

**Radar Signal Decomposition and Quality Assessment for Wide Area
Monitoring**

by

Xinyao Sun

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

Doctor of Philosophy

Department of Computing Science
University of Alberta

© Xinyao Sun, 2022

Abstract

The objective of signal decomposition is to extract and separate distinct signal components from a composite signal. Signal decomposition has been studied in many applications, such as image, video, audio, and speech signals. This thesis focuses on the category of signal decomposition on Interferometric Synthetic Aperture Radar (InSAR), a remote sensing technology that can monitor the earth from space. It provides measurements for thousands of square kilometres of ground, with a spatial resolution of around 10 m per pixel and a 1 mm precision on ground deformation estimation over time.

For wide-area monitoring, algorithms must handle tens of thousands of radar satellite images annually to measure ground stability over time. This thesis' primary focus is to combine traditional signal/image processing techniques with recent deep learning approaches to improve the InSAR processing pipeline to deliver faster and better results. The task is very challenging because ground surface displacement or deformation signals are encoded in observed InSAR phase measurements with other contaminant signals (e.g., atmospheric distortion, orbital error, and digital elevation model [DEM] error and noise). Each type of signal could be spatially correlated, temporally correlated, or both. It is also possible for the signals to be neither spatially nor temporally correlated. The phase values are wrapped by 2π , which causes a non-continuous processing domain. Moreover, there is no real-world ground truth to reference in the training or validation stages. This thesis explores and addresses the deformation signal extraction problem using different strategies.

We start by focusing on the *image filtering* problem of removing spatially inde-

pendent noise components. We demonstrate a novel deep learning model for Gaussian denoising in natural images and then adapt it to data from the InSAR modality. We designed a *teacher-student paradigm for supervised training* in the absence of real-world ground truth data. The framework uses a standard stack-based filtering method as the "teacher" (requiring more than 30 observations) and a deep differentiable model to learn the behaviour of the teacher method. After training, the student model can produce results comparable to, or even better than, those produced by its teacher method. Moreover, the student model relies on just a single pair of observations. Additionally, the proposed model is designed to provide a coherence map, which indicates the *signal quality at the pixel level*. Furthermore, we present an extension in the form of a novel *self-supervised framework*. This framework can be used to remove noise signals and estimate pixel-level quality using only noisy observations for training and inference.

In addition to the previous outcome, we investigate how to *separate deformation and DEM error signals* using a 2D optimization problem for each spatial location in a time series. In general, current approaches suffer from a non-continuous solution space. They are limited to small-scale displacement use cases, making them unsuitable for high-velocity scenarios such as mining, construction, and earthquakes. We propose a *two-stage optimization strategy* that effectively locates global optima by combining an iterative global coarse search with a stochastic derivative-free local fine search.

Almost all of the research on InSAR deforming signal estimation is based solely on temporal analysis and requires pre-removal of the atmospheric phase. We further investigate the possibility of *spatial-temporal cross-domain optimization* by developing an adaptive kernel that performs convolutional optimization on the entire 3D InSAR stack, resulting in accurate and robust deformation and DEM error signal extraction. The approach should be capable of processing wrapped phases directly and even working on phases that have not had their atmospheric component removed.

Despite these signal decomposition processes, accurately validating and optimizing the developed algorithms remains a challenge due to the lack of relevant ground truth data in a real-world environment. We developed a *stochastic InSAR simulator* to address this problem. The simulator provides a highly flexible modeling framework for generating various phase fringes and coherence distributions. This simulator is suitable for conducting thorough quantitative evaluations of various filtering and coherence estimation algorithms. The simulator features 2D and 3D modes that support stack and non-stack analysis. The 3D version is expected to simulate time-series deformation signals to evaluate signal separation methods. Additionally, to mimic realistic signals, we also study the *intelligent generative InSAR simulator* with adversarial training to learn the real-world deformation signal’s distribution and its correlations to the DEM error.

The main contributions of this thesis include the following:

1. Proposing novel signal decomposition and quality assessment approaches to argument, automate, and accelerate the InSAR processing pipeline for wide-area monitoring.
2. Introducing a stochastic InSAR data simulator for creating synthetic benchmark datasets to aid researchers in comparing the strengths and weaknesses of various algorithms and guiding future research.
3. Providing our industrial partner with the ability to use the established models and algorithms in their operations, rather than just theorizing about them.

Preface

The majority of this thesis' content has been published in peer-reviewed publications and conferences or is under review. A dual-stage framework for natural picture denoising based on adversarial training and feature matching was published at the *International Conference on Smart Multimedia* and is discussed in Chapter 3. In Chapter 4, two frameworks for restoring the phase and estimating the coherence of a SAR interferometric signal are discussed: 1) DeepInSAR, the first deep learning study, was published in *Remote Sensing*, and its commercial implementation has been deployed in the industry since 2019 and; 2) SRDInSAR, which is currently under review in *IEEE Geoscience and Remote Sensing Letters*, is a self-supervised residual distribution learning model that builds on our previous study, GenInSAR, which was also published in *IEEE Geoscience and Remote Sensing Letters*. Chapter 5 addresses two methods for extracting the deformation and DEM error signals from the InSAR time series: 1) The IGS-CMAES technique is a two-stage black-box optimization technique published in *Remote Sensing*. This technique is utilized exclusively for temporal analysis; 2) ConvArcFit is the first spatial-temporal analysis framework for dense signal estimations and has been partially deployed in the industry. Chapter 6 describes our stochastic InSAR simulator, which was recently published in the *Proceedings of the Springer Conference of the Arabian Journal of Geosciences*. Additionally, we illustrate a possible route for mimicking InSAR signals through recent intelligent generative modeling techniques. I lead a team pursuing this topic further, and our most recent results are being reviewed by *Remote Sensing*.

As a technical lead, I have assisted my supervisors in coordinating the remote

sensing research team since 2020. Beyond my thesis topics, I mentor and collaborate with team members on various wide-area monitoring research tasks, including learning-based non-stack phase unwrapping and water body detection using SAR images. I have chosen to use the plural first-person throughout this thesis to acknowledge the contributions of my supervisors and colleagues. The following section is a comprehensive list of contributions we made during my Ph.D. studies.

Journals

1. **Sun, X.**, Zheng, W., Ghuman, P., & Cheng, I. Self-supervised Residual Distribution Learning for SAR Interferometric Phase Restoration and Coherence Estimation. *IEEE Geoscience and Remote Sensing Letters*. (Submitted)
2. Zhou, Z., **Sun, X.**, Zheng, W., Yang, F., Ghuman, P., & Cheng, I.. Deep Generative Modeling for InSAR Signals. *Remote Sensing*, (Submitted)
3. **Sun, X.**, Zheng, W., Ghuman, P., & Cheng, I. ConvArcFit: Spatial-Temporal Optimization for Ground Deformation and DEM Error Estimation. *IEEE Transactions on Geoscience and Remote Sensing*. (Submitted)
4. **Sun, X.**, Zimmer, A., Mukherjee, S., Ghuman, P., & Cheng, I. (2021). IGS-CMAES: A Two-Stage Optimization for Ground Deformation and DEM Error Estimation in Time Series InSAR Data. *Remote Sensing*, 13(13), 2615.
5. **Sun, X.**, Zimmer, A., Mukherjee, S., Kottayil, N. K., Ghuman, P., & Cheng, I. (2020). DeepInSAR—A deep learning framework for SAR interferometric phase restoration and coherence estimation. *Remote Sensing*, 12(14), 2340.
6. Mukherjee, S., Zimmer, A., **Sun, X.**, Ghuman, P., & Cheng, I. (2020). An unsupervised generative neural approach for InSAR phase filtering and coherence estimation. *IEEE Geoscience and Remote Sensing Letters*.

Conferences

1. **Sun, X.**, Zimmer, A., Mukherjee, S., Kottayil, N. K., Ghuman, P., & Cheng, I. (2020). A Benchmark InSAR Simulator for Phase Filtering and Coherence Estimation. Conference of the Arabian Journal of Geosciences (CAJG)
2. **Sun, X.**, Kottayil, N. K., Mukherjee, S., & Cheng, I. (2018). Adversarial training for dual-stage image denoising enhanced with feature matching. In Lecture Notes in Computer Science (pp. 157-168). Springer International Publishing
3. Mukherjee, S., Zimmer, A., **Sun, X.**, Ghuman, P., & Cheng, I. (2019, August). CNN-Based Real-Time Parameter Tuning for Optimizing Denoising Filter Performance. In International Conference on Image Analysis and Recognition (pp. 112-125). Springer, Cham.
4. Kottayil, N. K., Zimmer, A., Mukherjee, **S.**, **Sun, X.**, Ghuman, P., & Cheng, I. (2018, October). Accurate Pixel-Based Noise Estimation for InSAR Interferograms. In 2018 IEEE SENSORS (pp. 1-4). IEEE.
5. Mukherjee, S., Zimmer, A., **Sun, X.**, Ghuman, P., & Cheng, I. (2018, October). CNN-based InSAR Coherence Classification. In 2018 IEEE SENSORS (pp. 1-4). IEEE.
6. Mukherjee, S., Zimmer, A., **Sun, X.**, Ghuman, P., & Cheng, I. (2018, October). CNN-based InSAR denoising and coherence metric. In 2018 IEEE SENSORS (pp. 1-4). IEEE.

Awards

1. Alberta Innovates Graduate Student Scholarship Data-Enabled Innovation, Nov. 2020
2. Alberta Graduate Excellence Scholarship, Dec. 2019
3. Ph.D. Early Achievement Award Runner-up, Nov. 2019

4. University of Alberta Doctoral Recruitment Scholarship, Sep. 2017
5. Selected by the University of Alberta to enter the national Colton Medal Competition, 2019

Invited Guest Lectures

1. "Introduction of Deep Learning and Remote Sensing Applications", Institute of Disaster Prevention Science and Technology, Beijing, China Feb. 2019
2. "Using Deep Learning for Radar Signal Filtering and Recognition", Institute of Disaster Prevention Science and Technology, Beijing, China Dec. 2019

Media and Text Interviews

1. "How Edmonton researchers are using artificial intelligence to try and prevent disaster before it strikes" *Star Metro Edmonton*
2. "Did the earth move for you?... measure it from space", *Edmonton Journal*.
3. "U of A researchers are using AI technology to predict disasters", *Gateway*.
4. "Bringing innovation to industry: Using AI for predicting disasters", *Faculty of Science website, University of Alberta*.
5. "Monitoring movement, mitigating risk", *Ottawa Aerospace Summit* 2018.

Funding Acknowledgment

1. Consortium for Aerospace Research and Innovation in Canada (CARIC)
2. Mitacs
3. NSERC - Collaborative Research and Development (CRD)
4. NSERC - Department of National Defence (DND)

“It is better to fail in originality than to succeed in imitation.”

-Herman Melville

Acknowledgements

I would first like to express my gratitude to my supervisors, Professors Irene Cheng and Anup Basu, for their encouragement, drive, and enthusiasm throughout my Ph.D. journey.

Second, I would like to thank Parwant Ghuman for providing me with internship opportunities at his company, 3vGeomatics. During this experience, I had the chance to work on various interesting and challenging projects. Additionally, I would like to express my gratitude to the 3VGeomatics research and development (R&D) team for their guidance and help, particularly Aaron Zimmer and Zheng Wang, who closely collaborated with and supported our research team.

I am grateful for the collaborative work of my colleagues at the University of Alberta: Subhayan Mukherjee; Navaneeth Kamballur Kottayil; Zhongrun Zhou; MD Samiul Islam; and Sachin Vijay Kumar. Additionally, I am grateful to my friends at the University of Alberta - Lihang, Jiyang, Chenqiu, Housam, Sarah, and Nasim, for the enjoyable discussions about life and research.

Additionally, I would like to thank the remaining members of my thesis committee - Drs. Nilanjan Ray, Pierre Boulanger, Witold Pedrycz and Gene Cheung, for their support, insightful comments, thoughtful questions, and excellent feedback on my research and thesis.

I want to express my heartiest gratitude to my family. Thanks to my parents and my wife, Xiaoqi (Fiona), for delivering Queenie Wancheng, our little angel. Your support during the pandemic was important to me.

Finally, I would want to convey my gratitude to everyone who directly or indirectly

supported my Ph.D. journey. Without you, I would be unable to make it.

Table of Contents

1	Introduction	1
1.1	Motivation	1
1.2	Challenges	2
1.3	Contribution Summary	3
1.3.1	Removing the noise term and estimating the signal quality	3
1.3.2	Extracting deformation and DEM error terms	3
1.3.3	InSAR data simulator	4
1.4	Thesis Structure	4
2	Background and Related Works	5
2.1	Natural Image Denoising	5
2.2	InSAR Filtering and Coherence Estimation	6
2.3	Time Series InSAR Deformation Estimation	9
2.4	InSAR Data Simulator	11
3	Adversarial training for DUAL-STAGE natural image denoising enhanced with feature matching	12
3.1	Introduction	12
3.2	Methods and Procedure	13
3.2.1	1 st Stage: Residual Learning Denoiser \mathbf{D}_{s1}	14
3.2.2	2 nd Stage: Full Image Reconstruction Denoiser \mathbf{D}_{s2}	14
3.2.3	Adversarial Training with Feature Matching	15

3.2.4	Learning Configuration	16
3.3	Results and Discussion	16
3.3.1	Training and Testing	17
3.3.2	Evaluation	18
3.4	Conclusions	18
4	SAR Interferometric Phase Restoration and Coherence Estimation	21
4.1	Introduction	21
4.1.1	Phase Noise Model	22
4.2	DeepInSAR—A Deep Learning Framework for SAR Interferometric Phase Restoration and Coherence Estimation	23
4.2.1	Method and Procedure	23
4.2.2	Experimental Setup	32
4.2.3	Results	33
4.2.4	Conclusions	48
4.3	Self-supervised Residual Distribution Learning for SAR Interferometric Phase Restoration and Coherence Estimation - SRDInSAR	49
4.3.1	Introduction	49
4.3.2	Method and Procedure	51
4.3.3	Experimental Setup	55
4.3.4	Results	55
4.3.5	Conclusions	61
5	Deformation and DEM Error Signals Separation in InSAR Time Series	63
5.1	Introduction	63
5.2	IGS-CMAES: A Two-Stage Optimization for Ground Deformation and DEM Error Estimation in Time Series InSAR Data	65
5.2.1	Introduction	65

5.2.2	Mathematical Modeling for InSAR Phase	66
5.2.3	Proposed Method	68
5.2.4	Results	81
5.2.5	Discussion	92
5.2.6	Conclusions	94
5.3	ConvArcFit: Spatial-Temporal Optimization for Ground Deformation and DEM Error	95
5.3.1	Introduction	95
5.3.2	Definition of Optimization Problem	95
5.3.3	Results and Future Work	99
6	InSAR Simulator	100
6.1	Introduction	100
6.2	Methods and Procedure	101
6.2.1	2D InSAR Simulator	101
6.2.2	3D InSAR Simulator	103
6.2.3	Intelligent InSAR Simulator - Learning from real world signals	105
6.2.4	Conclusion	107
7	Conclusion and Future Directions	108
7.1	Summary	108
7.2	Knowledge Translation and Commercialization	110
7.3	Limitations and Future Work	110
	Bibliography	112

List of Tables

3.1	Average PSNR and SSIM results of σ 30, 50, 70 for 14 widely used images.	17
3.2	Average PSNR and SSIM results of σ 30, 50, 70 for BSD200 images.	18
4.1	Phase Root Mean Square Error (RMSE) (radians) on 18 different types of Simulation dataset. S denotes Gaussian level of base noise and F represents phase fringes frequency. S and NS mean with and without low amplitude strips respectively. Values with bold fonts indicate the best performance.	39
4.2	Phase mean Structural Similarity Map (SSIM) on 18 different types of Simulation dataset. S denotes Gaussian level of base noise and F represents phase fringes frequency. S and NS mean with and without low amplitude strips respectively. Values with bold fonts indicate the best performance.	40
4.3	Coherence RMSE on 18 different types of Simulation dataset. S denotes Gaussian level of base noise and F represents phase fringes frequency. S and NS mean with and without low amplitude strips respectively. Values with bold fonts indicate the best performance. Values with bold fonts indicate the best performance.	41

4.4	Coherence mean SSIM on 18 different types of Simulation dataset. S denotes Gaussian level of base noise and F represents phase fringes frequency. S and NS mean with and without low amplitude strips respectively. Values with bold fonts indicate the best performance. . .	42
4.5	Running time T (in seconds) of different methods with image size 1000×1000	43
4.6	Quantitative Evaluation of SRDInSAR and existing methods	58
5.1	The key parameters of covariance matrix adaptation evolution strategy (CMAES) used in this work.	74
5.2	Quantitative assessment for simulated data with several widely-used local optimizers and their IGS-extended version	82
5.3	Quantitative comparison for simulated data using IGS-CMAES with two other Global Optimizers	91
5.4	Quantitative assessment of IGS-CMAES's outputs with reference results	92

List of Figures

3.1	Architecture of proposed AADNet with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.	13
3.2	The 14 widely used test images	17
3.3	Visual comparison examples - (Left to right): noisy image ($\sigma = 70$), denoised by AADNet and denoised by DnCNN-S. The zoom-in view shown at the bottom right of each image demonstrates that AADNet preserves fine detail better.	19
4.1	The architecture of the proposed Deep Interferometric Synthetic Aperture Radar (DeepInSAR) network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each Convolutional Neural Network (CNN) layer	24
4.2	Before and after preprocessing: amplitude images selected from three real-world site datasets. From left to right, it shows Site-A (1st and 2nd columns), Site-B (3rd and 4th columns), Site-C (5th and 6th columns) with two samples for each dataset. (1st row) Raw amplitude images after log transformation for better visualization, (2nd row) their corresponding histograms in log, (3rd row) histograms after proposed normalization and (4th row) corresponding normalized images.	28
4.3	Information and Gradient flow between modules.	31

4.4	Illustration of the PtSel method describing the three key steps in order [96].	32
4.5	We use S#-F#-S or S#-F#-NS to name simulation datasets generated using different distortion scenarios: S# denotes Gaussian level of base noise S; F# denotes frequency level of phase fringes F; S and NS mean with or without low amplitude strips respectively. (From left to right) A set of simulated images are selected from S1-F3-NS, S2-F2-NS, and S3-F1-S datasets. First row shows simulated ground truth with clean interferometric phase $[-\pi, \pi)$, second row is the noisy interferometric phase $[-\pi, \pi)$ —(Blue: $-\pi$; Red: $+\pi$), and third row is coherence (Black: 0; White: 1).	35
4.6	Examples of filtering and coherence estimation results on sample simulation images shown in Figure 4.5. (a–d) are filtering outputs and (e–h) are coherence estimations of S1-F3-NS, (i–l) are filtering outputs and (m–p) are coherence estimations of S2-F2-NS, and (q–t) are filtering outputs and (u–x) are coherence estimations of S3-F1-NS. Visual inspection on filtered outputs from different methods compared to ground truth phase images are given in Figure 4.5, 1st row. It can be seen that our model can preserve structural details better than others for increasing base noise levels and frequency of fringes (5th row). Our proposed method’s coherence estimation is most matched to ground truth (Figure 4.5, 3rd row), while other methods tend to predict inaccurate results on areas with highly dense fringes or low amplitude stripes.	37
4.7	Three representative noisy interferograms (Phase) selected from each of the three real datasets; Blue: $-\pi$; Red: $+\pi$	38
4.8	Filtered images and coherence maps generated by the reference methods and proposed DeepInSAR trained model for a Site-A image. . . .	44

4.9	Filtered images and coherence maps generated by the reference methods and proposed DeepInSAR trained model for a Site-B image. . . .	45
4.10	Filtered images and coherence maps generated by the reference methods and proposed DeepInSAR trained model for a Site-C image. . . .	46
4.11	Illustration of proposed SRDInSAR framework.	52
4.12	Filtered phase simulated InSAR images from SRDInSAR and five existing methods. Phase is colored between $-\pi$ (blue) to $+\pi$ (red)	56
4.13	Coherence estimation of simulated InSAR images from SRDInSAR and five existing methods. Coherence is colored between 0 (black) to 1 (white)	57
4.14	Filtered phase and coherence outputs for satellite InSAR images from SRDInSAR and six existing methods. Phase and coherence are colored from $-\pi$ (blue) to $+\pi$ (red), and 0 (black: low) to 1 (white: high), respectively	61
5.1	Illustration of phase jump and compassion real and imaginary MSE (RI-MSE) and mean square error (MSE) objective functions.	70
5.2	Illustration of proposed two-stage iterative grid search with covariance matrix adaptation evolution strategy (IGS-CMAES) method.	71
5.3	Loss landscape of objective function RI-MSE (Equation (5.7))) with two different real-world spatial and temporal baselines. Site (A) and Site (B) are two different real-world sites.	76
5.4	The landscape of loss function. Each row represents the loss landscapes that corresponds to real-world baselines with synthetic 0 deformation and 0 DEM error, and each column shows the results under different step sizes of grid sampling. The linear deformation rate x -axis and DEM error y -axis form a 2D solution space for an InSAR pixel location. The value is calculated by Equation (5.7) with selected real-world baselines.	77

5.5	A simple illustration of selecting candidate solutions by skipping the estimates (Estimate 2, 3) that are too close to any picked candidate solution (Estimate C).	79
5.6	Unwrapped phase L1 difference.	84
5.7	Reducing rates (ACC%) of unwrapped phase L1 difference less than π	85
5.8	<i>Cont.</i>	87
5.8	<i>Cont.</i>	88
5.8	<i>Cont.</i>	89
5.8	Visualization of estimated linear deformation rate and DEM on R1–R7 real-world stack’s PS pixels. IGS-CMAES is applied on wrapped interferogram directly. Reference results are generated using 3vGeomatics processing pipeline with unwrapped phase.	90
5.9	Proposed ConvArcFit optimization workflow.	96
5.10	Results on a real world stack; the model shows consistent results with input data with and without atmospheric removal.	99
6.1	Column A: low base noise, sparse fringes; B: median base noise, dense fringes; C: ellipse signals; D: polygon signals; E: phase bands; F: low amplitude bands; G: plenty of amplitude bands with building blocks.	102
6.2	Sample outputs from 3D Simulator by creating Gaussian Pyramids signals pairs.	104
6.3	The proposed baseline GAN architecture. The generator’s input is the noise map. Its outputs are a two-channel signal map (concatenation of synthetic linear deformation rate and DEM error signals) with the same spatial resolution. The discriminator is learned to tell if given two-channel signals from real distribution.	105
6.4	Sample outputs of baseline GAN.	106

6.5 Sample pair of real world motion rate and DEM height error signal
maps processed by full processing pipeline. 107

Abbreviations

AWGN Additive white Gaussian noise.

BM3D Block-matching and 3D filtering.

CMAES Covariance matrix adaptation evolution strategy.

DEM Digital Elevation Models.

DNN Deep Neural Network.

GAN Generative Adversarial Net.

InSAR Interferometric Synthetic Aperture Radar.

LAMBDA Least-squares Ambiguity Decorrelation Adjustment.

MAD Median Absolute Deviation.

ML Maximum Likelihood.

MRF Markov Random Field.

MSE Mean Squared Error.

PSInSAR Permanent Scatterer InSAR.

PSNR Peak Signal-to-Noise Ratio.

RMSE Root Mean Square Error.

SBAS Small Baseline Subset.

SLC Single Look Complex.

SSIM Structural Similarity Index.

STUN Spatio-Temporal Unwrapping Network.

TsInSAR Time Series InSAR.

Chapter 1

Introduction

Signal decomposition deals with extracting signals of interest from a composite observation. It is a fundamental procedure that applies to various applications and research areas. Depending on the application and acquisition technique, the objectives may include eliminating noise signals or extracting the target signals from images, videos, and audio files. The purpose of this thesis is to investigate the signal modality of interferometric synthetic aperture radar (InSAR). This remote sensing technique has attracted considerable interest in monitoring the earth from space. This technique can reconstruct the surface topography of the earth and measure ground displacement across wide areas. In recent decades InSAR has emerged as a promising technology for monitoring earth subsidence and uplift caused by human activities (e.g., urban infrastructure development, mining, oil and gas extraction) and natural phenomena (e.g., earthquakes, volcanic eruptions, and permafrost thawing). Numerous existing sophisticated SAR satellites, including Canadian Radar Sat-1/2, European ERS-1/2, ENVISAT, Sentinel-1A/1B, and Japanese ALOS, cover the entire globe. Depending on the satellite, InSAR can measure thousands of square kilometres of the ground area with a spatial resolution of around 10 meters per pixel and a precision of up to a millimetre for ground deformation assessment. There has been growing demand for robust, precise, and efficient algorithms to process such vast amounts of data and produce reliable earth monitoring analysis.

1.1 Motivation

SAR satellites transmit radar signals toward the globe and record their echoes to produce a 2D radar image. Each pixel represents a grid area on the earth's surface. The 2D signal is represented as a single-look complex (SLC) image, each pixel of which represents the phase and amplitude of the returning radar echo from each

resolution cell on the ground. An interferogram, also known as an InSAR image, is formed by combining two distinct SLC images via the point-wise product of one SLC with the complex conjugate of the other. Both SLCs were acquired over a prolonged period in the same region of interest. Their interferogram is also a complex number, with a phase value representing the phase difference between two temporally separated SLC pictures. However, phase values in InSAR images encode surface displacement signals and various other contaminant signals, including atmospheric distortion, orbital error, elevation model error, and noise. Each of these may be spatially correlated, temporally correlated, or both. It is also possible for the signals to be neither temporally nor spatially correlated. Therefore, signal decomposition plays a crucial role in the existing InSAR processing pipeline.

1.2 Challenges

A single InSAR image can have up to 100 million level pixels for wide-area monitoring and even more when the temporal analysis is performed on a stack of images. This raises an even greater challenge since the method must be highly parallelizable to process tens of thousands of radar images each year to determine ground displacement over time. A traditional InSAR processing pipeline typically requires additional observations (SLCs tokens at different times) or professional operators to manually review and tweak algorithm parameters as necessary to obtain a satisfactory final output. This is mostly because existing algorithms rely heavily on run-time optimization and are incapable of generalizing various use cases using the same setup.

Each observed phase pixel in an interferometric SAR image encodes a different type of signal [1] and can be represented by the following equation:

$$\phi = \phi_{def} + \Delta\phi_{topo} + \Delta\phi_{atmo} + \Delta\phi_{orbit} + \phi_{noise} \quad (1.1)$$

where ϕ_{def} represents phase components related to ground deformation motion, ϕ_{topo} represents topographic (DEM) error, ϕ_{atmo} refers to the difference in atmospheric distortion between two SLCs, ϕ_{orbit} refers to in-precise satellite orbits data when transforming the contributions of earth's ellipsoidal surface, and ϕ_{noise} denotes de-correlation noise.

It is well-known that the most valuable signal is ground deformation. In this thesis, we propose to approach ground deformation signal separation as a series of sub-tasks involving modeling and filtering various types of sub-signals. Moreover, we investigate traditional signal processing techniques with modern machine learning approaches to obtain a robust, effective ground deformation measurement. Our focus

is to enhance the performance of an existing InSAR processing pipeline by 1) reducing the number of observations, 2) accelerating the processing time, and 3) automating the analytic operations. Finally, obtaining ground truth information for real-world signals is always impossible. Synthetic data is frequently employed to conduct objective evaluations of various processing methods. Consequently, we study the potential of creating a comprehensive InSAR data simulator capable of producing realistic signals to avoid evaluation bias.

1.3 Contribution Summary

The signal decomposition work on InSAR signal data is divided into three sub-tasks in this thesis: 1) filtering the noise signal, 2) extracting the deformation signals and DEM height error signals, and 3) proper validation using the proposed InSAR simulator. Each work in this thesis is unique and demonstrates for the first time how modern computer vision and machine learning methodologies benefit the field of InSAR research. My contribution to this dissertation is mostly concentrated on the signal decomposition of observed ϕ signals utilizing different sub-modules:

1.3.1 Removing the noise term and estimating the signal quality

1. We demonstrate how to remove Gaussian noise from natural images using a two-stage adversarial learning architecture.
2. We develop a method for efficiently filtering and estimating the coherence of InSAR images in a supervised training setup.
3. We create a self-supervised learning model capable of removing phase noise and estimating coherence using only noisy observations.
4. We evaluate all suggested methods using qualitative analysis (expert feedback on real-world images) and quantitative analysis (numerical errors on simulated images).

1.3.2 Extracting deformation and DEM error terms

1. We employ a temporal-only analysis to decompose the deformation and DEM error signals for each pixel location according to the conventional processing literature.

2. We provide an optimization approach that combines spatial and temporal modeling to effectively extract dense signals of interest from a 3D signal stack.

1.3.3 InSAR data simulator

1. We develop a stochastic spatial-only 2D simulator capable of randomly generating synthetic interferometric phase and noise to analyze filtering and coherence estimation performance.
2. We extend the 2D simulator by injecting spatial and temporal baseline parameters to generate synthetic stack data to analyze time-series deformation and DEM error estimation capabilities.
3. We propose an intelligent generative model to discover the real-world signal distribution. Following training, the model can generate synthetic signals similar to the distribution of real-world data.

1.4 Thesis Structure

The remainder of the thesis is organized as follows: Chapter 2 discusses background research and related works for each of our focused sub-tasks. Chapter 3 presents our investigation of natural image restoration. Chapter 4 covers our two learning-based methods for interferometric phase restoration and coherence estimation, using supervised and unsupervised learning. Chapter 5 demonstrates how to extract deformation and DEM error using two-stage temporal and single-stage spatial-temporal optimization. Chapter 6 discusses our proposed 2D and 3D InSAR simulators, which we used to evaluate our experiments and illustrate the viability of simulating InSAR signals using an intelligent generation model. Finally, Chapter 7 summarizes our findings and indicates directions for future research.

Chapter 2

Background and Related Works

2.1 Natural Image Denoising

Images and videos can be contaminated with noises at various stages along the processing pipeline (e.g., acquisition, compression, and transmission). Noise can affect the quality of subsequent processing and user visualization. Thus, image denoising [2] is a necessary step for many applications. Although noise patterns vary, an additive white Gaussian noise (AWGN) is commonly discussed in the literature. Furthermore related denoising approaches have been exploited extensively by modeling image priors and optimization problems solving (e.g., non-local self-similarity (NSS) models [3] [4], sparse representations models [5] [6] and gradient-based models [7] [8]). These conventional methods target mainly low to medium noise levels, involve time-consuming optimization processes, and require handcrafted image priors. In general, conventional approaches are insufficient for denoising complex and diverse scene content with fine detail. Recently, deep neural networks (DNNs), especially deep convolutional neural networks (CNNs), have shown promising performance in image denoising. These deep CNNs use a discriminative denoising model (e.g., MLP [9], RED-Net [2] and DnCNN [10]). Their success is mainly attributed to CNNs' modeling capability and deep network training. These deep-learning-based discriminative methods perform better than conventional model-based methods. However, these discriminative learning approaches are insufficient when an image is distorted with high noise. Image denoising is a popular research topic in the literature [4] [5] [7] [11]. In general, these techniques follow image prior modeling to solve optimization problems. They remove noises but tend to over-filter the content, especially fine details. As deep learning methodology emerges, neural network-based discriminative denoising methods show better performance than conventional model-based techniques in many applications. One advantage of using neural networks, particularly CNN-based

models, is that the network parameters for image denoising can be learned from the training data based on pairs of clean and corrupted images rather than pre-defined handcrafted image priors. This powerful learning capability makes DNNs attractive. Asymmetrically encoder-decoder skip-layer connection was introduced [2] to take full advantage of deep learning while avoiding the computational cost of training too deep. This model shows faster training and better denoising performance. Residual network (ResNet) [12] has been designed for a similar purpose but trains an extremely deep CNN. As an alternative, DnCNN [10] adopts the residual learning formulation. It deploys identity shortcuts and a single residual unit to generate a residual image instead of directly outputting the latent clean image. A fixed-size kernel is used at each layer in a classical CNN architecture. All necessary features characterized by different kernel sizes can only be extracted by exploring multiple layers. In order to extract multi-scale features at the same layer, GoogLeNet [13] applies multiple sets of convolutional filters to the image. The resulting activations are then stacked together and passed to subsequent layers.

2.2 InSAR Filtering and Coherence Estimation

Numerous filtering methods have been proposed for interferometric phase filtering in recent decades. BoxCar is a well-known method because it is straightforward. It simply performs a moving average to estimate the variation of local pixel patterns. Authors have shown that this average process is a maximum-likelihood (ML) estimator for interferometric phase and coherence when all involved processes are stationary [19]. However, InSAR images are inherently non-stationary because of changing topography and ground displacement. While a BoxCar filter can be useful in a flat area, it is not suitable for areas with high slopes. In addition, BoxCar outputs are unsatisfactory due to its strong smoothing behaviour caused by simple averaging. BoxCar is vulnerable to loss of both spatial resolution and fine details. It is also vulnerable to significant phase and coherence estimation error. Other classic filters, such as median filter, 2D Gaussian filter, and multi-look processing, have similar limitations. Consequently, researchers started addressing the problem of non-stationary filtering for the interferometric phase. Generally speaking, the researchers' methods can be categorized into two groups depending on whether the filtering is done with or without domain transformation.

Lee filter [15] is a classic method that works in the original spatial domain. It adopts local fringe morphology modeling with anisotropic filtering, reducing the noise via local statistics and an adaptive window. Researchers have introduced an extension of

Lee’s method by using a minimum mean squared error estimator to exclude singular pixels within a selected direction [16]. Another statistical optimization framework has been proposed applying Bayesian estimation in the filtering process [17]. Some adaptive methods have also been proposed [18, 19]. For example, Vasile et al. designed an intensity-driven adaptive-neighborhood method for denoising interferometric phase images [18]. Yu et al. used a low-pass filter along with local fringe orientation with an adaptive contoured window [19]. Wang et al. indicated that phase fringe and noise frequency distribution are different, and hence noise can be detected without destroying the fringe signal [20]. Some works estimate the maximum posterior probability, as a filtered phase image can be obtained by modeling the image prior as a markov random field (MRF)[17, 21]. However, choosing appropriate properties as image prior remains an unsolved problem.

Goldstein filter [22] is the first frequency-domain method with Fourier transformation. One of its extensions [23] proposed a technique to preserve the signal in low noise (high coherence) areas by estimating the dominant component from the signal’s local power spectrum, which also adapts to the local direction of fringes. Researchers proposed other improvements to the Goldstein and Baran filters. For example, researchers tried to obtain a more accurate coherence estimation and overcome the original method’s under-filtering issue in low coherent regions [24] [25]. A joint method, using a modified Goldstein and simplified Lee filter, was invented [26]. This filter focuses on interferometric phase denoising under high-density fringes and a low coherent situation. Researchers showed that filtering with an adaptive multi-resolution technique is also necessary because of the different sizes and shapes of the interferogram [27]. This technique improves the filtering quality on the fringes via better frequency estimation and invalid estimation correction. In [28], authors first proposed a wavelet domain filter in complex domain (WInPF) based on a complex phase noise model. They proved that phase information and noise could be more easily separated in the wavelet domain. The success of WInPF was significant for the bulk of subsequent work. Other researchers applied a wavelet packet-based Wiener filter to further separate phase information in the wavelet packet domain [29]. This method achieves superior performance compared to the WInPF filter. Bian and Mercer proposed undecimated wavelet transform by treating image filtering as an estimation problem [30]. Overall, wavelet-domain filters preserve an adequate spatial resolution better than other methods and have high computational efficiency. Xu et al. [31] introduced a joint denoising filter via simultaneous regularization in the wavelet domain. Phase discontinuities are well preserved through this joint sparse constraint and its iterations.

The idea of non-local filtering is to explore more information from the data itself. In general, images contain repetitive structures such as corners and lines. Those redundant patterns in an image could be analyzed and explored to improve filtering performance. Recently, many studies deployed non-local techniques for SAR data filtering, from amplitude image despeckling [32–34] to interferometric phase denoising [35–39] and InSAR stack multi-temporal processing [40, 41]. Compared to the methods already mentioned, non-local-based methods always achieve state-of-the-art results. Non-local filtering, like previous approaches, adapts estimation to the local signal behaviour to deal with non-stationary images. Still, it also considers the entire image according to the image self-similarity property. The first non-local method applied to interferometric phase filtering was proposed by Deledalle et al. [32]. Image intensities and interferometric phase information are used to build a non-local means model with a probability criterion for estimating pixels. NL-InSAR [35] is the first InSAR application to use a non-local approach for the joint estimation of the reflectivity, interferometric phase and coherence map from an interferogram. In [36] and [42], researchers achieved better results on textural fine detail preservation by combining non-local filtering with pyramidal representation and singular value decomposition. A unified framework (NL-SAR) has been proposed in [39] as an extension of NL-InSAR, where an adaptive procedure is carried out to handle extremely high-resolution images. It is able to obtain the best non-local estimation with good quality on radar structures and discontinuities reconstruction. Recently, works on extending and modifying existing image restoration algorithms to suit interferometric phase domain achieved promising performance. In [20], a modified patch-based locally optimal Wiener (PLOW) method is proposed for interferometric phase filtering that achieves on par and better results than non-local means. Another famous algorithm, non-local block-matching 3D (BM3D) also inspired researchers to propose InSAR-BM3D [38], which delivered state-of-the-art results for InSAR phase filtering. The method is not proposed to do coherence estimation specifically. Instead, InSAR-BM3D computes the maximum likelihood estimates of coherence via stack-wise averages. Then the estimated coherence is used to determine the threshold at the collaborative filtering step. Hence, the accuracy of the coherence estimation likely affects the performance, which is highly dependent on how stationary the whole stack is.

Milestone works using CNNs have shown their ability to outperform almost all conventional algorithms on different visual-related tasks, including image restoration. Some recent SAR-based studies benefit from CNN, including the Fuzzy superpixels based Semi-supervised Similarity-constrained CNN (FS-SCNN) model [43], which uses an ensemble learning technique to achieve superior prediction on PolSAR image

classification tasks. Ma et al. [44] proposed an attention-based graph CNN to improve the SAR segmentation results. In [45], DeepLabv3+ [46], a well-known image semantic segmentation CNN model, was adopted for oil spill identification on SAR images. A direct automatic target recognition (D-ATR) deep CNN-based model outperformed all other conventional methods, it was proposed to obtain high accuracy, and fast processing for target recognition [47]. These works benefit from CNNs as superior feature extractors on SAR images. Anantrasirichai et al. [48] employed CNNs for the InSAR phase to volcano deformation monitoring via transfer learning from optical images. In this work, we propose our DeepInSAR architecture, a new deep learning-based model for SAR interferometric phase restoration and coherence estimation. The model is empowered by state-of-the-art deep learning techniques, relying on suitable phase-oriented solutions. We aim to design a more effective joint phase filter and coherence estimator by learning from the pre-generated training data. We pre-processed InSAR data into a single tensor to perform a multi-modal fusion analysis of both phase and amplitude information. A densely connected feature extractor is used to achieve multi-scale feature extraction and fusion. Two subsequent fully connected CNN perform phase filtering and coherence estimation from extracted features respectively. InSAR phase noise can be considered zero-mean additive noise. Therefore, we adopted the residual learning strategy, which has been proven effective for removing such types of noise [10]. Meanwhile, pre-activation and bottleneck [49], and batch normalization techniques [50], were used to enhance training efficiency and boost the model’s performance.

