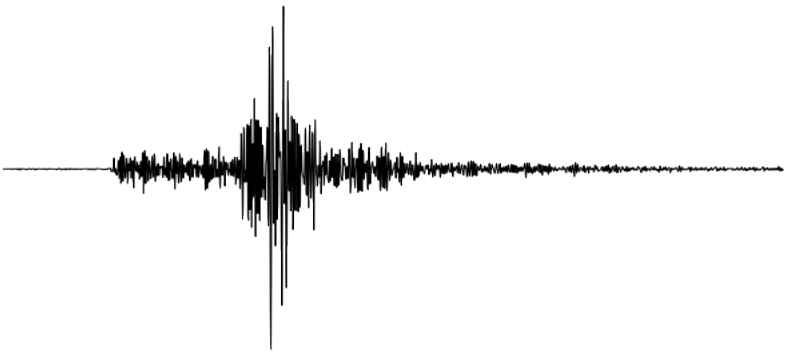
National Technical University of Athens
School of Electrical and Computer Engineering
MSc Data Science & Machine Learning

Denoising Distributed Acoustic Sensing data with Self-Supervised Deep Learning

Diploma Thesis

of

NIKOS MANIATIS



Supervisor: Konstantinos Karantzalos
Associate Professor, NTUA

Athens, July 2022

National Technical University of Athens
School of Electrical and Computer Engineering
MSc Data Science & Machine Learning

Denoising Distributed Acoustic Sensing data with Self-Supervised Deep Learning

Diploma Thesis

of

NIKOS MANIATIS

Supervisor: Konstantinos Karantzalos
Associate Professor, NTUA

Approved by the examination committee on 15th July, 2022.

(Signature)	*(Signature)*	*(Signature)*
.....	.....	.....
Konstantinos Karantzalos	Paraskevi Nomikou	Maria Pateraki
Associate Professor, NTUA	Associate Professor, UOA	Assistant Professor, NTUA

Athens, July 2022

National Technical University of Athens
School of Electrical and Computer Engineering
MSc Data Science & Machine Learning

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

(Signature)

.....
Nikos Maniatis
15th July, 2022

Abstract

Distributed Acoustic Sensing (DAS) is an emerging technology utilizing fiber optic cables for vibration measurements with various applications such as seismic signal analysis, pipeline monitoring, traffic monitoring (roads, railways and trains). Its ease of use and versatility, lie on the fact that it can be deployed in harsh and dangerous environments, such as submarine, glaciated or volcanic and due to its ability to turn existing commercial fiber optic cables into sensor arrays with temporal sampling of up to 1 thousand samples per second, and with a spatial sampling in the order of meters. However, new environments also come with new challenges as each new environment has the ability to introduce noise in various types, lowering the quality of the measurements and thus impeding with the data analysis workflows. In this work, we explore the possibility of removing incoherent noise from DAS recordings, utilizing the concept of J-invariance and modern self-supervised deep learning methods, without making assumptions regarding the noise characteristics. We apply this method to both synthetic and real world DAS data, from four different experiments, one of which took place in a volcanic environment in Iceland, and the rest come from three separate submarine DAS recordings in Greece. The results show exceptional denoising capability and great promise to be incorporated into seismological analysis data workflows, when the noise is incoherent.

Περίληψη

Το Distributed Acoustic Sensing (DAS) είναι μια αναδυόμενη τεχνολογία που χρησιμοποιεί καλώδια οπτικών ινών για λήψη δονητικών μετρήσεων που βρίσκει ποικίλες εφαρμογές όπως στην ανάλυση σημάτων σεισμικών δονήσεων, παρακολούθηση αγωγών αλλά και της κίνησης σε δρόμους, σιδηρόδρομους και άλλα. Η ευκολία της χρήσης και η ευελιξία της μεθόδου έγκειται στο γεγονός ότι μπορεί να τοποθετείται σε σκληρά και επικίνδυνα περιβάλλοντα, όπως υποθαλάσσια, ηφαιστειακά ή και πολικά, λόγω της δυνατότητας να μετατρέπει ήδη υπάρχουσες για εμπορικούς λόγους οπτικές ίνες σε συστοιχίες αισθητήρων με δειγματοληψία στο χρόνο μέχρι 1000 δείγματα το δευτερόλεπτο, και με χωρική δειγματοληψία της τάξης των μερικών μέτρων. Παρόλα αυτά, το κάθε περιβάλλον έχει τις δικές του προκλήσεις καθώς κάθε περιβάλλον έχει τη δυνατότητα εισαγωγής θορύβου στα δεδομένα με διαφορετικές μορφές, επηρεάζοντας την ποιότητα των μετρήσεων και παρεμποδίζοντας έτσι την διαδικασία ανάλυσης των δεδομένων. Σε αυτή την διπλωματική εργασία, εξερευνούμε τη δυνατότητα αφαίρεσης μη συνεκτικού θορύβου από μετρήσεις DAS, χρησιμοποιώντας την έννοια της $\mathcal{J}$-αναλλοιότητας και μοντέρνων τεχνικών ημι-επιβλεπόμενης βαθιάς μάθησης, χωρίς να προβούμε σε υποθέσεις για τα χαρακτηριστικά του θορύβου. Εφαρμόζουμε αυτή τη μέθοδο τόσο σε συνθετικά δεδομένα όσο και σε πραγματικά δεδομένα μετρήσεων DAS από 4 διαφορετικά πειράματα, ένα από τα οποία προέρχεται από ηφαιστειακό περιβάλλον της Ισλανδίας ενώ τα υπόλοιπα από 3 υποθαλάσσια καλώδια οπτικών ινών στην Ελλάδα. Τα αποτελέσματα δείχνουν αποτελεσματική αφαίρεση θορύβου και την καθιστούν πολλά υποσχόμενη να καθιερωθεί στην διαδικασία ανάλυσης σεισμολογικών δεδομένων, εφόσον ο θόρυβος δεν παρουσιάζει συνοχή στη δομή του.

to my parents

Acknowledgements

At first, I would like to express my sincere gratitude to my supervisor prof. Konstantinos Karantzalos, for his ever helpful feedback and for steering me in the right direction whenever he thought I needed it. I'm extremely grateful to PhD candidates Sara Klaasen and Ioannis Kakogeorgiou, for their constant support since day one, and their willingness to help. They were a core part of this thesis. Thanks to my family for their love and encouragement and for always believing in me not only throughout my studies, but my whole life. Without their support I wouldn't be where I am today. Finally, I must express my very profound gratitude to Celia, for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

Athens, July 2022

Nikos Maniatis

Contents

Abstract 1

Περίληψη 3

Acknowledgements 7

1 Introduction 17
- 1.1 English . . . 17
- 1.2 Ελληνικά . . . 19

2 Background 21
- 2.1 Distributed Acoustic Sensing . . . 21
- 2.2 Machine Learning . . . 23
- 2.3 Convolutional Neural Networks . . . 25
  - 2.3.1 Fully Connected Layers (Linear) . . . 25
  - 2.3.2 Activation or Non Linearity . . . 25
  - 2.3.3 Spatial Convolution . . . 26
  - 2.3.4 Spatial Pooling . . . 28
  - 2.3.5 UNET . . . 28
- 2.4 J-Invariant denoising . . . 29
  - 2.4.1 Simple Formulation . . . 29
  - 2.4.2 Mathematical Formulation . . . 31

3 Method 33
- 3.1 Model Architecture . . . 33
- 3.2 Data Acquisition . . . 35
  - 3.2.1 HCMR/NESTOR . . . 36
  - 3.2.2 Iceland . . . 37
  - 3.2.3 Santorini . . . 38
- 3.3 Synthetic Data Generation . . . 39
- 3.4 Training Procedure . . . 41

4 Results and Analysis 43
- 4.1 Ablation Study . . . 43
- 4.2 Results on synthetic data . . . 45
- 4.3 Results on real Distributed Acoustic Sensing (DAS) data . . . 48

4.3.1 HCMR/NESTOR 49
4.3.2 Iceland 51
4.3.3 Santorini 58
4.4 Comparison with traditional filtering methods 68

5 Discussion **73**
5.0.1 Reproducibility 73
5.0.2 Applications 74
5.0.3 Limitations & Future Work 75

Bibliography **81**

List of Abbreviations **83**

List of Figures

2.1 Rayleigh Scattering, image adapted from [1]. 22
2.2 Diagram showing the gauge length and the channel spacing, image adapted from [2]. 22
2.3 Plots of different activation functions. 27
2.4 The illustration of a spatial pooling operation in 2×2 regions by a stride of 2 in the high direction, and 2 in the width direction, without padding. . . . 27
2.5 Illustration of Max Pooling and Average Pooling Figure 2 above shows an example of max pooling operation and average pooling with a 2×2 pixel filter size from 4×4 pixel input. 28
2.6 U-net architecture. Blue boxes represent multi-channel feature maps, while while boxes represent copied feature maps. The arrows of different colors represent different operation. Picture coined from [3]. 29
2.7 Intuitive explanation of the concept of J-invariance. Image adapted from [4]. 30

3.1 Model Architecture. Image adapted from [4]. 33
3.2 Anti-aliased max-pooling. (Top) Pooling does not preserve shift-equivariance. It is functionally equivalent to densely-evaluated pooling, followed by sub-sampling. The latter ignores the Nyquist sampling theorem and loses shift-equivariance. (Bottom) We low-pass filter between the operations. This keeps the first operation, while anti-aliasing the appropriate signal. Anti-aliasing and subsampling can be combined into one operation, which we refer to as BlurPool. Image adapted from [5]. 34
3.3 Geographic location of the DAS cables HCMR and NESTOR. The inset shows the location of the region of interest within Greece. Image adopted from [4]. 37
3.4 Layout of the fiber-optic cable in Grímsvötn. (black line with numbers indicating distance in kilometers) deployed in the DAS-BúmmBúmm experiment in spring 2021. Locations of the research huts (GFUM) near one end of the cable and a GPS station at the other end are also shown, as are the years and approximate locations of previous fissure eruptions (orange and red). The site of Grímsvötn (red triangle) amid the Vatnajökull ice sheet in Iceland is indicated in the inset. Image taken from this article. 38

4.1 Results of the hyperparameter sweep 44
4.2 Parameter importance and correlation with validation loss 44
4.3 Training and Validation Loss Curves for training on synthetic data. 46

4.4 Synthetic examples of model performance. 46
4.5 Quantitative assessment of the model performance on synthetic data. . . . 47
4.6 Training and Validation loss curves for HCMR/NESTOR data. 49
4.7 J-invariant filtering results of HCMR data. 50
4.8 J-invariant filtering results of NESTOR data. 51
4.9 Wiggle-for-wiggle comparison along the HCMR cable. The waveforms correspond with panels a,b,c of 4.7. 52
4.10 J-invariant filtering results of Iceland data. 53
4.11 Wiggle-for-wiggle comparison along the Iceland cable. The waveforms correspond with panels a,b,c of 4.10. 53
4.12 Wiggle-for-wiggle comparison along the Iceland cable, on channel 121. . . 54
4.13 J-invariant filtering results of Iceland data. 55
4.14 Wiggle-for-wiggle comparison along the Iceland cable. The waveforms correspond with panels a,b,c of 4.13. 55
4.15 Wiggle-for-wiggle comparison along the Iceland cable, on channel 520. . . 56
4.16 J-invariant filtering results of Iceland data. 57
4.17 Wiggle-for-wiggle comparison along the Iceland cable. The waveforms correspond with panels a,b,c of 4.16. 57
4.18 J-invariant filtering results of Santorini data. 59
4.19 Moving coherence versus channels for Iceland and Santorini data. 60
4.20 Wiggle-for-wiggle comparison along the Santorini cable. The waveforms correspond with panels a,b,c of 4.18. 61
4.21 J-invariant filtering results of Santorini data. 61
4.22 Wiggle-for-wiggle comparison along the Santorini cable. The waveforms correspond with panels a,b,c of 4.21. 62
4.23 J-invariant filtering results of Santorini data. 63
4.24 Wiggle-for-wiggle comparison along the Santorini cable. The waveforms correspond with panels a,b,c of 4.23. 63
4.25 J-invariant filtering results of Santorini data. 64
4.26 Wiggle-for-wiggle comparison along the Santorini cable. The waveforms correspond with panels a,b,c of 4.25. 64
4.27 J-invariant filtering results of Santorini data. 65
4.28 Wiggle-for-wiggle comparison along the Santorini cable. The waveforms correspond with panels a,b,c of 4.27. 65
4.29 J-invariant filtering results of Santorini data. 66
4.30 Wiggle-for-wiggle comparison along the Santorini cable. The waveforms correspond with panels a,b,c of 4.29. 66
4.31 J-invariant filtering results of Santorini data. 67
4.32 Wiggle-for-wiggle comparison along the Santorini cable. The waveforms correspond with panels a,b,c of 4.31. 67
4.33 Comparison of our proposed method with two conventional image denoising methods on HCMR data. Event Corresponds with 4.7. 69

4.34 Comparison of our proposed method with two conventional image denoising methods on Santorini data. Event Corresponds with 4.29. 70
4.35 Time taken to process the file of figure 4.29. 70
4.36 Comparison of our proposed method with two conventional image denoising methods on Iceland data. Event Corresponds with 4.13. 71

List of Tables

3.1 Different DAS datasets and their parameters. 36
3.2 Catalogued events from the HCMR & NESTOR cables. 37

4.1 Final Set of Hyperparameters of the model 45
4.2 Information regarding the seismic events on Santorini DAS. 58

Chapter 1

Introduction

1.1 English

Distributed Acoustic Sensing (DAS) is an emerging technology that effectively transforms conventional fiber-optic cables into massive arrays of single component seismometers that enable the acquisition of dense, high-resolution datasets across tens of kilometers [1]. DAS systems send consecutive, coherent laser pulses down a fiber optic cable that can exist in a various range of environments. Due to naturally-occurring impurities within the core of the fiber, part of the light sent down the cable is backscattered and recorded back at the DAS unit, where the optical phase of the light is measured. Seismic waves impacting the cable induce compression/extension, causing a change in the optical phase of the backscattered light. An interferometric system in the DAS unit measures the optical phase changes of consecutive backscattered light profiles, which are proportional to changes in longitudinal strain along the fiber. Current DAS systems enable acquisition of strain rate at spatial samplings as small as 25 cm at rates in the tens of kHz range.

Deep learning, a branch of machine learning and artificial intelligence is nowadays considered as a core technology of today's "Fourth Industrial Revolution" [6]. Due to its learning capabilities from data, deep learning technology originated from artificial neural networks, has become a hot topic in the context of computing, and is widely applied in various application areas like healthcare, visual recognition, text analytics, cyber security, and many more. However, building an appropriate deep learning model is a challenging task, due to the dynamic nature and variations in real-world problems and data. Deep learning uses multiple layers to represent the abstractions of data to build computational models. While deep learning takes a long time to train a model due to a large number of parameters, it takes a short amount of time to run during testing as compared to other machine learning algorithms.

Combining these two relatively new fields to perform some kind of task, is no trivial matter, but the idea comes naturally as DAS produces large volumes of data, and machine learning methods (especially deep learning) benefit from that. A limited number of recent studies have explored leveraging Machine Learning approaches for analysis and processing of DAS ambient seismic noise datasets. In the majority of these studies, the main objective is event detection and classification, and the feature of interest is the complete seismic signal generated by a vehicle or person travelling past the array, which can be

differentiated from other signals of interest such as earthquakes, as they are more coherent. For example, the authors of [7] develop a combined ML approach based on feature extraction for vehicle detection, classification and speed estimation. On [8], the authors develop a convolutional neural network to automatically detect footstep signals in ambient seismic recordings from urban DAS arrays. In a similar approach, Huot and Biondi on [9] use a convolutional neural network to automatically detect car-generated seismic signals with the objective of removing them from the seismic recordings. In the majority of cases, these signals are complex and composed of a combination of useful seismic energy with other effects such as static deformation due to the vehicle load, optical effects originated by the recording instrument.

Another application of machine learning in DAS is event detection. Typically, this is done by non-learning algorithms, for example authors of [10] utilize the STA/LTA (Short Term Average/Long Term Average) algorithm, [11], where a short window of 0.3 sec length is divided by a long window of 60 sec length, and when this ratio surpasses a certain threshold then this is counted as an event. The authors of [12] try out different machine learning techniques to detect events while using images of the sensor signals and deep neural networks for pattern recognition. This requires a large amount of labelled data, that generally are not available as they require lots of human hours of labelling the data, hence pushing the need to design unsupervised of self-supervised learning techniques to circumvent this barrier.

DAS being a new approach of seismic data collection, also provides new perspectives and challenges with regard to nuisance signals (noise) that originate from instrumental, electronic, anthropogenic, or environmental sources. For example, submarine DAS recordings suffer from background noise coming from the coupling of the fiber optic cable to the sea floor, from passing ships, from currents and from various other sources. Since one often has little control on the exact placement of the cable, deployments are typically not optimised for the recording of specific signals of interest, enhancing the relative contribution of noise to the recordings. Noise affects all kinds of DAS applications, which is the reason we shifted our focus on adopting, re-applying and extending the work of [4], where the authors utilize the concept of $\mathcal{J}$-invariance and modern self-supervised methods to remove incoherent noise from DAS recordings.

The objective of this thesis is to re-implement the work of [4] in the PyTorch framework [13], as the original one was written in TensorFlow [14], reapply to a wide range of different DAS datasets and to explore further extensions. This master thesis is organized as follows: In chapter 2 we introduce all the necessary theoretical background regarding the the domains of deep learning and distributed acoustic sensing. In chapter 3 we deep-dive into the method we are about to follow as per the network architecture, the training procedure and we also describe the different datasets we used. In chapter 4 we present all the results from the different datasets, and also discuss the differences between them. Finally, in chapter 5 we discuss how one can reproduce the results on the same or in different DAS datasets, the limitations of our work, as well as several ideas for future improvements.

1.2 Ελληνικά

Το Distributed Acoustic Sensing (DAS) είναι μια αναδυόμενη τεχνολογία που αποτελεσματικά μετατρέπει συμβατικά καλώδια οπτικών ινών σε τεράστιες συστοιχίες αισθητήρων που επιτρέπουν την απόκτηση πυκνών συνόλων δεδομένων υψηλής ανάλυσης σε περιοχές δεκάδων χιλιομέτρων [1]. Τα συστήματα DAS στέλνουν διαδοχικούς, συνεκτικούς παλμούς λέιζερ σε ένα καλώδιο οπτικών ινών που μπορεί να υπάρχει σε διάφορα περιβάλλοντα. Λόγω φυσικών ατελειών και προσμίξεων στο εσωτερικό του καλωδίου, αλλάζει τοπικά ο δείκτης διάθλασης του υλικού και μέρος του φωτός που αποστέλλεται στο καλώδιο οπισθοσκεδάζεται και καταγράφεται στη μονάδα DAS, όπου μετράται η φάση του φωτός. Σεισμικά κύματα που προσκρούουν στο καλώδιο προκαλούν συμπίεση/επέκταση, προκαλώντας αλλαγή στην οπτική φάση του οπισθοσκεδασμένου φωτός. Ένα σύστημα στη μονάδα DAS μετρά τις αλλαγές οπτικής φάσης διαδοχικών προφίλ οπισθοσκέδασης του φωτός, τα οποία είναι ανάλογα σε αλλαγές στη διαμήκη τάση κατά μήκος της ίνας. Τα τρέχοντα συστήματα DAS επιτρέπουν την απόκτηση του ρυθμού παραμόρφωσης σε χωρικές δειγματοληψίες έως και 25 cm σε ρυθμούς στην περιοχή δεκάδων kHz.

Η βαθιά μάθηση, ένας κλάδος της μηχανικής μάθησης και της τεχνητής νοημοσύνης, θεωρείται στις μέρες μας η βασική τεχνολογία της σημερινής «Τέταρτης Βιομηχανικής Επανάστασης» [6]. Λόγω της ικανότητας μάθησης από δεδομένα, η τεχνολογία βαθιάς μάθησης προέρχεται από τεχνητά νευρικά δίκτυα, και έχει γίνει ένα επίκαιρο θέμα στο πλαίσιο της θεωρίας πληροφορικής και εφαρμόζεται ευρέως σε διάφορους τομείς όπως η υγειονομική περίθαλψη, η οπτική αναγνώριση (π.χ. αντικειμένων), η ανάλυση κειμένου, η ασφάλεια στον κυβερνοχώρο, και πολλά άλλα. Ωστόσο, η οικοδόμηση ενός κατάλληλου μοντέλου βαθιάς μάθησης είναι ένα δύσκολο έργο, λόγω της δυναμικής φύσης και των παραλλαγών στα προβλήματα και τα δεδομένα του πραγματικού κόσμου. Η βαθιά μάθηση χρησιμοποιεί πολλαπλά επίπεδα νευρωνικών δικτύων για να αναπαραστήσει με αφαιρετικό τρόπο τα δεδομένα, για τη δημιουργία υπολογιστικών μοντέλων. Ενώ η βαθιά εκμάθηση χρειάζεται πολύ χρόνο για την εκπαίδευση ενός μοντέλου λόγω του μεγάλου αριθμού παραμέτρων, χρειάζεται μικρό χρονικό διάστημα για να εκτελεστεί κατά τη διάρκεια της δοκιμής σε σύγκριση με άλλους αλγορίθμους μηχανικής μάθησης.

Ο συνδυασμός δύο σχετικά καινούριων επιστημονικών πεδίων προς επίτευξη κάποιου σκοπού, δεν είναι κάτι τετριμμένο, όμως η ιδέα μας έρχεται φυσικά καθώς το DAS παράγει μεγάλο όγκο δεδομένων, ενώ οι μέθοδοι μηχανικής μάθησης (και ειδικά βαθιάς μάθησης) οφελούνται από αυτά. Ένας περιορισμένος αριθμός από πρόσφατες μελέτες εξερευνούν τη χρήση μηχανικής μάθησης για ανάλυση και επεξεργασία από δεδομένα θορύβου υποβάθρου από DAS. Στην πλειοψηφία αυτών των μελετών, το κύριο αντικείμενο είναι η ανίχνευση και ταξινόμηση γεγονότων (π.χ. σεισμών) όπου το σήμα ενδιαφέροντος είναι αυτό που παράγεται από κάποιο άτομο ή όχημα που περνάει πάνω από την συστοιχία DAS, το οποίο μπορεί να διακριθεί σε σχέση με άλλα σήματα ενδιαφέροντος όπως σεισμοί, καθώς έχουν μεγαλύτερη συνοχή. Για παράδειγμα, οι συγγραφείς του [7], παρουσιάζουν μια συνδιαστική προσέγγιση με χρήση μηχανικής μάθησης βασισμένη σε εξαγωγή χαρακτηριστικών για εντοπισμό οχημάτων, ταξινόμηση αλλά και προσέγγιση της ταχύτητας. Στο [8], οι συγγραφείς ανέπτυξαν ένα συνελτικτικό νευρωνικό δίκτυο που ανιχνεύει αυτόματα σήματα από βηματισμούς (από

ανθρώπους) από μετρήσεις DAS σε αστικό περιβάλλον. Σε μια παρόμοια προσέγγιση, οι συγγραφείς στο [9] ανέπτυξαν ένα συνελικτικό νευρωνικό δίκτυο που ανιχνεύει σεισμικά σήματα που παράγονται από αυτοκίνητα, με κύριο σκοπό την αφαίρεσή τους από τις μετρήσεις.

Μια άλλη εφαρμογή της μηχανικής μάθησης στο DAS είναι η ανίχνευση συμβάντων. Συνήθως αυτό γίνεται από απλούς αλγόριθμους, όχι αλγόριθμους μηχανικής μάθησης, όπως για παράδειγμα ο STA/LTA[11] (Short Term Average / Long Term Average) όπου παίρνουμε τον κυλιόμενο μέσο όρο για ένα μικρό χρονικό παράθυρο, π.χ. 0,3 δευτερόλεπτα και το διαιρούμε με τον κυλιόμενο μέσο όρο για ένα πιο μεγάλο παράθυρο π.χ. μήκους 60 δευτερολέπτνω, και όταν αυτός ο λόγος υπερβαίνει ένα καθορισμένο όριο, τότε αυτό προσμετράται ως ένα συμβάν. Οι συγγραφείς του [12] δοκιμάζουν διάφορες τεχνικές μηχανικής μάθησης για τον εντοπισμό συμβάντων αντιμετωπίζοντας τα σήματα DAS ως εικόνες, χρησιμοποιώντας βαθιά νευρωνικά δίκτυα για αναγνώριση προτύπων. Αυτό απαιτεί την ύπαρξη μεγάλου όγκου επισημασμένων δεδομένων τα οποία γενικά δεν είναι διαθέσιμα καθώς απαιτούν πολλές ώρες ανθρώπινης εργασίας για να παρασκευαστούν, ωθόντας μας έτσι στην ανάγκη να στραφούμε σε τεχνικές μη επιβλεπόμενης ή αυτό-επιβλεπόμενης μάθησης για να ξεπεράσουμε αυτό το εμπόδιο.

Το DAS είναι μια νέα προσέγγιση συλλογής σεισμικών δεδομένων, που μαζί με τις νέες προοπτικές, παρέχει και προκλήσεις όσον αφορά τα σήματα θορύβου που μπορεί να προέρχονται από ανθρωπογενείς, περιβαλλοντικές, ηλεκτρονικές πηγές ή και από τα όργανα των μετρήσεων. Για παράδειγμα, μετρήσεις από υποβρύχιο DAS πάσχουν από θόρυβο που προέρχεται από το περιβάλλον όπως για παράδειγμα από τη σύζευξη του καλωδίου οπτικής ίνας με τον πυθμένα της θάλασσας, από διερχόμενα πλοία, από θαλάσσια ρεύματα και από διάφορες άλλες πηγές. Καθώς κατά τη διεξαγωγή του πειράματος έχουμε πολύ μικρό έλεγχο στην ακριβή τοποθεσία του καλωδίου, οι τοποθετήσεις συνήθως δεν είναι βελτιστοποιημένες για την καταγραφή σημάτων που μας ενδιαφέρουν, ενισχύοντας έτσι την σχετική συνεισφορά του θορύβου στις μετρήσεις. Ο θόρυβος επηρεάζει όλες τις εφαρμογές του DAS, γεγονός το οποίο μας έστρεψε την προσοχή στο να υιοθετήσουμε, να επανα-εφαρμόσουμε αλλά και να επεκτείνουμε το έργο των συγγραφέων στο [4], όπου χρησιμοποιούν την έννοια της $\mathcal{J}$-αναλλοιότητας καθώς και μοντέρνων μεθόδων αυτό-επιβλεπόμενης μάθησης για την αφαίρεση του μη συνεκτικού θορύβου από μετρήσεις DAS.

