Physics-based Data-driven Inference

by

Dhruv Patel

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(Mechanical Engineering)

May 2021

To the creator of the universe

and
my fabulous family

Acknowledgements

First and foremost, I thank Almighty Supreme Lord Shri Swaminarayan for blessing my life with endless joy, happiness, and fulfillment and for creating this beautiful world and giving me sense, strength, and directions to *infer* the mysteries of it.

Next, I would like to express my sincere gratitude to one of the most influential people in my professional life, my advisor, Professor Assad Oberai. He pushed me to grow as a researcher, an independent thinker, a better writer, and a speaker and had an infinite amount of patience with my endless research queries and delays. He provided a perfect research environment and mentorship I could have ever hoped for to explore and execute new research ideas and directions. His kind and caring nature has deeply influenced me and I believe our time together has made me not only a better researcher, but also a better human being and I am privileged to consider him as my guru, a mentor, a guide, and a friend.

I would also like to thank my doctoral committee members Professor Roger Ghanem, Professor Aiichiro Nakano, and Professor Satyandra Gupta for their time serving on my committee. Professor Ghanem's "Uncertainty Quantification" course was one of the highlights of my time at USC. My conversations with him during and after the class and his insightful comments and views about the field have tangibly shaped my research direction. I also really enjoyed the "Scientific Computing and Visualization" course of Professor Nakano and his passion for research and teaching has inspired me. Professor Gupta's feedback on my research during my qualifier exam was really helpful. I would also like to thank my qualifier committee members Professor Paul Newton and Dr. Joseph Lim for their valuable feedback.

I am grateful to have had the opportunity to work with great collaborators. Dr. Ravi Bonam and his team at IBM, Albany center were extremely helpful and provided us valuable

data. Dr. Bonam taught me many new things about lithography and was quick to help troubleshoot any problem. Equally fruitful was my collaboration with Dr. Vinay Duddalwar and his entire team at Keck Medicine, USC They were extremely kind and generous with their time and research help.

I am also grateful to IBM, NIH, and Army Research Office for providing financial support for the different parts of my Ph.D.

And of course, none of this would have been possible without the unwavering support and joyful companionship of other members of our research group. Starting with the very first week of joining the group I was touched by the kind and welcoming environment of group members starting with Mohit Tyagi, who stayed till midnight (despite his travel early next morning for his next job) to help me understand the code workflow to make sure that I have a smooth transition and start of the project before he leaves. Li Dong played (and is still playing) a perfect role of second mentor inside the group for me and I am forever indebted to him for his practical advice and life teachings about both technical as well as non-technical stuff as my graduate school life would not have been same without him. It was my absolute pleasure sharing office space at RPI with Nicholas Hugenberg, Justin Clough, and Yu Zhang. They all helped me keep sane among the chaos of grad school. Nicholas provided a good sounding-board for ideas and has helped me a lot with setting up the inversion code. Justin is my go-to man anytime I need any help with Linux or Vim or high-performance computing and his companionship had made my transfer to USC a lot smoother. I will always remember my late-night stays and fun chats in the office with Yu. Outside the office, Anirban Chandra has always been a constant pillar of support and I will always remember our never-ending conversations about graduate school, academia, and life in general. Moving to folks at USC, sharing the office with Harishankar Ramaswamy, Ragheb Raad, and Orazio Pinti has been an equally pleasing experience. Hari has been a great source to bounce back interesting ideas and insights. Ragheb and Orazio has always been willing to lend a helping hand and answer any questions. It was also my great pleasure

to share my graduate school life journey with wonderful postdoctoral scholars of the group: Dawei Song, Iman Asareh, and Deep Ray. I can recall many long and interesting discussions with Dawei often starting with some (seemingly) trivial question. Iman always provides interesting perspectives and I will remember our long conversations ranging from cuisines to Indian movies. Despite not being in the same location, Deep has been a great source of help and sounding board to bounce back ideas just a call away; brainstorming and working with him on different research problems has been a true pleasure and his systematic thinking and attention to detail are infectious. It is my privilege to call all these fine folks my friends.

It would be remiss of me if I do not thank my lovely friends outside the group, Sasidhar Potukuchi, Harshil Patel, Maulik Kotecha, and many more. They helped me keep things in perspective and provided a supporting hand in times of crisis.

Finally, I would like to express my utmost gratitude and appreciation to my family. My parents, Vasudevbhai Patel and Dharmishthaben Patel, inspired me to dream big and instilled in me the virtues to achieve them. Without their relentless support and sacrifices, I would not have reached where I am today. I also thank my elder brother Vivek Patel and sister-in-law Kiran Patel for always being there on my side during the ups and downs of my life and for their unconditional love. Special thanks go to two little champs Divy and Akshar for providing me much needed morale booster from time to time! Finally, my sincere gratitude and thanks to my both grandmas Ratan Ba and Madhu Ba, who played a huge role in my childhood. I feel deeply sorry for not being there on your bedside during your last moments and I sincerely hope I could live the rest of my life fulfilling your dreams.

Table of Contents

Dedication ii

Acknowledgements iii

List of Tables ix

List of Figures x

Abstract xv

Chapter 1: Introduction 1
- 1.1 Inverse Problems . . . 2
  - 1.1.1 Regularization methods . . . 3
  - 1.1.2 Bayesian inversion . . . 5
- 1.2 Machine Learning . . . 7
  - 1.2.1 Supervised v/s Unsupervised inversion . . . 8
    - 1.2.1.1 Supervised inversion . . . 8
    - 1.2.1.2 Unsupervised inversion . . . 8
  - 1.2.2 Deep Generative Models (DGM) . . . 9
- 1.3 Contents and Contributions of This Thesis . . . 10

Chapter 2: Efficient Solution Techniques for Bayesian Inverse Problems 13
- 2.1 Introduction . . . 13
- 2.2 Ill-posedness of Inverse Problems . . . 14
  - 2.2.1 Deterministic approach . . . 15
  - 2.2.2 Statistical approach . . . 15
    - 2.2.2.1 Prior modeling . . . 17
    - 2.2.2.2 Posterior characterization: . . . 19
- 2.3 Generative Adversarial Networks . . . 21
- 2.4 GAN as Prior in Bayesian inference . . . 24
- 2.5 Numerical Results . . . 27
  - 2.5.1 Inverse heat conduction: Inferring thermal conductivity . . . 27
    - 2.5.1.1 Rectangular dataset . . . 28
    - 2.5.1.2 MNIST . . . 31
    - 2.5.1.3 Bi-phase material microstructure . . . 32
  - 2.5.2 Elasticity imaging . . . 36

Chapter 3: Quantifying Uncertainty in Supervised Learning **39**
3.1 Introduction . . . 39
3.2 Problem Formulation . . . 43
3.2.1 Generative Adversarial Networks . . . 44
3.2.2 Quantifying Uncertainty in a Supervised Learning Task . . . 48
3.3 Experiments . . . 51
3.3.1 Image classification . . . 52
3.3.2 Image inpainting . . . 56
3.3.3 Inference problems in computational physics . . . 60
3.3.3.1 Forward Problem . . . 63
3.3.3.2 Inverse Problem . . . 64
3.3.3.3 Mixed Problem . . . 65
3.4 Expression for the Maximum A-Posteriori (MAP) Estimate . . . 66
3.5 Additional Results . . . 69
3.5.1 MNIST . . . 69
3.5.2 CelebA . . . 71
3.5.3 Physics-driven inference . . . 71
3.6 Architecture and Training Details . . . 71
3.7 Conclusions . . . 72

Chapter 4: Inferring Visco-Elastic Properties from Interior Time Harmonic Data **78**
4.1 Introduction . . . 78
4.2 Magnetic Resonance Elastography . . . 79
4.3 Problem Formulation . . . 80
4.3.1 Strong formulation . . . 80
4.3.2 Weak formulation . . . 82
4.4 CASE formulation . . . 83
4.4.1 Forward problem . . . 83
4.4.2 Inverse problem . . . 83
4.4.3 Iterative inversion . . . 86
4.5 Numerical Results . . . 86
4.5.1 Domain Decomposition . . . 89

Chapter 5: Circumventing the Solution of Inverse Problems in Mechanics through Deep Learning **92**
5.1 Introduction . . . 92
5.2 Computational Methods . . . 97
5.2.1 Workflow . . . 97
5.2.2 Generation of training and testing data sets . . . 100
5.2.3 Convolutional neural networkworkworks . . . 102
5.3 Results and Analysis . . . 104
5.3.1 Training and performance . . . 104
5.3.2 Analysis of convolution layers . . . 107
5.3.3 Performance on real data . . . 112

5.3.4 Comparison of ML-based and inverse problem-based approaches . . . 115
5.3.4.1 ML-based approach . . . 116
5.3.4.2 IP-based approach . . . 118
5.3.4.3 Hybrid approach . . . 119
5.4 Conclusions . . . 119

Chapter 6: Conclusions and Future Work **125**
6.1 Conclusions . . . 125
6.2 Future Work . . . 126

References **129**

List of Tables

2.1 Reconstruction results of elastography study 37

3.1 Comparison of different hybrid models. Arrows indicate which direction is better. Also a "-" indicates that the values were not reported in the orginal reference. 55

3.2 Hyper-parameters for WGAN-GP model 72

5.1 Performance of the CNN at different levels of noise (heterogeneity study). . . 105

5.2 Performance of the CNN at different levels of Gaussian noise (nonlinearity study). 107

5.3 Confusion matrix for physics based transfer learning 114

5.4 Wall-clock time for solving the nonlinear elasticity classification problem on a AMD Phenom II, 6 core processor. 117

List of Figures

1.1 A deep generative model is a function $\boldsymbol{g_\theta}(\boldsymbol{z})$ that takes a low-dimensional random vector $\boldsymbol{z} \in \mathbb{R}^k$ and produces a high-dimensional sample $\boldsymbol{g_\theta}(\boldsymbol{z}) \in \mathbb{R}^n$. The function is parameterized by a neural network with parameters $\boldsymbol{\theta}$. In the example shown in the figure, the generator $\boldsymbol{g} : \mathbb{R} \mapsto \mathbb{R}^2$ learns to map low-dimensional samples $\boldsymbol{z}$ (dark blue dots) drawn from a uniform distribution (light blue line), such that the distribution of $\boldsymbol{g}(\boldsymbol{z})$ (red dots) resembles the distribution of training samples (blue dots). 9

2.1 *Left:* Four typical samples of permeability in subsurface of earth. *Right:* Four typical samples of bi-phase material microstructure. Representing qualitative information of such prior samples into a quantitative form useful for prior characterization is non-trivial. 17

2.2 Schematics of Generative Adversarial Network 22

2.3 Parametric description of rectangular dataset 28

2.4 *Left*: Samples from true prior density (rectangular training dataset). *Right*: Samples from the learned prior density (from the trained WGAN) 29

2.5 Inferring thermal conductivity from noisy temperature measurement for rectangular dataset: Column (1) true conductivity field (2) temperature field (3) noisy version of temperature used as a measurement (4-6) MAP, mean, and pixel-wise standard deviation estimates of GAN-prior. 30

2.6 *Left panel:* Four representative samples from prior density of scaled MNIST conductivity field. *Right:* Four representative samples from the learned prior density (from the trained WGAN) 32

2.7 Inferring thermal conductivity from noisy temperature measurement for scaled MNIST conductivity dataset: Column (1) true conductivity field (2) temperature field (3) noisy version of temperature used as a measurement (4-6) MAP, mean, and pixel-wise standard deviation estimates of GAN-prior. 33

2.8 *Left panel:* Four representative samples from prior density of bi-phase material microstructure dataset. *Right:* Four representative samples from the learned prior density (from the trained WGAN) 35

2.9 Inferring thermal conductivity from noisy temperature measurement for bi-phase material microstructure: Column (1) true conductivity field (2) temperature field (3) noisy version of temperature used as a measurement (4-6) MAP, mean, and pixel-wise standard deviation estimates of GAN-prior. . . 36

2.10 Recovery of shear modulus field from noisy measurements of the displacement field. 38

3.1 Histogram of $\|\hat{\boldsymbol{x}} - \boldsymbol{x}^{\text{map}}\|$, a measure for out-of-distribution (OOD) data detection, on classification experiments on MNIST. The proposed method is able to successfully distinguish in-distribution (MNIST) and OOD (NotMNIST) test inputs. A large value of this parameter is a warning to the end user to disregard the classification results. 54

3.2 Histogram of $\|\text{var}(\boldsymbol{y})\|$ for MNIST dataset. 55

3.3 Variation of MNIST test set accuracy with likelihood variance. 56

3.4 Corner plot of prior (shown in blue contours) and posterior (shown in red contour) distributions for a GAN with the latent space dimension of 10. Posterior distribution corresponds to an MNIST digit classification task. 57

3.5 (a) Evolution of Wasserstein distance (between the true data distribution and the distribution defined by the samples generated by GAN) during training for different latent space dimensions (b) Effect of latent space dimension on prior approximation capability - Wasserstein distance (between the true data distribution and the distribution defined by the samples generated by GAN) at the end of training for different latent space dimension. 58

3.6 Estimate of the MAP, mean and pixel-wise variance from noisy occluded images using the proposed method. Note that the variance is peaked at the occluded region. 59

3.7 Estimate of the MAP (2nd row) and pixel-wise variance (3rd row) from the limited view of a noisy image (1st row) using the proposed method for image inpainting with a prior trained on MNIST images. An active learning strategy where subsequent measurement windows are selected at each iteration based on maximum value of variance (indicated by red rectangle). An accurate reconstruction of the original image is obtained with just 4 measurement windows. 59

3.8 Average reconstruction error (with 95% confidence interval) as a function of number of windows for variance-driven (adaptive) and random sampling strategies. 60

3.9 CelebA dataset: Estimate of the MAP (3rd row) , mean (4th row) and variance (5th row) from the limited view of a noisy image (2nd row) of a true image (1st row) using the variance-driven active learning strategy. 61

3.10 Realizations from the joint distribution of permeability (top row) and pressure field (bottom row). Left panel: samples from true joint density. Right panel: samples from the density learned by the WGAN model. 62

3.11 Forward propagation of uncertainty: Estimation of MAP (third column), mean (fifth column), and standard deviation (sixth column) of pressure field from the measurement of permeability field (second column). Two panels (top and bottom) show results for two different permeability measurements from the test set. 64

3.12 Inverse uncertainty quantification: Estimation of MAP (third column), mean (fourth column), and standard deviation (fifth column) of the permeability and pressure fields from the measurement of pressure field (second column). Two panels (top and bottom) show results for two different pressure measurements from the test set. 65

3.13 Inference with limited number of sparse measurements of pressure and permeability: From the top to bottom each panel corresponds to the case where measurements are made at 1%, 5%, 10%, and 20% of the total nodal locations. First column represents the true permeability and pressure fields. Second column shows the measured pressure and permeability fields. Third column shows the MAP estimate, and the fourth column shows difference between the MAP estimate and the ground truth. The last two columns represent the mean and the standard deviation. 67

3.14 L_1 norm of the standard deviation for permeability (left) and pressure (right) fields (shown in the last column of figure 3.13) as a function of the percentage of nodal locations where measurements were made. These plots were generated by considering ten different samples from the test set. Error bars indicate one standard deviation variation across ten different samples at each measurement level. 68

3.15 MNIST dataset: Estimate of the MAP (3rd row), mean (4th row) and variance (5th row) from the limited view of a noisy image (2nd row) using the proposed method. The window to be revealed at a given iteration (shown in red box) is selected using a variance-driven strategy. Top row indicates ground truth. For all digits measurement noise variance = 1. 70

3.16 Estimate of the MAP (2nd row), mean (3rd row) and variance (4th row) from a noisy image (1st row) using the proposed method. Note that all variance images are plotted on the same color scale and it highlights increasing level of uncertainty as more and more portion of an image is occluded. 71

3.17 CelebA dataset: Estimate of the y^{map} (3rd row), y^{mean} (4th row) and variance of y (5th row) from the limited view of a noisy image (2nd row) using the proposed adaptive method. The window to be revealed at a given iteration (shown in red box) is selected using a variance-driven strategy. Top row indicates ground truth. For all images measurement noise variance = 1. . . . 74

3.18 Inference with limited number of sparse measurements of pressure and permeability: From the top to bottom each panel corresponds to the case where measurements are made at 1%, 5%, 10%, and 20% of the total nodal locations. First column represents the true permeability and pressure fields. Second column shows the measured pressure and permeability fields. Third column shows the MAP estimate, and the fourth column shows difference between the MAP estimate and the ground truth. The last two columns represent the mean and the standard deviation. 75

3.19 Generator and discriminator architecture used for image classification task for both MNIST and NotMNIST datasaet. 76

3.20 Generator and discriminator architectures for (a) MNIST dataset and (b) CelebA dataset used in image inpainting and active learning tasks. Note that the same architecture as CelabA was used for the physics-driven inference problems described in section 3.3.3 except instead of three channels at the output of the generator and the input of the discriminator two channels were used (corresponding to permeability and pressure field) 77

4.1 Reconstruction results with elliptical inclusion. *Left panel:* real (top row) and imaginary (bottom row) components of shear modulus distribution for ground truth. *Middle panel:* real and imaginary components of displacement field corresponding to ground truth shear modulus shown on the left. *Right panel:* Real (top row) and imaginary (bottom) components of reconstructed shear modulus. 88

4.2 Reconstruction results from noisy measured displacement. *Left panel:* real (top row) and imaginary (bottom row) components of shear modulus distribution - ground truth. *Middle panel:* noisy measured/observed displacement field with 3% additive Gaussian noise. *Right panel:* Real (top row) and imaginary (bottom) components of reconstructed shear modulus. 88

4.3 L-curve for reconstruction results with 3% additive Gaussian noise. Note that optimal regularization parameter value (10^{-9}) is indicated by asterisk. 89

4.4 Domain Decomposition: Inverse problem in each sub-domain is solved by different processor in parallel. 90

5.1 Standard workflow for diagnosis based on ultrasound elastography. 93

5.2 Deep-learning based workflow for classifying malignant lesions. 95

5.3 Two typical shear modulus distributions (μ) for benign and malignant classes. 101

5.4 Two typical displacement distribution images for benign and malignant classes.101

5.5 Two typical nonlinear modulus (γ) distributions for benign and malignant classes. 103

5.6 Two typical "difference in displacement" images for benign and malignant classes.103

5.7 CNN architecture for the shear modulus heterogeneity study. 103

5.8 CNN architecture for the nonlinearity study. 106

5.9 Test accuracy at different noise levels for heterogeneity study. 106

5.10 Test accuracy at different noise levels for nonlinearity study. 106

5.11 Learned weights (top row) and corresponding Fourier transform (bottom row) of active convolution filters in the first layer/stage. 110

5.12 Learned weights and corresponding Fourier transform of some typical active convolution filters of second layer/stage. 110

5.13 Spectrum of filters that represent a first order derivative with a low-pass Gaussian filter (from left to right: $\theta = 0, 90, 45°$). 110

5.14 Spectrum of an actual filter and an approximation plotted along $k_x = k_y$. . . 111

5.15 Learned weights and corresponding Fourier transform of some typical active convolution filters of third layer/stage. 111

5.16 Typical 'difference in displacement' image of each class for real data 114

5.17 Hybrid ML/IP approach for solving a classification problem. 120

5.18 A typical material parameter distribution. 121

Abstract

The process of calculating the hidden *causes* behind an observed *effect* is an interesting mathematical problem. Such problems, commonly referred to as "inverse problems", appear in almost all areas of science and engineering. Due to their ubiquitous nature and associated algorithmic and computational challenges, such problems have gained substantial research interest in recent years. Much of these recent research efforts have been dedicated to tackle one of the central challenges of inverse problems: their ill-posed nature. There are two different schools of thought to tackle this ill-posedness and they lead to two different problem formulations: (i) deterministic and (ii) stochastic. The goal of this thesis is to advance the state-of-the-art of both these formulations.

We achieve this in the case of stochastic inverse problems by integrating a physics-based model (of the phenomena under study) with data-driven deep generative algorithms in a unified Bayesian framework. This allows us to tackle two fundamental challenges in stochastic inverse problems: (i) inferring high dimensional posterior distribution ("curse of dimensionality") and (ii) enabling the use of complex (not-easy-to-quantify) prior distributions. We demonstrate the effectiveness of this framework on a wide variety of applications such as heat conduction, subsurface flow modeling, elasticity imaging, microstructure identification, etc. Further, we propose an extension of this algorithm which allows for the quantification of uncertainty in the outputs of modern deep learning algorithms. We showcase the efficacy of this on a range of tasks such as image classification, denoising, and inpainting with quantified uncertainty estimates. We also showcase how such uncertainty estimates could be useful in downstream tasks such as optimal experimental design/active learning.

In the case of deterministic inverse problems we focus on elasticity imaging applications and apply adjoint-based optimization technique to infer the visco-elastic properties of the tissue from time-harmonic displacement field *without* requiring any boundary condition data. In order to reduce the overall computational *time* of solving such deterministic inverse problems, we propose a novel domain decomposition technique that can solve any large-scale inverse problem involving time-harmonic data in parallel. To reduce the total computational *cost* of elasticity imaging, we propose deep learning assisted simplified workflow that can circumvent the solution of complex and expensive inverse problems.

Chapter 1

Introduction

> The greatest progress that the human race has made lies in learning how to make correct inferences.
>
> *Friedrich Nietzsche*

Inferring the hidden causes behind an observed phenomenon has intrigued scientists for centuries. The quest for such deduction has led to numerous breakthroughs in science and has opened up many new possibilities. A historical example is that of Adams and Le Verrier. In the 1840's they observed the perturbation in the trajectory of Uranus and their calculations based on this observation led to the discovery of Neptune. This process of calculating from a set of observations the causal factors behind them is commonly referred to as inverse problems.

Inverse problems are some of the most interesting and important mathematical problems in science and engineering playing key role in many application domains such as geophysics [1–4], climate modeling [5, 6], astrophysics [7, 8], heat conduction [9, 10], medical imaging [11, 12], chemical kinetics [13], materials modeling [14], machine learning [15], disease diagnostics [16, 17] and so on. In different domains, inverse problems appear in different names such as system identification, data assimilation, history matching, design sensitivity

analysis, or PDE-constrained optimization. While each of them has their unique structure and characteristics, the general mathematical framework is the same.

1.1 Inverse Problems

While defining the term *inverse problem* it is important to consider the inverse of what? According to Keller [18], "We call two problems inverses of one another if the formulation of each involves all or part of the solution of the other. Often, for historical reasons, one of the two problems has been studied extensively for some time, while the other is newer and not so well understood. In such cases, the former is called the direct problem, while the latter is called the inverse problem". While this definition does not provide clear guidelines for classifying *direct* (often referred to as *forward*) and *inverse* problems, in the case of mathematical modeling of physical phenomena, however, there is a natural distinction between the two. For example, if one wants to predict the future state of a physical system from all the knowledge of its current state and the underlying physical laws (including the values of relevant parameters in that law), then such a problem can be referred to as *direct* or *forward* problem. On the other hand if one wants to infer the present state of the system from the observations of its future state (i.e. going backward in time) or inferring the value of parameters in the governing physical law from the observation of the evolution of the system then such problems are referred to as *inverse* problems.

To make this more precise consider the problem of heat conduction in a solid material. If we are given the geometry of the domain, spatial distribution of material properties (diffusivity), boundary condition, and initial condition (at time t_0) then we can use this information in conjunction with the transient heat conduction equation to predict the state (temperature) of this solid material in future (at time t). This problem is considered a direct or forward problem. Whereas if we are given the temperature distribution at final time t

and we are interested in finding the initial condition at time t_0 corresponding to this final temperature distribution, then that problem is considered as inverse problem.

From the application standpoint, there are two possible motivations for studying inverse problems : (i) to *know* the past state of the system or the parameters of the system from the observation of the current state of the system (ii) to *determine* the current state or parameters of the system such that it *steers* to the desired future state. Thus inverse problems are concerned with *determining* the causes for the *observed* or *desired* effect. The first motivation (related to *observed* effect) leads to parameter identification and data assimilation type problems whereas the second motivation (related to *desired* effect) leads to optimization and design sensitivity analysis type problems.