2.3 Time Series InSAR Deformation Estimation

Apart from filtering noisy InSAR observations, our primary focus was measuring ground deformation signals accurately. Previous research describes ground deformation as the phase difference between two radar observations. Until the 2000s, most algorithms focused on analyzing single interferograms derived from one pair of SAR images [51, 52]. Later, researchers noted that some radar targets’ back-scattering characteristics could maintain stability for a long time [51]. Then, researchers discovered that analysis of multiple acquisitions in time could overcome the conventional InSAR limitations, namely, temporal and geometric decorrelation and atmospheric disturbances. Using time-series InSAR (TSInSAR) techniques has emerged as a powerful strategy to monitor slow and subtle terrain displacements[53].

In general, the TSInSAR techniques use SAR images acquired on the same ground area but on different dates to construct a stack of N interferograms. The signal

phase of each interferometric resolution cell on the ground is a function of multiple-phase contributors. (e.g, uncompensated topographic component, deformation rate, and atmospheric bias) [54][51][55]. Hence the ground deformation can be estimated from the same resolution cell taken at different times [56]. However, each cell is represented as a wrapped phase, and the estimation with phase cycle ambiguities makes this task challenging. Extensive research has shown that, inevitably, there are many calculations, especially when processing wide-area regions [57]. Moreover, a range of processing methods requires manual inspection and expert interpretation[58] and could limit the timely dissemination of monitoring. Hence, it is crucial to have an accurate, efficient and robust algorithm.

One concern is that although the deformation estimation is applied after filtering, some locations are incoherent and impossible to derive any useful information from due to the temporal decorrelation. One solution is to only study the temporally coherent targets on the ground. Ferretti et al. [59, 60] proposed permanent scatter interferometry (PSI) in the early 2000s. This method estimates the deformation parameters lie on the identified permanent scatters that are coherent over the temporal stack. Werner et al. [61] applied interferometric point target analysis within PSI’s scope using a 2D regression algorithm to model the relationship between the perpendicular baseline and the temporal baseline. This algorithm estimates the terrain height and deformation using linear regression analysis. Another Integer Least Squares based technique - Spatio-Temporal Unwrapping Network (STUN) [62] is proposed to solve the phase ambiguities problem via Least-squares Ambiguity Decorrelation Adjustment (LAMBDA) method followed by a sequential least-squares search. Both approaches require extensive search or complex transform computation to eliminate the wrapped phase ambiguities. Persistent Scatterer Pairs InSAR (PSP)[63] and Quasi-Persistent Scatterer (QPS) [64] avoided the complex parameter modeling by directly searching the parameters in the solution space. Although these algorithms are simple and flexible, they have a trade-off for accuracy and efficiency. Especially for the large and complex deformation scenarios, PSP and QPS’s estimated results may be stuck in the local optimization. Hence, it is crucial to have an accurate, effective and robust deformation estimation algorithm. Another strategy is based on the small baseline subset (SBAS) algorithm that was first introduced by Beradion et al. [65], which only works on data pairs characterized by a small orbital separation (baseline) to suppress spatial decorrelation phenomena. It should be noted that SBAS based methods usually obtain deformation parameters using least squares (LS) optimization from the unwrapped phase [54, 66], where PSI-based methods [57, 67, 68] can deal with both the wrapped and unwrapped phases. Later, PS-SBAS-combined

methods [53, 69, 70] were proposed and relied on the unwrapped phase, assuming the small enough baselines and arcs to avoid the phase ambiguities of the phase gradient between two permanent scatters. The main limitation of SBAS based methods is their dependency on phase unwrapping. Phase unwrapping is time-consuming and requires a prior known displacement pixel as a reference, but that reference may introduce errors for subsequent parameter estimation [54]. PS-InSAR is a standard solution for deformation and DEM error signals separation. However, the sparsity of measurements becomes the main limitation, leaving it open to further research. This thesis first explores the PS-InSAR-based temporal-only signals estimation on each pixel location using the wrapped phase directly. Furthermore, we propose a more advanced solution that considers spatial information and gives dense per pixel signal measurements.

2.4 InSAR Data Simulator

One remaining challenge concerns how to precisely validate and eventually optimize the developed algorithms, as no ground truth data with a controlled environment is available. To date, the most commonly adopted evaluation strategy uses a couple of synthetic interferometric phases with simple geometric shape fringes [38] [37] or simulates the radar signals according to the pre-given DEM [20]. For filtering, coherence estimation tasks, non-InSAR data was also used for validation purposes in the early works [39][35]. The TS-InSAR deformation signal estimation task commonly adopts insufficient GPS data [54][58] or non-InSAR numerical optimization problems [57] as well for quantitative evaluation. A good InSAR simulator is important for objective evaluation and useful for developing a balanced training dataset for recent machine learning algorithms. Large-scale data is essential, especially if an application wants to take advantage of the recent learning-based approaches. It is known that the robustness of a trainable model’s learning capacity depends on diversity in datasets for sufficient training and testing in terms of the types of signal features and noise characteristics. Otherwise, a model could easily suffer from over-fitting, which biases the evaluation results. This field of research lacks large-scale datasets and benchmarks. In this thesis, we pursue designing an InSAR simulator that can generate large amounts of synthetic InSAR data, which is as realistic as possible.

Chapter 3

Adversarial training for DUAL-STAGE natural image denoising enhanced with feature matching

3.1 Introduction

Recently, deep neural networks (DNNs), especially deep convolutional neural networks (CNNs), have shown promising performance in image denoising. These deep-learning based discriminative methods show better performance than conventional model-based methods. However, when an image is distorted with high noise, these discriminative learning approaches are insufficient. In this thesis, we propose a dual-stage convolutional neural network, augmented with adversarial training, to address the shortcoming of current convolutional neural networks in image denoising. Our dual-stage approach - Adversarial Augmented Dual-stage neural Network (AADNet) includes two stages of discriminative denoising process: 1) a residual learning denoiser is used to generate a residual image from a noisy input image and construct a reference image, and 2) a full CNN model is designed to output the final denoised image by using the preliminary reference and noisy image pair. To improve the dual-stage denoising, we introduce an adversarial training framework. Adversarial training is first described in Generative Adversarial Nets (GANs) [71] and have become a popular approach. GANs are based on a two-player min-max game between two models. They aim to generate realistic synthetic images. Ideally, network discriminator is expected to learn distinct features from real data, which the generator can imitate. To achieve this, an optimal point should be reached when real data and synthesized data are indistinguishable. However, in practice, it is hard to train GANs as desired;

it is challenging to balance the relative capacities of the two models in an adversarial network. This is due to the lack of an unambiguous and computable convergence criterion. A solution is to add an augmented training process to GANs by directing the generator network towards a probable configuration of abstract discriminative features [72]. Different from current GANs, which aim to achieve generative tasks, our adversarial training is augmented with feature matching [72] and classification capability for image denoising, which enables our adversarial training to cover a diverse collection of image content. Our goal is to recover the latent clean image from its corrupted observation. The dual-stage denoising process uses trained data from the discriminator, while the discriminator receives input from the dual-stage denoisers for training and generates feedback. Our contribution lies in:

- Introducing a dual-stage net for image denoising, to address the shortcoming of single denoiser design deployed by existing CNN-based networks.
- Proposing adversarial training to provide feedback for the dual-stage image denoising. While current nets are trained for specific objects, our adversarial training framework can achieve better results with its feature matching and classification capability to cover a diverse collection of complex content with fine detail.

In our work, we propose an image denoising framework AADNet, which not only has the merits of existing methods, but also introduces a new dual-stage denoising net, augmented by adversarial training with embedded feature matching and classification.

3.2 Methods and Procedure

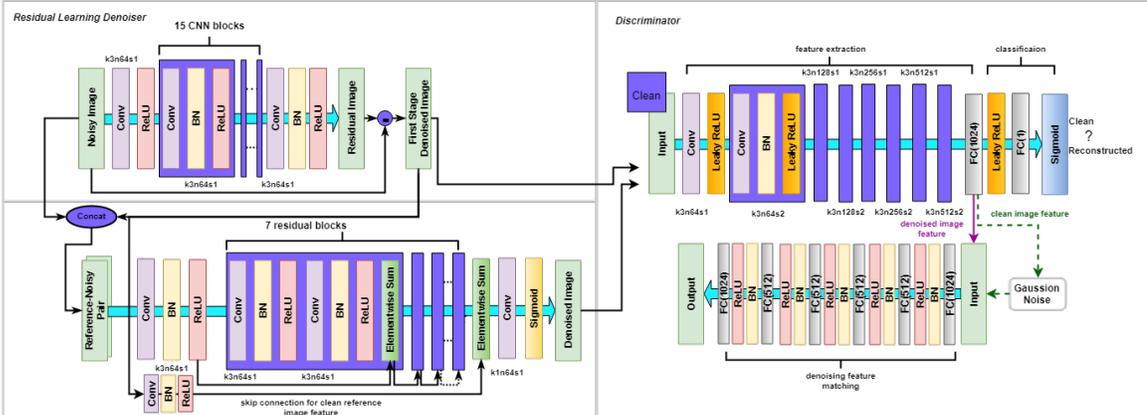


Figure 3.1: Architecture of proposed AADNet with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.

There are three major components in our AADNet: 1) A residual learning based denoiser (Fig.3.1 Top-left), 2) a full image reconstruction denoiser with skip-connections and ResNet blocks (Fig.3.1 Bottom-left), and 3) a discriminator for distinguishing denoised images from clean images (Fig.3.1 Right). All three components interact in an adversarial training environment. In a classical image degradation model $\mathbf{y} = \mathbf{x} + \mathbf{v}$ [10], the denoiser is trained to estimate the difference between the input noisy image \mathbf{y} and a clean image \mathbf{x} . At our first stage, a residual image $\hat{\mathbf{v}}$ is obtained from a noisy input image $\hat{\mathbf{y}}$. The preliminary output (reference) is denoted by $\hat{\mathbf{x}}_{s1} = \hat{\mathbf{y}} - \hat{\mathbf{v}}$. At the second stage, a full CNN model obtains a denoised image by inputting the reference & noisy image pair tensor (concatenating $\hat{\mathbf{y}}$ and $\hat{\mathbf{x}}_{s1}$ along the last dimension). The final denoised output is denoted by $\hat{\mathbf{x}}_{s2}$.

3.2.1 1st Stage: Residual Learning Denoiser \mathbf{D}_{s1}

\mathbf{D}_{s1} is trained to learn the mapping function $\mathbf{D}_{s1}(\mathbf{y}) = \mathbf{v}$. The clean image can be predicted as \mathbf{D}_{s1} 's output $\hat{\mathbf{x}}_{s1} = \mathbf{y} - \mathbf{D}_{s1}(\mathbf{y})$. The parameter in \mathbf{D}_{s1} is trained by computing the mean square error (MSE) or L2 loss (Eq.3.1), between the residual and the model predication.

$$\mathcal{L}_{L2}^{\mathbf{D}_{s1}}(\mathbf{x}, \mathbf{y}, \mathbf{D}_{s1}) = \|\mathbf{D}_{s1}(\mathbf{y}) - (\mathbf{y} - \mathbf{x})\|_2 \quad (3.1)$$

\mathbf{D}_{s1} is used for estimating the noise rather than reconstructing the de-noised image. Residual learning can be used for image denoising, but it usually over-filters and is not effective in recovering complex structure [10]. To resolve this issue, we introduce dual-stage denoising.

3.2.2 2nd Stage: Full Image Reconstruction Denoiser \mathbf{D}_{s2}

Existing end-to-end CNN based denoising methods use a noisy and clean image pair to train the model for full image reconstruction. In our AADNet, we have an output of \mathbf{D}_{s1} from the first stage as a preliminary denoised reference, which already has the Gaussian Noises partly or mostly removed. We concatenate the original noisy image and $\hat{\mathbf{x}}_{s1}$ to form a reference-noisy pair tensor as input to the (second stage) full image reconstruction denoiser \mathbf{D}_{s2} . The output is the final denoised image $\hat{\mathbf{x}}_{s2}$. Batch-normalization [73] and Res-Net[12] blocks, as well as Skip-connections [2], are used for improving training efficiency and stability. More detail about these techniques can be found in [12]. Note that, for an image denoising task, we want to avoid noisy information affecting subsequent layers during training. Therefore, different from other methods, which pass feature maps generated from the entire input, we select

feature maps generated from the preliminary denoised reference image only. These feature maps are passed to the end layer via skip-connection as illustrated in Fig.3.1 (bottom-left). The parameter in \mathbf{D}_{s2} is trained by minimizing the L2 loss (Eq.3.2) between the desired clean image and the model’s reconstruction.

$$\mathcal{L}_{L2}^{\mathbf{D}_{s2}}(\hat{\mathbf{x}}_{s1}, \mathbf{y}, \mathbf{D}_{s2}) = \|\mathbf{D}_{s2}(\hat{\mathbf{x}}_{s1}) - \mathbf{x}\|_2 \quad (3.2)$$

3.2.3 Adversarial Training with Feature Matching

AADnet is trained to recover the fine detail visually masked under noise degradation. We apply adversarial training using a discriminator \mathbf{D} to learn whether the input image is a clean image or an image reconstructed after denoising. We split \mathbf{D} into two components based on the denoising feature matching technique [72]. \mathbf{D} takes \mathbf{x} , $\hat{\mathbf{x}}_{s1}$ and $\hat{\mathbf{x}}_{s2}$ as input. A feature extractor then extracts a feature vector of length 1024 and then pass it to a classifier generating a logits for classification. The loss function for \mathbf{D} is given in Eq. 3.3:

$$\mathcal{L}_{\mathbf{D}}(\hat{\mathbf{x}}_{s1}, \hat{\mathbf{x}}_{s2}, \mathbf{x}, \mathbf{D}) = -\mathbb{E}[\log \mathbf{D}(\mathbf{x})] - \frac{1}{2}(\mathbb{E}[\log(1 - \mathbf{D}(\hat{\mathbf{x}}_{s1}))] + \mathbb{E}[\log(1 - \mathbf{D}(\hat{\mathbf{x}}_{s2}))]) \quad (3.3)$$

In order to maximize clean image content, we impose optimization *adv* loss $\mathcal{L}_{adv}^{\mathbf{D}_{s1}}$ and $\mathcal{L}_{adv}^{\mathbf{D}_{s2}}$ on each denoiser as shown in Eq. 3.4 & 3.5

$$\mathcal{L}_{adv}^{\mathbf{D}_{s1}}(\hat{x}_{s1}, \mathbf{D}) = \mathbb{E}[\log(1 - \mathbf{D}(\hat{\mathbf{x}}_{s1}))] \quad (3.4)$$

$$\mathcal{L}_{adv}^{\mathbf{D}_{s2}}(\hat{x}_{s2}, \mathbf{D}) = \mathbb{E}[\log(1 - \mathbf{D}(\hat{\mathbf{x}}_{s2}))] \quad (3.5)$$

As mentioned before, feature matching is necessary because current GANs are content or object specific [72], which is ineffective for real-world content covering a diverse collection of objects. A robust denoising net should target not only a narrow class of content, but also a wide collection of complex scenes. To address this issue, we embed a feature denoiser trained using extracted features from a feature extractor. Let F_{clean} denote a clean image feature vector. Feature matching denoiser \mathbf{FD} is trained to reconstruct the clean image feature vector from its corrupted version F_{clean}^C by minimizing the MSE loss function (Eq.3.6), where F_{clean}^C is generated by adding Gaussian noise to F_{clean} .

$$\mathcal{L}_{\mathbf{FD}}(F_{clean}, F_{clean}^C, \mathbf{FD}) = \|F_{clean} - \mathbf{FD}(F_{clean}^C)\|_2 \quad (3.6)$$

$F_{D_{s1}}$ and $F_{D_{s2}}$ denote feature vectors extracted from the dual-stage denoised output. When **FD** performs the inference process directly on $F_{D_{s1}}$ and $F_{D_{s2}}$, the difference between **FD**'s input feature vector and its output reconstructed feature vector is an additional cost to the two denoisers. Once **FD** is well trained with clean image features, minimizing the difference between denoised image features and its **FD** output will push the denoised image towards higher probability configuration following the clean data distribution in the feature space [72]. The two denoisers' feature matching loss functions are defined by Eq.3.7 & Eq.3.8 respectively.

$$\mathcal{L}_{\mathbf{FD}}^{\mathbf{D}_{s1}}(F_{D_{s1}}, \mathbf{FD}) = \|F_{D_{s1}} - \mathbf{FD}(F_{D_{s1}})\|_2 \quad (3.7)$$

$$\mathcal{L}_{\mathbf{FD}}^{\mathbf{D}_{s2}}(F_{D_{s2}}, \mathbf{FD}) = \|F_{D_{s2}} - \mathbf{FD}(F_{D_{s2}})\|_2 \quad (3.8)$$

Now the final loss functions $\mathcal{L}^{\mathbf{D}_{s1}}$ and $\mathcal{L}^{\mathbf{D}_{s2}}$ for the two denoisers have all the components as shown in Eq.3.9 & Eq.3.10. α , β and γ are hyperparameters. In the current implementation, we set them as 1, 10^{-3} and 10^{-3} respectively. All components described above can be trained in synchronization to form an end-to-end framework.

$$\mathcal{L}^{\mathbf{D}_{s1}} = \alpha\mathcal{L}_{\mathbf{L2}}^{\mathbf{D}_{s1}} + \beta\mathcal{L}_{adv}^{\mathbf{D}_{s1}} + \gamma\mathcal{L}_{\mathbf{FD}}^{\mathbf{D}_{s1}} \quad (3.9)$$

$$\mathcal{L}^{\mathbf{D}_{s2}} = \alpha\mathcal{L}_{\mathbf{L2}}^{\mathbf{D}_{s2}} + \beta\mathcal{L}_{adv}^{\mathbf{D}_{s2}} + \gamma\mathcal{L}_{\mathbf{FD}}^{\mathbf{D}_{s2}} \quad (3.10)$$

3.2.4 Learning Configuration

The parameters of all convolutional layers and fully connected layers are initialized using Xavier method [74]. Residual learning denoiser and Full image reconstruction denoiser are trained using Adam optimizer [75] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. Learning rate was set from 10^{-2} to 10^{-4} . The feature matching denoiser was also trained by Adam optimizer with the same configuration but learning rate was set from 10^{-5} to 10^{-6} . The discriminator was trained using stochastic gradient descent (SGD) with momentum 0.9. The reason why not use Adam optimizer for training discriminator is for improving the stability of the adversarial training. This is suggested by recent empirically studies related to GAN [76][77]. The learning rate was set from 10^{-1} to 10^{-4} . The network was trained using 40 epochs with min-batch size of 64.

3.3 Results and Discussion

We evaluate our AADNet using two well-known public benchmark datasets: 14 widely used images (Fig.3.2) and the BSD200 dataset [2]. The network was implemented using TensorFlow-1.4 [78] GPU version. All experiments were run on Compute Canada

Cedar GPU instance with NVIDIA P100 Pascal graphic processor and Intel Xeon E5-2650 v4 CPU. The network took about two days for training 40 epochs.



Figure 3.2: The 14 widely used test images

Table 3.1: Average PSNR and SSIM results of σ 30, 50, 70 for 14 widely used images.

σ	BM3D	EPLL	NCSR	PCLR	PGPD	WNNM	RED30	DnCNN-S	AADNet
PSNR									
30	28.49	28.35	28.44	28.68	28.55	28.74	29.17	29.20	29.30
50	26.08	25.97	25.93	26.29	26.19	26.32	26.81	26.81	27.11
70	24.65	24.47	24.36	24.79	24.71	24.80	25.31	25.32	25.58
SSIM									
30	0.8204	0.8200	0.8203	0.8263	0.8199	0.8273	0.8423	0.8377	0.8420
50	0.7427	0.7534	0.7415	0.7538	0.7442	0.7517	0.7733	0.7669	0.7830
70	0.6882	0.6717	0.6871	0.6997	0.6913	0.6975	0.7206	0.7206	0.7318

3.3.1 Training and Testing

Similar to related work in the literature, we used gray-scale images. We applied three noise levels, i.e., $\sigma = 30, 50$ and 70 , to train AADNet with additive white Gaussian noises (AWGN). 300 images from the Berkeley Segmentation Dataset were used. Preprocessing was done by cropping each image to 40×40 patches with stride of 10. The output was then randomly scaled, flipped and rotated to produce a larger training set (about 0.45M). For each noise level, we used extracted patches as ground truth and added AWGN to get noisy samples for training. All the patches were scaled to $[0,1]$. The reason to choose AWGN is because 1) there is no specific prior information on noise score, and 2) real-world noise can easily be approximated

Table 3.2: Average PSNR and SSIM results of σ 30, 50, 70 for BSD200 images.

σ	BM3D	EPLL	NCSR	PCLR	PGPD	WNNM	RED30	DnCNN-S	AADNet
PSNR									
30	27.31	27.38	27.23	27.54	27.33	27.48	27.95	28.55	28.61
50	25.06	25.17	24.95	25.30	25.18	25.26	25.75	26.30	26.43
70	23.82	23.81	23.58	23.94	23.89	23.95	24.37	24.94	25.08
SSIM									
30	0.7755	0.7825	0.7738	0.7827	0.7717	0.7807	0.7993	0.8152	0.8195
50	0.6831	0.6870	0.6777	0.6947	0.6841	0.6928	0.7167	0.7323	0.7451
70	0.6240	0.6168	0.6166	0.6336	0.6245	0.6346	0.6551	0.6785	0.6913

locally as AWGN [2]. During testing, there is no need to perform the feed-forward computation on our discriminator. The final denoised output is obtained by running inference on the dual-stage denoisers. Although the model is trained on local patches, there is no fully connected layer in our denoiser net. Note that AADNet can perform denoising on arbitrary image size.

3.3.2 Evaluation

To objectively assess the performance of AADNet, we use Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM) to measure dissimilarity between the original noise-free and denoised images. BM3D [4], EPLL [79], NCSR [80], PCLR [81], PGPD [82], WMMN [83], RED30 [2], and DnCNN-S [10] are compared with our AADNet. We used the publicly available source code of DnCNN-S to generate its scores. Scores of other methods are taken from [2]. Table 3.1 & 3.2 reports the PSNR and SSIM results of $\sigma = 30, 50,$ and 70 on the 14 commonly tested images and BSD200 images. Some visual examples are shown in Fig.3.3. The zoom-in view at the bottom right of each image demonstrates that AADNet preserves fine detail better.

3.4 Conclusions

Quantitative (PSNR, SSIM) and qualitative (visual) comparisons show that our AADNet achieves better results at all test noise levels in the BSD200 dataset and the 14 images widely used in the literature. There are only a few methods outperforming BM3D by more than 0.3dB [84], but AADNet has a higher PSNR (almost 1.0dB) com-

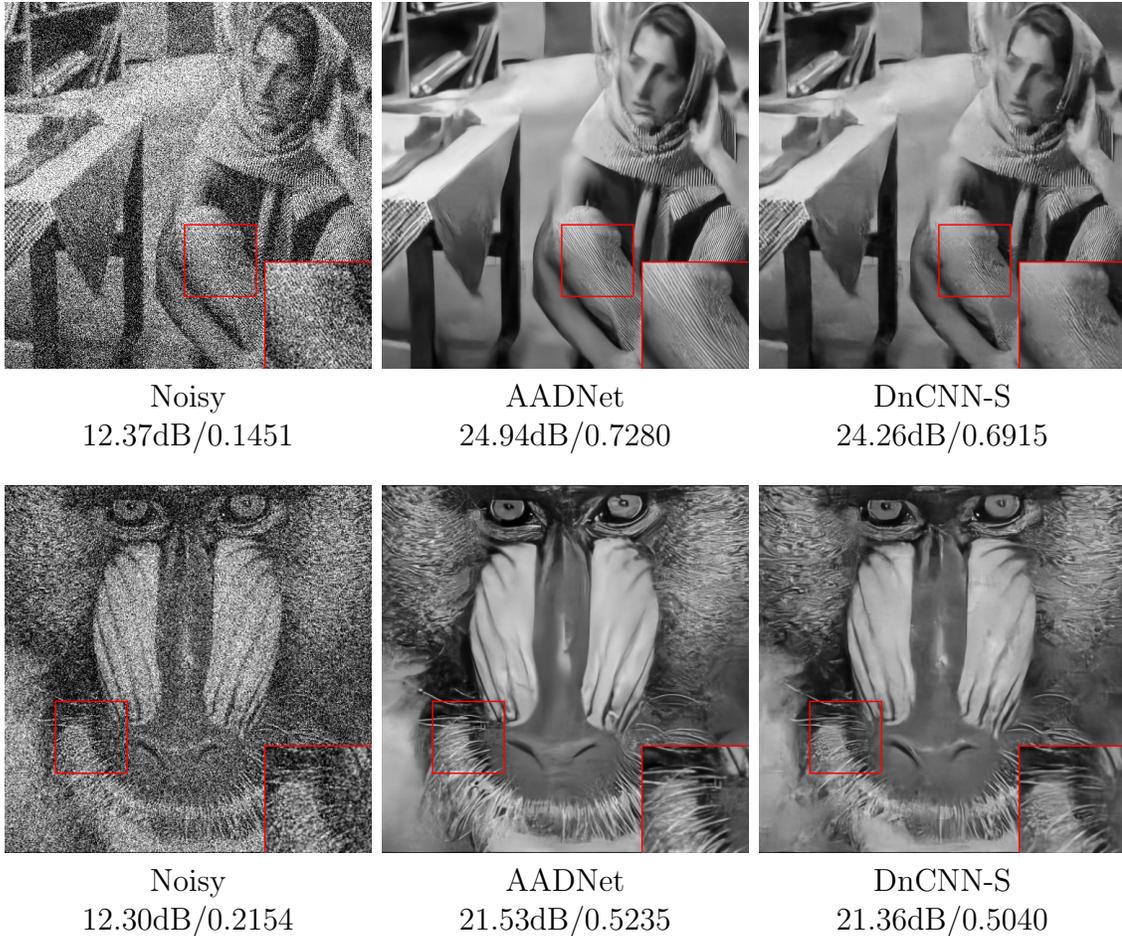


Figure 3.3: Visual comparison examples - (Left to right): noisy image ($\sigma = 70$), denoised by AADNet and denoised by DnCNN-S. The zoom-in view shown at the bottom right of each image demonstrates that AADNet preserves fine detail better.

pared with BM3D. When comparing to a pure residual learning network DnCNN-S, AADNet also outperforms at all noise levels. SSIM results indicate that AADNet surpasses other methods by a large margin, especially under high level noise conditions. Visual comparisons in Fig.3.3 show that AADNet can preserve sharp edges and fine structural details better than the original DnCNN-S on high noise level ($\sigma = 70$). Therefore, we conclude that AADNet is more robust than other methods in recovering image detail. AADNet’s better performance is attributed to our dual-stage denoising and adversarial training strategy.

In this work, we propose a novel adversarial augmented dual-stage denoising network, which outperforms existing state-of-the-art conventional and learning based approaches. The dual-stage is composed of residual learning for preliminary noise removal, and full image reconstruction for recovering fine detail. Dual-stage denoising

is complemented by adversarial training, with embedded feature matching and classification to augment learning. Experimental results confirm that our method gives better results at all test noise levels. AADNet is expected to perform well on image restoration tasks in general.

Chapter 4

SAR Interferometric Phase Restoration and Coherence Estimation

4.1 Introduction

Synthetic Aperture Radar (SAR) is a remote sensing technology that uses active microwaves to capture ground surface characteristics. An Interferometric SAR (InSAR) image or interferogram is created from two temporally separated single look complex (SLC) SAR images via the point-wise product of one SLC image with the complex conjugate of the other SLC image. Thus, each pixel in an interferogram indicates a phase difference between two coregistered SLC images. The phase difference encodes useful information, including deformation of the earth's surface and topographical signals, and has been successfully used to obtain the digital elevation model (DEM). InSAR final products have various purposes, such as civil engineering, topography mapping, infrastructure, oil/gas mining, natural hazards monitoring, and elevation change detection. In any SAR system, as the satellite circumnavigates the earth, the SAR sensor launches millions of radar signals toward the earth in the form of microwaves. The SAR image is represented as an SLC image generated from radar information echoed back from the ground. However, different ground surface compositions impact these radar signals in various ways. Some are reflected away from the satellite, some are absorbed by non-reflective materials, and some are reflected back to the satellite. Signal reflections can be noisy, resulting in SAR images with strong speckle noise. Furthermore, temporal and spatial variations between two SLC acquisitions cause decorrelation, affecting the interferometric phase [85]. Noisy SAR images make the interferometric phase filtering step on their output InSAR image more challenging. It is important to point out that the quality of the estimated inter-

ferogram directly affects to the whole processing pipeline. The phase noise affects all subsequent stages, from the phase-unwrapping operation to the motion signal modeling [29]. Therefore, restoration of interferometric phase image becomes a fundamental and crucial step to ensure measurement accuracy in remote sensing. In this regard, the coherence map of the interferogram is a crucial indicator showing the reliability of the interferometric phase [35]. Thus, interferometric phase filtering and coherence estimation are the main focuses of this thesis.

4.1.1 Phase Noise Model

Similar to the classic additive degradation mode in natural image restoration problem, an interferometric phase can also be characterized by the following equation:

$$\theta_y = \theta_x + v, \quad (4.1)$$

which has been validated in [15]. θ_y denotes the noisy observation, θ_x is clean phase component and v is the noise with zero mean and σ standard deviation, θ_x and σ are independent of each other. It follows a similar definition in the natural image analysis that clean signals are independent from noise signals. Unfortunately, it is not feasible to use natural image processing algorithms directly in the interferometric phase domain, because of branch cuts. According to the SAR interferometric phase calculation, the range of interferometric phase is within $[-\pi, \pi)$, which means that wrapped phase value could jump from negative to positive or positive to negative π , and they could represent high-frequency motion signals that should be well preserved. Therefore, we process the interferometric phase in the complex domain [20, 28] in this thesis. In other words, the phase noise model could be represented by real and imaginary channels, which are continuous values:

$$\begin{aligned} y_{Real} &= \cos(\theta_y) = Q\cos(\theta_x) + v_r = Qx_{Real} + v_r \\ y_{Imag} &= \sin(\theta_y) = Q\sin(\theta_x) + v_i = Qx_{Imag} + v_i. \end{aligned} \quad (4.2)$$

The noisy phase observation θ_y is decomposed into two components y_{Real} and y_{Imag} . v_r and v_i are zero-mean additive noise in the real and imaginary parts, and they are independent of the underlying clean phase signals θ_x . Q is a quality indicator that monotonically changes with coherence level [20]. We designed our filtering network based on the above complex phase model. During training, the network learns to filter both real and imaginary parts, and then the estimated clean phase $\tilde{\theta}_x$ can be reconstructed from filtered \tilde{x}_{Real} and \tilde{x}_{Imag} as follows:

$$\tilde{\theta}_x = \arctan\left(\frac{\tilde{x}_{Imag}}{\tilde{x}_{Real}}\right). \quad (4.3)$$

4.2 DeepInSAR—A Deep Learning Framework for SAR Interferometric Phase Restoration and Coherence Estimation

4.2.1 Method and Procedure

In this section, we describe our proposed DeepInSAR in detail. The main goal is to establish and validate the idea of using deep learning method to automate and accelerate both interferometric phase filtering and coherence estimation, which are conducted separately in most of existing approaches. Recently, deep learning studies especially CNNs have been dominating various fields of vision-related tasks. Generally, their excellent performance can be attributed to their powerful feature classification and ability to learn image priors during the training stage. The reasons why we choose to use CNN for InSAR filtering and coherence estimation are (1) CNN is effective for capturing spatial feature characterization with a lot of trained parameters, (2) many achievements in deep learning can be borrowed to benefit better training and generalization, as well as to speed up and improve the output data quality, and (3) powerful GPUs could speed up CNN training and runtime inference. Deep CNN is well suited to be deployed on modern GPUs for parallel computation. All these advantages make deep learning techniques promising for InSAR phase filtering and coherence estimation, where real-time processing and high-quality outcome of large resolution radar images are required.

Figure 4.1 illustrates the architecture of the proposed DeepInSAR network. At a high-level, our deep model includes multiple modules for handling different sub-tasks. The amplitudes and their interferometric phases of two SLC SAR images are combined by concatenating into a single tensor during a preprocessing step. The output is subsequently fed into a densely connected feature extractor. Dense connectivity helps extract useful features under different scales and composite multi-scale features are suitable for different end tasks [86]. Two *feature to image* transformations are achieved by sub-networks performing—(1) phase filtering using residual learning strategy [10] and (2) coherence estimation. The model is expected to learn optimal discriminative functions, mapping from noisy observations to both latent clean phase signals and coherence, by a feed-forward neural network.

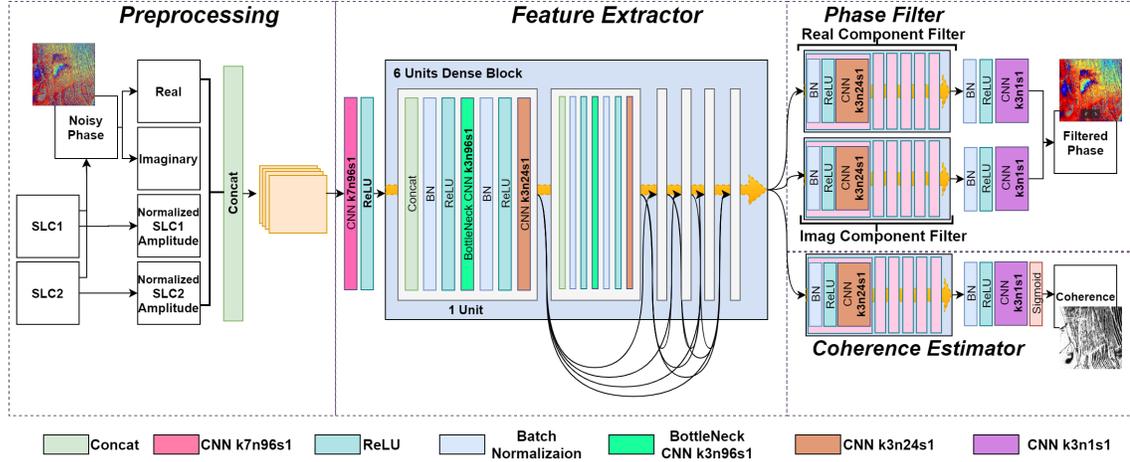


Figure 4.1: The architecture of the proposed Deep Interferometric Synthetic Aperture Radar (DeepInSAR) network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each Convolutional Neural Network (CNN) layer

Preprocessing of Radar Data

Referring to our noise model in Equation (4.2), we propose to fully utilize all the information from two SLCs rather than only analyzing interferometric phase. As shown in the *Preprocessing Module* in Figure 4.1, the raw input contains two noisy co-registered SLC SAR images S_1 and S_2 . Interferometric phase image I is calculated as:

$$I = (A_{S_1} \odot A_{S_2}) e^{j(\varphi^{S_2} - \varphi^{S_1})} = A_I e^{j\Delta\varphi}, \quad (4.4)$$

where A is amplitude and φ is phase. In fact, the phases in SLC images look like random noise from one pixel to another because each pixel is a complicated function of scattering features located on the ground surface. However, interferometric phase $\Delta\varphi$ represents phase-difference fringes illustrating changes in distance between ground and satellite antenna, which are valuable information needed for InSAR related applications, but they are often contaminated by noise. Intuitively, we want to incorporate amplitude images, because they usually show recognizable patterns like buildings, mountains, and valleys, which are useful spatial characterizations and hence informative for denoising and coherence estimation. For phase filtering, our proposed DeepInSAR aims to learn a mapping function $\mathcal{F}_{oc} : observation \mapsto clean$. As shown in Equation (4.2), \mathcal{F}_{oc} can include noisy y_{Real} , y_{Imag} and Q as observations. In this work, we further use two SLC's amplitude value to replace the Q in the observations, because we learn from Reference [87] that coherence magnitude $|\gamma|$ can be approximated based on two SLC's amplitude:

$$|\gamma| = \frac{|\sum_{m=1}^M \sum_{n=1}^N A_{S1}(m, n) A_{S2}(m, n)|}{\sqrt{\sum_{m=1}^M \sum_{n=1}^N |A_{S1}(m, n)|^2 \sum_{m=1}^M \sum_{n=1}^N |A_{S2}(m, n)|^2}}, \quad (4.5)$$

where M, N represent estimator window size. This widely used coherence estimator shows a potential mapping $(A_{S1}, A_{S2}) \mapsto |\gamma|$. Moreover, As mentioned in Section 4.1.1, Q is related to $|\gamma|$. Here we hypothesize that there is a mapping chain $(A_{S1}, A_{S2}) \mapsto |\gamma| \mapsto Q$. Hence, instead of using any handcrafted sampling estimator to estimate Q . We proposed to use a deep model to approximate the mapping function \mathcal{F}_{oc} , in a simplified end-to-end manner by treating both SLC amplitudes together with interferometric phase as input observation to the network. Theoretically, sufficient and well-reasoned input would help the model learn a proper mapping function to estimate latent clean signals more precisely. The same should also support estimating the quality of signals (coherence).

Unfortunately, in real-world SAR image, the range of amplitude values could be extremely broad, that is, from 0 to 1×10^6 , and the scale of the values also varies across different target sites and types of radar sensor. This is one of the reasons why learning-based studies are not pursued for SAR analysis because using uncontrolled amplitude values to train a deep discriminative model is not effective. In general, the learning-based method requires each input dimension to have a similar distribution with low and controlled variance, which has been suggested by many deep learning studies [10, 74]. Unnormalized input data can lead to an awkward loss function topology and place more emphasis on certain parameter gradients resulting in a poor training. Hence, for a CNN layer, all the input pixels should be in the same scale. The amplitude values in raw SAR images are not suitable as input data for a deep model. In this work, we introduce an adaptive method to normalize all amplitude values to lie between 0 to 1. The model saturates potential outliers as well as keeps most dynamic changes in the original image without destroying or cutting off any essential ground characteristics.

Knowing that if data roughly follows a normal distribution, the standard Z score of each data point can be calculated as the position of a raw score in terms of its distance from the mean, when measured in standard deviation units [88]. However, SAR amplitude values follow Rayleigh distribution [89] with potential extremes in the distribution tail. Hence, the mean is not statistically robust in our case, and it is easily influenced by outliers. In this study, we apply a modified Z score [90] which estimates Z score based on Median Absolute Deviation (MAD). The MAD value of SLC amplitude A is calculated as:

$$MAD = \text{median}(|A_i - \tilde{A}|), \quad (4.6)$$

where \tilde{A} is the median of the data. Next, we transform the data into the modified Z score domain:

$$A_i^{mz} = \frac{0.6745 * (A_i - \tilde{A})}{MAD}. \quad (4.7)$$

A^{mz} represents each pixel’s modified Z score and 0.6745 is the 0.75th quartile of the standard normal distribution, to which the MAD converges. For outlier detection, researchers commonly use absolute values of modified Z scores to threshold the data, where data points with $|Z|$ score greater than 3.5 are potential outliers and are ignored [90]. In Figure 4.2, there are 6 SLC amplitude images selected from three real-world datasets captured by TerraSAR-X in StripMap mode [91], with 2 SLCs taken at different time for each stack. By observing their raw amplitude values and histograms as shown in the 1st and 2nd rows of Figure 4.2, data points are close to Rayleigh distribution as mentioned above. So simply cutting off according to the modified Z score might cause loss of information located on the right tail of high amplitude values. Although logarithm transformation could help us visualize the images better, there is no fixed base number for all images because they might differ by order of magnitude. In our proposed normalization method, we adopt modified Z score as a transformation function to force all values to be close to 0 first and then all potential outliers will be far from 0 and greater than 3.5. To give a standard input data distribution for training the neural network, we apply a hyperbolic tangent \tanh non-linear function as:

$$\hat{A} = \frac{1}{2}(\tanh(\frac{A^{mz}}{7}) + 1) \quad (4.8)$$

to bind all input amplitudes with a controlled variance. A good property of hyperbolic tangent $\tanh(x)$ function is that the input value between -1 to 1 will be enhanced and others will be saturated. In our case, we divide A^{mz} by 7 (two times of 3.5) to make the majority of data points lie between -1 to 1 . Then ground characteristics could be potentially enhanced after \tanh operations. Secondly, data points with relatively high amplitude are still kept on the right tail, and for those extremely high values, likely outliers, are saturated close to 1. Note that, we further normalize the transformed data to the range $[0, 1]$, because we use a Rectified Linear Unit (ReLU) activation for introducing nonlinearity in the CNN to learn complex features. Non-negative input is recommended to avoid saturated neuron at an early training stage when using ReLU activation in the early layers [92]. As shown in the 3rd row in Figure 4.2, after our

proposed data normalization, all amplitude values lie in the range 0 to 1 are properly delivered without losing and breaking essential details. One can also observe this in the 4th row of Figure 4.2. The final observation \mathbf{o} is a tensor $[y_{real}, y_{imag}, \hat{A}_{S1}, \hat{A}_{S2}]$, and is the input to the proposed DeepInSAR.