Το αντικείμενο αυτής της διπλωματικής εργασίας είναι η υλοποίηση της δουλειάς των συγγραφέων στο [4] στη δομή PyTorch [13], αφού η αρχική δουλειά ήταν υλοποιημένη σε Tensorflow [14], η εφαρμογή σε μια μεγάλη γκάμα από διαφορετικά σύνολα δεδομένων DAS και η εξερεύνηση μελλοντικών επεκτάσεων. Η διπλωματική εργασία είναι οργανωμένη ως εξής: Στο κεφάλαιο 2 εισάγουμε όλο το απαραίτητο θεωρητικό υπόβαθρο σχετικά με τους κλάδους της βαθιάς μάθησης και του DAS. Στο κεφάλαιο 3 εμβαθύνουμε στην μέθοδο που πρόκειται να ακολουθήσουμε όσον αφορά την αρχιτεκτονική του νευρωνικού δικτύου, την διαδικασία εκπαίδευσης αλλά και περιγράφουμε τα διαφορετικά σύνολα δεδομένων που χρησιμοποιήθηκαν. Στο κεφάλαιο 4 παραθέτουμε και αναλύουμε τα αποτελέσματα από όλα τα σύνολα δεδομένων καθώς και τις διαφορές αυτών. Τέλος, στο κεφάλαιο 5 συζητάμε πως μπορεί κανείς να αναπαράξει αυτά τα αποτελέσματα, στα ίδια ή σε διαφορετικά σύνολα δεδομένων DAS, τα όρια της μεθόδου που ακολουθήσαμε καθώς και διάφορες ιδέες για μελλοντικές βελτιώσεις.

Chapter 2

Background

2.1 Distributed Acoustic Sensing

Distributed acoustic sensing (DAS) is an emerging geophysical method that uses an optoelectronic instrument connected to an optical fiber to measure strain along the fiber, effectively repurposing it as a seismic array. [15] The optoelectronic instrument, called an interrogator unit (IU), sends laser pulses into the optical fiber and measures subtle phase shifts in Rayleigh scattered light at each point along the fiber, as measured in the time or frequency domain. In this way, the strain field acting on a fiber coupled to Earth can be sampled at meter-scale spatial resolution over tens of linear fiber kilometers.

DAS enables scientists to test hypotheses using high-density and large aperture experiments. Deploying traditional sensing systems (nodal arrays, geophones, seismometers) is not always logistically feasible due to space constraints, land access issues, extreme temperatures, or power limitations. By contrast, DAS is used to study a variety of geoscientific processes and locations (urban areas, offshore, glaciers, wells, volcanoes).

For the past decade, DAS has been increasingly utilized in exploration geophysics related to the oil and gas industry, geothermal energy, and CO2 sequestration ([16], [17] and [18]). Much of this effort has focused on vertical seismic profile imaging [19], time-lapse imaging [18], and continuous microseismicity monitoring [20], as well as some on geomechanical strain monitoring [21].

Over the past few years, applications have grown in near-surface geophysics for engineering, infrastructure, and environmental studies, particularly those requiring long-term monitoring. Note that researchers are increasingly collecting relatively high sample rate data for long durations, leading to data volumes per experiment that are much larger than traditional seismic experiments. For example, an experiment can generate up to 1 TB of data per day, depending on the sampling rate. Before DAS may be more widely utilized by geoscientists, a variety of community-scale challenges and needs must be addressed by the Earth science community.

DAS refers to any method in which optical interferometry is applied to laser light traveling inside of an optical fiber to measure strain or strain rate at many positions along the fiber. DAS is also referred to as distributed vibration sensing, coherent optical time-domain reflectometry (OTDR), coherent optical frequency-domain reflectometry (OFDR), or phase-sensitive OTDR. The DAS instrument is referred to as an optoelectronic IU and

has a field form factor that fits on a workbench. All IUs generate, send, and receive laser pulses to and from an attached fiber sensor. Laser light commonly falls in the near-infrared wavelength range (~1,550 nm) and is pulsed, but some DAS instruments use continuous and swept-frequency light sources. Refractive index heterogeneities in the fiber's silica glass core cause Rayleigh scattering [15] as pictured in 2.1. In (a) we see

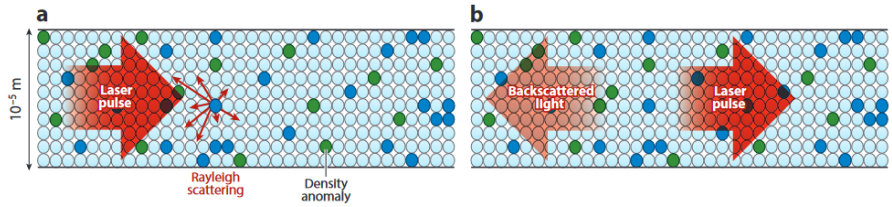


Figure 2.1. *Rayleigh Scattering, image adapted from [1].*

a Rayleigh scattering event at sites of index of refraction change (blue/green) inside the core of a single-mode optical fiber laser, while in (b) we can see that most light continues to propagate down the fiber, but distributed acoustic sensing utilizes the backscattered energy. Rayleigh scattering is well characterized by the telecommunications industry because of its transmission impact, which can cause a drop in signal strength of 0.15 – 0.20 *dB/km* for near-infrared wavelengths such as those commonly used for DAS. Only a small amount of the scattered light returns to the IU.

There are many ways to implement DAS. A common technique uses an IU to repeatedly inject laser pulses into an optical fiber and analyze the phase of the Rayleigh backscattered signal with OTDR. OTDR assumes the mean fiber path follows a simple out-and-back trajectory between the IU and Rayleigh scattering point. A known two-way travel time of light in the fiber provides the necessary information to map each subset of the backscattered profile in time to a subset of linear fiber distance. OTDR multiplexes the time-continuous backscattering into an array of independent signals returning from consecutive fiber segments, called gauges.

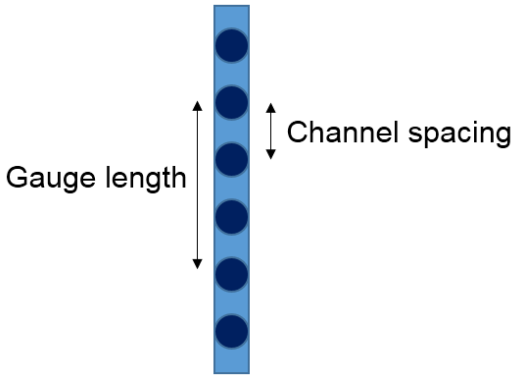


Figure 2.2. *Diagram showing the gauge length and the channel spacing, image adapted from [2].*

The gauge length is the corresponding spatial increment of fiber sampled by each

signal, typically about 1 to 40 m long [22]. The spatial axis of a DAS data set is reported in channels, often equal or subset from the gauge length; (figure 2.2) however, spatial resolution is fixed by the gauge length.

Backscattering phases, as opposed to amplitudes, are proportional to the change in path length over the gauge length. DAS recordings capture dynamic strains at acoustic frequencies ($f > 1,000\ Hz$) and broadband seismic frequencies ($f = 0.001 - 1,000\ Hz$), and they have also shown promise for studies at ultra-low frequencies that would classically be called static strain or geodesy (periods of hours to weeks) but that relax away over long enough timescales ([23], [24]).

Inside the IU, optical interferometry is applied to the backscattered signal to measure phase or phase rate. The exact units depend on the particular DAS approach. In [25] was described the first distributed optical strain sensing instrument. According to this approach, a pair of laser pulses separated in frequency (f_1 and f_2) are launched one after the other and the backscattered signal is measured at the beat frequency ($\Delta f = |f_1 - f_2|$). The temporal separation of the pulses results in a backscattered signal that combines light from location x_1 (first pulse) with location x_2 (second pulse) separated by the gauge length. The backscattered signal phase $\Delta\Phi$ is linearly related to the gauge strain ϵ_{xx}, following:

$$\epsilon_{xx}(t, x) = \frac{\lambda}{4\pi n_c x_g \psi}\Delta\Phi$$

where t and x locate the axial strain measurement along the fiber axis ($+x$ direction), λ is the frequency used for measurement (beat frequency here), n_c and ψ are the refractive index and Pockels coefficient of the single mode fiber glass ($\psi = 0.79$), and x_g is the gauge length. This assumes $\Delta\Phi$ is related only to the fiber's dynamic mechanical strain. Optical dispersion effects are easily considered for multi-frequency setups or ignored for single-frequency ones. Thermo-optical effects and thermal strain are ignored because the deformation measurement timescale in seismology is much less than the thermal variation timescales; however, these must be accounted for in low-frequency DAS strain measurements or faster thermal cycles.

2.2 Machine Learning

Machine learning is an area of applied statistics that studies algorithms which computer systems can use, in order to estimate functions from data, without explicit instructions. These learning algorithms, can loosely be classified as **supervised** or **unsupervised**, based on the kind of data we use. This is not exhaustive though, as the method we use in this thesis is classified as **self-supervised** learning, which learns from unlabeled sample data. It can be regarded as an intermediate form between supervised and unsupervised learning.

Supervised learning algorithms use data points that each one contains features and is associated with a label. That means that every sample is tagged with the answer that we are trying to predict. So for example, labeled image data set with animals would tell the model which images are dogs,cats,etc. Given a specific training set, a supervised

learning algorithm try to learn the function that maps the features of the input to the label. In math notation, we denote the input features as x^i and the label as y^i. A data set $\mathcal{D}$ contains many data points, so $\mathcal{D} = \{(x^i, y^i)\}$ where $i = 1, ..., n$. Also, we denote X the input space, Y the output space. Our goal is to learn a function $f : X \mapsto Y$ using the dataset $\mathcal{D}$ in order for $f(x)$ to correctly determine the label y of x. Based on the possible values of output space, the supervised learning problems can be grouped in classification and regression problems.

Classification problems have categorical output values. Examples of classification problems are email spam detection [26] or image classification [27]. Regression problems have output values that are real numbers. Examples of regression problems are house price prediction [28] or prediction of wind energy production [29].

Unsupervised learning algorithms use data points without an explicit target associated with them. The machine learning algorithm attempt to identify interesting structural properties and extract useful features of the data. Two of the most important unsupervised techniques are clustering and autoencoders. Clustering algorithms try to discover clusters of similar samples. Examples of such algorithms are k-means [30] and DBSCAN [31].

As mentioned before, **self-supervised** methods can be regarded as an intermediate form between supervised and unsupervised learning. At the same time, however, it does not require the explicit use of labeled input-output pairs. Instead, correlations, metadata embedded in the data, or domain knowledge present in the input are implicitly and autonomously extracted from the data. These supervisory signals, generated from the data, can then be used for training. Self-supervised Learning is similar to unsupervised learning in that it does not require labels in the sample data. Unlike unsupervised learning, however, learning is not done using inherent data structures. We also note that autoencoders, such as the UNet [3] we are using, are in practice a self-supervised pretext task (referring to training a model for a task other than what it will actually be trained and used for), but due to historical reasons sometimes it is still referred as unsupervised learning. Examples of applications of self-supervised learning is denoising of images [32], speech recognition [33] or language processing where the models can be used to translate texts or answer questions, among other things [34].

Nowadays, deep learning has achieved major breakthroughs in many fields, including computer vision ([27], [35]) and natural language processing ([34], [36]). These achievements became possible mainly due to the following reasons:

- High availability of massive data sets.
- Increased performance of computer processors, and especially GPUs-TPUs.
- More complex and deeper neural network architectures.

But before deep learning dominates the field, there was a major drawback in the classical approaches that were used. The feature representations from the input , had to be manually extracted , based on the task. For example, in a computer vision task, the engineer had to manually use specific algorithms that extracts edges,corners,blobs,etc. from

images. This unfortunately put up barriers to anyone want to pursue machine learning, as it requires a lot of domain knowledge. In contrast, artificial neural networks have automated the feature extraction procedure. They are composed from many neurons, that are connected to each other, inspired by the functionality of the brain, and can construct features from raw input. Now, it is not a prerequisite to be an expert in the domain of the problem, in order to use machine learning techniques. Proof of this argument, is in fact this master thesis. Deep learning brought a fundamental change, and enabled more and more people to get involved in the area.

2.3 Convolutional Neural Networks

In order to introduce our neural network architecture in the next section, we first need to define the different components neural networks are made of, mainly the different layers, activation functions and how they can be used.

2.3.1 Fully Connected Layers (Linear)

Mathematically, we can think of a linear layer as a function which applies a linear transformation on a vector input of dimension $\mathcal{I}$ and output a vector of dimension $\mathcal{O}$. Usually the layer has a bias parameter.

$$y_i = \sum_{j=1}^{\mathcal{I}} (A_{ij} x_j) + b_i$$

The linear layer is motivated by the basic computational unit of the brain called neuron. Approximately 86 billion neurons can be found in the human nervous system and they are connected with approximately $10^{14} - 10^{15}$ synapses. Each neuron receives input signals from its dendrites and produces output signal along its axon. The linear layer is a simplification of a group of neuron having their dendrites connected to the same inputs. Usually an activation function, such as sigmoid, is used to mimic the 1-0 impulse carried away from the cell body and also to add non linearity. However we consider here that the activation function is the identity function that output real values.

2.3.2 Activation or Non Linearity

The capacity of the neural networks to approximate any functions, especially non-convex, is directly the result of the non-linear activation functions. Every kind of activation function, takes a vector and performs a certain fixed point-wise operation on it. Here are some common activation functions, also seen on figure 2.3.

- **Sigmoid**: The Sigmoid non-linearity has the following mathematical form:

$$y = \sigma(x) = 1/(1 + e^{-x})$$

  It takes a real value and squashes it between 0 and 1. However, when the neuron's activation saturates at either tail of 0 or 1, the gradient at these regions is almost

zero. Thus, the backpropagation algorithm fail at modifying its parameters and the parameters of the preceding neural layers.

- **Hyperbolic Tangent**: The TanH non-linearity has the following mathematical form:

$$y = 2 \cdot \sigma(2x) - 1$$

It squashes a real-valued number between -1 and 1. However it has the same drawback than the sigmoid.

- **Rectified Linear Unit**: The ReLU [37] has the following mathematical form:

$$y = \max(0, x)$$

The ReLU has become very popular in the last few years, because it was found to greatly accelerate the convergence of stochastic gradient descent compared to the sigmoid/tanh functions due to its linear non-saturating form (e.g. a factor of 6 in [27]). In fact, it does not suffer from the vanishing or exploding gradient. An other advantage is that it involves cheap operations compared to the expensive exponentials. However, the ReLU removes all the negative information and thus appears not suited for all datasets and architectures.

- **Sigmoid Linear Unit**: The SiLU [38] function, also known as the swish function, is the one we used in our architecture and has the following mathematical form:

$$y = x \cdot \sigma(x)$$

where $\sigma(x)$ is the sigmoid function we defined earlier. The SiLU activation functions tends to work better than ReLU on deeper models across a number of challenging datasets. For example, simply replacing ReLUs with Swish units improves top-1 classification accuracy on ImageNet by 0.9% for Mobile NASNet-A and 0.6% for Inception-ResNet-v2, according to [38]. The simplicity of Swish and its similarity to ReLU make it easy for practitioners to replace ReLUs with Swish units in any neural network.

2.3.3 Spatial Convolution

Regular Neural Networks, only made of linear and activation layers, do not scale well to full images. For instance, images of size $3 \times 224 \times 224$ (3 color channels, 224 wide, 224 high) would necessitate a first linear layer having $3 \cdot 224 \cdot 224 + 1 = 150,129$ parameters for a single neuron (e.g. output). Spatial convolution layers take advantage of the fact that their input (e.g. images or feature maps) exhibits many spatial relationships. In fact, neighboring pixels should not be affected by their location within image. Thus, a convolutional layer learns a set of N_k filters $F = f_1, ..., f_{N_k}$, which are convolved spatially

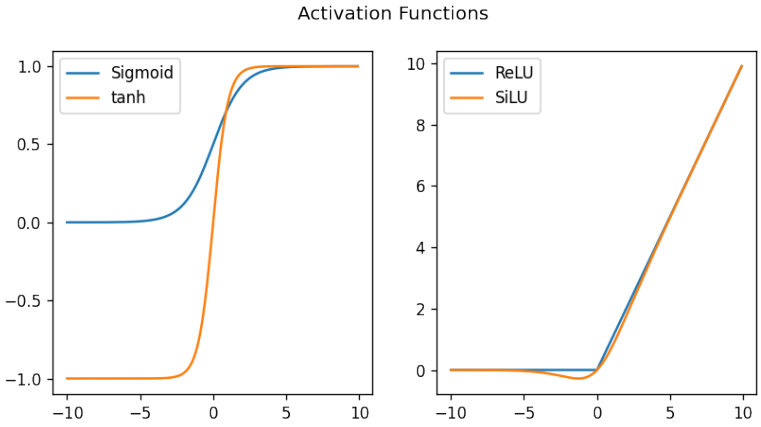


Figure 2.3. *Plots of different activation functions.*

with input image x, to produce a set of N_k 2D feature maps z:

$$z_k = f_k * x$$

where $*$ is the convolution operator. When the filter correlates well with a region of the input image, the response in the corresponding feature map location is strong. Unlike conventional linear layer, weights are shared over the entire image reducing the number of parameters per response and equivariance is learned (i.e. an object shifted in the input image will simply shift the corresponding responses in a similar way). Also, a fully connected layer can be seen as a convolutional layer with filter of sizes $1 \times 1 \times inputSize$. It is important to highlight that a spatial convolution is not defined by the spatial size of the input feature maps (e.g. wide and high), neither by the size of the output feature maps, but by the number of filters (e.g. number of output channels), the properties of its filters (e.g. number of input channels, wide, high) and the properties of the convolution (e.g. padding, stride).

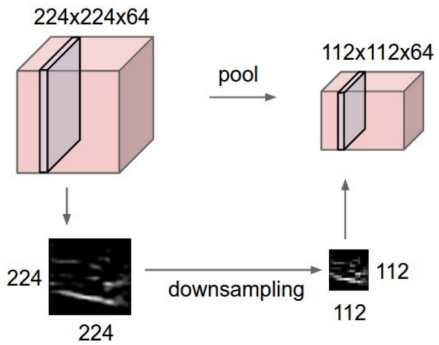


Figure 2.4. *The illustration of a spatial pooling operation in 2×2 regions by a stride of 2 in the high direction, and 2 in the width direction, without padding.*

2.3.4 Spatial Pooling

In Convolutional Neural Networks, a pooling layer is typically present to provide invariance to slightly different input images and to reduce the dimension of the feature maps (e.g. wide, high):

$$p_R = P_{i \in R}(z_i)$$

where P is a pooling function over the region of pixels R. Common pooling functions are average and max pooling (figure 2.5). Max pooling is preferred as it avoids cancellation of negative elements and prevents blurring of the activations and gradients throughout the network since the gradient is placed in a single location during backpropagation. The spatial pooling layer is defined by its aggregation function, the high and width dimensions of the area where it is applied, and the properties of the convolution (e.g. padding, stride).

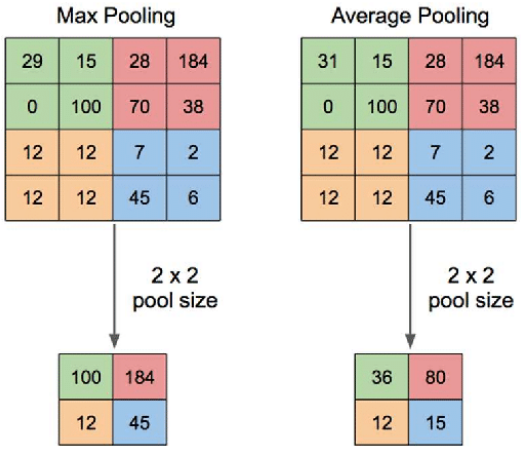


Figure 2.5. *Illustration of Max Pooling and Average Pooling Figure 2 above shows an example of max pooling operation and average pooling with a* 2×2 *pixel filter size from* 4×4 *pixel input.*

2.3.5 UNET

U-net was originally invented and first used for biomedical image segmentation [3]. Its architecture (figure 2.6) can be broadly thought of as an encoder network followed by a decoder network. Unlike classification where the end result of the the deep network is the only important thing, semantic segmentation not only requires discrimination at pixel level but also a mechanism to project the discriminative features learnt at different stages of the encoder onto the pixel space. Upsampling in CNN might be new when used to classification and object detection architectures, but the idea is fairly simple. The intuition is that we would like to restore the condensed feature map to the original size of the input image, therefore we expand the feature dimensions. Upsampling is also referred to as transposed convolution, upconvolution, or deconvolution. There are a few ways of upsampling such as Nearest Neighbor, Bilinear Interpolation, and Transposed

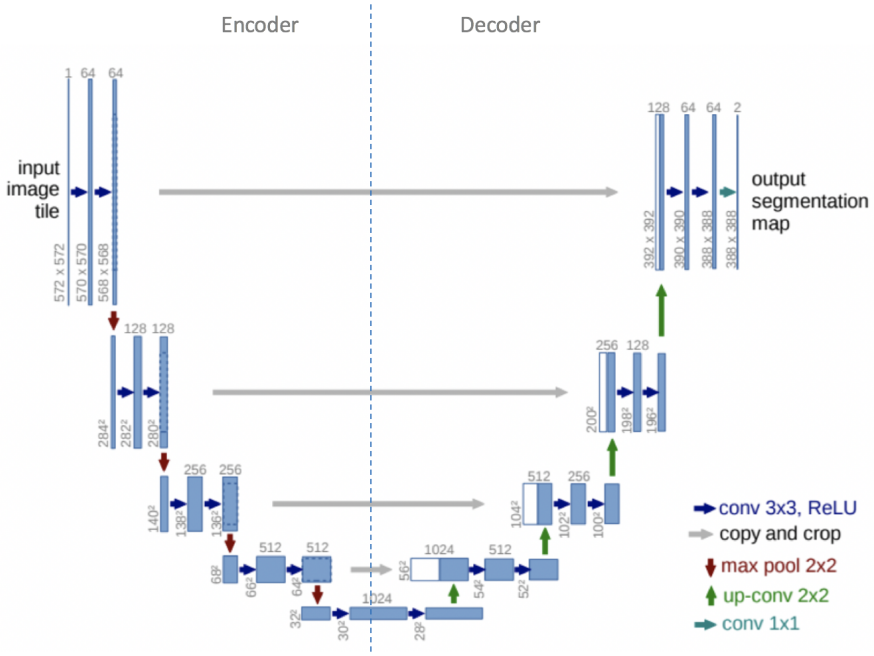


Figure 2.6. *U-net architecture. Blue boxes represent multi-channel feature maps, while while boxes represent copied feature maps. The arrows of different colors represent different operation. Picture coined from [3].*

Convolution from simplest to more complex. Specifically, we would like to upsample it to meet the same size with the corresponding concatenation blocks from the left. You may see the gray and green arrows, where we concatenate two feature maps together.

The main contribution of U-Net in this sense is that while upsampling in the network we are also concatenating the higher resolution feature maps from the encoder network with the upsampled features in order to better learn representations with following convolutions. Since upsampling is a sparse operation we need a good prior from earlier stages to better represent the localization.

In summary, unlike classification where the end result of the very deep network is the only important thing, semantic segmentation not only requires discrimination at pixel level but also a mechanism to project the discriminative features learnt at different stages of the encoder onto the pixel space.

2.4 J-Invariant denoising

2.4.1 Simple Formulation

We would often like to reconstruct a signal from high dimensional measurements that are corrupted, undersampled, or otherwise noisy. Devices like high-resolution cameras, electron microscopes or Distributed Acoustic Sensing IUs are capable of producing measurements in the thousands to millions of feature dimensions. But when these devices are pushed to their limits, taking videos with ultra-fast frame rates at very low-illumination, probing individual molecules with electron microscopes or getting thousands of strain

measurements per seconds, each individual feature can become quite noisy.

Nevertheless, the objects being studied are often very structured and the values of different features are highly correlated. Speaking loosely, if the "latent dimension" of the space of objects under study is much lower than the dimension of the measurement, it may be possible to implicitly learn that structure, denoise the measurements, and recover the signal without any prior knowledge of the signal or the noise.

Traditional denoising methods each exploit a property of the noise, such as Gaussianity, or structure in the signal, such as spatiotemporal smoothness, self-similarity, or having low-rank. The performance of these methods is limited by the accuracy of their assumptions. For example, if the data are genuinely not low rank, then a low rank model will fit it poorly. This requires prior knowledge of the signal structure, which limits application to new domains and modalities.

J-invariant denoising was first introduced in [32] by Batson and Royer, and then later applied to DAS recording in [4], and the only assumption it makes is that the noise exhibits statistical independence across different dimensions of the measurement, while the true signal exhibits some correlation. For a broad class of functions ("$\mathcal{J}$-invariant") it is then possible to estimate the performance of a denoiser from noisy data alone. This allows us to calibrate $\mathcal{J}$-invariant versions of any parameterised denoising algorithm, from the single hyperparameter of a median filter to the millions of weights of a deep neural network.

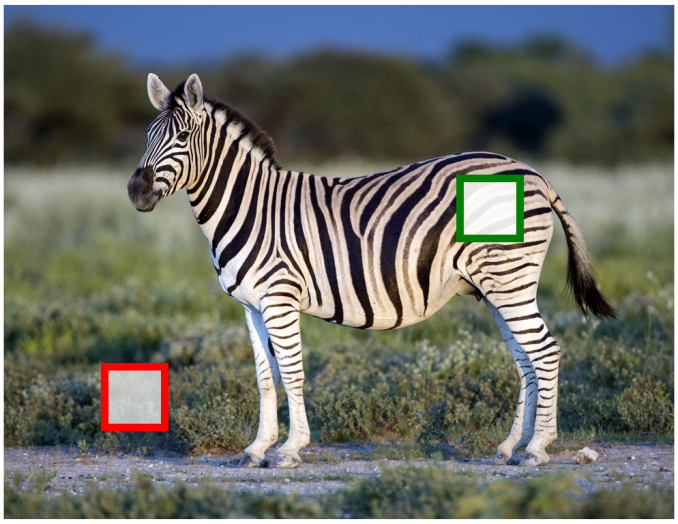


Figure 2.7. *Intuitive explanation of the concept of J-invariance. Image adapted from [4].*

In order to better understand intuitively the concept of J-invariance, we consider the image on figure 2.7, featuring a coherent signal such as the stripes on the zebra (green patch), where the contents of the patch have been removed. If the signal of interest exhibits sufficiently long-range coherence, with respect to the size of the patch, then the contents of the signal within the patch can be accurately predicted. On the contrary, uncorrelated and incoherent features, that exhibit coherence only locally, with respect to the size of the patch, are not informative for predicting the contents of the removed patch

(red patch). Any kind of learner, be it human or artificial, faced with the task to recover the hidden data from the red patch, will therefore only be able to use coherent signals in the input data. This means that the contents of the patch are not immediately required to perform a given action on the patch. This approach from [32], circumvents the need of clean training data.

2.4.2 Mathematical Formulation

The authors from [32] propose to train an image denoiser g using a single noisy image y that results from an unknown clean image x such that $x = \mathbb{E}[y|x]$, where $\mathbb{E}[\cdot]$ denotes the expectation operator. The denoiser derivation relies on the assumption that g is J-invariant.

Definition: *Let $\mathcal{J}$ be a partition of the feature space, and let $J \in \mathcal{J}$. We write z_J for z restricted to its features in J. We say that g is $\mathcal{J}$-invariant if, $\forall\ J \in \mathcal{J}$ and $\forall\ z$, $g(z)_J$ does not depend on the values of z_J.*

Since the term feature space is somehow ambiguous, we will explain that concept. As DAS denoising is closely related to image denoising, the term "image" can be used for both cases. The feature space of the image is the set of all possible combinations of features, while a feature is a piece of information about the content of an image; typically about whether a certain region of the image has certain properties. Features may be specific structures in the image such as points, edges or objects. Since we can construct new features from existing ones, feature spaces can be infinite, so that is why we select a partition $\mathcal{J}$ of the feature space, and we work on that partition.