Apart from these motivating applications, there is inherent mathematical appeal and rigor in studying inverse problems. This appeal comes from their characteristic ill-posedness. A problem is considered ill-posed if it does not satisfy any one or all three requirements of well-posedness (due to Hadamard [19]). These requirements are:

1. A solution exists.

2. The solution is unique.

3. The solution depends continuously on its parameters.

While some inverse problems do satisfy either the first or the first two requirements for well-posedness, most do not satisfy the third requirement making them ill-posed. Broadly speaking there are two schools of thought to tackle this ill-posedness: (i) Regularization methods and (ii) Bayesian inversion.

1.1.1 Regularization methods

The most widely used approach to tackle the ill-posedness in inverse problems is via regularization. In this approach, an extra regularization/penalty term is added to the inverse problem objective which promotes a particular structure in the inferred solution. The choice

of the functional form of this term leads to different regularization and as a result different inferred solutions and hence care should be taken while designing/selecting it.

To make things more concrete let us consider a closed and bounded domain $\Omega \in \mathbb{R}^n$. There is a physical process $\mathbb{F}$ mapping cause $\mathbb{X}$ to effect $\mathbb{Y}$ and the mathematical model for this process is given as $\hat{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\eta}$, where $\boldsymbol{x} \in \mathbb{R}^{n_x}$ represents discrete representation of the parameter of the problem. It is also the quantity we are interested in inferring. This could be the spatial distribution of material property inside the domain (for example, diffusivity in the case of heat conduction), the geometry of the domain, boundary condition or initial condition. $\hat{\boldsymbol{y}} \in \mathbb{R}^{n_y}$ represents discrete representation of observed/measured quantity (for example, temperature at time t in the case of heat conduction) and $\boldsymbol{f} \colon \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_y}$ represents the forward operator mapping parameters to observations (for example, discretized version of transient heat equation in the case heat conduction). Now, to tackle the ill-posedness via regularization method, the inverse problem is posed as an optimization problem with an added regularization/penalty term and an optimal solution of this optimization problem is considered as the desired solution of inferred parameter.

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} \frac{1}{2}||\hat{\boldsymbol{y}} - \boldsymbol{f}(\boldsymbol{x})||^2 + \frac{\lambda}{2}||\boldsymbol{x} - \boldsymbol{x_0}||_R^2 \tag{1.1}$$

where the first term is referred to as the data-misfit term and it captures the discrepancy between the measurement and the response/effect of the inferred parameter. The second term is referred to as the regularization term and it promotes the desired structure in the inferred parameter (or penalizes unwanted variation in it). λ refers to the regularization parameter and it balances the relative weight of the data misfit and regularization term.

For most practical problems we have some a-priori knowledge about the inferred solution and the functional form of regularization term is selected to reflect that prior knowledge. For example, if we are interested in inferring the viscosity field ($\boldsymbol{x}$) of fluid flow from the pressure and velocity measurements ($\hat{\boldsymbol{y}}$) and we know from our prior knowledge that for a given application viscosity varies smoothly inside the domain with an average value of

variation centered around zero then smoothness-promoting regularization term such as H^1 regularization should be selected, which penalizes the gradient of inferred field i.e. we can choose $||\nabla \boldsymbol{x}||^2$ as regularization term in Eq. (1.1). Similarly, if we are dealing with the inverse problem arising in Computed Tomography (CT) and we know from our prior knowledge of analyzing multiple CT images that the inferred density images ($\boldsymbol{x}$) manifest piece-wise constant behavior then we can select Total Variation (TV) regularization which promotes such behavior.

For most of interesting inverse problems, Eq. (1.1) does not admit an analytical solution and hence gradient-based optimization algorithms (such as BFGS or conjugate gradient descent) are employed to obtain the solution. This will require taking gradient of the right hand side of Eq. (1.1) with respect to $\boldsymbol{x}$. Taking this gradient is not straightforward as it involves taking the gradient of $\boldsymbol{f}(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ which might involve the integro-differential operator $\boldsymbol{f}$ and hence often the adjoint method is employed.

1.1.2 Bayesian inversion

Bayesian inversion provides a completely new way of tackling ill-posedness in inverse problems. In this approach both known measurement as well as unknown inferred parameter is interpreted as random variable. This interpretation naturally gives rise to a statistical framework in which the process of finding the unknown inferred parameter given the (possibly noisy version of) measurement is posed as finding the conditional probability density.

To be more precise let $\mathcal{X}$ represent random variable corresponding to unknown parameter we are interested in inferring and let the random variable $\mathcal{Y}$ represent known measurement. Given a specific instance of random variable $\mathcal{Y} = \hat{\boldsymbol{y}}$, we are interested in inferring conditional

density $p_{\mathcal{X}}(\boldsymbol{x}|\boldsymbol{y})$. In Bayesian inference this conditional density is computed by invoking Bayes' rule:

$$p_{\mathcal{X}}^{\text{post}}(\boldsymbol{x}|\hat{\boldsymbol{y}}) = \frac{p_{\mathcal{Y}}^{\text{like}}(\hat{\boldsymbol{y}}|\boldsymbol{x})p_{\mathcal{X}}^{\text{prior}}(\boldsymbol{x})}{p_{\mathcal{Y}}(\hat{\boldsymbol{y}})} \tag{1.2}$$

$$\propto p_{\eta}^{\text{like}}(\hat{\boldsymbol{y}} - \boldsymbol{f}(\boldsymbol{x}))p_{\mathcal{X}}^{\text{prior}}(\boldsymbol{x}) \tag{1.3}$$

where, $p_{\mathcal{X}}^{\text{prior}}(\boldsymbol{x})$ is called the *prior density*. It expresses our knowledge about random unknown parameter $\mathcal{X}$ *prior* to observing measurement $\mathcal{Y} = \hat{\boldsymbol{y}}$. $p_{\mathcal{Y}}^{\text{like}}(\hat{\boldsymbol{y}}|\boldsymbol{x})$ is called the likelihood density as it expresses the likelihood of different measurement outcomes given parameter $\mathcal{X} = \boldsymbol{x}$. $p_{\mathcal{X}}^{\text{post}}(\boldsymbol{x}|\boldsymbol{y})$ is called the posterior density as it expresses conditional probability of parameter $\mathcal{X}$ after (*post*) observing the measurement $\mathcal{Y} = \hat{\boldsymbol{y}}$. $p_{\mathcal{Y}}(\hat{\boldsymbol{y}})$ is called the evidence term as it represents the probability of evidence/measurement. Eq. (2.3) is obtained from Eq. (2.2) by considering the fact that $\hat{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\eta}$ with $\boldsymbol{\eta} \sim p_{\eta}$.

As explained above in the case of inverse problems we are interested in inferring the posterior density $p_{\mathcal{X}}^{\text{post}}(\boldsymbol{x}|\boldsymbol{y})$ and by looking at Bayes' formula in Eq. (2.3), we can say that solving such a statistical/Bayesian inverse problem may be broken into three steps:

1. Based on all prior knowledge about unknown parameter $\mathcal{X}$, find an appropriate probability density $p_{\mathcal{X}}^{\text{prior}}(\boldsymbol{x})$.

2. Find/Compute the likelihood density $p_{\eta}^{\text{like}}(\hat{\boldsymbol{y}} - \boldsymbol{f}(\boldsymbol{x}))$ based on known/assumed density for model/measurement error p_{η}.

3. Compute posterior probability density.

For most of the practical problem of interest however we are not interested in computing posterior density itself, but are more interested in computing some statistics (such as mean or variance) or confidence intervals with respect to it.

Let $s(\boldsymbol{x})$ denote some desired quantity of interest (QoI)/estimator with respect to posterior. It can be computed as,

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim p^{\text{post}}(\boldsymbol{x}|\boldsymbol{y})}[s(\boldsymbol{x})] = \int_{\Omega_{\mathcal{X}}} s(\boldsymbol{x}) p^{\text{post}}(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x} \tag{1.4}$$

Computing this estimator in general involves integration over $\mathbb{R}^{n_x}$ dimensional space. For most of the practical science and engineering problem n_x is very large making use of traditional numerical integration methods based on quadrature rule useless. So, instead one has to rely on sampling based techniques and often times efficient sampling techniques such as Markov Chain Monte Carlo (MCMC) is employed in practice.

1.2 Machine Learning

While many algorithms are developed over the years based on regularization methods and Bayesian inversion to solve real-world and large-scale inverse problems. Still these algorithms face many practical challenges such as choosing appropriate prior distribution, characterizing high-dimensional posterior distribution, performing reconstruction in the presence of unknown forward map, choosing appropriate value of hyper-parameters (such as regularization parameter), etc. In recent years machine learning (ML) and deep learning (DL) in particular has revolutionized many domains ranging from image recognition [20, 21], natural language processing [22], disease diagnosis [23], reinforcement learning [24] to high energy physics [25], computational chemistry [26], lithography [27], and medical imaging [28]. These algorithms are extremely powerful in extracting pattern from high-dimensional data. Moreover, they are quite flexible in nature and can work with any type of data making them a tempting choice for investigating inverse problems. Due to these desirable properties recently there has been significant interest in using these deep learning based models to solve inverse problems. These works can broadly be classified in two categories.

1.2.1 Supervised v/s Unsupervised inversion

1.2.1.1 Supervised inversion

This refers to scenarios where we have access to samples from joint distribution of parameter $\boldsymbol{x}$ and measurement $\boldsymbol{y}$. Once we have the dataset $\mathcal{S} = \{(\boldsymbol{x_i}, \boldsymbol{y_i})\}_{i=1}^{N}$, where $(\boldsymbol{x_i}, \boldsymbol{y_i}) \sim p_{\mathcal{XY}}(\boldsymbol{x}, \boldsymbol{y})$ of samples from joint distribution we can train a deep neural network (also sometimes referred to as *reconstruction network*) $\mathcal{NN}_\theta(\cdot)$ to learn the mapping $\mathcal{Y}$ to $\mathcal{X}$. Denoising auto-encoders [29], U-Net based inversion [30], Deep convolutional framelets [31], Unrolled optimization-based inversion [32, 33], and Neumann networks [34] are some examples of the supervised inversion. While quite efficient and accurate this type of method is quite sensitive to perturbation in the measurement process and new network needs to be trained for each type new forward map/measurement process is used.

1.2.1.2 Unsupervised inversion

This second family of method considers scenario where we do not have samples from joint distribution, but only have samples from marginal distribution. In other words we do not have matching dataset of $\boldsymbol{x}$ and $\boldsymbol{y}$. This type of inversion method can be subdivided in three sub-categories:

1. Methods that only use samples of parameter $\boldsymbol{x}$: plug-and-play (PnP) approach [35], Regularization by Denoising (RED) [36], Learned Denoising-Based Approximate Messaging Passing (LDAMP) [37], and Compressed Sensing using Generative Models (CSGM) [38] are some popular approaches falling into this sub-category.

2. Methods that only use samples of measurement $\boldsymbol{y}$: Self-supervision based inversion [39, 40], Generalized Stein's unbiased risk estimator (GSURE) based approaches [41], Noise2Noise [42], and AmbientGAN [43] are some popular examples of this sub-category.

3. Methods that use unpaired parameter $\boldsymbol{x}$ and measurement $\boldsymbol{y}$ samples: CycleGAN [44] based approaches [45, 46] fall into this sub-category.

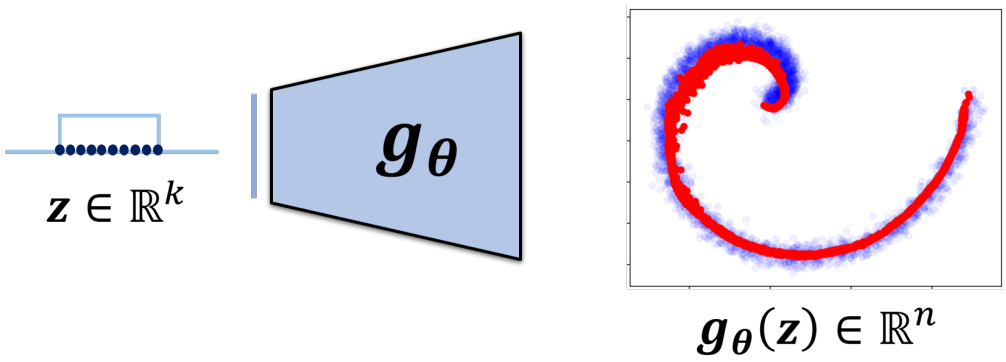


Figure 1.1: A deep generative model is a function $\boldsymbol{g_\theta}(\boldsymbol{z})$ that takes a low-dimensional random vector $\boldsymbol{z} \in \mathbb{R}^k$ and produces a high-dimensional sample $\boldsymbol{g_\theta}(\boldsymbol{z}) \in \mathbb{R}^n$. The function is parameterized by a neural network with parameters $\boldsymbol{\theta}$. In the example shown in the figure, the generator $\boldsymbol{g} : \mathbb{R} \mapsto \mathbb{R}^2$ learns to map low-dimensional samples $\boldsymbol{z}$ (dark blue dots) drawn from a uniform distribution (light blue line), such that the distribution of $\boldsymbol{g}(\boldsymbol{z})$ (red dots) resembles the distribution of training samples (blue dots).

1.2.2 Deep Generative Models (DGM)

A central goal of learning from data paradigm is to succinctly model high-dimensional data distribution in a way that permits efficient learning and sampling. Deep generative models provide a flexible and expressive framework of achieving this goal. Their capability of learning complex high-dimensional data distribution coupled with their special structure which allows for dimensionality reduction makes them a perfect candidate for solving high-dimensional and complex inverse problems (in both statistical/Bayesian and deterministic/regularization setting).

DGMs represent complex and high-dimensional probability distribution by applying a deterministic and non-linear transformation to a distribution which is easy-to-sample from. Figure 1.1 shows an illustrative example of a deep generative model that maps $\mathbb{R} \mapsto \mathbb{R}^2$. While the output of this simple generative model lies in $\mathbb{R}^2$, modern deep generative models are capable of generating samples in millions of dimensions. Furthermore, the generator function is typically a convolutional neural network and is therefore continuous and differentiable almost everywhere allowing use of efficient gradient based algorithms for their training. Some

examples of some popular DGMs are Variational Auto-Encoders (VAE) [47], Generative Adversarial Networks (GAN) [48], Normalizing Flows [49, 50]. DGMs have shown tremendous success in solving inverse problem specially at learning complex prior distribution and have been successfully applied in various applications such as compressive sensing [38, 43], phase retrieval [51], geophysics [52], blind deconvolution [53] to name a few.

1.3 Contents and Contributions of This Thesis

The goal of this thesis is to propose novel and general-purpose physics-based data-driven algorithms to push the state-of-the-art of deterministic as well as stochastic inverse problems. We achieve this in the case of stochastic inverse problems by integrating physics-based models with data-driven deep generative algorithms in a unified Bayesian framework. This allows us to tackle two fundamental challenges in Bayesian inversion: (i) inferring high dimensional posterior distribution ("curse of dimensionality") and (ii) enabling the use of complex prior distributions. In the case of deterministic inverse problems, we achieve this goal by implementing an adjoint-based optimization technique that does not require any boundary condition data. Furthermore, to reduce the total computational time, we propose a novel domain decomposition technique to solve large-scale time-dependent deterministic inverse problem in parallel.

In the next chapter, we will begin our survey of inverse problems with special emphasis on Bayesian inverse problems. Most existing Bayesian inversion techniques struggle to scale to large dimensions restricting their potential use in real-world applications. Further, at the time of this study, there was an open and important challenge of quantifying prior density in situations where this information is available in the qualitative form. We made an original contribution to the field and proposed a novel algorithm addressing both these issues. In this chapter, we provide details of this algorithm along with its first applications to a wide

array of domains such as inverse heat conduction, elasticity imaging, material microstructure identification etc.

In the following chapter, we extended this algorithm to the emerging and important field of Bayesian deep learning and demonstrate how with minor modifications our algorithm can perform any supervised learning tasks (such as image classification, inpainting, denoising, and out-of-distribution detection) with quantified uncertainty estimates and can outperform many existing Bayesian deep learning methods. Moreover, we showcase how our proposed algorithm can simultaneously solve forward and inverse uncertainty quantification problems. We demonstrate the effectiveness of this algorithm for subsurface flow modeling task involving Darcy's law.

In Chapter 4 we turn our attention to deterministic inverse problems and tackle the problem arising in elasticity imaging of inferring the visco-elastic material properties of tissue from time-harmonic displacement data with incomplete or no information of boundary condition. Many existing techniques make additional assumptions about the domain and/or boundary to account for the incomplete boundary information. We instead implement an adjoint-based optimization algorithm leveraging coupled adjoint-state equation method [54], which enables solution of inverse problem without boundary condition. Moreover, to reduce total computational time for solving such inverse problems, we propose a novel domain decomposition algorithm capable of solving any large-scale deterministic inverse problem in parallel involving time harmonic data.

This leads to chapter 5, where we consider alleviating the need of solving the expensive inverse problem in elasticity imaging and propose a novel data-driven elastography workflow leveraging the power of deep learning. This simplified workflow could be applicable to any situation where classification is done based on the output of the inverse problem. In this process we analyze the learning process of the deep learning models and reveal an interesting connection of trained filters of deep models to traditional strain imaging techniques. Further, we provide the first demonstration of domain randomization in bio-mechanical imaging.

Finally, we conclude this thesis with a summary and an outlook for promising future directions.

Chapter 2

Efficient Solution Techniques for Bayesian Inverse Problems

> It ain't what you don't know that gets you into trouble. It's what you know for sure that just ain't so.
>
> *Mark Twain*

2.1 Introduction

Inverse problems refer to the process of inferring the latent parameters (the "cause") from a set of measured observations (the "effect") for a given system. Such problems arise in various areas of science and engineering such as geophysics [1–4], climate modeling [5], chemical kinetics [13], heat conduction [9], astrophysics [7, 8], materials modeling [14], and the detection and diagnosis of disease [16, 17]. Due to their enormous practical importance and associated algorithmic and computational challenges it has gained substantial research interest in recent years [55, 56].

The algorithmic and computational challenges in inverse problems primarily stem from their *non-local* and *non-causal* nature. To understand this let us first consider the "forward problem" (which refers to the process of computing the "effect" given the "cause"). The laws

of nature are typically expressed as forward problems in the form of differential equations which are *local* in the sense that the physical condition of a system at a given point (in space and time) can be expressed as value of a function and its derivative at that point. Another typical feature of the forward problem is *causality*: the later condition of a system only depends on its previous states. Inverse problems on the other hand are most often *non-causal* and/or *non-local* in nature.

2.2 Ill-posedness of Inverse Problems

This *non-causality* and *non-locality* of inverse problems greatly contributes to their instability. To understand this, consider the heat conduction problem in a solid material. Given the geometry of the domain, material property (conductivity) distribution, boundary condition, and initial temperature distribution at time T_0 one can solve the diffusion equation to determine the final temperature distribution in domain at time T. Some small perturbation in the initial temperature distribution (T_0) smears out over time and leaves the final temperature distribution at time T almost unaltered. Thus the forward problem is stable with respect to perturbation in its parameters (initial condition here).

Going in the inverse (*non-causal*) direction however (of determining the initial temperature distribution at time T_0 from the measured temperature at time T) is challenging, as we find that vastly different initial conditions may have produced the same final temperature distribution. In general, inverse problems are considered ill-posed as they do not satisfy any one or all of the three conditions of well-posedness (proposed by Hadamard): existence, uniqueness, and stability.

There are two popular and widely different approaches to tackle this ill-posedness: (1) deterministic approach based on regularization techniques, (2) statistical approach based on Bayesian inference.

2.2.1 Deterministic approach

In the first approach the inverse problem is posed as an optimization problem with an added regularization/penalty term which promotes a particular structure in the inferred solution. Let $\boldsymbol{x} \in \Omega_x \subset \mathbb{R}^{n_x}$ be our parameter of interest and $\hat{\boldsymbol{y}} \in \Omega_y \subset \mathbb{R}^{n_y}$ be our observed quantity. They are related via a forward map $\hat{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\eta}$, where $\boldsymbol{f}$ is the forward map (which often takes the form of a differential equation) and $\boldsymbol{\eta}$ captures model and/or measurement error. Then in order to infer the parameter $\boldsymbol{x}$ from observation $\hat{\boldsymbol{y}}$ via regularization approach, we can solve the following optimization problem.

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} \frac{1}{2}||\hat{\boldsymbol{y}} - \boldsymbol{f}(\boldsymbol{x})||_Y^2 + \frac{\lambda}{2}||\boldsymbol{x} - \boldsymbol{x_0}||_R^2 \tag{2.1}$$

where, λ is regularization parameter, $\boldsymbol{x_0}$ is the reference value of parameter of interest, and $||\cdot||_R$ and $||\cdot||_Y$ are appropriate norms for regularization and data misfit terms respectively. In the context of inverse heat conduction problem described above, $\boldsymbol{x}$ refers to the initial temperature distribution at time T_0 and $\hat{\boldsymbol{y}}$ refers to (possibly noisy) measured temperature at time T with $\boldsymbol{f}$ representing time-dependent diffusion equation. By solving the above optimization problem using efficient gradient-based algorithms (like BFGS or conjugate gradient descent) one can obtain a unique solution $\boldsymbol{x}^*$ for a given measurement $\hat{\boldsymbol{y}}$. While this deterministic approach is quite efficient and useful for many applications, it only gives point estimates and hence lacks the ability of computing error/confidence interval in the inferred solution. Access to such uncertainty information in the inferred solution is critical in many applications where high-stakes decisions are made based on the inferred solution of the inverse problem.

2.2.2 Statistical approach

In order to obtain such quantified uncertainty estimates in the inferred solution it is essential to pose the inference problem in a probabilistic setting, where every known as well as

unknown quantity is expressed as a random variable and for a given measurement $\mathcal{Y} = \hat{\boldsymbol{y}}$, instead of inferring a unique parameter $\boldsymbol{x}^*$, we infer the whole probability distribution $p(\boldsymbol{x}|\hat{\boldsymbol{y}})$. A principled way of inferring this conditional probability distribution is via Bayesian inference. In Bayesian inference this conditional density is computed by invoking Bayes' rule:

$$p_{\mathcal{X}}^{\text{post}}(\boldsymbol{x}|\hat{\boldsymbol{y}}) = \frac{p_{\mathcal{Y}}^{\text{like}}(\hat{\boldsymbol{y}}|\boldsymbol{x}) p_{\mathcal{X}}^{\text{prior}}(\boldsymbol{x})}{p_{\mathcal{Y}}(\hat{\boldsymbol{y}})} \tag{2.2}$$

$$\propto p_{\eta}^{\text{like}}(\hat{\boldsymbol{y}} - \boldsymbol{f}(\boldsymbol{x})) p_{\mathcal{X}}^{\text{prior}}(\boldsymbol{x}) \tag{2.3}$$

where, $p_{\mathcal{X}}^{\text{prior}}(\boldsymbol{x})$ is called the *prior density*. It expresses our knowledge about random unknown parameter $\mathcal{X}$ *prior* to observing measurement $\mathcal{Y} = \hat{\boldsymbol{y}}$. $p_{\mathcal{Y}}^{\text{like}}(\hat{\boldsymbol{y}}|\boldsymbol{x})$ is called likelihood density as it expresses the likelihood of different measurement outcomes given parameter $\mathcal{X} = \boldsymbol{x}$. $p_{\mathcal{X}}^{\text{post}}(\boldsymbol{x}|\boldsymbol{y})$ is called the posterior density as it expresses conditional probability of parameter $\mathcal{X}$ after (*post*) observing the measurement $\mathcal{Y} = \hat{\boldsymbol{y}}$. $p_{\mathcal{Y}}(\hat{\boldsymbol{y}})$ is called the evidence term as it represents the probability of evidence/measurement. Eq. (2.3) is obtained from Eq. (2.2) by considering the fact that $\hat{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\eta}$ with $\boldsymbol{\eta} \sim p_{\eta}$.

