, M. B. Bertram, and H. C. Bland, 2019, Design and deployment of a prototype multicomponent distributed acoustic sensing loop array: SEG International Exposition and Annual Meeting, SEG, D023S007R002.

- Isaac, H. J., and D. C. Lawton, 2016, Brooks revisited: CREWES Research Report, **28**.
- Isaac, J. H., and D. C. Lawton, 2006, A case history of time-lapse 3d seismic surveys at cold lake, alberta, canada: *Geophysics*, **71**, B93–B99.
- , 2014, A case history of experimental time-lapse 3c 2d seismic reflection data for reservoir monitoring at cold lake, alberta, canada: *Interpretation*, **2**, SE47–SE54.
- Jia, Z., W. Lu, M. Zhang, and Y. Miao, 2019, Separating ground-roll from land seismic record via convolutional neural network: SEG 2018 Workshop: SEG Maximizing Asset Value Through Artificial Intelligence and Machine Learning, Beijing, China, 17-19 September 2018, Society of Exploration Geophysicists and the Chinese Geophysical Society, 60–63.
- Jiang, J., H. Ren, and M. Zhang, 2021, A convolutional autoencoder method for simultaneous seismic data reconstruction and denoising: *IEEE Geoscience and Remote Sensing Letters*, **19**, 1–5.
- Jiang, Z., J. Sheng, J. Yu, G. T. Schuster, and B. E. Hornby, 2007, Migration methods for imaging different-order multiples: *Geophysical Prospecting*, **55**, 1–19.
- Jordan, M., 1986, Serial order: a parallel distributed processing approach. technical report, june 1985-march 1986: Technical report, California Univ., San Diego, La Jolla (USA). Inst. for Cognitive Science.
- Jun, H., and Y. Cho, 2022, Repeatability enhancement of time-lapse seismic data via a convolutional autoencoder: *Geophysical Journal International*, **228**, 1150–1170.
- Karrenbach, M., S. Cole, A. Ridge, K. Boone, D. Kahn, J. Rich, K. Silver, and D. Langton, 2019, Fiber-optic distributed acoustic sensing of microseismicity, strain and temperature during hydraulic fracturing: *Geophysics*, **84**, D11–D23.
- Kaur, H., S. Fomel, and N. Pham, 2020a, Seismic ground-roll noise attenuation using deep learning: *Geophysical Prospecting*, **68**, 2064–2077.
- Kaur, H., N. Pham, and S. Fomel, 2020b, Improving the resolution of migrated images by approximating the inverse hessian using deep learning: *Geophysics*, **85**, WA173–WA183.
- , 2020c, Separating primaries and multiples using hyperbolic radon transform with deep

- learning, *in* SEG Technical Program Expanded Abstracts 2020: Society of Exploration Geophysicists, 1496–1500.
- Kearey, P., and M. Brooks, 1991, An introduction to geophysical exploration.
- Kolkman-Quinn, B. J., and D. C. Lawton, 2022, Detection threshold of a shallow co2 plume with vsp data from the cami field research station: Presented at the SEG/AAPG International Meeting for Applied Geoscience & Energy, OnePetro.
- Koster, K., P. Gabriels, M. Hartung, J. Verbeek, G. Deinum, and R. Staples, 2000, Time-lapse seismic surveys in the north sea and their business impact: *The Leading Edge*, **19**, 286–293.
- Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012, Imagenet classification with deep convolutional neural networks: *Advances in neural information processing systems*, **25**, 1097–1105.
- Krogh, A., and J. Vedelsby, 1994, Neural network ensembles, cross validation, and active learning: *Advances in neural information processing systems*, **7**.
- Larner, K., R. Chambers, M. Yang, W. Lynn, and W. Wai, 1983, Coherent noise in marine seismic data: *Geophysics*, **48**, 854–886.
- Lawton, D. C., J. Dongas, K. Osadetz, A. Saeedfar, and M. Macquet, 2019, Development and analysis of a geostatic model for shallow co2 injection at the field research station, southern alberta, canada: *Geophysics and Geosequestration*. Cambridge University Press, Cambridge, **280**, 296.
- LeCun, Y., Y. Bengio, and G. Hinton, 2015, Deep learning: *nature*, **521**, 436–444.
- LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner, 1998, Gradient-based learning applied to document recognition: *Proceedings of the IEEE*, **86**, 2278–2324.
- Levin, S. A., 1984, Principle of reverse-time migration: *Geophysics*, **49**, 581–583.
- Li, D., S. Peng, Y. Guo, Y. Lu, and X. Cui, 2021, Co2 storage monitoring based on time-lapse seismic data via deep learning: *International Journal of Greenhouse Gas Control*, **108**, 103336.