Filtering with Residual Learning

Residual learning is designed for solving performance degradation problem on very deep neural networks [93]. In our interferometric phase filtering, we apply a similar idea but without using too many skip-connections within the network. We only create identity shortcuts for predicting the residuals of both real and imaginary channels. Instead of directly outputting the estimated clean components, the proposed model is trained to predict residuals. The model implicitly filters the latent clean signals with hidden operations within the deep neural network. For each of the real and imaginary channels, we have the loss function below:

$$\begin{aligned}\mathcal{L}(\mathbf{W}_{fe}, \mathbf{W}_{real}) &= \frac{1}{2} \|\mathcal{R}_{real}(\mathbf{o}; \mathbf{W}_{fe}, \mathbf{W}_{real}) \\ &\quad - (y_{real} - x_{real})\|_F^2 \\ \mathcal{L}(\mathbf{W}_{fe}, \mathbf{W}_{imag}) &= \frac{1}{2} \|\mathcal{R}_{imag}(\mathbf{o}; \mathbf{W}_{fe}, \mathbf{W}_{imag}) \\ &\quad - (y_{imag} - x_{imag})\|_F^2,\end{aligned}\tag{4.9}$$

where \mathbf{W}_{fe} , \mathbf{W}_{real} and \mathbf{W}_{imag} are the trainable parameters in the model corresponding to feature extractor, real and imaginary channels respectively. For both real and imaginary channels filtering, during the training iterations, our model aims to learn a residual mapping $\mathcal{R}(\mathbf{o}) \approx y - \frac{y-v}{Q}$ according to our noise model (Equation (4.1)). Then the clean components can simply be reversed by $x = y - \mathcal{R}(\mathbf{o})$. (y, x) represents noise-free training sample (patch) pairs. Residual mapping is much easier to learn than the original unreferenced mapping. It has been shown to output excellent results in many low-level vision image inverse restoration problems such as image super-resolution [94] and image denoising [10]. To the best of our knowledge, we are the first to use residual learning and CNN to do InSAR phase filtering. The model now learns a residual mapping $\mathcal{R} : observations \mapsto residuals$ on real and imaginary channels respectively. Furthermore, it is known that phase noise variance σ_θ^2 could be approximated by coherence magnitude $|\gamma|$ [87]:

$$\sigma_\theta^2 = \frac{\pi^2}{3} - \pi \arcsin(|\gamma|) + \arcsin^2(|\gamma|) - \frac{Li_2(|\gamma|^2)}{2},\tag{4.10}$$

where Li_2 is Euler’s dilogarithm. Our input tensor for phase filtering includes two SLCs’ amplitudes, which correlated to coherence magnitude. Hence, our designed observation input is well-reasoned for predicting phase residuals.

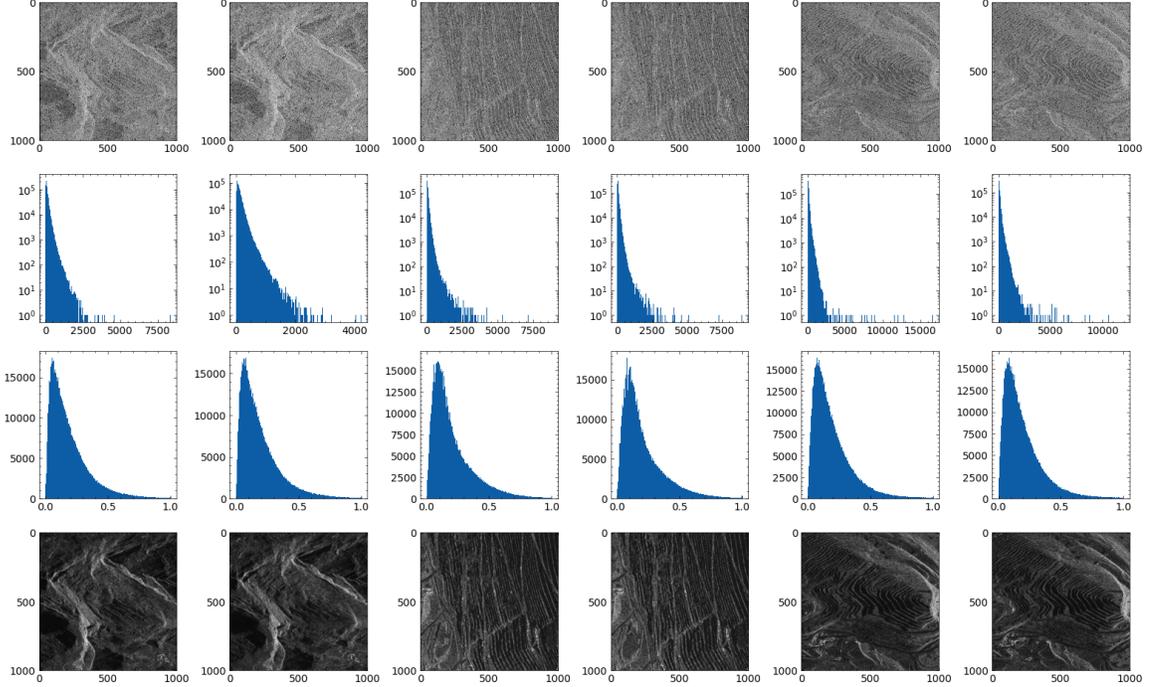


Figure 4.2: Before and after preprocessing: amplitude images selected from three real-world site datasets. From left to right, it shows Site-A (1st and 2nd columns), Site-B (3rd and 4th columns), Site-C (5th and 6th columns) with two samples for each dataset. (1st row) Raw amplitude images after log transformation for better visualization, (2nd row) their corresponding histograms in log, (3rd row) histograms after proposed normalization and (4th row) corresponding normalized images.

Coherence Estimation

Coherence map is estimated from two co-registered SAR images and is usually used as an indicator of phase quality. Demarcation of image regions based on the degree of contamination (“coherence”) is an important component of the InSAR processing pipeline. 0 coherence denotes complete decorrelation. On the other hand, successful and accurate deformation is measurable with high coherence. Lower quality of interferometry corresponds to decreasing coherence level and increasing level of noise on the phase. Interferometric fringes can only be observed where image coherence prevails. Filtered output is usually combined with coherence map for further processing, because coherence map could tell how much useful signals are potentially within this area. Some of the filtering studies also require coherence map in the

filtering process. However, most of them use Maximum Likelihood (ML) estimator (Equation (4.5)) or its extensions, which are usually significantly biased when using small window sizes. These methods can lose resolution and increase computational cost with large window sizes. Generally speaking, an area on the ground is treated as coherent, when it appears to have similar surface characterization within all images under analysis. However, between two SAR acquisitions, subareas will decorrelate if the land surface is disturbed. Therefore, CNN is a very good candidate to handle this spatial and non-local based analysis, especially on our input o , where almost all necessary information is available for learning the features and capturing mapping functions. During training, the model can learn to capture prior knowledge on all training samples and represent the knowledge as network weights. Intuitively, our method takes a more reliable and robust non-local analysis compared to conventional non-stack based work, which only considers one interferogram. It is also more time efficient than stack-based method because there is no requirement for doing heavy runtime analysis after training is done. In our model, we have a separate module in the proposed DeepInSAR for coherence estimation by using the same features extracted from observations \mathbf{o} as shown in Figure 4.3. Because coherence lies in the range $[0,1]$, we calculate sigmoid cross entropy loss, given logits obtained from last convolution layer’s output $\mathbf{c} = \mathcal{F}_{oh}(\mathbf{o}; \mathbf{W}_{fe}, \mathbf{W}_{coh})$:

$$\mathcal{L}(\mathbf{W}_{fe}, \mathbf{W}_{coh}) = \mathbf{z} * -\log(\sigma(\mathbf{c})) + (1 - \mathbf{z}) * -\log(1 - \sigma(\mathbf{c}))$$

$$\text{where } \sigma(\mathbf{c}) = \frac{e^{\mathbf{c}}}{e^{\mathbf{c}} + 1}. \quad (4.11)$$

\mathbf{z} is the reference coherence map that can be pre-calculated by any existing coherence estimator in order to generate training dataset for real images.

Shared Feature Extractor with Dense Connectivity

Natural images exhibit repetitive patterns, such as geometric and photometric similarities, which provide cues to improve the filtering performance. This concept is also valid for InSAR interferometric phase and SAR amplitude images. However, it should be noted that though, CNNs perform well for visual related tasks, it is known that as CNNs become increasingly deep, both input and gradient information can vanish and “wash out.” Recent work ResNet [93, 95] have addressed this problem by building shorter connections between layers close to the input and those close to the output. By doing this, CNNs can be substantially deep but still have accurate performance as well as efficient training. We adopt a dense connected CNN introduced in Reference [86] as a shared feature extractor before the real-imaginary filter

and coherence estimator. In the single-look interferometric phase, the latent noise level is related to the coherence magnitude [87]. A shared feature extractor for both phase filter and coherence estimation modules is expected to capture this relationship in latent space because weights in the feature extractor \mathbf{W}_{fe} are updated based on the gradient feedback back-propagated from both phase residual prediction and coherence estimation as shown in Figure 4.3. During training, the model can encode non-local image prior by updating network parameters according to both phase filter and coherence estimator loss. After training, the model can directly produce filtering and coherence output with a learned discriminative network function without any runtime non-local analysis.

Furthermore, because of the dense connectivity, our feature extractor follows multi-supervision that learns to extract common feature parameters for all related subsequent tasks [50]. In case of dense connectivity, each layer in the feature extractor is connected to every other layer in a feed-forward manner. During gradient back-propagation, each layer’s weight is updated based on all subsequent layers’ gradients [86]. As shown in Figure 4.1, features extracted by each layer in the feature extractor module of DeepInSAR are based on all preceding layers’ output. At the same time, its own output is passed to all subsequent layers as input. In our network, all feature maps extracted at different depth levels are passed to both phase filter and coherence estimator as a single concatenated tensor. Note that, as per deep CNNs’ working mechanism, early layers extract most detailed and low complexity features with a small perceptual field. With increasing depth, later layers in the feature extractor start extracting high level and complex features with a larger perceptual field. Therefore, a densely connected CNN feature extractor allows each sub-module to perform its own task with multi-scale and multi-complexity features. The proposed DeepInSAR also achieves a deep supervision by allowing each layer in the feature extractor to have direct access to the gradients from both sub-modules. Dense connectivity guarantees the model to get better feature propagation and enables feature reuse and fusion, which is really important for InSAR phase filtering and coherence estimation. In real-world images, ground data sites contain very different scale level characteristics. That is why most existing methods require user-defined window sizes to extract image characteristics. Therefore, all these methods suffer from the inability to choose a generic optimal window size, and fail to automatically generalize to different data sites. In our case, we use a dense CNN based feature extractor to intelligently select the best multi-level features for subsequent modules. The experiments in Section 4.2.3 show that our model is capable of generalizing on phase filtering and coherence estimation for different scale features in one image, as well as performing effectively

on new site images.

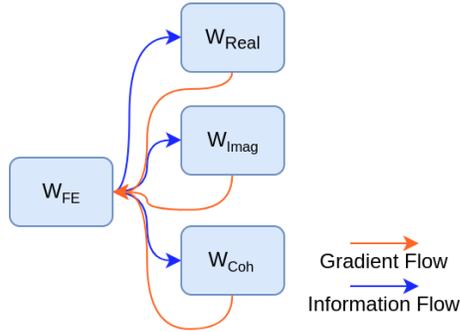


Figure 4.3: Information and Gradient flow between modules.

Teacher-Student Framework

Based on our findings, the main reason why deep learning techniques have not been pursued widely in InSAR filtering and coherence estimation so far is the lack of ground truth image data (reference without noise) for training such models. For training our proposed DeepInSAR model, we need image pairs as described in Section 4.2.1. However, there is no ground truth for real-world InSAR images. Therefore we introduce a teacher-student framework to make it feasible to train DeepInSAR for real-world images. From the literature, stack-based methods, like PtSel [96], always give reliable results. PtSel is an industry level algorithm for coherence estimation and interferometric phase filtering, which searches similar pixels across a stack of interferograms in both spatial and temporal domains. There are three key steps of PtSel algorithm (Figure 4.4) to generate the coherence map for a stack of interferograms. Next, the filtering process is replacing each interferograms’ pixel data by the weighted mean of the phase values of its neighboring pixels, where the weight is the PtSel generated coherence value. Despite the accuracy of stack-based methods, it requires historic SLCs and intensive online parallel searching using a high-end GPU farm, which limits its ability to be integrated into a time-critical InSAR processing chain. The stack-based methods have to wait for several months to collect sufficient data before it can start processing a new site. Although existing stack or non-stack based methods are powerful, most of them require human expert to ensure intermediate output quality because they are incapable of automatically detecting and removing all possible real-world noise patterns from InSAR data.

We introduce a deep neural network to replace the manual pre-processing, that is, feature extraction; and post-processing, that is, quality inspection, with a single intelligent trainable model. Similar to training an object classification neural network

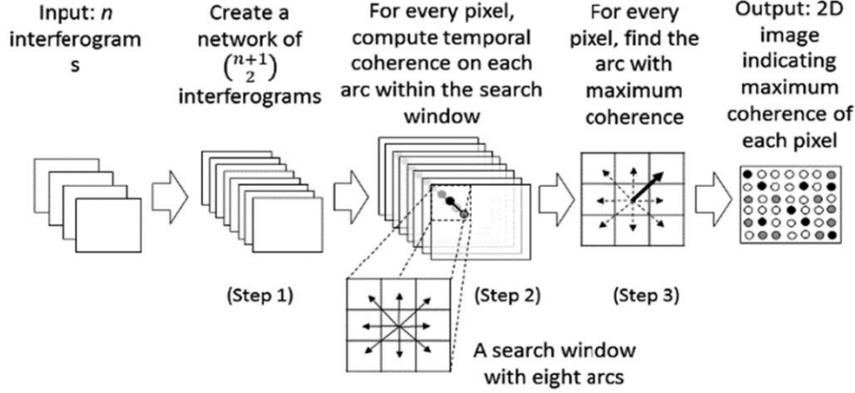


Figure 4.4: Illustration of the PtSel method describing the three key steps in order [96].

model, a large human labeled dataset is required in our approach. Human thus acts as a teacher to teach the model how to classify objects by providing human labeled data. For InSAR phase restoration and coherence estimation, we adopt the PtSel method to create filtered phase images for reference, coherence maps with human tuning and full stack processing to make sure the results are sufficiently reliable. The detail of the PtSel algorithm and its GPU implementation can be found at References [96, 97]. In our approach, PtSel with expert supervision becomes the teacher of the proposed DeepInSAR model, which is a student. We are able to demonstrate that, after training, (1) the student DeepInSAR can generate on par or even better results than its teacher method—PtSel, using the same test data sets, (2) our model only requires feed-forward inference on a single pair of SLCs, while PtSel requires more than thirty SLCs; and (3) our model can output filtering and coherence results after a one pass computation, while PtSel requires back and forward tuning processes and needs the phase unwrapping step, which is time consuming.

4.2.2 Experimental Setup

We compared our method with a number of other non-stack based methods, which can also perform both phase filtering and coherence estimation. They are (1) BoxCar filter, (2) NL-SAR [39] and (3) NL-InSAR [35]. We used publicly available implementations of these methods found in <https://github.com/gbaier/despeckCL>. Note that all parameters were set, when applicable, as suggested by the authors of the original papers, or else chosen to optimize the performance. We implemented the proposed DeepInSAR using Tensorflow-GPU 1.10; the code is available at: <https://github.com/Lucklyric/DeepInSAR>. In order to maximize the randomness of the

training patch samples, for a given training dataset, the model was trained on randomly extracted image patches with a size of 128×128 on the fly [98]. Network parameters were updated using Adam optimizer with a batch size of 64 and 0.001 initial learning rate. The model was trained on two NVIDIA 1080 GPUs for 6 hours with 1.6×10^5 iterations. To fairly compare the computational time, we executed all methods on the same GPU with an i7-8700K processor and 32GB RAM. It is worth noting that we built and trained our model using common hyper-parameter settings in our experimental setup because the work presented in this thesis is mainly for validating the feasibility of using deep learning techniques to do InSAR phase filtering and coherence estimation. It is expected that more extensive hyper-parameter tuning will further improve the performance of our proposed deep model based on the findings in References [86, 94]. We conducted our experiments using both simulated and real-world data to assess the effectiveness and robustness of the proposed model. In this section, we also discuss learning capacity and generalization ability, which are essential criteria for evaluating a learning model.

4.2.3 Results

Results on Simulation Data

In this section, we present quantitative results using simulated data. Simulated data allows us to evaluate the filtered quality in a controlled environment by comparing with the simulated ground truth. Ground truth is treated as an optimal teacher for training our proposed DeepInSAR; we can objectively demonstrate our model’s capability to learn proper phase filtering and coherence estimation for new simulated testing images, with ground truth available. The simulation strategy is similar to the work for generating the interferometric phase in Reference [38]. Instead of synthesizing a limited known patterns, the additional advantage is to extend the simulation for randomly generated irregular motion signals, ground reflective phenomena, as well as non-stationary noisy conditions. We designed a synthetic InSAR generator to randomly simulate a pair of SLC SAR images with the following procedure:

- Generate first SLC image S_1 with 0 phase value. The amplitude value grows from 0.1 to 1 from the left-most column in the image to the right column following a Rayleigh distribution. This leads to a linearly growing of coherence from left to right.
- Generate second SLC image S_2 by adding random Gaussian bubbles as synthetic motion signals to the phase. The amplitude value is equal to S_1 ’s amplitude value.

- Add random low-value amplitude bands (less than 0.3) on S_1 and S_2 to simulate stripe-like low amplitude incoherence areas.
- Generate noisy SLCs S_1^{noisy} and S_2^{noisy} by adding independent additive Gaussian noise v to both real and imaginary channels of S_1 and S_2 .
- Calculate clean and noisy interferometric phase I and I^{noisy} .
- Calculate ground truth coherence using clean amplitude, phase, and the standard deviation of base noise v .

Our simulated image generator includes a set of parameters for controlling the complexity of the interferometric phase at different distortion levels. We generated 18 different configurations, by combining (1) three base Additive White Gaussian Noise (AWGN) levels of v (S1, S2, S3), (2) three fringe frequency levels of phase fringes (F1, F2, F3), and (3) with or without low amplitude strips (S, NS). For example, the dataset, which has a relatively high level of base noise, and low fringe frequency with low amplitude stripes, is denoted by S3-F1-S. Sample images are shown in the first column of Figure 4.5. We generated 100 samples with 1000×1000 image resolution under each configuration. Half of them were used for training and the rest were for testing. In this experiment, in order to assess the learning capacity and generalization ability of our proposed DeepInSAR model, a single model was trained on all 18 datasets with the noise-free ground truth images (teacher). Because all amplitude stripes and motion signals are randomly generated, all images between training and testing datasets were distinct. Figure 4.5 shows randomly selected samples from our simulation dataset. Our data generator is inspired by the noise simulation strategy described in Reference [99]. Basically, we simulate speckle noise by adding uncorrelated zero-mean Gaussian random variables to the real and imaginary parts of both synthetic SLCs before multiplying them for interferogram generation. To get the ground truth coherence for the simulated interferogram, we make an empirical mapping to it from the standard deviation of those random variables and the ground truth amplitude. This is because increasing the noise will decrease the coherence, and decreasing the amplitude will also decrease the coherence. In this case, each pixel in the generated interferogram is composed of 4 zero-mean Gaussian random variables with identical standard deviation. The source code of our simulator and full resolution simulated samples used in the experiments are available online at <https://github.com/Luckylyric/InSAR-Simulator>.

Visual comparisons with BoxCar, NL-InSAR, NL-SAR, and our proposed DeepInSAR methods are presented in Figure 4.6. Each two rows show the phase filtering

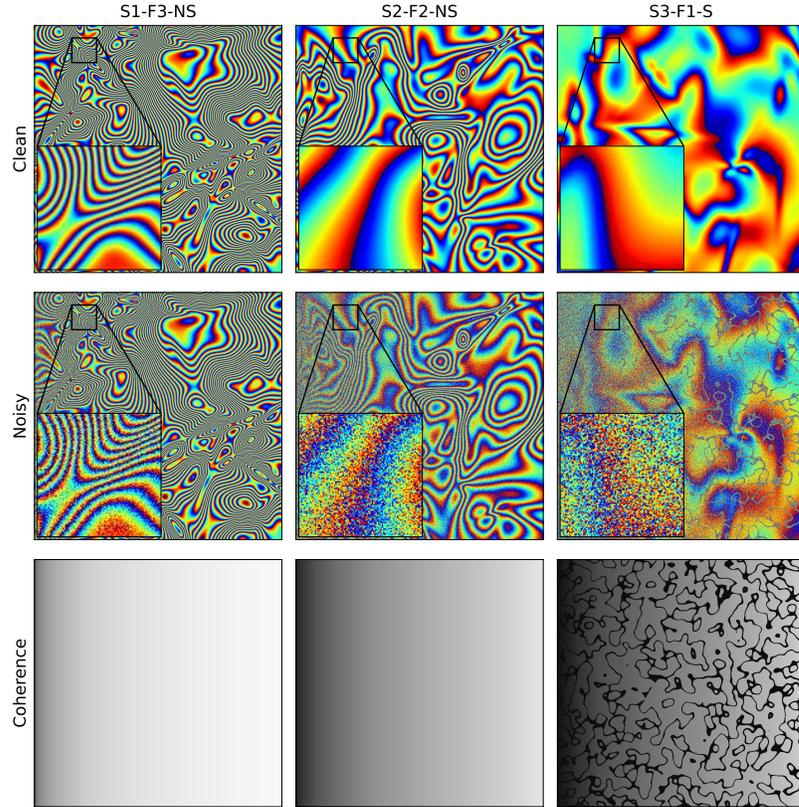


Figure 4.5: We use $S\#-F\#-S$ or $S\#-F\#-NS$ to name simulation datasets generated using different distortion scenarios: $S\#$ denotes Gaussian level of base noise S ; $F\#$ denotes frequency level of phase fringes F ; S and NS mean with or without low amplitude strips respectively. (From left to right) A set of simulated images are selected from $S1-F3-NS$, $S2-F2-NS$, and $S3-F1-S$ datasets. First row shows simulated ground truth with clean interferometric phase $[-\pi, \pi)$, second row is the noisy interferometric phase $[-\pi, \pi)$ —(Blue: $-\pi$; Red: $+\pi$), and third row is coherence (Black: 0; White: 1).

and coherence estimation of the three images in Figure 4.5 respectively, where (a–d) are filtering outputs and (e–h) are coherence estimations of S1-F3-NS, (i–l) are filtering outputs and (m–p) are coherence estimations of S2-F2-NS, and (q–t) are filtering outputs and (u–x) are coherence estimations of S3-F1-NS. Visual inspection on the filtered outputs compared to ground truth clean phase images in Figure 4.5 shows that our model can preserve phase structural details better than other methods for increasing base noise levels (Figure 4.6q–t) and frequency of fringes (Figure 4.6a–d). As we can observe, all methods work fairly well on low-level noise (S1) and low-level fringe frequency (F1) cases. However, with increasing distortion level, all other methods perform rather poorly. The BoxCar filter loses resolution and produces noticeably squiggly artifacts (Figure 4.6j,r). In particular, when distortion is with high base noise (S3) and high fringe frequency (F3), our model only loses insignificant detail especially in relatively low coherent regions on the left (Figure 4.6a,q). Although NL-InSAR can guarantee strong noise suppression with detail preservation on high frequency fringes (Figure 4.6c), it over-smooths the image when phase distortion level keeps increasing (2nd row of Figure 4.5); fringe structures are washed out when both distortion level and fringe frequency are high (Figure 4.6k). For coherence estimation, our proposed DeepInSAR is most matched to ground truth (Coherence row in Figure 4.5). BoxCar and NL-SAR tend to output low coherence on fast moving areas (Figure 4.6f,h). NL-InSAR and NL-SAR fail to compute correct coherence around low amplitude strips (Figure 4.6w,x). NL-InSAR also shows inaccurate coherence estimation between the phase jumps (Figure 4.6h,p).

We also use objective assessment to evaluate the performance of our method. Our test datasets include $18 \times 50 = 900$ simulated images with noisy and ground truth phase images, as well as corresponding coherence indices. The results obtained from BoxCar, NL-InSAR, NL-SAR and our proposed DeepInSAR are compared. We computed both Root Mean Square Error (RMSE) in radians (Table 4.1), and mean Structural Similarity Map (SSIM) between the filtered phase image and noise-free ground truth to quantitatively evaluate the filtering performance (Table 4.2). RMSE and mean SSIM are also used to assess coherence estimation (Tables 4.3 and 4.4). Numerical results further confirm our observations that the proposed DeepInSAR significantly outperforms all other methods on most of the 18 different distortion levels. From the simplest (S1-F1-NS) to the most challenging (S3-F3-S) simulation task, all methods have decreasing the performance on both phase filtering and coherence estimation. However, the proposed DeepInSAR has the least performance degradation and consistently gives better results than the other methods with a total mean of RMSE (radians) 0.8536 and mean SSIM score 0.8666 for phase filtering. The statis-

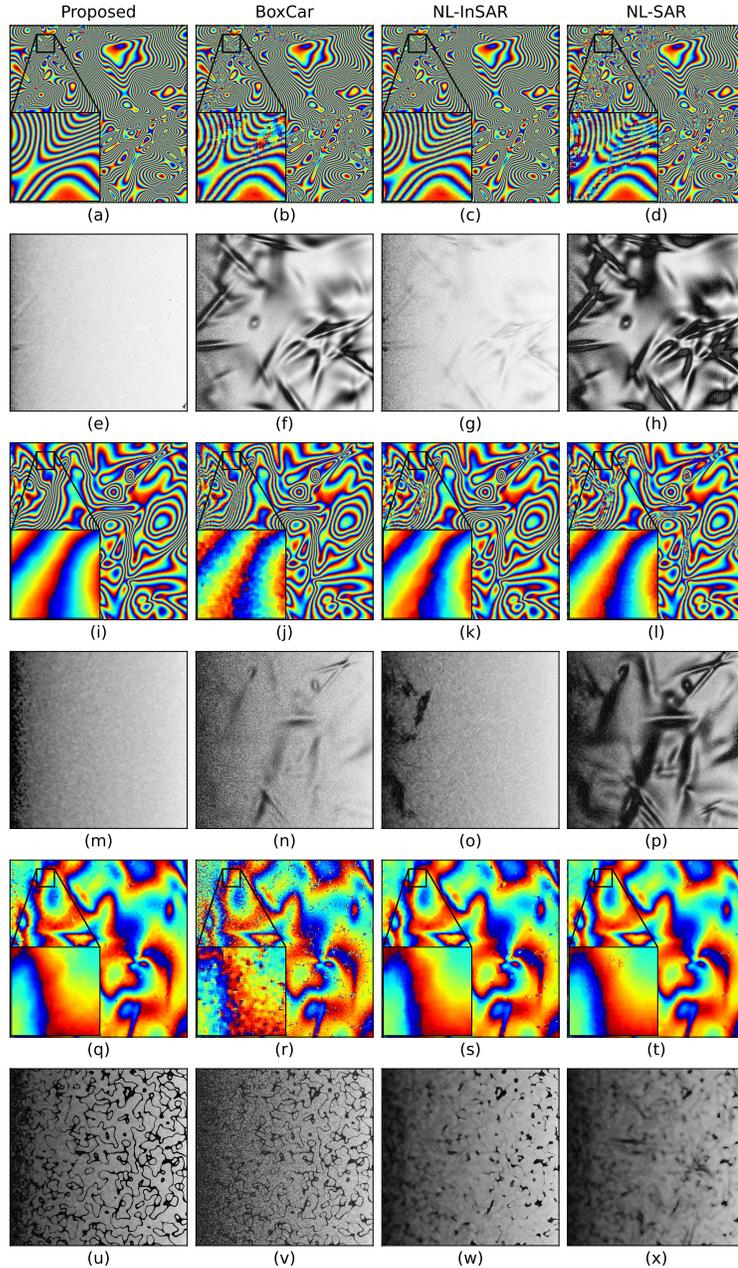


Figure 4.6: Examples of filtering and coherence estimation results on sample simulation images shown in Figure 4.5. (a–d) are filtering outputs and (e–h) are coherence estimations of S1-F3-NS, (i–l) are filtering outputs and (m–p) are coherence estimations of S2-F2-NS, and (q–t) are filtering outputs and (u–x) are coherence estimations of S3-F1-NS. Visual inspection on filtered outputs from different methods compared to ground truth phase images are given in Figure 4.5, 1st row. It can be seen that our model can preserve structural details better than others for increasing base noise levels and frequency of fringes (5th row). Our proposed method’s coherence estimation is most matched to ground truth (Figure 4.5, 3rd row), while other methods tend to predict inaccurate results on areas with highly dense fringes or low amplitude stripes.

tical analysis proves that our proposed model can effectively remove the noise and at the same time maintain the structural information effectively. The accuracy of our coherence measurement also shows superior performance with a total mean of phase RMSE 0.2167 and mean SSIM score 0.7984. Coherence computations in all other methods are biased as the data complexity increases, especially when they deal with dense phase fringes (F3) and low amplitudes strips (S).

Results on Real Data

Real complex features and noise patterns cannot be fully replicated by simulation data. However, we can conclude from simulation data experiments that if we can give the model close to clean reference data for teaching DeepInSAR, the model can learn latent mapping from training samples. As mentioned in Section 2.2.5, we use PtSel with expert supervision to generate clean reference phases and coherence maps for three real-world datasets captured by TerraSAR-X in StripMap mode [91]: (1) Site-A—27 SLCs, (2) Site-B—37 SLCs, and (3) Site-C—103 SLCs. We used a cropped version of these datasets with size 1000×1000 pixels. For coherence estimation, because the window-based PtSel coherence estimator is biased [96], we applied binary threshold 0.5 on PtSel’s coherence output to transform the original regression problem into a classification task. During the inference step, we use coherence estimator’s sigmoid output as the confidence level to represent final coherence magnitude. To demonstrate the generalization ability of the proposed DeepInSAR on real word InSAR data, we trained the model using images from two sites and tested its robustness on the third site. Three representative interferograms selected from each of the three real datasets are shown in Figure 4.7.

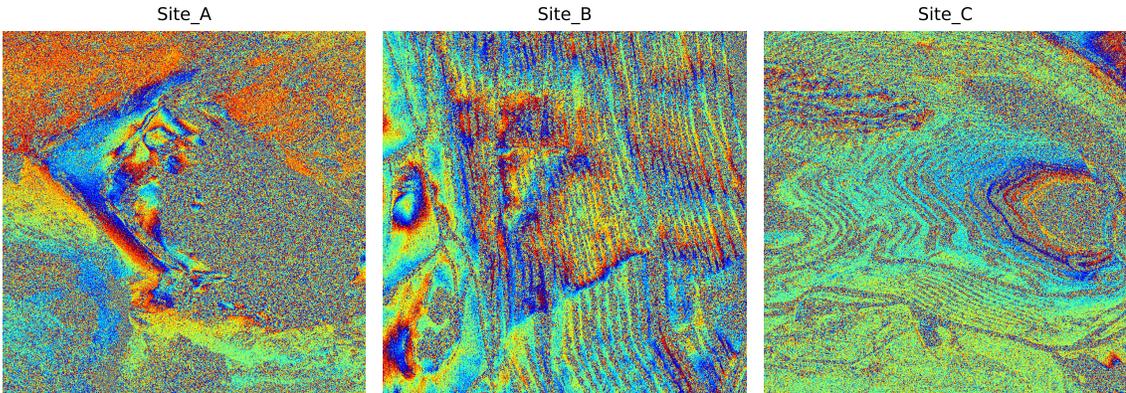


Figure 4.7: Three representative noisy interferograms (Phase) selected from each of the three real datasets; Blue: $-\pi$; Red: $+\pi$.

Table 4.1: Phase Root Mean Square Error (RMSE) (radians) on 18 different types of Simulation dataset. S denotes Gaussian level of base noise and F represents phase fringes frequency. S and NS mean with and without low amplitude strips respectively. Values with bold fonts indicate the best performance.

Phase RMSE (Radians)						
Sim Configuration		Methods				
		BoxCar	NL-SAR	NL-InSAR	Proposed	
S1	S	F1	0.7469	0.8401	0.8373	0.6939
		F2	1.0697	1.2012	0.9572	0.7422
		F3	1.0699	1.2054	1.0354	0.7890
	NS	F1	0.6675	0.7751	0.7088	0.6570
		F2	0.9906	1.1015	0.8284	0.6938
		F3	0.9623	1.1348	0.9138	0.7261
S2	S	F1	0.8409	0.8782	0.9105	0.8091
		F2	1.1252	1.2319	1.0859	0.8854
		F3	1.2096	1.2801	1.1890	0.9593
	NS	F1	0.7863	0.8199	0.8256	0.7715
		F2	1.0567	1.1687	0.9854	0.8297
		F3	1.1251	1.2186	1.0855	0.8785
S3	S	F1	0.9542	0.9332	0.9648	0.9370
		F2	1.1920	1.2657	1.1883	1.0239
		F3	1.3080	1.3430	1.2940	1.1156
	NS	F1	0.8886	0.8672	0.8976	0.8709
		F2	1.1307	1.2203	1.1159	0.9555
		F3	1.2398	1.2927	1.2120	1.0259
Average		1.0202	1.0988	1.0020	0.8536	

Filtered phases and estimated coherence obtained using BoxCar, NL-InSAR, NL-SAR, PtSel, and our trained DeepInSAR are shown in Figures 4.8–4.10, which are the outputs of three real sites given in Figure 4.7. We use qualitative comparison because we do not have noise-free real images for quantity evaluation. The Box-Car filter tends to blur fringe edges in all the visual samples, mainly because of its

Table 4.2: Phase mean Structural Similarity Map (SSIM) on 18 different types of Simulation dataset. S denotes Gaussian level of base noise and F represents phase fringes frequency. S and NS mean with and without low amplitude strips respectively. Values with bold fonts indicate the best performance.

Phase SSIM						
Sim Configuration		Methods				
		BoxCar	NL-SAR	NL-InSAR	Proposed	
S1	S	F1	0.9424	0.8897	0.8566	0.9511
		F2	0.7372	0.6266	0.7723	0.9333
		F3	0.6937	0.5989	0.6888	0.9015
	NS	F1	0.9665	0.8923	0.9505	0.9585
		F2	0.8075	0.7413	0.8887	0.9493
		F3	0.7999	0.7074	0.8117	0.9303
S2	S	F1	0.8898	0.8590	0.8358	0.9122
		F2	0.6624	0.5681	0.6746	0.8647
		F3	0.5150	0.4684	0.5202	0.7976
	NS	F1	0.9221	0.8902	0.9023	0.9312
		F2	0.7357	0.6577	0.7825	0.8966
		F3	0.6152	0.5647	0.6398	0.8527
S3	S	F1	0.8026	0.8168	0.7939	0.8349
		F2	0.5717	0.4989	0.5748	0.7670
		F3	0.3747	0.3555	0.3919	0.6675
	NS	F1	0.8570	0.8722	0.8508	0.8824
		F2	0.6463	0.5736	0.6621	0.8211
		F3	0.4612	0.4375	0.4938	0.7463
Average		0.7223	0.6677	0.7273	0.8666	

low-pass behaviour and it under-filters near incoherent areas, which can be easily observed when zooming in. In Figure 4.8, there appears minor loss of resolution in thin strips (when zoomed in) for the proposed compared to PtSel (stack-based) but is still much better than all other methods that use a single interferogram. NL-InSAR has more stripping artifacts that cause streaks in the phase along incoherence boundaries,

Table 4.3: Coherence RMSE on 18 different types of Simulation dataset. S denotes Gaussian level of base noise and F represents phase fringes frequency. S and NS mean with and without low amplitude strips respectively. Values with bold fonts indicate the best performance. Values with bold fonts indicate the best performance.

		Coherence RMSE				
Sim Configuration	Methods					
	BoxCar	NL-SAR	NL-InSAR	Proposed		
S1	S	F1	0.4360	0.4532	0.3827	0.2125
		F2	0.5418	0.6356	0.3526	0.1838
		F3	0.5321	0.6251	0.3639	0.1850
	NS	F1	0.2119	0.3472	0.1436	0.2045
		F2	0.5458	0.6515	0.1907	0.1633
		F3	0.5444	0.6494	0.2565	0.1564
S2	S	F1	0.4284	0.4522	0.4136	0.2688
		F2	0.4887	0.5564	0.3802	0.2699
		F3	0.4784	0.5463	0.3869	0.2774
	NS	F1	0.2052	0.3303	0.1878	0.2011
		F2	0.4768	0.5664	0.2749	0.2038
		F3	0.4766	0.5600	0.3175	0.2166
S3	S	F1	0.3780	0.3988	0.3834	0.2549
		F2	0.4251	0.4836	0.3726	0.2553
		F3	0.4244	0.4678	0.3805	0.2591
	NS	F1	0.2052	0.2522	0.2086	0.1920
		F2	0.4117	0.4904	0.3116	0.1955
		F3	0.4207	0.4817	0.3419	0.1998
Average		0.4240	0.4971	0.3139	0.2167	

which also shown up in its coherence output (Figure 4.9). It also results in artifacts that follow the benches rather than the fringe lines (Figure 4.10). NL-SAR can significantly remove the noise, but it also yields over-filtering that breaks some fringes and also merges small scale signals with neighboring fringes (Figure 4.8). Overall, though non-local based NL-SAR and NL-InSAR can provide as sharp and visually

Table 4.4: Coherence mean SSIM on 18 different types of Simulation dataset. S denotes Gaussian level of base noise and F represents phase fringes frequency. S and NS mean with and without low amplitude strips respectively. Values with bold fonts indicate the best performance.