We then take a piece J of that partition, this for example can be the patch on figure 2.7, and some z_j, for example the different pixel values, inside the patch. We say that our denoiser g is $\mathcal{J}$-invariant if for all the possible pieces in the partition of the feature space, and for all the features z, the output of the denoiser $g(z)_J$ (on J) when given a feature z as input, does not depend on the actual values of z_J. In other words, the output of the denoiser inside the patch, when given the rest of the picture, does not depend on the contents of the patch.

We can then assume that the minimiser of $\mathbb{E}\|g(y) - y\|^2$ over the space of $\mathcal{J}$-invariant functions, let it be $g*$, is able to verify that: $g^*(y)_J = \mathbb{E}\left[x_J \mid y_{J^c}\right]$, $\forall\ J \in \mathcal{J}$, where J^c is the complement of J. This result, when compared to the optimal denoiser, $\mathbb{E}\left[x_J \mid y\right]$, clearly shows the couplings between the independence of the noise, the spatial coherence of the clean image and the partition $\mathcal{J}$.

In our case, the set of $\mathcal{J}$-invariant functions g is explored with a neural network, which can be seen as a function f_∂ with lots of parameters ∂, which are the various weights, biases and so on. The training process is the tuning of all the parameters ∂ to minimize the objective function that is defined on our problem. The neural network f_∂ is made $\mathcal{J}$-invariant by defining it as:

$$g(\cdot) = \sum_{J \in \mathcal{J}} \Pi_J \left(f_\partial \left(\Pi_{J^c}(\cdot)\right)\right)$$

where $\Pi_A(z)$ is the projection operator that does not modify the values of the elements of z in A but sets the elements in A^c to zero (being $\mathbb{E}[z]$ in our case). This operator can be mathematically formulated as:

$$\Pi_A(z) = \begin{cases} 1, & z \in A \\ 0, & z \in A^c \end{cases}$$

From the previous equation, it is implied that:

$$g\left(\Pi_{J^c}(\cdot)\right) = \Pi_J\left(f_\partial\left(\Pi_{J^c}(\cdot)\right)\right)$$

meaning that the minimisation of $\|g(y) - y\|^2$ with respect to ∂ can be performed efficiently by training the neural network f_∂ with a suitable learning objective, which we will define later on.

In [32], the authors focus primarily on single-image denoising applications, with a brief exploration of multi-image denoising using Deep Learning architectures. In the present work, we apply the concept of J-invariance to batches of DAS data (which are analogous to images). As we will demonstrate, performing the training on a sufficiently diverse set of DAS data enables direct application of the trained model on new data without retraining. This is explored as we test our model in datasets it was not trained on (Iceland, Santorini), with satisfactory results even without retraining.

Chapter 3

Method

3.1 Model Architecture

The neural network architecture we used for our denoising objective was based on the commonly used UNet architecture [3], that we also introduced on subsection 3.2.3 featuring 4 blocks of downsampling and convolutional layers in the encoder, and 4 blocks of up-sampling, concatenation, and convolutional layers in the decoder. The original architecture can be seen on figure 2.6 while ours on figure 3.1. Each one of the 4 downsampling blocks features an anti-aliased downsampling layer with a stride of 4 along the time axis, no downsampling was performed on the channel axis, followed by two convolutional layers with a number of filters that is doubled for each block. Normally, these encoder operations are reversed in the decoder by first bilinear upsampling with a factor 4, concatenating the output of the diametrically opposed block, and two convolutional layers with a number of filters that is halved for each block. The output layer is a convolutional layer with a single filter and a linear activation. All convolutional layers (except the output one) feature a kernel of size 3×5 (DAS channels $\times$ time samples). For weight initialization, the default PyTorch method was used, the Kaiming method [39] which is an initialization method for neural networks that takes into account the non-linearity of activation functions.

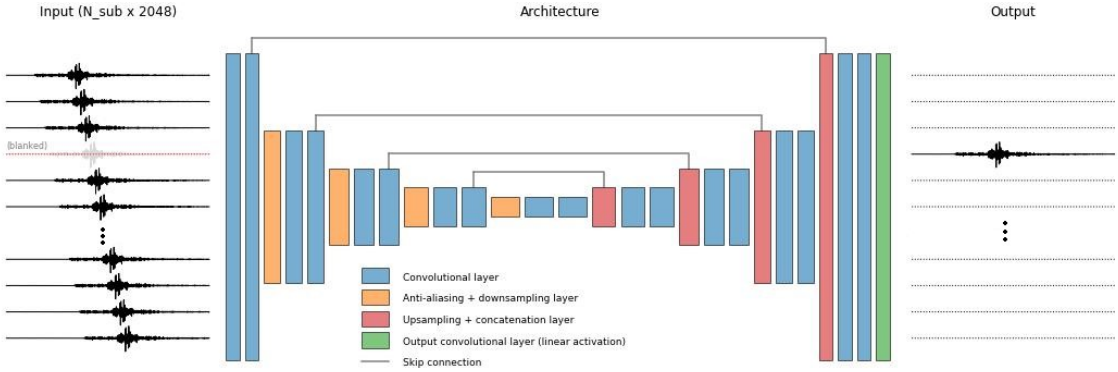


Figure 3.1. *Model Architecture. Image adapted from [4].*

We decided to make some modifications in the original UNet architecture in order to make it more efficient towards our specific learning objective. The first, was to change

the original convolutional layers with anti-aliasing ones. As the PyTorch [13] framework, in which our work was implemented, does not provide these components out of the box, we added this kind of layers by using an implementation of [5]. According to [5], modern convolutional networks are not shift invariant, as small input shifts or translations can cause drastic changes in the output. Commonly used downsampling methods, such as max-pooling, strided-convolution, and average pooling, ignore the sampling theorem. The well known signal processing fix is anti-aliasing by low-pass filtering before downsampling, which when used right managed to achieve increased accuracy in ImageNet [27] classification for a wide range of architectures, and also better generalization in terms of stability and robustness to input corruptions. The latter is why we decided to adopt this approach and indeed worked as promised.

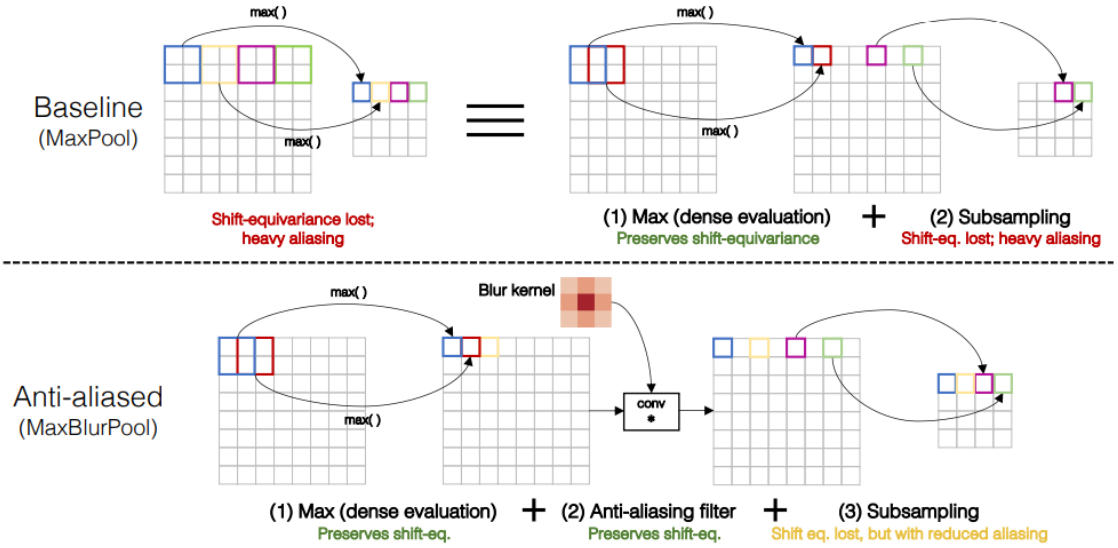


Figure 3.2. *Anti-aliased max-pooling. (Top) Pooling does not preserve shift-equivariance. It is functionally equivalent to densely-evaluated pooling, followed by subsampling. The latter ignores the Nyquist sampling theorem and loses shift-equivariance. (Bottom) We low-pass filter between the operations. This keeps the first operation, while anti-aliasing the appropriate signal. Anti-aliasing and subsampling can be combined into one operation, which we refer to as BlurPool. Image adapted from [5].*

An example of anti-aliasing a common spatial resolution reducing method, in order for it not to break shift-invariance, is shown in figure 3.2, where we perform anti-aliasing of the common Max Pooling method. The authors from [5] noticed that the Max Pooling operation is equivalent to a Max operation, followed by a subsampling operation, and while this loses shift-equivariance, by keeping the first operation and anti-aliasing with a low-pass filter the appropriate signal they manage to reduce aliasing effects. This procedure can be seen on the bottom of figure 3.2. Then, anti-aliasing and subsampling can be combined into one operation, which is referred to as BlurPool. We adopted this operation in our UNet, instead of the conventional Max Pooling layer.

Another modification we decided to implement, was to replace the ReLU [37] activation function, with SiLU [38], for all convolutional layers except the output layer. According

to [38], the SiLU (or swish) activation function tends to work better than ReLU, on deeper models across a number of challenging datasets. For example, simply replacing ReLUs with Swish units improves top-1 classification accuracy on ImageNet [27] by 0.9% for Mobile NASNet-A [40] and 0.6% for Inception-ResNet-v2 [41]. The simplicity of Swish and its similarity to ReLU make it easy for practitioners to replace ReLUs with Swish units in any neural network.

Every mini-batch of the input consists of a sample y_k of N_{sub} neighbouring channels, which after hyperparameter tuning was selected at 11 channels, corresponding with $N_{sub}\times x_g = 11\times x_g$ meters of cable, where x_g is the gauge length that corresponds to each dataset and with 41 s of recordings at 50 Hz (2048 samples). An entire single DAS waveform is defined as J_k, chosen at random out of the 11 samples. Therefore, the remaining 10 samples can be denoted as J_k^c. In order to achieve $\mathcal{J}$-invariance to the model, we need an projection operator such as the $\Pi_A(J_k)$ we mentioned earlier. Intuitively, such an operation can be imagined as the one that hides the contents of the patch, while if applied to the complement of the image reveals the contents of the patch and hides the rest of the image.

The mathematical formulation of this operator when applied to the input y_k is:

$$u_k = \Pi_{J_k^c}(y_k)$$

so, if y_k consists of 11 waveforms, then u_k consists of 11 waveforms out of which 1 is blanked, as can be seen in figure 3.1. Then, u_k is input into the neural network. Since the neural network output $f_\partial(u_k)$ is essentially an image that consists of 11 full waveforms, we also apply the same operator to $f_\partial(u_k)$ to the complement, giving:

$$v_k = \Pi_{J_k}(f_\partial(u_k))$$

so that v_k consists of 1 full waveform and 10 blank ones. This way, we can compute the MSE loss between the output v_k and $\Pi_{J_k}(y_k)$. Then in accordance to the theory we laid out in subsection 2.4.2, we define the loss $\mathcal{L}$ computer over a mini-batch $\{y_k\}$ as:

$$\mathcal{L}(\{y_k\}) = \frac{1}{|K|}\sum_{k\in K}\|u_k - \Pi_{J_k}(y_k)\|^2$$

where $|K|$ is the batch size. We choose the ADAM optimizer [42] to minimize this loss function by backpropagating the loss and updating the weights and biases of the neural network. We initially also considered the Stochastic Gradient Descent algorithm [43], but ADAM was more effective as we will see in the results section.

3.2 Data Acquisition

In this section we will describe the different environments in which each experiment took place, what parameters were used to obtain the DAS recordings as well as differences and similarities between the datasets. On table 3.1 we can see the the technical

parameters of the four different datasets, that describe the experiment as per their spatial and temporal sampling, and the length of the fiber optic cable.

Dataset Name	Length (km)	Number of Channels	Gauge Length (m)	Channel Spacing (m)	Sampling Rate (Hz)
HCMR	13.21	688	19.2	19.2	166.66
NESTOR	26.21	1365	19.2	19.2	200
Iceland	13.82	1728	10	8	200
Santorini	44.54	5568	10	8	1000

Table 3.1. *Different DAS datasets and their parameters.*

For every dataset, we took some preprocessing steps in order to be consistent as every dataset contains DAS recordings that were recorded under different circumstances. We filter the data in a 1-10 Hz frequency band and downsample in time to 50 Hz. Depending on the dataset, we select a subset of the channels that exhibits better SNR and is more consistent, for the submarine datasets for example we make sure that the fiber optic cable is already located at the sea floor level and not in the surface as that would introduce noise from the surface waves. Then for each of the events we select, we extract the data within a 41 second time window (2048 samples at 50 Hz) centered approximately around the first arrival of the first wave. Finally, the data of each channel is normalised by its standard deviation.

3.2.1 HCMR/NESTOR

The DAS data from HCMR and NESTOR datasets, were acquired on two adjacent dark optical fibers, situated on the central Hellenic shear zone, near a triple junction: the Kefalonia Transform Fault to the north-west, and the Hellenic Trench and Mediterranean Ridge to the south-east [44], an area of great seismic activity. From 1st January of 2022 until the 24th of June 2022, 646 earthquakes took place in a 200 km radius from Methoni, only counting the ones with magnitude 2 and above (source is the seismology lab of university of Athens online catalogue).

These cables are intended for the HCMR (Hellenic Center for Marine Research) and NESTOR (Neutrino Extended Submarine Telescope with Oceanographic Research [45]) projects. DAS data were acquired on April 18 and 19, 2019 on the HCMR cable and from April 19 to 25 on the NESTOR cable. The HCMR and NESTOR cables span 13.2 and 26.2 km, respectively: from a common landing point, they follow the same path for the first kilometer (at different conduits) inside the shallow Methoni Bay, and then diverge in different directions toward the bottom of the East Ionian Sea, as can be seen on figure 3.3.

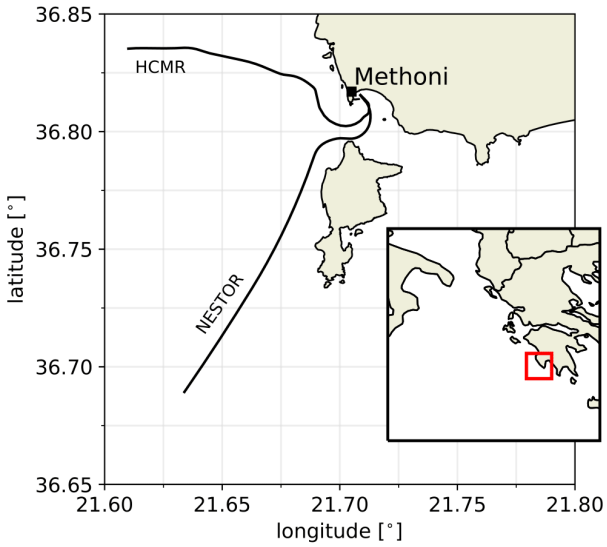


Figure 3.3. *Geographic location of the DAS cables HCMR and NESTOR. The inset shows the location of the region of interest within Greece. Image adopted from [4].*

Both cables recorded several local earthquakes during the measurement campaign, some of which were located and catalogued from [46], as evident on table 3.2. These

Cable name	Origin time (UTC)	Magnitude (local)	Location (latitude, longitude, depth [km])
NESTOR	22/04/2019 19:26:06	3.3	37.4185, 20.6897, 11.0
	23/04/2019 17:29:40	3.6	37.7753, 20.7658, 7.0
	21/04/2019 22:11:47	2.0	36.8335, 22.0382, 2.0
	23/04/2019 19:25:51	2.6	37.2528, 21.4593, 9.0
HCMR	18/04/2019 21:44:42	3.7	37.57, 20.66, 8.0
	19/04/2019 03:30:19	2.6	37.1523, 20.6662, 1.0

Table 3.2. *Catalogued events from the HCMR & NESTOR cables.*

cables were interrogated using an old generation Febus A1-R DAS interrogator, developed by Febus Optics. This single-pulse phase-based system produces longitudinal strain-rate measurement. Raw data was processed using gauge length and spatial sampling of 19.2 m for both cables, equivalent to 688 and 1,365 channels of strain-rate equally spaced along the HCMR and NESTOR cables, respectively. For our purpose, we selected 650 channels from HCMR and 880 from NESTOR. Strain-rate was computed at intervals of 6 ms (166.66 Hz frequency) for HCMR and 5 ms (200 Hz frequency) for NESTOR, producing 68 and 74 GB of data for HCMR and NESTOR, respectively.

3.2.2 Iceland

The Iceland dataset is different than the other datasets we consider in this work, as it is not a submarine fiber optic cable. This experiment took place in a volcanic environment

in Grímsvötn, Iceland. Grímsvötn is Iceland's most active volcano, with its most recent eruption occurring in 2011. It is situated underneath the Vatnajökull ice cap, and our DAS dataset was acquired by trenching a fiber-optic cable 50 cm into the snow cover. The total length of the cable was 12 km, and its layout can be seen in figure 3.4.

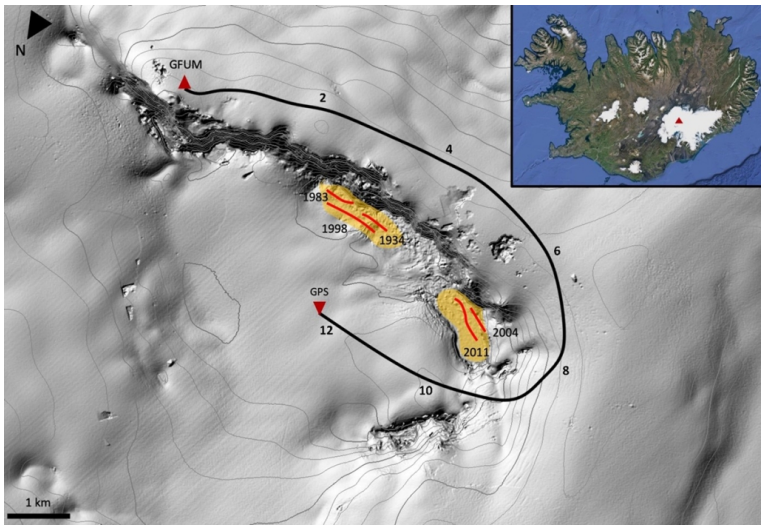


Figure 3.4. *Layout of the fiber-optic cable in Grímsvötn. (black line with numbers indicating distance in kilometers) deployed in the DAS-BúmmBúmm experiment in spring 2021. Locations of the research huts (GFUM) near one end of the cable and a GPS station at the other end are also shown, as are the years and approximate locations of previous fissure eruptions (orange and red). The site of Grímsvötn (red triangle) amid the Vatnajökull ice sheet in Iceland is indicated in the inset. Image taken from this article.*

A Silixa iDAS^{TM} interrogation unit was running for one month from May to June 2021, generating approximately one terabyte of data. Raw data was processed using gauge length of 10 m and spatial sampling of 8 m, equivalent to 1728 channels. For our purpose, we selected a subset of 1000 channels from the first 1500 channels, depending on the location each event took place. Strain-rate measurements were obtained with a frequency of 200 Hz, producing approximately 1 TB of data in total. Due to the deep trenching, the dataset has a relatively high signal-to-noise ratio. Physical signals were easily visible within the data by looking at images.

The signals from the events we selected, represent events that may have been caused by a wide range of phenomena, such as volcanic and geothermal activity, ice-quakes, snow avalanches, and resonance of the sub-glacial lake and the overlying ice sheet. This means they can't be characterised by magnitude as they weren't catalogued the way normal earthquakes are. We hand-picked 6 events in total.

3.2.3 Santorini

Lastly, this dataset is another submarine fiber optic cable, from the island of Santorini which is located in the southernmost part of Cyclades, in the Aegean Sea. Santorini is of great seismic interest, as is volcanic in nature, with the last eruption being in 1950. The caldera of Santorini lies in the center of the Christiana-Santorini-Kolumbo volcanic field,

which comprises the extinct Christiana Volcano, the Santorini Caldera, the polygenetic submarine Kolumbo Volcano, as well as the Kolumbo Volcanic Chain. Just to get some perspective of the seismic activity in the general vicinity of Santorini, from 1st January of 2022 until the 24th of June 2022, 1023 earthquakes took place in a 200 km radius from the center of Santorini island, only counting the ones with magnitude 2 and above.

The fiber optic cable, which is of commercial telecommunication purposes, starts off at Thira mainland, and after almost 10 kilometers enters the sea, which is evident in the DAS recordings from the surface waves' noise, and ends up in the mainland of Ios island, after getting close to the Kolumbo submarine volcano. The path it takes in order to get there is slightly curved with the radius curvature being large enough so that it can be considered straight.

The same Silixa iDAS^{TM} interrogation unit as in the Iceland dataset was used, it was running for two months from November to December 2021, generating almost one terabyte of data every day (due to the large sampling rate). Raw data was processed using gauge length of 10 m and spatial sampling of 8 m, equivalent to 5568 channels, giving a fiber optic cable length of 44.5 kilometers, one of the largest to our knowledge. Strain-rate measurements were obtained with a sampling frequency of 1000 Hz.

For our purpose, we selected a subset of 1000 channels in total, out of the 5568, starting from channel 1700 until 2700. This was because the cable enters the sea at approximately channel 1100, and we had to make sure that it is located at the sea floor. Another reason is that due to the great length of this fiber optic cable, the further away the channel is from the IU, Rayleigh scatterers tend to accumulate, producing more and more noise rendering the further away channels less useful.

The earthquake events that took place in the general area were catalogued by the seismology lab of university of Athens online catalogue. During the 2 month period of the experiment, lots of events took place, including an astounding magnitude of 6 event, that took place 300 kilometers from Santorini that was also recorded. Due to the large volume of events, we analyzed 8 hours of data and hand picked 14 events, information about which can be seen on table 4.2.

3.3 Synthetic Data Generation

If we were to use our raw DAS recordings as inputs to our model, then the percentage of recordings that is noise, would be much more significant than the portion of the recordings that is actual events. This is a problem similar to class imbalance in supervised learning problems. That means that the model will not be able to distinguish between noise and events, and can potentially learn to output noise and suppress it, which is the opposite of our learning objective.

In order to circumvent that problem and to gain a first order understanding of the denoiser, we generate a synthetic dataset with "clean" waveforms corrupted by Gaussian white noise with a controlled signal-to-noise ratio (SNR). The clean strain rate waveforms are obtained from three-component broadband seismometer recordings of the Pinon Flats Observatory Array (PFO, [47]), California, USA, of 82 individual earthquakes. These

earthquakes are manually selected based on a visual evaluation of SNR and waveform diversity.

Since DAS recordings measure strain-rate, which is essentially the difference of strain of two points separated by the gauge length, both in distance and in time, we can calculate the strain rate for two seismometers given their distance. The strain rate $\dot{\varepsilon}$ recorded by DAS at a location x can be expressed as:

$$\dot{\varepsilon}(x) = \frac{1}{x_g}\left[\dot{u}\left(x + \frac{x_g}{2}\right) - \dot{u}\left(x - \frac{x_g}{2}\right)\right]$$

where x_g is the gauge length and $\dot{u}$ the particle velocity. To simulate DAS strain rate recordings, as we mentioned we need to take two different stations. These are separated by 50 m, so we divide the difference of their respective waveform recordings by their distance. Owing to the low noise floor of these shallow borehole seismometers, the resulting strain rate waveforms exhibit an extremely high SNR.

As the different DAS recordings and the recordings from the seismometers are sampled at different frequencies, we need all of them to be sampled in the frequency for our model to achieve its full potential. For that reason, we agree to downsample all our DAS recordings to 50 Hz in a frequency band of 1-10 Hz.

The PFO broadband stations are sampled at a 40 Hz frequency, so in order to simulate a 1-10 Hz frequency band sampled at 50 Hz (i.e. a frequency range of 0.04-0.4 times the Nyquist frequency), we filter the synthetic waveforms in a 0.8-8 Hz frequency band and apply no resampling. All the waveforms are then scaled by their individual standard deviations.

Up to this point, we have simulated one channel of DAS recordings. In order for the synthetic dataset to work, we need 11 waveforms of recording of neighbouring channels, as seen in figure 3.1. Also, different events have different magnitudes, meaning the waves that are sensed from our fiber optic cables have different apparent speeds (velocities) depending on the event characteristics. The velocity of the wave fronts define the time delta that it takes for the event to reach two different channels that are separated by x_g meters.

In order to account for all that, during training of the model, we create a synthetic sample by randomly choosing one "clean" waveform, among with a random apparent wave speed v in the range of $\pm 0.2 - 10 km \cdot s^{-1}$. This will help the model better generalize to events approaching with different apparent speeds. A total of 11 copies (N_{sub}) of the selected waveform are created, and each are offset in time in accordance with the moveout, which is given by the following equation:

$$\Delta N_i = int\left(\frac{i \cdot x_g \cdot f}{v}\right),\ 0 \leq i < 11$$

ΔN_i being the time offset in number of samples of the i-th copy, x_g the gauge-length, f the sampling frequency (set at 50Hz) and v the apparent wave speed.

The waveforms are then cropped within a window of 2048 time samples, positioned randomly around the arrival of the first wave. We selected a sampling rate of 50 Hz, as

41 seconds of recording is enough to contain a whole earthquake event and also 2048 samples is a power of two, (2^{11}), which is preferred in inputs of neural networks as they are trained in GPUs (Graphics Processing Unit) which can take advantage of optimizations related to efficiencies in working with powers of two.

Lastly, a SNR value is sampled from a log uniform over the values 0.01-10, and the waveforms are re-scaled such that the maximum amplitude of the signal is $2\sqrt{SNR}$. These scaled waveforms are then superimposed onto Gaussian white noise with unit variance, filtered in a 1-10 Hz frequency band, and scaled by the total variance.

In order to avoid over-fitting in this limited dataset, and to help the model achieve better generalization, we also do some trivial data augmentation. Data augmentation in data analysis are techniques used to increase the amount of data by adding slightly modified copies of already existing data (or newly created synthetic data from existing data). The techniques we use to augment our data, are random polarity flips and time reversals of the final processed sample. Polarity flips mean that we flip the waveform with respect to the x-axis. Time reversal means that we reverse the time axis and feed the waveform to the model starting from the end of the event.

3.4 Training Procedure

In order to improve the rate of convergence on the real world DAS datasets and to help our model to better distinguish between noise and events and learn to not output incoherent noise, we first train the model on the synthetic dataset. We split the dataset 80-20 in a training and a validation set, and use these separate sets to generate training and validation synthetics as we describe in section 3.3. In the synthetic data generation there are random procedures taking place, like the selection of apparent speeds, SNR as well as the selection of waveforms. For this reason, we create a new set every epoch, in order to mitigate overfitting. A batch size of 32 was selected. We trained for 2000 epochs where the validation set loss saturates. Training on the synthetic dataset took just approximately 7 hours on a single Nvidia RTX 3070 (laptop) GPU. This model is then saved, and used for the analysis of the synthetic dataset in section 4.2.

At this point, we are ready to re-train our neural network to the real world DAS dataset we aim to denoise. As we mentioned earlier, both the Santorini and the Iceland datasets, have a channel spacing that is a subset of the gauge length, causing data leakage in between neighbouring channels. This data leakage essentially prohibits us from training on these datasets. A way to circumvent this problem, is to select channels that are not immediate neighbours. This solution wastes lots of channels and prohibits us from leveraging the great spatial resolution of DAS, one of the benefits of the method.