- Li, Y., and T. Alkhalifah, 2022, Target-oriented time-lapse elastic full-waveform inversion constrained by deep learning-based prior model: *IEEE Transactions on Geoscience and Remote Sensing*, **60**, 1–12.
- Li, Z., Z. Li, P. Wang, and M. Zhang, 2017, Reverse time migration of multiples based on different-order multiple separation: *Geophysics*, **82**, S19–S29.
- Lines, L., A. Bourgeois, and J. Covey, 1984, Traveltime inversion of offset vertical seismic profiles; a feasibility study: *Geophysics*, **49**, 250–264.
- Lines, L. R., and R. T. Newrick, 2004, *Fundamentals of geophysical interpretation*.
- Liu, N., L. Wu, J. Wang, H. Wu, J. Gao, and D. Wang, 2022, Seismic data reconstruction via wavelet-based residual deep learning: *IEEE Transactions on Geoscience and Remote Sensing*, **60**, 1–13.
- Liu, Y., X. Chang, D. Jin, R. He, H. Sun, and Y. Zheng, 2011, Reverse time migration of multiples for subsalt imaging: *Geophysics*, **76**, WB209–WB216.
- Liu, Z., Y. Chen, and G. Schuster, 2020, Deep convolutional neural network and sparse least-squares migration: *Geophysics*, **85**, WA241–WA253.
- Lu, S., 2021, Migration using sea surface-related multiples: Challenges and opportunities: *Geophysics*, **86**, WC11–WC19.
- Lu, S., N. Whitmore, A. Valenciano, and N. Chemingui, 2011, Imaging of primaries and multiples with 3d seam synthetic: Presented at the 2011 SEG Annual Meeting, OnePetro.
- 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.
- Lumley, D. E., 2001, Time-lapse seismic reservoir monitoring: *Geophysics*, **66**, 50–53.
- Macquet, M., D. Lawton, K. Osadetz, G. Maidment, M. Bertram, K. Hall, B. Kolkman-Quinn, J. M. Parra, F. Race, G. Savard, et al., 2022, Overview of carbon management canada’s pilot-scale co2 injection site for developing and testing monitoring technologies for carbon capture and storage, and methane detection: *CSEG Recorder*, **47**, 1–27.

- Malcolm, A. E., B. Ursin, and M. V. de Hoop, 2008, Subsalt imaging with internal multiples, *in* SEG Technical Program Expanded Abstracts 2008: Society of Exploration Geophysicists, 2461–2465.
- Mallat, S. G., 1989, A theory for multiresolution signal decomposition: the wavelet representation: IEEE transactions on pattern analysis and machine intelligence, **11**, 674–693.
- Margrave, G. F., 1998, Theory of nonstationary linear filtering in the fourier domain with application to time-variant filtering: Geophysics, **63**, 244–259.
- Margrave, G. F., and R. J. Ferguson, 1999, Wavefield extrapolation by nonstationary phase shift: Geophysics, **64**, 1067–1078.
- McMechan, G. A., 1983, Migration by extrapolation of time-dependent boundary values: Geophysical prospecting, **31**, 413–420.
- Meddins, R., 2000, Introduction to digital signal processing: Newnes.
- Mestayer, J., B. Cox, P. Wills, D. Kiyashchenko, J. Lopez, M. Costello, S. Bourne, G. Ugueto, R. Lupton, G. Solano, et al., 2011, Field trials of distributed acoustic sensing for geophysical monitoring, *in* Seg technical program expanded abstracts 2011: Society of Exploration Geophysicists, 4253–4257.
- Moya, A., and K. Irikura, 2010, Inversion of a velocity model using artificial neural networks: Computers & geosciences, **36**, 1474–1483.
- Mustafa, A., M. Alfarraj, and G. AlRegib, 2021, Joint learning for spatial context-based seismic inversion of multiple data sets for improved generalizability and robustness: Geophysics, **86**, O37–O48.
- Nemeth, T., C. Wu, and G. T. Schuster, 1999, Least-squares migration of incomplete reflection data: Geophysics, **64**, 208–221.
- O’Shea, K., and R. Nash, 2015, An introduction to convolutional neural networks: arXiv preprint arXiv:1511.08458.
- Pennington, W. D., 2000, “do no harm!”—seismic petrophysical aspects of time-lapse monitoring: Presented at the 2000 SEG Annual Meeting, OnePetro.