As explained above in the case of inverse problems we are interested in inferring the posterior density $p_{\mathcal{X}}^{\text{post}}(\boldsymbol{x}|\boldsymbol{y})$ and computing relevant statistics (such as confidence intervals) with respect to it and by looking at Bayes' formula in Eq. (2.3), we can say that solving such a statistical/Bayesian inverse problem may be broken into three subtasks:

1. Based on all prior knowledge about unknown parameter $\mathcal{X}$, find an appropriate probability density $p_{\mathcal{X}}^{\text{prior}}(\boldsymbol{x})$.

2. Find/Compute the likelihood density $p_{\eta}^{\text{like}}(\hat{\boldsymbol{y}} - \boldsymbol{f}(\boldsymbol{x}))$ based on known/assumed density for model/measurement error p_{η}.

3. Develop methods to characterize the posetrior probability density.

Computing the likelihood density (step 2) is the most easy part in above process. In the case of physics-driven Bayesian inverse problems (which is the central focus of this chapter),

we typically have a well-defined functional form of forward operator $\boldsymbol{f}$, which is deterministic in nature. Further, we have a reasonable understanding of model/measurement error $\boldsymbol{\eta}$ for a given measurement process leading to reliable estimate of p_η making step 2 of above process straightforward. However, step 1 and 3 are quite challenging and require significant consideration. So, we briefly describe some of the major challenges associated with each of them below.

2.2.2.1 Prior modeling

Construction of feasible and reliable prior density is arguably one of the most important yet challenging task of statistical inversion. The major problem with finding an appropriate prior density lies usually in the nature of prior information. Oftentimes, this prior information of the unknown parameter of interest is *qualitative* in nature and the challenge is to translate this qualitative information into a *quantitative* form so that it can be encoded in a prior density.

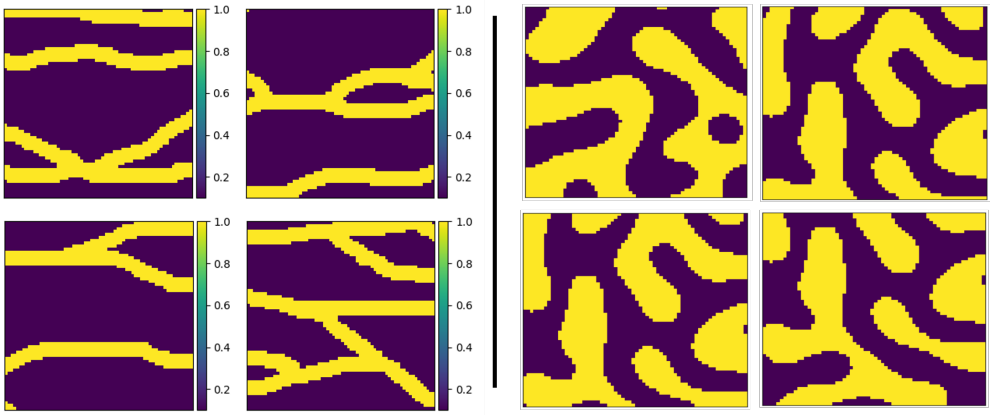


Figure 2.1: *Left:* Four typical samples of permeability in subsurface of earth. *Right:* Four typical samples of bi-phase material microstructure. Representing qualitative information of such prior samples into a quantitative form useful for prior characterization is non-trivial.

For example, a geophysicist studying the subsurface structure of earth in a given region might know from his/her experience that the permeability of earth in that region might have a channeled structure due to groundwater streams with elevated permeability value

in specific channelized format dispersed in low permeability background region (Figure 2.1 left panel). Translating this *qualitative knowledge* into a *quantitative form* that can be used as prior density is not obvious and straightforward. Similarly, a material scientist studying the microstructure of materials might have an *intuitive feeling* about how microstructure should look like (Figure 2.1 right panel) and what is considered as a 'valid' microstructure. Transforming such *feeling* into an *equation* useful for prior characterization is not easy. In this chapter we will tackle this challenge and propose new algorithms to characterize prior density from qualitative prior information represented in the form of samples.

A second challenge associated with prior modeling is its *subjective* nature. As different people/modelers might have different prior knowledge and belief about the unknown parameters of interest, they may use different prior density reflecting their prior belief about these parameters. This eventually leads to different posterior densities making it difficult to reliably evaluate and compare final inferred posterior statistics and confidence intervals. Thus, it is desirable to have an algorithm which can *objectively* capture the prior information about the parameters directly from available data of these parameters. We will show how method developed in this chapter can address this issue in a reliable and efficient manner.

Finally, the third challenge with prior modeling is what to do when the modeler does not have sufficient domain knowledge or prior information about the parameter of interest to formulate a reliable prior density. There are many situations where such scenario arises where one only has access to samples of parameters but does not have sufficient prior knowledge of these parameters to fully characterize the prior density and in such situations often "uninformative priors" [1] are used to represent one's lack of belief about any specific parameter values. Yet in these situations it is desirable to somehow use the indirect information available in the forms of samples of parameters to represent prior density.

[1]The term "uninformative prior" is somewhat of a misnomer, as it does represent our lack of belief about parameter values, which is akin to specifying some amount of specific information. As many Bayesians would tell every prior is informative and there is no such thing as a prior with "truly no information".

2.2.2.2 Posterior characterization:

The third and final step in the statistical inversion technique is to characterize the posterior distribution. In Bayesian inversion most often we are not interested in the posterior distribution itself, but are interested in computing some statistics with respect to this posterior. So, in order to completely understand all the intricacies of this posterior characterization step, let us first briefly consider how any statistics are computed with respect to posterior distribution.

Let $s(\boldsymbol{x})$ denote some desired quantity of interest (QoI)/statistics. A common example of one such QoI is mean of the posterior distribution often denoted as $\bar{\boldsymbol{x}}$ and it can be computed as,

$$\bar{\boldsymbol{x}} = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p^{\text{post}}(\boldsymbol{x}|\boldsymbol{y})}[\boldsymbol{x}] = \int_{\Omega_{\mathcal{X}}} \boldsymbol{x} p^{\text{post}}(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x} \tag{2.4}$$

In general any QoI $s(\boldsymbol{x})$ of posterior distribution can be computed as,

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim p^{\text{post}}(\boldsymbol{x}|\boldsymbol{y})}[s(\boldsymbol{x})] = \int_{\Omega_{\mathcal{X}}} s(\boldsymbol{x}) p^{\text{post}}(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x} \tag{2.5}$$

Computing these QoIs requires solving an integration problem in $\mathbb{R}^{n_x}$, where n_x is the dimension of the space in which random variable $\mathcal{X}$ lives. For the physics-driven inverse problems (which are the primary focus of this thesis), the value of n_x is very high ($\mathcal{O}(10^4 - 10^7)$) as for such problems $\mathcal{X}$ represents nodal value of discretized version of parameter field and for most interesting and practically useful problems this discretization is relatively fine with large number of nodes resulting in very high dimension of random variable $\mathcal{X}$. This means integrating the above integral using numerical/quadrature-based method is simply infeasible (even on supercomputers) due to exponential dependence of number of quadrature points on dimension n_x and hence we have to rely on methods that approximate this intractable integral.

There are two popular approaches to approximate this posterior integral: (i) variational inference and (ii) Monte Carlo integration. In variational inference the intractable posterior

integral is approximated by a tractable integral. This is done via variational principles which results in an optimization problem. While easy-to-understand and relatively efficient, variational inference provides biased estimate of posterior statistics and hence is not a preferred choice for physics-driven inverse problems. Monte Carlo integration on the other hand provides unbiased estimate of posterior statistics by computing the sum described in Eq. (2.6) in the limit of large number of samples (N).

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim p^{\text{post}}(\boldsymbol{x}|\hat{\boldsymbol{y}})}[s(\boldsymbol{x})] = \int_{\mathbb{R}^{n_x}} s(\boldsymbol{x}) p^{\text{post}}(\boldsymbol{x}|\hat{\boldsymbol{y}}) d\boldsymbol{x} \approx \frac{1}{N}\sum_{i=1}^{N} s(x_i), \qquad \text{where } x_i \sim p^{\text{post}}(\boldsymbol{x}|\hat{\boldsymbol{y}}) \quad (2.6)$$

Since taking large number of samples is computationally prohibitive, an effective method called Markov Chain Monte Carlo (MCMC) is often employed in practice. This method systematically generates samples from the posterior distribution. While the number of samples required to estimate posterior statistics with MCMC sampling is much less than random Monte Carlo sampling, the number of samples required for reliable statistics is still significantly large. Furthermore, each sample generated via MCMC requires the solution of the forward problem, which in the case of physics-driven inverse problems involves solving a PDE. This makes using MCMC based sampler for estimating posterior statistics in high dimension computationally prohibitive. Other than this computational challenge, it is also difficult to even design MCMC algorithms that converge in high dimension. This so-called "curse of dimensionality" is one of the main impediment in solving practical large-scale physics-driven Bayesian inverse problems and restricting their potential integration in downstream tasks. In fact, due to this "curse of dimensionality" often some summary posterior statistics (like average value of QoI in some small specific region) is reported in practice. It is thus extremely desirable to have access to algorithms which can overcome this "curse of dimensionality" and can compute posterior statistics in high dimensions.

In this chapter, we propose novel algorithm which efficiently tackles this "curse of dimensionality" enabling posterior characterization in high dimension and at the same time is

capable of modeling complex prior distributions which are difficult to represent in an analytical form. We achieve this by leveraging ideas from deep generative modeling. Specifically we use a deep Generative Adversarial Network (GAN) as a prior in Bayesian update and reformulate the resulting inference problem in the low-dimensional latent space of the GAN. In the next section we provide brief introduction of GAN followed by how GAN can be used as a prior in Bayesian inference. We finish this chapter with numerical results from diverse applications such as inverse heat conduction, elasticity imaging, material microstructure identification, etc.

2.3 Generative Adversarial Networks

GANs [48] are a class of deep generative models, which are trained in an adversarial fashion to generate samples from a distribution $p^g_{\mathcal{X}}$ which approximates some target distribution $p_{\mathcal{X}}$ of the data $\mathcal{X}$. Typically, GAN comprise of a generator function $\boldsymbol{g}\colon \boldsymbol{z} \in \Omega_{\mathcal{Z}} \subset \mathbb{R}^{n_z} \mapsto \boldsymbol{x} \in \Omega_{\mathcal{X}} \subset \mathbb{R}^{n_x}$, where typically, $n_z \ll n_x$ and $\boldsymbol{z} \sim p_{\mathcal{Z}}$ with $p_{\mathcal{Z}}$ being an easy-to-sample from distribution, typically a Gaussian (with zero mean and unit variance) or a uniform distribution. The generator up-scales the lower dimensional latent vector $\boldsymbol{z}$ through successive application of non-linear transformations defined via deep neural network. In particular, the generator is a function $\boldsymbol{g}(\boldsymbol{z}, \boldsymbol{\theta})$, $\boldsymbol{g} : \Omega_{\mathcal{Z}} \times \mathbb{R}^{N_\theta} \mapsto \Omega_{\mathcal{X}}$ which is approximated by a neural network with weight parameters $\boldsymbol{\theta}$ whose values are determined during the training process described below. The number of weights, N_θ, is a measure of the capacity of the generator which increases with increasing N_θ.

The other component of a GAN is a discriminator function, which is also approximated by successive non-linear transformations defined via neural network. However, these transformations are designed to down-scale the original input. The final few layers of the discriminator are fully connected and lead to a single scalar-valued field. Thus the discriminator, $d(\boldsymbol{x}, \boldsymbol{\phi})$, $d : \Omega_{\mathcal{X}} \times \mathbb{R}^{N_\phi} \mapsto \mathbb{R}$. Here $\boldsymbol{\phi}$ is the vector of the weights of the discriminator, and the number

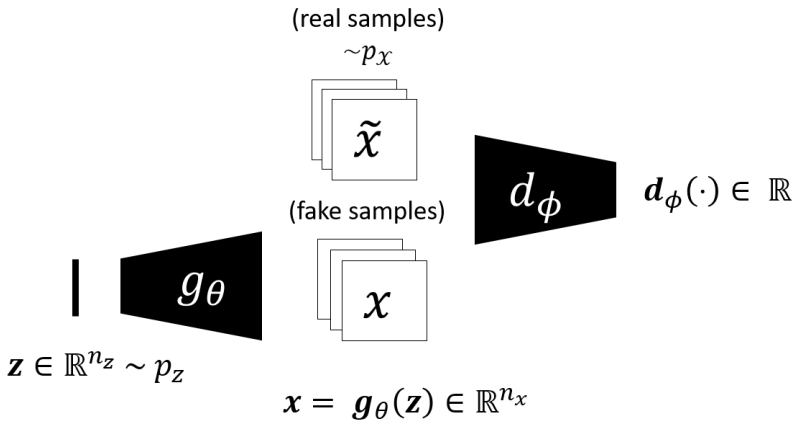


Figure 2.2: Schematics of Generative Adversarial Network

of weights, N_ϕ, is a measure of the capacity of the discriminator. The discriminator is trained so that it attains large positive values for inputs selected from the data distribution $p_{\mathcal{X}}$ and small values for inputs sampled from the distribution defined by the samples generated by the generator $p^g_{\mathcal{X}}$. This is made precise in the description of the objective function used to train the GAN, which is described below. Figure 2.2 shows the schematics of GAN.

The generator and the discriminator are trained in an adversarial manner. The training data for the discriminator is comprised of the set of real instances of $\boldsymbol{x}$, sampled from $p_{\mathcal{X}}$, and a set of "fake" instances generated by the generator, $\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{z}, \boldsymbol{\theta})$, with $\boldsymbol{z}$ sampled from $p_{\mathcal{Z}}$. The weights of the discriminator are determined by requiring it attain large values for the true instances of $\boldsymbol{x}$ and small values for the "fake" instances. On the other hand, the weights of the generator are determined by passing its output $\boldsymbol{g}(\boldsymbol{z})$ through the discriminator and requiring it to be considered as "real" i.e. by requiring it to attain high values. Thus while the generator is trained to "fool" the discriminator, the discriminator is trained so as not to be fooled by the generator. Different types of GANs can be obtained by appropriately selecting the objective function within these broad guidelines. In fact, the training objective of several GANs can be interpreted as the variational minimization of an appropriate divergence [57]. In this work, we work with the Wasserstein GAN (WGAN) [58, 59] which minimizes the

Wasserstein 1-distance between $p_{\mathcal{X}}$ and $p^g_{\mathcal{X}}$. We next describe the training objective of this WGAN below.

The objective function for the Wasserstein GAN (WGAN) is given by

$$L(\boldsymbol{\theta}, \boldsymbol{\phi}) \equiv \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{\mathcal{X}}}[d(\boldsymbol{x}, \boldsymbol{\phi})] - \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_Z}[d(\boldsymbol{g}(\boldsymbol{z}, \boldsymbol{\theta}), \boldsymbol{\phi})]. \tag{2.7}$$

The discriminator is trained to maximize this objective function, while the generator is trained to minimize it. This leads to the following min-max problem to determine the optimal values of the weights denoted by $(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*)$,

$$(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*) = \mathop{\text{argmin}}_{\boldsymbol{\theta}}(\mathop{\text{argmax}}_{\boldsymbol{\phi}}(L(\boldsymbol{\theta}, \boldsymbol{\phi}))), \tag{2.8}$$

under the constraint

$$\|d(\boldsymbol{x}, \boldsymbol{\phi})\|_{\text{Lip}} \leq 1. \tag{2.9}$$

In the original work on WGANs [58] this inequality constraint was approximately imposed by clipping the weights of the discriminator. This approach was then improved by replacing the inequality constraint with an equality constraint on the gradient of the discriminator with respect to $\boldsymbol{x}$ [59], and this version was referred to as WGAN-GP (gradeint penalty). In practise this constraint was enforced weakly through a penalty term that was added to the loss function for the generator. In our experiments we use this version of the WGAN. In the limit of infinite capacity, that is $N_\theta, N_\phi \to \infty$, the discriminator $d(\boldsymbol{x})$, $d : \Omega_{\mathcal{X}} \mapsto \mathbb{R}$, and the generator $\boldsymbol{g}(\boldsymbol{z})$, $\boldsymbol{g} : \Omega_Z \mapsto \Omega_{\mathcal{X}}$ can represent all continuous bounded functions, $\mathcal{C}_b$, over their respective domains. In this limit the finite-dimensional min-max is problem is placed by its infinite-dimensional counterpart:

$$(d^*, \boldsymbol{g}^*) = \mathop{\text{argmin}}_{\boldsymbol{g}\in\mathcal{C}_b}\Big(\mathop{\text{argmax}}_{d\in\mathcal{C}_b}\big(\mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{\mathcal{X}}}[d(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_Z}[d(\boldsymbol{g}(\boldsymbol{z}))]\big)\Big), \tag{2.10}$$

under the constraint

$$\|d(\boldsymbol{x})\|_{\text{Lip}} \leq 1. \tag{2.11}$$

The term within the large parenthesis in (2.10) is precisely the Wasserstein-1 distance [60](Remark 6.5). It is also known as Kantorovich-Rubinstein dual characterization of Wasserstein-1 distance. Therefore, we may write,

$$\boldsymbol{g}^* = \underset{\boldsymbol{g}\in\mathcal{C}_b}{\operatorname{argmin}} W_1(p_{\mathcal{X}}, \boldsymbol{g}_{\#}p_Z), \tag{2.12}$$

where W_1 is the Wasserstein-1 distance, and $\boldsymbol{g}_{\#}p_Z$ is the push-forward of p_Z by $\boldsymbol{g}$. In the Wasserstein-1 distance, the convergence of a sequence of probability measures, implies weak convergence [60](Theorem 6.8). Therefore, if $\boldsymbol{g}^*$ is the limit of a class of generators, $\boldsymbol{g}$, for which $W_1(p_{\mathcal{X}}, \boldsymbol{g}_{\#}p_Z) \to 0$, then for all continuous bounded functions $h \in \mathcal{C}_b(\Omega_{\mathcal{X}})$, we have,

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{\mathcal{X}}}[h(\boldsymbol{x})] = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_Z}[h(\boldsymbol{g}^*(\boldsymbol{z}))]. \tag{2.13}$$

In the following section we make use of this equality and demonstrate how WGAN may be used to solve the desired inference problem.

2.4 GAN as Prior in Bayesian inference

We now return to the problem of computing any QoI $s(\boldsymbol{x})$ with respect to posterior distribution as described in Eq. (2.5). We consider a WGAN whose generator has converged to a distribution that is weakly equivalent to $p_{\mathcal{X}}$. As before, we let $\boldsymbol{z} \sim p_{\mathcal{Z}}(\boldsymbol{z})$ characterize the latent vector space of this GAN, and let $\boldsymbol{g}^*(\boldsymbol{z})$ denote such generator. Assuming that p_η, and $s \in \mathcal{C}_b(\Omega_{\mathcal{X}})$ we have,

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{\mathcal{X}}^{post}}[s(\boldsymbol{x})] = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{\mathcal{X}}^{prior}}[\frac{s(\boldsymbol{x})p_\eta(\hat{\boldsymbol{y}}-\boldsymbol{f}(\boldsymbol{x}))}{p_{\mathcal{Y}}(\boldsymbol{y})}] \qquad \text{(from Eq.(2.5) and (2.2))} \tag{2.14}$$

$$= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{\mathcal{X}}}[\frac{s(\boldsymbol{x})p_\eta(\hat{\boldsymbol{y}}-\boldsymbol{f}(\boldsymbol{x}))}{p_{\mathcal{Y}}(\boldsymbol{y})}] \qquad \text{(using } p_{\mathcal{X}} \text{ as prior distribution)} \tag{2.15}$$

$$= \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{\mathcal{Z}}}[\frac{s(\boldsymbol{g}^*(\boldsymbol{z}))p_\eta(\hat{\boldsymbol{y}}-\boldsymbol{f}(\boldsymbol{g}^*(\boldsymbol{z})))}{p_{\mathcal{Y}}(\boldsymbol{y})}] \tag{2.16}$$

$$\text{(by letting } h(\boldsymbol{x}) = \frac{s(\boldsymbol{x})p_\eta(\hat{\boldsymbol{y}}-\boldsymbol{f}(\boldsymbol{x}))}{p_{\mathcal{Y}}(\boldsymbol{y})} \text{ in Eq. (2.13))}$$

$$= \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{\mathcal{Z}}^{\text{post}}}[s(\boldsymbol{g}^*(\boldsymbol{z}))], \tag{2.17}$$

where

$$p_{\mathcal{Z}}^{\text{post}}(\boldsymbol{z}|\boldsymbol{y}) \equiv \frac{1}{p_{\mathcal{Y}}(\boldsymbol{y})} p_\eta(\hat{\boldsymbol{y}}-\boldsymbol{f}(\boldsymbol{g}^*(\boldsymbol{z})))p_Z(\boldsymbol{z}). \tag{2.18}$$

Equation (2.17), which is the main result of this manuscript, implies that sampling from the posterior distribution of $\boldsymbol{x}$ is equivalent to sampling from the posterior distribution for $\boldsymbol{z}$ and passing the sample through the generator $\boldsymbol{g}^*$. That is,

$$\boldsymbol{x} \sim p_{\mathcal{X}}^{\text{post}}(\boldsymbol{x}|\hat{\boldsymbol{y}}) \Rightarrow \boldsymbol{x} = \boldsymbol{g}^*(\boldsymbol{z}), \boldsymbol{z} \sim p_{\mathcal{Z}}^{\text{post}}(\boldsymbol{z}|\boldsymbol{x}). \tag{2.19}$$

Since the dimension of $\boldsymbol{z}$ is typically much smaller than that of $\boldsymbol{x}$, this represents an efficient approach to sampling from the posterior of $\boldsymbol{x}$.

The left hand side of (2.17) is an expression for a QoI of the posterior. The right hand side of this equation describes how this QoI may be evaluated by sampling $\boldsymbol{z}$ (instead of $\boldsymbol{x}$) from $p_{\mathcal{Z}}^{\text{post}}$. In practise this is accomplished by generating an MCMC approximation, $p_{\mathcal{Z}}^{\text{mcmc}}(\boldsymbol{z}|\boldsymbol{y}) \approx p_{\mathcal{Z}}^{\text{post}}(\boldsymbol{z}|\boldsymbol{y})$ using the definition in (2.18), and thereafter sampling from this distribution. This circumvents the calculation of the evidence term $p_{\mathcal{Y}}(\boldsymbol{y})$, which would

otherwise be necessary when using (2.18) directly. Using this approach, we conclude that any QoI for the posterior can be approximated as

$$\overline{s(\boldsymbol{x})} \equiv \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{\mathcal{X}}^{\text{post}}}[s(\boldsymbol{x})] \approx \frac{\sum_{n=1}^{N_{\text{samp}}} s(\boldsymbol{g}(\boldsymbol{z}))}{N_{\text{samp}}}, \quad \boldsymbol{z} \sim p_Z^{\text{mcmc}}(\boldsymbol{z}|\hat{\boldsymbol{y}}). \tag{2.20}$$

where N_{samp} is the number of samples. For all the numerical experiments in this paper we have used this approach to evaluate QoIs.

Summary We have described a method for probing the posterior distribution when the prior is approximated by a WGAN. The steps of our algorithm are:

1. Train a WGAN using the sample set $\mathcal{S} \coloneqq \{\boldsymbol{x_i}\}_{i=1}^{N}, \boldsymbol{x_i} \sim p_{\mathcal{X}}(\boldsymbol{x})$ to learn the prior distribution.

2. Reformulate the posterior distribution in the latent space of the WGAN.

3. Run a Markov chain Monte Carlo algorithm to generate samples from this low-dimensional posterior distribution.

4. Use MCMC-generated samples to compute QoIs that quantify the uncertainty in the inference.

In the following section we apply the above algorithm to a variety of physics-driven inverse problems with applications to thermal imaging, elasticity imaging, and microstructure identification where we draw inferences from noisy measurements and quantify uncertainty in these inferences. We consider measurements obtained using synthetic as well as experimental data.