Coherence SSIM						
Sim Configuration		Methods				
		BoxCar	NL-SAR	NL-InSAR	Proposed	
S1	S	F1	0.5598	0.5444	0.7150	0.9056
		F2	0.4580	0.2859	0.5979	0.9007
		F3	0.3180	0.1280	0.4455	0.9104
	NS	F1	0.6695	0.7234	0.9497	0.9069
		F2	0.5134	0.4225	0.6767	0.9040
		F3	0.3524	0.2859	0.4318	0.8977
S2	S	F1	0.3621	0.5057	0.6257	0.7349
		F2	0.3100	0.2860	0.4596	0.7649
		F3	0.2340	0.1967	0.2930	0.7508
	NS	F1	0.3061	0.7688	0.8752	0.7864
		F2	0.2422	0.3986	0.4855	0.7584
		F3	0.1756	0.1931	0.1853	0.7994
S3	S	F1	0.2555	0.5082	0.5734	0.6908
		F2	0.2311	0.2952	0.4029	0.7323
		F3	0.1840	0.1728	0.2275	0.7072
	NS	F1	0.1782	0.8209	0.8195	0.7475
		F2	0.1524	0.3910	0.4391	0.7119
		F3	0.1246	0.1552	0.1416	0.7617
Average		0.3126	0.3935	0.5192	0.7984	

appealing filtered phase as DeepInSAR on high coherence areas, in medium and low coherence areas, they tend to flatten the phase and create artifacts in highly noisy areas (Figure 4.8). Both methods have lower overall variance and less blurring than the BoxCar filter, though NL-InSAR has high variance in the estimates between the coherence/amplitude boundaries with streaky artifacts. Our proposed DeepInSAR

Table 4.5: Running time T (in seconds) of different methods with image size 1000×1000 .

	BoxCar	NL-SAR	NL-InSAR	Proposed
T (s)	1.16	12.77	19.36	0.46

shows a good balance between noise removal and structural preservation. Regarding to coherence estimation, the proposed DeepInSAR consistently gives better contrast and less spurious high coherence points within the low coherence areas in all the visual samples. It would be easier to be used as the weighting mask for the subsequent InSAR processing e.t. phase unwrapping, compared to other methods. In NL-SAR’s and NL-InSAR’s coherence outputs, there are also artifacts showing high coherence dots in low coherent areas. The limitation is caused by NL-InSAR’s numerical instability algorithm and preferential treatment of amplitude, when the amplitude similarities disagree with the phase similarities. Explanation of NL-InSAR’s weakness is also discussed in Reference [100, 101]. Comparing to these non-stack based methods, our DeepInSAR offers both strong noise suppression and detail preservation as well as gives clear high contrast coherence estimation. It performs on par and even better than its stack-based teacher method—PtSel. PtSel’s coherence estimation is biased toward low coherence in the dynamic areas (Figures 4.8 and 4.10), because it requires the target remaining stable over a long period of time [96].

The High-level fringe frequency indicates fast-moving areas on the ground. These areas usually introduce many phase jumps ($-\pi$ to $+\pi$) in the wrapped interferogram. As aforementioned, structural information is one of the most important information that any phase filtering method should preserve. This is because the performance of subsequent InSAR processing, for example, phase unwrapping, is heavily affected by the distorted fringe structure. Many gradient-based phase-unwrapping methods reply on the phase gradients and derivatives, which are types of structural information [20]. As an effective InSAR phase filter, it should preserve the structural details as much as possible [26], and our proposed method demonstrates this capability. For such evaluation, SSIM is a better metric comparing to RMSE for assessing how much structure information has been preserved after filtering. The mean SSIM score (Table 4.2) indicates that our method preserves excellent details even on highly dense fringes (F3), where all reference methods show decreasing performance as the fringe density increases. Our model shows more noticeable improvement under the SSIM metric than the RMSE metric. It is because RMSE estimates absolute errors and the SSIM

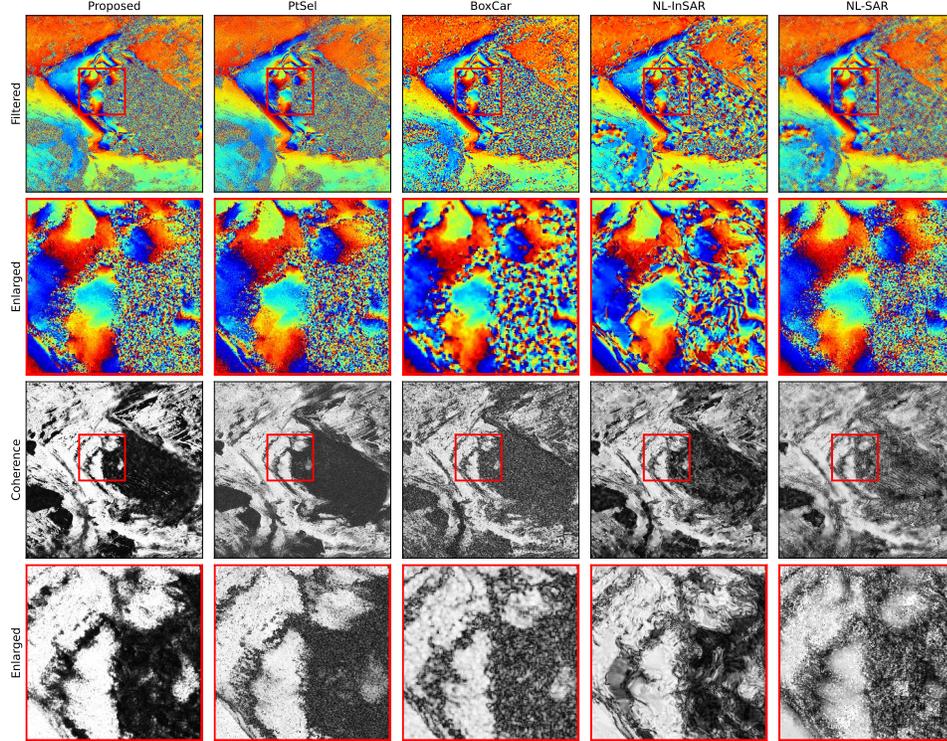


Figure 4.8: Filtered images and coherence maps generated by the reference methods and proposed DeepInSAR trained model for a Site-A image.

provides scores, which focus on the structural similarity. If a filter is over-filtering or breaking the boundary between phase jumps, it may only show insignificant RMSE changes, but will introduce a significant SSIM degradation. Furthermore, if a filter fails to fully suppress the noise signals, the residual noise in the output image will also be reflected more sensitively by SSIM score as in the natural image [102]. This is also the main reason why we use the SSIM metric in the comparisons. Note that, the structural information of coherence is not as important as the filtered phase, because the coherence values are mostly used as a threshold or weighting metric for subsequent processing. However, we still added SSIM metric for coherence estimation to enrich the experimental analysis. Table 4.4 shows that the proposed DeepInSAR can predict coherence map most matched to the ground truth. It demonstrates why our method can give high contrast and clear boundaries between extremely low and high coherence areas in both simulation and real site outputs. We believe that a method which can precisely recover the structural information in coherence map must also benefit the subsequent processing with a more detailed and precise coherence indication.

Moreover, in Figure 4.7, we used three very different real site interferogram examples. Similarly, all test simulation data were generated randomly. Both quantitative

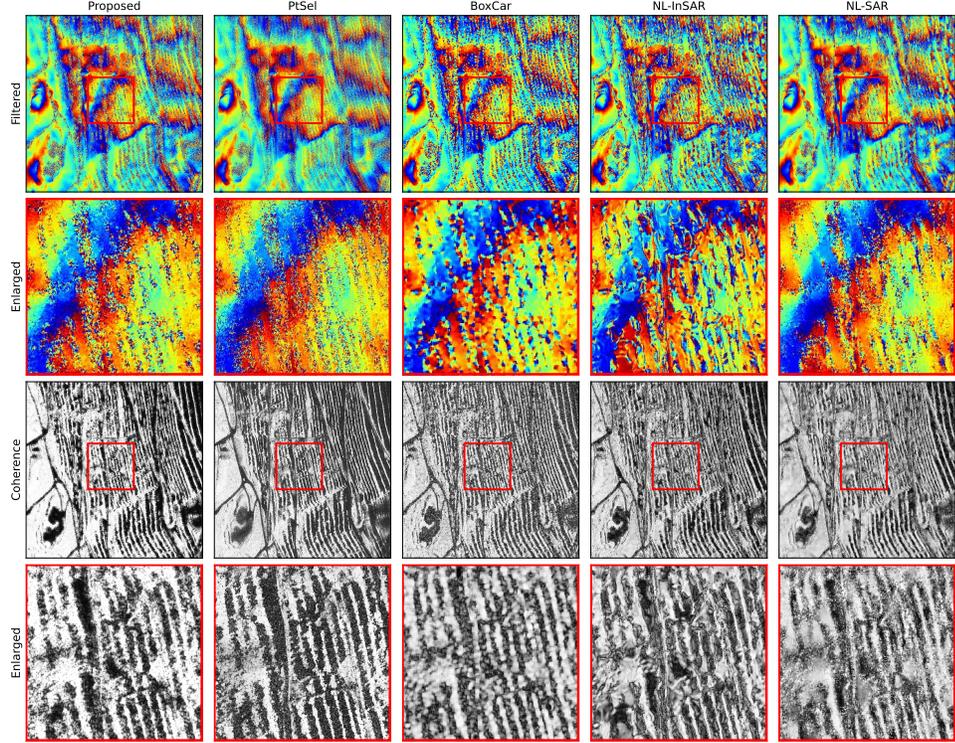


Figure 4.9: Filtered images and coherence maps generated by the reference methods and proposed DeepInSAR trained model for a Site-B image.

and qualitative results confirm that our trained DeepInSAR model generalizes well to new InSAR data without any human supervision or parameter adjustment, which is required by other methods. As an example, when we adjusted the searching window size to a smaller size, NL-SAR and NL-InSAR were able to filter well on those highly dense fringes, but facing under-filtering problem on slow motion areas. During the experiments, we had to manually tune the set of parameters for the reference methods in order to get reasonable results. Their coherence estimators also have similar limitations. In comparison, our proposed model’s coherence output is closest to the ground truth in all different distortion cases. For instance, all three referenced methods tend to give better results when using (1) a small window size on highly dense fringe areas but (2) need a large window size on low frequency motion. There is no fixed size, which works for all 18 simulated distortion levels. However, we show that our learning based DeepInSAR works well for all 18 simulated datasets with a single trained model. It has successfully learned the mapping from noisy observations (18 different distortions) to latent clean signals and coherence magnitudes, when we give it proper training samples to explore. Using densely connected feature extractor gives DeepInSAR the ability to intelligently handle multi-scale signal characteristics with

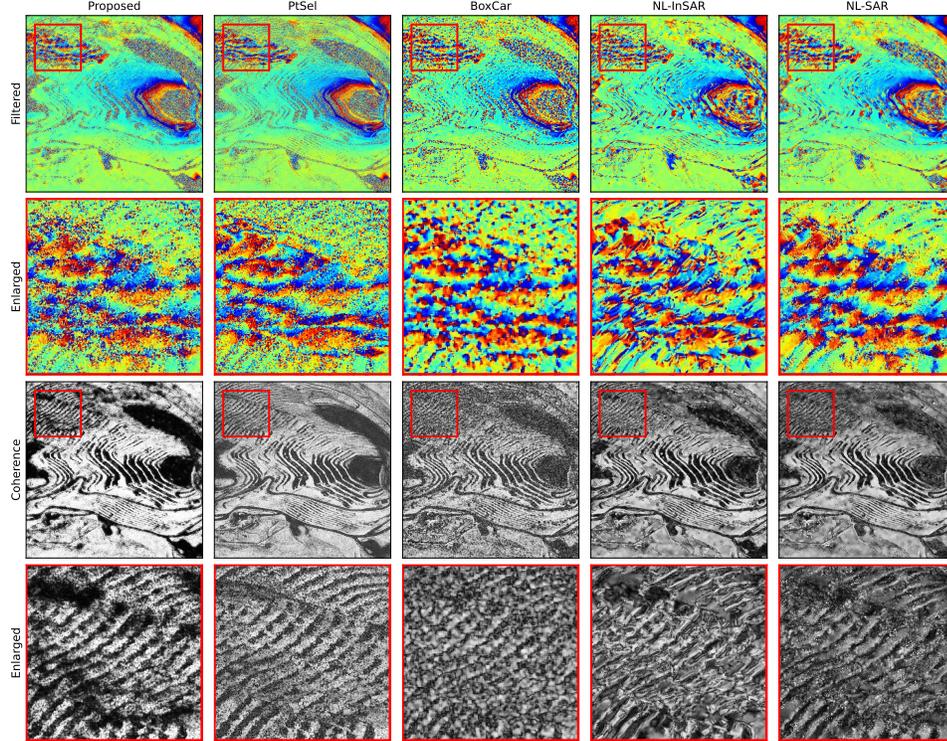


Figure 4.10: Filtered images and coherence maps generated by the reference methods and proposed DeepInSAR trained model for a Site-C image.

a single model. Since the simulated signal patterns are random, therefore simulated motion patterns, noise conditions and low reflective strips, are irregular among all training and testing images. The evaluation output from the test dataset shows that our trained model does not suffer from the over-fitting issue and only shows a small generalization error, which however has not affected its better performance. It learns well from the teacher and the model can be generalized to new InSAR data. From the operational point of view, NL-InSAR has large amount of artifacts that it produces in the phase and coherence. There are many instances where it does a good job, but in an industrial setting reliability is more important. NL-SAR is better in terms of reliability, but much worse in terms of resolution and is therefore also not an efficient option. The proposed DeepInSAR balances well on noise reduction and fringe preservation. At the same time, it gives a high level of bi-modality in the coherence estimates between the incoherence and coherent pixels.

Furthermore, besides the superior performance compared to other non-stack methods, under a teacher-student framework, DeepInSAR can achieve results comparable to or better than its teacher method with a learned discriminating neural network. The PtSel algorithm (teacher) has several limitations—(1) It relies on temporal in-

formation, which means that non-local linear motion can make it hard to pick a neighbourhood suitable for all interferograms, causing under-filtering in these areas. As a result, the algorithm has to wait for many more days of sufficient data before starting the process; (2) It has bias toward filtering results—PtSel looks for similar nearby pixels to perform filtering. If it does not find enough of such pixels, then the filtering is toward averaging, giving worse result compared to another pixel which can find lot of similar neighbors. PtSel’s filtering and coherence output is regarded as state-of-the-art in the literature, but it fails to give optimal output across the test input image because of its biased adaptive kernel estimation. On the other hand, the proposed DeepInSAR successfully distills the knowledge from training samples and generalizes the model to new unseen InSAR images with a simple feed-forward inference, without any human expert supervision, or intensive online searching on a stack of interferograms as required by PtSel. Our proposed DeepInSAR model captures coherence in the fast-moving areas even better than PtSel and produces excellent delineation in the coherence with better contrast, which helps subsequent stages in the InSAR processing pipeline, that is, when threshold and weighting are required on the estimated coherence in the phase unwrapping stage. With respect to the average running time (T) in seconds, as seen from Table 4.5, the proposed method requires significant less running time than other non-stack methods because only feed-forward computation is needed after training. After testing different parameter settings (e.g., number of iterations and patch size), reference methods sometimes get better results after running for a longer time. However, it is not always the case, which means that these methods have limited potential of full automation without human intervention. The proposed method shows better results with much faster processing time. It is worth mentioning that PtSel outputs used for training and visual comparison are generated using a Titan XP GPU farm. This is because PtSel requires high-end GPUs for intensive parallel searching on a stack of SLCs (>30). In comparison, our method can run on a consumer level system, and perform filtering and coherence estimation using only two SLCs. Taking filtering, coherence performance and flexibility into consideration, the proposed DeepInSAR is very competitive and suitable for real-world InSAR applications.

Lastly, it is worth mentioning that, in this work, our InSAR simulator is mainly designed for quantitative evaluation and analysis, because there is no ground truth data for real-world images. The proposed simulator can generate randomly composite irregular motion signals, ground reflective phenomena, as well as non-stationary noise conditions with different controlled configurations. It is an ideal scenario for objectively assess our proposed DeepInSAR’s learning capacity and ability of gener-

alization. However, as a data-driven technique, when we want to apply the proposed DeepInSAR framework on real-world InSAR data, we need to make sure the training data distribution is similar to real-world scenarios. Existing simulator is designed to give controlled experimental environments for quantitative analysis, but it is still not able to fully replicate the real-world complex features and noise patterns. That is also the reason why we propose the teacher-student framework, which has been validated to be useful for adapting the proposed DeepInSAR to a real-world phase filtering and coherence estimation pipeline. This is also one of the contributions we would like to highlight. We show the potential benefits in the InSAR industry that the proposed DeepInSAR framework has the ability to transform conventional methods, which might require higher computational resources, more input observations, and human supervision, into a differentiable deep neural network model by learning from their outputs. In future work, we plan to investigate a Generative Adversarial Network (GAN) [71] based InSAR simulator for generating more realistic synthetic data. We believe that it will certainly confirm the operationalization aspect of the proposed DeepInSAR.

4.2.4 Conclusions

In this work, we propose a learning-based DeepInSAR framework to address two important research issues: InSAR phase filtering and coherence estimation, in a single process. Our model works well when using either simulated or real data, under different synthetic distortion and real noisy pattern levels. To quantitatively assess the proposed method, we designed an InSAR simulator, which can generate motions and noise patterns randomly. The proposed DeepInSAR outperforms existing non-stack based methods for both tasks by giving the most matched filtered phase and coherence map comparing to the ground truth data. SSIM scores (0.8666 for phase filtering and 0.7986 for coherence estimation) also show superior DeepInSAR performance that can preserve well the phase fringe structure after filtering, and at the same time gives sharp and clear coherence map. Numerical results show that the proposed DeepInSAR can generalize well on new unseen images once it has been trained, and thus can be applied in various real-world InSAR applications. We also presented a teacher-student training strategy, which allows the proposed DeepInSAR to augment, automate and accelerate existing un-differentiable methods using a differentiable deep neural network. Our trained model can obtain the same or better filtering and coherence estimation results only on a single pair of SLC images compared to its teacher algorithm, which requires a stack of SLCs(>30), achieving significantly higher com-

putational efficiency. Comparing to other non-stack based methods, our model gives most robust results on both filtering and coherence estimation (1) without any human supervision and (2) with real-time performance. In addition, the proposed DeepInSAR gives a high level of bi-modality coherence estimation that nicely distinguishes the incoherence and coherent pixels, which benefits the subsequent phase unwrapping. To the best of our knowledge, the proposed DeepInSAR is the first work that uses deep neural network to perform InSAR filtering and coherence estimation jointly using both amplitude and phase information of only two co-registered SLC SAR images. In future work, we will investigate how well the proposed DeepInSAR framework can benefit subsequent InSAR analytic stages along the processing pipeline.

4.3 Self-supervised Residual Distribution Learning for SAR Interferometric Phase Restoration and Coherence Estimation - SRDInSAR

4.3.1 Introduction

To the best of our knowledge, DeepInSAR [103] is the first successful deep learning framework for jointly performing phase filtering and coherence estimation on SAR interferometric images. Afterward, numerous learning-based models (e.g, [104] [105] and [106]) have referenced the DeepInSAR and continue to explore this topic by incorporating state-of-the-art model architecture or integrating pre- and post-processing procedures to enhance results. We believe that DeepInSAR demonstrates the advancements associated with using a differentiable neural network to mimic both expert human decision-making behaviour and the performance of a sophisticated indifferentiable algorithm. Apart from the advancement in the application of a deep learning model to the problem of InSAR signal restoration and estimation of coherence. All prior studies, including DeepInSAR, have relied on supervised learning. A downside of the supervised setting is the restricted accessibility of the training dataset. Possibly, the quality of ground truth training labels poses a performance barrier. Additionally, developing a large-scale training dataset is a time-consuming and costly process. No known InSAR simulator model can fully replicate clean and noisy real-world signals for training and then apply a model to real-world data. Indeed, the benefits of learning approaches have not been fully explored because most of the present work is focused only on maximizing a mapping function between input and output signals. Rather than using corrupted images as inputs and clean images as targets in typical supervised learning, researchers [107] invented Noise2Noise training,

which employs pairs of corrupted images as training data and does not require clean references. However, each training image requires at least two distinct realizations of the corruption, which is still unfeasible in real-world cases, particularly ones involving InSAR data. Afterward, the Noise2Void [108] training method improved the self-supervised denoising algorithm by eliminating the requirement for image pairs and focusing the training only on individual noisy images, assuming that the corruption is zero-mean and independent between pixels. The researchers defined a blind-spot technique in which the receptive field of the network excludes the central pixel. As a result, the model is forced to recover the noisy version of the center pixel value using only the surrounding context and not a separate reference image. Noise2Void is the first truly self-supervised work that trains directly on the body of data to be denoised. It can be used in situations where other approaches are impractical. Its most notable use is in the InSAR field, where it is typically not easy to acquire training samples, whether clean or noisy pairs.

We take inspiration from the recent Noise2Void training but adapt it for phase filtering in InSAR. Due to the extra problem of estimating coherence, our research group published GenInSAR [109], a generative model that predicts a value distribution rather than the target value directly using mixture density networks (MDNs) [110]. GenInSAR introduces the process of learning the InSAR phase value distribution and can be used to approximate the parameters - mean and standard deviation - of a bivariate (Real, Imaginary) Gaussian distribution of each pixel location. GenInSAR is built as a proof of concept for using self-supervised training to solve InSAR filtering and coherence estimation challenges. However, some issues related to Noise2Void and GenInSAR remain unsolved: 1) The model is trained on patches with center pixel masking, resulting in a limited model architecture that constrains the output dimension to 1x1. Additionally, only a few output pixels can contribute to the loss function, restricting training efficiency dramatically. 2) GenInSAR implements the Noise2Void concept by masking the center pixels to prevent the model from learning similar outputs; however, this approach results in a trained model that ignores the center pixel information, which has been addressed in the original Noise2Void study [108]. This is especially detrimental in InSAR tasks, as each pixel can represent hundreds of square meters of the ground area regarding the data’s resolution; thus, each pixel’s information should be considered in conjunction with its surrounding context. 3) The final notable drawback of GenInSAR is its unsatisfactory estimation of coherence, which our previous study recognized as a high priority for future development. GenInSAR’s coherence formula is based on a handcrafted derivation that makes numerous assumptions and is not rigorously constrained inside

the range $[0, 1]$, theoretically violating the coherence definition.

4.3.2 Method and Procedure

This thesis proposes Self-supervised Residual Distribution (SRD) InSAR filtering and coherence estimation to address each of GenInSAR’s limitations. A high-level architecture of the proposed SRDInSAR framework is illustrated in Figure 4.11. Our improvements to GenInSAR can be summarized as follows: 1) We propose a random masking mechanism rather than using the center pixel masking approach adopted by GenInSAR and Noise2Void. This approach allows a greater number of pixels to contribute to the loss function and ensures that the training is compatible with any modern fully convolutional image to image architecture (e.g., UNet [111], DenseNet [112], ResNet [12]) to offer effective and adaptable training. 2) We propose training the model to estimate the residual distributions of both real and imaginary channels, drawing inspiration from residual learning in natural image restoration and distribution learning in GenInSAR. This prevents the model from ignoring the target pixel value throughout the training and inference stages. 3) Following GenInSAR’s approach of estimating the distribution of each pixel location using the MDN layer, we take it a step further by directly approximating pixel coherence using Monte Carlo sampling from the estimated pixel distribution [113]. Compared to GenInSAR’s handcrafted formula, our approach strictly follows the theoretical model of InSAR coherence and produces flawless $[0, 1]$ value outputs. The next sections detail each of the advancements.

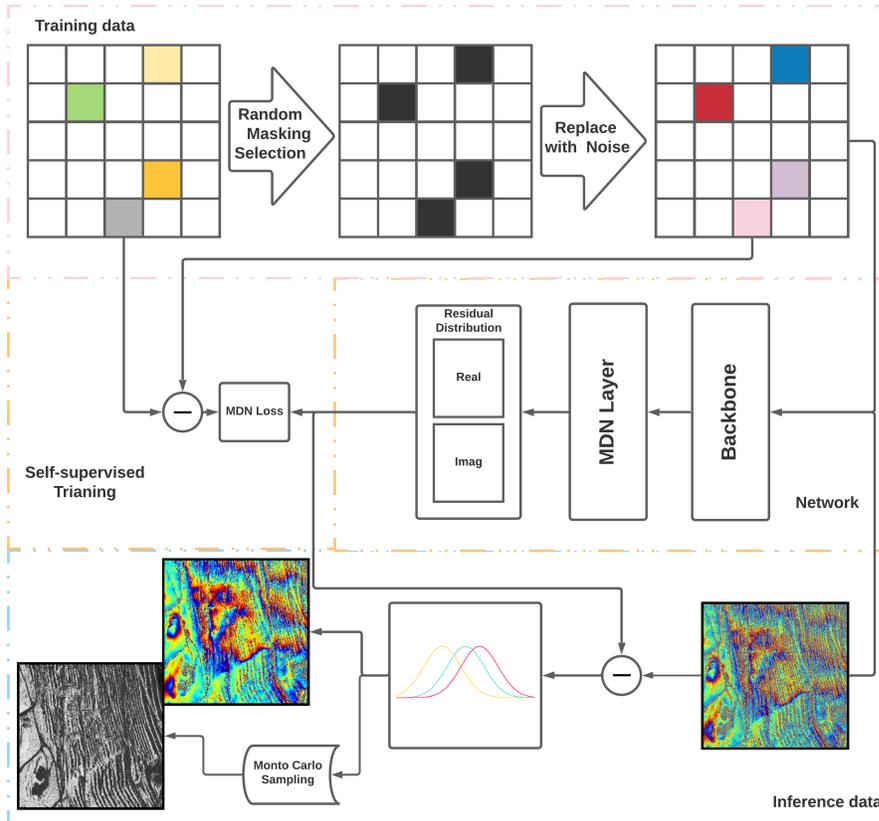


Figure 4.11: Illustration of proposed SRDInSAR framework.

Random Masking

GenInSAR begins by transforming the noisy interferometric phase θ_{noisy} into r_{noisy} and i_{noisy} real and imaginary channels as introduced in Section 4.1. Masking the center pixel value in all training patches, adheres to Noise2Void’s fundamental notion of blind-spot training. GenInSAR can then be trained to estimate the central pixel’s value distribution using the MDN output layer. The model’s architecture must be perfectly output in 1 x 1 dimensions to enable parallelism inference after training. This means that GenInSAR can only use restricted network architectures with a certain model depth and a certain number of down-sample steps depending on the size of the training patch. Additionally, regardless of the size of the input patch, this architecture results in only a few pixels contributing to the loss function, as it is unable to apply a fully convolutional neural network (CNN) for effective image-to-image training. Modern image-related deep learning research exclusively focuses on image-to-image correspondences, creating an incompatibility with GenInSAR’s ability to benefit from recent advancements in network architecture research. In

the proposed SRDInSAR, we perform random masking for each training patch, as illustrated in Figure 4.11. As a result, each pixel location has a probability of ω being selected as a masked pixel as a preprocessing step in the training stage. Then, just these pixel regions are considered while calculating loss. Compared to the GenInSAR, this approach performs more stochastically as a data argument since it masks off more than just the center pixel information. Moreover, because there are no hard constraints on the output dimensions during training, this configuration easily fits any image-to-image architecture. As seen in the high-level design, we can select the backbone module for feature extraction only prior to the MDN layer, which provides better adaptability and flexibility than GenInSAR. On the other hand, additional pixels contribute to the loss function during the training stage to further improve training efficiency.

Residual Distribution

The random masking method can still not resolve the issue of the model disregarding target pixel information during training or inference due to the fact that a 0 value is used as the input of masked pixels. Thus, rather than predicting the clean signal component, we propose using residual learning to estimate the residual components directly, as demonstrated in many natural image restoration works. Residual learning was also employed in our previous works, ADDNet [114] and DeepInSAR [103], and shows exceptional performance in terms of restoration quality and training efficiency. This work develops a residual distribution learning algorithm by utilizing MDN. In our GenInSAR work, we employ the MDN layer to produce the parameters (μ, σ) of a bi-variate (real and imaginary) Gaussian distribution of the center masked pixel. Following training, the estimated means of distribution are used to recover the filtered pixel phase value as $\theta_{filtered} = \arctan(\frac{\mu_{Imag}}{\mu_{Real}})$. To incorporate residual learning, we first substituted random phase values (represented as two channels- ε_{Real}^n and ε_{Imag}^n) for each selected masked location, whose original noisy observations are $(\alpha_{Real}, \alpha_{Imag})$. During the training, we force the model to parameterize a distribution $(\mu^\omega, \sigma^\omega)$ by maximizing the Gaussian density that best encompasses the residual value $\omega = \varepsilon - \alpha$ for both channels in all training samples. To this end, our filtered phase during inference can be derived as $\theta_{filtered} = \arctan(\frac{\alpha_{Imag} - \mu_{Imag}^\omega}{\alpha_{Real} - \mu_{Real}^\omega})$. This strategy forces the model to consider the input value of all selected pixel locations. This is particularly true for wrapped interferograms, because 0 is a valid signal value in both real and imaginary channels. It is illogical to hardcode 0 values for all masked center pixels in InSAR data, and we can avoid this in SRDInSAR versus GenInSAR by applying

a residual learning strategy.

Monte Carlo Coherence Inference

Coherence is defined as an ensemble average derived for each pixel using a collection of observations. This requires the simultaneous acquisition of a large number of interferograms under identical conditions, which is impossible in the real world. The most frequently used technique for assessing coherence is to employ a small window to form a maximum likelihood estimator (MLE) [115]. It is predicated on the premise that observations within a small region stay stationary, which is not true in most real-world scenarios. Furthermore, a well-known limitation of MLE algorithms is the loss of resolution when large window sizes are used, and the estimation bias when small window sizes are used. It is an inevitable trade-off when exchanging *ensemble averages* with *spatial averages*. Later, our industrial partner 3vGeomatics developed the temporal analysis technique PtSel [96], which chooses temporal persistent scatter as stationary observations through the stack in order to achieve high-resolution estimations. We employ residual distribution learning to estimate the phase value distribution of each pixel location, which is modeled by the output of MDN layers. To mimic the behaviour of repeated captures of the same location under the same situation, we chose the Monte Carlo strategy to draw a number of phase samples from the anticipated distribution. The coherence index can then be determined as follows by utilizing the sampled interferometric phase:

$$\gamma = \frac{1}{N} \left| \sum_{k=1}^N \exp^{j(\theta_k)} \right|$$

where,

$$\theta_k = \arctan\left(\frac{\alpha_{Imag} - \omega_{Imag}^k}{\alpha_{Real} - \omega_{Real}^k}\right) \tag{4.12}$$

$$\omega_{Real}^k \sim N(\mu_{Real}, \sigma_{Real})$$

$$\omega_{Imag}^k \sim N(\mu_{Imag}, \sigma_{Imag})$$

Although N signals must be sampled for each location, it is worth noting that the calculation can be executed entirely on GPUs without considerably increasing processing time. GenInSAR produces coherence through the use of a handcrafted formula based on assumptions. Furthermore, its output value is not mathematically restricted between 0 and 1. In contrast, Monte Carlo inference extensively uses the estimated distribution to approximate ensemble averaging, as defined by the InSAR definition of coherence.

4.3.3 Experimental Setup

GenInSAR’s experimental setup [109] is replicated here. Likewise, we compare our strategy to a range of different non-stack-based approaches, including the classic algorithms: BoxCar, Goldstein [116], NLIInSAR [35], and NLSAR [39]. These approaches provide open-source OpenCL 1.2 implementations, as specified in previous research [103]. Additionally, we compare our SRDIInSAR to two unsupervised works (CNN-InSAR [117] and GenInSAR) to ensure a direct and fair comparison. Our SRDIInSAR is built on PyTorch-1.8.1 and CUDA 10.2. To maximize the randomness of the training patch samples, we train our SRDIInSAR on randomly extracted image patches having dimensions of 128x128, following the strategy described in [114][103]. As previously stated, our SRDIInSAR could use any image-to-image backbone network. For demonstration purposes, we adopt a classic UNet [111] architecture in our study. At each training step, each pixel in the training patch has a 20%-30% chance of being chosen as a masked location and replaced by random noise to form residual learning. Finally, it is worth mentioning that we construct and train our models using standard hyper-parameter settings in our experiments. All algorithms are trained and evaluated on the same machine, which is equipped with an NVIDIA 1080 GPU, an Intel i7-8700K processor, and 32 GB of RAM. We conduct experiments using both simulated and real-world data to determine the applicability and robustness of the proposed model.

4.3.4 Results

Using Simulated InSAR Data

We employed the same InSAR simulator as in our previous publications [109] [103]; this simulator can simulate ground truth interferograms with Gaussian bubbles, roads, and buildings. We introduced Gaussian noise to simulated noise-free images using input patches from the noisy versions to establish self-supervised training. To objectively compare SRDIInSAR to other recent methods, we split the dataset in the same way for all learning-based models. We assessed all approaches using the same testing dataset, which consisted of 60 noisy simulated images having a resolution of 1000x1000.

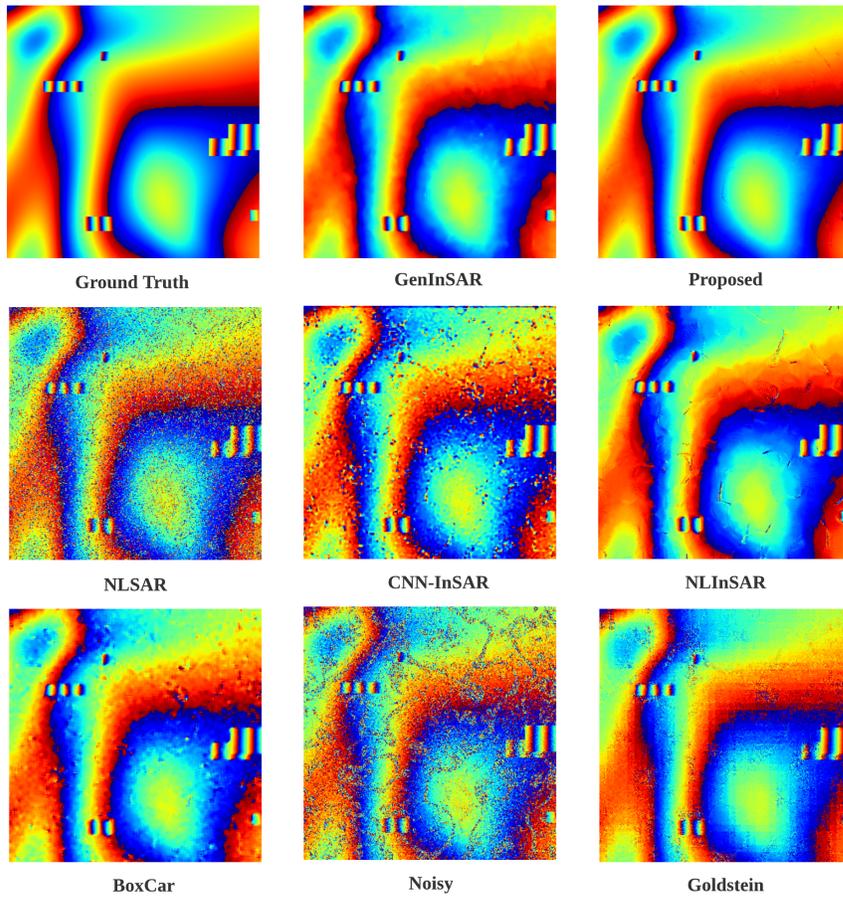


Figure 4.12: Filtered phase simulated InSAR images from SRDInSAR and five existing methods. Phase is colored between $-\pi$ (blue) to $+\pi$ (red)

Figure 4.12 depicts a visual examination of a cropped area from test images. According to the phase filtering results, SRDInSAR suppressed noisy signals well while avoiding the breaking of original fringes. SRDInSAR’s results are the most accurate, even for extremely low coherent areas, compared to all other approaches. Conventional techniques produce mostly artifacts and flush out phase fringes in excessively noisy regions. GenInSAR is able to reduce the noise; however, there was significant over-filtering around the sharp fringe edges. Compared to other methods, SRDInSAR’s filtering results in the best preservation of fine features and the least degree of over-smoothing near branch cuts.

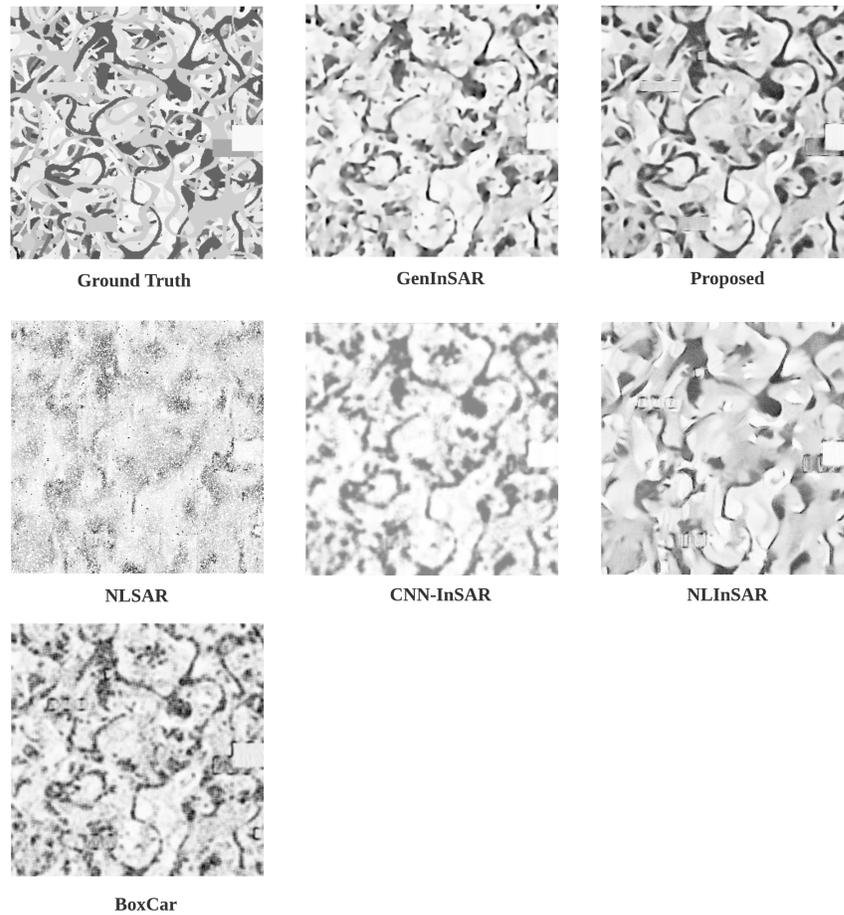


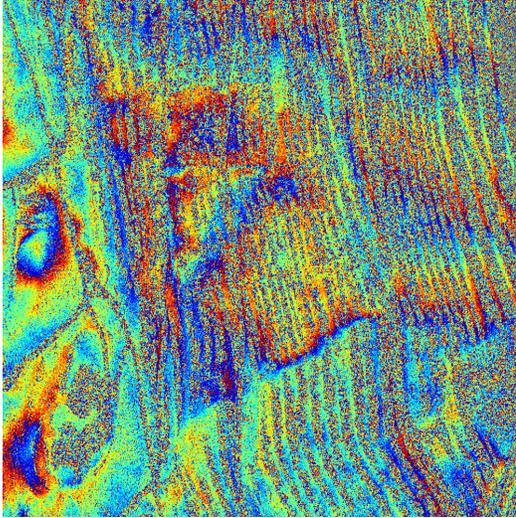
Figure 4.13: Coherence estimation of simulated InSAR images from SRDInSAR and five existing methods. Coherence is colored between 0 (black) to 1 (white)

As seen in Figure 4.13, our proposed SRDInSAR’s coherence estimation has the best match to ground truth. The other two unsupervised models (CNN-InSAR and GenInSAR) show considerable contrast and resolution loss. Moreover, BoxCar and NL-SAR exhibit low coherence in high-speed areas and perform poorly in strip noise zones. Additionally, NL-InSAR produces an imprecise estimation around the region of phase jumps.