We decided to train only on the HCMR and NESTOR datasets, and since the denoising capabilities of the final model were satisfying, we decided to keep it this way. We load the model that was trained in the synthetic dataset, and continue training on the 21 recorded events on HCMR and 8 events on NESTOR (some of which were catalogued, available on table 3.2). We manually kept out of training 4 and 2 events for validation respectively. During training of the real-world DAS data, we generate a batch of samples from randomly

selected events and central DAS channels. We then take 5 DAS channels on either side of a central channel and randomly select one of them as a target channel, to be blanked from the input and to be reconstructed by the model. We additionally perform polarity flips and time reversals on the set of 11 waveforms to augment the dataset.

The model's performance saturates after just 50 epochs, which take approximately 15 minutes in the same GPU. In order to get the $\mathcal{J}$-invariant reconstructions of the DAS data along the entire cable, we create 11 channel input samples centered around a target channel (that is blanked), and by sliding that window from one channel to another we reconstruct all of the channels.

Chapter 4

Results and Analysis

4.1 Ablation Study

With the term ablation study, we refer to the study in which the performance of an Artificial Intelligence system is investigated by removing certain components, to understand the contribution of the component to the overall system. In our case, we try several values for certain hyperparameters of the system, to see how important they are to the final outcome, and also to determine what set of hyperparameters we will use in our final model, that serves the best final results.

For example, we would like to know the importance of factors such as: Learning Rate, Number of Epochs, the number of waveforms that are inserted in the neural network with every batch, the batch size and also different models and optimizers. Since our model is already simple enough and can be trained in logical amounts of time, we will not be removing additional layers and components of it, but rather replacing different types of components, such as replacing normal convolution layers with anti aliasing ones, to see if they do a good job in confining aliasing effects.

Experiment tracking was done with the Weights and Biases (wandb) [48] framework, which makes it possible to define a grid of hyperparameters to draw values from, and perform a sweep either in a random manner or by using Bayesian Hyperparameter Optimization. The latter was used, as it is the most suitable option when the task includes a large set of different combinations of hyperparameters. Bayesian optimization builds a probabilistic model of the function mapping from hyperparameter values to the objective evaluated on a validation set.

To gain a first order understanding of how each hyperparameter affects the validation loss, we prepare and run a sweep for a wide range of values of the most important hyperparameters of our model, for a total of 13 runs on 25% of the training and 25% of the validation data sets, because if we were to use the whole dataset it would increase the runtime massively.

The results on each run can be seen on figure 4.1, by following each line, it is possible to see what combination of hyperparameters resulted in what value for the validation loss and also how much time it took. As this is not an exactly helpful way to understand how each parameter affects the validation loss, wandb also offers the ability to see the importance and the correlation of each hyperparameter with respect to the validation loss,

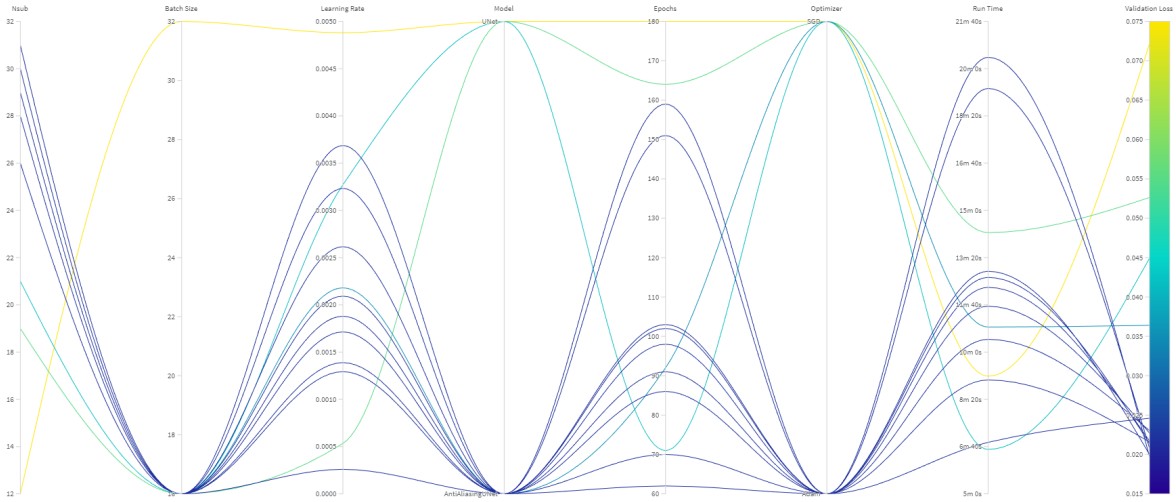


Figure 4.1. *Results of the hyperparameter sweep*

in a more aggregated way, as is evident on figure 4.2.

Correlation is the linear correlation between the hyperparameter and the chosen metric (in this case *current_val_loss*). So a high correlation means that when the hyperparameter has a higher value, the metric also has higher values and vice versa. Green values show positive correlation while red values show negative correlation. Correlation is a great metric to look at but it can't capture second order interactions between inputs and it can get messy to compare inputs with wildly different ranges.

Therefore we also calculate an importance metric where we train a random forest with the hyperparameters as inputs and the metric as the target output and report the feature importance values for the random forest. The importance column shows you the degree to which each hyperparameter was useful in predicting the chosen metric.

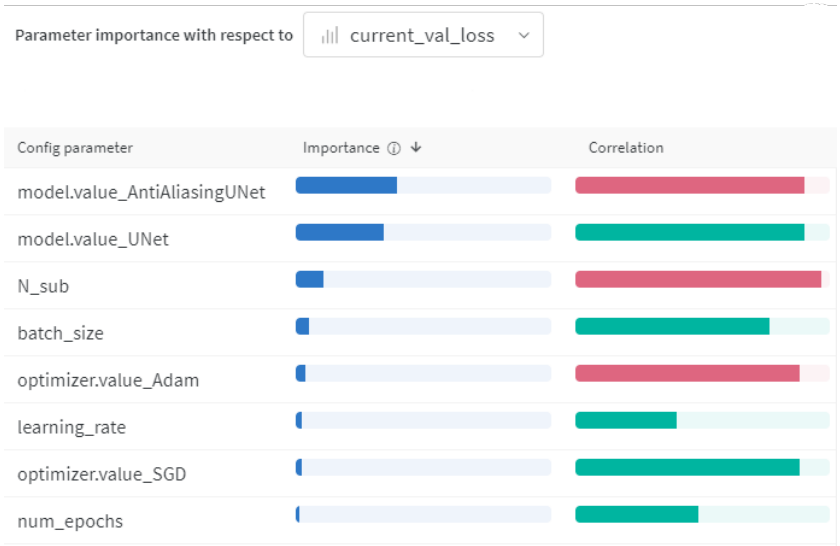


Figure 4.2. *Parameter importance and correlation with validation loss*

In our case, it is evident that the most important parameter in the sweep, was the anti aliasing UNet, which as we mentioned before is the same as the simple UNet, only with

different CNN's, replaced with anti aliasing ones from Adobe [5]. Apart from the different model, the parameter *N_sub*, which is the number of waveforms (or different channels) that are input in the model is the most important, which intuitively seems correct, as for every event the model receives information from more neighbouring channels. As this parameter also affects the batch input size, it should be fine tuned along with the batch size, because if we select high values for both of the parameters, it will affect the input/output of the data through the network.

Therefore, for our final model, we used the Anti Aliasing UNet, along with Adam optimizer. For the rest of the hyperparameters, we took the results from the best run (from figure 4.1, the one with the lowest validation loss) as a baseline: $N\_sub = 30$, $learning\_rate = 0.002$, $num\_epochs = 200$, $batch\_size = 16$. Consequently, while experimenting we decided to decrease the *N_sub* to 11, as it increased the image size dramatically, hence increasing training and most importantly, inference time. To compensate for that, we increased the batch size to 32. Experimenting with these values we found that the training and validation curves were really noisy, and also we could tell from the validation metrics there was room for improvement, so we decided to decrease the learning rate to $5 \cdot 10^{-4}$ and train for more epochs. We did a large run of 3000 epochs while keeping checkpoints, and selected the best model at 2000 epochs, finally arriving at the final set of hyperparameters, agreeing with [4].

N sub	Batch Size	Epochs	Learning Rate	Model	Hidden Layers	Optimizer
11	32	2000	$5 \cdot 10^{-4}$	Anti Aliasing UNet	4	Adam

Table 4.1. *Final Set of Hyperparameters of the model*

The training and validation loss curves for the total run, can be seen at figure 4.3. As is evident, throughout the training, the validation loss is higher than the training loss, despite some statistically insignificant instances that are normal to arise due to the randomness that takes place when generating the synthetic data, as mentioned in section 3.3.

4.2 Results on synthetic data

We start off with a qualitative assessment of our algorithm's performance on synthetic data. We start by taking an event from the test samples, and as these events exhibit a really high SNR (Signal to Noise Ratio), we generate the input samples as described in section 3.3, by taking a value of SNR so that $SNR = \{0.1, 1, 10\}$ and an apparent wave speed of 1.5 km/s. In the first column of figure 4.4, in the clean signal, the P- and S-waves are clearly distinguishable from the background noise, with the S- wave exhibiting a considerably higher amplitude than the P- wave.

We generate the input sample by adding a small amount of Gaussian white noise of $SNR = 10$, filtered in a $1 - 10$ Hz frequency band (panels b and i) and we can still clearly

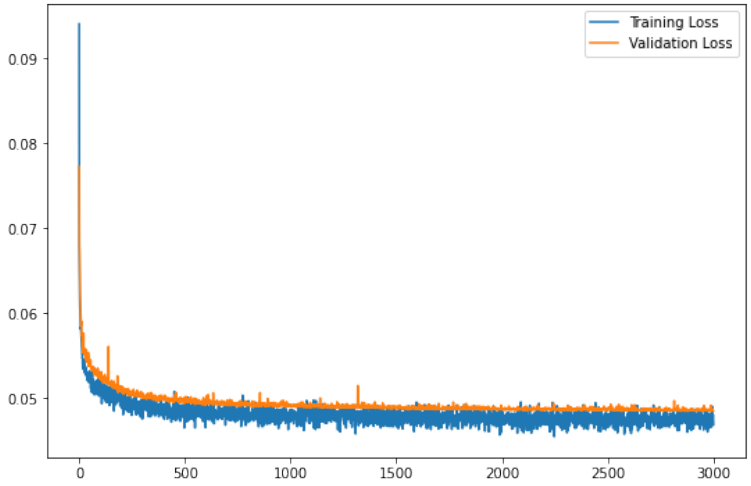


Figure 4.3. *Training and Validation Loss Curves for training on synthetic data.*

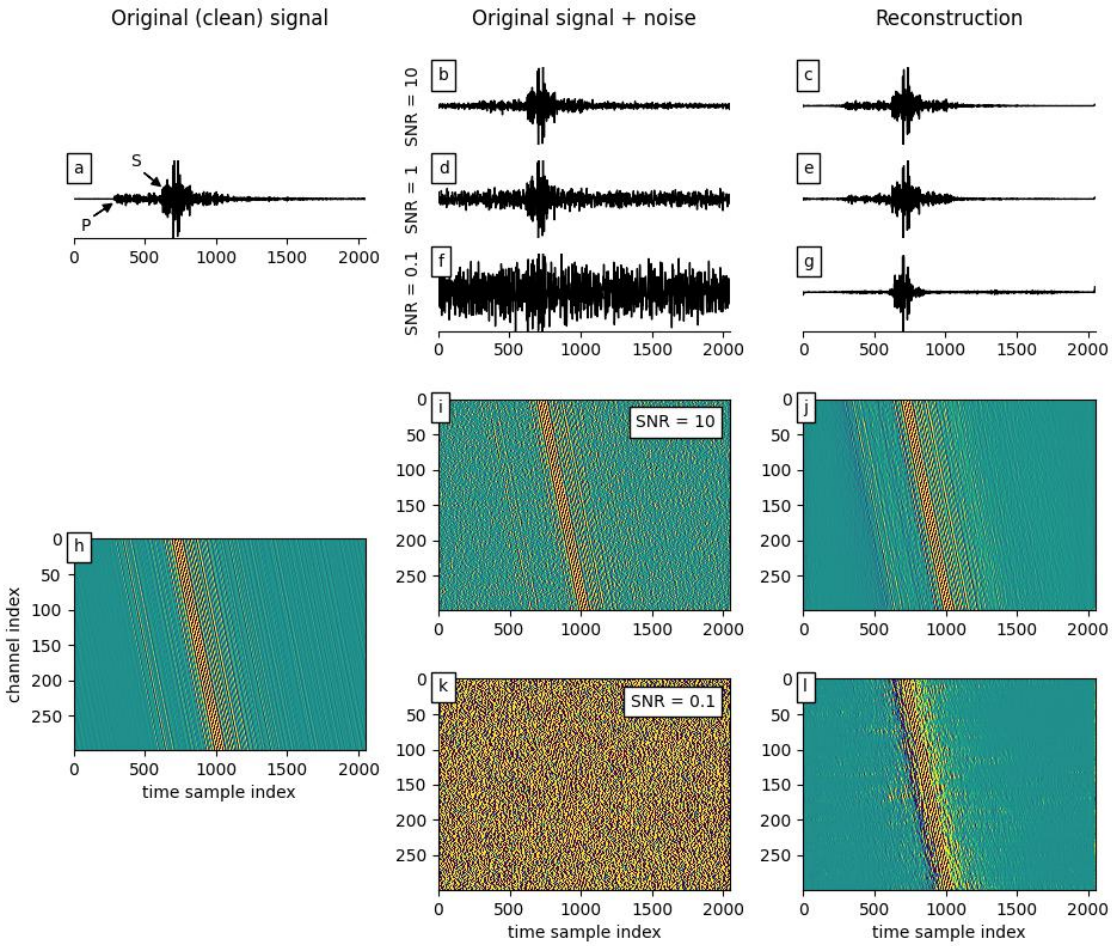


Figure 4.4. *Synthetic examples of model performance.*

see the waveform features. When this mildly corrupted sample is fed to the model, the reconstruction is nearly indistinguishable to the input and to the original input.

For the second input sample, we select a lower $SNR = 1$. In this waveform, the P-wave and a portion of the S-wave are vanished within the noise but the peak strains are

still visible. In such a scenario, it would be impossible for us to accurately pick the P- and S- waves arrival times, which are important metrics as they allow us to estimate the distance from the seismic source. After J-invariant filtering, the signal's P-wave train is lifted out of the noise level. The onset of the P- and S-waves becomes much more clear, permitting a crude estimation of their arrival times. Moreover, details of the S-wave are fairly well recovered (panels d and e).

Lastly, we select an extremely poor SNR value, $SNR = 0.1$. In this case, even the S-wave portion of the waveform is nearly indistinguishable from the noise. Without additional processing of the signal, it would be nearly impossible for an anomaly detection algorithm (e.g. STA/LTA [11] or [49]) to detect an event with such poor SNR. After J-invariant filtering, the S-wave train is recovered, albeit with a much lower amplitude than the original signal. The P-wave can no longer be distinguished in a single waveform (panel g), but from the reconstruction (panel l) it is apparent that small amounts of the P-wave energy are recovered. In spite of the imperfect reconstruction, the SNR of the reconstructed signal is sufficient to identify this event with detection algorithms.

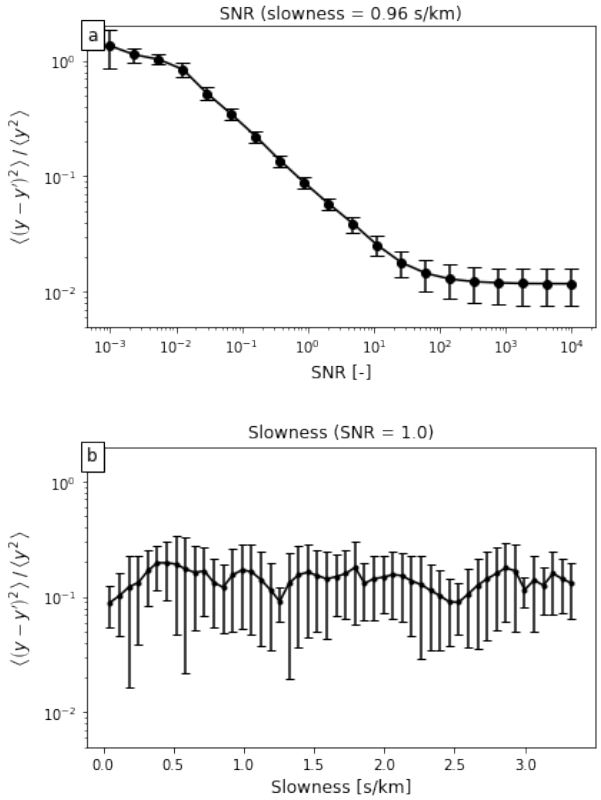


Figure 4.5. *Quantitative assessment of the model performance on synthetic data.*

We continue with a quantitative assessment of the model performance by computing the scaled variance of the residuals, defined as $R = \left\langle (y - y')^2 \right\rangle / \left\langle y^2 \right\rangle$, with y being the "clean" signal and y' the reconstruction. We compute this quantity for a range of values of SNR and slowness (reciprocal of wave velocity), in figure 4.5. The error bars are calculated

from 300 samples generated from the same clean waveform. As desired, the model output becomes more accurate when the SNR is high, which saturates towards the end of the SNR range. Towards the lower end of the SNR range the scaled variance approaches 1, indicating that the model essentially produces zero-centred random noise with small variance, so that: $\left\langle (y - y')^2 \right\rangle \approx \left\langle y^2 \right\rangle$. This is a highly desired outcome: when provided with purely random, incoherent noise, the model should output zero. In other words, the prior learnt by the decoder of the auto-encoding network is zero, which prevents the generation of non-existing signals driven by a dominant non-zero prior.

As detailed in 3.3, the time-offset of the waveform between neighbouring channels is governed by the slowness. For low slowness values, the offset between neighbouring waveforms is minimal, so that a reasonably accurate reconstruction can be generated from simply copying a non-blanked waveform from the model input. This is obviously undesired, and so we investigate this hypothesis by systematically varying the slowness (and correspondingly the time-offset between channels). As is apparent from 4.5 (b), this hypothesis can be safely discarded: the scaled residuals remain constant over a wide range of slowness, varying from 0.1 to 3.3 $s\ km^{-1}$ (small to large time-offset, respectively). However, we do find a small but systematic drop in scaled residuals at fixed intervals. Further investigation reveals that these occur at integer multiples of $1/9.6\ s\ km^{-1}$. The time-offset between neighbouring channels, is given as $\Delta N = int(Lf/u)$, L being the gauge length of $8m$ and f the sampling frequency of 50 Hz. For $u < 9.6\ m/s$, the offset between neighbouring channels is 0. For $960 \leq u < 1920\ m/s$, the offset is 1, etc. For channels that are separated by i gauge lengths, these jumps in time-offset occur at integer multiples of $u = iLf$. In other words, the method by which the synthetic samples are generated causes a jagged, non-exact offset between channels due to integer rounding. Only when the slowness is an integer multiple of $1/Lf = 1/960\ s\ m^{-1}$ is the offset between the channels exactly as given by the theoretical move-out. ween the channels exactly as given by the theoretical move-out. When this condition is satisfied, the offset between close and far channels is fully consistent, and correspondingly the model performance improves. This suggests that not only does the model refrain from simply copying the input data, but that it also considers both far and close channels to assess the move-out, which is then used to reproduce the correct time-offset of the reconstruction. In the real-world DAS data the wavefield is not discretised (i.e. the arrival of waves at a given channel is exact) and so this time-offset rounding does not occur.

4.3 Results on real Distributed Acoustic Sensing (DAS) data

As explained in section 3.4, after training on the synthetic dataset, we move on to train our model on real DAS data, specifically on 21 events from HCMR and 8 events from NESTOR, adapted from [4], (6 of which were located and catalogued, [46]), out of which 8 and 2 events were manually selected for validation, respectively. The training and validation loss (MSE) curves can be seen on 4.6. As expected, the validation loss reflects the trend of the training loss, being a little higher in magnitude.

Since we are retraining the model, we keep the same values for the hyperparameters,

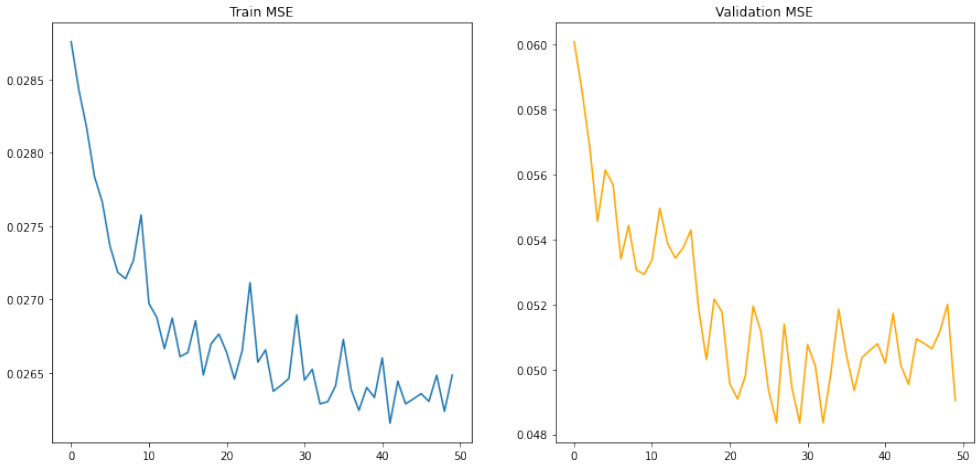


Figure 4.6. *Training and Validation loss curves for HCMR/NESTOR data.*

but we train for only 50 epochs. We did not train the model further or separately for the other datasets, as the contribution of training specifically for each dataset was negligible to the results, and thus it did not outweigh the computational cost to do so. This is the case, as the data, even so coming from different environments and with differences in their configurations (gauge length, channel spacing and sampling rate) are of the same nature, and so the reconstructions are satisfactory even without retraining. It does make sense to retrain for each specific dataset, though, in order to capture small nuances each data acquisition experiment has.

For each qualitative assessment of the denoising reconstructions we present, in order go get a measure of the model's performance without ground truth (as there is no ground truth), we compute the local waveform coherence before and after J-invariant filtering and assess the gain in coherence. We define the mean local waveform coherence CC around the k-th DAS channel, as:

$$\mathrm{CC}_k = \frac{1}{4N^2}\left[\sum_{i,j=-N}^{+N} \max\left(\frac{x_{k+i} * x_{k+j}}{\sqrt{\sum_t x_{k+i}^2 \sum_t x_{k+j}^2}}\right) - 2N - 1\right]$$

where x_n denotes the waveform at the n-th channel, $*$ denotes cross-correlation, and $\sum_t x^2$ denotes the sum over all time samples in x. The bin size N is set to 11, for all reconstructions. The coherence gain is then defined as the local coherence computed for the J-invariant reconstruction, divided by that of the input data. As such, coherence gains above 1 indicate improved waveform coherence compared to the input data, which is beneficial for coherence-based seismological analyses, such as template matching or beamforming.

4.3.1 HCMR/NESTOR

We then continue with a qualitative assessment of the denoising results, by considering two events in the validation set of the HCMR data, and two in the validation set of

NESTOR data, as can be seen on figures 4.7 and 4.8, respectively.

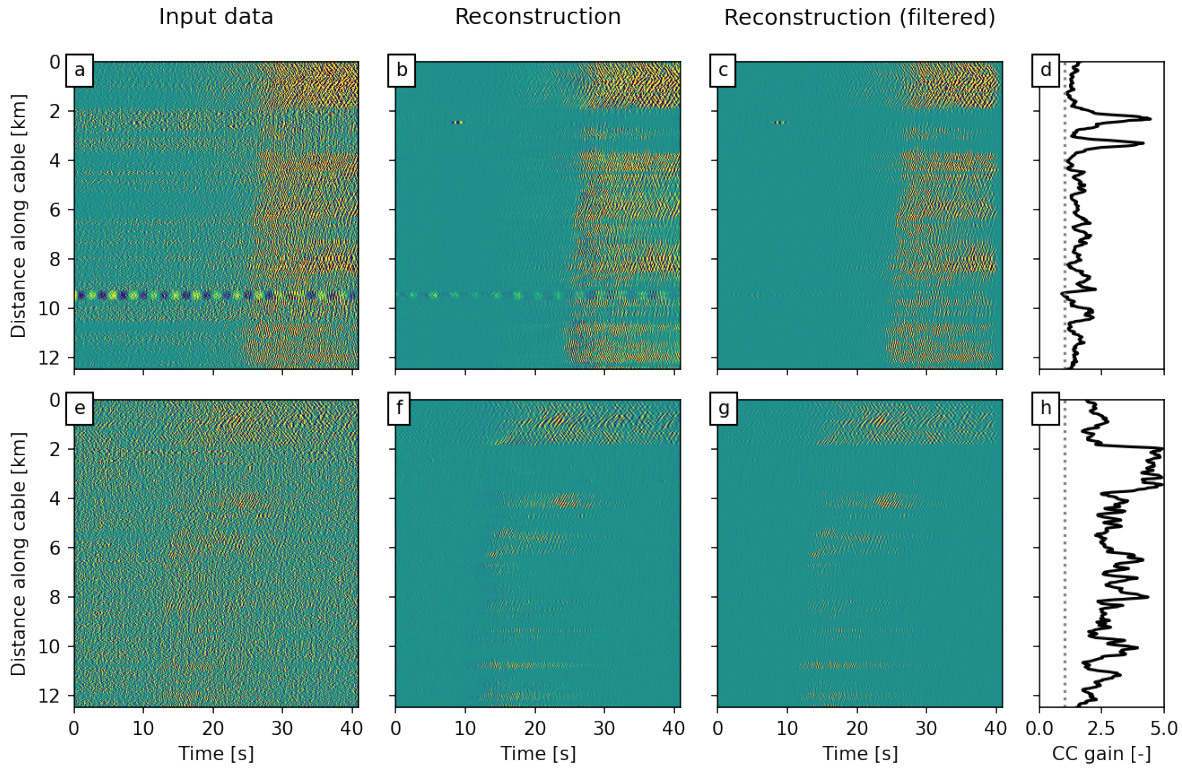


Figure 4.7. *J-invariant filtering results of HCMR data.*

Looking at the J-invariant reconstructions, they practically always exhibit (much) higher waveform coherence. Along some cable segments this quantity is inflated due to an absence of coherent signals in both the input data and the reconstruction, which is particularly apparent in panel 4.7(h) between 2 and 4 km distance. Nonetheless along other segments, such as between 0 and 2 km in panel 4.7(h) or around 15 km in panels 4.8(d) and (h), the local coherence of recorded earthquake signals have improved substantially (the vertical dotted line marks a gain of 1).

When considering the SNR of the DAS data as shown e.g. in panel 4.7 (a), we see that there are segments of the cable that exhibit better SNR than others (e.g. at 0.5, 3.9, and 10.8 km along the HCMR cable). This along-cable variation in SNR may be due to variations in environmental noise, cable-ground coupling degree, or orientation of the cable with respect to the wave propagation direction [50].