In all cases we use a Wasserstein GAN-GP [59] to learn the prior density We also ensure that the target images are not chosen from the set used to train the GAN. We sample from the posterior using Hamiltonian Monte Carlo (HMC) [61] and implement it using Tensorflow-probability library [62]. We use initial step size of 1.0 for HMC and adapt it following [63]

based on the target acceptance probability. We use 64k samples with burn-in period of 50%. We select these parameters to ensure convergence of chains. Using the HMC sampler we compute the quantities of interest.

2.5 Numerical Results

2.5.1 Inverse heat conduction: Inferring thermal conductivity

We first consider a non-linear coefficient inversion problem for elliptical PDEs. This problem arises in many fields such as subsurface flow modeling [64], electrical impedance tomography [65], inverse heat conduction etc. Depending upon application domain, the interpretation of inferred parameter and measurement takes different meaning. Here we focus on heat conduction application where the goal is to infer the conductivity distribution inside the domain given (partial or noisy) measurement of temperature inside the domain. The exact same approach is applicable to other coefficient inverse problems described above as well. For the current application the forward model is described by the steady-state heat conduction equation:

$$-\nabla \cdot (k(\boldsymbol{s})\nabla u(\boldsymbol{s})) = f(\boldsymbol{s}), \qquad \boldsymbol{s} \in (0,1)^2 \tag{2.21}$$

$$u(\boldsymbol{s}) = 0, \qquad \boldsymbol{s} \in \partial(0,1)^2 \tag{2.22}$$

where $k(\boldsymbol{s})$, $u(\boldsymbol{s})$, and $f(\boldsymbol{s})$ denote thermal conductivity, temperature, and the heat source respectively. The goal of coefficient inversion problem is to recover the thermal conductivity at each location given the noisy (and potentially partial) measurement of temperature.

We consider three different types of dataset $\mathcal{S}$: (i) rectangular dataset: a parametric dataset generated by sampling from an underlying parametric description (explained in detail below), (ii) MNIST: a non-parametric dataset of hand-written digits, and (iii) microstructure:

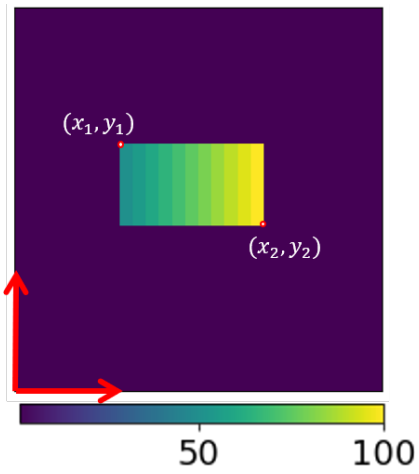


Figure 2.3: Parametric description of rectangular dataset

a non-parametric dataset of a two-phase material generated by solving the Cahn-Hilliard equation.

2.5.1.1 Rectangular dataset

Let $\Omega := [0, 1] \times [0, 1]$ be a bounded domain in $\mathbb{R}^2$ and let $k(x, y)$ and $u(x, y)$ denote the conductivity and temperature value at point $(x, y) \in \Omega$ respectively. In the case of rectangular dataset we consider the conductivity field which is constant everywhere in Ω except in a rectangular region where it varies linearly from left to right.

Figure 2.3 shows parametric description of this dataset. Specifically, there are four parameters which completely describe this dataset of conductivity field. These are:

1. $\xi_1 = x$-coordinate of the top left corner of rectangle

2. $\xi_2 = y$-coordinate of the top left corner of rectangle

3. $\xi_3 = x$-coordinate of the bottom right corner of rectangle

4. $\xi_4 = y$-coordinate of the bottom right corner of rectangle

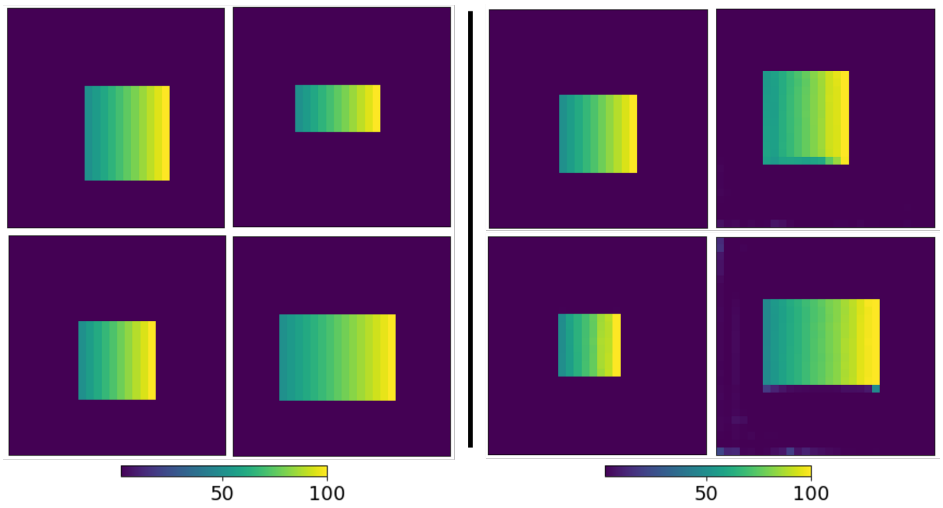


Figure 2.4: *Left*: Samples from true prior density (rectangular training dataset). *Right*: Samples from the learned prior density (from the trained WGAN)

Given these four parameters the conductivity value at any point (x, y) inside the domain Ω can be represented as,

$$k(x,y) = 1 + \mathbb{1}_{\{(x,y):\xi_1 \le x \le \xi_3, \xi_2 \le y \le \xi_4\}}(50 + 50((x-\xi_1)/(\xi_3-\xi_1)) \tag{2.23}$$

We vary the four parameters described above by sampling them from a uniform distribution to generate dataset $\mathcal{S}$ of 10,000 different images of conductivity field of dimension 28x28. Specifically $\xi_1, \xi_2 \sim \mathcal{U}[0.2, 0.4]L$ and $\xi_3, \xi_4 \sim \mathcal{U}[0.6, 0.8]L$, where L is the length of the domain Ω, which is set to 1 unit. Once $\{\xi_i\}_{i=1}^4$ are sampled as above, the conductivity field can be obtained using Eq. (2.23).

Four representative samples from this dataset are shown in Figure 2.4 (left panel). We train WGAN using the dataset $\mathcal{S}$ to learn the prior density. The realizations from the learned prior are shown in Figure 2.4 (right panel), which are qualitatively similar to the true samples. The generator of the trained WGAN is then used to do posterior inference as described in Section 2.4.

In Figure 2.5 we have shown the true thermal conductivity, the measured temperature (with and without noise), the fields inferred using our proposed algorithm (MAP, mean and

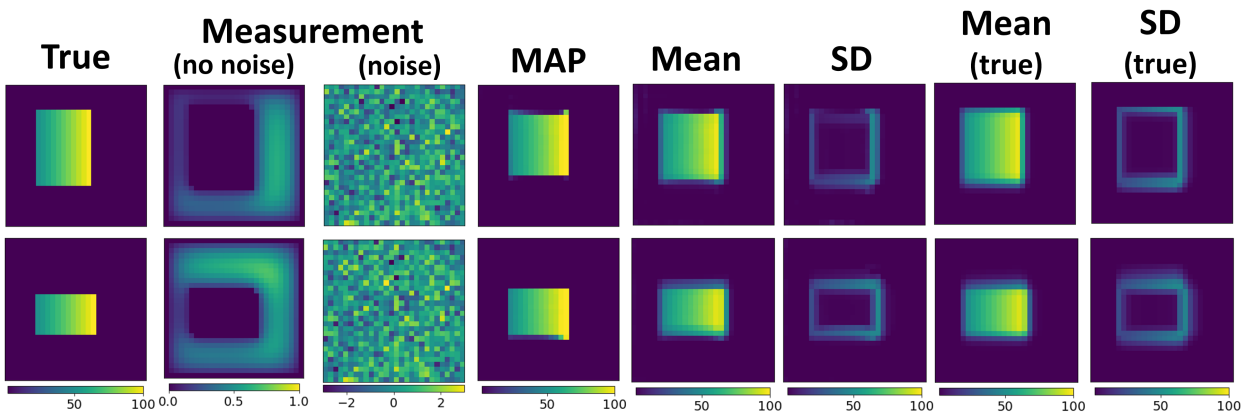


Figure 2.5: Inferring thermal conductivity from noisy temperature measurement for rectangular dataset: Column (1) true conductivity field (2) temperature field (3) noisy version of temperature used as a measurement (4-6) MAP, mean, and pixel-wise standard deviation estimates of GAN-prior.

standard deviation) and the noisy temperature. For the parametric case, the true mean and SD are determined by Monte Carlo sampling in the parametric space. This can be done as follows:

We are interested in computing some statistics $s(\boldsymbol{k})$ with respect to the posterior distribution of conductivity $p(\boldsymbol{k}|\hat{\boldsymbol{u}})$ with $\boldsymbol{k} = \boldsymbol{h}(\boldsymbol{\Xi})$, where $\boldsymbol{\Xi} := (\xi_1, \xi_2, \xi_3, \xi_4)$ and $\boldsymbol{h}$ is as described in Eq. (2.23). $\boldsymbol{\Xi} \sim p(\boldsymbol{\Xi}) = p(\xi_1)p(\xi_2)p(\xi_3)p(\xi_4)$. So,

$$\mathop{\mathbb{E}}_{\boldsymbol{k}\sim p(\boldsymbol{k}|\hat{\boldsymbol{u}})}[s(\boldsymbol{k})] = \mathop{\mathbb{E}}_{\boldsymbol{\Xi}\sim p(\boldsymbol{\Xi})}[s(\boldsymbol{h}(\boldsymbol{\Xi}))\delta(\hat{\boldsymbol{u}} - \boldsymbol{f}(\boldsymbol{h}(\boldsymbol{\Xi})))] \tag{2.24}$$

where $\boldsymbol{f}$ is the forward map described in Eq. (2.21) mapping conductivity $\boldsymbol{k}$ to temperature $\boldsymbol{u}$ and δ represents delta distribution. So, the process of computing "true" statistics shown in Figure 2.5 can be computed by following these steps.

1. Sample $\{\xi_i\}_{i=1}^4$ from their respective prior densities. That is $\xi_1, \xi_2 \sim \mathcal{U}[0.2, 0.4]L$ and $\xi_3, \xi_4 \sim \mathcal{U}[0.6, 0.8]L$ with L=1 and $\boldsymbol{\Xi} \sim p(\boldsymbol{\Xi})$ (Note that together all $\{\xi_i\}_{i=1}^4$ define joint density $p(\boldsymbol{\Xi}) = p(\xi_1)p(\xi_2)p(\xi_3)p(\xi_4)$).

2. Compute $\boldsymbol{k} = \boldsymbol{h}(\boldsymbol{\Xi})$ and $\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{k})$ using Eq. (2.23) and Eq. (2.21) respectively.

3. Compute "true" posterior statistics using Eq. (2.24).

From Figure 2.5 we observe that

- the MAP is close to the true distribution in every case, even in the presence of significant noise;
- the estimated mean and SD are close to the true values for the case where the latter can be determined;
- the SD is large in regions where the conductivity has sharp gradients;

2.5.1.2 MNIST

Next, we consider a non-parametric MNIST dataset [66]. MNIST is a dataset of handwritten digits of dimension 28×28. Let $\Omega := [0,1] \times [0,1]$ be a bounded domain in $\mathbb{R}^2$ and let $k(x,y)$ and $u(x,y)$ denote the conductivity and temperature value respectively at point $(x,y) \in \Omega$. For this study we discretized the domain Ω in 28×28 equally spaced grid points and considered the scaled version of pixel intensity value of MNIST image at (x,y) as conductivity value k at that nodal point. We scaled the MNIST image in such a way that the conductivity value at any point in the domain is between 1 and 10.

Figure 2.6 (left panel) shows four representative samples from the true prior density (training set) of the scaled MNIST conductivity field. We use 50,000 images of this scaled MNIST dataset as our training set $\mathcal{S}$ to train a WGAN. The samples from the trained WGAN are shown in Figure 2.6 (right panel). It is evident from this figure that the generated samples are qualitatively similar to samples from true prior density. The generator of trained WGAN is then used to do posterior inference as described in Section 2.4.

We next choose a sample of conductivity field $\boldsymbol{k}^{test}$ from the test set. We then solve the forward heat conduction problem described in Eq. (2.21) to obtain corresponding temperature field $\boldsymbol{u}^{test}$. We then add Gaussian noise to this temperature field to mimic the real world scenario of noisy measurement i.e. $\hat{\boldsymbol{u}} = \boldsymbol{f}(\boldsymbol{u}) + \boldsymbol{\eta}$. Finally we use the steps described

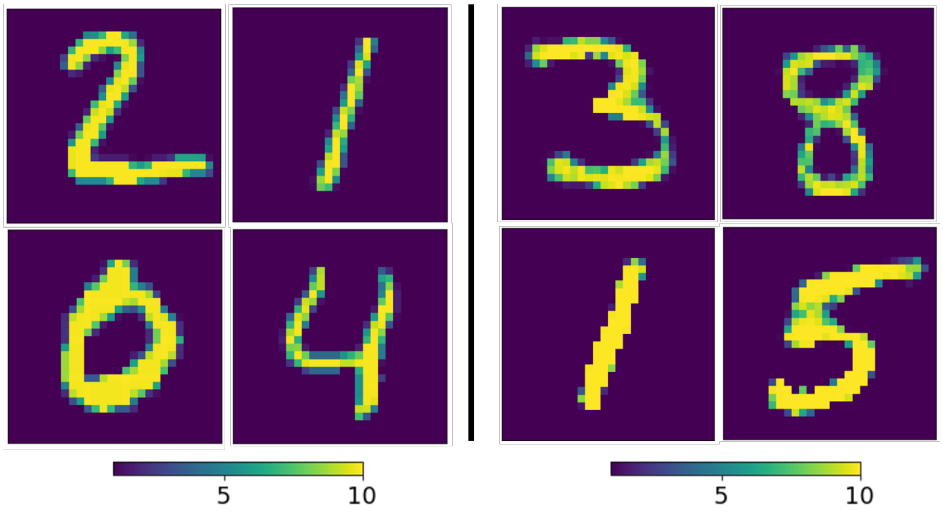


Figure 2.6: *Left panel:* Four representative samples from prior density of scaled MNIST conductivity field. *Right:* Four representative samples from the learned prior density (from the trained WGAN)

in section 2.4 to perform posterior characterization and compute important statistics like $\boldsymbol{k}^{\text{map}}$, $\boldsymbol{k}_{\boldsymbol{mean}}$ and standard deviation.

Figure 2.7 shows inference results for scaled MNIST conductivity dataset. The top row shows result for noisy measurement and the bottom row shows inference results for noisy *and* partial measurement. By comparing the true conductivity field with the mean and the pixel-wise standard deviation estimated by the GAN-based priors we conclude that the GAN-based prior has converged to the true posterior even in the presence of a very high level of measurement noise. The last row of Figure 2.7 further highlights the efficacy of the proposed method for highly ill-posed inverse problems where a significant portion of measurement is not available for inference. The algorithm not only performs excellent reconstruction but also produces a higher level of uncertainty in the locations where measurement is not available.

2.5.1.3 Bi-phase material microstructure

In this study, we considered the more complex non-parametric dataset of the material microstructure of a bi-phase material. The purpose of this study is two fold: (i) to demonstrate

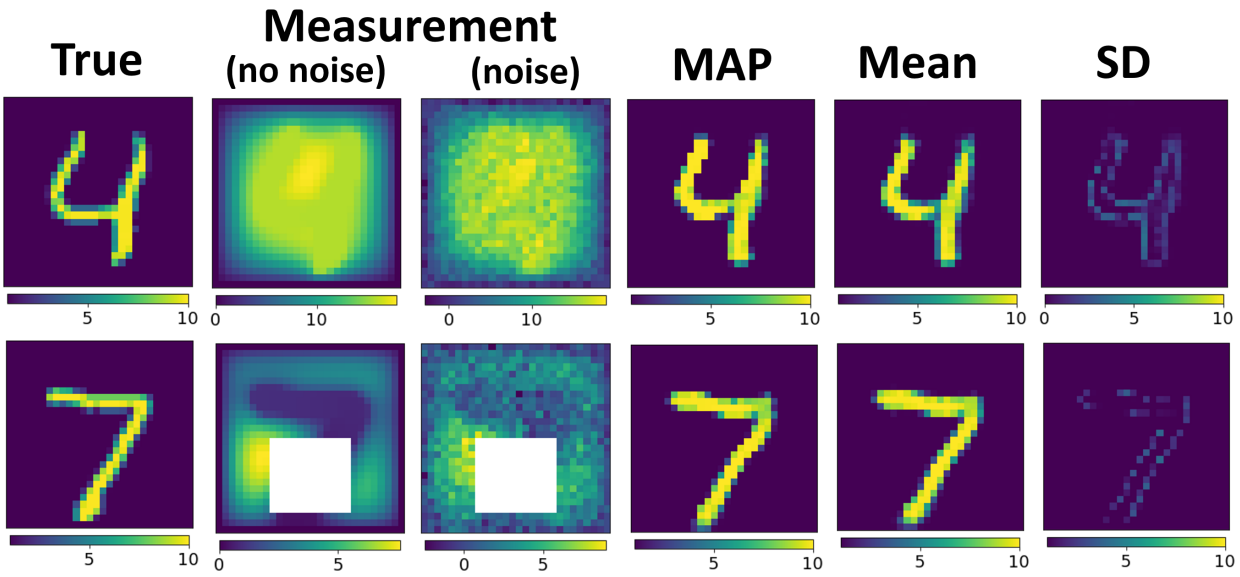


Figure 2.7: Inferring thermal conductivity from noisy temperature measurement for scaled MNIST conductivity dataset: Column (1) true conductivity field (2) temperature field (3) noisy version of temperature used as a measurement (4-6) MAP, mean, and pixel-wise standard deviation estimates of GAN-prior.

the excellent effectiveness of GAN-based priors with highly complex and non-parametric samples of prior and (ii) to solve an interesting and important inverse problem of identifying optimal microstructure of material which produces desired quantity of interest. For the scope of this study, the desired quantity of interest is temperature field $\boldsymbol{u}$, however, any other function of $\boldsymbol{u}$ or conductivity $\boldsymbol{k}$ could also be used in the same fashion. Next we describe the process of generating this dataset.

The given dataset which represents the material microstructure of a bi-phase material at spinodal decomposition stage. Spinodal decomposition occurs when when one thermodynamic phase spontaneously (i.e., without nucleation) separates into two phases. It is observed, for example, when mixtures of metals or polymers separate into two coexisting phases, each rich in one species and poor in the other. The dynamics of the spinodal decomposition are typically modeled using Cahn-Hilliard equation. The Cahn-Hilliard (CH) equation represents the process through which a binary fluid separates spontaneously into

two distinct phases and forms domains pure in each component. If c represents concentration with $c = \pm 1$ indicating domains for each phase, then the CH equation can be written as,

$$\frac{\partial c}{\partial t} = D\nabla^2(c^3 - c - \gamma\nabla^2 c) \tag{2.25}$$

where, D is a diffusion coefficient and $\sqrt{\gamma}$ is the length of transition region between two domains. In the current study we choose the value of $D = 10$ units and $\gamma = 5$ units. We solve the above time-dependent differential equation with initial condition represented as $c_0(x, y) \sim \mathcal{U}\{-1, 1\}$ (where $c_0(x, y)$ indicates concentration value at nodal location (x, y) at time $t = 0$ and $\mathcal{U}\{\cdot, \cdot\}$ indicates discrete uniform distribution with two possible values) by forward Euler method with step size of 0.005 and total number of iterations=15000 (i.e. total time of 75 units). The spatial discretization (laplacian) in above equation was approximated via convolution operation with edge wrap kernel of size 3×3. The physical domain size is $[0, 600] \times [0, 600]$ and is discretized in 600 nodes in each directions. The solution of CH equation gives concentration c at each node. We scale this concentration value to represent conductivity value at each nodal point. This ensures that each phase of bi-material has distinct conductivity value. We choose this value to be 1 and 100 mimicking highly conductive material dispersed in low conductivity material. Once this conductivity field of dimension 600×600 is obtained, we create a sub-dataset of 62,500 images of dimension 64×64 by uniformly cropping this smaller dimension images from a bigger dataset. We work with images of dimension 64×64 instead of 600×600 to ensure that the generator and the discriminator are of moderate size with reasonable number of parameters allowing training of the WGAN on a single GPU.

Figure 2.8 (left panel) shows four representative samples from the training set and the right panel shows samples from the generator of a trained WGAN. This highlights the ability of the WGAN to generate samples from highly complex and non-parametric dataset.

Next, just like previous two studies descibed above, we take a sample of conductivity field from test set and solve the forward heat conduction problem described in Eq. (2.21) to

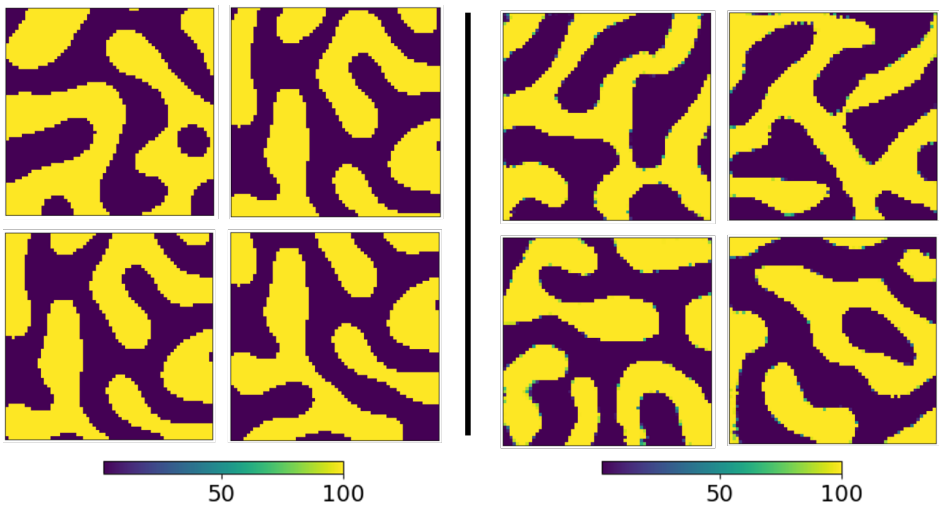


Figure 2.8: *Left panel:* Four representative samples from prior density of bi-phase material microstructure dataset. *Right:* Four representative samples from the learned prior density (from the trained WGAN)

obtain temperature field and then add Gaussian noise to it. Given this noisy measurement then we infer the posterior statistics of conductivity field.

Figure 2.9 shows the results of inference problem with GAN priors. As can be observed even with this highly complex dataset our proposed algorithm is capable of excellent recovery. Furthermore, in the regions of the domain where it makes inaccurate prediction (bottom left column of both rows, for example) it produces high standard deviation values indicating that the reconstruction is less reliable in that region and should be used with caution. This is extremely useful in many downstream tasks where decisions are made based on the output of Bayesian inference.

Another way of interpreting results shown in Figure 2.9 is by interpreting this problem as an inverse design problem where the goal is to find the optimal microstructure for a given objective function/quantity of interest. In this context, the desired quantity of interest in Figure 2.9 is measured temperature field (third column) and the MAP output shows the corresponding optimal microstructure. One can similarly consider more sophisticated quantity of interests as well in the current framework without any major modification.