- Pevzner, R., S. Glubokovskikh, R. Isaenkov, P. Shashkin, K. Tertyshnikov, S. Yavuz, B. Gurevich, J. Correa, T. Wood, and B. Freifeld, 2022, Monitoring subsurface changes by tracking direct-wave amplitudes and traveltimes in continuous distributed acoustic sensor vsp datamonitoring by traveltime and amplitude: *Geophysics*, **87**, A1–A6.
- Pham, N., and X. Wu, 2019, Missing sonic log prediction using convolutional long short-term memory: Presented at the SEG International Exposition and Annual Meeting, OnePetro.
- Piriaux, J., and B. Lombard, 2001, A new interface method for hyperbolic problems with discontinuous coefficients: One-dimensional acoustic example: *Journal of Computational Physics*, **168**, 227–248.
- Poole, T. L., A. Curtis, J. O. Robertsson, and D.-J. van Manen, 2010, Deconvolution imaging conditions and cross-talk suppression: *Geophysics*, **75**, W1–W12.
- Richardson, A., 2018, Seismic full-waveform inversion using deep learning tools and techniques: arXiv preprint arXiv:1801.07232.
- Rickett, J., and D. Lumley, 2001, Cross-equalization data processing for time-lapse seismic reservoir monitoring: A case study from the gulf of mexico: *Geophysics*, **66**, 1015–1025.
- Rioul, O., and M. Vetterli, 1991, Wavelets and signal processing: *IEEE signal processing magazine*, **8**, 14–38.
- Ronneberger, O., P. Fischer, and T. Brox, 2015, U-net: Convolutional networks for biomedical image segmentation: *International Conference on Medical image computing and computer-assisted intervention*, Springer, 234–241.
- Roy, P., X. Zhu, and W. Fei, 2020, Machine learning assisted seismic inversion: Presented at the SEG International Exposition and Annual Meeting, OnePetro.
- Rumelhart, D. E., G. E. Hinton, and R. J. Williams, 1985, Learning internal representations by error propagation: Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.
- Sava, P., and I. Vlad, 2011, Wide-azimuth angle gathers for wave-equation migration: *Geophysics*, **76**, S131–S141.

- Schleicher, J., P. Hubral, M. Tygel, and M. S. Jaya, 1997, Minimum apertures and fresnel zones in migration and demigration: *Geophysics*, **62**, 183–194.
- Schulte, B., 2012, Overview on the fundamentals of imaging: *CSEG Recorder*, **37**, 40.
- Schuster, G. T., 2002, Joint migration of primaries and multiples by a semi-natural green’s functions: *Utah Tomography and Modeling/Migration Consortium Annual Report*, 141–164.
- , 2003, Imaging the most bounce out of multiples: 65th EAGE Conference & Exhibition, European Association of Geoscientists & Engineers, cp–141.
- , 2020, *Seismic imaging, overview*: Springer International Publishing.
- Schuster, M., and K. K. Paliwal, 1997, Bidirectional recurrent neural networks: *IEEE transactions on Signal Processing*, **45**, 2673–2681.
- Sengbush, R., 1983, Multiple reflections, *in* *Seismic Exploration Methods*: Springer, 103–123.
- Seydoux, L., R. Balestrieri, P. Poli, M. d. Hoop, M. Campillo, and R. Baraniuk, 2020, Clustering earthquake signals and background noises in continuous seismic data with unsupervised deep learning: *Nature communications*, **11**, 3972.
- Sheriff, R. E., and L. P. Geldart, 1995, *Exploration seismology*: Cambridge university press.
- Siahkoohi, A., 2022, *Deep generative models for solving geophysical inverse problems*: PhD thesis, Georgia Institute of Technology.
- Siahkoohi, A., M. Louboutin, and F. J. Herrmann, 2019a, The importance of transfer learning in seismic modeling and imaging: *Geophysics*, **84**, A47–A52.
- Siahkoohi, A., D. J. Verschuur, and F. J. Herrmann, 2019b, Surface-related multiple elimination with deep learning: Presented at the SEG International Exposition and Annual Meeting, OnePetro.
- Sochacki, J., R. Kubichek, J. George, W. Fletcher, and S. Smithson, 1987, Absorbing boundary conditions and surface waves: *Geophysics*, **52**, 60–71.
- Sun, J., 2000, Limited-aperture migration: *Geophysics*, **65**, 584–595.
- Sun, J., K. A. Innanen, and C. Huang, 2021, Physics-guided deep learning for seismic inver-