At locations where the apparent SNR is high, we can attempt to make a wiggle-for-wiggle comparison between the input data and the reconstructions on figure 4.9. When doing so, we see that the model correctly attenuates the random noise in the first 25 s, and subsequently increases its amplitude to match the recorded signals. Overall the reconstructions exhibit a lower maximum amplitude than the input data, which is as expected (the model removes the contribution of the noise to the recorded data). The phase of the large-amplitude arrivals seems to be matched fairly well, which is important

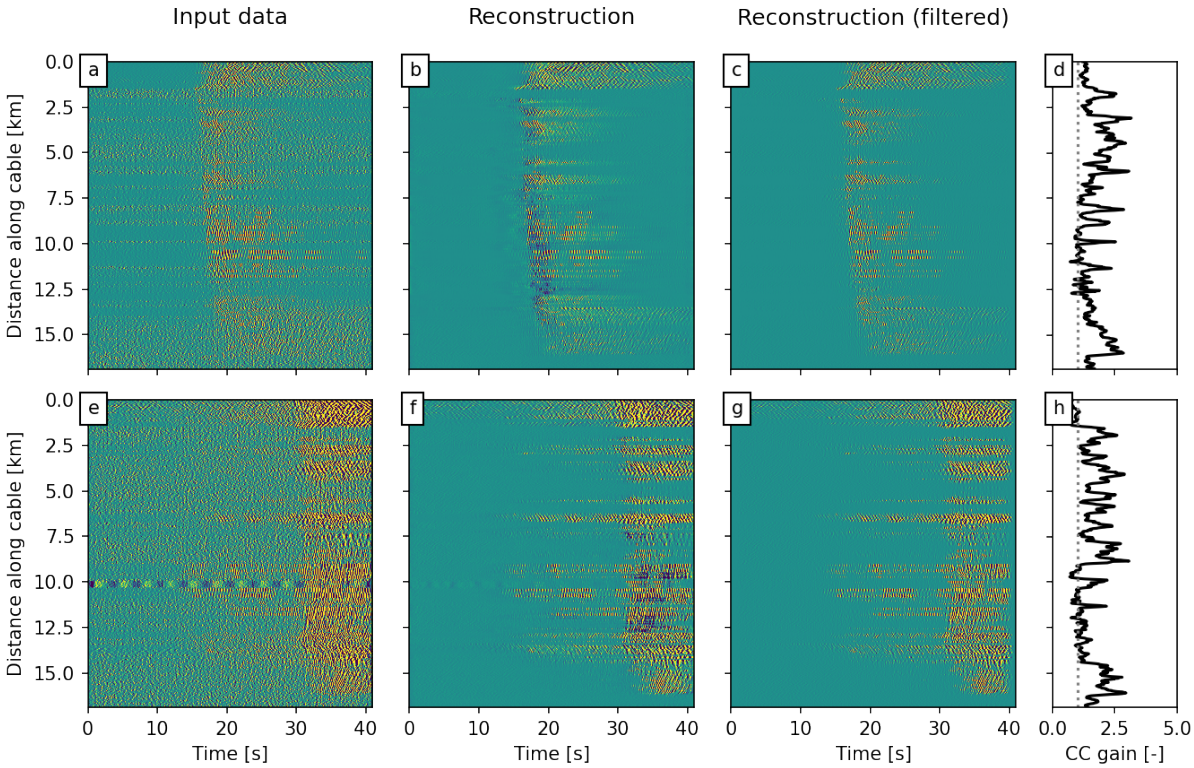


Figure 4.8. *J-invariant filtering results of NESTOR data.*

for seismological methods that rely on phase information, such as beamforming. This can be quantitatively expressed as the correlation coefficient computed for the waveforms after 25 s, which is overall satisfactory (in the range of 0.66 to 0.89 as indicated in figure 4.9).

4.3.2 Iceland

We then proceed, with the results from the fiber optic cable in Grímsvötn, Iceland. This dataset has the same sampling rate as the NESTOR fiber optic cable that we saw earlier, so the difference is only in spatial sampling, which for this cable was set at 8 meters, while gauge length was set at 10 meters, as we mentioned before. Therefore, in principle we should get satisfactory results, just by using the model that was trained on the HCMR/NESTOR dataset, for inference, but we should pay attention to how well the reconstructed signal matches the phase of the input signal as different channel spacing parameters between the dataset we train and the one we perform inference on can cause miss-matches in phase.

This dataset, coming from a 12 km fiber optic cable trenched at about 50 cm into the snow cover, exhibited great SNR due to the deep trenching. Physical signals such as hammer tests initially, and micro-seismic events later, are clearly visible by looking at images. So it's interesting to see how our pretrained model will denoise this "cleaner" waveforms.

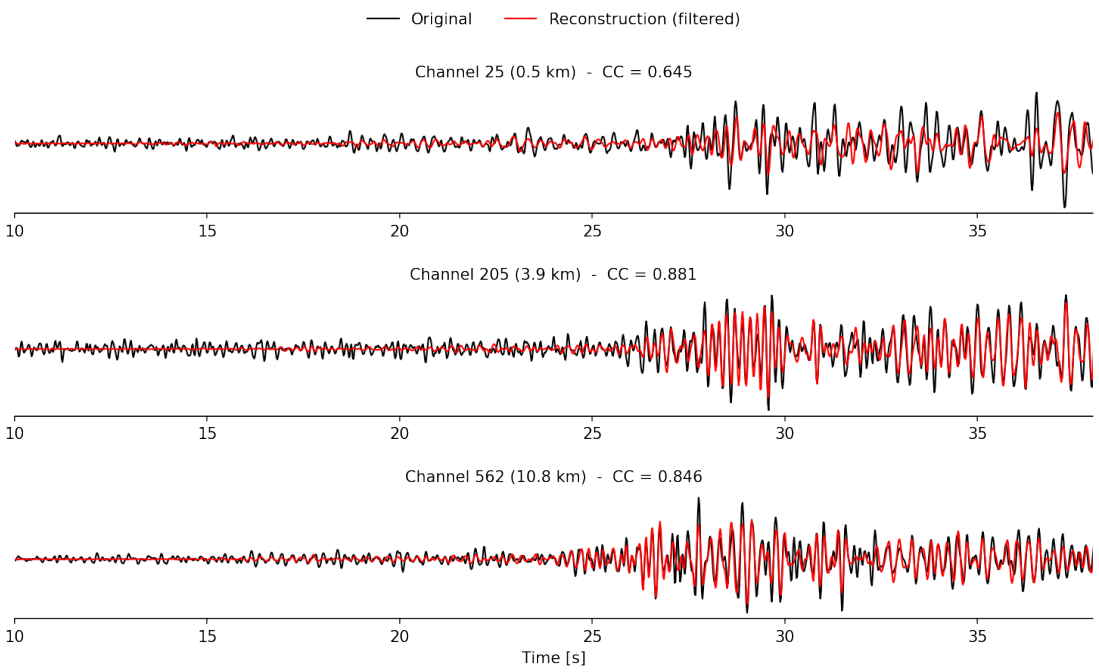


Figure 4.9. *Wiggle-for-wiggle comparison along the HCMR cable. The waveforms correspond with panels a,b,c of 4.7.*

The example signals, represent events that may have been caused by a wide range of phenomena, such as volcanic and geothermal activity, ice-quakes, snow avalanches, and resonance of the sub-glacial lake and the overlying ice sheet. This means they can't be characterised by magnitude as they weren't catalogued the way normal earthquakes are. We present 6 different events in total.

We start our qualitative assessment of the model's denoising results in figure 4.10 with two small in magnitude, possibly local events, with the first event (panels a,b,c) being larger in magnitude than the second (panels e,f,g). In the second event, the location of said event is denoted with an arrow as it can be too small to notice (panels e & g). Even though the background noise is smaller than in our other datasets, our model reconstructs the data in a satisfying degree, with the events still clearly visible. For both cases, the CC gain (panels d and h) is always higher than 1, as expected, showing us that the local coherence of the recorded event signals have improved substantially.

When considering the SNR of the DAS data as shown, we see that there are segments of the cable that exhibit better SNR than other (e.g. between 0 and the 1st km distance along the cable). This variation may be due to the cable's orientation with respect to the wave propagation direction, as this is also the area in which the two events in figure 4.10 were recorded.

We then attempt to make a wiggle-for-wiggle comparison in the high SNR area (figure 4.11), showcasing three different channel regions of the optic fiber cable. With the background noise being smaller in magnitude, the model does a really good job suppressing it, while successfully reconstructing the signal peaks of the event, proving us that this learning algorithm has successfully learned the noise characteristics, even though we

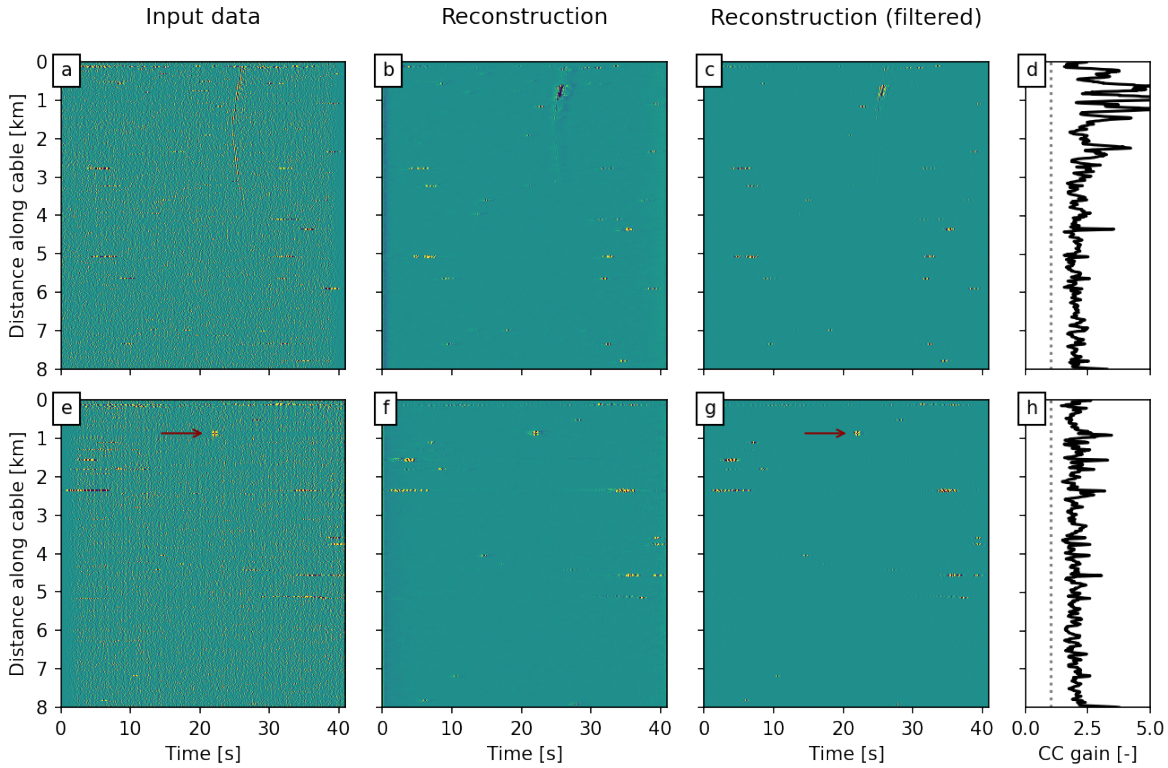


Figure 4.10. *J-invariant filtering results of Iceland data.*

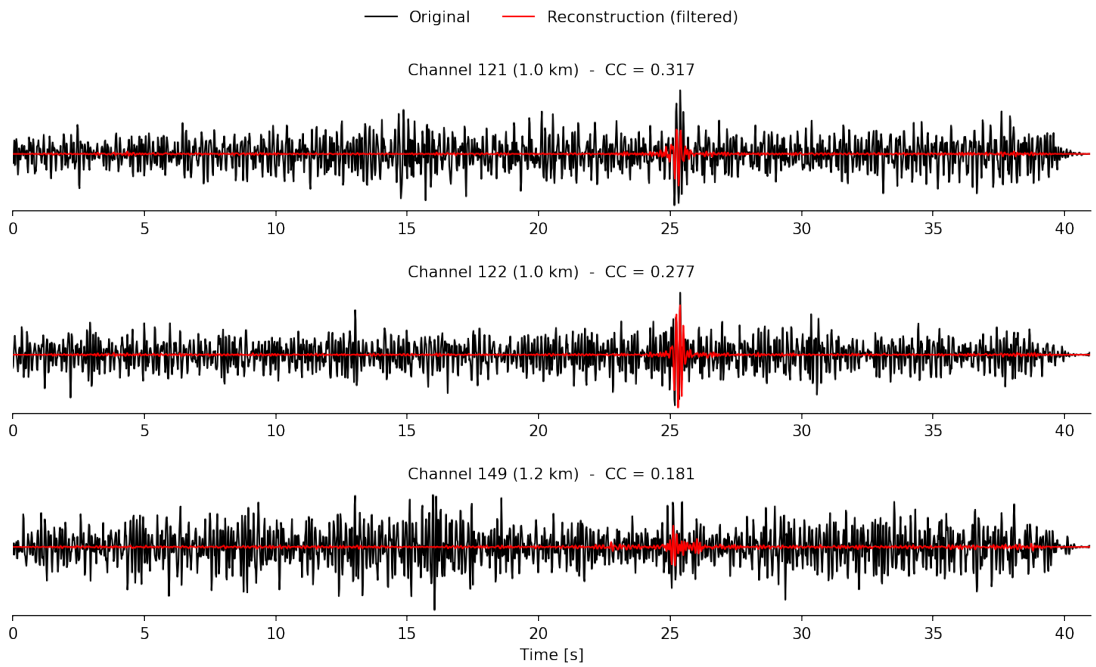


Figure 4.11. *Wiggle-for-wiggle comparison along the Iceland cable. The waveforms correspond with panels a,b,c of 4.10.*

made no assumptions regarding said characteristics. The Pearson correlation coherence values are significantly lower than in submarine DAS data, the reason being that the noise is of smaller magnitude, so the model outputs zero when provided with incoherent noise which is a highly desired outcome (that was also validated in the synthetic examples results, when we examined the scaled variance of residuals R for a range of different values of SNR, figure 4.5). Thus, it is normal to have lower coherence values between the original signal and the reconstructed, as they are weaker. In order to examine how the

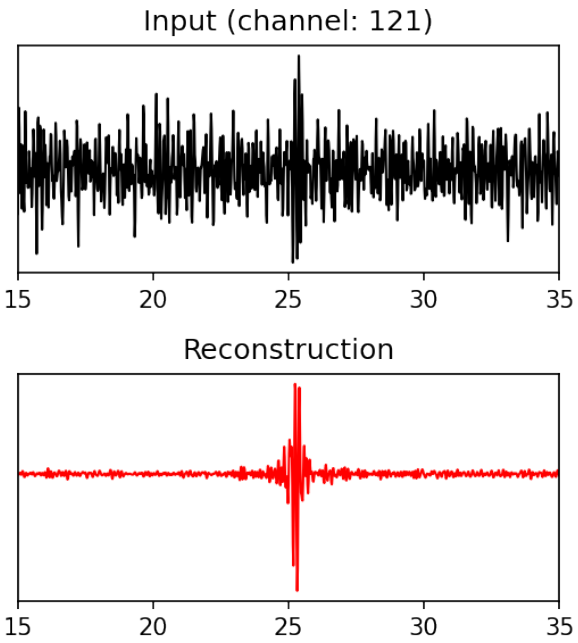


Figure 4.12. *Wiggle-for-wiggle comparison along the Iceland cable, on channel 121.*

reconstructed signal matches the phase of the original one, on figure 4.12 we examine a shorter period of time at a channel of interest, and by visual examination we conclude that the phase is matched really well. This is also a great example of the great denoising capabilities of our model, as it can reconstruct the signal of the coherent event even when the biggest portion of it is corrupted by the background noise. We continue with two higher magnitude events on figure 4.13. Again, the denoising results are satisfactory, with the CC gain being above 1 along the fiber optic cable. As usual, we attempt to do a wiggle-for-wiggle comparison in the area of interest, on figure 4.14. The reconstructions on these channels showcase the excellent denoising capabilities of our model. Despite them showing no coherence at all with the original signal, it is easily visible how well the reconstruction matches the phase of the original signal. Again, on figure 4.15 we zero in on a short time window of interest, this time on channel 520, to aid the reader to visually compare the quality of the reconstruction versus the original signal.

Finally, we present another two relatively big events on figure 4.16, exhibiting high CC gain values which is always above 1, except only a handful of channels out of the 1000 that are shown in the image. Similarly to the previous 4 events, the reconstructed

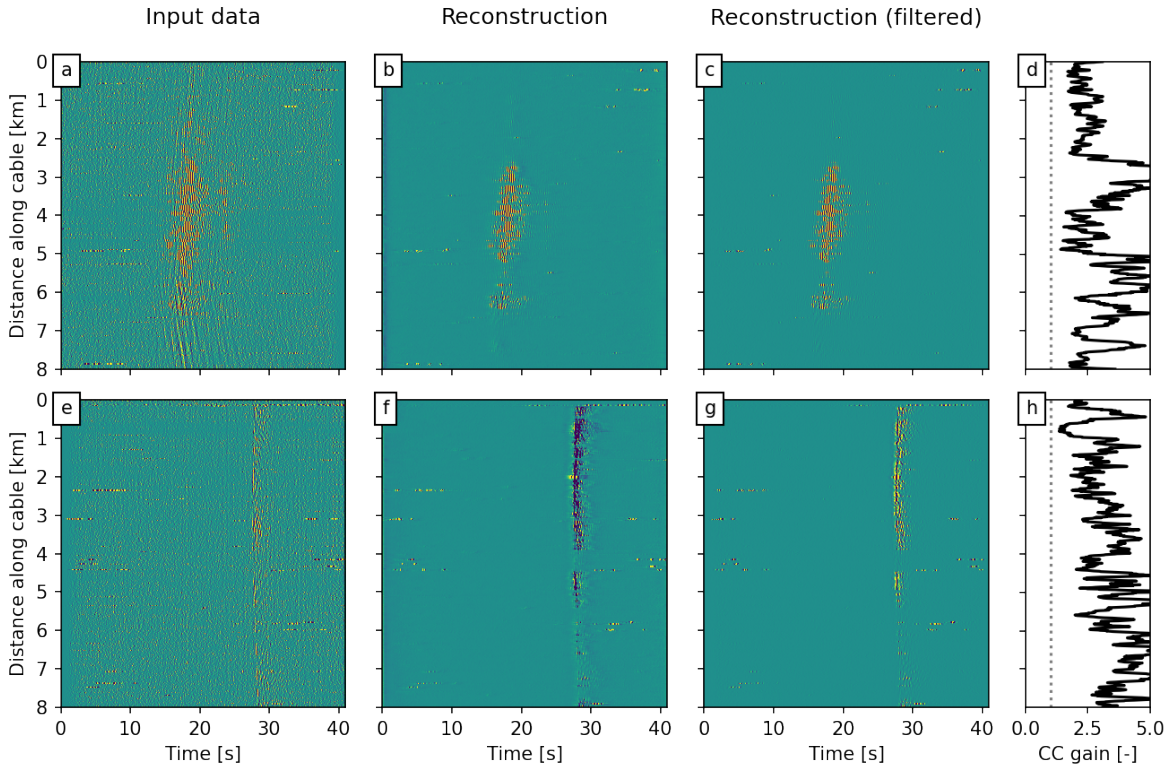


Figure 4.13. *J-invariant filtering results of Iceland data.*

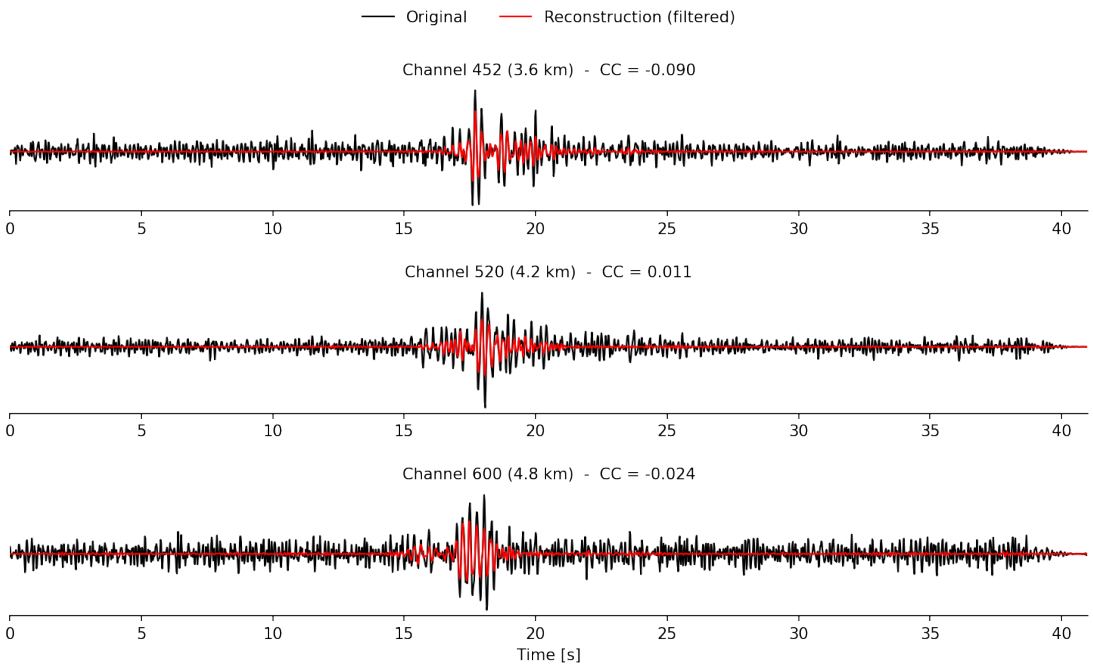


Figure 4.14. *Wiggle-for-wiggle comparison along the Iceland cable. The waveforms correspond with panels a,b,c of 4.13.*

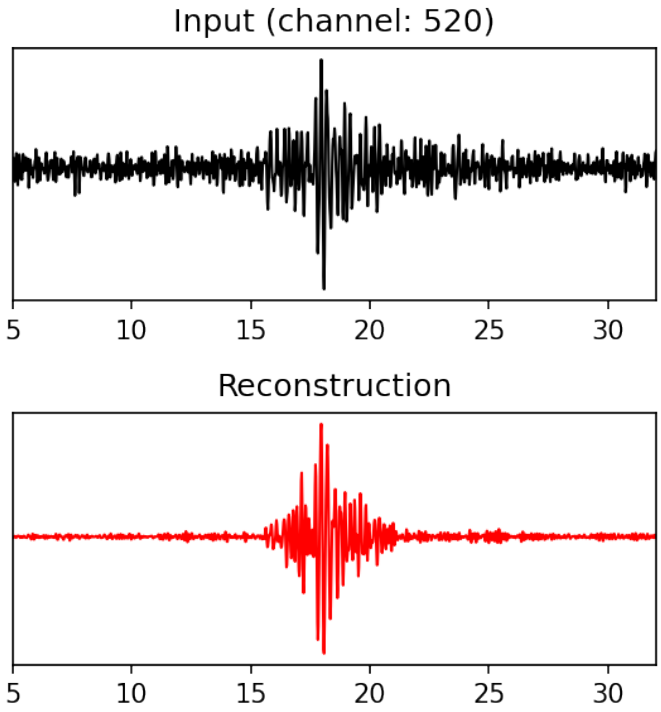


Figure 4.15. *Wiggle-for-wiggle comparison along the Iceland cable, on channel 520.*

waveforms on figure 4.17 match the phase of the original waveforms without any flaws.

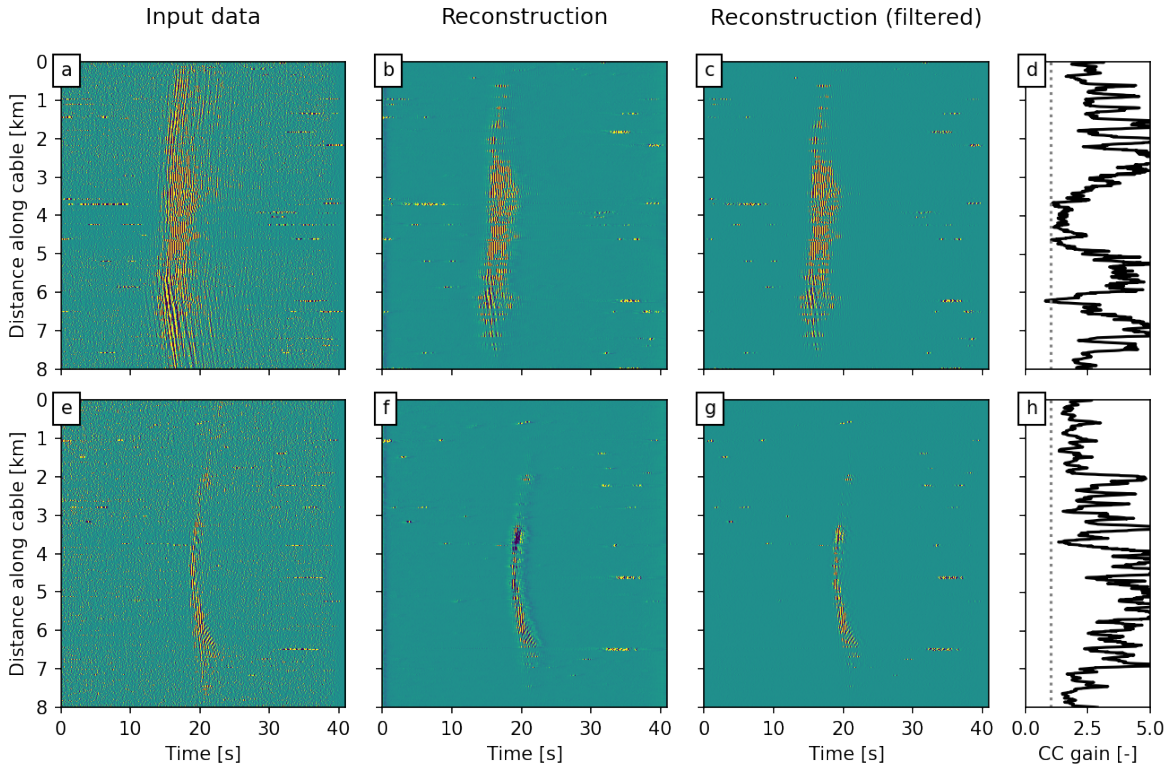


Figure 4.16. *J-invariant filtering results of Iceland data.*

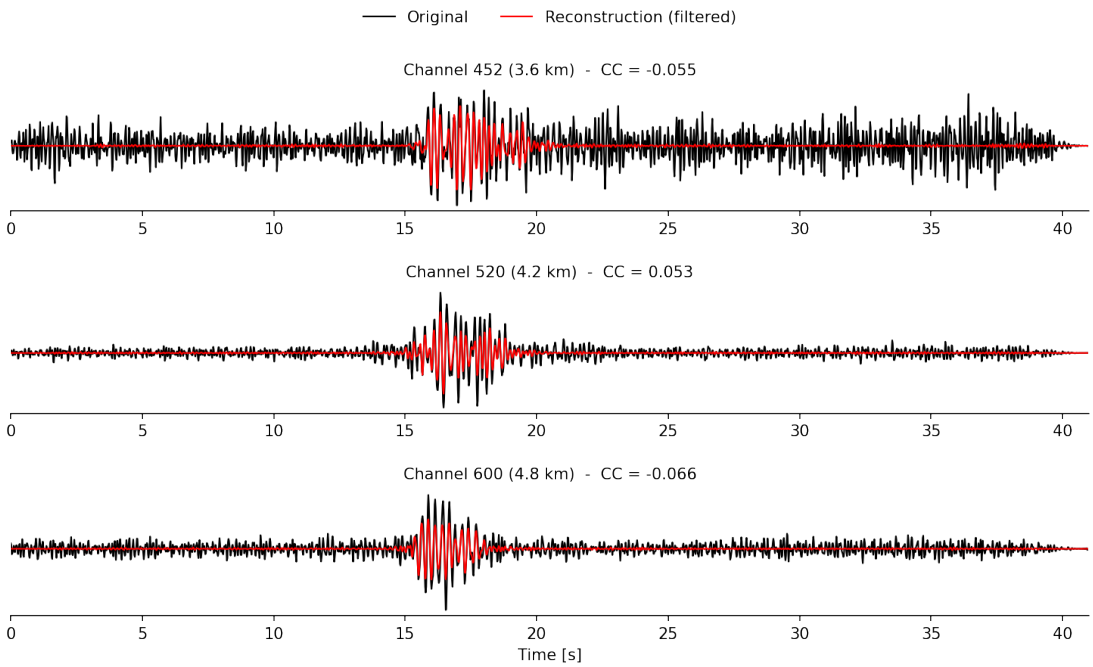


Figure 4.17. *Wiggle-for-wiggle comparison along the Iceland cable. The waveforms correspond with panels a,b,c of 4.16.*

4.3.3 Santorini

We continue our analysis, by testing our model in another DAS dataset, from another experiment in Greece, that took place in the island of Santorini. For this experiment, the same Silixa interrogation unit as in the Iceland dataset was used, with parameters such as gauge length and channel spacing remaining the same, at 10m and 8m respectively, while the sampling rate was set at an impressive 1kHz, one of the highest sampling rates in DAS experiments. As we also mentioned in section 3.2.3, this 44.5 kilometer cable is being used for telecommunications.