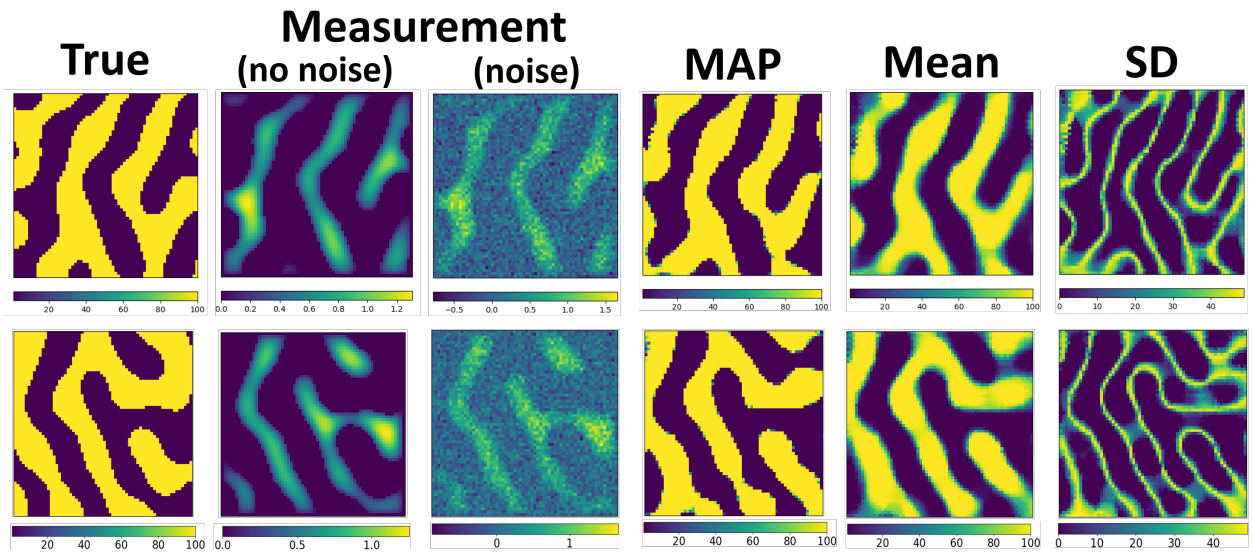


Figure 2.9: Inferring thermal conductivity from noisy temperature measurement for bi-phase material microstructure: Column (1) true conductivity field (2) temperature field (3) noisy version of temperature used as a measurement (4-6) MAP, mean, and pixel-wise standard deviation estimates of GAN-prior.

2.5.2 Elasticity imaging

We provide results for an imaging problem involving experimental measurements. Elasticity imaging is a technique of inferring mechanical properties of tissue from displacement data collected via different medical imaging modalities [67]. The forward problem is given by elasticity problem which solves for the displacement field of an incompressible linear elastic solid.

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{0} \qquad \text{in } \Omega \tag{2.26}$$

$$\boldsymbol{u} = \boldsymbol{u_D} \qquad \text{on } \Gamma_D \tag{2.27}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{\tau} \qquad \text{on } \Gamma_N \tag{2.28}$$

where, $\boldsymbol{\sigma} = 2\boldsymbol{\mu}(\nabla^s \boldsymbol{u} + (\nabla \cdot \boldsymbol{u})\mathbb{1})$ for plane-stress incompressible linear elastic solid, $\boldsymbol{\mu} \in^{N \times N}$ is the discretized shear modulus and $\boldsymbol{u} \in^{N \times N \times 2}$ is the discretized displacement field.

We are interested in recovering the shear modulus field $\boldsymbol{\mu}$, given a noisy observation of displacement field $\boldsymbol{u}$. For this study, we used experimental data obtained from a physical phantom experiment [68]. The phantom was manufactured from a mixture of gelatin, agar, and oil and contained a spherical inclusion with an elevated shear modulus compared to the background. The phantom was subjected to uni-axial loading and the interior deformation was measured using ultrasound.

The sample set $\mathcal{S}$ contained 3,000 images of elliptical inclusions centered around different locations inside discretized into 56^2 grid points. The ratio of the shear modulus of inclusion to that of the background was varied between 1:1 and 8:1 to account for a wide range of possibilities. A WGAN was trained using this sample set and the learned GAN prior was used in Bayesian inference in conjunction with the experimentally measured displacement field (shown in the leftmost panel of Figure 2.10).

Table 2.1: Reconstruction results of elastography study

Quantity of Interest	True value	MAP-based reconstruction	Mean-based reconstruction
Diameter of inclusion	10 mm	10.22 ± 0.49 mm	10.06 ± 0.27 mm
Avg. value of SM inside inclusion	10.7 kPa	10.3 ± 0.35 kPa	10.5 ± 0.34 kPa
Vertical distance of inclusion from top	35 mm	34.4 ± 0.85 mm	35.1 ± 0.3 mm

The reconstruction results are shown in the middle panel of Figure 2.10 and reveal the circular inclusion. These results compare very well with the true physical quantities for the inclusion (Table 2.1). Since the measurement data for this study was obtained experimentally, we do not know the true measurement noise. Given this, we tested the robustness of our algorithm with different assumed values of measurement noise (different variance of additive Gaussian noise). As shown in the Figure 2.10, the algorithm produces consistent reconstruction results across different noise values, with elevated pixel-wise standard deviation for the case where higher value of noise is assumed. The mean and the standard deviation reported in the quantitative results shown in Table 2.1 are computed using reconstruction results obtained from all three different assumed noise levels.

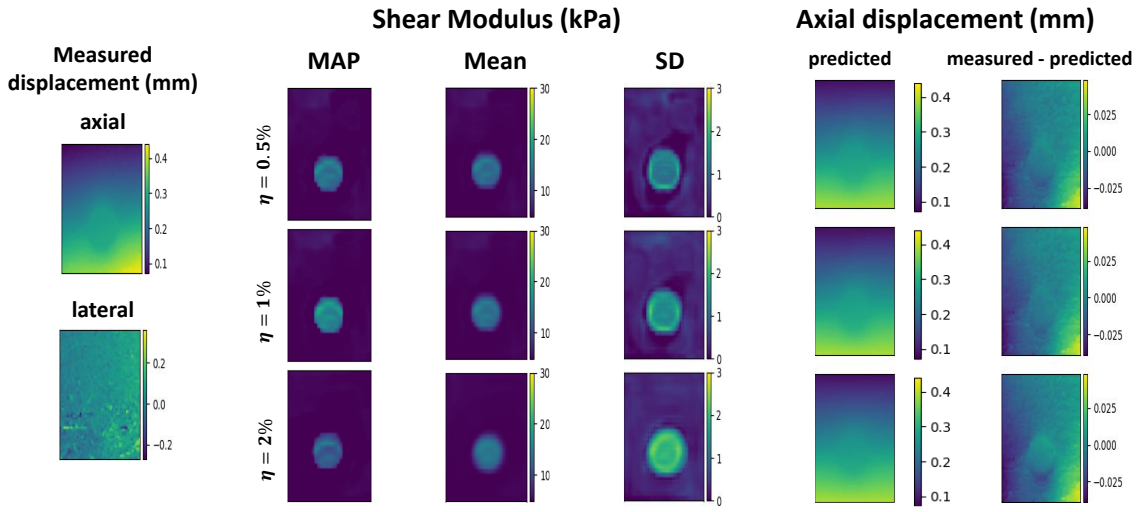


Figure 2.10: Recovery of shear modulus field from noisy measurements of the displacement field.

Chapter 3

Quantifying Uncertainty in Supervised Learning

> Although our intellect always longs for clarity and certainty, our nature finds uncertainty fascinating.
>
> ON WAR
>
> *Karl Von Clausewitz*

3.1 Introduction

Quantifying uncertainty in an inference problem amounts to making a prediction and quantifying the confidence in that prediction. With the recent adoption of machine learning and computational modeling in high-stakes applications such as medical diagnosis [23, 69, 70], climate modeling [71, 72], autonomous driving [73, 74], and finance [75, 76], it has become increasingly important to develop algorithms that can quantify uncertainty in an inference.

The knowledge of uncertainty in a prediction can directly influence the downstream action that depends on the inference. Consider the example of an autonomous vehicle that uses machine learning models for image segmentation, inpainting, and localization to extract useful features directly from raw sensory inputs [77]. The output of these models is then fed

Portions of this chapter are under review as: D. V. Patel and A. A. Oberai, "GAN-based priors for quntifying uncertainty in supervised learning," SIAM Journal of Uncertainty Quantification.

to a higher-level decision making system, which is either based on fixed set of rules ("if there is a vehicle in front, slow down") or on learning principles such as reinforcement learning. In this framework, mistakes made by lower level components, for example, incorrectly reconstructing a 10 mph speed limit sign as 70 mph, can propagate through the decision making system and lead to devastating outcomes. Similar examples can be drawn from other areas like medical imaging, high frequency trading and natural sciences [78].

The knowledge of uncertainty can also be useful in determining the optimal location of a sensor. Consider an image recovery problem, where the goal is to infer the signal, and associated uncertainty, using limited measurement data. In this problem a user can leverage information about the spatial distribution of uncertainty to choose the location with maximum uncertainty as next measurement location. This task falls within the fields of active learning and/or design of experiments [79] and is particularly useful in applications where each measurement requires significant time and/or resources.

In this chapter we focus on quantifying uncertainty in supervised learning problems, where pair-wise instances of the vector of input features, $\boldsymbol{x} \in \mathbb{R}^P$, and the vector of the quantities to be inferred, $\boldsymbol{y} \in \mathbb{R}^N$, are used to train a model. Thereafter for a given noisy measurement of the input features, denoted by $\hat{\boldsymbol{x}}$, the distribution of $\boldsymbol{y}$ is desired. To fix ideas the reader might think of $\boldsymbol{x}$ as an image, and $\boldsymbol{y}$ as the corresponding label class; though many other interpretations are possible. There has been significant work on quantifying uncertainty for such problems using techniques of machine learning in general, and deep learning in particular. A convenient way to classify these approaches is to consider whether in the learning phase, a given approach learns the conditional distribution $p_Y(\boldsymbol{y}|\hat{\boldsymbol{x}})$ or the joint distribution $p_U(\boldsymbol{u})$, where $\boldsymbol{u} = [\boldsymbol{x}, \boldsymbol{y}]$.

There is substantially more work on deep learning methods that learn the conditional distribution $p_Y(\boldsymbol{y}|\hat{\boldsymbol{x}})$. A common theme among these is treating the parameters of the neural network as random variables and applying Bayes' rule to infer their posterior distribution given the training data. Then, for a new input, a prediction can be made by marginalizing

this posterior distribution [80, 81]. For many applications this marginalization is intractable due to the high dimension of the integral and various approximations of the posterior have been proposed. These include using a Laplace approximation [81], using the Hamiltonian Monte Carlo method for efficient sampling [82–84], and using a variational method to approximate the posterior with a Gaussian distribution [85, 86]. These variational methods have recently been combined with the re-paramaterization trick leading to a popular class of algorithms called variational autoencoders [47, 87–90]. In [91], authors make an interesting connection between MC dropout and Gaussian processes and demonstrate how MC dropout in the forward mode acts as a variational approximation of Gaussian processes. These ideas, which were initially proposed for image classification, have been recently extended to other image inference tasks [92, 93]. Another approach for uncertainty quantification involves generating ensembles of deep networks that are randomized for example with different initial guess for the weights [94]. Yet another approach treats only the weight in last layer of the network as stochastic and then determines these through a stochastic inference-based model [95].

In contrast to the methods described above, there are those that aim to learn and work with the joint distribution $p_U(\boldsymbol{u})$, where $\boldsymbol{u} = [\boldsymbol{x}, \boldsymbol{y}]$, and use this model to make inferences [96, 97]. The advantage of learning the joint distribution is its generality. Consider an algorithm that learns $p_U(\boldsymbol{u})$ in the training phase. Then given a noisy measurement $\hat{\boldsymbol{x}}$, it infers the conditional distribution $p_U(\boldsymbol{u}|\hat{\boldsymbol{x}})$. Now if the user wishes to solve another inference problem where $\boldsymbol{x}$ and $\boldsymbol{y}$ now refer to different set of components of $\boldsymbol{u}$, then the process of learning $p_U(\boldsymbol{u})$ does not have to be repeated. That is, once the model has learnt the joint distribution, it can be used for multiple different downstream inference tasks. Consider for example the case described in Section 3.3, where $\boldsymbol{u}$ is the vector of the nodal values of pressure and permeability in a nonlinear Darcy flow problem. Then, once p_U is learned, it can be used to solve the forward problem of determining pressure, given the permeability, by selecting $\boldsymbol{x}$ to be permeability and $\boldsymbol{y}$ to be pressure. It can also be used to solve the inverse

problem of determining permeability, given pressure, by selecting $\boldsymbol{x}$ to be pressure and $\boldsymbol{y}$ to be permeability. It can also be used to solve a "mixed" problem where partial measurements of both pressure and permeability are made. Clearly, this is not true if the model learns only the conditional distribution.

There is relatively little work on modeling the joint distribution, $p_U(\boldsymbol{u})$, to perform uncertainty quantification in deep learning. Most of this work falls into the category of hybrid algorithms where the the joint distribution is split into conditional and marginal distributions, $p_U(\boldsymbol{u}) = p_Y(\boldsymbol{y}|\boldsymbol{x})p_X(\boldsymbol{x})$ [98, 99]. In doing so, these algorithms use deep generative algorithms like normalizing flows to learn the marginal distribution, and generalized linear models or residual networks to learn the conditional distribution. In contrast to this, in this chapter we propose a truly "joint" algorithm that directly learns $p_U(\boldsymbol{u})$ from i. i. d. samples using generative adversarial networks (GANs). By directly learning this joint distribution with a single model we hope to accommodate more complex relations between $\boldsymbol{x}$ and $\boldsymbol{y}$ as we do not make any assumption about the conditional distribution $p_Y(\boldsymbol{y}|\boldsymbol{x})$. Once this joint distribution is learned, and we are given a noisy measurement $\hat{\boldsymbol{x}}$, we use Bayes' rule to infer $p_U(\boldsymbol{u}|\hat{\boldsymbol{x}})$. In doing so, we utilize the strength of a GAN in two ways. First, we use it to learn the complex prior distribution of $\boldsymbol{u}$ from its i. i. d. samples. Second, we use it as a tool for dimension reduction by mapping the Bayesian inference problem to the latent space for the GAN, which is typically of much smaller dimension than the dimension of $\boldsymbol{u}$, which allows us to perform efficient posterior sampling.

We note that there is a growing body of work focused on using GANs and other deep generative models to solve inverse/inference problems where the forward model is explicitly known [38, 100–102], typically through some physics-based principles. This includes the use of generative models for determining the posterior distribution in a Bayesian framework [103, 104] and for sampling conditional distributions [105–108]. In contrast to this in the problems considered in this chapter the knowledge of the forward model is not assumed and is inferred from the pairwise data instances.

Our Contribution

In this chapter we focus on the problem of quantifying uncertainty in supervised learning problems and make following contributions:

- We propose a novel way of performing uncertainty quantification using Bayesian inference involving priors without explicit analytical expressions and high dimensional posterior distributions. We utilize the approximate joint distribution learned by a GAN as a prior in a Bayesian update and reformulate the inference problem in the low-dimensional latent space of the GAN to achieve this goal.

- We demonstrate how uncertainty information can be useful in detecting out-of-distribution samples and in active learning/design of experiments for computer vision problems, and problems driven by physical models and processes.

The layout of the remainder of this chapter is as follows. In the following section we describe the problem of interest and develop the new method of quantifying uncertainty for supervised learning problems. Thereafter, in Section 3.3 we apply this method to develop novel algorithms for image classification and image inpainting, and physics-based inference, with quantitative measures of uncertainty. We also demonstrate the utility of quantifying uncertainty in the detection of out-of-distribution (OOD) samples and in active learning. We end with conclusions in Section 3.7.

3.2 Problem Formulation

We consider the problem where we wish to infer the vector $\boldsymbol{y} \in \Omega_Y \subset \mathbb{R}^N$ from the noisy measurement of a related vector $\boldsymbol{x} \in \Omega_X \subset \mathbb{R}^P$, where the domains Ω_Y and Ω_X are closed and bounded. We denote by $\boldsymbol{u} = [\boldsymbol{x}, \boldsymbol{y}] \in \Omega_U \equiv \Omega_Y \times \Omega_X \subset \mathbb{R}^Q$, (where $Q = N + P$), the joint vector and recognize that the measurement has the form

$$\hat{\boldsymbol{x}} = \mathbb{1}_{\boldsymbol{x}}\boldsymbol{u} + \boldsymbol{\eta}, \tag{3.1}$$

where $\mathbb{1}_{\boldsymbol{x}}$ extracts components of $\boldsymbol{x}$ from $\boldsymbol{u}$, and $\boldsymbol{\eta}$ is the noise vector drawn from the distribution p_η. Further we assume that we have access to the sample set $\mathcal{S} = \{\boldsymbol{u}^{(1)}, \cdots, \boldsymbol{u}^{(S)}\}$ which contains multiple realizations of $\boldsymbol{u}$ drawn from the distribution p_U. Our goal is to use the prior information encoded in $\mathcal{S}$ and the new, noisy measurement $\hat{\boldsymbol{x}}$ to determine the distribution for the corresponding vector $\boldsymbol{y}$, and perhaps also the distribution for $\boldsymbol{x}$ (the de-noised version of $\boldsymbol{x}$). We note that learning priors from data has rich history and has been successfully used in various physics-driven inference problems [109–111].

Since the prior information is built from the samples of joint distribution p_U, this class of problems is one of supervised learning where training requires *pair-wise* instances of $\boldsymbol{x}$ and $\boldsymbol{y}$. An example problem in this class is that of image classification, where $\boldsymbol{x}$ represents an image and $\boldsymbol{y}$ represents the corresponding one-hot encoded label vector. Another example is that of image inpainting, where $\boldsymbol{x}$ represents the portion of an image that is revealed and $\boldsymbol{y}$ represents the portion that is occluded.

The method developed in this chapter utilizes the approximate distribution learned by a generative adversarial network (GAN) as a prior in a Bayesian update. It relies on the ability of a GAN to learn a density from i. i. d. samples drawn from $p_U(\boldsymbol{u})$. In the section below we provide a brief introduction to GANs and show the weak convergence of the distribution learned by the GAN to true data distribution under certain assumptions. Thereafter, in Section 3.2.2 we utilize this result to develop a method for quantifying uncertainty in supervised learning tasks.

3.2.1 Generative Adversarial Networks

GANs [48] are a class of deep generative models, which are trained in an adversarial fashion and are used to generate samples from a distribution p_U^g which approximates some target distribution p_U. Typically, GAN comprise of a generator $\boldsymbol{g}$ that maps a latent vector $\boldsymbol{z} \in \Omega_Z \subset \mathbb{R}^M$ to $\boldsymbol{u} \in \Omega_U$, where typically, $M \ll Q$. The components of the latent vector are selected from a simple distribution, typically a Gaussian or a uniform distribution. The

generator up-scales these components through successive application of non-linear transformations at each layer of a neural network. In particular, the generator is a function $\boldsymbol{g}(\boldsymbol{z}, \boldsymbol{\theta})$, $\boldsymbol{g} : \Omega_Z \times \mathbb{R}^{N_\theta} \mapsto \Omega_U$ which is approximated by a neural network with weight parameters $\boldsymbol{\theta}$ whose values are determined during the training process described below. The number of weights, N_θ, is a measure of the capacity of the generator which increases with increasing N_θ.

The other component of a GAN is a discriminator function, which is also approximated by successive non-linear transformations in the form of a neural network. However, these transformations are designed to down-scale the original input. The final few layers of the discriminator are fully connected and lead to a single scalar-valued field. Thus the discriminator, $d(\boldsymbol{u}, \boldsymbol{\phi})$, $d : \Omega_U \times \mathbb{R}^{N_\phi} \mapsto \mathbb{R}$. Here $\boldsymbol{\phi}$ is the vector of the weights of the discriminator, and the number of weights, N_ϕ, is a measure of the capacity of the discriminator. The discriminator is trained so that it attains large positive values for inputs selected from the true distribution p_U and small values for inputs generated by the generator. This is made precise in the description of the objective function used to train the GAN, which is described below.

The generator and the discriminator are trained in an adversarial manner. The training data for the discriminator is comprised of the set of real instances of $\boldsymbol{u}$, sampled from p_U, and a set of "fake" instances generated by the generator, $\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{z}, \boldsymbol{\theta})$, with $\boldsymbol{z}$ sampled from p_Z (an easy-to-sample-from distribution. Typically, a Gaussian distribution with zero mean and identity covariance matrix). The weights of the discriminator are determined by requiring it assume large values for the true instances of $\boldsymbol{u}$ and small values for the "fake" instances. On the other hand, the weights of the generator are determined by passing its output through the discriminator and requiring it to be considered as "real." Thus while the generator is trained to "fool" the discriminator, the discriminator is trained so as not to be fooled by the generator. Different types of GANs can be obtained by appropriately selecting the objective function within these broad guidelines. In fact, the training objective of several GANs can

be interpreted as the variational minimization of an appropriate divergence [57]. In this work, we work with the Wasserstein GAN [58, 59] which is described below.

The objective function for the Wasserstein GAN (WGAN) is given by

$$L(\boldsymbol{\theta}, \boldsymbol{\phi}) \equiv \mathop{\mathbb{E}}_{\boldsymbol{u}\sim p_U}[d(\boldsymbol{u}, \boldsymbol{\phi})] - \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_Z}[d(\boldsymbol{g}(\boldsymbol{z}, \boldsymbol{\theta}), \boldsymbol{\phi})]. \tag{3.2}$$

The discriminator is trained to maximize this objective function, while the generator is trained to minimize it. This leads to the following min-max problem to determine the optimal values of the weights denoted by $(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*)$,

$$(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*) = \operatorname*{argmin}_{\boldsymbol{\theta}}(\operatorname*{argmax}_{\boldsymbol{\phi}}(L(\boldsymbol{\theta}, \boldsymbol{\phi}))), \tag{3.3}$$

under the constraint

$$\|d(\boldsymbol{u}, \boldsymbol{\phi})\|_{\text{Lip}} \leq 1. \tag{3.4}$$

In the original work on WGANs [58] this inequality constraint was approximately imposed by clipping the weights of the discriminator. This approach was then improved by replacing the inequality constraint with an equality constraint on the gradient of the discriminator with respect to $\boldsymbol{u}$ [59], and this version was referred to as WGAN-GP (gradeint penalty). In practise this constraint was enforced weakly through a penalty term that was added to the loss function for the generator. In our experiments we use this version of the WGAN.

In the limit of infinite capacity, that is $N_\theta, N_\phi \to \infty$, the discriminator $d(\boldsymbol{x})$, $d : \Omega_U \mapsto \mathbb{R}$, and the generator $\boldsymbol{g}(\boldsymbol{z})$, $\boldsymbol{g} : \Omega_Z \mapsto \Omega_U$ can represent all continuous bounded functions, $\mathcal{C}_b$, over their respective domains. In this limit the finite-dimensional min-max is problem is placed by its infinite-dimensional counterpart:

$$(d^*, \boldsymbol{g}^*) = \underset{\boldsymbol{g}\in\mathcal{C}_b}{\operatorname{argmin}} \Big(\underset{d\in\mathcal{C}_b}{\operatorname{argmax}} \big(\underset{\boldsymbol{u}\sim p_U}{\mathbb{E}}[d(\boldsymbol{u})] - \underset{\boldsymbol{z}\sim p_Z}{\mathbb{E}}[d(\boldsymbol{g}(\boldsymbol{z}))] \big) \Big), \tag{3.5}$$

under the constraint

$$\|d(\boldsymbol{u})\|_{\text{Lip}} \leq 1. \tag{3.6}$$

The term within the large parenthesis in (3.5) is precisely the Wasserstein-1 distance [60]. Therefore, we may write,

$$\boldsymbol{g}^* = \underset{\boldsymbol{g}\in\mathcal{C}_b}{\operatorname{argmin}} W_1(p_U, \boldsymbol{g}_{\#}p_Z), \tag{3.7}$$

where W_1 is the Wasserstein-1 distance, and $\boldsymbol{g}_{\#}p_Z$ is the push-forward of p_Z by $\boldsymbol{g}$. In the Wasserstein-1 distance, the convergence of a sequence of probability densities, implies weak convergence [60]. Therefore, if $\boldsymbol{g}^*$ is the limit of a class of generators, $\boldsymbol{g}$, for which $W_1(p_U, \boldsymbol{g}_{\#}p_Z) \to 0$, then for all continuous bounded functions $f \in \mathcal{C}_b(\Omega_U)$, we have,

$$\underset{\boldsymbol{u}\sim p_U}{\mathbb{E}}[f(\boldsymbol{u})] = \underset{\boldsymbol{z}\sim p_Z}{\mathbb{E}}[f(\boldsymbol{g}^*(\boldsymbol{z}))]. \tag{3.8}$$

In the following section we make use of this equality and demonstrate how WGAN may be used to solve the desired inference problem.