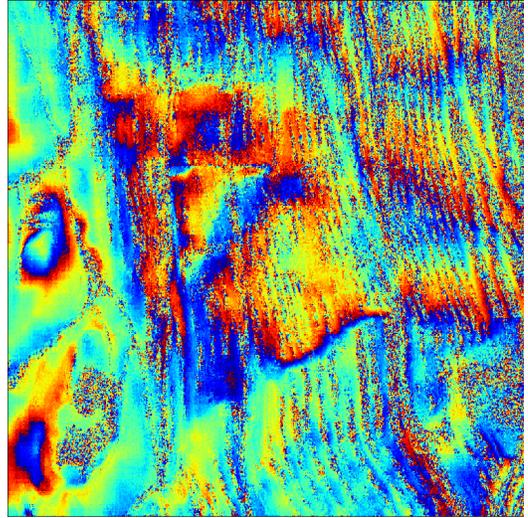
Table 4.6: Quantitative Evaluation of SRDInSAR and existing methods

Method Name	Phase RMSE	Coherence RMSE	Cosine Error
BoxCar	1.025	0.143	0.025
Goldstein	1.260	N/A	0.048
NLSAR	1.537	0.301	0.132
NLInSAR	0.850	0.159	0.014
CNN-InSAR	1.392	0.200	0.073
GenInSAR	0.687	0.138	0.005
Proposed (SRDInSAR)	0.557	0.025	0.003

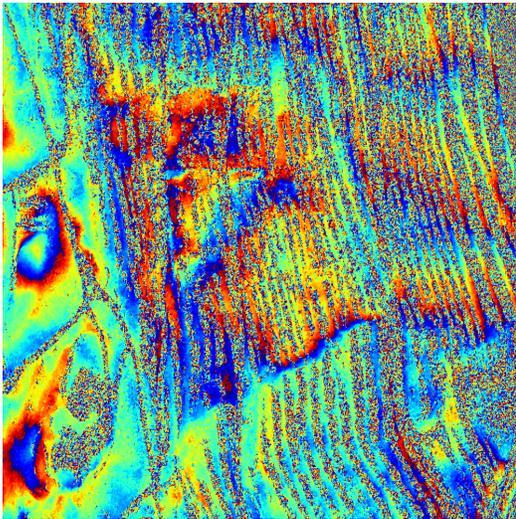
The quantitative evaluations are summarized in Table 4.6. The root mean square error (RMSE) of the InSAR phase and coherence, and the phase cosine error, are used as metrics. These are frequently used measurements in the literature for objectively assessing the performance of phase filtering and coherence estimation [109] [117]. In terms of filtering quality and coherence estimation, our SRDInSAR technique outperformed unsupervised methods GenInSAR and CNNInSAR. SRDInSAR also surpasses other established techniques, as evidenced by visual examination and numerical results.



(a) Noisy Phase



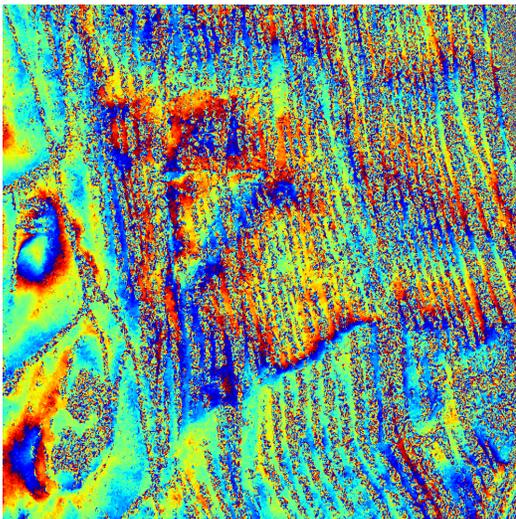
(b) Goldstein Phase



(c) Proposed (SRDInSAR) Phase



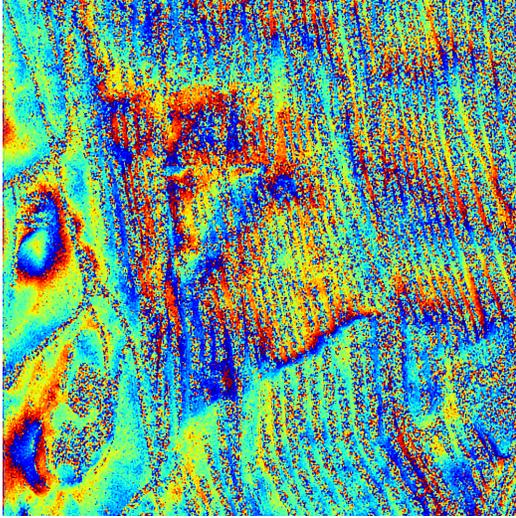
(d) Proposed (SRDInSAR) Coherence



(e) GenInSAR Phase



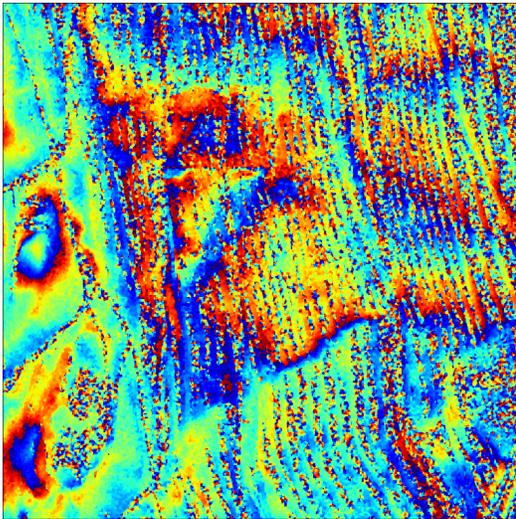
(f) GenInSAR Coherence



(g) CNN-InSAR Phase



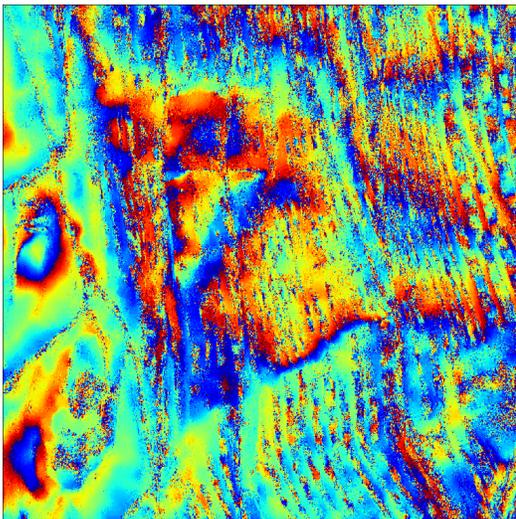
(h) CNN-InSAR Coherence



(i) Boxcar Phase



(j) Boxcar Coherence



(k) NLSAR Phase



(l) NLSAR Coherence

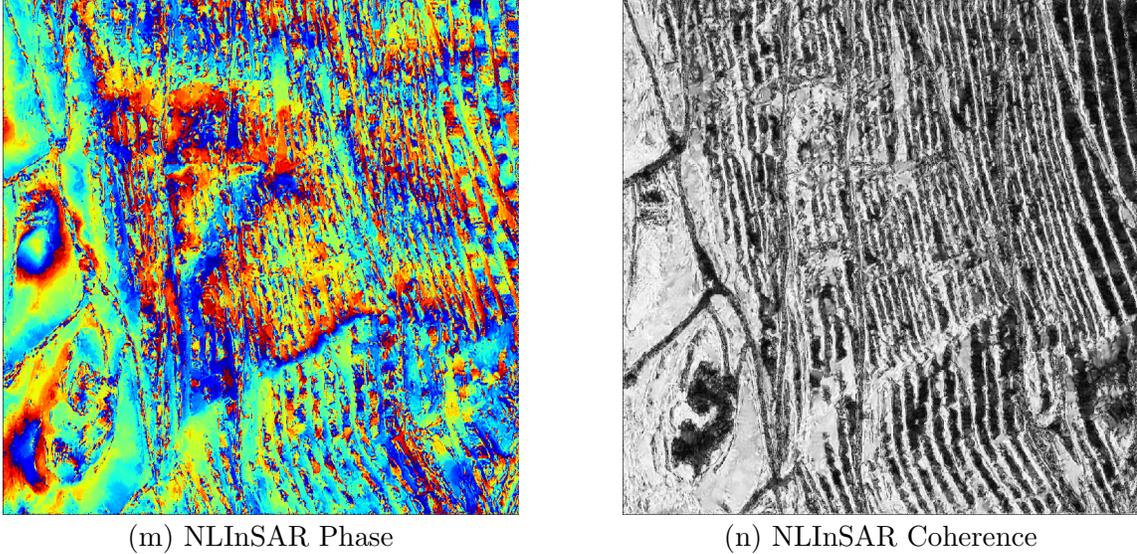


Figure 4.14: Filtered phase and coherence outputs for satellite InSAR images from SRDInSAR and six existing methods. Phase and coherence are colored from $-\pi$ (blue) to $+\pi$ (red), and 0 (black: low) to 1 (white: high), respectively

Using Real-World Satellite InSAR Data

We use 300 interferograms with varied resolutions to conduct real-world data experiments [109]. All trainable models were tested using interferograms of a mining site with a resolution of 1000x1000 pixels. A qualitative investigation of real-world data, as represented in Figure 4.14, confirmed our findings from simulated data experiments. For phase filtering, the proposed SRDInSAR outperformed the conventional methods in terms of noise reduction and resolution preservation. BoxCar’s low-pass behaviour caused fringe edges to be blurred in all visual samples. NL-InSAR and NL-SAR resulted in over-filtering and artifacts in places with significant noise. CNN-InSAR showed under-filtering and was comparable in coherence estimation to BoxCar, both suffered from a significant loss of resolution in their coherence estimate. While GenInSAR could produce a filtered phase similar to that of SRDInSR in areas of high coherence. GenInSAR’s estimation was less contrasted than that of SRDInSAR. Our method indicated an obvious distinction between coherent and incoherent areas.

4.3.5 Conclusions

This study presents the SRDInSAR as a self-supervised framework for InSAR phase filtering and coherence estimation. We employ random masking residual distribution learning and Monte Carlo sampling to improve the training efficiency, model flexibility, and inference performance of the baseline GenInSAR method. SRDInSAR is

evaluated quantitatively and qualitatively to demonstrate that it is a state-of-the-art unsupervised approach suitable for industrial deployment. Currently, we only examine the interferometric phase, but this will be expanded in the future to incorporate paired SAR images with amplitude information.

Chapter 5

Deformation and DEM Error Signals Separation in InSAR Time Series

5.1 Introduction

Over the years, there has been an increasing interest in interferometric synthetic aperture radar (InSAR) techniques. An InSAR interferogram represents the phase difference between two SAR images, taken at different temporal times looking at the same ground location on Earth. It has provided significant advances in measuring the Earth's surface deformation and creating precise digital elevation models (DEM). In early studies, most InSAR applications focused on analyzing a single interferogram derived from a pair of SAR images [51, 52]. Later, researchers noted that some radar targets' backscattering characteristics could maintain stability for a long period [51]. Hence, involving the analysis of multiple acquisitions in time could reduce the effects of temporal and geometric decorrelation and atmospheric disturbances. Since then, time-series InSAR (TSInSAR) techniques have emerged as powerful strategies to monitor slow and subtle terrain displacements [53]. Several studies [56, 59, 67, 118] have investigated the signal model of interferometric phase and have shown that observed interferometric phases are affected by different factors: imaging geometry, topography, atmospheric delay and ground deformation. Among these factors, the deformation and topography components are valuable contributors because they contain information for monitoring the ground movement and describing surface height. When an external DEM is adopted to remove the phase created at the earth's curvature step, DEM error should be estimated to revise the external DEM model and thus reduce the topography effects. However, deformation and DEM error are also known to be more challenging to be estimated than other types of signals [119]. In

general, TSInSAR techniques use SAR images acquired for the same ground area on different dates to construct a stack of N interferograms. The signal phase of single-referenced (i.e., master) images' resolution cell on the ground is a function of multiple-phase contributors; and the signals of interest can be estimated from the same resolution cell, taken at separate times [56]. However, each cell is represented as a wrapped phase, and the ambiguities regarding estimation with the phase cycle make this task challenging. There are inevitably heavy calculations, especially when processing wide-area regions [57]. Moreover, a range of processing methods requires manual inspection and specialist interpretation [58] to achieve quality control and could limit the timely dissemination of monitoring. Therefore, it is crucial to have an accurate, efficient and robust algorithm.

Estimating ground deformation and DEM error are usually defined as an ill-posed optimization problem by its very nature. One commonly cited difficulty is that temporal signals may be incoherent and impossible to derive any useful information due to the temporal decorrelation in a real-world scenario. One way of resolving this issue is to estimate temporal coherence and then only study the temporally coherent targets on the ground. Ferretti et al. [59, 60] proposed permanent scatterer interferometry (PSI) in the early 2000s; it estimates the deformation parameters lying on the identified permanent scatters that are coherent over the temporal stack. Under PSI's scope, Werner et al. [61] applied interferometric point target analysis (IPTA) with a 2D regression algorithm to model the relations between the perpendicular baseline and temporal baseline. It estimates the terrain height and deformation using linear regression analysis. Another integer least squares-based technique—Spatio-Temporal Unwrapping Network (STUN) [62]—was proposed to solve the phase ambiguity problem via the least-squares ambiguity decorrelation adjustment (LAMBDA) method followed by a sequential least-squares search. However, both approaches require extensive search or complex transform computation to resolve the phase ambiguities. Persistent scatterer pairs (PSP) InSAR [63] and quasi-persistent scatterer (QPS) [64] avoided the complex parameter modeling by directly searching the parameters in the solution space. Although these algorithms are simple and flexible, they have a trade-off on accuracy. Especially for the large and complex deformation scenarios, their estimates are prone to be trapped in a local optimal solution [57].

Berardino et al. [65] introduced a small baseline subset (SBAS) algorithm to produce a mean deformation map of multilooked coherent pixels. However, it obtains deformation parameters using least square (LS) optimization from the unwrapped phase [54, 66], while PSI-based methods [57, 67, 68] can process both the wrapped and unwrapped phases. Although integrated PS-SBAS methods [53, 69, 70] have

been proposed, they rely on the unwrapped phase and assume that the baselines and arcs are small enough to avoid phase ambiguity of the phase gradient between two permanent scatterers. The main limitation of SBAS-based methods is their dependency on phase unwrapping, which is very time-consuming and requires a previously known displacement pixel as a reference, which may introduce errors for subsequent parameter estimation [54]. It is worth mentioning that the wrapped phase can cause a non-continuous searching space. Hence, retrieval of each absolute phase contribution from wrapped measures that are ambiguous by integer-multiples of 2π is a very challenging task [120]. Searching for the solution in the unwrapped phase is more straightforward than in the wrapped phase. However, phase unwrapping itself is a computation-consuming step that ideally should be avoided in time-sensitive applications. Moreover, those methods are just doing a 2D phase unwrapping without considering any temporal information. Hence, those methods can be error-prone and lead to significant unwrapping errors and an inability to measure fast motion in subsequent processing.

5.2 IGS-CMAES: A Two-Stage Optimization for Ground Deformation and DEM Error Estimation in Time Series InSAR Data

5.2.1 Introduction

A new direction has recently emerged in this topic. In [57], the authors tried to tackle parameter estimation by a stochastic optimizer—simulated annealing (SA). SA is a random search-based black-box optimizer that works on a given acceptance criterion (object function) to guide the search direction in a solution space. Following the traditional direct search methods, it avoids complex signal modeling and, at the same time, borrows the advantages of SA to reduce the local extrema effects when dealing with the wrapped phase. This work shows promising and better results compared to conventional gradient-based or direct grid searching algorithms by deploying a novel gradient-free optimization technique. However, SA is still a local optimizer, similar to many other stochastic optimizers, that is not guaranteed for global convergence. Note that, in real scenarios, the value range of estimated signals can be broad, e.g., mining sites and urban infrastructure, which can easily generate hundreds of meters DEM error due to open-pit mining. It also holds for the deformation rate because the ground is continuously disturbed, and pit walls tend to sag. To the best of our knowledge, most previous works have not considered such a large range of values.

Hence, when target areas have large-scale deformation and DEM error, those methods could become inefficient and not robust to precisely estimate the results because of the increasing searching space.

Investigations in [57] have demonstrated that it is feasible to use gradient-free optimization for deformation rate and DEM error estimation on a set of PS locations in TSInSAR. Their work’s main objective is to explore further the potential of using another advanced stochastic optimizer—covariance matrix adaptation evolution strategy (CMAES)—for this task. Moreover, we also demonstrate a global optimization strategy to enable a broad range of possible deformation rates and DEM error estimation. Our main contributions are: (1) We reformat the task as a two-stage 2D black-box optimization task. At the first exploration stage, an iterative grid search (IGS) policy is proposed to obtain coarse candidate solutions, which have a high chance to be close to global optima. Next, we employ CMAES for a fast local optimization at the second exploitation stage. (2) We present a hybrid benchmark simulation dataset that combines synthetic motion signals and DEM errors to real-world baseline parameters. Our proposed IGS-CMAES method has been assessed on both simulated data and real-world satellite data. We also compared our IGS-CMAES with various local and global optimization methods. The comparison results demonstrate the effectiveness and robustness of our method.

In this work, we first briefly review the mathematical phase model and definition of our optimization problem. Then, the CMAES algorithm is introduced, followed by a detailed explanation of the proposed IGS-CMAES method. Lastly, experimental results and discussion are presented before the conclusion.

5.2.2 Mathematical Modeling for InSAR Phase

Interferometric phase modeling has been investigated in the literature [56, 59, 67, 118]. Interferometric phase measurements are affected by various factors—imaging geometry, topography, atmospheric delay and ground deformation. For a given pixel location l in TSInSAR, interferometric phase can be represented in a differential interferogram [1] as follows:

$$\begin{aligned}\phi_l &= \phi_{def,l} + \Delta\phi_{topo,l} + \Delta\phi_{atmo,l} + \Delta\phi_{orbit,l} + \phi_{noise,l} \\ \hat{\phi}_l &= \arctan2(\sin(\phi_l), \cos(\phi_l))\end{aligned}\tag{5.1}$$

where ϕ_{def} represents phase components related to ground deformation motion, ϕ_{topo} is the topographic phase contribution (DEM error when applied external DEM), ϕ_{atmo} refers to the differences of atmospheric distortion between two single-look-complex (SLC) scenes, ϕ_{orbit} denotes imprecise satellite orbit data when forward modeling

the contributions of the Earth's ellipsoidal surface, and ϕ_{noise} describes decorrelation noise. The observed phase is also wrapped as $\hat{\phi}_l$. A general PSI processing chain eventually removes the flat-earth phase using satellite orbit data, and a collection of spatial and temporal filtering routines are adopted to reduce noise and remove other contaminant signals that are not deformation or DEM error [38, 39, 103]. Several established techniques are also described to mitigate the atmospheric phase contribution by deploying the toolbox for reducing atmospheric InSAR noise (TRAIN) [121, 122]. It is known that both ϕ_{orbit} and ϕ_{atmo} could create long-scale correlated signals in the spatial domain. Hence, those phase contributions can be further reduced by giving double difference phase between two neighbourhood PS pixels [57] as follows:

$$\Delta\phi_{i,j}^k = \Delta\phi_{topo,(i,j)}^k + \Delta\phi_{def,(i,j)}^k \quad k = 1, \dots, N \quad (5.2)$$

It describes the arc's double difference phase constructed by a pair of PS pixels i, j in interferogram k from a time-series stack with a length of N . In interferogram k , $\Delta\phi_{topo,(i,j)}^k$ is the relative height between pixel i and j , and $\Delta\phi_{def,(i,j)}^k$ is the relative deformation, respectively.

Furthermore, for each PS pixel, its topographic phase component can be modeled as a linear function of the spatial perpendicular baseline (B_{\perp}) according to the geometry relation of InSAR for each interferogram as:

$$\phi_{topo}^k = -\frac{4\pi}{\lambda} \frac{B_{\perp}^k}{R \cdot \sin\theta} \cdot h = conv_{topo} \cdot B_{\perp}^k \cdot h \quad (5.3)$$

where λ is the transmitted radar wavelength, R is slant range distance, θ denotes satellite incidence angle. Here, h is the orthometric height between two SLCs. We use $conv_{topo}$ to denote the unit conversion factor for a given stack. Similar to average ground deformation rate (m_r), which is used for the modeling deformation phase as follows:

$$\begin{aligned} \phi_{def}^k &= -\frac{4\pi}{\lambda} (d^{first} - d^{second})^k \cdot m_r \\ &= -\frac{4\pi}{\lambda} \Delta days^k \cdot m_r \\ &= conv_{def} \cdot \Delta days^k \cdot m_r \end{aligned} \quad (5.4)$$

where $\Delta days$ is the temporal baseline between two acquisitions on distinct days (d^{first}, d^{second}) used to form the interferogram, and $conv_{def}$ is unit conversion factor. We can substitute Equations (5.3) and (5.4) into Equation (5.2), and then add an extra integer variable w to handle phase ambiguity between pixel i and j because

of phase wrapping:

$$\Delta\phi_{i,j}^k = conv_{def} \cdot \Delta days^k \cdot \Delta m_{r(i,j)}^k + conv_{topo} \cdot B_{\perp}^k \cdot \Delta h_{(i,j)}^k + 2\pi \cdot w_{i,j}^k \quad k = 1, \dots, N \quad (5.5)$$

Due to the nature of microwave, the observed phase is wrapped as in Equation (5.1), there is an extra integer variable w that refers to phase ambiguity. It leads to N equations with more than N parameters that have to be resolved. That is why Equation (5.5) cannot be solved efficiently by a simple matrix inversion. As aforementioned, conventional PSI techniques consider this parameter fitting task as a 2D regression problem or an integer optimization problem. Those methods try to map the integer least square (ILS) and 2D solution search to the wrapped phase domain [63, 64, 123]. However, algorithms based on direct search are straightforward and do not require any complex modeling. They suffer from inefficient computation and are easily affected by local optima when dealing with complex baseline situation.

5.2.3 Proposed Method

Definition of Optimization Problem

In PSI frameworks such as DePSI (Delft PSI processing package) and StaMPS (Stanford method for persistent scatterers) [124, 125], signal separation is one of the essential steps in the whole processing pipeline. Parameters estimation is applied in an iterative manner combined with a collection of spatial and temporal filtering routines to obtain a precise final estimate. A more efficient and accurate parameter estimation algorithm could accelerate the whole processing pipeline by reducing iterates. Detailed descriptions of PSI methodologies can be found in [1, 51]. Our main focus is studying parameter estimation of linear deformation rate and DEM error upon PS time series, for the following reasons. (1) There are many state-of-the-art methods for filtering random noise and suppressing atmosphere components from a stack of interferograms [103–105, 109, 121]. (2) Recently, satellite facilities can provide accurate enough orbits for practical usage [126–128]. (3) It is very common to divide a complicated optimization problem into sub-problems, which are easier to be solved than the original problem [129, 130]. (4) Arc-based methods have to apply prior knowledge to pick reference points and then resolve each coherent pair’s parameters [54]. The main limitation of these methods is that it only gives the relative signal estimation between two PS points, which requires prior known deformation information of at least one PS point to derive the final estimations. It might not be feasible in real scenarios, where there is no information about the monitoring area. Moreover, it may potentially introduce accumulated error when a poor reference point is selected.

Our preprocessed observed phase is already filtered, and the atmospheric phase and orbit phase have been removed using 3vGeomatics’s preprocessing chain described in [131]. The pre-removal of these two large spatial correlated signals allows us to work on each PS signal directly instead of using arcs. To this end, our phase model becomes:

$$\phi^k = conv_{def} \cdot \Delta days^k \cdot m_r + conv_{topo} \cdot B_{\perp}^k \cdot h_e \quad k = 1, \dots, N \quad (5.6)$$

Note that, we replace the h (height) with h_e (DEM error), because we adopt external DEM to pre-remove the topographic phases. The outcome h_e will be used to refine external DEM as $DEM_{refined} = DEM - h_e$. We define the average deformation rate and DEM error estimation problem as a two variables optimization task that consists of an objective function. Please note that DEM studied in this work is in range Doppler coordinates (RDC), which can be transformed to geographic coordinates by the geocoding step.

The objective of our optimization task is to minimize the residuals between observed target phase ϕ_t and reconstructed phase ϕ_r , where ϕ_r is calculated using Equation (5.6) with estimated m_r and h_e . The typical objective functions for evaluating value difference in the continuous domain are mean absolute error (MAE) and mean square error (MSE). However, these Euclidean-based metrics are not suitable in the interferometric phase domain because of branch cuts. The value range of the wrapped phase is bounded by $[-\pi, +\pi)$, which results in interferometric phase value jumping from negative to positive or positive to negative π . In this work, we consider the wrapped phase difference with real and imaginary MSE (RI-MSE) (Equation (5.7)).

$$\mathcal{J}_{\beta} = \frac{1}{2N} \sum_{k=1}^N ((\sin(\phi_o^k) - \sin(\phi_r^k))^2 + (\cos(\phi_o^k) - \cos(\phi_r^k))^2) \quad k = 1, \dots, N \quad (5.7)$$

As shown in Figure 5.1, there are two phasors (ϕ_t Green, and ϕ_r Orange) plotted in a polar coordinate system. If we treat ϕ_r as the reconstructed phasor and ϕ_t as the target phasor, a Euclidean-based metric such as MSE would increase linearly with angle value difference. In terms of phase, the MSE objective function tends to guide the optimizer to move ϕ_r anticlockwise as shown in Figure 5.1 (MSE guidance direction), especially when their difference is close to π . RI-MSE can be interpreted as the distance between two phasors on the unit polar coordinate circle. During optimization, the optimizer adjusts variables in order to force the reconstructed phasor close to the target phasor based on the perspective of projections on two axes. This approach has been commonly deployed as a good indication of wrapped phase distances in recent InSAR phase filtering studies [38, 103, 132].

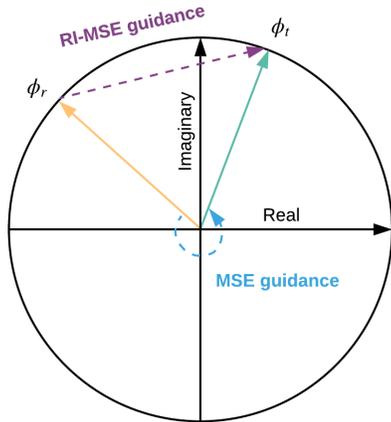


Figure 5.1: Illustration of phase jump and compassion real and imaginary MSE (RI-MSE) and mean square error (MSE) objective functions.

At this point, our signal separation task is formulated as a parameter fitting problem by optimizing an objective function. As aforementioned, there are still N equations with more than N interferograms to be estimated for each pixel pair because of the wrapped phase. That is why Equation (5.5) cannot be solved efficiently by conventional methods. Previous studies [63, 64, 123] point out that although the direct search-based algorithms are simple without using complicated modeling, they suffer from local optima when dealing with complex baselines and large solution space.

We propose a two-stage algorithm that combines a global coarse searching followed by a fine local optimization. As shown in Figure 5.2, we first adopt an IGS policy to obtain a set of coarse candidate solutions, which are expected to approximate the global optima. We then apply CMAES for a fast fine local optimization starting with each candidate solution. Lastly, the best result with the minimal objective function value is picked as the final estimate.

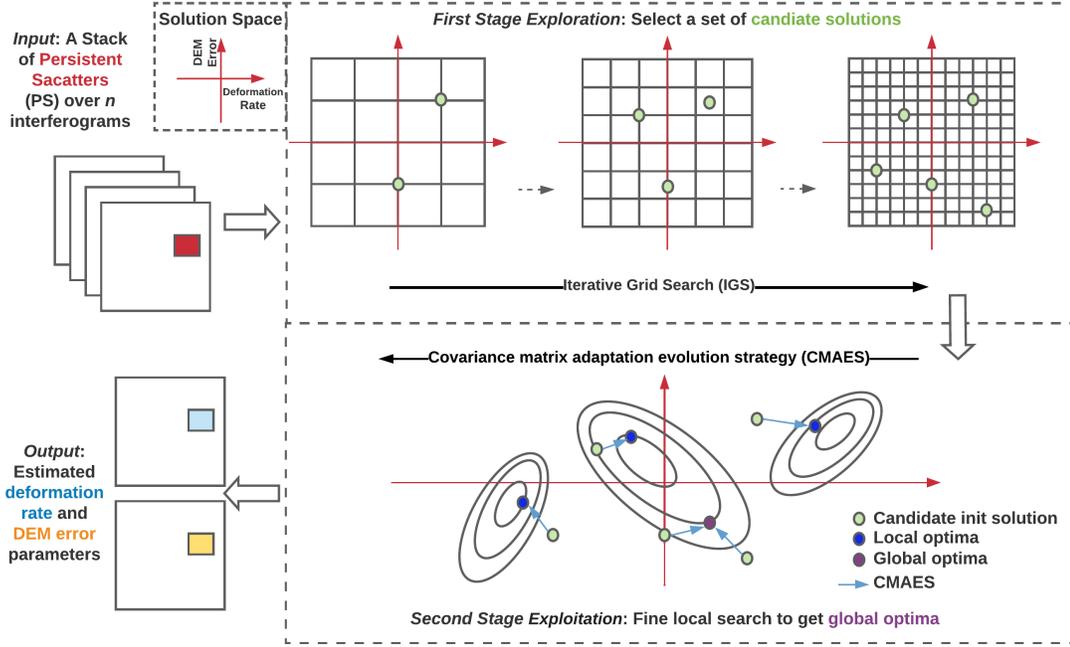


Figure 5.2: Illustration of proposed two-stage iterative grid search with covariance matrix adaptation evolution strategy (IGS-CMAES) method.

CMAES

In [57], researchers tried to adopt a black-box optimizer—SA algorithm, to solve the deformation fitting optimization problem. Inspired by it, we propose to use CMAES [133] to take advantage of using gradient-free optimization for our task. CMAES is an evolutionary-based stochastic optimization algorithm, which has shown state-of-the-art performance in derivative-free optimization and performed best among more than 100 classic and modern optimizers on a wide range of black-box functions [134, 135]. According to our phase model Equation (5.6), it can be seen that different temporal baselines ($conv_{def}$) and spatial baselines ($conv_{topo}$) result in very different objective functions. Its robust performance on optimizing unknown functions of CMAES is the major reason we choose it in our approach. When dealing with a few objective variables (two in our case), CMAES also obtains better speed than other methods [133, 136, 137]. A brief workflow of CMAES optimization is described as follows:

CMAES is an iterative algorithm, and there are three main steps in each iteration (t): (1) sample n candidate solutions from a multivariate normal distribution $N(m_t, \sigma_t^2 C_t)$; (2) calculate function values for each sampling solution, (3) update the distribution parameters (m_t, σ_t, C_t) accordingly. In this work, we use CMAES to min-

imize our loss function \mathcal{J}_β (Equation (5.7)) with two objective variables $\hat{\mathbf{v}} = [m_r, h_e]$ as shown below:

$$\hat{\mathbf{v}} = \underset{\mathbf{v}}{\operatorname{argmin}} \mathcal{J}_\beta \quad (5.8)$$

We first give CMAES an initial solution point as a start searching location $v_0 = [m_{r(0)}, h_{e(0)}]$ with an initial sampling distribution $N_0(m_0, \sigma_0^2 C_0)$, where $m_0 = v_0$, $C_0 = I \in \mathbb{R}^2$ and σ_0 denotes the initial step size, which is set to 0.01 in our study. Then, at each iteration t we generate S candidate solutions v_k sampled from N_t as:

$$v_k = m_t + \sigma_t y_k, y_k \sim N(0, C_t) \quad (5.9)$$

where k is the index of the randomly sampled candidate solutions with a total number of S . Here, each $y_k \in \mathbb{R}^2$ can be treated as a searching direction. Next, calculate all candidate solutions' objective function values $\mathcal{J}(v_k)$ and sort them as:

$$\mathcal{J}(v_{1:S}) \leq \mathcal{J}(v_{2:S}) \leq \dots \leq \mathcal{J}(v_{S:S}) \quad (5.10)$$

The subscript indicates the rank of those samples out of S . The optimizer will stop if the best solution reaches the termination criteria $\mathcal{J}(v_{1:S}) < \tau$, where τ is a threshold, which is set with a small value 10^{-11} . Otherwise, it uses the top μ ($\mu < S$) solutions to update the distribution parameters. Note that the rank order is only based on comparing the objective function itself, known as objective value-free ranking. In this work, we select top $\mu = \lfloor \frac{S}{4} \rfloor$ solutions with the lowest objective functions to update distribution parameters m_t and C_t , where the mean of the new distribution m_{t+1} is updated as

$$m_{t+1} = \sum_{k=1}^{\mu} w_k v_{k:S} = m_t + \sum_{k=1}^{\mu} w_k (v_{k:S} - m_t) \quad (5.11)$$

If we set $w_k = \frac{1}{\mu}$ for all candidate solutions, then the updates are treated as the maximum-likelihood estimation of all selected solutions. In this work, we want to emphasize the solutions with the lowest objective function and defined the function below as our weighting function:

$$w_i = 1 - \frac{\mathcal{J}(v_{k:S})}{\sum_{k=1}^{\mu} \mathcal{J}(v_{k:S})} \quad (5.12)$$

In this case, the updates in Equation (5.11) can be treated as a stochastic approximation of the natural gradient of m , which is recently used for optimizing deep neural networks on a reinforcement learning task [138]. The covariance matrix C is updated using:

$$p_{t+1} = (1 - c)p_t + \sqrt{c(2 - c)} \sqrt{\mu_w} \frac{m_{t+1} - m_t}{\sigma_t} \quad (5.13)$$

$$C_{t+1} = (1 - c_1 - c_\mu)C_t + c_1 p_{t+1} p_{t+1}^\top + c_\mu \sum_{k=1}^{\mu} w_k y_{i:S} y_{k:S}^\top \quad (5.14)$$

where p can be treated as the evolution path, which is 0 at the beginning and $\mathbf{1}$ is the symbol for vector transpose. It is similar to a mutation used for updating covariance matrix, and c is the learning rate for updating p . We set $c = 0.5$, which is the number of variables divided by 4 as recommended in [139]. μ_w is equal to $\frac{1}{\sum_{i=1}^{\mu} w_i^2}$, which is used for weighting intermediate recombination to force the second term as a random vector selected from $N(0, C_t)$. c_1 and c_μ are two other learning rates which are set to be $\frac{2}{n^2} = 0.5$ and $\frac{\mu_w}{n^2}$, respectively. To update step size σ , we adopt CMEAES's default cumulative step size adaption (CSA):

$$s_{t+1} = (1 - c_\sigma)s_t + \sqrt{c_\sigma(2 - c_\sigma)} \sqrt{\mu_w C}^{-1} \frac{1}{2} \frac{m_{t+1} - m_t}{\sigma_t} \quad (5.15)$$

$$\sigma_{t+1} = \sigma_t \exp\left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|s_{t+1}\|}{\mathbb{E}\|N(0, I)\|} - 1\right)\right) \quad (5.16)$$

where s is another conjugate evolution path that is similar to p but ignoring the scale fact. c_σ and d_σ are two parameters for controlling the changing magnitude of σ are set to $\frac{2}{n^2} = 0.5$ and $1 + \sqrt{\frac{\mu_w}{n^2}}$, respectively. We adopt the default configuration defined in the literature [140] for this study, and the key parameters of the CMAES algorithm are listed in Table 5.1.

Once the distribution parameters are updated, the optimizer will start another new iteration until it reaches the termination condition. The main advantage of CMAES over classical ES is the use of correlated mutations instead of axis-parallel ones. It can learn appropriate mutation distribution steadily and has a high probability of reaching optima by using adapted covariance matrix C to adjust searching direction [136]. However, CMAES is still a local search optimizer similar to SA and many other stochastic optimizers. They may get stuck in local optima, and the convergence to global optima is not guaranteed [135, 139]. Many works have pointed out that the initial search position is essential for stochastic optimization. Restart-based methods are very classical but useful in many optimization frameworks and show benefits in finding the global optima. There are also many CMAES extensions [134] using restart strategies to prevent premature convergence on complicated tasks. The main limitation of those approaches is that they need extra search in the hyper-parameter space of the population size, and the initial step-size seems inefficient. Moreover, the subsequent restarted search usually relies on previous search results, making the whole optimization process hard to be parallelized. However, there is evidence to

support the fundamental idea behind restarts; the initial search location is essential for finding the global optima and saving the computations. To address these issues, we propose a pre-stage exploration search for picking a set of potential candidate solutions for CMAES. Then, CMAES will use those initial solutions for fine local optimization as the second exploitation stage.

Table 5.1: The key parameters of covariance matrix adaptation evolution strategy (CMAES) used in this work.

Parameter	Definition	Value
S	Number of candidate solutions at each iteration	30
σ_0	Initial step size	0.01
μ	Number of selected top ranked solutions	7
τ	Threshold value to terminate the optimization	10^{-11}
c	Learning rate for updating evolution path	0.5
c_1	Learning rate for updating covariance matrix	0.5
c_σ	Learning rate for updating step size	0.5

Iterative Grid Search (IGS)

To find good candidate solutions for fine local optimization, we introduce an IGS strategy in this work. In Figure 5.3, we examine the loss landscapes computed using Equation (5.7). We inject 0 deformation and 0 DEM error as ground truth with two site baseline parameters ($conv_{def}, conv_{topo}, \Delta days, B_\perp$) for generating simulated wrapped interferogram using Equation (5.6). We then oversample both deformation rate (1000 samples) and DEM error (1000 samples) to better illustrate the surface of the corresponding loss value landscape ($1e6$ evaluated solutions). By observing Figure 5.3, it can be seen that both landscapes are non-convex, rugged and contain many local optima. This is the reason why a simple gradient-based optimizer is hard to find global optima. The desired algorithm is expected to converge to global optima effectively and avoid brute force grid search because naïve sampling is impractical when high precision estimation and real-time performance are required. Comparing landscapes of the two sites also shows that different real-world baselines could result in very distinct objective functions. Some site baseline parameters may produce much

more challenging objective functions than others.

In fact, a straightforward grid search is commonly used in industry processing pipelines. For a ± 26 cm/year linear deformation rate and ± 200 m DEM error study case, a typical grid search is applied as S_s with 0.5 cm and 2 m step size (respective to deformation rate and DEM error) in order to pick the best solution. It generates $(2 \times 200/2) + (2 \times 26)/0.5 = 20,800$ number of objective function evaluations in total for each PS location and has a limited precision bottleneck regarding the step size. Increasing the step size can reduce the computations but also decrease the precision. Furthermore, a large step will result in poor estimation for complicated baselines because of the objective function’s ruggedness. We treat κS_s as sampling step size for deformation rate and DEM error, where a scaling factor κ is used to control the density of grid search. In an optimization task, the number of objective function evaluations (**search cost**) is an essential metric for assessing a stochastic optimizer’s performance. Especially when higher precision is required, dense grid search is extremely insufficient. Moreover, from an optimization point of view, determining whether the true global optima reached is a fundamental challenge as a stochastic optimization algorithm. As a consistency check, the algorithm can be run from several different random starting points to ensure the result of each run converges to the global optimal. Rather than using randomly selected starting points, in this work, we hypothesize that using a low-precision grid search can select a set of initial candidate solutions that are potentially close to the global optima. To support our assumption, we plot the loss landscapes with different sampling steps sizes by setting $\kappa = [1, 3, 5, 8]$ as shown in Figure 5.4.