For our analysis, we selected 14 events that happened in the 19th and the 20th of October in 2021, that were catalogued from the seismology lab of the University of Athens[1] Information regarding the magnitude, depth, distance from the start of the fiber optic cable, date/time and location of the events are available on table 4.2.

Origin Time (GMT)	Depth (km)	Magnitude	Figure	Location
10/19 05:32:34.590	30.0	6.0	4.21 (Upper)	147.0 km SE of Karpathos
10/19 12:32:34.490	8.0	2.5	4.21 (Lower)	15.2 km NE of Thira
10/19 20:01:59.790	19.0	3.6	4.23 (Upper)	52.1 km SSE of Thira
10/19 20:08:12.770	14.0	2.9	4.23 (Lower)	52.4 km SSE of Thira
10/19 20:41:54.270	7.0	2.6	4.25 (Upper)	54.3 km SSE of Thira
10/19 20:43:13.910	12.0	3.0	4.25 (Lower)	55.2 km SSE of Thira
10/19 20:53:46.600	22.0	2.8	4.27 (Upper)	54.9 km SSE of Thira
10/19 20:59:17.640	7.0	1.8	4.27 (Lower)	28.1 km NNE of Thira
10/19 21:17:59.950	10.0	2.0	4.18 (Upper)	47.2 km S of Thira
10/19 21:35:24.030	13.0	2.6	4.29(Upper)	53.4 km S of Thira
10/19 21:47:09.760	7.0	3.3	4.29 (Lower)	23.4 km SE of Iraklion
10/20 01:29:35.740	7.0	3.4	4.18 (Lower)	24.4 km SE of Iraklion
10/20 02:44:05.280	8.0	4.3	4.31 (Upper)	23.8 km SE of Iraklion
10/20 03:50:53.280	3.0	3.1	4.31 (Lower)	23.9 km SE of Iraklion

Table 4.2. *Information regarding the seismic events on Santorini DAS.*

We start our qualitative analysis of the J-invariant reconstructions on figure 4.18. We can clearly see that the quality of the reconstructions remains at a satisfactory degree, while not like in the other datasets. We also see that the local waveform coherence gain, computed as that of the reconstructed image over the original one, is only fluctuating around the value of 1. This is a problem that we encountered in all of the events of this dataset that seemed strange at first. The only logical explanation we can give about the fact that we don't have any coherence gain, is that in this dataset, the gauge length is not equal to the channel spacing. This means, that interrogating unit averages the measurements taken every 10 meters (gauge length) while in the data we have defined that one channel is 8 meters away from the next and the previous one. This means, that in this 2 meter difference, the channels are overlapping which results in information leakage from on channel to the surrounding ones. That explains the fact that if we try and visualize neighbouring channels, one can barely make a difference between the two

[1]Link available here.

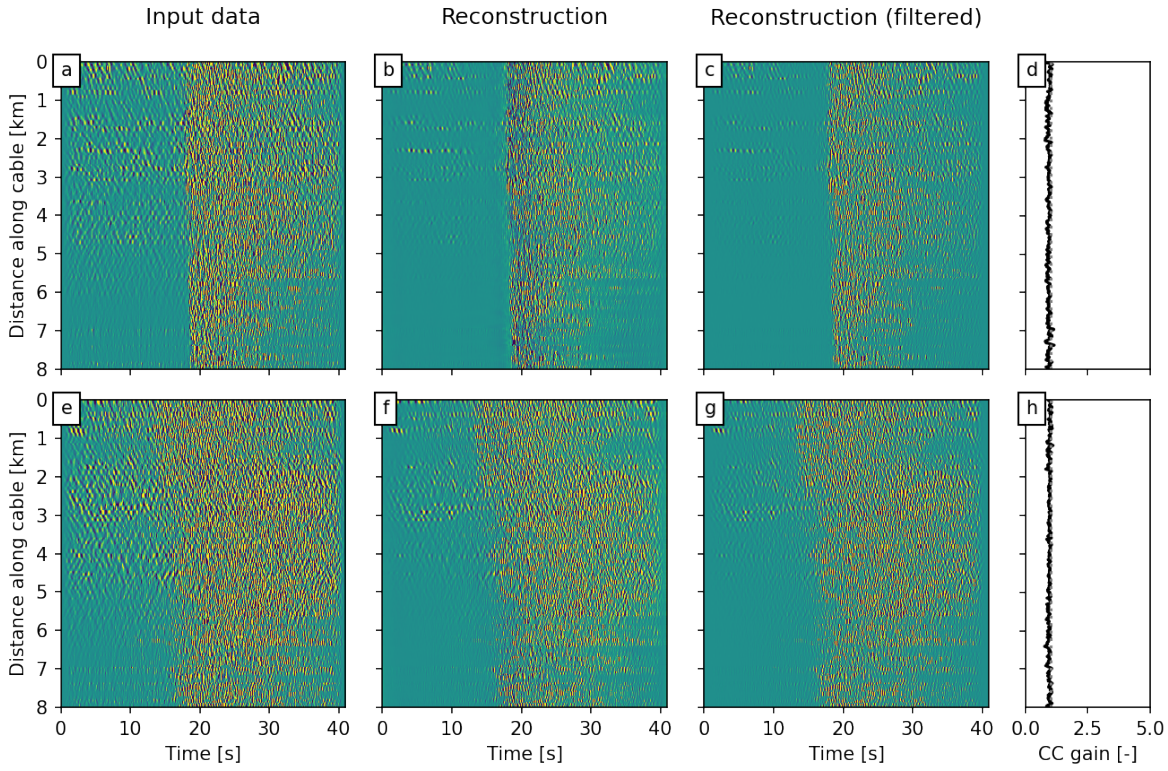


Figure 4.18. *J-invariant filtering results of Santorini data.*

waveforms. The fact that the gauge length is no equal to the channel spacing, is also true for the Iceland dataset, but the alarming difference between the two datasets is that in Iceland the initial sampling rate was set at 200 Hz, while in Santorini it was at 1 kHz, so 5 times more granular. This difference in sampling rate, might be an amplifying factor in the data leakage problem.

This means, that if neighboring channels are really similar, by calculating the moving coherence we expect to see that they are highly coherent with each other. In order for the seismic events not to interfere with what we aim to achieve here, we select a 60 second period where no events take place, so just the background noise, and we calculate the moving coherence for the 1000 channels of interest, for Santorini, and also for Iceland data for comparison. As is evident on figure 4.19, the Santorini data exhibit much higher coherence in between channels than the Iceland data. Therefore, it is almost impossible for our denoising algorithm to contribute to the coherence gain. A higher coherence gain would mean that the channels of the reconstructed waveforms are nearly identical between them.

The most probable cause for this higher coherence in between the DAS channels in the Santorini dataset, is the fact that it is an underwater cable, and while the HCMR/NESTOR datasets are also underwater cables, the circumstances can be different. Authors of [51] showed that underwater DAS earthquake recordings are dominated by Scholte-waves (surface waves that are propagated at an interface between a fluid and an elastic solid

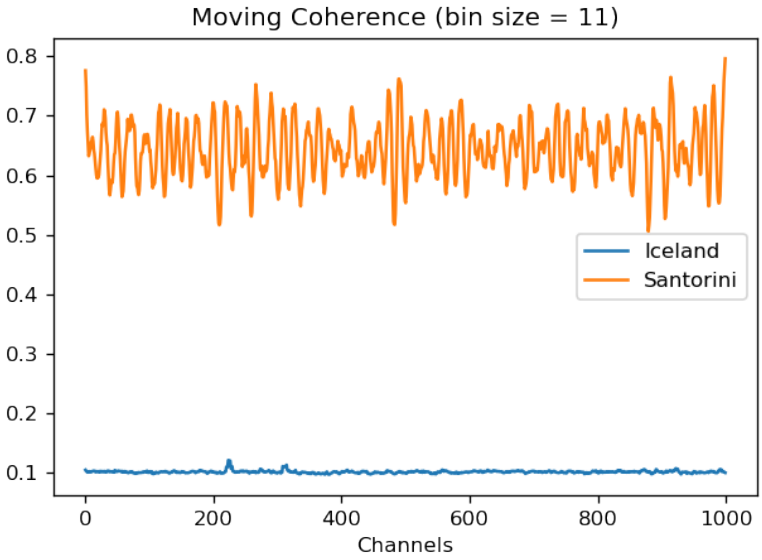


Figure 4.19. *Moving coherence versus channels for Iceland and Santorini data.*

medium, such as the cable between the sand and the water), indicating that acoustic and seismic waves are converted and scattered at the ocean—solid-earth interface. Moreover, interactions between the water-column and solid-earth generate several noise sources, i.e., surface gravity waves and microseisms, which constitute coherent noise that could affect earthquake monitoring with underwater DAS measurements. Therefore, this kind of noise exhibits coherent characteristics, basically violating our $\mathcal{J}$-invariance assumptions that the noise does not exhibit coherent characteristics and is statistically independent.

In order to further demonstrate that our algorithm actually helps denoise the events from the Santorini DAS data, we attempt to make a wiggle for wiggle comparison for the higher event of figure 4.18. As is evident on figure 4.20, our J-invariant reconstructions manage to suppress the background noise leading to the start of the event.

We move forward with another set of events. This time, of less denoising significance, as one of the two events was a stunning magnitude 6 earthquake, that took place more than 300 kilometers away from our cable. This means, that such a big event, causes high amplitudes in the waveforms, that the background noise is a few orders of magnitude lower than the peaks of the waveform, so it is barely visible. The second event on figure 4.21 is a magnitude 2.5 earthquake, but this took place only 15 kilometers North East of Thira (Santorini), meaning that due to its close proximity to our DAS cable, it was captured with similar characteristics as the M=6 event.

For both events, qualitatively we can see that there is a hint of background in the input waveforms, that were removed in the reconstructions. Of course, one does not need to perform denoising to see these events, but it is interesting to see how our model behaves in a plethora of situations.

As usual, we continue with a wiggle-for-wiggle comparison, on figure 4.22. We notice that the amplitudes are matched to a satisfying degree, also taking note that the algorithm

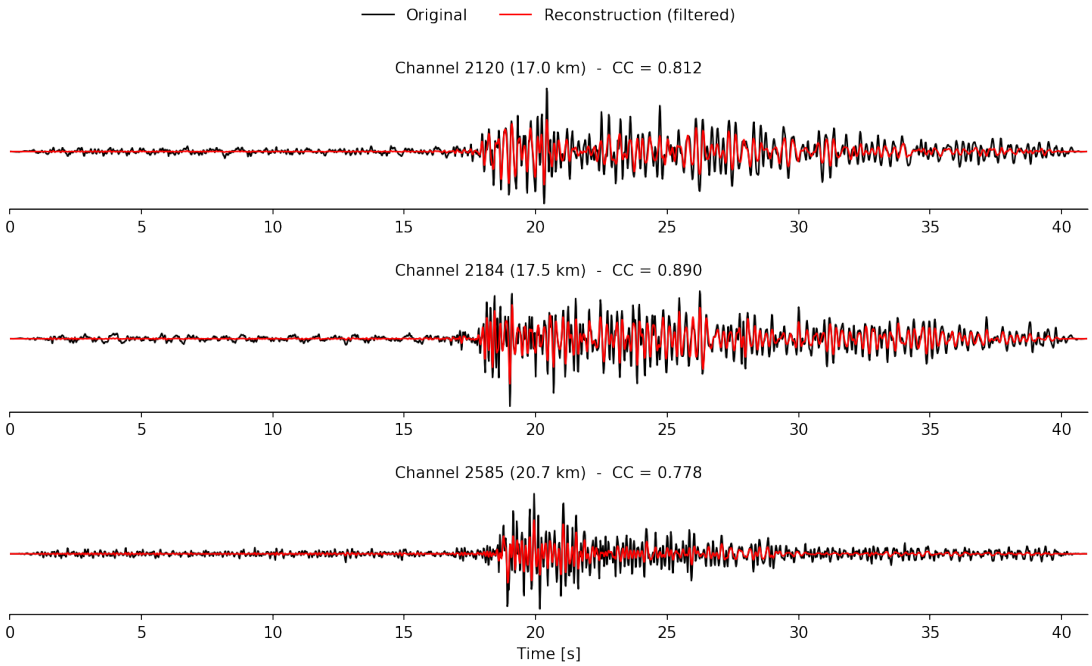


Figure 4.20. *Wiggle-for-wiggle comparison along the Santorini cable. The waveforms correspond with panels a,b,c of 4.18.*

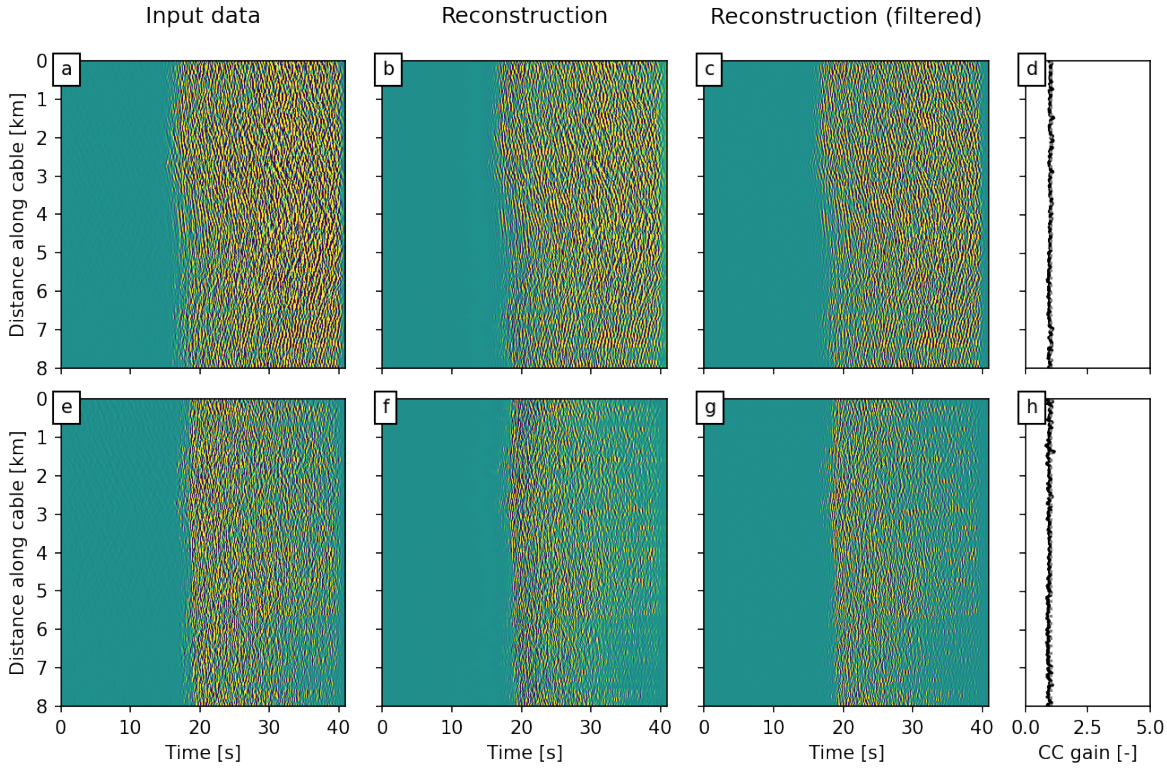


Figure 4.21. *J-invariant filtering results of Santorini data.*

was not trained to such massive events, and also that when removing the noise, the contribution of the noise is also removed from the waveform peaks that were caused by the event. Also, quantitatively, we can see that the correlation coefficients fall in the right range.

We notice that in this dataset, since the cable was placed on the ocean floor, all of the events captured were with high magnitude. Similarly, we present the reconstructions of

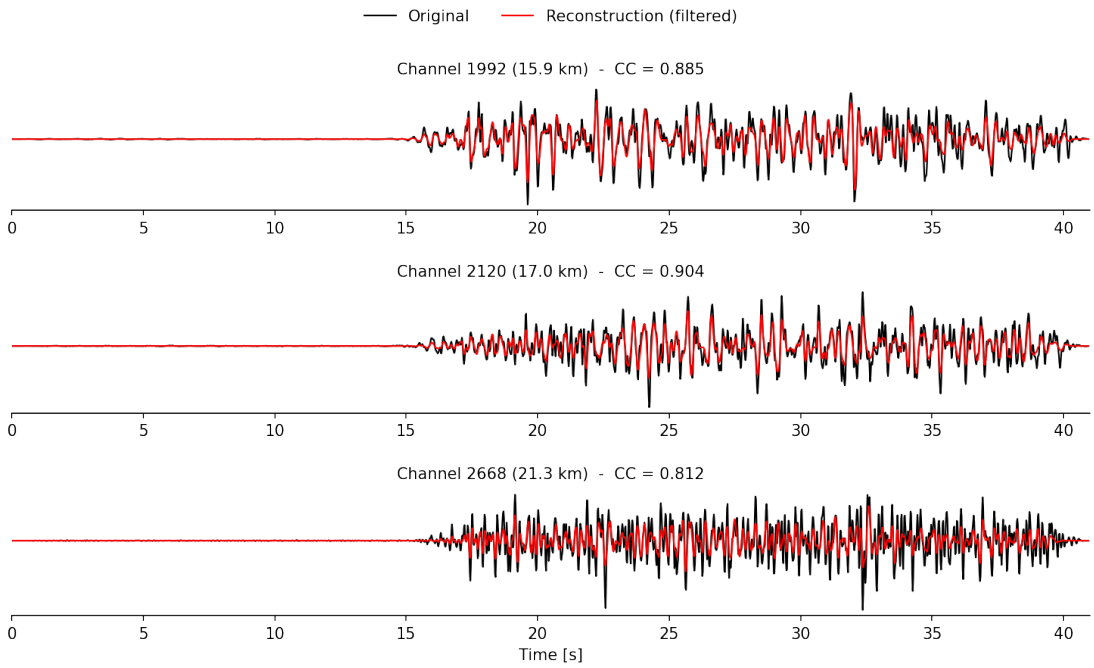


Figure 4.22. *Wiggle-for-wiggle comparison along the Santorini cable. The waveforms correspond with panels a,b,c of 4.21.*

the rest of the events.

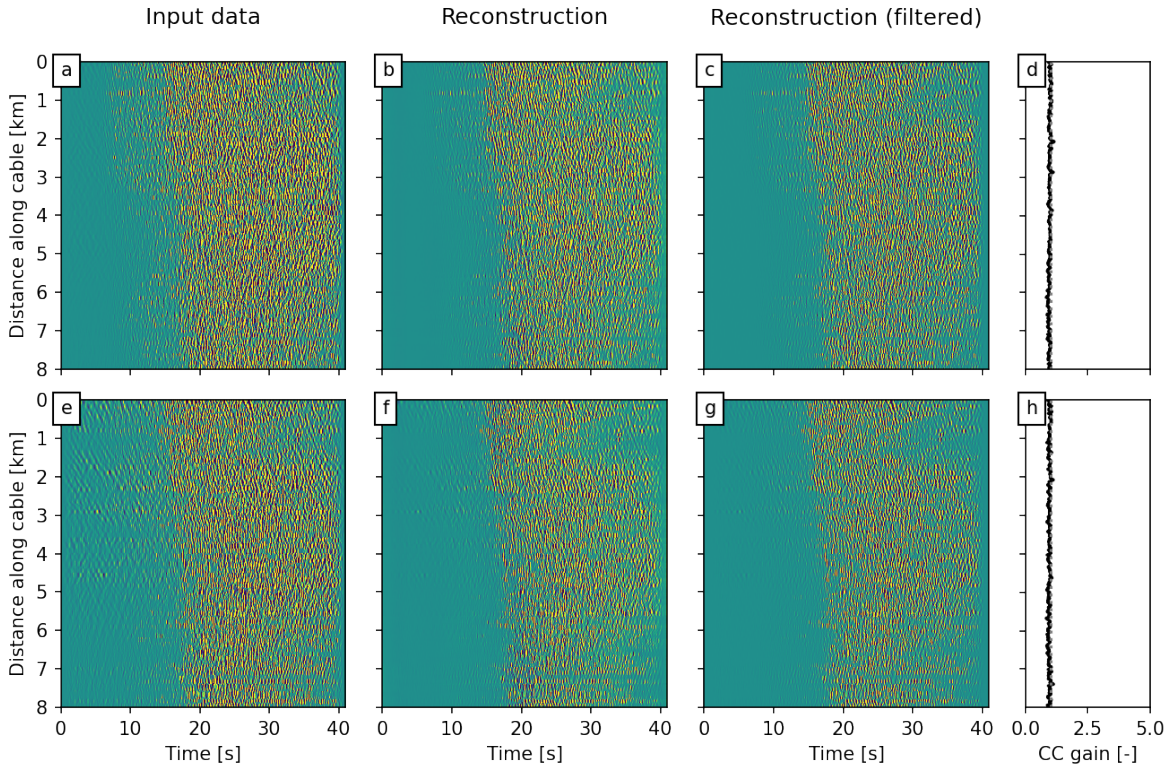


Figure 4.23. *J-invariant filtering results of Santorini data.*

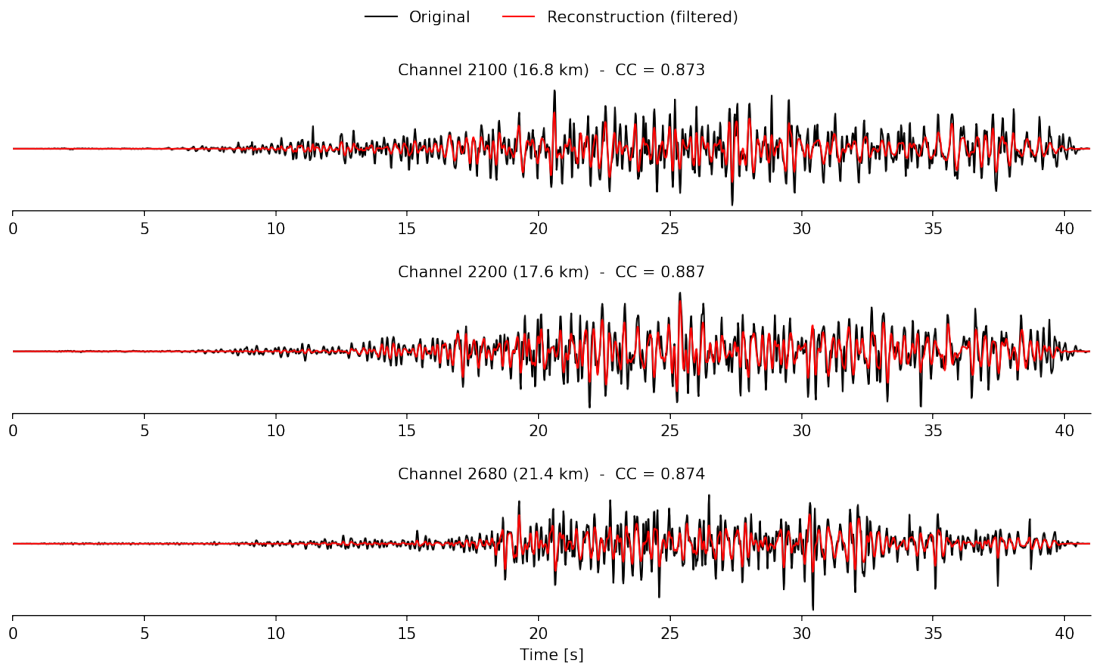


Figure 4.24. *Wiggle-for-wiggle comparison along the Santorini cable. The waveforms correspond with panels a,b,c of 4.23.*