It is worth noting that (3.8), is valid only when the optimization problem (3.5)-(3.6) is solved exactly, and when the generator and discriminator networks have infinite capacity. When training a WGAN in practice neither of these conditions are satisfied. This is because the expectations in (3.5) are approximated by finite sums over the number of real and fake samples, and the $\boldsymbol{g}(\boldsymbol{z})$ and $d(\boldsymbol{x})$ are approximated with neural networks which are constructed using a finite number of weights. This introduces an error in (3.8). Quantifying this error

is important in understanding the convergence properties of WGAN and WGAN-GP, and therefore the performance of our method. We point the reader to [112, 113] for recent work on this topic.

In deriving (3.8) we have assumed that $W_1(p_U, \boldsymbol{g}^*_\# p_Z) = 0$. It is useful to consider when this might be the case. Clearly, this can happen when p_U is supported on some M-dimensional manifold (also see [114] for a discussion). It can also hold if the generator produces space-filling curves. As an example, consider the space-filling and measure-preserving Peano curves. As described in [115], these curves have the following properties:

1. They are a map $\boldsymbol{g} : \Omega_Z \mapsto \Omega_U$, where Ω_Z is the unit interval in one dimension and Ω_U is the unit interval in N-dimensions.

2. They are Lipschitz continuous of order $1/N$.

3. They are measure-preserving. That is for any integrable function $f(\boldsymbol{u})$, $\int_{\Omega_Z} f(\boldsymbol{g}(\boldsymbol{z}))d\boldsymbol{z} = \int_{\Omega_U} f(\boldsymbol{u})d\boldsymbol{u}$.

Thus if the generator is able to generate Peano curves, then for the special case of uniform latent and target distributions, we can have $M = 1$ and Q can be an arbitrary large finite integer.

3.2.2 Quantifying Uncertainty in a Supervised Learning Task

We now return to the inference problem and propose a method for solving it using the generator of a WGAN as a prior. Using Bayes' rule we may write the posterior distribution of $\boldsymbol{u} = [\boldsymbol{x}, \boldsymbol{y}]$ as,

$$\begin{aligned} p_U^{\text{post}}(\boldsymbol{u}|\boldsymbol{x}) &= \frac{1}{\mathbb{Z}} p^{\text{l}}(\boldsymbol{x}|\boldsymbol{u}) p_U^{\text{prior}}(\boldsymbol{u}) \\ &= \frac{1}{\mathbb{Z}} p_\eta(\hat{\boldsymbol{x}} - \mathbb{1}_{\boldsymbol{x}}(\boldsymbol{u})) p_U(\boldsymbol{u}), \end{aligned} \tag{3.9}$$

where p^{l} is the likelihood of $\boldsymbol{x}$ given $\boldsymbol{u}$, $\mathbb{Z}$ is the prior-predictive distribution of $\boldsymbol{u}$. This term is also referred to as model evidence, and for a given measurement $\hat{\boldsymbol{x}}$, evaluates to a constant. In deriving the second line of the equation above we have made use of the expression for the measurement (3.1) and chosen p_U as the prior.

We consider a WGAN whose generator has converged to a distribution that is weakly equivalent to p_U. As before, we let $\boldsymbol{z} \sim p_Z(\boldsymbol{z})$ characterize the latent vector space of this GAN, and let $\boldsymbol{g}^*(\boldsymbol{z})$ denote its generator. Then choosing $f(\boldsymbol{u}) = \frac{l(\boldsymbol{u})p_\eta(\hat{\boldsymbol{x}} - \mathbb{1}_{\boldsymbol{x}}(\boldsymbol{u}))}{\mathbb{Z}}$, and assuming that both $p_\eta, l \in \mathcal{C}_b(\Omega_U)$, from (3.8) we have

$$\mathop{\mathbb{E}}_{\boldsymbol{u}\sim p_U}[\frac{l(\boldsymbol{u})p_\eta(\hat{\boldsymbol{x}} - \mathbb{1}_{\boldsymbol{x}}(\boldsymbol{u}))}{\mathbb{Z}}] = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_Z}[\frac{1}{\mathbb{Z}}p_\eta(\hat{\boldsymbol{x}} - \mathbb{1}_{\boldsymbol{x}}(\boldsymbol{g}^*(\boldsymbol{z})))l(\boldsymbol{g}^*(\boldsymbol{z}))]. \tag{3.10}$$

Now making use of (3.9) on the LHS of the equation above, we have

$$\mathop{\mathbb{E}}_{\boldsymbol{u}\sim p_U^{\mathrm{post}}}[l(\boldsymbol{u})] \quad = \quad \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_Z^{\mathrm{post}}}[l(\boldsymbol{g}^*(\boldsymbol{z}))], \tag{3.11}$$

where

$$p_Z^{\mathrm{post}}(\boldsymbol{z}|\boldsymbol{x}) \equiv \frac{1}{\mathbb{Z}}p_\eta(\hat{\boldsymbol{x}} - \mathbb{1}_{\boldsymbol{x}}(\boldsymbol{g}^*(\boldsymbol{z})))p_Z(\boldsymbol{z}). \tag{3.12}$$

Equation (3.11), which is the main result of this chapter, implies that sampling from the posterior distribution of $\boldsymbol{u}$ is equivalent to sampling from the posterior distribution for $\boldsymbol{z}$ and passing the sample through the generator $\boldsymbol{g}^*$. That is,

$$\boldsymbol{u} \sim p_U^{\mathrm{post}}(\boldsymbol{u}|\boldsymbol{x}) \Rightarrow \boldsymbol{u} = \boldsymbol{g}^*(\boldsymbol{z}), \boldsymbol{z} \sim p_Z^{\mathrm{post}}(\boldsymbol{z}|\boldsymbol{x}). \tag{3.13}$$

Since the dimension of $\boldsymbol{z}$ is typically much smaller than that of $\boldsymbol{u}$, this represents an efficient approach to sampling from the posterior of $\boldsymbol{u}$.

The left hand side of (3.11) is an expression for a quantity of interest (QoI) of the posterior. The right hand side of this equation describes how this QoI may be evaluated by sampling $\boldsymbol{z}$ (instead of $\boldsymbol{u}$) from p_Z^{post}. In practise this is accomplished by generating an MCMC approximation, $p_Z^{\text{mcmc}}(\boldsymbol{z}|\boldsymbol{x}) \approx p_Z^{\text{post}}(\boldsymbol{z}|\boldsymbol{x})$ using the definition in (3.12), and thereafter sampling from this distribution. This circumvents the calculation of the prior-predictive distribution of $\boldsymbol{u}$ (denoted by $\mathbb{Z}$), which would otherwise be necessary when using (3.12) directly. Using this approach, we conclude that any QoI for the posterior can be approximated as

$$\overline{l(\boldsymbol{u})} \equiv \mathop{\mathbb{E}}_{\boldsymbol{u}\sim p_U^{\text{post}}}[l(\boldsymbol{u})] \approx \frac{\sum_{n=1}^{N_{\text{samp}}} l(\boldsymbol{g}(\boldsymbol{z}))}{N_{\text{samp}}}, \quad \boldsymbol{z} \sim p_Z^{\text{mcmc}}(\boldsymbol{z}|\boldsymbol{x}). \tag{3.14}$$

where N_{samp} is the number of samples. For all the numerical experiments in this paper we have used this approach to evaluate QoIs.

In practise, the MCMC approximation of the posterior, the numerical approximation of the forward map, and the WGAN approximation of the prior, all introduce perturbations in the different components of posterior distribution in the Bayesian update. A natural question to consider is whether the posterior distribution is stable with respect to these perturbations. As described in [116] the answer depends on the type of problem being solved (discrete or continuous), the amount of observed data, and the metric used to bound the perturbations. Depending on these choices, there are cases where the inference in robust and those where it is brittle [117].

We note that this approach allows us to compute QoIs for the entire vector $\boldsymbol{u}$, which includes the vector $\boldsymbol{y}$, which is not observed, as well as the vector $\boldsymbol{x}$, of which a noisy measurement, $\hat{\boldsymbol{x}}$, is available. While it is clear that parameters related to $\boldsymbol{y}$ are useful, in some instances it is also useful to estimate QoIs related to $\boldsymbol{x}$. A case in point is the image classification problem considered in the following section. In this problem $\hat{\boldsymbol{x}}$ represents the input image and $\boldsymbol{y}$ represents its label. Here computing parameters associated with $\boldsymbol{y}$ provide information about the label for an image. In addition, computing $\boldsymbol{x}^{\text{map}}$ is useful since large

values of the quantity $\|\boldsymbol{x}^{\text{map}} - \hat{\boldsymbol{x}}\|$, which measures the distance between the mode of the posterior distribution and the input image, are strongly correlated with input images that lie outside of the range of the prior, thus enabling the detection of out of distribution (OOD) samples.

Summary We have described a method for probing the posterior distribution when the prior is approximated by a WGAN. The steps of our algorithm are:

1. Train a WGAN using the sample set $\mathcal{S}$ to learn the prior distribution.
2. Reformulate the posterior distribution in the latent space of the WGAN.
3. Run a Markov chain Monte Carlo algorithm to generate samples from this low-dimensional posterior distribution.
4. Use MCMC-generated samples to compute QoIs that quantify the uncertainty in the inference.

In the following section we apply the above algorithm to a broad class of problems where we draw inferences from noisy measurements and quantify uncertainty in these inferences. Wherever possible, we compare our predictions with related methods and/or benchmark solutions and also highlight the role of uncertainty quantification in downstream tasks. In Appendix 3.4, we also derive a computationally efficient approach for estimating the MAP for the posterior density of the latent vector under the assumptions of Gaussian noise and prior.

3.3 Experiments

In this section we show numerical experiments in two widely different application domains: computer vision and computational physics. For experiments in computer vision, we apply our method to two broad classes of problems: image classification and image in-painting.

For experiment in computational physics, we demonstrate how our method can be used to solve forward, inverse and mixed problems with quantified uncertainty estimates. In each case we apply our method to determine important QoIs that include $\boldsymbol{y}^{\text{mean}}$, $\boldsymbol{y}^{\text{map}}$, $\text{var}(\boldsymbol{y})$, and $\|\hat{\boldsymbol{x}} - \boldsymbol{x}^{\text{map}}\|$. Thereafter, we use these QoIs to answer important questions like: Is the input to the inference problem consistent with the prior data it was trained on? Do we have confidence in the inference? How do we utilize this knowledge in order to design the next measurement?

In all cases we use a Wasserstein GAN-GP [59] to learn the prior density (architecture described in Appendix 3.6). We also ensure that the target images are not chosen from the set used to train the GAN. We sample from the posterior using Hamiltonian Monte Carlo (HMC) [61] and implement it using Tensorflow-probability library [62]. We use initial step size of 1.0 for HMC and adapt it following [63] based on the target acceptance probability. We use 64k samples with burn-in period of 50%. We select these parameters to ensure convergence of chains. Using the HMC sampler we compute the quantities of interest.

3.3.1 Image classification

The objective of this task is to infer the label $\boldsymbol{y}$ along with its uncertainty for a given input image $\hat{\boldsymbol{x}}$. This predictive uncertainty estimation is crucial in deep learning applications where high-stakes decisions are made based on the output of a model [118]. It has been shown that in real-world scenarios, where a model might encounter inputs that are anomalous to its training data distribution, many models produce overconfident predictions [119] raising serious concerns about AI safety [120]. In this situation, it is desirable that such out-of-distribution (OOD) data points are detected upfront before making any prediction. A useful probabilistic predictive model should therefore flag all OOD data points, maintain high levels of accuracy on in-distribution data points, and provide a measure of confidence in its predictions.

In order to achieve this goal, we compute three different quantities: $\|\hat{\boldsymbol{x}} - \boldsymbol{x}^{\text{map}}\|$ for OOD detection, $\boldsymbol{y}^{\text{mean}}$ for prediction, and $\text{var}(\boldsymbol{y})$ as a measure of confidence in the prediction.

We consider the MNIST [121] database of hand-written digits and use 55k images and the corresponding labels to train a WGAN-GP. We note that the dimension of $\boldsymbol{u}$ is 794 whereas the dimension of $\boldsymbol{z}$ is 100, giving a reduction by almost a factor of 8. Thereafter, we use the MNIST test set to test the performance of our algorithm for in-distribution data, and NotMNIST[1] test set for OOD data. Our approach of learning and inferring the joint distribution is closely related to hybrid models and hence we compare our performance against the most recent hybrid models.

We determine whether a given test image is OOD based on a rejection rule. If this condition is satisfied then following [98] we set the probability of each label to be equal. We then quantify the performance of the rejection rule by reporting the average entropy of the labels for all test samples from the OOD set and the false positive rate (FPR = # of in-distribution samples rejected as OOD/ # of in-distribution samples). Thereafter, for all samples that are correctly identified as in-distribution, we report the accuracy of predicting the label, which is determined from $\boldsymbol{y}^{\text{mean}}$. We consider two rejection rules: $\|\hat{\boldsymbol{x}} - \boldsymbol{x}^{\text{map}}\| > c_1$, and $\|\hat{\boldsymbol{x}} - \boldsymbol{x}^{\text{map}}\| + \|\text{var}(\boldsymbol{y})\| > c_2$.

The performance of $\|\hat{\boldsymbol{x}} - \boldsymbol{x}^{\text{map}}\| > c_1$ rejection rule is shown in Figure 3.1, where we observe that it perfectly segregates the in-distribution and OOD samples. This is also apparent in Table 3.1, where it yields zero FPR and maximum entropy. Its accuracy for the in-distribution samples is also quite high. This accuracy can be further improved by using the combined rejection rule $\|\hat{\boldsymbol{x}} - \boldsymbol{x}^{\text{map}}\| + \|\text{var}(\boldsymbol{y})\| > c_2$, since it rejects some incorrectly labeled in-distribution samples with high variance as OOD. However, this comes at the cost of a slightly higher FPR. The usefulness of $\|\text{var}(\boldsymbol{y})\|$ as a predictor of accuracy is evident in Figure 3.2, where we observe that most correctly labeled samples have low variance (avg.

[1] available at: http://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html

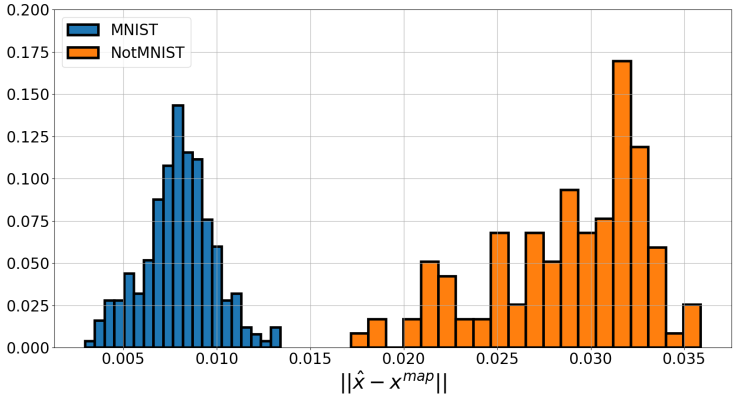


Figure 3.1: Histogram of $\|\hat{\boldsymbol{x}} - \boldsymbol{x}^{\text{map}}\|$, a measure for out-of-distribution (OOD) data detection, on classification experiments on MNIST. The proposed method is able to successfully distinguish in-distribution (MNIST) and OOD (NotMNIST) test inputs. A large value of this parameter is a warning to the end user to disregard the classification results.

value = 0.0026 ± 6e-3) when compared with their incorrectly labeled counterparts (avg. value = 0.025 ± 5e-3).

In Table 3.1 we compare the performance of our GAN-based approach with other approaches which perform uncertainty quantification by modeling the joint distribution using on flow-based models [98, 99]. The explicit nature of these models allows the joint density to be decomposed into generative and discriminative components, enabling a way to explore the generative-discriminative trade-off by introducing a weighted likelihood objective with a scaling parameter λ. Values of $\lambda < 1$ favor discriminative performance, while $\lambda > 1$ favors generative performance. In this context our approach may be regarded as one where the generative and discriminative components are equally weighted, and is therefore close to the choice $\lambda = 1$. Given this, in Table 3.1, we have compared our approach with hybrid models where $\lambda \approx 1$. We note that with the $\|\hat{\boldsymbol{x}} - \boldsymbol{x}^{\text{map}}\| \leq c_1$ rejection rule our model performs competitively with both the scaled [98] and the un-scaled hybrid models [99] for both in-distribution and OOD datasets. With the $\|\hat{\boldsymbol{x}} - \boldsymbol{x}^{\text{map}}\| + \|\text{var}(\boldsymbol{y})\| > c_2$ rejection rule it outperforms both hybrid models giving maximum accuracy and entropy but with non-zero FPR.

Table 3.1: Comparison of different hybrid models. Arrows indicate which direction is better. Also a "-" indicates that the values were not reported in the orginal reference.

	Configuration /	**MNIST**		**NotMNIST**
	Rejection rule	Acc ↑	FPR ↓	Entropy ↑
DIGLM [98] ($\lambda = 10.0/\mathrm{D}$)	$\log p(x)$	95.99 %	-	**2.300**
Residual Flow[99] ($\lambda = 1$)	Coupling	95.42%	-	-
	$+ 1 \times 1$ Conv	94.22%	-	-
	Residual	98.69%	-	-
Ours	$\|\hat{\boldsymbol{x}} - \boldsymbol{x}^{\mathrm{map}}\|$	96.81%	0	**2.300**
	$+ \parallel \mathrm{var}(\boldsymbol{y}) \parallel$	**99.57%**	0.064	**2.300**

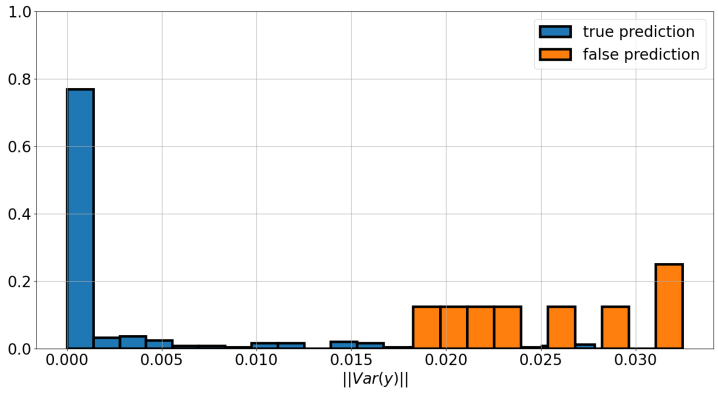


Figure 3.2: Histogram of $\|\mathrm{var}(\boldsymbol{y})\|$ for MNIST dataset.

Results in the table 3.1, figure 3.2, and 3.1 were obtained for likelihood variance (η) of 1. This value of likelihood variance is the optimal one and it was selected by performing classification for MNIST test set images for different values of η and selecting the one with the highest accuracy. Figure 3.3 shows the variation of MNIST test set accuracy with different likelihood variances.

We note that all the results described in this section are produced using a GAN with the latent space dimension of 100. However, in order to visualize how the prior latent distribution changes to posterior, we trained a separate GAN with the latent space dimension of 10.

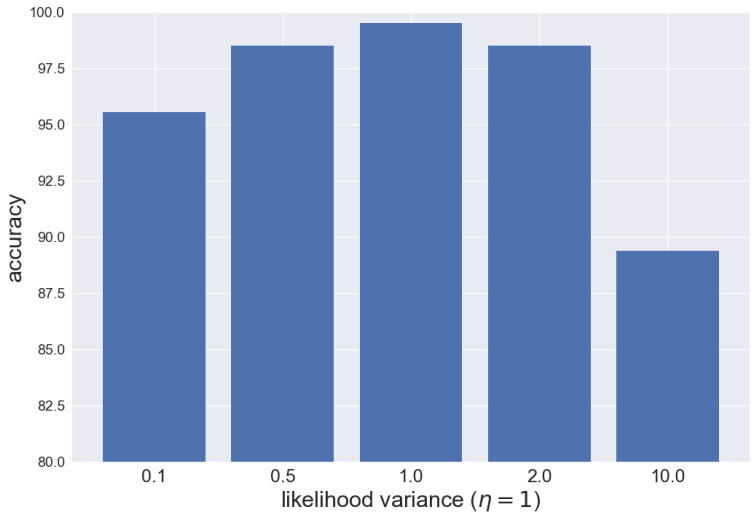


Figure 3.3: Variation of MNIST test set accuracy with likelihood variance.

Figure 3.4 shows the corner plot for the prior and the posterior distributions for such a GAN for an MNIST digit classification task. Figure 3.4 shows how prior distribution shifts and concentrates to a specific region of posterior due to likelihood term.

In order to evaluate the effect of latent space dimension on learning the prior distribution itself we trained four different GAN models on MNIST dataset with different latent space dimensions (5, 10, 50, 100). Figure 3.5 shows the variation of Wasserstein distance with latent space dimension. As can be observed from the figure as the latent space dimension increases the Wasserstein distance decreases indicating that GAN is able to approximate prior distribution more accurately. We note that increased la

3.3.2 Image inpainting

In this class of problems the quantity to be inferred, $\boldsymbol{y}$, is the occluded region of an image, and the measurement, $\hat{\boldsymbol{x}}$, is the noisy version of its visible portion. The goal is to recover the entire image, $\boldsymbol{u} = [\boldsymbol{x}, \boldsymbol{y}]$. While there has been great interest in recent years in developing efficient deep learning-based image inpainting algorithms [122–124], most of it has focused

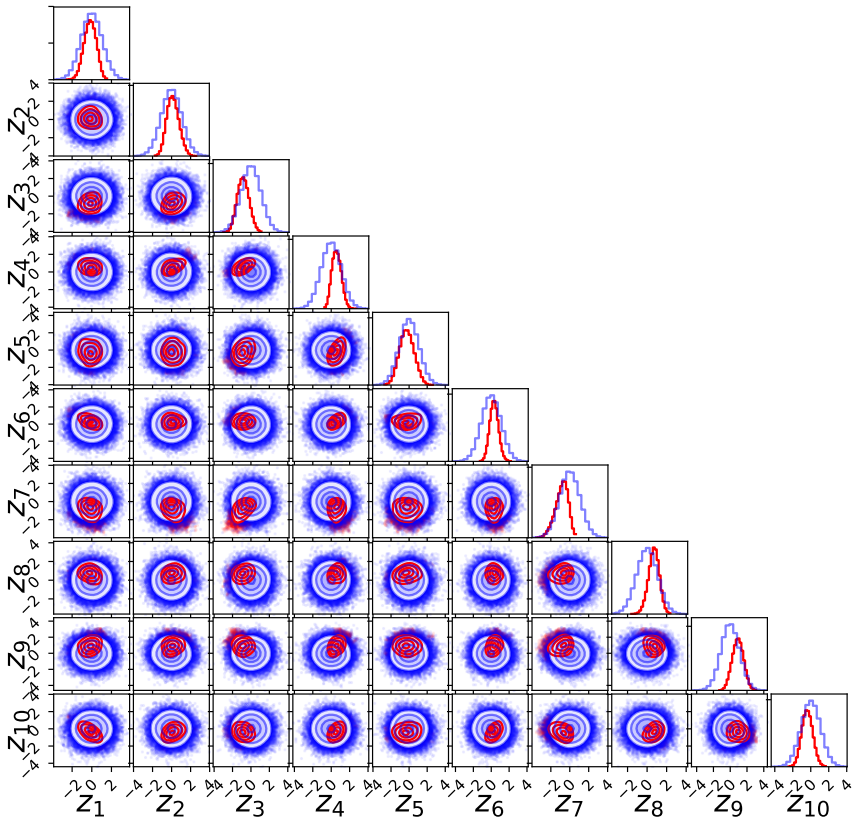
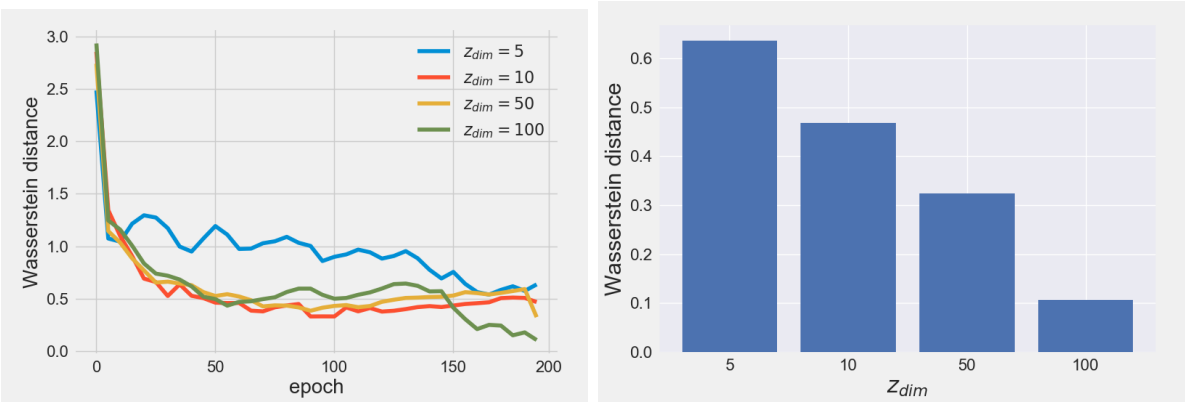


Figure 3.4: Corner plot of prior (shown in blue contours) and posterior (shown in red contour) distributions for a GAN with the latent space dimension of 10. Posterior distribution corresponds to an MNIST digit classification task.

on deterministic algorithms that lack the ability to quantify uncertainty in a prediction. In contrast, we use the algorithm described in Section 3.2.2 to perform probabilistic image inpainting.