- sion 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.
- Suraj, A. A., M. Francis, T. Kavya, and T. Nirmal, 2014, Discrete wavelet transform based image fusion and de-noising in fpga: *Journal of Electrical Systems and Information Technology*, **1**, 72–81.
- Torres, K., and M. Sacchi, 2021, Deep learning based least-squares reverse-time migration: First International Meeting for Applied Geoscience & Energy, Society of Exploration Geophysicists, 2709–2713.
- , 2022, Least-squares reverse time migration via deep learning-based updating operators: *Geophysics*, **87**, 1–80.
- Trad, D., 2022, Gpu applications for modelling, imaging, inversion and machine learning.: Presented at the Geoconvention 2022.
- Tu, N., and F. J. Herrmann, 2015, Fast imaging with surface-related multiples by sparse inversion: *Geophysical Journal International*, **201**, 304–317.
- Valenciano, A. A., and B. Biondi, 2003, 2-d deconvolution imaging condition for shot-profile migration, *in* SEG Technical Program Expanded Abstracts 2003: Society of Exploration Geophysicists, 1059–1062.
- Vamaraju, J., J. Vila, M. Araya-Polo, D. Datta, M. Sidahmed, and M. K. Sen, 2021, Mini-batch 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 preprint arXiv:2104.01626.
- Verschuur, D. J., A. Berkhout, and C. Wapenaar, 1992, Adaptive surface-related multiple elimination: *Geophysics*, **57**, 1166–1177.
- Versteeg, R., 1994, The marmousi experience: Velocity model determination on a synthetic complex data set: *The Leading Edge*, **13**, 927–936.

- Wang, X.-Y., H.-Y. Yang, and Z.-K. Fu, 2010, A new wavelet-based image denoising using undecimated discrete wavelet transform and least squares support vector machine: *Expert Systems with Applications*, **37**, 7040–7049.
- Wang, Y., and D. C. Lawton, 2022a, Acoustic and exact elastic impedance variations during co2 injection at the cami.frs: CREWES Research Report, **34**.
- , 2022b, Time-lapse attenuation-attribute variations during co2 injection using das vsp data from the cami.frs: CREWES Research Report, **34**.
- Wang, Y., and I. B. Morozov, 2020, Time-lapse acoustic impedance variations during CO2 injection in Weyburn oilfield, canada: *Geophysics*, **85**, M1–M13.
- Wang, Y., Y. Zheng, Q. Xue, X. Chang, T. W. Fei, and Y. Luo, 2017, Reverse time migration of multiples: Reducing migration artifacts using the wavefield decomposition imaging condition: *Geophysics*, **82**, S307–S314.
- Wang, Z., 1997, Feasibility of time-lapse seismic reservoir monitoring: The physical basis: *The Leading Edge*, **16**, 1327–1330.
- Wapenaar, K., and J. Van Ijsseldijk, 2021, Employing internal multiples in time-lapse seismic monitoring, using the Marchenko method: 82nd EAGE Annual Conference & Exhibition, European Association of Geoscientists & Engineers, 1–5.
- Watanabe, T., S. Shimizu, E. Asakawa, and T. Matsuoka, 2004, Differential waveform tomography for time-lapse crosswell seismic data with application to gas hydrate production monitoring, *in* SEG Technical Program Expanded Abstracts 2004: Society of Exploration Geophysicists, 2323–2326.
- Weglein, A. B., 1995, Multiple attenuation: Recent advances and the road ahead, *in* SEG Technical Program Expanded Abstracts 1995: Society of Exploration Geophysicists, 1492–1495.
- Weglein, A. B., and W. H. Dragoset, 2005, Multiple attenuation: Society of Exploration Geophysicists Tulsa.
- Whitmore, N., A. Valenciano, W. Sollner, and S. Lu, 2010, Imaging of primaries and multi-