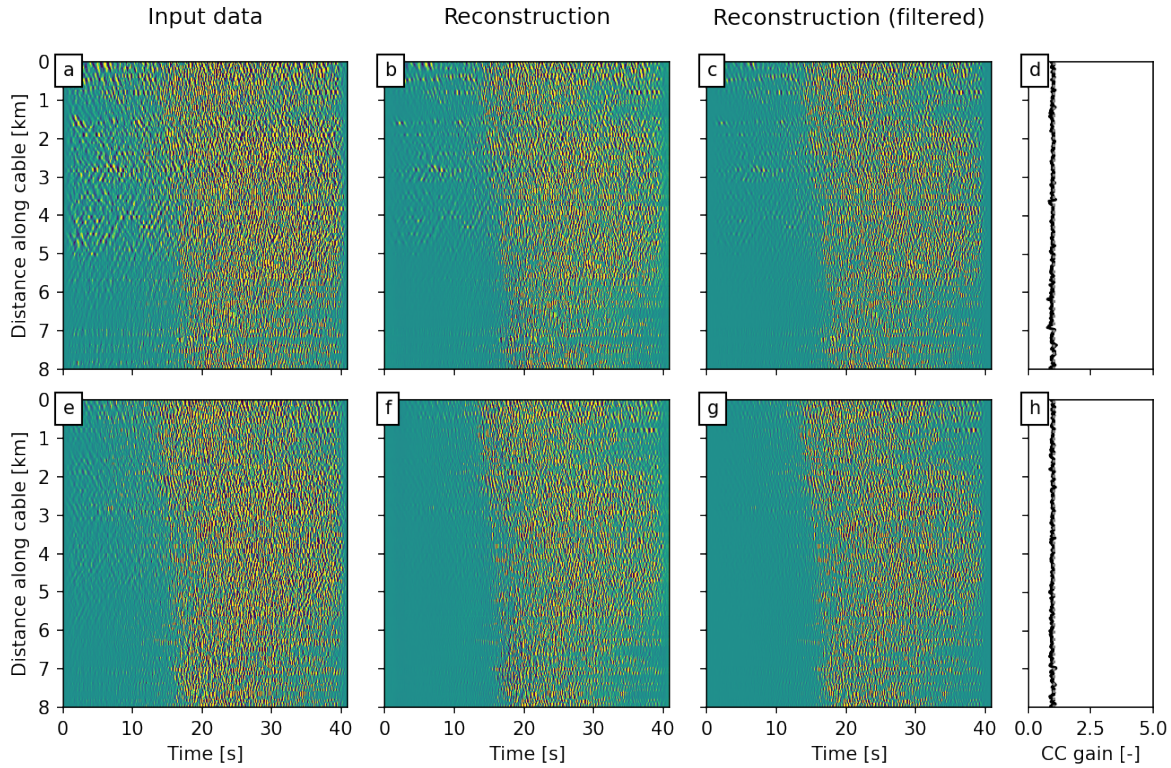


Figure 4.25. *J-invariant filtering results of Santorini data.*

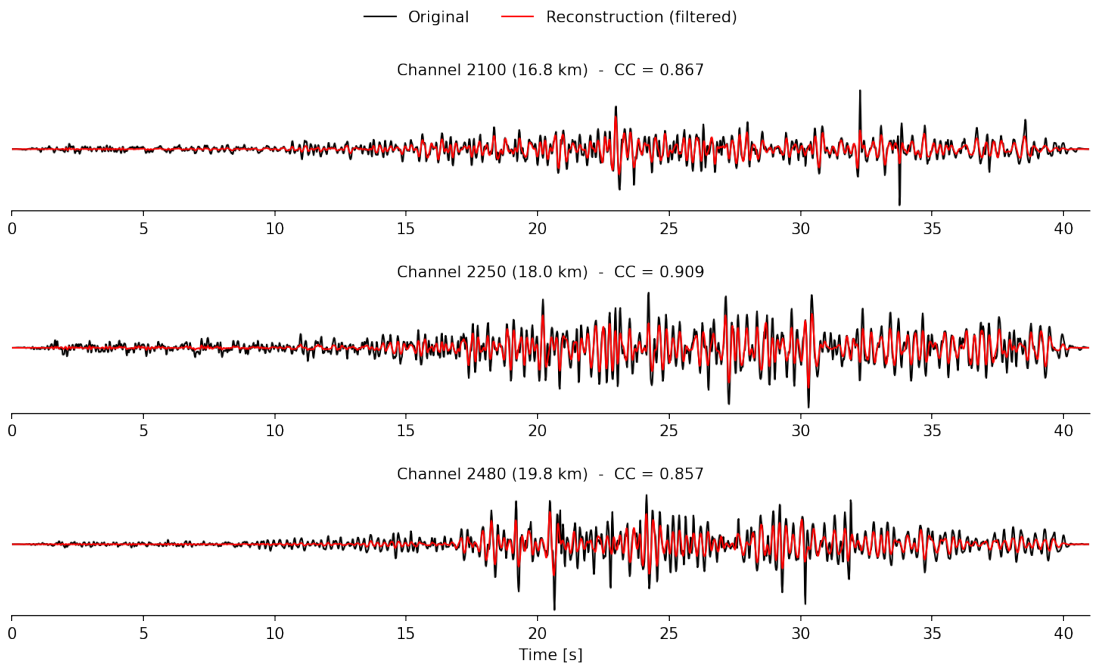


Figure 4.26. *Wiggle-for-wiggle comparison along the Santorini cable. The waveforms correspond with panels a,b,c of 4.25.*

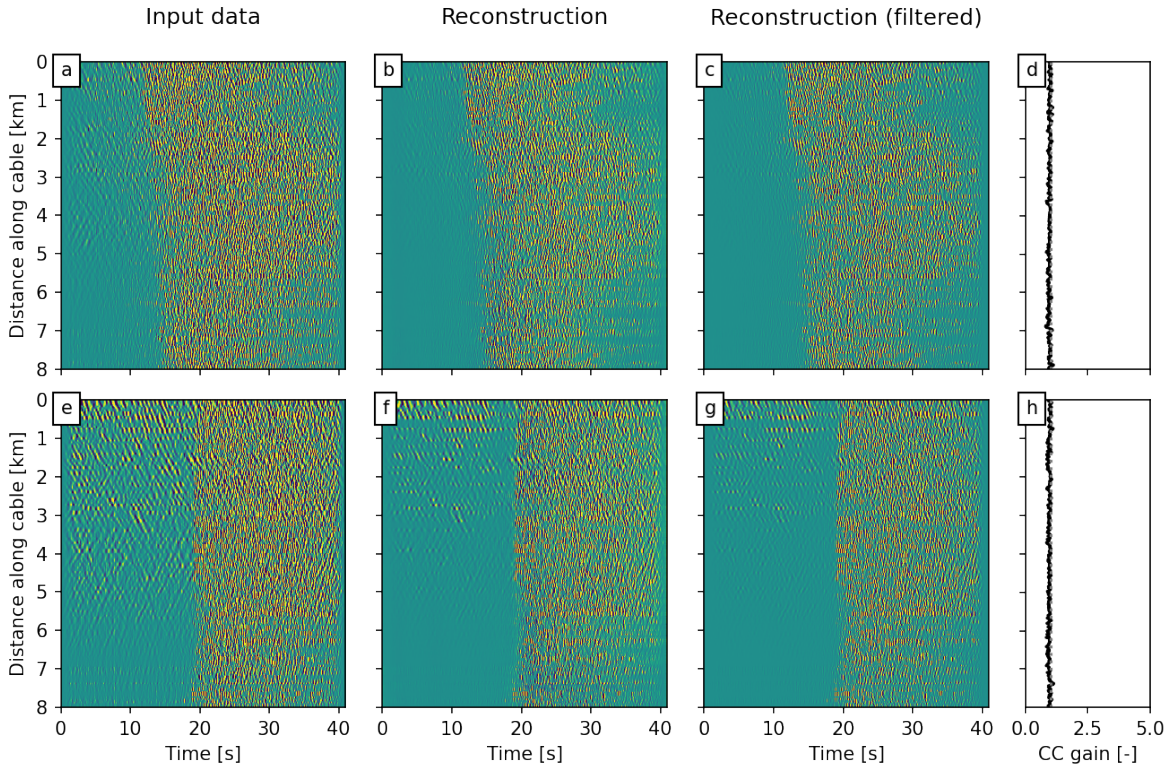


Figure 4.27. *J-invariant filtering results of Santorini data.*

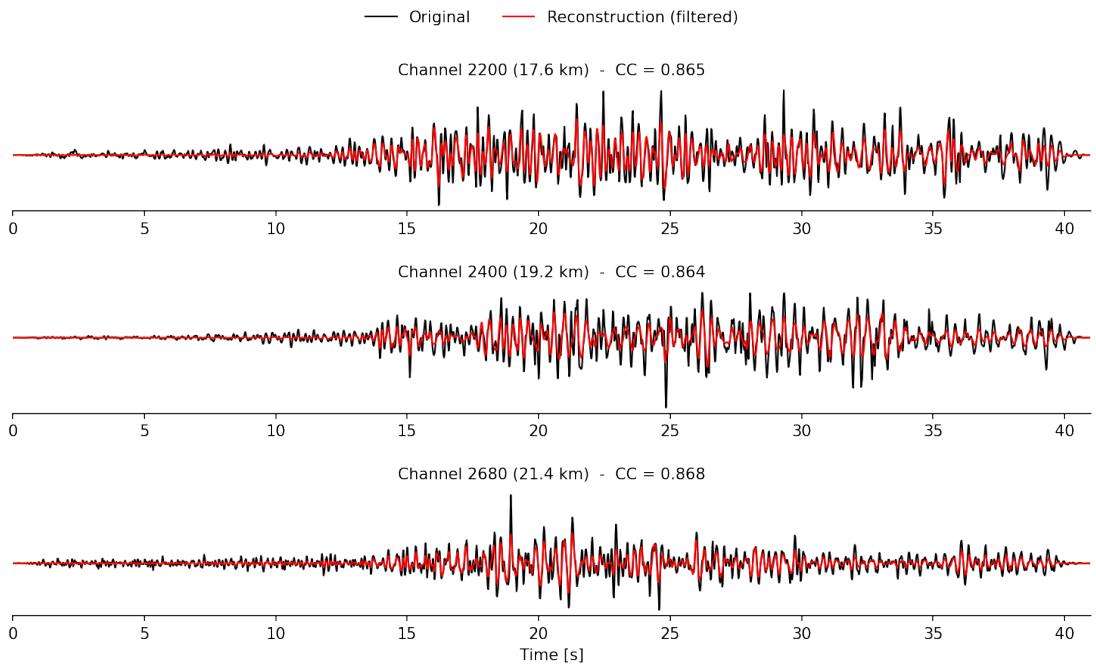


Figure 4.28. *Wiggle-for-wiggle comparison along the Santorini cable. The waveforms correspond with panels a,b,c of 4.27.*

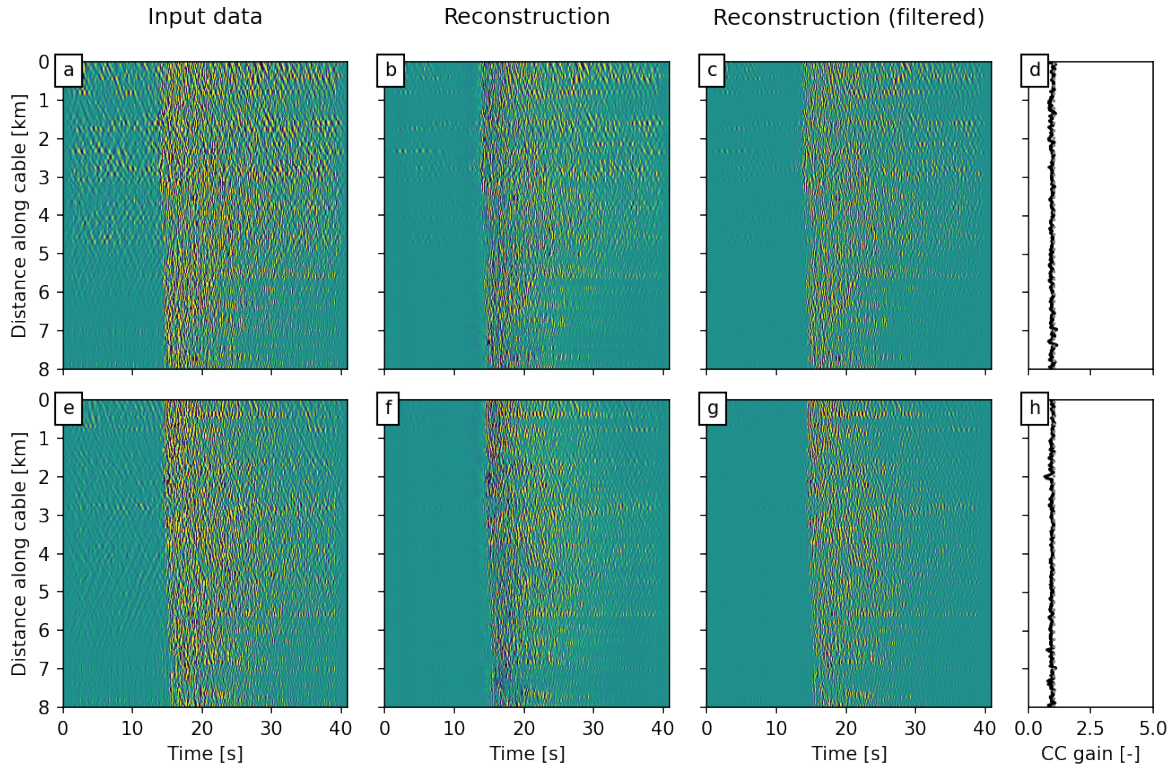


Figure 4.29. *J-invariant filtering results of Santorini data.*

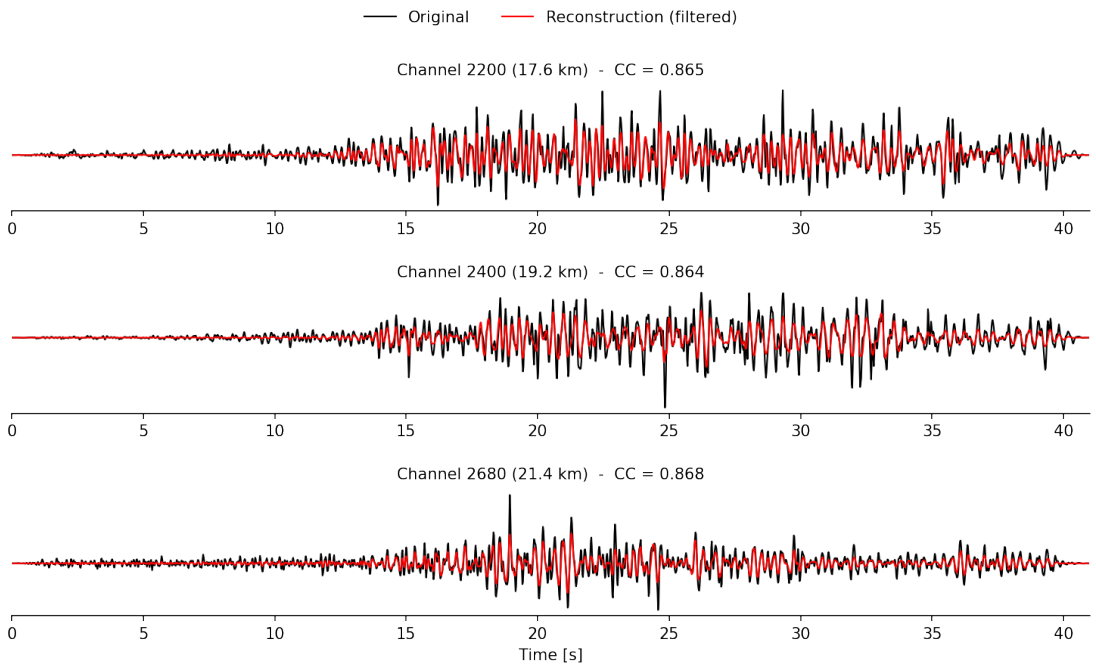


Figure 4.30. *Wiggle-for-wiggle comparison along the Santorini cable. The waveforms correspond with panels a,b,c of 4.29.*

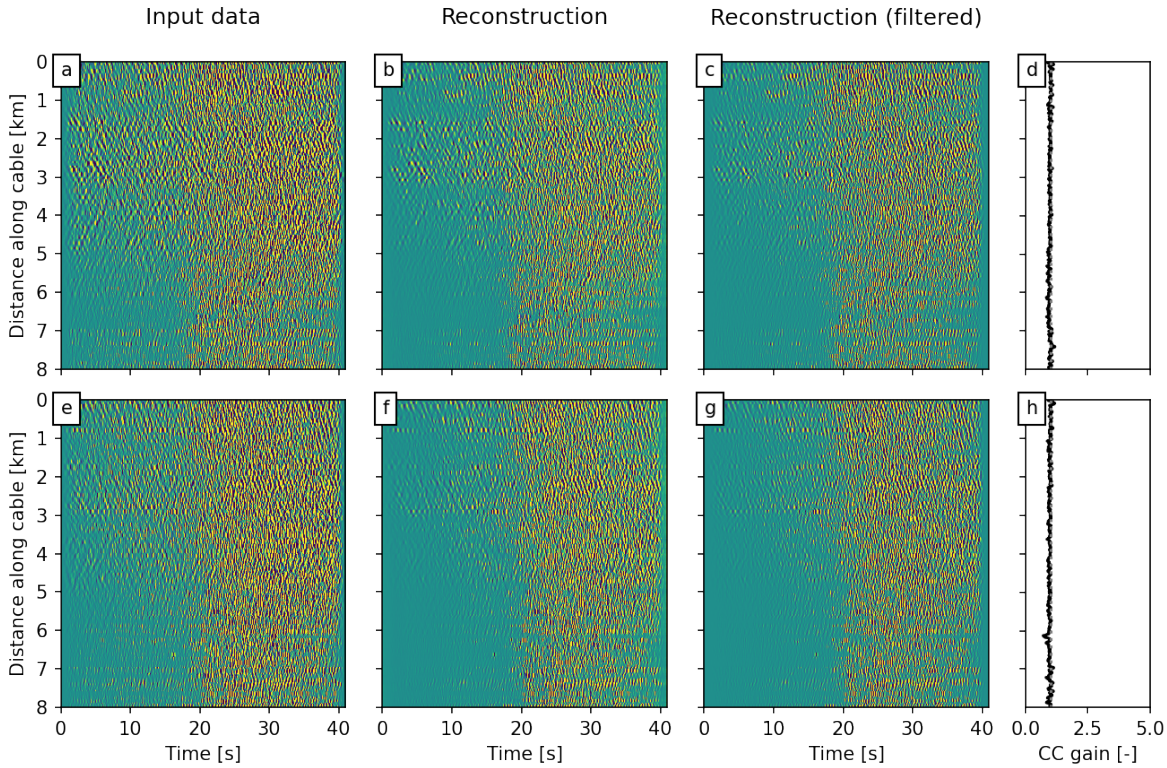


Figure 4.31. *J-invariant filtering results of Santorini data.*

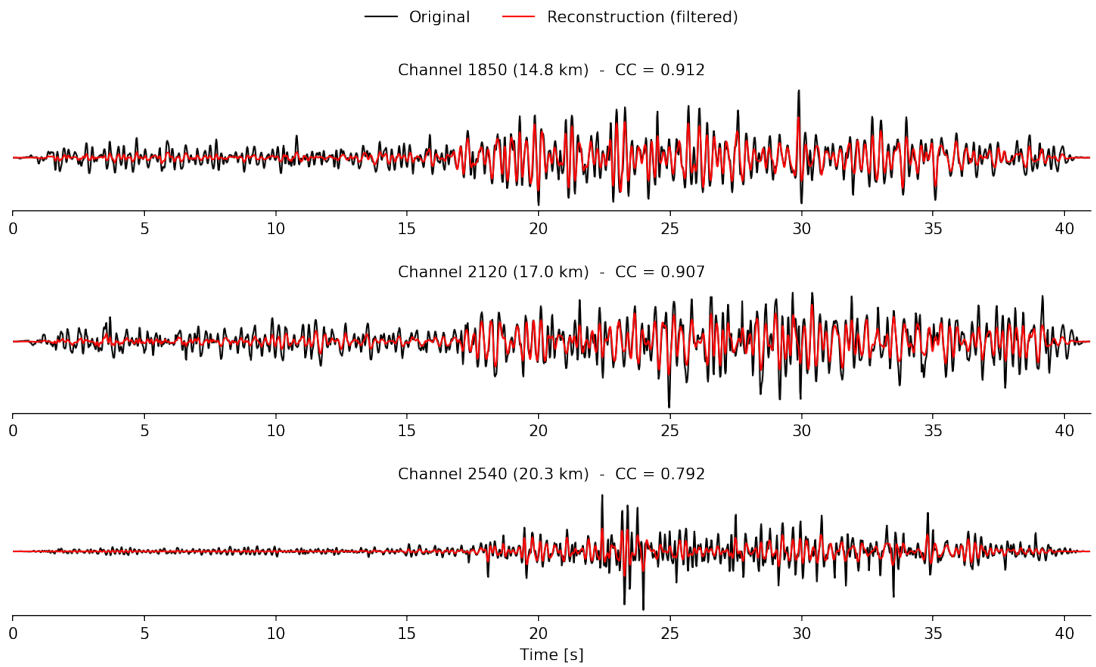


Figure 4.32. *Wiggle-for-wiggle comparison along the Santorini cable. The waveforms correspond with panels a,b,c of 4.31.*

4.4 Comparison with traditional filtering methods

Since our proposed filtering approach for DAS data is closely related to image denoising and enhancement, we compare our method with two commonly used image processing techniques, namely non-local means (NLM) [52] as implemented in scikit-image [53] and block-matching 3D-transform shrinkage (BM3D) [54]. These two non-learning algorithms also served as a benchmark for the study of noise2self [32].

In simple words, the NLM method is based on a simple principle: replacing the color of a pixel with an average of the colors of similar pixels. But the most similar pixels to a given pixel have no reason to be close at all. It is therefore licit to scan a vast portion of the image in search of all the pixels that really resemble the pixel one wants to denoise.

BM3D is a denoising method introduced in 2015, that is based on the fact that an image has a locally sparse representation in transform domain. This sparsity is enhanced by grouping similar 2D image patches into 3D groups. Collaborative filtering is the name of the BM3D grouping and filtering procedure. It is realized in four steps: 1) finding the image patches similar to a given image patch and grouping them in a 3D block 2) 3D linear transform of the 3D block; 3) shrinkage of the transform spectrum coefficients; 4) inverse 3D transformation. This 3D filter therefore filters out simultaneously all 2D image patches in the 3D block.

For this comparison, we select 3 events recorded by the three aforementioned datasets we provided the results for, HCMR (4.7), Iceland (4.13) and Santorini (4.29), and their respective comparisons on figures: 4.33, 4.36 and 4.34. We select each of these events, and process the data with NLM and BM3D using a noise variance from the first 20 seconds of data. We then compare the denoised data with our J-invariant reconstruction after bandpass filtering. To be fair with the other methods' reconstructions, we also bandpass filter the outputs from NLM and BM3D.

For a more detailed visual comparison, we focus on a shallow segment of the cable that recorded many fine-detailed features. While these low-amplitude details (e.g. in figure 4.33) do not significantly contribute to other metrics like SNR or MSE loss, they are critical for the detection of body wave arrivals. DAS is strongly sensitive to surface waves, both owing to their horizontal inclination and slow phase velocity, and exhibits much less sensitivity to body waves. As a result, P- and S-waves are recorded as comparatively low amplitude features, and so the recovery and preservation of these is important.

This effect is less apparent on Iceland data (figure 4.36), as in that dataset the SNR is significantly higher, and background noise is at low levels.

When comparing the performance of the various denoising methods, it is immediately clear that our proposed method preserves fine-grained details with much higher fidelity than the BM3D and particularly the NLM method. In terms of the performance on the high-amplitude wave train, NLM and BM3D essentially copy the input without altering the amplitudes, which is questionable given that these are affected by the same noise levels as prior to the arrival of these waves. Moreover, our method processes the Santorini event of figure 4.34 in 9.43 seconds, while for the rest of the methods it takes 13.95 seconds for NLM and 89.55 seconds for BM3D, also in the time comparison plot in figure 4.35. These

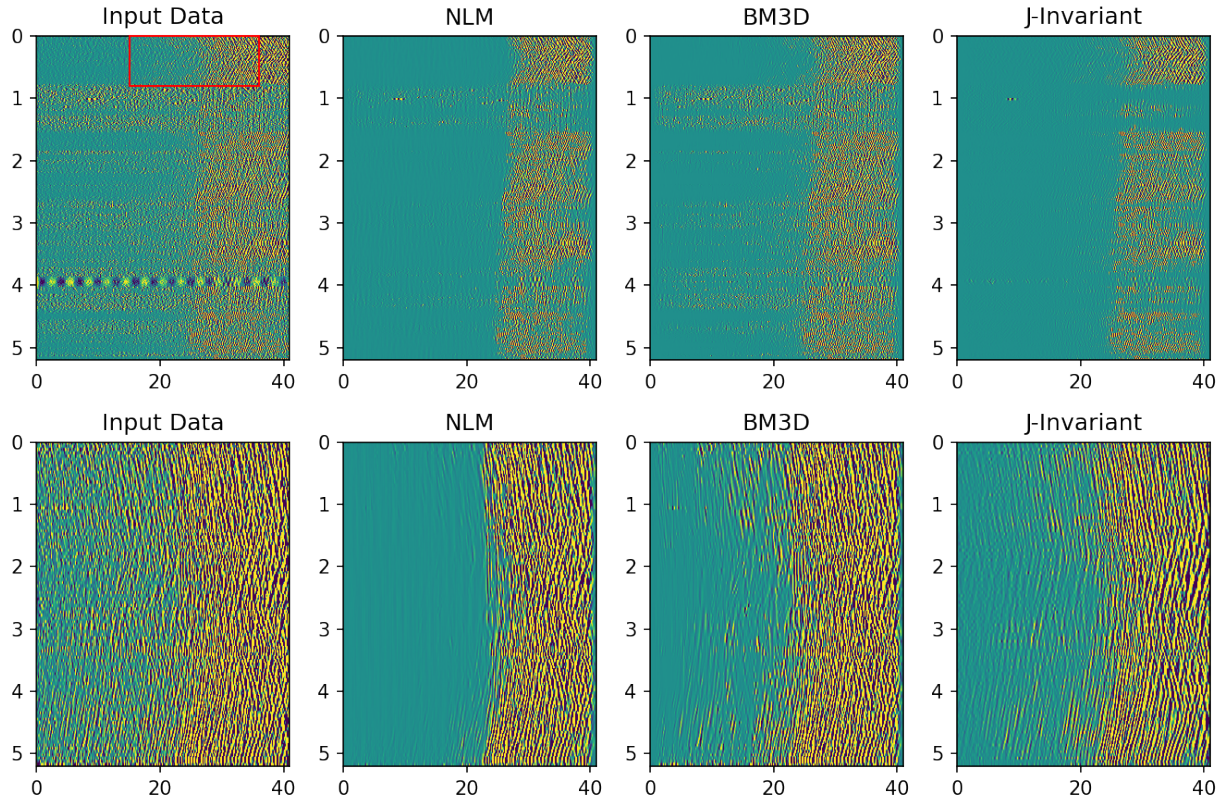


Figure 4.33. *Comparison of our proposed method with two conventional image denoising methods on HCMR data. Event Corresponds with 4.7.*

speed gains, along with the enhanced precision of the method render our J-invariant denoiser superior to traditional image denoising methods.