We consider MNIST dataset and use 55k images to train a WGAN-GP as the prior. We generate measurements by selecting an image from the test set, occluding a significant region and then adding Gaussian noise. With this as input we use our algorithm to generate samples from the posterior distribution of the entire image (both occluded and retained regions). From these samples we evaluate the relevant QoIs, $\boldsymbol{u}^{\text{map}}$, $\boldsymbol{u}^{\text{mean}}$, and $\text{var}(\boldsymbol{u})$. These results are shown in Figure 3.6. They highlight that the map and mean images are close to the true image, even in the presence of significant occlusion and noise. The image of



(a) (b)

Figure 3.5: (a) Evolution of Wasserstein distance (between the true data distribution and the distribution defined by the samples generated by GAN) during training for different latent space dimensions (b) Effect of latent space dimension on prior approximation capability - Wasserstein distance (between the true data distribution and the distribution defined by the samples generated by GAN) at the end of training for different latent space dimension.

pixel-wise variance reveals that we are most uncertain along the boundaries of the digits and around the occlusion window.

In Figure 3.7, we demonstrate how uncertainty may be used in active learning/design of experiment, where the goal is to determine the optimal location for a measurement. We begin with an input where the entire image is occluded and in every subsequent step, we allow for a small 7×7 pixel window to be revealed. We select the window with the largest average pixel-wise variance. As the iterations progress, the MAP estimate converges to the true digit, and the variance decreases. In about 4 iterations we arrive at a very good guess for the digit. The performance of this approach is quantified in Figure 3.8, where we have plotted reconstruction error versus the number windows for this strategy, and a strategy where the subsequent window is selected randomly. The variance-driven strategy consistently performs better. We are not aware of any other methods for computing uncertainty in recovered images that have been applied to drive an active learning task in image inpainting. While methods based on dropout [125] or variational inference [126] could be extended to accomplish this, this has not been done thus far.

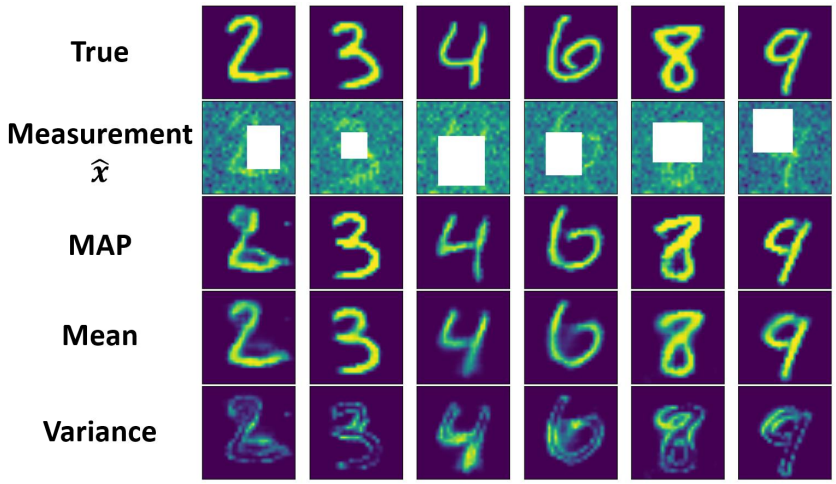


Figure 3.6: Estimate of the MAP, mean and pixel-wise variance from noisy occluded images using the proposed method. Note that the variance is peaked at the occluded region.

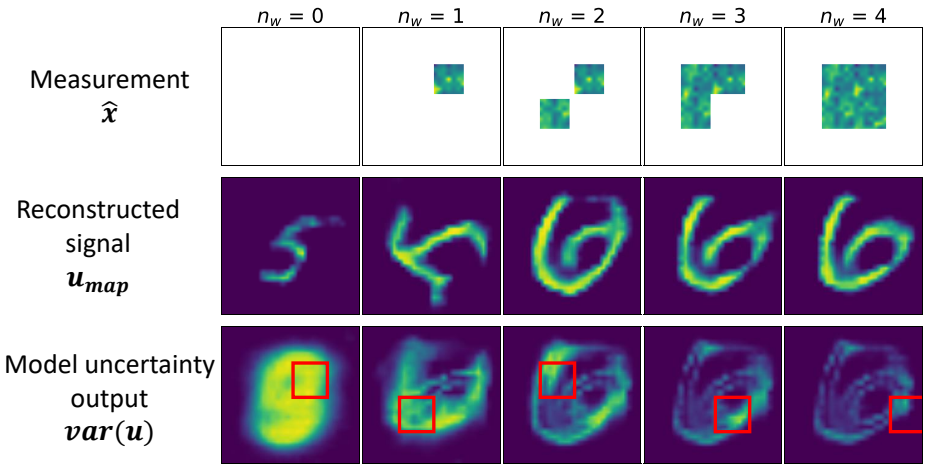


Figure 3.7: Estimate of the MAP (2nd row) and pixel-wise variance (3rd row) from the limited view of a noisy image (1st row) using the proposed method for image inpainting with a prior trained on MNIST images. An active learning strategy where subsequent measurement windows are selected at each iteration based on maximum value of variance (indicated by red rectangle). An accurate reconstruction of the original image is obtained with just 4 measurement windows.

Next, we present results of the variance-based window selection strategy applied the more challenging CelebA dataset [127] in Figure 3.9. We use 200k images for training the WGAN-GP model as prior and use images from the test set for inference. We note that

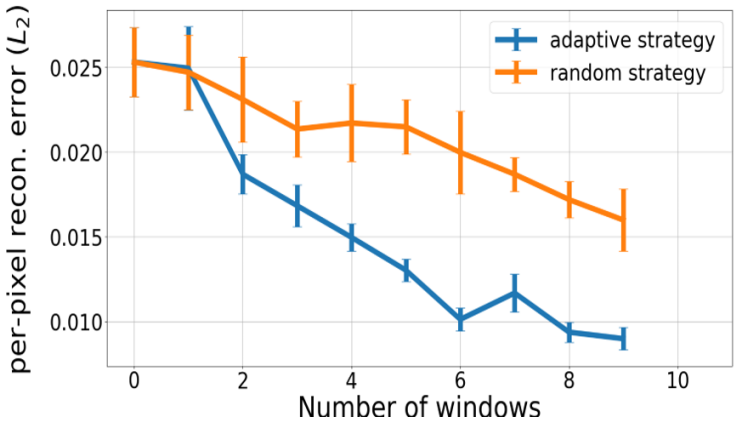


Figure 3.8: Average reconstruction error (with 95% confidence interval) as a function of number of windows for variance-driven (adaptive) and random sampling strategies.

the dimension of $\boldsymbol{u}$ is 12,288 whereas the dimension of $\boldsymbol{z}$ is 100, giving a parameter model order reduction by 123. The detailed architecture and hyper-parameters are described in Appendix 3.6. As we observe from Figure 3.9, the algorithm produces realistic images at each iteration; however, the initial variance is large indicating large uncertainty. As more windows are sampled using the active learning strategy, the variance reduces and by the 7th iteration a good approximation of the true image is obtained, even though only a small, noisy portion is revealed. Additional results for the MNIST and CelebA datasets are presented in Appendix 3.5.

3.3.3 Inference problems in computational physics

The quantification of uncertainty in computer simulations of physical systems is an important ingredient in reliably integrating such simulators in the design, development, and decision-making processes. There are two intimately coupled components of uncertainty quantification in physical systems. The first, often referred to as forward uncertainty quantification (UQ), pertains to the propagation of uncertainty from model parameters to model output. The second, commonly referred to as inverse uncertainty quantification, is concerned with the estimation of model parametric uncertainty from observed data. Different sets of

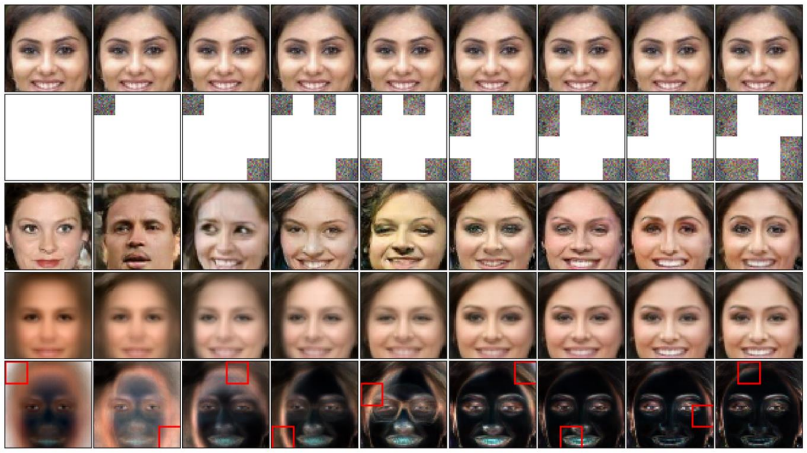


Figure 3.9: CelebA dataset: Estimate of the MAP (3rd row) , mean (4th row) and variance (5th row) from the limited view of a noisy image (2nd row) of a true image (1st row) using the variance-driven active learning strategy.

numerical methods are developed for each of these components. For example, polynomial chaos expansion [13], local sensitivity analysis [128], and moment methods [129] are typically used for forward UQ problems, whereas Bayesian approach is used for inverse UQ problems [130]. In this section we show how by virtue of learning the joint density $p_U(\boldsymbol{u})$ our proposed algorithm can perform both forward as well as inverse uncertainty quantification tasks. Furthermore, this algorithm requires no knowledge of the underlying physical model connecting input model parameters to model outputs and hence could be an attractive choice in situations where there is limited or no knowledge of the physical system.

To illustrate the main idea we consider the following model problem which represents the steady-state flow in random heterogeneous media.

$$\nabla \cdot \boldsymbol{\tau}(\boldsymbol{s}) = 0, \qquad \boldsymbol{s} \in \Omega \tag{3.15}$$

$$-\nabla p(\boldsymbol{s}) = \frac{\boldsymbol{\tau}(\boldsymbol{s})}{K(\boldsymbol{s})} + \frac{|\boldsymbol{\tau}(\boldsymbol{s})|\boldsymbol{\tau}(\boldsymbol{s})}{K^{1/2}(\boldsymbol{s})} + |\boldsymbol{\tau}(\boldsymbol{s})|^2\boldsymbol{\tau}(\boldsymbol{s}), \qquad \boldsymbol{s} \in \Omega \tag{3.16}$$

$$p(\boldsymbol{s}) = p_D(\boldsymbol{s}), \qquad \boldsymbol{s} \in \Gamma_D \tag{3.17}$$

$$\nabla p(\boldsymbol{s}) \cdot \boldsymbol{n} = h(\boldsymbol{s}), \qquad \boldsymbol{s} \in \Gamma_G \tag{3.18}$$

where $\boldsymbol{\tau}$ is flux vector field, K is permeability (hydraulic conductivity), p is pressure (hydraulic head), and $\boldsymbol{n}$ represents unit normal vector. Equation (3.15) is a statement of conservation of mass and Equation (3.16) represents a non-linear constitutive relationship that is valid for low Reynolds number [131, 132]. The domain of interest is a square, $\Omega = (0,1) \times (0,1)$ with Dirichlet data of $p_D = 1$ and $p_D = 0$ on the left and right boundaries, respectively, and Neumann data $h = 0$ on the top and bottom boundaries.

For our experiments, we do not solve or use the knowledge of these equations. Rather, we consider the open-source dataset from [132] which contains paired set of the nodal values of permeability and pressure field discretized on a uniform grid of 64×64 nodes. The permeability field is defined as a combination of multiple channels with binary values of 0.01 and 1. That is, $K(\boldsymbol{s}) = 0.01\mathbb{1}_{\Omega_A} + \mathbb{1}_{\Omega_B}$ where $\Omega_A \cap \Omega_B = \phi$ and $\Omega_A \cup \Omega_B = \Omega$. The pressure field p is obtained by solving equations (3.15-3.18) using the finite element method [133].

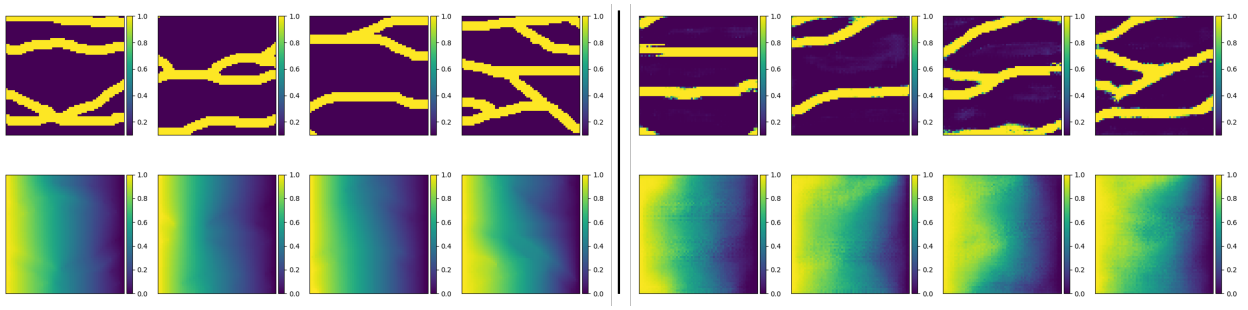


Figure 3.10: Realizations from the joint distribution of permeability (top row) and pressure field (bottom row). Left panel: samples from true joint density. Right panel: samples from the density learned by the WGAN model.

In order to solve the forward and inverse UQ problems arising in subsurface flow using the algorithm proposed in Section 3.2.2, we first train a WGAN-GP with a training set $\mathcal{S} = \{\boldsymbol{u}_1, \cdots, \boldsymbol{u}^{(S)}\}$ which contains 4,096 realizations of $\boldsymbol{u}$ drawn from the joint distribution, p_U, of the permeability and pressure fields. Details about the architecture and hyperparameters used are provided in Appendix 3.6. Figure 3.10 shows four such realizations from the true density and the learned density. Depending upon the nature of the inference problem (forward or inverse), the measured field, $\boldsymbol{x}$, and inferred field, $\boldsymbol{y}$, take on different meanings. In the forward problem, where we are interested in propagating uncertainty in the permeability field to the pressure field, $\boldsymbol{x}$ denotes the permeability and $\boldsymbol{y}$ denotes the pressure. While in the inverse problem, where we measure pressure and infer permeability, $\boldsymbol{x}$ denotes pressure and $\boldsymbol{y}$ denotes permeability. We note that the dimension of $\boldsymbol{u}$ is 8,192 whereas the dimension of latent space $\boldsymbol{z}$ is 100, thus allowing for efficient posterior sampling in a drastically reduced-dimensional space. Further, for all problems we assume that the noise in the measured vector is independent with a Gaussian density with zero mean and variance of 0.5.

3.3.3.1 Forward Problem

In this subsection, given the noisy permeability field, we determine the corresponding distribution of the pressure field.

We accomplish this by using the WGAN-GP trained using the sample set $\mathcal{S}$ as described above, and then for a given measurement of the permeability field from the test set, we use equation (3.14) in conjunction with WGAN prior to compute desired QoIs. Figure 3.11 shows results for two different permeability fields selected from the test set. It is observed that the MAP and mean estimates for the pressure closely match the true pressure field. Furthermore, the standard deviation, which is an estimate for the uncertainty in the recovered pressure, is correlated with the error in the MAP estimate. It is worth noting that no explicit information

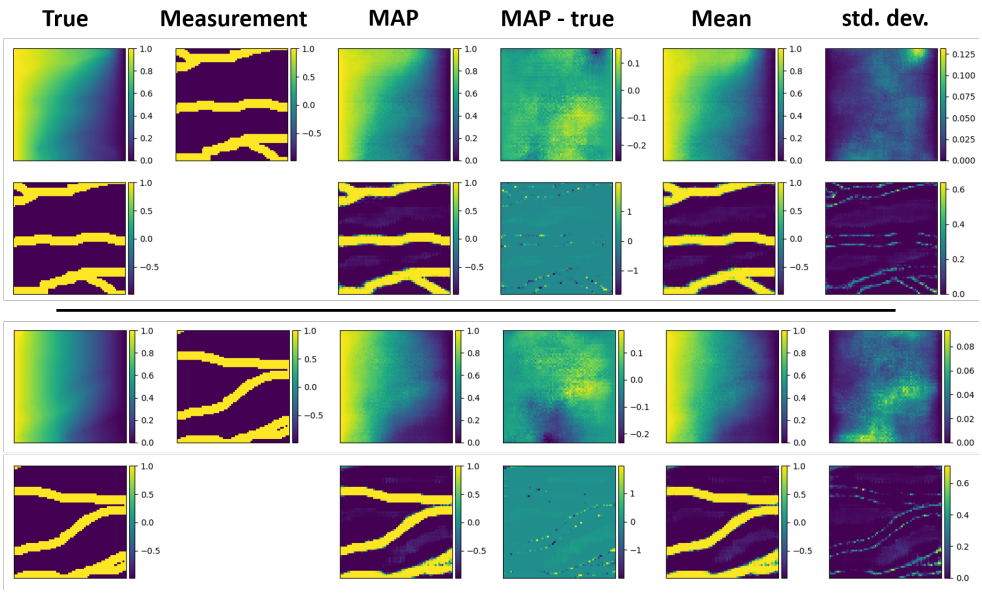


Figure 3.11: Forward propagation of uncertainty: Estimation of MAP (third column), mean (fifth column), and standard deviation (sixth column) of pressure field from the measurement of permeability field (second column). Two panels (top and bottom) show results for two different permeability measurements from the test set.

regarding the underlying PDE was provided to our algorithm. Instead, this information, that is the map from the permeability to the pressure, was learned directly from the pairwise data.

3.3.3.2 Inverse Problem

We solve the inverse problem of determining the distribution for the permeability field given a noisy measurement of the pressure field. We use the same WGAN generator to approximate the prior that was used in the forward problem. Next, given a noisy measurement of pressure field, we infer the posterior distribution of $\boldsymbol{u}$ using equation (3.13) and compute desired QoIs that include the MAP, mean, and standard deviation using equation (3.14). Figure 3.12 shows results the results for two different pressure measurements from the test set. We note that in contrast to the forward problem, the inverse problem is highly ill-posed, and this behavior is apparent in the inferred QoIs for the permeability. While the MAP estimate is able to reproduce the general structure of the network of channels of the true permeability, it misses some small branches. Further, the standard deviation is also large, indicating that

the relatively small uncertainty in the pressure field has resulted in significant uncertainty in the inferred permeability field. It is interesting to note that in regions where the MAP estimate is inaccurate, the standard deviation is also large. Thus warning the end user to not trust the inferred MAP estimate in these regions.

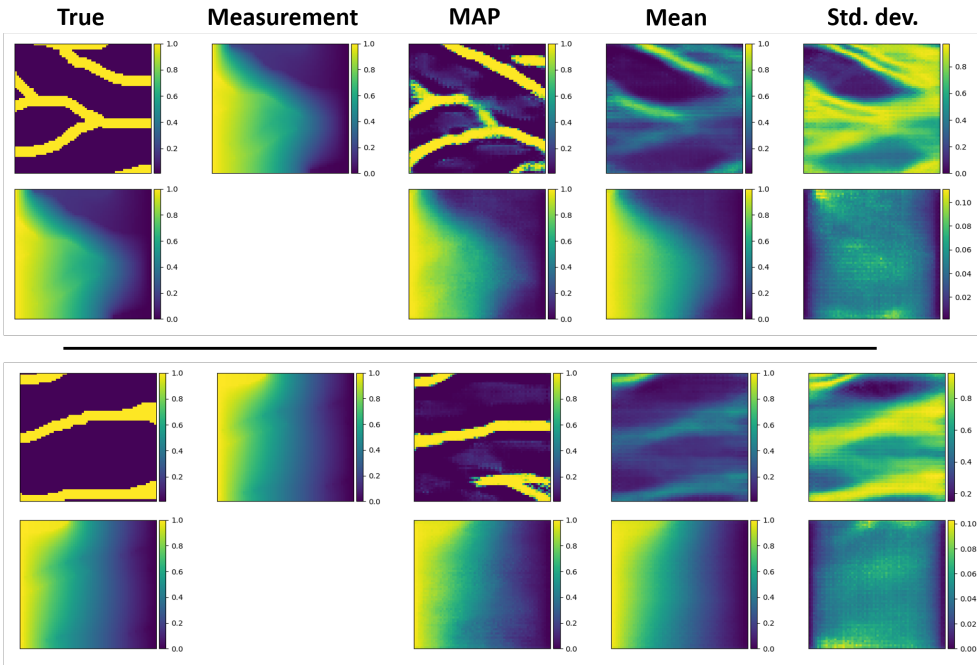


Figure 3.12: Inverse uncertainty quantification: Estimation of MAP (third column), mean (fourth column), and standard deviation (fifth column) of the permeability and pressure fields from the measurement of pressure field (second column). Two panels (top and bottom) show results for two different pressure measurements from the test set.