- ples using a dual-sensor towed streamer, *in* SEG Technical Program Expanded Abstracts 2010: Society of Exploration Geophysicists, 3187–3192.
- Whitmore, N. D., 1983, Iterative depth migration by backward time propagation, *in* SEG Technical Program Expanded Abstracts 1983: Society of Exploration Geophysicists, 382–385.
- Wiggins, J. W., 1988, Attenuation of complex water-bottom multiples by wave-equation-based prediction and subtraction: *Geophysics*, **53**, 1527–1539.
- Wrona, T., I. Pan, R. L. Gawthorpe, and H. Fossen, 2018, Seismic facies analysis using machine learning: *Geophysics*, **83**, O83–O95.
- Wu, B., G. Yao, J.-J. Cao, D. Wu, X. Li, and N.-C. Liu, 2022a, Huber inversion-based reverse-time migration with de-primary imaging condition and curvelet-domain sparse constraint: *Petroleum Science*.
- Wu, D., Q. Li, X. Zhang, J. Li, and H. Wu, 2018, Least-squares reverse time migration with adaptive moment estimation method: Presented at the 2018 SEG International Exposition and Annual Meeting, OnePetro.
- Wu, D., Y. Wang, J. Cao, N. V. da Silva, and G. Yao, 2021, Least-squares reverse-time migration with sparsity constraints: *Journal of Geophysics and Engineering*, **18**, 304–316.
- Wu, H.-C., W.-L. Fan, C.-S. Tsai, and J. J.-C. Ying, 2022b, An image authentication and recovery system based on discrete wavelet transform and convolutional neural networks: *Multimedia Tools and Applications*, **81**, 19351–19375.
- Wyatt, K. D., 1981, Synthetic vertical seismic profile: *Geophysics*, **46**, 880–891.
- Xiong, Z., K. Ramchandran, and M. T. Orchard, 1998, Wavelet packet image coding using space-frequency quantization: *IEEE transactions on image processing*, **7**, 892–898.
- Yao, G., B. Wu, N. V. da Silva, H. A. Debens, D. Wu, and J. Cao, 2022, Least-squares reverse time migration with a multiplicative cauchy constraint: *Geophysics*, **87**, S151–S167.
- Yilmaz, Ö., 2001, *Seismic data analysis: Processing, inversion, and interpretation of seismic data*: Society of exploration geophysicists.

- Yoon, D., Z. Yeoh, and J. Byun, 2020, Seismic data reconstruction using deep bidirectional long short-term memory with skip connections: *IEEE Geoscience and Remote Sensing Letters*, **18**, 1298–1302.
- Yoon, K., K. J. Marfurt, and W. Starr, 2004, Challenges in reverse-time migration: SEG International Exposition and Annual Meeting, SEG, SEG–2004.
- Yu, J., J. Hu, G. T. Schuster, and R. Estill, 2006, Prestack migration deconvolution: *Geophysics*, **71**, S53–S62.
- Yuan, C., X. Zhang, X. Jia, and J. Zhang, 2020, Time-lapse velocity imaging via deep learning: *Geophysical Journal International*, **220**, 1228–1241.
- Zhang, D., and G. T. Schuster, 2014, Least-squares reverse time migration of multiples: *Geophysics*, **79**, S11–S21.
- Zhang, R., R. Ghosh, M. K. Sen, and S. Srinivasan, 2013, Time-lapse surface seismic inversion with thin bed resolution for monitoring co2 sequestration: A case study from cranfield, mississippi: *International Journal of Greenhouse Gas Control*, **18**, 430–438.
- Zhang, W., J. Gao, Y. Cheng, Z. Li, X. Jiang, and J. Zhu, 2022, Deep-learning for accelerating prestack correlative least-squares reverse time migration: *Journal of Applied Geophysics*, **200**, 104645.
- Zhang, Y., Y. Liu, X. Liu, and X. Zhou, 2020, Reverse time migration using water-bottom-related multiples: *Geophysical Prospecting*, **68**, 446–465.
- Zhong, Z., A. Y. Sun, and X. Wu, 2020, Inversion of time-lapse seismic reservoir monitoring data using cyclegan: a deep learning-based approach for estimating dynamic reservoir property changes: *Journal of Geophysical Research: Solid Earth*, **125**, e2019JB018408.
- Zhou, R., L. Huang, and J. Rutledge, 2010, Microseismic event location for monitoring co 2 injection using double-difference tomography: *The Leading Edge*, **29**, 208–214.
- Zhou, Z.-H., 2021, Ensemble learning, *in* *Machine learning*: Springer, 181–210.