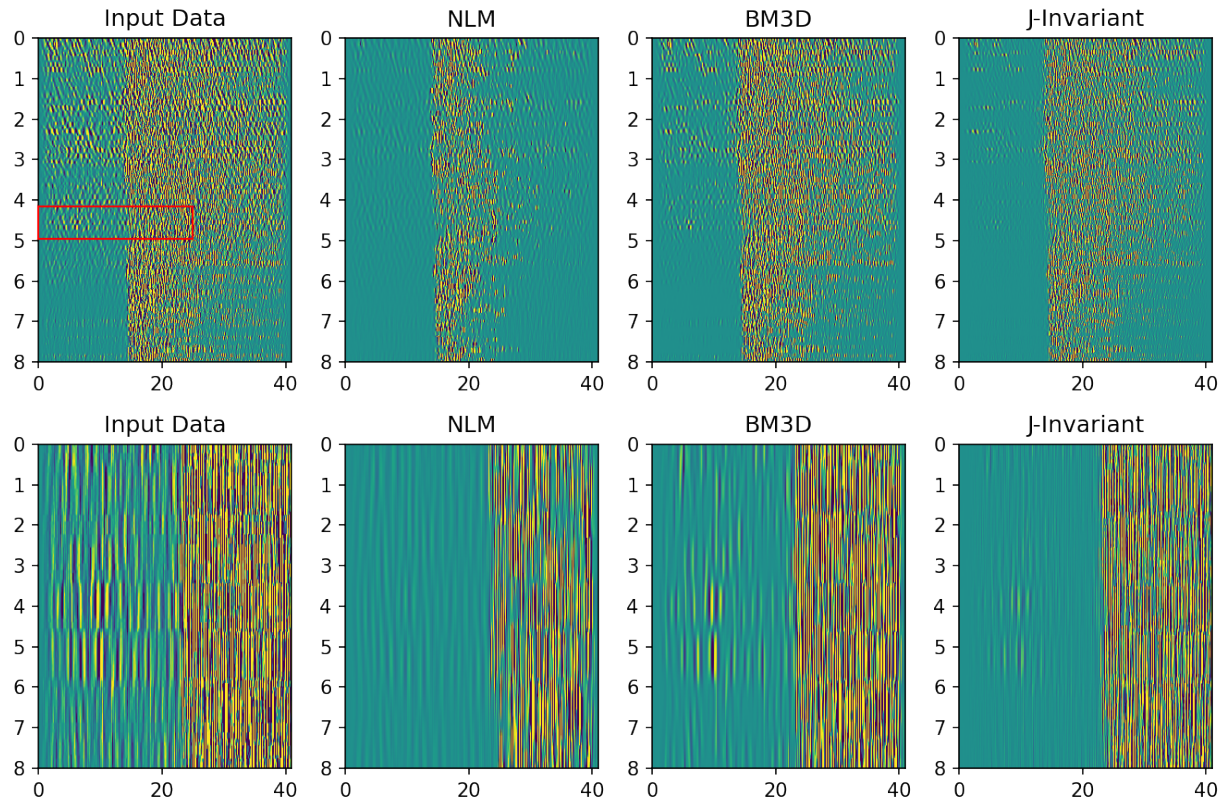


Figure 4.34. *Comparison of our proposed method with two conventional image denoising methods on Santorini data. Event Corresponds with 4.29.*

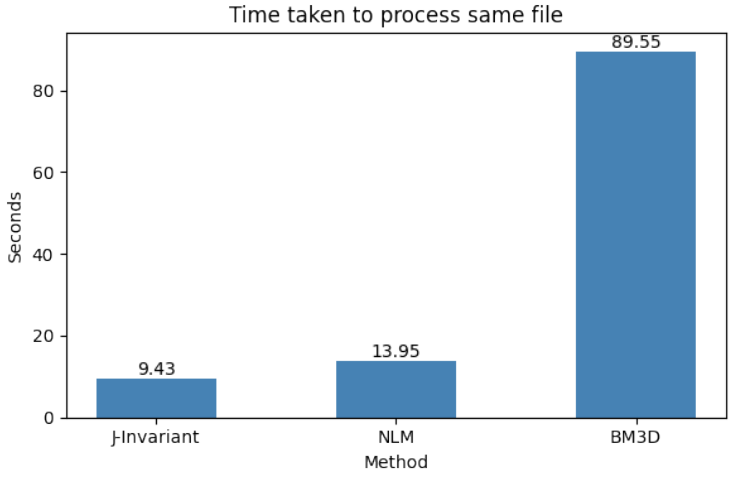


Figure 4.35. *Time taken to process the file of figure 4.29.*

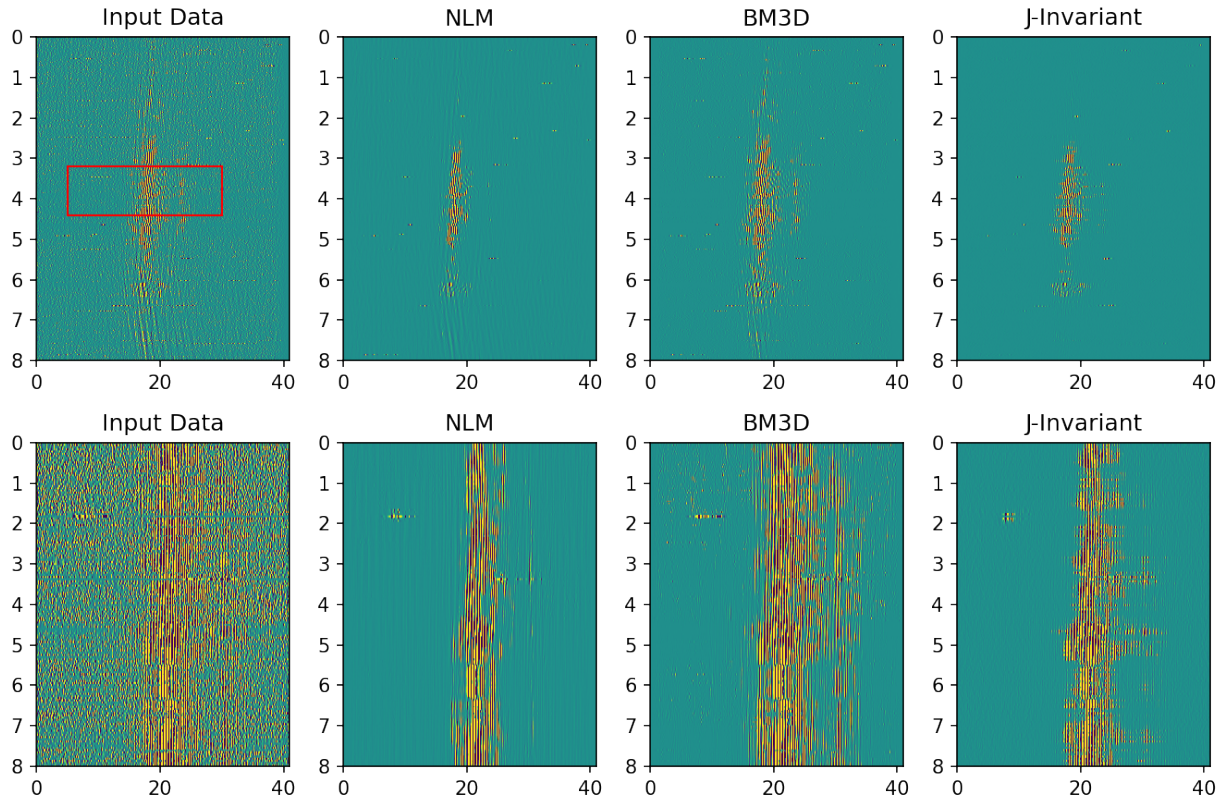


Figure 4.36. *Comparison of our proposed method with two conventional image denoising methods on Iceland data. Event Corresponds with 4.13.*

Chapter 5

Discussion

5.0.1 Reproducibility

The code that produced all the plots and the pretrained models that we used, are widely available at this GitHub repository. While in some cases the pretrained model provided in the GitHub repository will perform reasonably well out of the box, in practice, retraining the model is recommended, particularly in the following scenarios:

- A new experiment was conducted (e.g. in a different location, or with different interrogator settings like the gauge length or sampling rate).
- The conditions during one experiment strongly vary. This can be the case when new noise sources are introduced (construction works on-land, microseismic noise, etc.), or when the signal-to-noise ratio significantly changes.
- One particular event of interest, such as a major earthquake, occurs. In this case the model can be trained in a single sample; instead of training on a large data set and optimising the model parameters for a (potentially) wide data range, the training is done on a very specific data range. Consequently, the model will try to achieve the best denoising performance for this specific data set, at the cost of generalisation. This practice was also followed in noise2self [32], where the authors performed training in a single image at a time.

In order to achieve the levels of denoising we showed in the previous section, the data are required to be in a specific format, that can be achieved by performing the preprocessing steps we mentioned earlier. Specifically, the data are assumed to be Euclidean, meaning that the spacing between each sample is constant in both space and time (constant gauge length and time-sampling frequency).

The pretrained model was trained on a dataset that was bandpass filtered in a 1-10 Hz pass band, and was sampled at 50 Hz with a gauge length (and channel spacing) of 19.2 m. For data with a different gauge length, retraining of the model is recommended, though not required, as we didn't do any retraining in the Santorini and Iceland datasets, as there the gauge length was not equal to channel spacing, thus it would not be successful. However, retraining is not necessary if the ratio of the frequency pass band to the Nyquist frequency remains fixed. To give an example, a pass band of 1-10 Hz sampled at 50 Hz

is the same as a pass band of 5-50 Hz sampled at 250 Hz. A last requirement is that the DAS data be approximately normalised before being passed onto the neural network.

5.0.2 Applications

The Distributed Acoustic Sensing method generally generates large volumes of data, of the order of terabytes per fibre per day. To be exact, the Santorini dataset that has the highest sampling rate and hence the largest file sizes, produces one 326.251 MegaByte file every 30 seconds, meaning 940 GigaBytes per day.

With DAS being a data-driven approach, Deep Learning methods are ideally suited for automated processing and analysis of DAS data. However, since DAS is still an emerging technology, large labelled datasets are lacking, hence prohibiting any supervised learning approach. In all three datasets we used in this work, all the events were selected by hand either by visual examination of the images (e.g. Iceland), or by looking at specific dates and times where we have information regarding events from earthquake monitoring websites (e.g. Santorini). Because this solution is of course not scalable and cannot be automated, the possible ways such a problem can be tackled is either by creating labelled data, (e.g. with computer simulations [55]), or by applying an unsupervised or self-supervised learning method.

The J-invariant filtering approach that we detailed in this study is entirely self-supervised and, after pretraining on synthetic data, can be trained using only a small dataset of recorded events, to capture the statistical properties of the specific experiment. This opens up a plethora of application in seismology and DAS signal analysis. First and foremost, it improves the performance of conventional earthquake detection algorithms such as the commonly used STA/LTA method [11], or of newer ones such as [49] that uses computer vision to detect events. Taking an extreme example, as we mentioned in the results section on figure 4.4 (f and g), for an event whose biggest portion is corrupted by noise it would be nearly impossible for a conventional algorithm such as those we just mentioned to detect the earthquake signal that is completely obscured by noise. Particularly for the analysis of microseismicity, the improved SNR after J-invariant filtering would massively improve catalogue completeness, provide better earthquake location estimates, and draw a more complete picture of the evolution of seismicity in time and space. Such improved SNR could not be obtained by single-waveform frequency-based methods, since the signal of interest and the background noise share a common frequency band.

A second application of the method pertains to waveform coherence based methods like template matching [56] and beamforming [57]. In template matching a given time-series is analysed for the occurrence of a previously recorded and identified signal (the template). By cross-correlating the time-series with the template in a sliding time window, events similar to the template can be identified through a high correlation coefficient above some predefined threshold. This approach has recently been applied successfully to DAS data [58], detecting numerous small earthquakes induced by geothermal energy extraction operations. Out of the 116 detections, 68 were identified within the level of the background noise, demonstrating the sensitivity of the template matching technique.

It is possible that by preprocessing the data with our proposed J-invariant filtering approach many more events could be identified, not only due to a higher SNR of the target timeseries, but also due to a higher SNR of the template waveform. The improved SNR of both cases would lower the threshold above which a detection is deemed significant. Inherent to the template matching method, a detection automatically provides a rough location estimate as well, helping to rapidly build (micro-)earthquake catalogues from DAS experiments. All of the above also applies to the detection of volcanic or tectonic tremor and very-low frequency earthquakes, two seismological features that often exist at or below the ambient noise level, similar to the events we demonstrated on the Iceland data, namely the micro event on figure 4.10.

Similar to template matching, seismic beamforming and backprojection relies on waveform coherence to assess the move-out of a seismic signal propagating across a seismometer or DAS array (see e.g. [56], [57], [59]). The quality of these analyses depends on the resolution with which the phase shift can be determined. This resolution may improve at higher signal frequencies, which in turn suffer from stronger attenuation and waveform decorrelation. There thus exists a trade-off between the beamforming/backprojection resolution and the coherence/SNR of the signal of interest, as a function of frequency. By employing J-invariant filtering, coherent signals at higher frequencies can be amplified, helping to shift the trade-off towards higher frequencies and to improve the resolution.

5.0.3 Limitations & Future Work

Aside from the numerous applications that may benefit from the proposed method, we acknowledge certain limitations and point out potential extensions of the method to guide future endeavours.

Firstly, the amplitudes of the earthquake signals are not always fully recovered. To some extent this is expected, as the model attempts to remove the contribution of the noise to the signal. However, for signals that are at or below the noise floor, the true amplitude of the signal cannot be reliably estimated. This can be seen in most of the wiggle-for-wiggle comparisons we performed in the previous section. The reason for this is simple: whether a signal's amplitude is 10 or 10000 times smaller than the noise level, the resulting superposition of signal plus noise looks the same, i.e. the coherent signal contributes negligibly to the overall signal. In this case of a very low SNR, it cannot be inferred what the original signal amplitude was, other than that it is upper-bounded by the noise level. Thus, one should exercise caution interpreting the amplitudes of J-invariant reconstructions of low SNR signals. For high SNR signals this seems to be much less of a problem, as evidenced by the low scaled residuals for high SNR samples on figure 4.5 (a), and also on the high quality reconstructions we have for the Iceland dataset that exhibits high SNR in general, as in figure 4.17.

Next, the underlying principle of our method relies on spatio-temporal signal coherence. Any parts of the signal that are incoherent will not be reconstructed and therefore be filtered from the input. This is useful for incoherent noise sources, given that the signals of interest are strongly coherent. However, it is common in (submarine) DAS

to also observe strongly coherent nuisance sources like ocean gravity waves and related phenomena. For many seismological applications these are considered part of the background noise, and are therefore not desired in the output. Our method does not address the separation of multiple coherent signals. However, once incoherent noise has been removed from the input, it may be easier to separate the remaining coherent signals by other means, as usually they contain known characteristics. This is especially the case with the Santorini dataset, as the noise there exhibits coherent characteristics.

Then, we applied our proposed method to a DAS array with constant spacing between the recording channels. Although the model makes no assumption regarding the geometry of the array (it applies to both straight and curved cables provided that the radius curvature is sufficiently large), it does implicitly require that the data are evenly distributed along the trajectory of the cable. This limits the application of the method to DAS arrays and, practically speaking, linear seismic arrays with constant inter-station distance. Fortunately, this limitation can potentially be circumvented by treating the array as a graph with the receiver station location as node attributes, or with station distances as edge attributes, and performing the learning task on the graph [58], [60].

Last but not least, the biggest assumption we made while training the data, is that the gauge length is equal to the channel spacing. While this is true for HCMR and NESTOR data, it is not true for the Santorini and Iceland datasets. This essentially prohibited us from training in these datasets as information from one channel is leaked to the neighbouring ones. This problem can be circumvented, by not picking nearby channels in the training dataset, hence limiting the amount of spatial information we have in the data, which is one of the biggest assets of DAS. While experimenting, this gave good reconstructions in the Iceland dataset, but it was not effective in the Santorini dataset, mostly because of the coherent characteristics of noise in that experiment, as we saw in figure 4.19, the neighbouring channels are highly correlated between them, even in large distances, since the bin size (i.e. the amount of channels that the moving coherence was calculated from) was set at 11. Also, a new metric to better quantify the quality of the reconstruction is also needed, because as we saw in the Santorini data, the high inter-channel coherence caused no CC gain uplift, despite the denoising results being satisfactory, meaning that the results weren't captivated in that metric, due to the noise being highly coherent.

A great step towards making this model better, would be replacing the tried and tested UNet architecture with a more modern one, such as Vision Transformers [61]. Recently, vision transformers (ViTs) have appeared as a competitive alternative to CNNs, yielding similar levels of performance; if not better. According to [35], regarding computer vision applications, while CNNs perform better when trained from scratch, off the-shelf vision transformers using default hyperparameters are on par with CNNs when pretrained on ImageNet, and outperform their CNN counterparts when pretrained using self-supervision. This essentially means that vision transformers have the capability to outperform CNN architectures, but are a lot harder to train. This promising architecture when combined with the large amounts of data DAS experiments produce, have the capability to overcome the limitations we stated above.

Bibliography

[1] N. J. Lindsey and E. R. Martin, "Fiber-optic seismology," *Annual Review of Earth and Planetary Sciences*, vol. 49, no. 1, pp. 309–336, 2021.

[2] T. Dean, T. Cuny, and A. Hartog, "Determination of the optimum gauge length for borehole seismic surveys acquired using distributed vibration sensing," 06 2015.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[4] "A Self-Supervised Deep Learning Approach for Blind Denoising and Waveform Coherence Enhancement in Distributed Acoustic Sensing data," Mar. 2021. Publisher: figshare.

[5] R. Zhang, "Making convolutional networks shift-invariant again," in *ICML*, 2019.

[6] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, p. 53, Mar. 2021.

[7] B. G. Gorshkov, K. Yüksel, A. A. Fotiadi, M. Wuilpart, D. A. Korobko, A. A. Zhirnov, K. V. Stepanov, A. T. Turov, Y. A. Konstantinov, and I. A. Lobach, "Scientific applications of distributed acoustic sensing: State-of-the-art review and perspective," *Sensors*, vol. 22, no. 3, 2022.

[8] S. Jakkampudi, J. Shen, W. Li, A. Dev, T. Zhu, and E. R. Martin, "Footstep detection in urban seismic data with a convolutional neural network," *The Leading Edge*, vol. 39, no. 9, pp. 654–660, 2020.

[9] *Machine Learning Algorithms for Automated Seismic Ambient Noise Processing Applied to DAS Acquisition*, vol. All Days of *SEG International Exposition and Annual Meeting*, 10 2018. SEG-2018-W20-03.

[10] S. A. Klaasen, P. Paitz, N. Lindner, J. Dettmer, and A. Fichtner, "Distributed Acoustic Sensing in Volcano-Glacial Environments - Mount Meager, British Columbia," preprint, Geophysics, Apr. 2021.

[11] R. Allen, "Automatic phase pickers: Their present use and future prospects.," *Bulletin of the Seismological Society of America 72 (6B): S225-42*, 1928.

[12] M. Bublin, "Event detection for distributed acoustic sensing: Combining knowledge-based, classical machine learning, and deep learning approaches," *Sensors*, vol. 21, p. 7527, 11 2021.

[13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

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

[15] A. H. Hartog, "An Introduction to Distributed Optical Fibre Sensors," p. 471.

[16] J. Muggleton, R. Hunt, E. Rustighi, G. Lees, and A. Pearce, "Gas pipeline leak noise measurements using optical fibre distributed acoustic sensing," *Journal of Natural Gas Science and Engineering*, vol. 78, p. 103293, 2020.

[17] Z. Li and Z. Zhan, "Pushing the limit of earthquake detection with distributed acoustic sensing and template matching: a case study at the Brady geothermal field," *Geophysical Journal International*, vol. 215, pp. 1583–1593, 09 2018.

[18] K. Harris, D. White, D. Melanson, C. Samson, and T. M. Daley, "Feasibility of time-lapse vsp monitoring at the aquistore co2 storage site using a distributed acoustic sensing system," *International Journal of Greenhouse Gas Control*, vol. 50, pp. 248–260, 2016.

[19] T. Daley, D. Miller, K. Dodds, P. Cook, and B. Freifeld, "Field testing of modular borehole monitoring with simultaneous distributed acoustic sensing and geophone vertical seismic profiles at citronelle, alabama," *Geophysical Prospecting*, vol. 64, no. 5, pp. 1318–1334, 2016.

[20] F. Walter, D. Gräff, F. Lindner, P. Paitz, M. Köpfli, M. Chmiel, and A. Fichtner, "Distributed acoustic sensing of microseismic sources and wave propagation in glaciated terrain," *Nature Communications*, vol. 11, p. 2436, Dec. 2020.

[21] Y. Tan, S. Wang, M. Rijken, K. Hughes, I. Lim Chen Ning, Z. Zhang, and Z. Fang, "Geomechanical template for distributed acoustic sensing strain patterns during hydraulic fracturing," *SPE Journal*, vol. 26, pp. 1–12, 04 2021.

[22] T. Dean, T. Cuny, and A. H. Hartog, "The effect of gauge length on axially incident p-waves measured using fibre optic distributed vibration sensing," *Geophysical Prospecting*, vol. 65, no. 1, pp. 184–193, 2017.

[23] M. W. Becker, C. Ciervo, M. Cole, T. Coleman, and M. Mondanos, "Fracture hydromechanical response measured by fiber optic distributed acoustic sensing at milliHertz frequencies: Fracture Hydromechanics From DAS," *Geophysical Research Letters*, vol. 44, pp. 7295–7302, July 2017.

[24] M. W. Becker and T. I. Coleman, "Distributed acoustic sensing of strain at earth tide frequencies," *Sensors*, vol. 19, no. 9, 2019.

[25] J. P. Dakin and C. Lamb, "Distributed fibre optic sensor system," Feb. 1990.

[26] I. Idris, A. Selamat, N. T. Nguyen, S. Omatu, O. Krejcar, K. Kuca, and M. Penhaker, "A combined negative selection algorithm–particle swarm optimization for an email spam detection system," *Engineering Applications of Artificial Intelligence*, vol. 39, 03 2015.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

[28] B. Park and J. K. Bae, "Using machine learning algorithms for housing price prediction: The case of fairfax county, virginia housing data," *Expert Systems with Applications*, vol. 42, no. 6, pp. 2928–2934, 2015.

[29] I. Sanchez, "Short-term prediction of wind energy production," *International Journal of Forecasting*, vol. 22, pp. 43–56, 03 2006.

[30] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, vol. 5.1, pp. 281–298, Jan. 1967. Publisher: University of California Press.

[31] M. Ester, H. P. Kriegel, J. Sander, and X. Xiaowei, "A density-based algorithm for discovering clusters in large spatial databases with noise," 12 1996.

[32] J. Batson and L. Royer, "Noise2Self: Blind Denoising by Self-Supervision," p. 10.

[33] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," 2020.

[34] E. Wilcox, P. Qian, R. Futrell, R. Kohita, R. Levy, and M. Ballesteros, "Structural supervision improves few-shot learning and syntactic generalization in neural language models," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Online), pp. 4640–4652, Association for Computational Linguistics, Nov. 2020.

[35] C. Matsoukas, J. F. Haslum, M. Söderberg, and K. Smith, "Is it time to replace cnns with transformers for medical images?," *CoRR*, vol. abs/2108.09038, 2021.

[36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

[37] A. F. Agarap, "Deep learning using rectified linear units (relu)," *CoRR*, vol. abs/1803.08375, 2018.

[38] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for Activation Functions," Oct. 2017. Number: arXiv:1710.05941 arXiv:1710.05941 [cs] version: 1.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," 2015.

[40] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," 2017.

[41] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *CoRR*, vol. abs/1602.07261, 2016.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[43] S. Ruder, "An overview of gradient descent optimization algorithms," 2016.

[44] I. Finetti, "Structure, stratigraphy and evolution of central mediterranean," *Bollettino di Geofisica Teorica ed Applicata*, vol. 24, no. 96, pp. 247–312, 1982.

[45] G. Aggouras, E. Anassontzis, A. Ball, G. Bourlis, W. Chinowsky, E. Fahrun, G. Grammatikakis, C. Green, P. Grieder, P. Katrivanos, P. Koske, A. Leisos, J. Ludvig, E. Markopoulos, P. Minkowsky, D. Nygren, K. Papageorgiou, G. Przybylski, L. Resvanis, I. Siotis, J. Sopher, T. Staveris, V. Tsagli, A. Tsirigotis, and V. Zhukov, "Operation and performance of the NESTOR test detector," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 552, pp. 420–439, Nov. 2005.

[46] I. Lior, A. Sladen, D. Rivet, J. Ampuero, Y. Hello, C. Becerril, H. F. Martins, P. Lamare, C. Jestin, S. Tsagkli, and C. Markou, "On the Detection Capabilities of Underwater Distributed Acoustic Sensing," *Journal of Geophysical Research: Solid Earth*, vol. 126, Mar. 2021.

[47] UC San Diego, "Piñon flats observatory array," 2014.

[48] L. Biewald, "Experiment tracking with weights and biases," 2020. Software available from wandb.com.

[49] S. Thrastarson, R. Torfason, S. Klaasen, P. Paitz, Y. Cubuk Sabuncu, K. Jónsdóttir, and A. Fichtner, "Detecting Seismic Events with Computer Vision: Applications for Fiber-Optic Sensing," preprint, Geophysics, Dec. 2021.

[50] H. F. Wang, X. Zeng, D. E. Miller, D. Fratta, K. L. Feigl, C. H. Thurber, and R. J. Mellors, "Ground motion response to an ML 4.3 earthquake using co-located distributed acoustic sensing and seismometer arrays," *Geophysical Journal International*, vol. 213, pp. 2020–2036, June 2018.

[51] A. Sladen, D. Rivet, J. P. Ampuero, L. De Barros, Y. Hello, G. Calbris, and P. Lamare, "Distributed sensing of earthquakes and ocean-solid Earth interactions on seafloor telecom cables," *Nature Communications*, vol. 10, p. 5777, Dec. 2019.

[52] A. Buades, B. Coll, and J.-M. Morel, "Non-Local Means Denoising," *Image Processing On Line*, vol. 1, pp. 208–212, Sept. 2011.

[53] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014.

[54] M. Lebrun, "An Analysis and Implementation of the BM3D Image Denoising Method," *Image Processing On Line*, vol. 2, pp. 175–213, Aug. 2012.

[55] L. Shiloh, A. Eyal, and R. Giryes, "Efficient Processing of Distributed Acoustic Sensing Data Using a Deep Learning Approach," *Journal of Lightwave Technology*, vol. 37, pp. 4755–4762, Sept. 2019.

[56] N. S. Senobari, G. J. Funning, E. Keogh, Y. Zhu, C. M. Yeh, Z. Zimmerman, and A. Mueen, "Super-Efficient Cross-Correlation (SEC-C): A Fast Matched Filtering Code Suitable for Desktop Computers," *Seismological Research Letters*, vol. 90, pp. 322–334, 11 2018.

[57] P. Goldstein and R. J. Archuleta, "Array analysis of seismic signals," 1987.

[58] Z. Li and Z. Zhan, "Pushing the limit of earthquake detection with distributed acoustic sensing and template matching: a case study at the Brady geothermal field," *Geophysical Journal International*, vol. 215, pp. 1583–1593, Sept. 2018. _eprint: https://academic.oup.com/gji/article-pdf/215/3/1583/25817111/ggy359.pdf.

[59] L. Meng, A. Inbal, and J. P. Ampuero, "A window into the complexity of the dynamic rupture of the 2011 mw 9 tohoku-oki earthquake," *Geophysical Research Letters*, vol. 38, p. L00G07, 08 2011.

[60] A. A. Hutchison and A. Ghosh, "Very low frequency earthquakes spatiotemporally asynchronous with strong tremor during the 2014 episodic tremor and slip event in cascadia," *Geophysical Research Letters*, vol. 43, no. 13, pp. 6876–6882, 2016.

[61] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked Autoencoders Are Scalable Vision Learners," Tech. Rep. arXiv:2111.06377, arXiv, Dec. 2021. arXiv:2111.06377 [cs] type: article.

List of Abbreviations

DAS	Distributed Acoustic Sensing
BPF	Band Pass Filter
MSE	Mean Squared Error
HCMR	Hellenic Center for Marine Research
NESTOR	Neutrino Extended Submarine Telescope with Oceanographic Research
wandb	Weights and Biases [48]
STA/LTA	Short Term Average/Long Term Average
CNN	Convolutional Neural Networks
ViT	Vision Transformers
IU	Interrogator Unit
(OTDR)	optical time-domain reflectometry
(OFDR)	optical frequency-domain reflectometry
ML	Machine Learning
DL	Deep Learning
GPU	Graphical Processing Unit