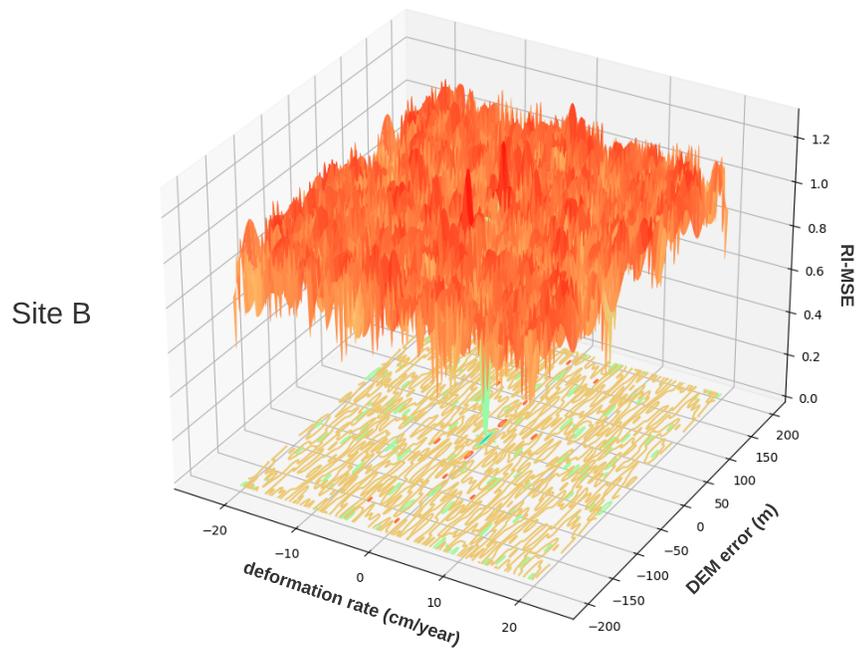
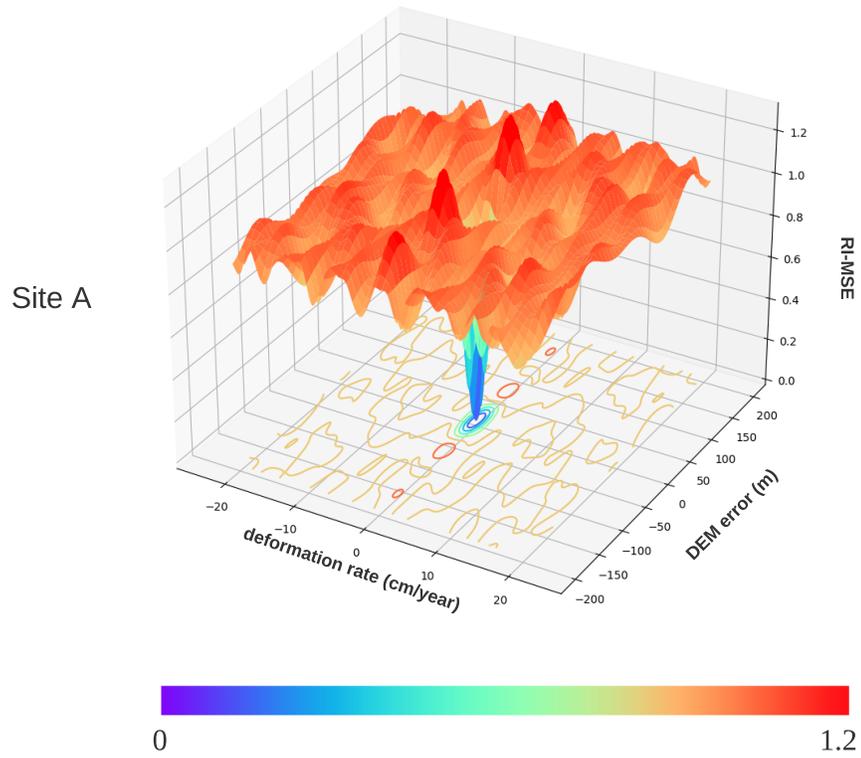


Figure 5.3: Loss landscape of objective function RI-MSE (Equation (5.7)) with two different real-world spatial and temporal baselines. Site (A) and Site (B) are two different real-world sites.

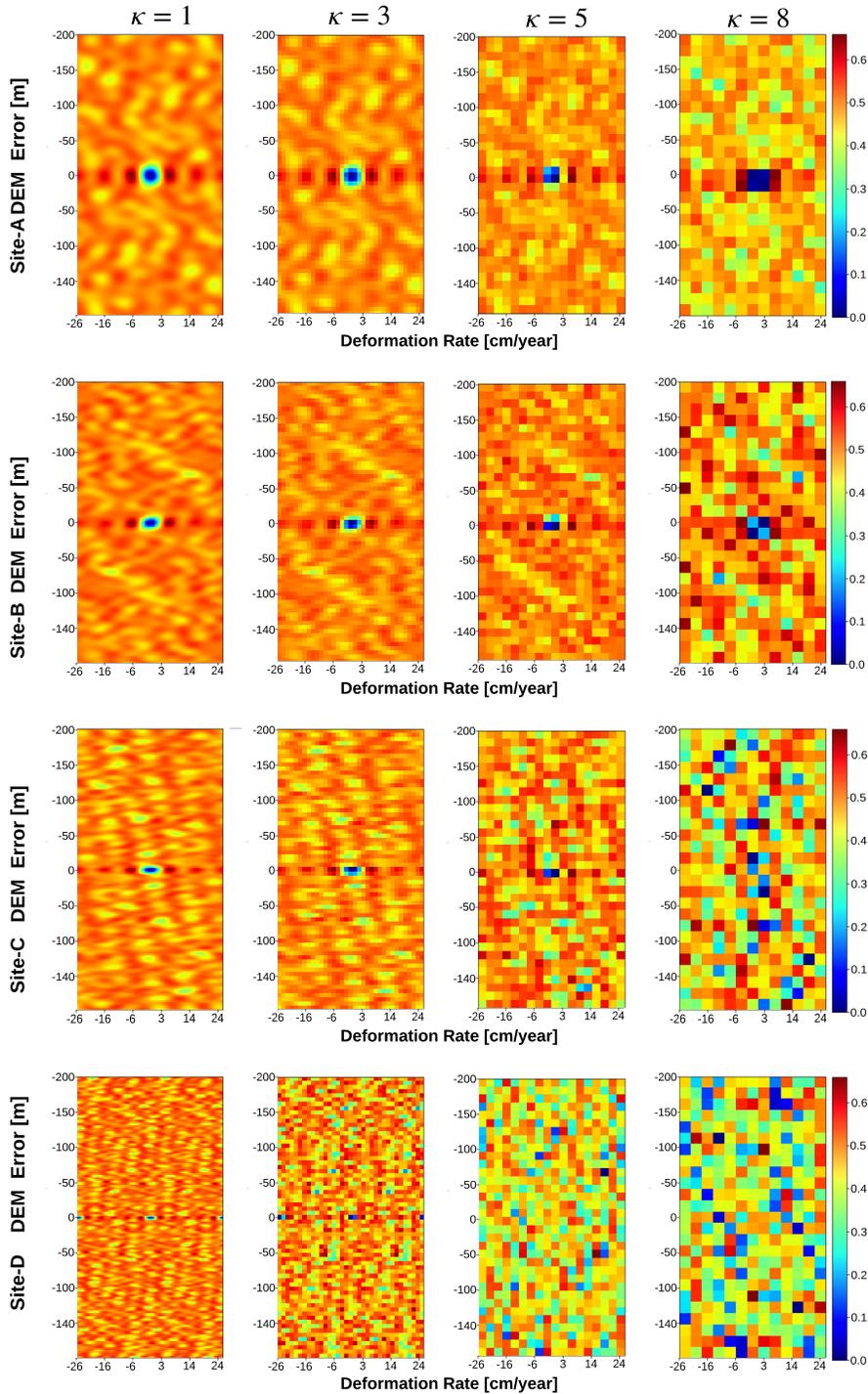


Figure 5.4: The landscape of loss function. Each row represents the loss landscapes that corresponds to real-world baselines with synthetic 0 deformation and 0 DEM error, and each column shows the results under different step sizes of grid sampling. The linear deformation rate x -axis and DEM error y -axis form a 2D solution space for an InSAR pixel location. The value is calculated by Equation (5.7) with selected real-world baselines.

By observing the resulting loss landscapes at different sampling scales, we notice that global optima can be roughly located with a less dense grid sampling. As seen in Site-A and Site-B from Figure 5.4, $\kappa = 5$ or 8 is sufficient to estimate the location of global optima. However, when dealing with complicated baselines such as Site-C and Site-D, it is necessary to require $\kappa = 1$ or 3 in order to roughly locate the optimal solution. A large sampling step size (large κ) leads to low precision, but a small sampling rate (small κ) brings massive computations. Figure 5.4 shows that there is no fixed optimal sampling scale κ for different baselines. Hence, in this work, we propose an iterative strategy by performing grid sampling from large κ to small units until N acceptable initial solutions have been picked. We define a threshold ω for accepting sampled estimates, whose values are less than ω as acceptable initial solutions.

IGS performs grid sampling with different κ iteratively, and the key steps at each iteration are: (1) sort all sampled estimates based on their loss values (Equation (5.7)); (2) if the smallest one is greater than ω , the algorithm will iterate to the next grid search level. Otherwise, it pushes the estimate to candidate solution list one by one until the length of N or no solution loss is less than ω . Moreover, solutions close to global or local optima might have similar loss values, as shown in Figure 5.5 (Estimate C, 2, 3). To avoid selecting solutions from the same local area in the landscape, we define a *CheckDistance* procedure to skip solutions that are too close to any existing accepted solutions. A threshold value ψ is adopted to determine if two solutions are too close to each other by assessing two estimates L2 distances in the solution space. As shown in Figure 5.5, if point C is already labeled as a candidate solution, points 2 and 3 will not be considered. Instead, the algorithm will consider points 4 and 5 because they are not close to any existing candidate solution.

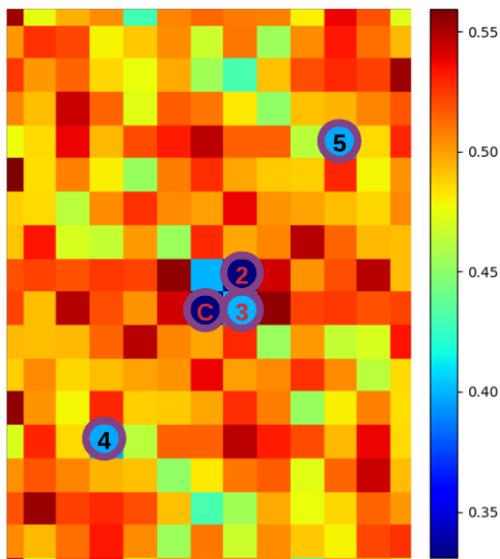


Figure 5.5: A simple illustration of selecting candidate solutions by skipping the estimates (Estimate 2, 3) that are too close to any picked candidate solution (Estimate C).

IGS-CMAES

Once we get a certain number of *acceptable* initial solutions, CMAES is then applied for local fine optimization. In fact, there are existing CMAES extension works that are proposed based on a repeat mechanism. The key concept of those methods is repeating CMAES process by adjusting the initial point and population size according to the results from previous runs. They offer better performance but increase the computation time because the adjustments are not parallelizable. In our work, CMAES runs only once for each candidate solution as a starting location. Hence, the local search can be performed in parallel to save overall execution time compared to repeat-based algorithms. Our complete IGS-CMAES optimization procedure is given in Algorithm 1.

To summarize, we propose to use IGS to globally select acceptable initial solutions, which are with high possibility of close to the global optima. We then apply CMAES, started from each selected initial solution, to perform the local refine search. Lastly, the final result is the best estimate among all local solutions. Overall, IGS-CMAES can save unnecessary function evaluations compared to dense grid search but still cover the global optimal space. At the same time, it also preserves the benefits of accuracy and efficiency from the local search.

Algorithm 1: IGS-CMAES for deformation rate and DEM error estimation.

```

input :  $\phi_o \in \mathbb{R}^k, conv_{def}, conv_{topo}, \Delta days \in \mathbb{R}^k, B_{\perp} \in \mathbb{R}^k, \omega = 0.3, N_c = 5$ 
// First Exploration Stage: select candidate initial solutions
 $V_{init} = \{\}$ ;
for  $\kappa \leftarrow [8, 7, 6, 5, 4, 3, 2]$  do
     $S \leftarrow GridSampling(\kappa)$ ;
     $L(i, j) \leftarrow \mathcal{J}(S(i, j))$ ;
    for  $v_c \leftarrow L_{min}$  to  $L_{max}$  do
        if  $v_c < \omega$  && DistanceCheck( $v_c, V_{init}$ ) is True then
            push ( $v_c$ ) into  $v_{init}$ ;
            if  $size(V_{init}) > N_c$  then
                Break; // End for Initial Solutions Selections
            else
                if  $size(V_{init}) \leq N_c$  then
                    Continue; // Next Grid Sampling with smaller  $\kappa$ 
// Second Exploitation Stage: perform Local CMAES search
 $V_{final} = \{\}$ ;
for  $v \leftarrow V_{init}$  do
    Run CMAES local starting from  $v$ ;
    Push final solution  $v_s$  into  $V_{final}$ ;
output:  $v_s$  with lowest  $\mathcal{J}$  in  $V_{final}$ 

```

5.2.4 Results

Experimental Setup

This section empirically demonstrates the effectiveness and robustness of the proposed IGS-CMAES using both simulated and real-world TSInSAR data. All our experiments are based on seven real-world stacks R (R1–R7), captured by TerraSAR-X in StripMap mode [91]. Each stack represents a different ground site and contains 31 SLCs except R7, which has 17 SLCs. We designed a simulator that generates TSInSAR signals by injecting synthetic deformation rate and DEM error with real-world baseline parameters. Moreover, we apply an industry 3vGeomatics’s industry-standard InSAR processing pipeline [131, 141] to perform data pre-processing as well as generate the reference results to assess the performance of the proposed IGS-CMAES method on real data. The experiments were designed to estimate two important parameters: optimization cost and accuracy. Optimization cost is defined as the number of objective function evaluations. Accuracy is assessed directly by comparing with synthetic deformation rate and DEM error values in the simulation setup. Because there is no ground truth in the real-world scenario, we adopt mean phase residual (MPR) between the wrapped reconstructed and input interferograms to evaluate performance. The details of both experiments and results are presented in the following sub-sections, and the code is available at: https://github.com/Lucklyric/TSInSAR-PF-IGS_CMAES (accessed on 23 June, 2021).

Simulation Data

Our simulator generates interferometric phase by injecting synthetic deformation rate and DEM error into seven real-world temporal and spatial baselines R1–R7. For each dataset, in order to assess the robustness of different optimizers, we sample 30 deformation rates and 60 DEM errors from two uniform distributions - $\mathcal{U}(-26, 26)$ cm/year and $\mathcal{U}(-200, 200)$ m. It gives a total of 1800 possible signal pairs for each of the seven baselines, which are further used to generate synthetic interferometric pixel stacks using Equation (5.6), where $conv_{def}$, $\Delta days$, $conv_{topo}$ and B_{\perp} are real-world baselines from the R1–R7 stacks. In order to objectively evaluate different methods, we use (1) root-mean-square error (RMSE) between the simulated (ground truth) and the estimated deformation rate (cm/year) and DEM error (m), (2) L1 unwrapped phase difference (L1-UPD) between the simulated (unwrapped ground truth) and the reconstructed phase calculated based on the estimated deformation rate and DEM error, (3) the accuracy (ACC%) of reducing rate that counts the

percentage of how many test cases that the method’s unwrapped L1 phase error is less than π . We performed our experiments from three perspectives: (1) apply four widely used local optimizers—(a) least square (LS), (b) Nelder–Mead [142], (c) conjugate gradient (CG) [143] and d) Broyden–Fletcher–Goldfarb–Shanno (BFGS) [144], to show how challenging our task is when applying conventional local optimizers; (2) adopt the IGS strategy into all local optimizers used in the first experiment to showcase the improvements in comparison with our coarse search strategy; (3) compare to the other two global optimization methods—direct grid search (baseline method) and dual simulated annealing (Dual-SA) [145]—to investigate the novelty of our IGS-CMAES algorithm.

Table 5.2: Quantitative assessment for simulated data with several widely-used local optimizers and their IGS-extended version

Baseline	Categories	LS	IGS-LS	Nelder-Mead	IGS-Nelder-Mead	CG	IGS-CG	BFGS	BFGS-LS
R1	m_r -RMSE (cm/year)	12.2382	2.9481	12.4695	2.0734	12.5197	1.3484	12.4414	1.2893
	h_e -RMSE (m)	107.0381	0.0011	105.6981	0.0001	107.6607	0.0000	106.5213	0.0000
R2	m_r -RMSE (cm/year)	13.2260	2.3247	13.3805	2.5936	13.6642	1.3439	13.6011	1.3143
	h_e -RMSE (m)	107.2283	0.0016	106.3737	0.0002	107.4976	0.0000	107.5918	0.0000
R3	m_r -RMSE (cm/year)	13.0308	1.9413	13.1002	1.6877	13.3366	1.4307	13.2444	1.0897
	h_e -RMSE (m)	107.8291	0.0016	106.5126	0.0002	108.3554	0.0000	108.7696	0.0000
R4	m_r -RMSE (cm)	12.7119	1.9372	13.1455	2.2679	13.1560	0.8951	13.2614	2.0387
	h_e -RMSE (m)	107.5101	0.0032	106.5326	0.0002	108.2290	0.0000	108.4366	0.0000
R5	m_r -RMSE (cm/year)	12.6042	2.1055	12.6375	1.9330	12.8295	0.8136	12.8457	1.5362
	h_e -RMSE (m)	106.7188	0.0022	105.9384	0.0002	109.1707	0.0000	107.6000	0.0000
R6	m_r -RMSE (cm/year)	12.4571	1.9549	12.5720	1.7121	12.5872	0.6817	12.6906	0.7807
	h_e -RMSE (m)	105.3393	0.0062	106.0447	0.0003	108.2960	0.0000	109.1020	0.0000
R7	m_r -RMSE (cm/year)	12.5991	14.9993	12.6070	14.1472	12.7588	6.7126	12.6561	3.2887
	h_e -RMSE (m)	106.6014	0.0006	106.0582	0.0001	108.0011	0.0000	12.6561	0.0000

From Table 5.2, it is obvious that none of the local optimizers works for any stack. They show significant errors on both types of signal estimation. During the experiments, all local optimizers prone to converge to local optima around the initial solution. However, comparing to IGS-extended versions, which run each local optimization following IGS policy, there is a consistent agreement to reduce the errors in deformation and DEM error estimations. As shown in Figures 5.6 and 5.7, a significant decrease in L1-UPD and improvement in ACC is observed, when IGS is adopted.

The proposed IGS policy effectively improves all local optimizer performance in the unwrapped phase domain, except R7, which is a special case with fewer SLCs. Nevertheless, all IGS-optimizers produce acceptable results. However, due to the limitation of those local optimizers themselves, the results are still not satisfactory.

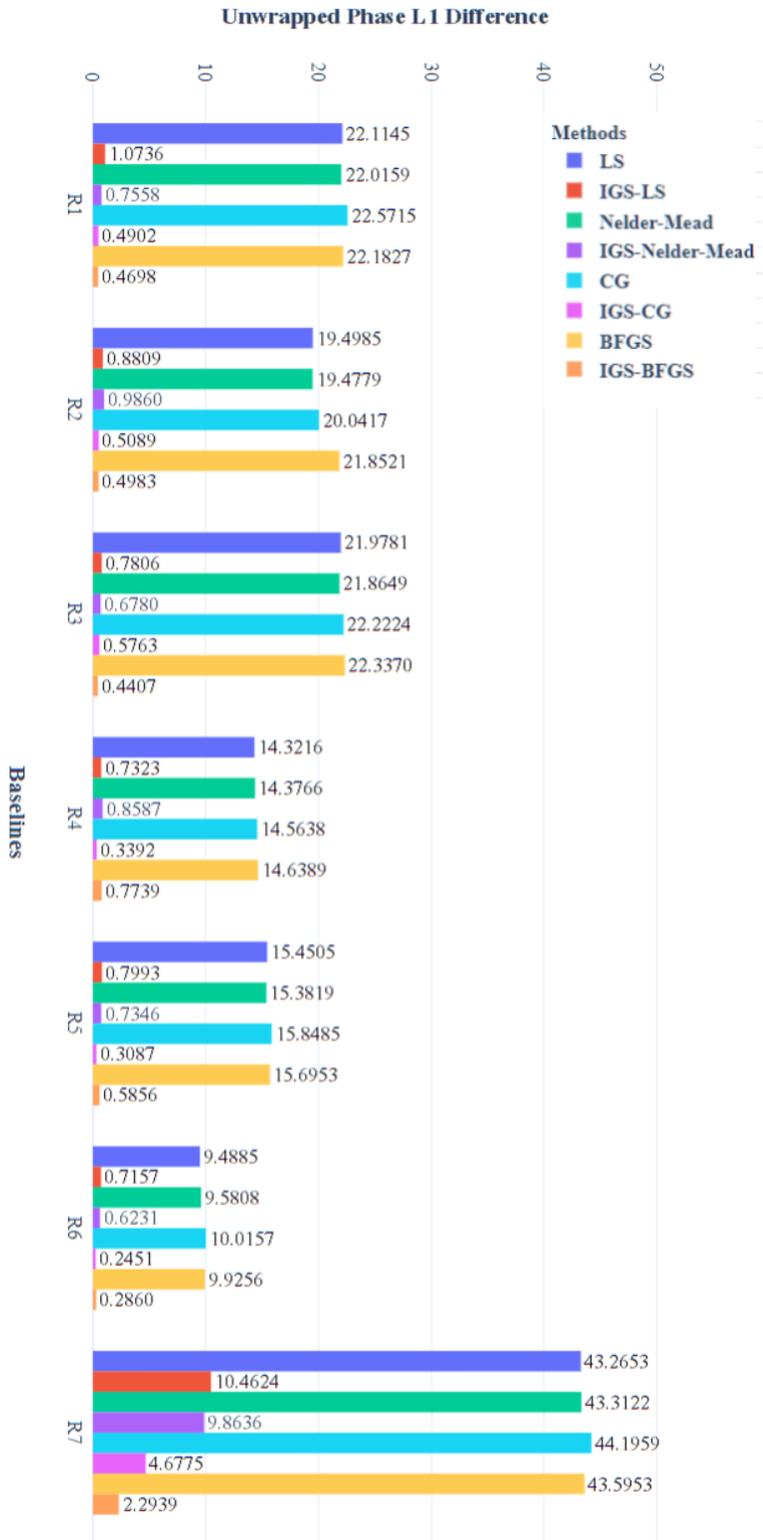


Figure 5.6: Unwrapped phase L1 difference.

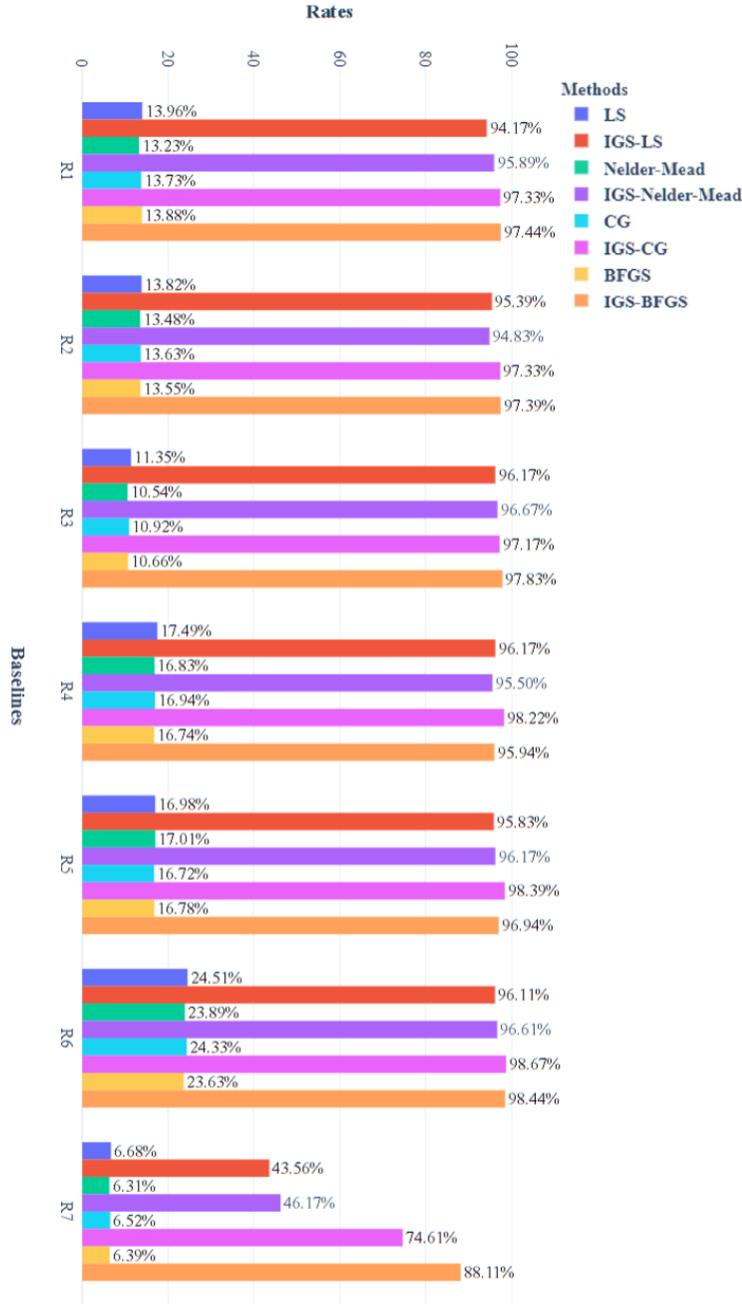


Figure 5.7: Reducing rates (ACC%) of unwrapped phase L1 difference less than π .

In this work, we also provide comparison between the proposed IGS-CMAES and other two global algorithms—grid search and Dual-SA with an extra metric—the mean number of objective function evaluation (NFev). Both methods are designed for global optimization. The comparisons reported in Table 5.3 show that all three methods have an on-par accuracy on deformation estimation when using R1-R6. However, Dual-SA fails to give proper estimates of DEM error except for R6. The error of grid

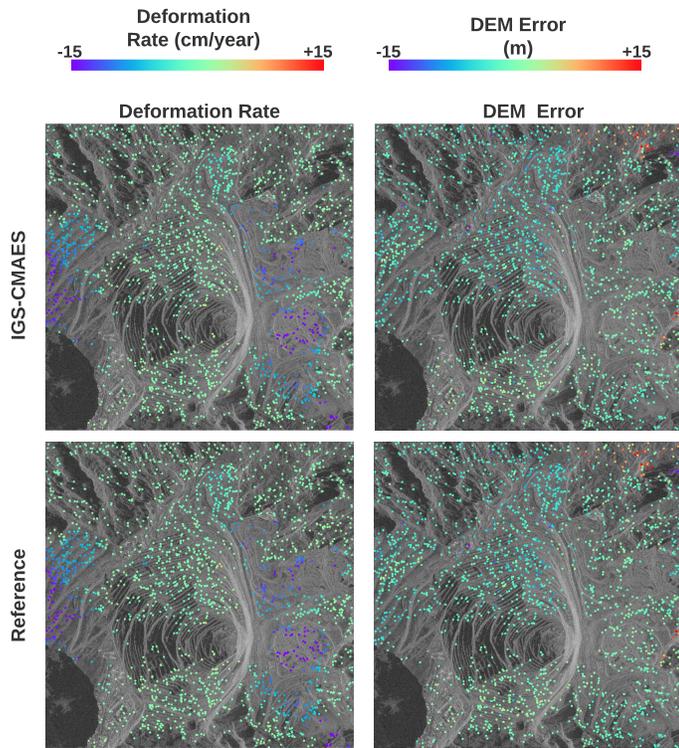
search is mainly due to its limited sampling precision and it fails to provide proper deformation estimation when working on limited temporal information R7, which also troubles Dual-SA. In contrast, our IGS-CMAES consistently offers robust and accurate results, and achieves better generalization ability than other two methods. Furthermore, IGS-CMAES has demonstrated a significant improvement in NFev by saving more than 85% objective function computation costs compared to grid search, while maintaining similar and even better results. Our method surpasses Dual-SA on all baselines with substantial improvement in both accuracy and efficiency.

Real-World Data

In this section, we present results obtained when applying IGS-CMAES to real interferometric SAR acquisitions. All input interferograms from the R1–R7 have been preprocessed by the 3vGeomatics processing pipeline [131] with proper filtering, earth flattening and atmospheric phase removal as discussed in Section 5.2.3. There are 2000 PS pixels selected from each stack, which results in 14,000 test data points. Because there is no ground-truth data for real-world data, we adopt the final results from the processing pipeline as reference output. Such pipeline involves a phase unwrapping step and then uses least-square optimization to approximate the linear deformation rate and DEM error on the unwrapped phase directly. Phase unwrapping is known to be very time-consuming, but we can treat those outputs as reference results. It is worth mentioning that our proposed IGS-CMAES is designed to work on the wrapped phase directly in order to save the computations in unwrapping.

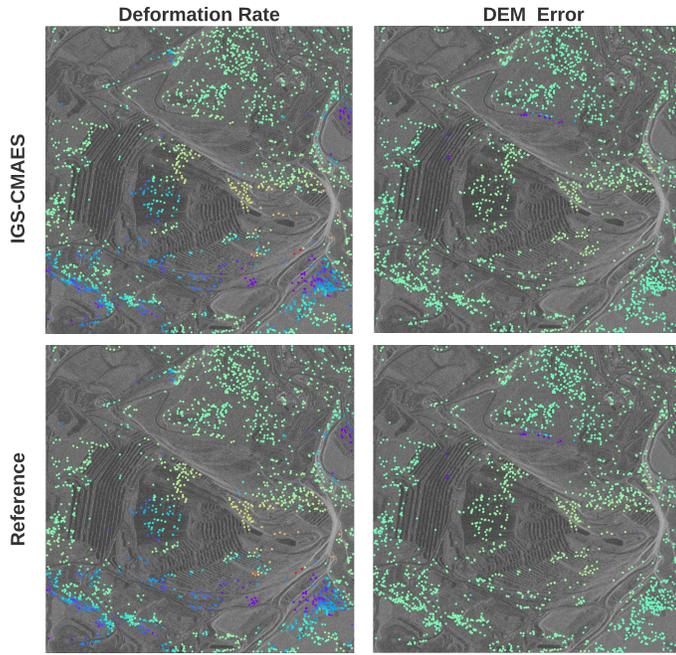
We estimate the linear deformation rate and DEM error using the proposed model on selected PS locations. Visual outputs and comparison to the reference results can be found in Figure 5.8. It is shown that the estimates of IGS-CMAES match the reference output pretty well in R1-R6. There are a few apparent disagreements on deformation rate in R(7), where reference results indicate small movements, but IGS-CMAES predicts high positive displacements (red dots). After careful examination, we present our numerical analysis in Table 5.4. We calculate m_r -RMSE and h_e -RMSE to quantify the difference between the IGS-CMAES and reference results. The only significant mismatch is the deformation rate in R7 (0.190598), which correlates with the observation in Figure 5.8. All other categories stay commensurate with a little disparity. We further investigate the wrapped phase residuals (WPR) between the wrapped reconstructed phase and input interferogram, and our method shows lower residuals than the reference output. This finding can also be confirmed by checking RI-MSE (Equation (5.7)), which is presented to show the phase distance in the polar coordinate system (Figure 5.1). Experimental results reveal that our

IGS-CMAES offers more performance advantages on global convergency than the reference output. Moreover, our method can be applied to wrapped interferograms directly while achieving equal performance compared to the reference method, which requires phase unwrapping. Lastly, the algorithm continuously performs around 3800 NFev for each baseline. Therefore, we are confident that IGS-CMAES can serve as an efficient optimizer and provide accurate global fitting.

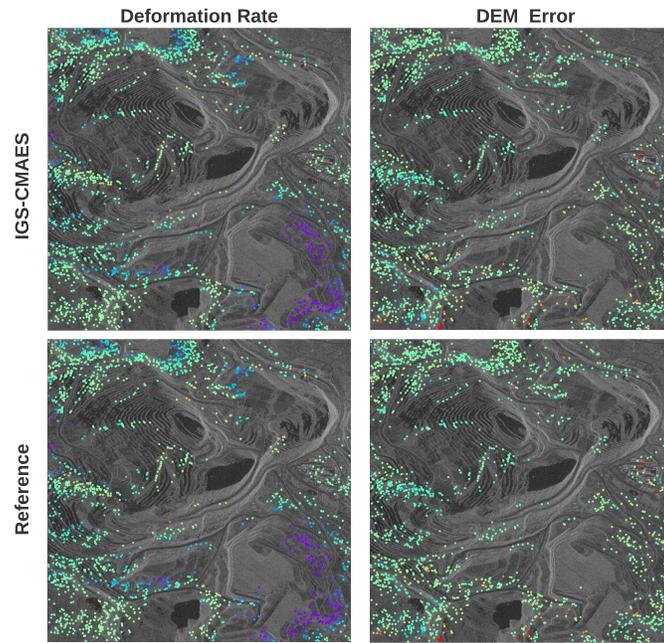


(a) R1

Figure 5.8: *Cont.*

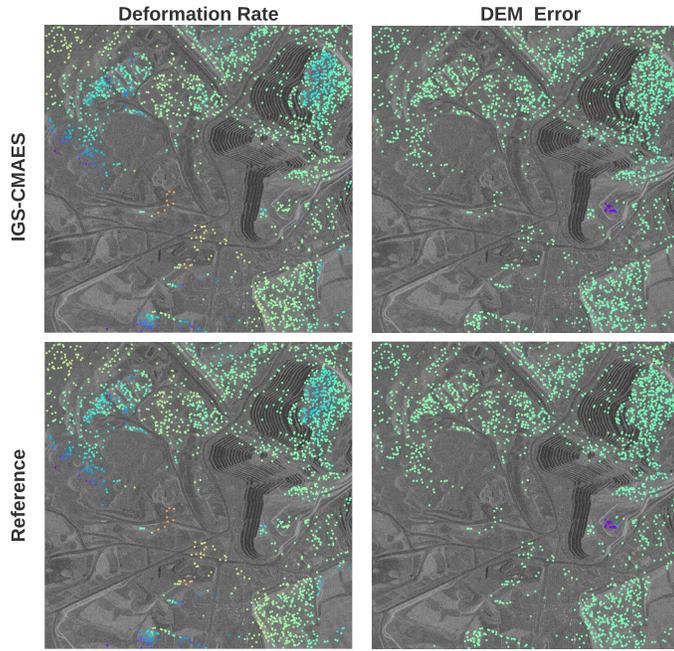


(b) R2

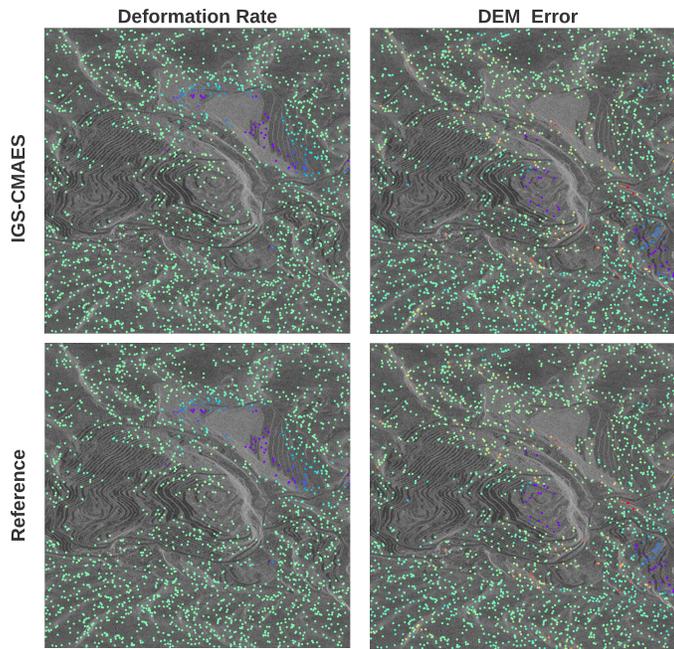


(c) R3

Figure 5.8: *Cont.*

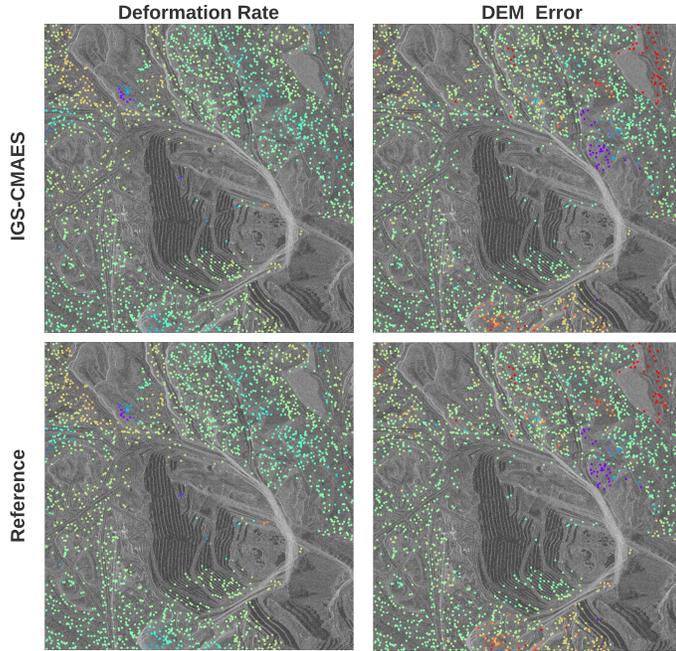


(d) R4

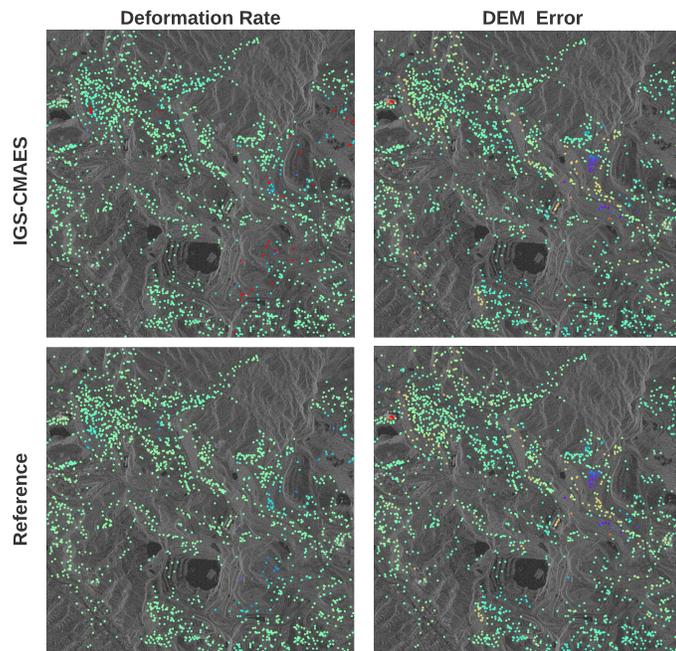


(e) R5

Figure 5.8: *Cont.*



(f) R6



(g) R7

Figure 5.8: Visualization of estimated linear deformation rate and DEM on R1–R7 real-world stack’s PS pixels. IGS-CMAES is applied on wrapped interferogram directly. Reference results are generated using 3vGeomatics processing pipeline with unwrapped phase.

Table 5.3: Quantitative comparison for simulated data using IGS-CMAES with two other Global Optimizers

Baseline	Categorie	IGS-CMAES	Grid-Search	Dual-SA
R1	m_r -RMSE	0.0284	0.0967	0.6336
	h_e -RMSE	0.0000	0.5611	31.7574
	L1-UWPD	0.0424	0.1191	1.3113
	ACC	99.94%	100%	96.88%
	NFev	2725.03	20800	4109.38
R2	m_r -RMSE	0.1137	0.0967	0.1792
	h_e -RMSE	0.0000	0.5610	6.7533
	L1-UWPD	0.0424	0.1051	0.1279
	ACC	99.94%	100%	99.35%
	NFev	2585.18	20800	4103.06
R3	m_r -RMSE	0.1991	0.0967	0.4899
	h_e -RMSE	0.0000	0.5613	15.5361
	L1-UWPD	0.0791	0.1180	0.7239
	ACC	99.61%	100%	96.33%
	NFev	2687.90	20800	4112.38
R4	m_r -RMSE	0.2276	0.0967	0.0588
	h_e -RMSE	0.0000	0.5617	7.5336
	L1-UWPD	0.0848	0.0806	0.0534
	ACC	99.56%	100%	99.69%
	NFev	2500.69	20800	4089.58
R5	m_r -RMSE	0.2844	0.0967	0.1556
	h_e -RMSE	0.0000	0.5617	5.8893
	L1-UWPD	0.1065	0.0864	0.0853
	ACC	99.44%	100%	99.52%
	NFev	2520.12	20800	4093.36
R6	m_r -RMSE	0.0000	0.0967	0.0000
	h_e -RMSE	0.0000	0.5617	0.0000
	L1-UWPD	0.0000	0.0591	0.0000
	ACC	100%	100%	100.00%
	NFev	2381.48	20,800	4070.47
R7	m_r -RMSE	1.6145	7.7633	13.9020
	h_e -RMSE	0.0000	0.5609	104.14
	L1-UWPD	1.1259	5.5327	27.8864
	ACC	94.88%	70%	39.03%
	NFev	3576.92	20800	4121.54

Table 5.4: Quantitative assessment of IGS-CMAES’s outputs with reference results

Baseline	NFev	RI-MAE	RI-MAE	WPR	WPR	mr-RMSE (cm)	he-RMSE (m)
		IGS-CMAES (rad)	Reference (rad)	IGS-CMAES (rad)	Reference (rad)		
R1	4023.67	0.305124	0.307999	0.431716	0.435595	0.114381	0.061021
R2	3960.59	0.306338	0.307771	0.438660	0.439584	0.080547	0.053405
R3	3872.19	0.238092	0.242657	0.371183	0.374736	0.109501	0.052438
R4	3510.90	0.120080	0.119850	0.238335	0.238453	0.034719	0.048583
R5	3563.92	0.120687	0.122859	0.249803	0.250186	0.059676	0.122966
R6	3496.90	0.161789	0.161793	0.321250	0.321475	0.025550	0.102940
R7	3817.64	0.191073	0.210659	0.326159	0.352141	0.190598	0.061347

5.2.5 Discussion

Average ground deformation rate and DEM error estimation concurrently using TSIn-SAR stack can naturally be treated as a 2D optimization task. However, due to the wrapped phase, the resulted objective function could contain many local optima that results in a pure local optimizer easily stuck to semi-optimal solutions. Moreover, varying spatial and temporal baseline parameters ($conv_{def}$, $\Delta days$, $conv_{topo}$ and B_{\perp}) also lead to different objective functions to be resolved. Hence, both global convergence and generalization ability are considered in our method. The proposed IGS policy tackles this optimization problem by iteratively sampling to select candidate solutions at a coarse level. It avoids inefficient naïve grid search but also retains a global exploration. Our simulation experiments confirm the benefits of the proposed IGS policy in Table 5.2 and Figures 5.6 and 5.7. The failing local optimizers tend to exhibit significant improved results after IGS boosting. However, due to the bottlenecks of local optimizers, their results can only be as good as dense grid search and Dual-SA (Table 5.3).

Under the scope of global optimization, the performance of naive dense grid search highly relies on the sampling step size. An inevitable trade-off between precision and computation efficiency limits its usage when millimetric accuracy is required. SA algorithm is demonstrated in a recent study [57] on small-scale motion detection, which inspires us to adopt a stochastic gradient-free optimizer in this work. Its global extension Dual-SA combining Classical SA, and Fast SA has shown better results than other conventional optimizers. However, SA-based methods highly depend on randomness, and Dual-SA’s global search policy relies on hyperparameter tuning to generalize to different objective functions [134, 145]. It can be observed from Table

5.3, that Dual-SA results show unstable performance when dealing with different baselines. In contrast, CMAES leads to fast local convergence with adaptive searching direction using the covariance matrix and has shown superior performance on many local optimization benchmarks. In addition, we can achieve fast convergence with CMAES by setting a small population size and step size because we have conducted sufficient exploration during the first IGS stage. At the second stage of optimization, CMAES can directly focus on local optimization for exclusive exploitation.

In our real data experiments, reference results are generated by 3vGeomatics’s industry level processing pipeline. The pipeline has served industry customers for years, and it is reasonable to treat its outputs as empirically validated references. Besides its robustness, this pipeline requires phase unwrapping to eliminate phase ambiguities before signal estimation. As aforementioned, phase unwrapping itself is an expensive step and has a dependency on prior known displacement pixel as the reference [54]. Due to our data nature, where no ground truth is available, it does not allow us to perfectly assess IGS-CMAES’s estimates. Hence, we use reference results that are based on wrapped phase. The proposed IGS-CMAES achieves a comparable output but on wrapped phase directly. It substantially improves existing work by skipping the complicated unwrapping step and preserve a robust estimation.

Lastly, we want to discuss the disagreement in the R7 stack, which shows a few miss-matched estimations between IGS-CMAES and the reference output. This is an interesting finding, as the statistics in Table 5.4 suggest that IGS-CMAES has very similar results as the reference one on all other stacks. Though significant differences happen in R7, IGS-CMAES shows consistent lower phase residuals between the wrapped and input phases. To this end, we can only conclude that the proposed IGS-CMAES provides robust convergence from an optimization point of view. However, our optimization tasks and objective function are defined following the literature of the linear deformation model. It is worth mentioning that R7 is the only stack with just 17 SLC acquisitions comparing to all other stacks (31 SLC). A limited number of terms in Equation (5.7) results in a potential situation that the objective function’s optimal solution might not cover all the latent truth estimates. The hyperthesis might not hold due to insufficient temporal information resulting in overfitting of the data, but the unwrapping used for generating the reference phase provides a spatial regularization that reduces the amount of overfitting. Fortunately, enough number of SLC acquisitions results in more formatted interferograms—which is commonplace nowadays to obtain sufficient SLCs for one site. Reference results are based on the unwrapped phase, and phase unwrapping is a step that incorporates spatial analysis, which is not the scope of this work. However, considering spatial analysis can reduce

the effect on limited temporal access. This understanding guides us to incorporate both spatial and temporal analysis for deformation and DEM error estimation in future work. Overall, based only on temporal analysis, IGS-CMAES delivers robust results on stacks with sufficient temporal observations. Its effectiveness in solving optimization tasks is well validated in our method by using both simulation and real data experiments.

5.2.6 Conclusions

Estimating ground deformation and DEM error with TSInSAR data is an ill-posed problem. In this work, we provided two main contributions: (1) designing a two-stage architecture suitable for interferometric phase processing and (2) introducing a benchmark hybrid simulation dataset by combining real-world baseline parameters and synthetic ground truth signals for an effective evaluation. It is known that, when the availability of exclusive time series data increases, it is necessary to design more efficient and effective algorithms. Typically, research in these fields is conducted by solving complex optimization problems, which are extremely computationally intensive and time-demanding. By considering parameter estimation as an optimization problem, we presented an exploration process to acquire sufficient global information that will guide us to the optimal solution using IGS coarse search. After that, a group of candidate solutions is passed to the second exploitation step, where the best estimation is obtained using an effective local CMAES refined search. The combined two-stage optimization delivers a high degree of accuracy and efficiency without being influenced by local extrema. Our method was evaluated using simulated and real data, and the result outperforms traditional local and global optimizers. Furthermore, IGS-CMAES offers the advantage of avoiding phase unwrapping, which is often time-consuming and prone to error. It also generalizes well to different real site baselines without retuning the model configurations. In conclusion, this study demonstrates that our proposed two-stage black-box optimization framework IGS-CMAES successfully addresses two research tasks concurrently: linear deformation estimation and DEM error correction with TSInSAR data. When sufficient temporal information is provided, investigations on real data demonstrate that IGS-CMAES achieves comparable performance to an industry-standard processing pipeline, which requires a phase unwrapping process. Further developments of this work will focus on the improvements by considering spatial information when dealing with limited temporal data.

5.3 ConvArcFit: Spatial-Temporal Optimization for Ground Deformation and DEM Error

5.3.1 Introduction

Our previous IGS-CMAES aligns with established literature by analyzing just temporal information and processing each pixel location independently. However, it is known to have some limitations: 1) Accuracy is primarily determined by temporal coherence. As a result, those techniques are often limited to persistent scatters (PS), which are time-coherent targets contained within a stack of N interferograms. They impose limitations on the ability to produce dense motion maps, which include more information than sparse estimation. 2) For other spatially correlated signals, such as atmospheric signals, the techniques must either use the double differences strategy on arcs or remove the atmospheric signal entirely as a preprocessing step. Each option involves additional computations and could introduce accumulated method errors, affecting the final deformation signal estimation’s accuracy. In this study, to address the constraints of temporal-only analysis, we present ConvArcFit, a new spatial-temporal optimization approach. It is a 3D convolutional optimization technique involving the use of a stack of 2D interferograms rather than PS points. Figure 5.9 summarizes the architecture of ConvArcFit. It is designed to estimate wrapped interferograms’ motion rates and DEM errors.

5.3.2 Definition of Optimization Problem

We consider an optimization problem on a 3D wrapped phase (stack of interferograms) in this work. We use ϕ with shape (H, W, N) to denote the wrapped input phase with spatial resolution $H \times W$ and N as the depth of the stack. Our target solutions are two signal maps - linear motion rate mr and DEM error he with same size as $H \times W$ for dense estimates at each ground location. The optimization is performed by finding the solutions that maximize the spatial coherence of phase residuals and temporal coherence of phase arc residuals. For each ground location (x, y) in interferogram k , we have input phase $\phi_t(x, y, k)$ and reconstruct phase $\phi_r(x, y, k)$ that can be calculated using estimated mr and he by Equation 5.6. Phase residual $\phi_\delta(x, y, k)$ is point difference between reconstruct phase and input phase as $\phi_r(x, y, k) - \phi_t(x, y, k)$. At the same time, arc $\phi_{Arc}(x, y, k)$ is the phase difference to a neighbourhood location (\hat{x}, \hat{y}) as $\phi(x, y, k) - \phi(\hat{x}, \hat{y}, k)$. Then, arc residual $\phi_{\delta, Arc}(x, y, k)$ are double difference phase arcs between input phase arc $\phi_{t, Arc}(x, y, k)$ and reconstruct phase arc $\phi_{r, Arc}(x, y, k)$. For an arbitrary discrete 3D patch with shape (P, P, N) (P is patch size and N is

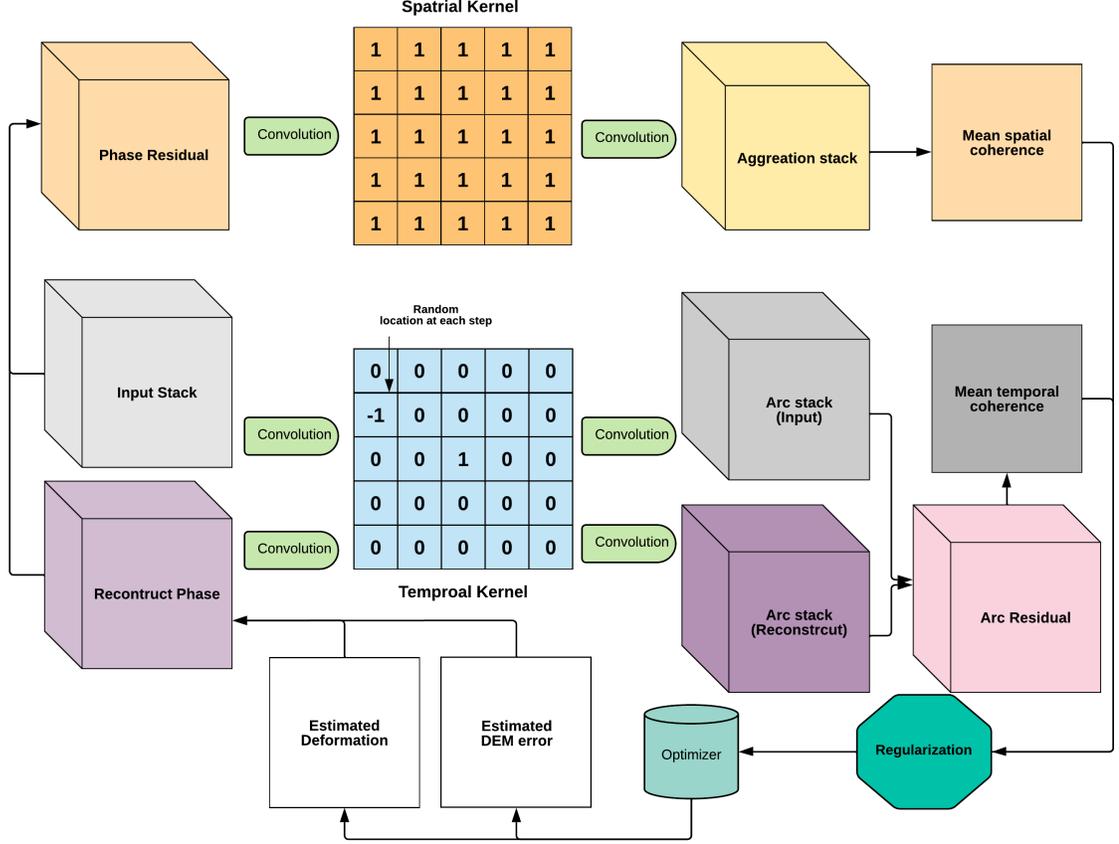


Figure 5.9: Proposed ConvArcFit optimization workflow.

stack depth), we have an objective function \mathcal{J} for minimizing optimization:

$$\mathcal{J}(x, y) = \mathcal{J}_{spatial}(x, y) + \mathcal{J}_{temporal}(x, y) \quad (5.17)$$

where, the spatial loss is the negative mean BoxCar coherent coefficient through the stack:

$$\mathcal{J}_{spatial}(x, y) = -\frac{1}{N} \sum_{k=1}^N \left(\frac{1}{P^2} \left| \sum_{\hat{x}=1}^P \sum_{\hat{y}=1}^P e^{j(\phi_{\delta}(\hat{x}, \hat{y}, k))} \right| \right) \quad (5.18)$$

and, temporal loss is negative residual phase arc temporal coherent coefficient:

$$\mathcal{J}_{temporal}(x, y) = -\frac{1}{N} \left| \sum_{k=1}^N e^{j(\phi_{\delta, Arc}(x, y, k))} \right| \quad (5.19)$$

Its contribution to spatial coherence can be considered to be one [20] due to the atmosphere's spatial smoothness. On the other hand, the atmospheric phase can also be eliminated using temporal coherence arc formulation [57]. As a result, the

estimated motion rate and DEM error approach the correct value, increasing spatial and temporal coherence. This property enables us to assess how well the estimations match the data without considering the effect of the atmosphere.

Design of Convolution Kernels

As shown in Figure 5.9, we define two convolution kernels - a spatial kernel and a temporal kernel to efficiently accomplish our proposed optimization task. The spatial kernel is a standard sum aggregation kernel that sums all the values in a window. The temporal kernel is a special dynamic kernel given that it only sets the center pixel at one and others at zero. For every iteration, one of the surrounding pixels is chosen at random to be -1. Both spatial and temporal kernels are used as separate convolutions to make a depth-wise computation. We obtain the spatial coherence tensor ($[H \times W \times N]$) by applying the spatial kernel to the phase residual ϕ_δ stack and simply calculating the mean of the entire stack (Equation 5.18). The temporal kernel is applied to obtain the arcs map ($[H \times W \times N]$) for both the input and ($\phi_{r,Arc}$) reconstruct phases ($\phi_{r,Arc}$). Two arcs stacks can be subtracted to obtain a double difference arcs stack ($\phi_{\delta,Arc}$), which can measure mean temporal coherence (Equation 5.19). These convolutional computations can take advantage of new deep learning frameworks to process our kernels convolutionally on GPUs.

Focal Regularization

It is well established that our solution space is discontinuous as a result of the wrapping phase. Additionally, as mentioned in Section 4.1.1, incoherent areas can retain noise signals even after filtering. As a result, simply optimizing \mathcal{J} still leaves the possibility of becoming stuck in local optima or shifting to other global optimas because of 2π phase jumps. Two regularization terms are proposed to overcome the issue. The first one is the total variation of the predicted 2D signal maps for both mr and he in order to minimize the image gradients' L1 norm:

$$\mathcal{R}_{tv}(y) = \sum_{i,j} (|y_{i+1,j} - y_{i,j}| + |y_{i,j+1} - y_{i,j}|) \quad (5.20)$$

It smooths the estimated signals (y_{mr}, y_{he}) and helps make the loss function more convex, making it easier to locate the solution when it is far away. Apart from that, both types of signals are also penalized by L1 norm regularization:

$$\mathcal{R}_{L1}(y) = \sum_{i,j} |y_{i,j}| \quad (5.21)$$

This term is primarily used to avoid a global phase shift. Because all phases are wrapped by 2π , shifting the global results by the multiplier of 2π results in the same value \mathcal{J} . The L1 term is applied to penalize the estimations based on their absolute distance from zero. Here, our updated objective function is as follows:

$$\mathcal{J} = \mathcal{J}_{spatial} + \mathcal{J}_{temporal} + \sigma(\mathcal{R}_{tv}(mr) + \mathcal{R}_{tv}(he)) + \gamma(\mathcal{R}_{L1}(mr) + \mathcal{R}_{L1}(he)) \quad (5.22)$$

where σ and γ are the regularization weights, which serve as the hyper-parameters for balancing the penalty algorithm. Properly setting the weights of σ and γ is critical because they affect the final outcome. A low weight results in negligible benefits when dealing with large-scale signals, while a high weight may result in biased estimation. In this study, we further examine an adaptive method inspired by the innovative focal loss [146] that has been widely employed in image classification tasks. Instead of altering the cross-entropy loss, we propose employing a focal modulating term to dynamically scale our regularizations in run time. We define two adaptive functions for automatically adjusting the regularization weights at the pixel level.

$$FL_{tv} = \alpha_{tv}^{max}(1 + \mathcal{J}_{spatial})^{\beta_{tv}} + \alpha_{tv}^{min} \quad (5.23)$$

$$FL_{L1} = \alpha_{L1}^{max}(-\mathcal{J}_{temporal})^{\beta_{L1}} + \alpha_{L1}^{min} \quad (5.24)$$

where for both regularization terms, β is the tunable focusing parameter as explained in [146]. The idea is that when the estimated outputs result in low spatial coherence for a given pixel location, FL_{tv} gradually increases the weight of the regularization term for total variation. This process assists the model in gaining more attention from neighbourhood estimates. Simultaneously, it lowers regularization effects to avoid biased estimates when spatial coherence is large. In the presence of an L1 regularization factor, the same manner is followed for temporal arc coherence. When the temporal arc is relatively coherent, a heavier weighted L1 regularization will be used to prevent the issue of global phase shifting. When the temporal arc is incoherent, FL_{L1} reduces L1 regularization to prevent unexpected penalties for large magnitude estimations. Since both $\mathcal{J}_{spatial}$ and $\mathcal{J}_{temporal}$ have the same spatial resolution as the original input, regularization is conducted adaptively at the pixel level based on how well the present results fit the input. Furthermore, two global minimum (α^{min}) and maximum (α^{max}) parameters are defined for each type of regularization to restrict the scaling factor's border. To this end, our final objective function is the following:

$$\mathcal{J} = \mathcal{J}_{spatial} + \mathcal{J}_{temporal} + FL_{tv}(\mathcal{R}_{tv}(mr) + \mathcal{R}_{tv}(he)) + FL_{L1}(\mathcal{R}_{L1}(mr) + \mathcal{R}_{L1}(he)) \quad (5.25)$$

5.3.3 Results and Future Work

We adopted a straightforward demonstration by running ConvArcFit on a single stack at two stages - before and after atmospheric removal. Results in Figure 5.10 demonstrate that the model is capable of producing consistent results in both scenarios. The model can differentiate linear motion rate and DEM error signal even when the atmospheric signals are spatially presented, because the designed objective function is insensitive to any spatially smooth signals.

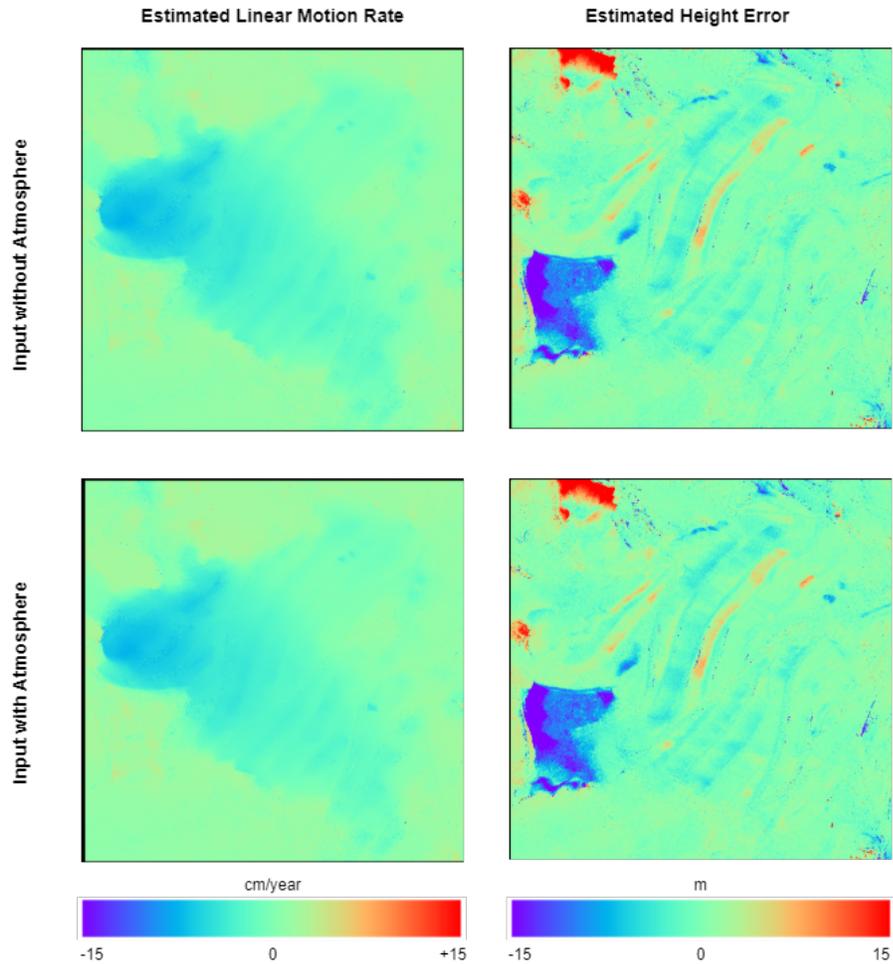


Figure 5.10: Results on a real world stack; the model shows consistent results with input data with and without atmospheric removal.

This approach has been operationalized in the industry since 2019 and has become a standard module in the company's processing pipeline. Our technical reports contain extensive experimental details.

Chapter 6

InSAR Simulator

6.1 Introduction

Despite decades of InSAR-related research, including InSAR phase filtering, coherence estimation, the deformation and DEM error signal extraction. There is a remaining challenge to precisely validate and eventually optimize the developed algorithms with no ground truth data, and a controlled environment is available [16]. To date, the most commonly adopted evaluation strategy is using a couple of synthetic interferometric phases with simple geometric shape fringes or simulating the radar signals according to the pre-given digital evaluation model (DEM) [38] [41]. Note that there is a lack of large-scale datasets and benchmarks for this research field. Large-scale data is important, especially if an application wants to take advantage of the recent learning-based approaches. It is known that the robustness of a trainable model’s learning capacity requires diversity in datasets for sufficient training and testing in terms of the types of signal features and noise characteristics [93]. We address this problem by introducing an ultimate InSAR simulator toolbox that provides highly configurable simulation strategies to generate a wide variety of phase fringes and coherence distributions. It allows researchers to customize the key configuration parameters and then adopt a mechanism to randomly generate In-SAR data with a collection of distinct synthetic features. We will first discuss a spatial only 2D simulator that generates random clean and noisy pairs used for evaluating non-stack InSAR filters and coherence estimators. It is worth mentioning that our 2D simulator has been adopted by state-of-the-art methods for extensive analysis of several conventional and deep-learning approaches. Next, we extend it to generate random motion and DEM error signals as well as spatial and temporal baselines to produce a stack of interferograms for time-series InSAR research. In the end, we also present the potential to use recent intelligent generative models to learn the real signal dis-

tribution by training it with the full processing pipelines’ outputs. Furthermore, an adversarial training framework is introduced to let the generator be able to create synthetic signals as realistic as possible. To this end, we hope our proposed simulator toolbox can help researchers validate and compare the strengths and weaknesses of different algorithms in a low-bias assessment environment and guide future studies.

6.2 Methods and Procedure

6.2.1 2D InSAR Simulator

SAR satellites produce single-look complex (SLC) images. The returning radar echo from the ground is represented as a complex number in each SLC pixel. The signal model and its degradation version can be characterized:

$$SLC_{clean} = A\cos(\theta) + jA\sin(\theta), SLC_{noisy} = A\cos(\theta) + v_r + j(A\sin(\theta) + v_i) \quad (6.1)$$

where A is clean amplitude and θ is clean phase. Noise components v_r and v_i are additive white Gaussian noise (AWGN) in real and imaginary channels respectively

2. The corresponding clean interferogram is calculated as:

$$I = (A_{SLC_1} \odot A_{SLC_2})e^{j(\theta_{SLC_2} - \theta_{SLC_1})} \quad (6.2)$$

As illustrated in Eq. 6.2, interferometric phase is the wrapped phase difference between two SLCs. Hence, we set one of SLC’s phase component to 0 and only generate synthetic signals on the other one. The key steps of our simulator are: 1) Generate SLC_1 as a $H \times W$ 2D complex image with 0 phase value and with amplitude value growing from r_{min} to r_{max} moving from the left-most column to the right-most side following a Rayleigh distribution. 2) Generate SLC_2 with the same resolution and amplitude value as SLC_1 , then add a set of signals "S" to the phase. 3) Add independent additive white Gaussian noise v with standard deviation σ to the real and imaginary channels of both SLC_1 and SLC_2 . 4) Generate ground truth clean and noisy interferogram as defined by Eq. 6.2. In the proposed simulator we provide the following types of signals with their fully configurable parameters: 1) Gaussian bubble: range of spatial scale and range of Gaussian amplitudes. 2) Ellipse: range of heights and range of radii for both ellipse’s first and second axis. 3) Polygon: range of heights and range of radii for the edges and the range of number of polygon edges. 4) Buildings: range of height, width and depth factors which are used to simulate the height of a building. 5) Amplitude stripes: the approximate thickness of the amplitude that is altered in a band region of the SLC. 6) Phase stripes: the

approximate thickness of the phase band in the interferometric phase. Furthermore, the simulator is designed to add signals in random locations, and there is an extra parameter n defining the number of such type of signals added to the SLC. Lastly, the ground truth coherence is calculated with Monto Carlo approximation by maximum-likelihood coherence metric:

$$\frac{\sum_i^N z_1^{(i)} z_2^{*(i)}}{\sqrt{\sum_i^N |z_1^{(i)}|^2 \sum_i^N |z_2^{(i)}|^2}} \quad (6.3)$$

where, z is sampled with amount of N (set to a large number to reduce bias) for each location (r, c) in the synthetic interferogram.

$$z = A_{slc}^{r,c} e^{j\theta} + v_r + jv_i, \text{ where } (v_r, v_i) \sim N(0, \sigma^2) \quad (6.4)$$

We adopt an empirical relationship between amplitude and coherence for a specific noise σ to convert the amplitude to coherence for each point.

Results

After fixing the random seed as well as the configuration, we generated 1800 samples with 1000x1000 resolution as a benchmark dataset. Visualization of selected patches' simulated phase (under different configurations) of clean and noisy interferograms as well as the coherence are shown in Figure 6.1

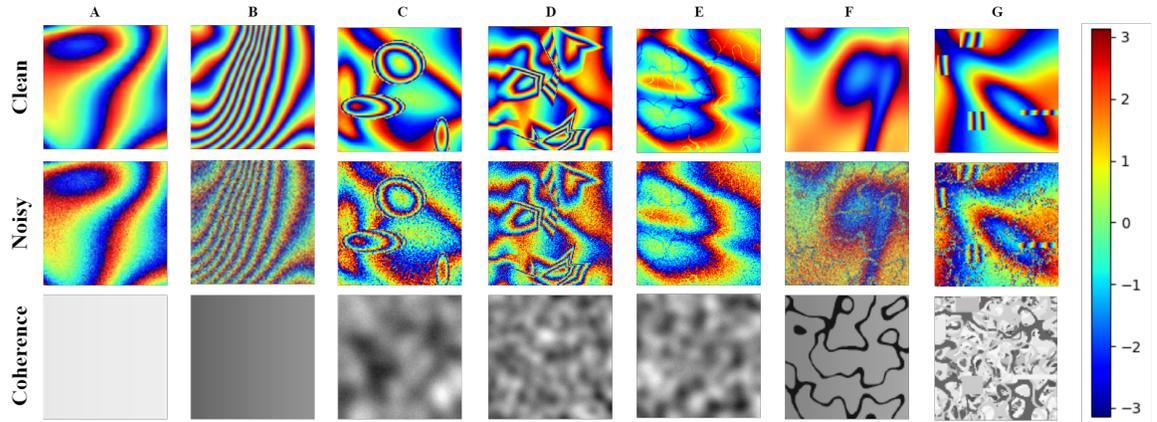


Figure 6.1: Column A: low base noise, sparse fringes; B: median base noise, dense fringes; C: ellipse signals; D: polygon signals; E: phase bands; F: low amplitude bands; G: plenty of amplitude bands with building blocks.

Our proposed benchmark dataset has been adopted to evaluate widely used by our own and other researchers [103–105, 109, 117].

6.2.2 3D InSAR Simulator

To address time-series (TS)InSAR research, we further developed a 3D extension of our previous 2D simulator. Adapting the 2D simulator for temporal simulation requires taking into account InSAR temporal baselines. Thus, rather than generating noise-free synthetic signals, we propose directly generating synthetic linear motion rate and DEM height error signals. In this section, we consider only the phase components that are used in TSInSAR techniques. We propose to use the same simulation strategy to generate 2D signals representing synthetic motion and DEM errors (mr and he). Furthermore, we designed it to generate handcrafted Gaussian Pyramids for improved objective evaluation. As shown in the first row of Figure 6.2. This is for the purpose of validating and comparing different algorithms in a controlled environment, as well as determining how well the model can handle a variety of signal scales and possible combinations of two signal types. In the example image, motion rates are sampled from $[-26, +26]$ cm/year, and height errors are sampled from $[-120, +120]$ m. Both types of signals were generated on different spatial scales from top to bottom, with the highest amplitude value in the center. Because the values of the motion rate and height error Gaussian bubble signals are sampled in the opposite direction from the center to the outer (from low to high or from high to low), a comprehensive combination of the two signals is presented. Following that, we generate synthetic interferograms using the Equation 6.5.

$$\phi = conv_m \cdot \Delta days \cdot mr + conv_h \cdot B_{\perp} \cdot he \quad (6.5)$$

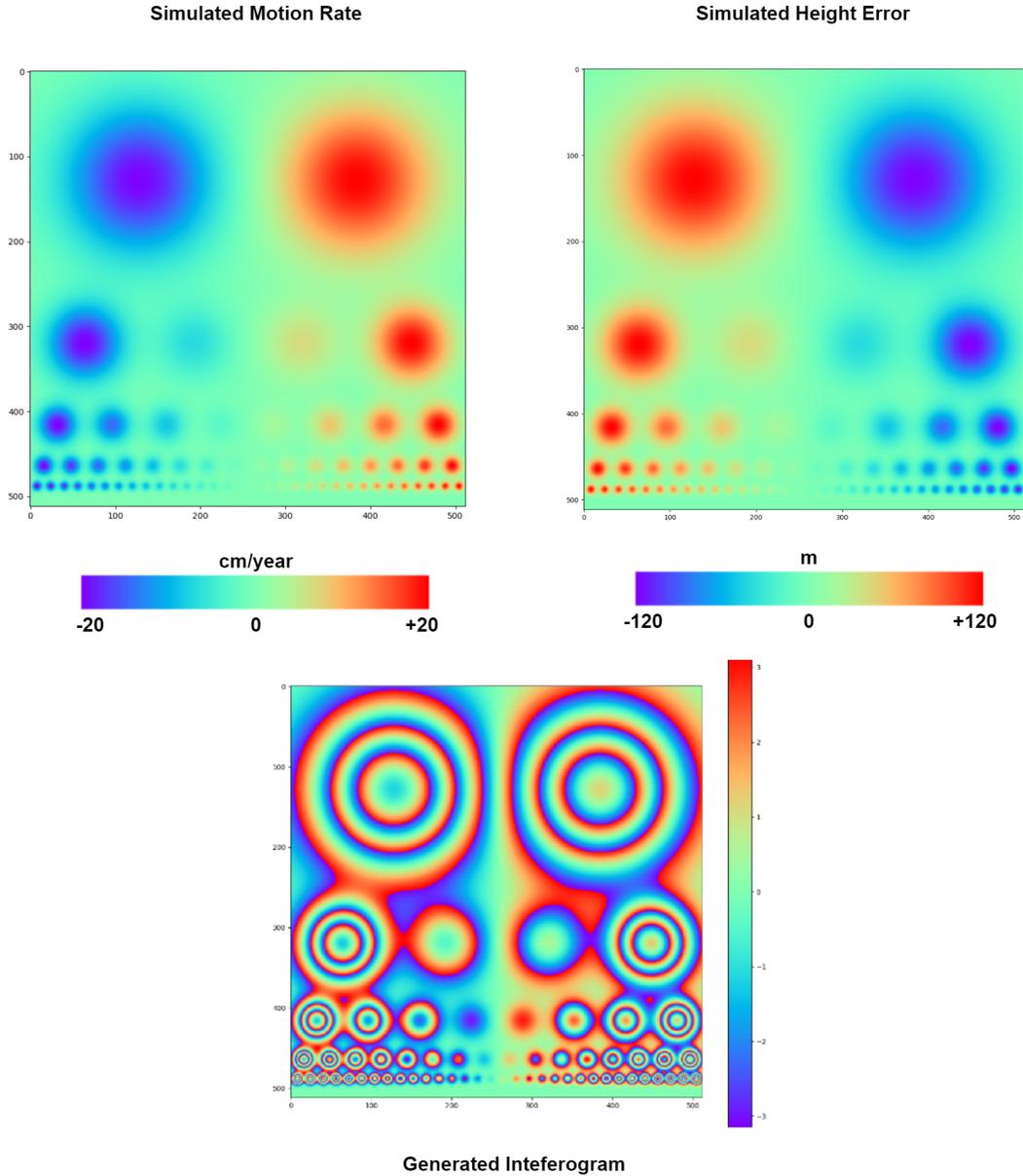


Figure 6.2: Sample outputs from 3D Simulator by creating Gaussian Pyramids signals pairs.

where, mr and he are 2D signal maps with size $H \times W$, and $\Delta days$ and B_{\perp} are 1D vector whose length equals the depth of the stack N , $conv_m$ and $conv_h$ are two constant values (recall Equations 5.3 and 5.4). Because these baseline values are satellite parameters that can be randomly sampled according to their real distribution or use real-world satellite baseline parameters. The simulation outputs are a stack of interferograms with a shape of $H \times W$ and a depth of N . For a quick demonstration, we chose one real site baseline parameter and generated a full stack of synthetic

interferograms, as shown in the second row of Figure 6.5. This can easily be improved to: 1) generate additional signal shapes and stochastic signals as our 2D simulator to augment the synthetic features; and 2) generate a simulation benchmark dataset to evaluate our signal decomposition algorithm and make quantitative comparisons to other methods.

6.2.3 Intelligent InSAR Simulator - Learning from real world signals

To generate realistic signal patterns, we continue to study the possibility of developing an intelligent generative model similar to the one we investigated in our natural image denoising work. We have obtained many fully processed deformation and DEM error signals from the industry in the past. These delivered outputs can be considered an adequate assessment of the real-world signals' ground truth. Numerous works based on Generative Adversarial Networks (GANs) have demonstrated promising outcomes in the generation of synthetic natural images [147][71][77]. This work demonstrates how to learn from real-world InSAR data using a baseline GAN built of a ResNet-based Generator and a Patch Discriminator. Figure 6.3 provides an overview of our baseline model.

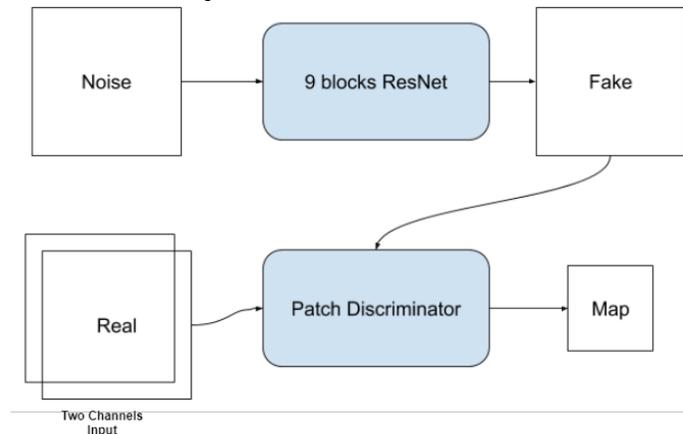


Figure 6.3: The proposed baseline GAN architecture. The generator’s input is the noise map. Its outputs are a two-channel signal map (concatenation of synthetic linear deformation rate and DEM error signals) with the same spatial resolution. The discriminator is learned to tell if given two-channel signals from real distribution.

In our baseline model, our generator is a nine-layer ResNet [148] that takes noise input with shape $[H, W, 1]$ and generates synthetic 2D signal pairs of motion rate and DEM error as a two-channel tensor with shape $[H, W, 2]$. Additionally, we utilize the patch discriminator, which was introduced by PatchGAN [149]. It penalizes structure

only at the scale of local image patches. Instead of classifying the entire image, the PatchGAN discriminator attempts to classify whether each $N \times H$ patch in an image is real or fake. It scans the image constitutionally and averages all predictions to get the final discrimination result. Assuming pixels are independent when separated by more than a patch area effectively represents the image as a Markov random field [149]. The key reason for employing Patch Discriminator is that we believe the spatial feature qualities are also essential for the signal’s structure information. Because the contours of the motion and height error signals may change depending on the topography, the textures of the signal maps often provide more value.

Results

We train the model using randomly extracted 256x256 image patches from night pairs of the real-world signal map with a deformation rate of 1000x1000 and a DEM height error of 1000x1000. Each training iteration includes 18 256x256 patches as real data training samples and the corresponding noise samples from a uniform distribution ranging from -1 to 1. The generator learns to map random noise to synthetic motion rate and DEM error signal pairs by using adversarial training. A discriminator can be trained to determine whether or not a particular signal sample is real. Following training, the generator should generate synthetic signals that are indistinguishable from real signals. Figure 6.5 shows a real training pair sample of deformation and DEM height error signals. After 1e6 training iterations, the sample output of generator is shown in Figure 6.4. Our baseline model produces signals that are visually similar to those observed in real-world data.

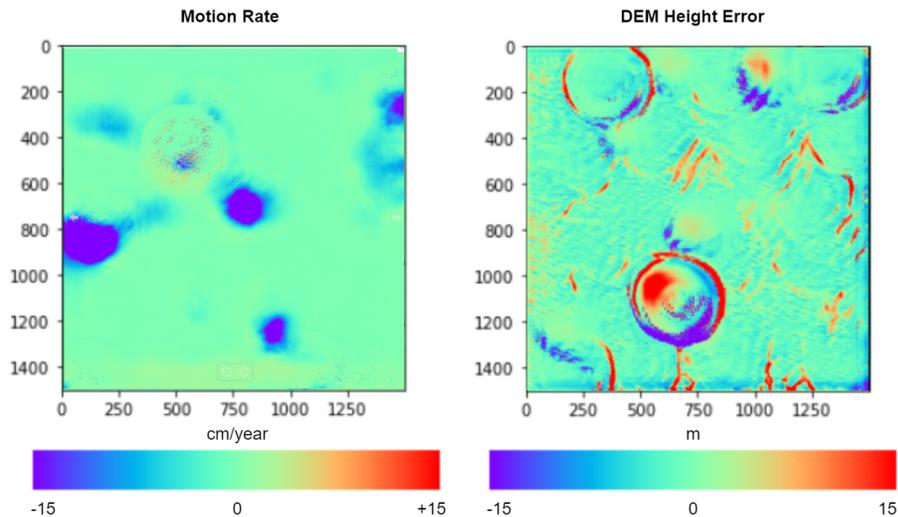


Figure 6.4: Sample outputs of baseline GAN.

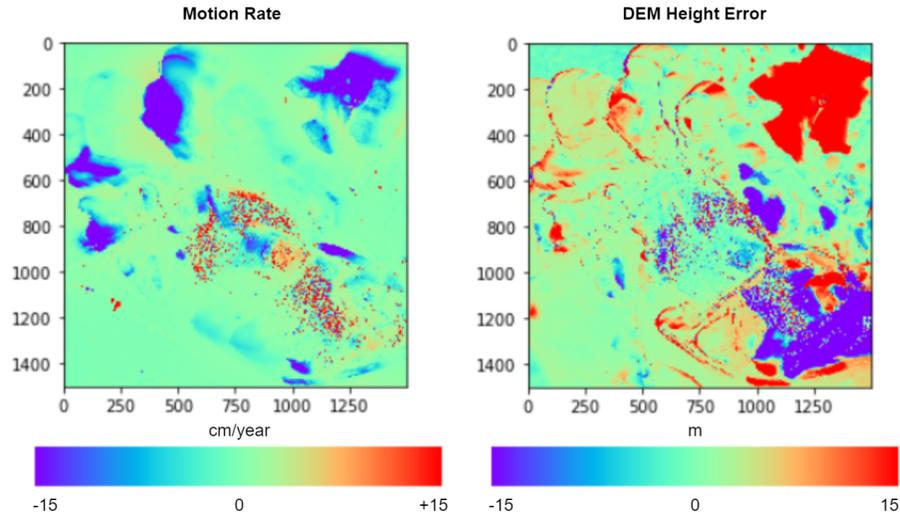


Figure 6.5: Sample pair of real world motion rate and DEM height error signal maps processed by full processing pipeline.

6.2.4 Conclusion

Our stochastic InSAR simulator is capable of generating distinctive kinds of signals based on specified configurations and combining them to produce an almost infinite number of combinations. It results in various irregular fringes in the interferometric phase and offers a number of controls for controlling the complexity of the generated phase as well as different levels of distortion. To the best of our knowledge, this InSAR simulator is the first study that represents the large-scale performance evaluation benchmark for InSAR filtering and coherence estimation. The source code for the InSAR-Simulator is available at <https://github.com/Lucklyric/InSAR-Simulator>. By introducing our simulator and benchmark, we hope to assist researchers in comparing the strengths and shortcomings of existing or new algorithms and provide guidance for developing InSAR phase filtering and coherence estimation. Additionally, we demonstrate the concept of employing a recent intelligent deep generative model to learn the real signal distribution for deep modeling synthetic signals. I am now leading the team that continues to work in this direction; our technical report and [150] include the most recent quantitative and qualitative analysis contributions.

Chapter 7

Conclusion and Future Directions

In this thesis, we present numerous methods for decomposing observed InSAR data into distinct types of signals.

7.1 Summary

We begin by addressing the topic of spatially independent noise components in natural images. We present a novel deep learning model based on Gaussian denoising of natural images and adapt it for use with the InSAR modality technique. In the absence of ground truth data for a real-world scenario, we established a teacher-student framework - DeepInSAR, for supervised training. The design allows the use of a conventional stack-based filtering approach as the teacher (which requires over 30 observations) and a deep differentiable model to acknowledge the behaviour of teacher method. Following training, the student model can produce results that are equivalent to or better than those provided by the teacher method when only one pair of observations is used. Additionally, the proposed model is designed to provide a coherence map which reflects the pixel-level signal quality in conjunction with filtering.

To circumvent the requirement for clean reference data in DeepInSAR, we presented GenInSAR, a self-supervised method for training the model to predict the pixel distribution using information about surrounding pixels. While preliminary findings seem positive, they are subject to some significant limitations: 1) Because the model is trained on center-masked patches, it ignores information about the center pixel during training, and 2) GenInSAR's coherence estimate lacks appropriate resolution and contrast. As a result, we created SRDInSAR, which overcomes each of GenInSAR's shortcomings and provides a more flexible framework for modeling and inference efficiency. In addition, SRDInSAR surpasses GenInSAR and other men-

tioned techniques in simulation and real-world data experiments, proving its readiness for industrial deployment.

It is widely established that time-series InSAR (TSInSAR) data can be used to estimate slow and subtle terrain displacements after noise suppression. However, when exclusive time series data availability increase, it is necessary to develop more efficient and effective algorithms. Typically, research in these fields is conducted through challenging optimization problems, which require a significant amount of processing power and time. We present a two-stage black-box optimization approach for predicting the average ground deformation rate and the terrain digital elevation model (DEM) error simultaneously. The approach begins with an iterative grid search (IGS) to generate coarse candidate solutions, refined using a covariance matrix adaptive evolution strategy (CMAES). Comparing our work quantitatively and qualitatively with a range of optimizers demonstrates our work’s stability and effectiveness. The suggested IGS-CMAES method outperforms previously published algorithms in terms of accuracy while requiring much fewer objective function evaluations. It provides broad-area surveillance in scenarios that need high precision and real-time processing. To strengthen usability and performance, ConvArcFit, a novel spatial-temporal optimization technique, is then presented to overcome the constraints of temporal-only processing. This technique optimizes 3D convolutional signals in which a stack of 2D interferograms replaces the PS points. ConvArcFit’s results reveal that it can effectively extract dense deformation and DEM error signals on the wrapped phase. Additionally, it operates on signals that do not require atmospheric phase reduction, saving processing time.

Despite these signal decomposition procedures, accurately evaluating and ultimately enhancing the resulting algorithms remains a challenge due to a lack of sufficient ground truth data in real-world scenarios. To address this issue, we provide a stochastic InSAR simulator that enables the production of a wide variety of phase fringes and coherence distributions through a highly flexible simulation framework. This simulator is well-suited for conducting exhaustive quantitative studies of various filtering and coherence estimation algorithms. In 2D and 3D modes, the simulator enables both stack and non-stack analysis. The 3D version assesses signal separation methods by simulating time-series deformation signals. Additionally, we evaluate a newly built adversarial generative model for learning the distribution of real-world deformation signals and their correlations to the DEM error to reproduce realistic signals.

7.2 Knowledge Translation and Commercialization

Apart from academic research, our ultimate goal is to apply our results to a real-world InSAR processing pipeline in order to tackle real-world challenges.

By collaborating with the Research and Development (R & D) group at 3vGeomatics, we set up a solid procedure for algorithm validation and operationalization. The amount of data generated by real-world InSAR can exceed ten gigabytes per site, depending on the type of radar and its resolution. We always begin with a subset of data to conduct proof-of-concept and validate our methods. Later, we must adapt the proposed algorithms to integrate with 3vGeomatics' distributed map-reduce processing architecture in order to handle massive special scale stacks with resolutions greater than 10K x 10K. DeepInSAR has been developed into an in-house processing module after months of training on all accessible data at the company. Its outputs are verified and evaluated by InSAR operators before becoming a standard processing module for their clients. ConvArcFit has also been operationalized in 3vGeomatics in a simplified version for deformation and DEM error signal separation, ConvFit, does not leverage Arc coherence to save computations and already provides industry-leading results.

The commercialized versions of DeepInSAR and ConvArcFit have been operationalized to process PB-level data annually to speed up wide-area monitoring applications worldwide. Additionally, our current InSAR data simulator has been adopted and cited by many researchers and publications to benefit them in properly evaluating their algorithms.

7.3 Limitations and Future Work

My entire thesis is devoted to various proposed methods and their improved versions for continually overcoming limitations. We believe that there are always research gaps that can be explored to improve the methodology.

Due to the supervised training setup used in our DeepInSAR work, the output of the teacher technique is critical. While DeepInSAR has been proven and deployed in a real-world InSAR processing chain, it may be of limited benefit to academics or organizations that lack the resources to collect reference data. To this end, we proposed SRDInSAR as a follow-up to GenInSAR. SRDInSAR requires only a noisy version for self-supervised model training. The current SRDInSAR is undergoing an industry migration process for large-scale validation. The findings from DeepInSAR also show a potential direction for future work by expanding the SRDInSAR to incor-

porate amplitude information from paired SAR images, not just the interferometric phase.

By examining just temporal data and processing each pixel location independently, our suggested IGS-CMAES aligns with the existing literature. It is well-known to have various downsides, including the following: 1) Accuracy is heavily influenced by temporal coherence, and it is only effective for sparse results. 2) Stacks containing additional spatially correlated signals, such as atmospheric signals, are not supported.

ConvArcFit is then presented to address these issues by integrating spatial and temporal analyses. However, ConvArcFit requires considerable computations due to the online convolutional optimization, which is why its simplified form, ConvFit, is used for operations first. Additionally, our current research focuses exclusively on linear deformation modeling, which is not always applicable to areas with non-linear motions. Further investigation may be necessary to determine whether a distribution model can be utilized to approximate the non-linear motion rate.

Our stochastic InSAR simulator can be utilized to simulate a wide variety of phase fringes and coherence using a highly customizable simulation framework with hand-crafted features. To mimic real-world signals, we proposed a path employing an intelligent deep generative model to learn the distributions of real-world signals. However, our current training data is derived from the company’s whole processing pipeline outputs, which are not perfect due to missing data, noise, and method flaws. However, building a learning-based generative model with distorted data distribution remains an open research question to advance our intelligent InSAR simulator even more. This direction is an active, ongoing topic that our research group is pursuing.

Bibliography

- [1] A Hooper, P. Segall, and H Zebker, “Persistent scatterer insar for crustal deformation analysis, with application to volcán alcedo, galápagos,” *Journal of Geophysical Research*, vol. 112, no. B07407, p. 19, 2007.
- [2] X. Mao, C. Shen, and Y.-B. Yang, “Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections,” in *Advances in neural information processing systems*, 2016, pp. 2802–2810.
- [3] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, IEEE, vol. 2, 2005, pp. 60–65.
- [4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering,” *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [5] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Transactions on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006.
- [6] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online dictionary learning for sparse coding,” in *Proceedings of the 26th annual international conference on machine learning*, ACM, 2009, pp. 689–696.
- [7] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.
- [8] Y. Weiss and W. T. Freeman, “What makes a good model of natural images?” In *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*, IEEE, 2007, pp. 1–8.
- [9] H. C. Burger, C. J. Schuler, and S. Harmeling, “Image denoising: Can plain neural networks compete with bm3d?” In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, pp. 2392–2399.
- [10] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

- [11] P. Chatterjee and P. Milanfar, “Clustering-based denoising with locally learned dictionaries,” *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1438–1451, 2009.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [13] C. Szegedy *et al.*, “Going deeper with convolutions,” *Cvpr*, 2015.
- [14] M. Seymour and I. Cumming, “Maximum likelihood estimation for sar interferometry,” in *Geoscience and Remote Sensing Symposium, 1994. IGARSS’94. Surface and Atmospheric Remote Sensing: Technologies, Data Analysis and Interpretation., International*, IEEE, vol. 4, 1994, pp. 2272–2275.
- [15] J.-S. Lee, K. P. Papathanassiou, T. L. Ainsworth, M. R. Grunes, and A. Reigber, “A new technique for noise filtering of sar interferometric phase images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 5, pp. 1456–1465, 1998.
- [16] C.-F. Chao, K.-S. Chen, and J.-S. Lee, “Refined filtering of interferometric phase from insar data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 12, pp. 5315–5323, 2013.
- [17] G. Ferraiuolo and G. Poggi, “A bayesian filtering technique for sar interferometric phase fields,” *IEEE Transactions on image processing*, vol. 13, no. 10, pp. 1368–1378, 2004.
- [18] G. Vasile, E. Trouvé, J.-S. Lee, and V. Buzuloiu, “Intensity-driven adaptive-neighborhood technique for polarimetric and interferometric sar parameters estimation,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 6, pp. 1609–1621, 2006.
- [19] Q. Yu, X. Yang, S. Fu, X. Liu, and X. Sun, “An adaptive contoured window filter for interferometric synthetic aperture radar,” *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 1, pp. 23–26, 2007.
- [20] Y. Wang, H. Huang, Z. Dong, and M. Wu, “Modified patch-based locally optimal wiener method for interferometric sar phase filtering,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 114, pp. 10–23, 2016.
- [21] F. Baselice, G. Ferraioli, V. Pascazio, and G. Schirinzi, “Joint insar dem and deformation estimation in a bayesian framework,” in *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, IEEE, 2014, pp. 398–401.
- [22] R. M. Goldstein and C. L. Werner, “Radar interferogram filtering for geophysical applications,” *Geophysical research letters*, vol. 25, no. 21, pp. 4035–4038, 1998.
- [23] I. Baran, M. Stewart, and P. Lilly, “A modification to the goldstein radar interferogram filter,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 9, pp. 2114–2118, 2003.

- [24] R. Song, H. Guo, G. Liu, Z. Perski, and J. Fan, "Improved goldstein sar interferogram filter based on empirical mode decomposition," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 2, pp. 399–403, 2014.
- [25] M. Jiang *et al.*, "The improvement for baran phase filter derived from unbiased insar coherence," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 7, pp. 3002–3010, 2014.
- [26] Q. Wang, H. Huang, A. Yu, and Z. Dong, "An efficient and adaptive approach for noise filtering of sar interferometric phase images," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 6, pp. 1140–1144, 2011.
- [27] B. Cai, D. Liang, and Z. Dong, "A new adaptive multiresolution noise-filtering approach for sar interferometric phase images," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 2, pp. 266–270, 2008.
- [28] C. Lopez-Martinez and X. Fabregas, "Modeling and reduction of sar interferometric phase noise in the wavelet domain," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 12, pp. 2553–2566, 2002.
- [29] X. Zha, R. Fu, Z. Dai, and B. Liu, "Noise reduction in interferograms using the wavelet packet transform and wiener filtering," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 3, pp. 404–408, 2008.
- [30] Y. Bian and B. Mercer, "Interferometric sar phase filtering in the wavelet domain using simultaneous detection and estimation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 4, pp. 1396–1416, 2011.
- [31] G. Xu, M.-D. Xing, X.-G. Xia, L. Zhang, Y.-Y. Liu, and Z. Bao, "Sparse regularization of interferometric phase and amplitude for insar image formation based on bayesian representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 4, pp. 2123–2136, 2015.
- [32] C.-A. Deledalle, L. Denis, and F. Tupin, "Iterative weighted maximum likelihood denoising with probabilistic patch-based weights," *IEEE Transactions on Image Processing*, vol. 18, no. 12, p. 2661, 2009.
- [33] S. Parrilli, M. Poderico, C. V. Angelino, and L. Verdoliva, "A nonlocal sar image denoising algorithm based on lmmse wavelet shrinkage," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 2, pp. 606–616, 2012.
- [34] D. Cozzolino, S. Parrilli, G. Scarpa, G. Poggi, and L. Verdoliva, "Fast adaptive nonlocal sar despeckling," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 2, pp. 524–528, 2014.
- [35] C.-A. Deledalle, L. Denis, and F. Tupin, "Nl-insar: Nonlocal interferogram estimation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 4, pp. 1441–1452, 2011.
- [36] R. Chen, W. Yu, R. Wang, G. Liu, and Y. Shao, "Interferometric phase denoising by pyramid nonlocal means filter," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 4, pp. 826–830, 2013.

- [37] X. X. Zhu, R. Bamler, M. Lachaise, F. Adam, Y. Shi, and M. Eineder, “Improving tandem-x dems by non-local insar filtering,” in *EUSAR 2014; 10th European Conference on Synthetic Aperture Radar; Proceedings of*, VDE, 2014, pp. 1–4.
- [38] F. Sica, D. Cozzolino, X. X. Zhu, L. Verdoliva, and G. Poggi, “Insar-bm3d: A nonlocal filter for sar interferometric phase restoration,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 6, pp. 3456–3467, 2018.
- [39] C.-A. Deledalle, L. Denis, F. Tupin, A. Reigber, and M. Jäger, “Nl-sar: A unified nonlocal framework for resolution-preserving (pol)(in) sar denoising,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 4, pp. 2021–2038, 2015.
- [40] X. Su, C.-A. Deledalle, F. Tupin, and H. Sun, “Two-step multitemporal non-local means for synthetic aperture radar images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 10, pp. 6181–6196, 2014.
- [41] F. Sica, D. Reale, G. Poggi, L. Verdoliva, and G. Fornaro, “Nonlocal adaptive multilooking in sar multipass differential interferometry,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 4, pp. 1727–1742, 2015.
- [42] X. Lin, F. Li, D. Meng, D. Hu, and C. Ding, “Nonlocal sar interferometric phase filtering through higher order singular value decomposition,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 4, pp. 806–810, 2015.
- [43] Y. Guo, Z. Sun, R. Qu, L. Jiao, F. Liu, and X. Zhang, “Fuzzy superpixels based semi-supervised similarity-constrained cnn for polsar image classification,” *Remote Sensing*, vol. 12, no. 10, p. 1694, 2020.
- [44] F. Ma, F. Gao, J. Sun, H. Zhou, and A. Hussain, “Attention graph convolution network for image segmentation in big sar imagery data,” *Remote Sensing*, vol. 11, no. 21, p. 2586, 2019.
- [45] M. Krestenitis, G. Orfanidis, K. Ioannidis, K. Avgerinakis, S. Vrochidis, and I. Kompatsiaris, “Oil spill identification from satellite images using deep neural networks,” *Remote Sensing*, vol. 11, no. 15, p. 1762, 2019.
- [46] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *ECCV*, 2018.
- [47] Z. Cui, C. Tang, Z. Cao, and N. Liu, “D-atr for sar images based on deep neural networks,” *Remote Sensing*, vol. 11, no. 8, p. 906, 2019.
- [48] N. Anantrasirichai, J. Biggs, F. Albino, and D. Bull, “A deep learning approach to detecting volcano deformation from satellite imagery using synthetic datasets,” *Remote Sensing of Environment*, vol. 230, p. 111 179, 2019.
- [49] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *European conference on computer vision*, Springer, 2016, pp. 630–645.

- [50] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *ICML 2015*, 2015.
- [51] J. J. Sousa, A. J. Hooper, R. F. Hanssen, L. C. Bastos, and A. M. Ruiz, “Persistent scatterer insar: A comparison of methodologies based on a model of temporal deformation vs. spatial correlation selection criteria,” *Remote Sensing of Environment*, vol. 115, no. 10, pp. 2652–2663, 2011.
- [52] S. Usai, “The use of man-made features for long time scale insar,” in *IGARSS’97. 1997 IEEE International Geoscience and Remote Sensing Symposium Proceedings. Remote Sensing-A Scientific Vision for Sustainable Development*, IEEE, vol. 4, 1997, pp. 1542–1544.
- [53] A. Hooper, “A multi-temporal insar method incorporating both persistent scatterer and small baseline approaches,” *Geophysical Research Letters*, vol. 35, no. 16, 2008.
- [54] T. R. Lauknes, H. A. Zebker, and Y. Larsen, “Insar deformation time series using an $L_{\{1\}}$ -norm small-baseline approach,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 1, pp. 536–546, 2010.
- [55] F. Gini, F. Lombardini, and M. Montanari, “Layover solution in multibaseline sar interferometry,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 4, pp. 1344–1356, 2002.
- [56] T. Reza, A. Zimmer, J. M. D. Blasco, P. Ghuman, T. K. Aasawat, and M. Ripeanu, “Accelerating persistent scatterer pixel selection for insar processing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 1, pp. 16–30, 2017.
- [57] W. Duan, H. Zhang, and C. Wang, “Deformation estimation for time series insar using simulated annealing algorithm,” *Sensors*, vol. 19, no. 1, p. 115, 2019.
- [58] N. Anantrasirichai *et al.*, “Detecting ground deformation in the built environment using sparse satellite insar data with a convolutional neural network,” *IEEE Transactions on Geoscience and Remote Sensing*, 2020.
- [59] A. Ferretti, C. Prati, and F. Rocca, “Nonlinear subsidence rate estimation using permanent scatterers in differential sar interferometry,” *IEEE Transactions on geoscience and remote sensing*, vol. 38, no. 5, pp. 2202–2212, 2000.
- [60] A. Ferretti, C. Prati, and F. Rocca, “Permanent scatterers in sar interferometry,” *IEEE Transactions on geoscience and remote sensing*, vol. 39, no. 1, pp. 8–20, 2001.
- [61] C. Werner, U. Wegmuller, T. Strozzi, and A. Wiesmann, “Interferometric point target analysis for deformation mapping,” in *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No. 03CH37477)*, IEEE, vol. 7, 2003, pp. 4362–4364.
- [62] M. K. Bert, “Radar interferometry: Persistent scatterers technique,” *The Netherlands: Springer*, 2006.

- [63] M. Costantini, S. Falco, F. Malvarosa, and F. Minati, “A new method for identification and analysis of persistent scatterers in series of sar images,” in *IGARSS 2008-2008 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, vol. 2, 2008, pp. II-449.
- [64] M. Costantini, S. Falco, F. Malvarosa, F. Minati, F. Trillo, and F. Vecchioli, “Persistent scatterer pair interferometry: Approach and application to cosmospymed sar data,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 7, pp. 2869–2879, 2014.
- [65] P. Berardino, G. Fornaro, R. Lanari, and E. Sansosti, “A new algorithm for surface deformation monitoring based on small baseline differential sar interferograms,” *IEEE Transactions on geoscience and remote sensing*, vol. 40, no. 11, pp. 2375–2383, 2002.
- [66] H. Fattahi and F. Amelung, “Dem error correction in insar time series,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 7, pp. 4249–4259, 2013.
- [67] L. Zhang, X. Ding, and Z. Lu, “Modeling psinsar time series without phase unwrapping,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 1, pp. 547–556, 2010.
- [68] A Peltier, M Bianchi, E Kaminski, J.-C. Komorowski, A. Rucci, and T Staudacher, “Psinsar as a new tool to monitor pre-eruptive volcano ground deformation: Validation using gps measurements on piton de la fournaise,” *Geophysical Research Letters*, vol. 37, no. 12, 2010.
- [69] C. Patrascu, A. A. Popescu, and M. Datcu, “Sbas and ps measurement fusion for enhancing displacement measurements,” in *2012 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2012, pp. 3947–3950.
- [70] A. Hooper, H. Zebker, P. Segall, and B. Kampes, “A new method for measuring deformation on volcanoes and other natural terrains using insar persistent scatterers,” *Geophysical research letters*, vol. 31, no. 23, 2004.
- [71] I. Goodfellow *et al.*, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [72] D. Warde-Farley and Y. Bengio, “Improving generative adversarial networks with denoising feature matching,” 2016.
- [73] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, 2015, pp. 448–456.
- [74] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feed-forward neural networks,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.
- [75] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

- [76] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5769–5779.
- [77] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *International Conference on Machine Learning*, 2017, pp. 214–223.
- [78] M. Abadi *et al.*, “Tensorflow: A system for large-scale machine learning.,” in *OSDI*, vol. 16, 2016, pp. 265–283.
- [79] D. Zoran and Y. Weiss, “From learning models of natural image patches to whole image restoration,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, 2011, pp. 479–486.
- [80] W. Dong, L. Zhang, G. Shi, and X. Li, “Nonlocally centralized sparse representation for image restoration,” *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1620–1630, 2013.
- [81] F. Chen, L. Zhang, and H. Yu, “External patch prior guided internal clustering for image denoising,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 603–611.
- [82] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng, “Patch group based nonlocal self-similarity prior learning for image denoising,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 244–252.
- [83] S. Gu, L. Zhang, W. Zuo, and X. Feng, “Weighted nuclear norm minimization with application to image denoising,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2862–2869.
- [84] A. Levin and B. Nadler, “Natural image denoising: Optimality and inherent bounds,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, IEEE, 2011, pp. 2833–2840.
- [85] R. F. Hanssen, *Radar interferometry: data interpretation and error analysis*. Springer Science & Business Media, 2001, vol. 2.
- [86] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks.,” in *CVPR*, vol. 1, 2017, p. 3.
- [87] R. Bamler and P. Hartl, “Synthetic aperture radar interferometry,” *Inverse problems*, vol. 14, no. 4, R1, 1998.
- [88] R. E. Shiffler, “Maximum z scores and outliers,” *The American Statistician*, vol. 42, no. 1, pp. 79–80, 1988.
- [89] Z. Sun and C. Han, “Heavy-tailed rayleigh distribution: A new tool for the modeling of sar amplitude images,” in *IGARSS 2008-2008 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, vol. 4, 2008, pp. IV–1253.
- [90] B. Iglewicz and D. C. Hoaglin, *How to detect and handle outliers*. Asq Press, 1993, vol. 16.

- [91] W. Pitz and D. Miller, “The terrasars-x satellite,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 2, pp. 615–622, 2010.
- [92] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315–323.
- [93] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning.,” in *AAAI*, vol. 4, 2017, p. 12.
- [94] R. Timofte, V. De Smet, and L. Van Gool, “A+: Adjusted anchored neighborhood regression for fast super-resolution,” in *Asian Conference on Computer Vision*, Springer, 2014, pp. 111–126.
- [95] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Highway networks,” *arXiv preprint arXiv:1505.00387*, 2015.
- [96] T. Reza, A. Zimmer, J. M. D. Blasco, P. Ghuman, T. K. Aasawat, and M. Ripeanu, “Accelerating persistent scatterer pixel selection for insar processing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 1, pp. 16–30, 2018.
- [97] T. Reza, A. Zimmer, P. Ghuman, T. k. Aasawat, and M. Ripeanu, “Accelerating persistent scatterer pixel selection for insar processing,” in *2015 IEEE 26th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, 2015, pp. 49–56. DOI: 10.1109/ASAP.2015.7245704.
- [98] T. F. Y. Vicente, L. Hou, C.-P. Yu, M. Hoai, and D. Samaras, “Large-scale training of shadow detectors with noisily-annotated shadow examples,” in *European Conference on Computer Vision*, Springer, 2016, pp. 816–832.
- [99] J. W. Goodman, *Speckle phenomena in optics: theory and applications*. Roberts and Company Publishers, 2007.
- [100] A. Zimmer and P. Ghuman, “Cuda optimization of non-local means extended to wrapped gaussian distributions for interferometric phase denoising,” *Procedia Computer Science*, vol. 80, pp. 166–177, 2016.
- [101] X. X. Zhu, G. Baier, M. Lachaise, Y. Shi, F. Adam, and R. Bamler, “Potential and limits of non-local means insar filtering for tandem-x high-resolution dem generation,” *Remote Sensing of Environment*, vol. 218, pp. 148–161, 2018.
- [102] Z. Wang and A. C. Bovik, “Mean squared error: Love it or leave it? a new look at signal fidelity measures,” *IEEE signal processing magazine*, vol. 26, no. 1, pp. 98–117, 2009.
- [103] X. Sun, A. Zimmer, S. Mukherjee, N. K. Kottayil, P. Ghuman, and I. Cheng, “Deepinsar—a deep learning framework for sar interferometric phase restoration and coherence estimation,” *Remote Sensing*, vol. 12, no. 14, p. 2340, 2020.
- [104] Y. Chen, L. Bruzzone, L. Jiang, and Q. Sun, “Aru-net: Reduction of atmospheric phase screen in sar interferometry using attention-based deep residual u-net,” *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

- [105] L. Pu, X. Zhang, Z. Zhou, J. Shi, S. Wei, and Y. Zhou, “A phase filtering method with scale recurrent networks for insar,” *Remote Sensing*, vol. 12, no. 20, p. 3453, 2020.
- [106] Q. Zhang, T. Wang, Y. Pei, and X. Shi, “Selective kernel res-attention unet: Deep learning for generating decorrelation mask with applications to tandem-x interferograms,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 8537–8551, 2021.
- [107] J. Lehtinen *et al.*, “Noise2noise: Learning image restoration without clean data,” *arXiv preprint arXiv:1803.04189*, 2018.
- [108] A. Krull, T.-O. Buchholz, and F. Jug, “Noise2void-learning denoising from single noisy images,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2129–2137.
- [109] S. Mukherjee, A. Zimmer, X. Sun, P. Ghuman, and I. Cheng, “An unsupervised generative neural approach for insar phase filtering and coherence estimation,” *IEEE Geoscience and Remote Sensing Letters*, 2020.
- [110] C. M. Bishop, “Mixture density networks,” 1994.
- [111] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.
- [112] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer, “Densenet: Implementing efficient convnet descriptor pyramids,” *arXiv preprint arXiv:1404.1869*, 2014.
- [113] A. Shapiro, “Monte carlo sampling methods,” *Handbooks in operations research and management science*, vol. 10, pp. 353–425, 2003.
- [114] X. Sun, N. K. Kottayil, S. Mukherjee, and I. Cheng, “Adversarial training for dual-stage image denoising enhanced with feature matching,” in *International Conference on Smart Multimedia*, Springer, 2018, pp. 357–366.
- [115] Z. Yanjie and V. Prinet, “Insar coherence estimation,” in *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, vol. 5, 2004, pp. 3353–3355.
- [116] Z. Suo, J. Zhang, M. Li, Q. Zhang, and C. Fang, “Improved insar phase noise filter in frequency domain,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 2, pp. 1185–1195, 2015.
- [117] S. Mukherjee, A. Zimmer, N. K. Kottayil, X. Sun, P. Ghuman, and I. Cheng, “Cnn-based insar denoising and coherence metric,” in *2018 IEEE SENSORS*, IEEE, 2018, pp. 1–4.
- [118] F. Hu and J. Wu, “Improvement of the multi-temporal insar method using reliable arc solutions,” *International journal of remote sensing*, vol. 39, no. 10, pp. 3363–3385, 2018.

- [119] D. Bekaert, R. Walters, T. Wright, A. Hooper, and D. Parker, “Statistical comparison of insar tropospheric correction techniques,” *Remote Sensing of Environment*, vol. 170, pp. 40–47, 2015.
- [120] H. A. Zebker, J. Villasenor, *et al.*, “Decorrelation in interferometric radar echoes,” *IEEE Transactions on geoscience and remote sensing*, vol. 30, no. 5, pp. 950–959, 1992.
- [121] C. Yu, Z. Li, and N. T. Penna, “Interferometric synthetic aperture radar atmospheric correction using a gps-based iterative tropospheric decomposition model,” *Remote Sensing of Environment*, vol. 204, pp. 109–121, 2018.
- [122] C. Yu, Z. Li, N. T. Penna, and P. Crippa, “Generic atmospheric correction model for interferometric synthetic aperture radar observations,” *Journal of Geophysical Research: Solid Earth*, vol. 123, no. 10, pp. 9202–9222, 2018.
- [123] M. Costantini, F. Minati, F. Trillo, and F. Vecchioli, “Enhanced psp sar interferometry for analysis of weak scatterers and high definition monitoring of deformations over structures and natural terrains,” in *2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS*, IEEE, 2013, pp. 876–879.
- [124] J. J. Sousa *et al.*, “Ps-insar processing methodologies in the detection of field surface deformation—study of the granada basin (central betic cordilleras, southern spain),” *Journal of Geodynamics*, vol. 49, no. 3-4, pp. 181–189, 2010.
- [125] B. Kampes and S. Usai, “Doris: The delft object-oriented radar interferometric software,” in *Proc. 2nd Int. Symp. Operationalization of Remote Sensing*, Citeseer, 1999, pp. 16–20.
- [126] M. Schlögl, B. Widhalm, and M. Avian, “Comprehensive time-series analysis of bridge deformation using differential satellite radar interferometry based on sentinel-1,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 172, pp. 132–146, 2021.
- [127] R. Torres *et al.*, “Gmes sentinel-1 mission,” *Remote Sensing of Environment*, vol. 120, pp. 9–24, 2012.
- [128] S Azadnejad, Y. Maghsoudi, and D. Perissin, “Evaluation of polarimetric capabilities of dual polarized sentinel-1 and terrasars-x data to improve the psinsar algorithm using amplitude dispersion index optimization,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 84, p. 101950, 2020.
- [129] H.-L. Liu, F. Gu, and Q. Zhang, “Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems,” *IEEE transactions on evolutionary computation*, vol. 18, no. 3, pp. 450–455, 2013.
- [130] M. Wu, K. Li, S. Kwong, Q. Zhang, and J. Zhang, “Learning to decompose: A paradigm for decomposition-based multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 376–390, 2018.

- [131] D Holden, S Donegan, and A Pon, “Brumadinho dam insar study: Analysis of terrasar-x, cosmo-skymed and sentinel-1 images preceding the collapse,” in *Proceedings of the 2020 International Symposium on Slope Stability in Open Pit Mining and Civil Engineering*, Australian Centre for Geomechanics, 2020, pp. 293–306.
- [132] N. K. Kottayil, A. Zimmer, S. Mukherjee, X. Sun, P. Ghuman, and I. Cheng, “Accurate pixel-based noise estimation for insar interferograms,” in *2018 IEEE SENSORS*, IEEE, 2018, pp. 1–4.
- [133] N. Hansen, S. D. Müller, and P. Koumoutsakos, “Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es),” *Evolutionary computation*, vol. 11, no. 1, pp. 1–18, 2003.
- [134] I. Loshchilov, “Cma-es with restarts for solving cec 2013 benchmark problems,” in *2013 IEEE Congress on Evolutionary Computation*, Ieee, 2013, pp. 369–376.
- [135] N. Hansen, “Benchmarking a bi-population cma-es on the bbob-2009 function testbed,” in *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, ACM, 2009, pp. 2389–2396.
- [136] N Hansen, *The cma evolution strategy: A comparing review towards a new evolutionary computation vol 192 ed l jose a et al*, 2006.
- [137] I. Loshchilov, M. Schoenauer, and M. Sebag, “Alternative restart strategies for cma-es,” in *International Conference on Parallel Problem Solving from Nature*, Springer, 2012, pp. 296–305.
- [138] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, “Evolution strategies as a scalable alternative to reinforcement learning,” *arXiv preprint arXiv:1703.03864*, 2017.
- [139] N. Hansen and S. Kern, “Evaluating the cma evolution strategy on multimodal test functions,” in *International Conference on Parallel Problem Solving from Nature*, Springer, 2004, pp. 282–291.
- [140] N. Hansen, “Tutorial: Covariance matrix adaptation (cma) evolution strategy,” *Institute of Computational Science, ETH Zurich*, 2006.
- [141] D. Cusson, K. Trischuk, D. Hébert, G. Hewus, M. Gara, and P. Ghuman, “Satellite-based insar monitoring of highway bridges: Validation case study on the north channel bridge in ontario, canada,” *Transportation Research Record*, vol. 2672, no. 45, pp. 76–86, 2018.
- [142] F. Gao and L. Han, “Implementing the nelder-mead simplex algorithm with adaptive parameters,” *Computational Optimization and Applications*, vol. 51, no. 1, pp. 259–277, 2012.
- [143] J. Nocedal and S. J. Wright, “Sequential quadratic programming,” *Numerical optimization*, pp. 529–562, 2006.

- [144] D.-H. Li and M. Fukushima, “A modified bfgs method and its global convergence in nonconvex minimization,” *Journal of Computational and Applied Mathematics*, vol. 129, no. 1-2, pp. 15–35, 2001.
- [145] Y. Xiang, S. Gubian, B. Suomela, and J. Hoeng, “Generalized simulated annealing for global optimization: The gensa package.,” *R Journal*, vol. 5, no. 1, 2013.
- [146] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
- [147] C. Ledig *et al.*, “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.
- [148] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” *arXiv preprint arXiv:1602.07261*, 2016.
- [149] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.
- [150] Z. Zhou, X. Sun, Z. Wang, P. Ghuman, and I. Cheng, “Deep generative modeling for insar signals,” *Remote Sensing (prior to submission)*,
- [151] A. Ferretti, A. Fumagalli, F. Novali, C. Prati, F. Rocca, and A. Rucci, “A new algorithm for processing interferometric data-stacks: Squeesar,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 9, pp. 3460–3470, 2011.
- [152] A. Parizzi and R. Brcic, “Adaptive insar stack multilooking exploiting amplitude statistics: A comparison between different techniques and practical results,” *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 3, pp. 441–445, 2011.
- [153] M. Schmitt and U. Stilla, “Adaptive multilooking of airborne single-pass multi-baseline insar stacks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 305–312, 2014.
- [154] A. Pepe, Y. Yang, M. Manzo, and R. Lanari, “Improved emcf-sbas processing chain based on advanced techniques for the noise-filtering and selection of small baseline multi-look dinsar interferograms,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 8, pp. 4394–4417, 2015.
- [155] A. Buades, B. Coll, and J.-M. Morel, “A review of image denoising algorithms, with a new one,” *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.
- [156] D. Rolnick, A. Veit, S. Belongie, and N. Shavit, “Deep learning is robust to massive label noise,” *arXiv preprint arXiv:1705.10694*, 2017.

- [157] L. Zhang, X. Ding, Z. Lu, H.-S. Jung, J. Hu, and G. Feng, “A novel multitemporal insar model for joint estimation of deformation rates and orbital errors,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 6, pp. 3529–3540, 2013.
- [158] Z. Lu and D. Dzurisin, “Insar imaging of aleutian volcanoes,” in *InSAR imaging of Aleutian volcanoes*, Springer, 2014, pp. 87–345.
- [159] P. Blanco-Sanchez, J. J. Mallorquí, S. Duque, and D. Monells, “The coherent pixels technique (cpt): An advanced dinsar technique for nonlinear deformation monitoring,” in *Earth Sciences and Mathematics*, Springer, 2008, pp. 1167–1193.
- [160] M. F. Møller, *A scaled conjugate gradient algorithm for fast supervised learning*. Aarhus University, Computer Science Department, 1990.
- [161] A. McCardle, J. McCardle, and F Ramos, “Large scale deformation monitoring and atmospheric removal in mexico city,” in *Fringe 2009 Workshop, Frascati, Italy*, 2009.
- [162] X. Sun, A. Zimmer, S. Mukherjee, P. Ghuman, and I. Cheng, “Igs-cmaes: A two-stage optimization for ground deformation and dem error estimation in time series insar data,” *Remote Sensing*, vol. 13, no. 13, p. 2615, 2021.