3.3.3.3 Mixed Problem

As described in the Introduction, one advantage of working with the joint distribution is that the designation of the measured and inferred fields can be changed easily without having to relearn the joint density p_U. This feature is highlighted in a "mixed problem" where measurements of pressure and permeability are made at randomly selected sparse locations (see Figure 3.13). Remarkably, our approach to solving this problem is very similar to that of solving the forward and inverse problems. The only change is in the definition of the

measured field, $\boldsymbol{x}$ and the inferred field $\boldsymbol{y}$. In the mixed problem, both $\boldsymbol{x}$ and $\boldsymbol{y}$ contain some components of the pressure and permeability fields. The top panel in Figure 3.13 shows the case where pressure and permeability measurements are made at only 1% of the total nodal locations of the domain (40 out of 4,096 nodes). Even with such sparse observations, the model is able to make reasonable posterior estimates for the MAP and the mean. Further, as before, regions where the MAP estimate is inaccurate are correlated with regions of relatively large values of standard deviation. As the number of measurements is increased (from top to bottom in the figure), the MAP and the mean estimates become more accurate and the standard deviation is reduced. This is clearly seen in Figure 3.14 where we have plotted the L_1 norm of standard deviation as a function of the percentage of nodes where observations are made. Both plots in Figure 3.14 were generated by considering 10 different samples from the test set. Additional results for this experiment are shown in Appendix 3.5.

3.4 Expression for the Maximum A-Posteriori (MAP) Estimate

The techniques described in Section 3.2.2 focus on sampling from the posterior distribution and computing approximations to QoIs. These techniques can be applied in conjunction with any distribution used to model noise and the latent space vector; that is, any choice of p_η (likelihood) and p_Z (prior). In this section we consider the special case when Gaussian models are used for noise and the latent vector. In this case, we can derive a simple optimization algorithm to determine the maximum a-posteriori estimate (MAP) for $p_Z^{\text{post}}(\boldsymbol{z}|\boldsymbol{x})$. This point is denoted by $\boldsymbol{z}^{\text{map}}$ in the latent vector space and represents the most likely value of the latent vector in the posterior distribution. It is likely that the operation of the generator on $\boldsymbol{z}^{\text{map}}$, that is $\boldsymbol{g}(\boldsymbol{z}^{\text{map}})$, will yield a value that is close to $\boldsymbol{y}^{\text{map}}$, and may be considered as an inexpensive approximation of the most likely solution ($\boldsymbol{y}^{\mathbf{map}}$) of the inference problem.

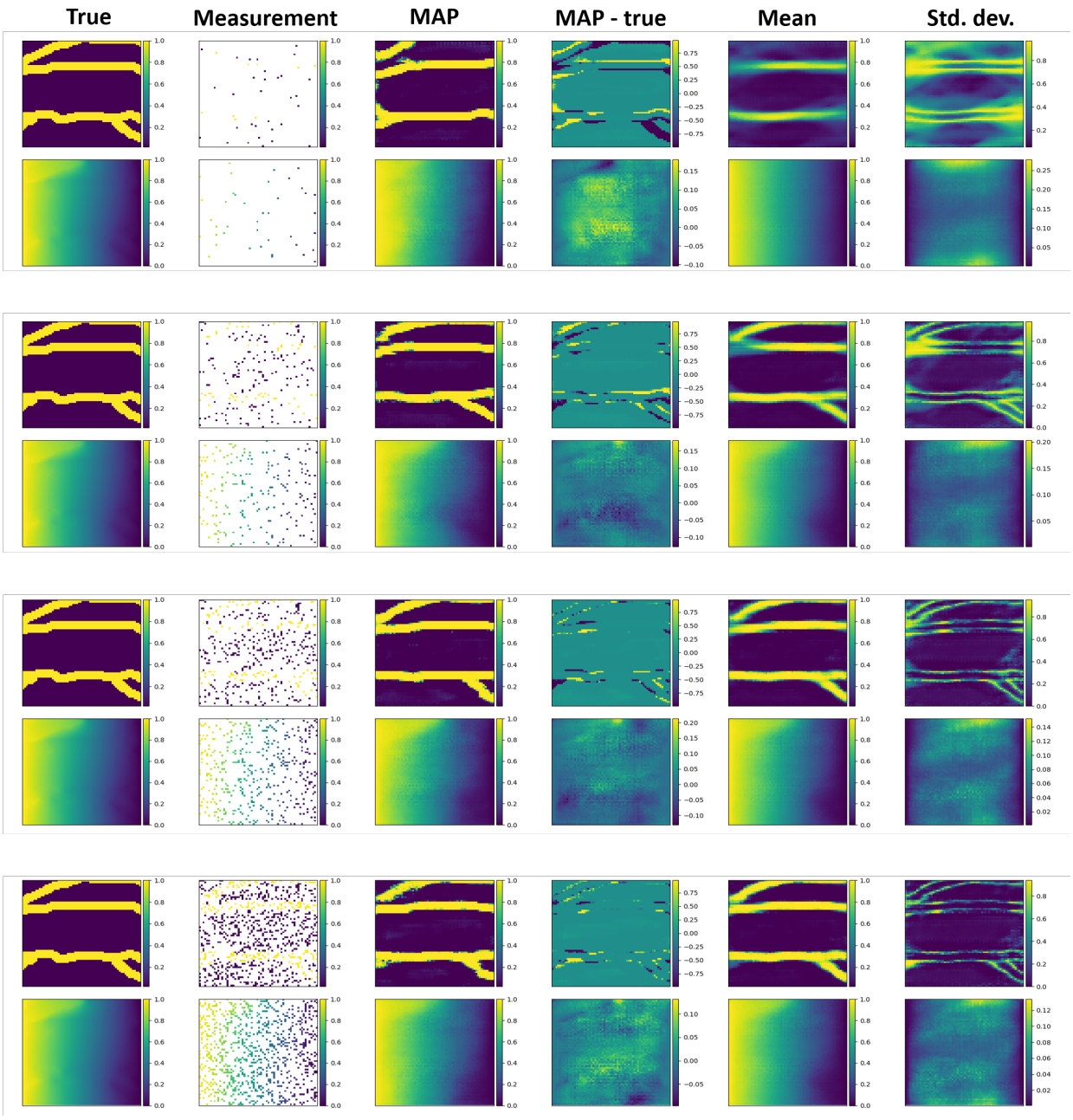


Figure 3.13: Inference with limited number of sparse measurements of pressure and permeability: From the top to bottom each panel corresponds to the case where measurements are made at 1%, 5%, 10%, and 20% of the total nodal locations. First column represents the true permeability and pressure fields. Second column shows the measured pressure and permeability fields. Third column shows the MAP estimate, and the fourth column shows difference between the MAP estimate and the ground truth. The last two columns represent the mean and the standard deviation.

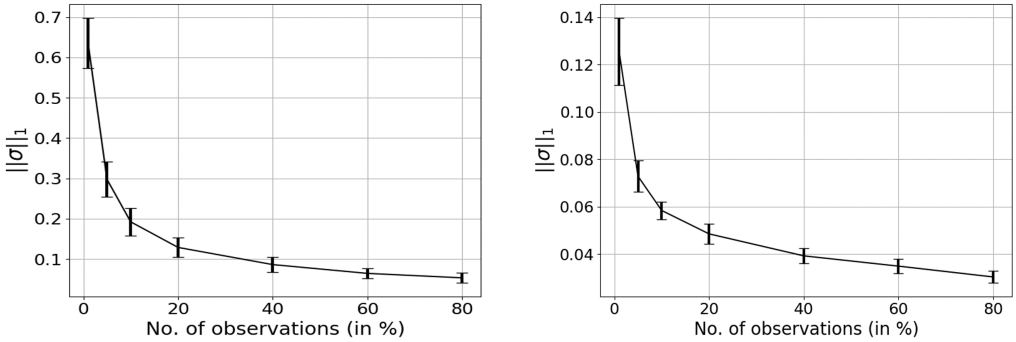


Figure 3.14: L_1 norm of the standard deviation for permeability (left) and pressure (right) fields (shown in the last column of figure 3.13) as a function of the percentage of nodal locations where measurements were made. These plots were generated by considering ten different samples from the test set. Error bars indicate one standard deviation variation across ten different samples at each measurement level.

We consider the case when the components of the latent vector are i. i. d. sampled from a normal distribution with zero mean and unit variance. This is often the case in many typical applications of GANs. Further, we assume that the components of noise vector are defined by a normal distribution with zero mean and a covariance matrix $\boldsymbol{\Sigma}$. Using these assumptions in (2.18), we have

$$p_Z^{\text{post}}(\boldsymbol{z}|\boldsymbol{x}) \propto \exp\Big(-\overbrace{\frac{1}{2}\big(|\boldsymbol{\Sigma}^{-1/2}(\hat{\boldsymbol{x}} - \mathbb{1}_{\boldsymbol{x}}(\boldsymbol{g}(\boldsymbol{z})))|^2 + |\boldsymbol{z}|^2\big)}^{\equiv r(\boldsymbol{z})}\Big). \tag{3.19}$$

The MAP estimate for this distribution is obtained by minimizing the negative of the argument of the exponential. That is

$$\boldsymbol{z}^{\text{map}} = \underset{\boldsymbol{z}}{\operatorname{argmin}}\, r(\boldsymbol{z}). \tag{3.20}$$

This minimization problem may be solved using any gradient-based optimization algorithm. The input to this algorithm is the gradient of the functional r with respect to $\boldsymbol{z}$, which is given by

$$\frac{\partial r}{\partial \boldsymbol{z}} = \boldsymbol{H}^T \boldsymbol{\Sigma}^{-1} (\mathbb{1}_{\boldsymbol{x}}(\boldsymbol{g}(\boldsymbol{z})) - \hat{\boldsymbol{x}}) + \boldsymbol{z}, \tag{3.21}$$

where the matrix $\boldsymbol{H}$ is defined as

$$\boldsymbol{H} \equiv \mathbb{1}_{\boldsymbol{x}} \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{z}}. \tag{3.22}$$

Here $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{z}}$ is the derivative of the generator output with respect to the latent vector. In evaluating the gradient above we need to evaluate the operation of the matrix $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{z}}$ on a vector, and not the matrix themselves. The operation of $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{z}}$ on a vector can be determined using a back-propagation algorithm within the GAN.

Once $\boldsymbol{z}^{\text{map}}$ is determined, one may evaluate $\boldsymbol{g}(\boldsymbol{z}^{\text{map}})$ by using the GAN generator. This represents the value of the field we wish to infer at the most likely value value of latent vector. Note that this is not the same as the MAP estimate of $p_U^{\text{post}}(\boldsymbol{u}|\boldsymbol{x})$.

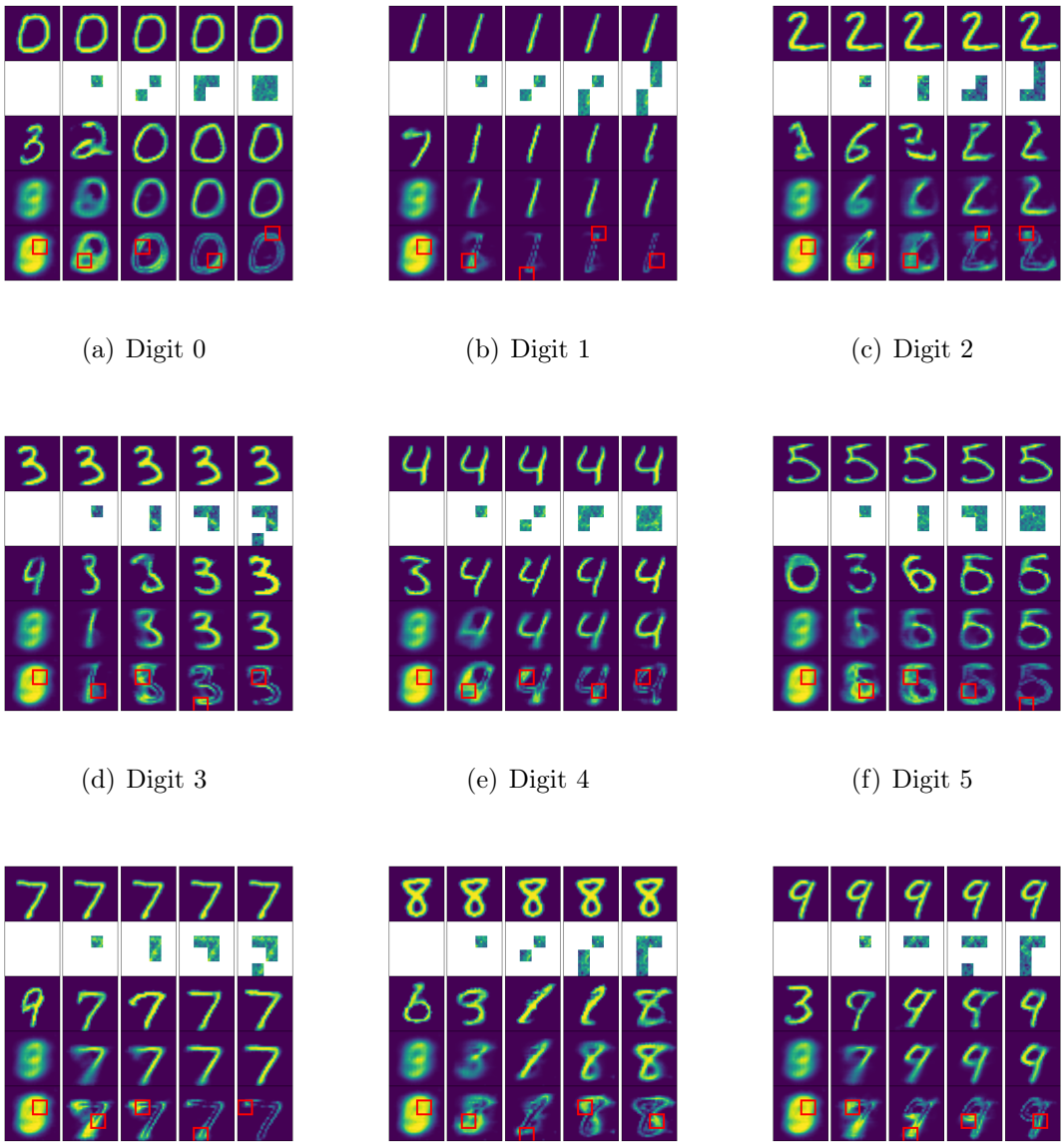
3.5 Additional Results

In this section we provide additional results for both MNIST and CelebA dataset for different tasks discussed in the main paper.

3.5.1 MNIST

First, in Figure 3.15, we provide additional examples for variance-based adaptive measurement window selection procedure described in Section 3.3.2.

Figure 3.16 shows additional results for the inpainting + denoising task, where an MNIST digit is occluded with masks of different sizes at different locations. Note that the variance is



(a) Digit 0 (b) Digit 1 (c) Digit 2

(d) Digit 3 (e) Digit 4 (f) Digit 5

(g) Digit 7 (h) Digit 8 (i) Digit 9

Figure 3.15: MNIST dataset: Estimate of the MAP (3rd row), mean (4th row) and variance (5th row) from the limited view of a noisy image (2nd row) using the proposed method. The window to be revealed at a given iteration (shown in red box) is selected using a variance-driven strategy. Top row indicates ground truth. For all digits measurement noise variance = 1.

high where the occlusion mask is located indicating lower confidence in reconstructed image in that location.

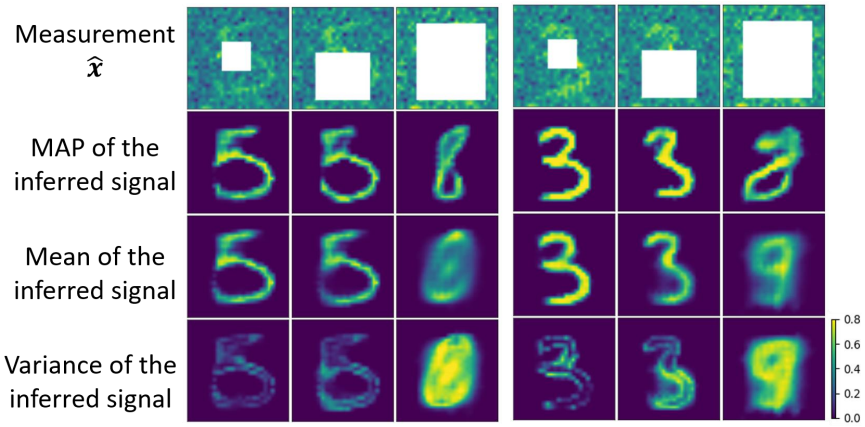


Figure 3.16: Estimate of the MAP (2nd row), mean (3rd row) and variance (4th row) from a noisy image (1st row) using the proposed method. Note that all variance images are plotted on the same color scale and it highlights increasing level of uncertainty as more and more portion of an image is occluded.

3.5.2 CelebA

In Figure 3.17, we provide additional examples for variance-based adaptive measurement window selection procedure described in Section 3.3.2 applied to the CelebA dataset.

3.5.3 Physics-driven inference

In Figure 3.18 , we provide additional examples for the mixed problem considered in Section 3.3.3.3.

3.6 Architecture and Training Details

We use the WGAN-GP model for learning the prior density. The tuned value of hyper-parameters is shown in Table 3.2.

The architecture of the GANs used to learn the prior density for the MNIST and CelebA data-sets is shown in Figures 3.19 and 3.20, respectively. Some notes regarding nomenclature used in these figures:

Table 3.2: Hyper-parameters for WGAN-GP model

Task	Image classification	Image inpainting and active learning		Physics-driven inference problems
Dataset	MNIST/ NotMNIST	MNIST	CelebA	Binary channels
Epochs	100	1000	500	500
Learning rate	0.0002	0.0002	0.0001	0.0001
Batch size	64	64	64	64
n_{critic}/n_{gen}	2	5	5	5
Momentum params. (β_1, β_2)	0.5, 0.999	0.5, 0.999	0.5, 0.999	0.5, 0.999

- Conv (H × W × C | s=n) indicates convolutional layer with filer size of HxW and number of filters=C with stride(s)=n.
- FC (x,y) indicates fully connected layer with x neurons in input layer and y neurons in output layer.
- BN = Batch norm, LN = Layer norm.
- TrConv = Transposed Convolution.
- LReLU = Leaky ReLU with α=0.2.

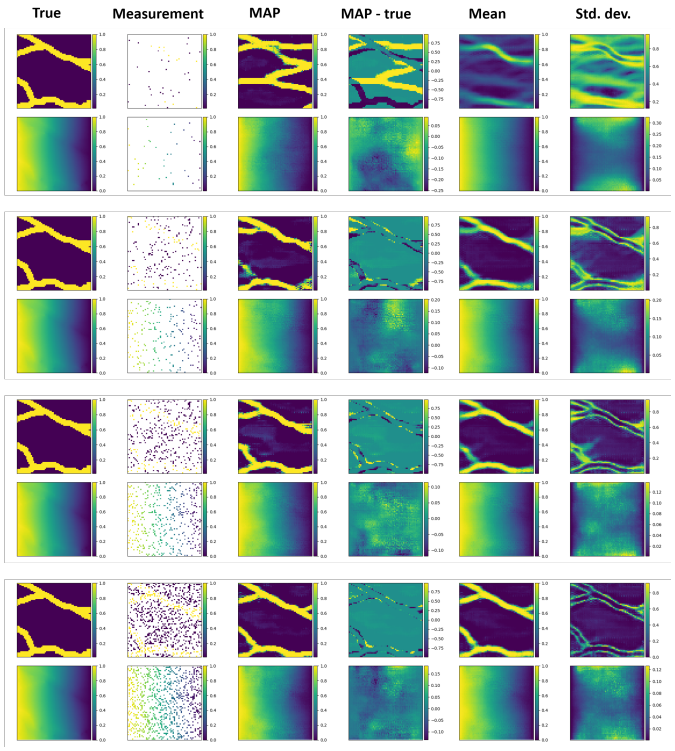
3.7 Conclusions

The ability to quantify the uncertainty in an inference problem is useful in identifying measurements that are outliers, in developing confidence in the inference and in designing strategies to improve the confidence. In this paper we have described how this may be accomplished when using deep neural networks to solve problems supervised learning. Our approach to quantifying uncertainty relies on formulating a Bayesian inference problem in which a WGAN is used to model the prior density, and solving the problem in the latent space of the WGAN. It derives its efficiency by mapping the posterior distribution to the latent space of the WGAN, whose dimension is often much smaller than that of the inferred

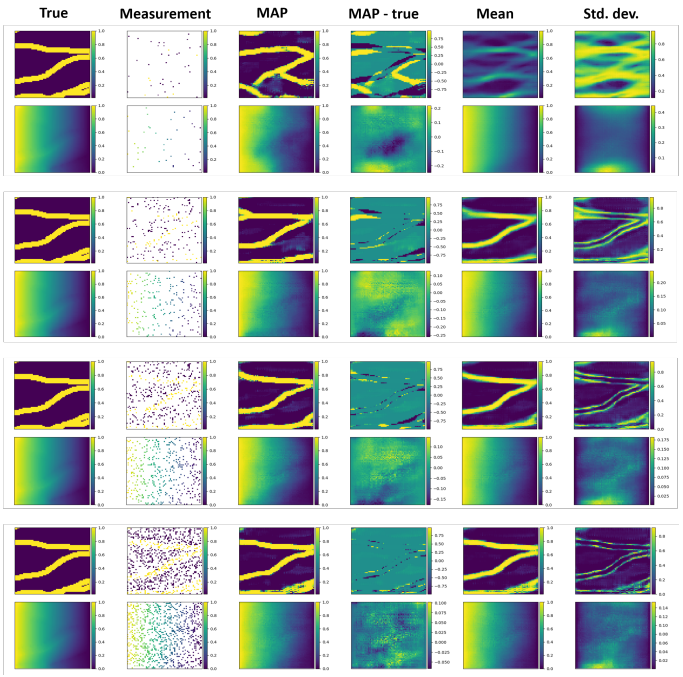
field. We have presented applications of this approach to image classification and image inpainting problems, and a problem motivated by a physical principle. Further, since GANs can learn complex distributions for a wide variety of fields from their samples, this approach can easily be applied to a large range of inference problems.

(a) (b) (c) (d) (e) (f)

Figure 3.17: CelebA dataset: Estimate of the y^{map} (3rd row), y^{mean} (4th row) and variance of y (5th row) from the limited view of a noisy image (2nd row) using the proposed adaptive method. The window to be revealed at a given iteration (shown in red box) is selected using a variance-driven strategy. Top row indicates ground truth. For all images measurement noise variance = 1.



(a)



(b)

Figure 3.18: Inference with limited number of sparse measurements of pressure and permeability: From the top to bottom each panel corresponds to the case where measurements are made at 1%, 5%, 10%, and 20% of the total nodal locations. First column represents the true permeability and pressure fields. Second column shows the measured pressure and permeability fields. Third column shows the MAP estimate, and the fourth column shows difference between the MAP estimate and the ground truth. The last two columns represent the mean and the standard deviation.

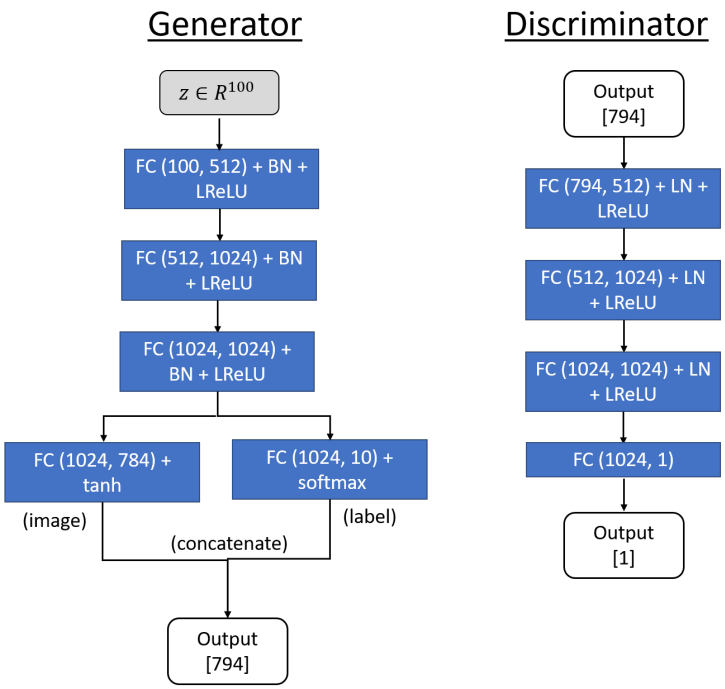
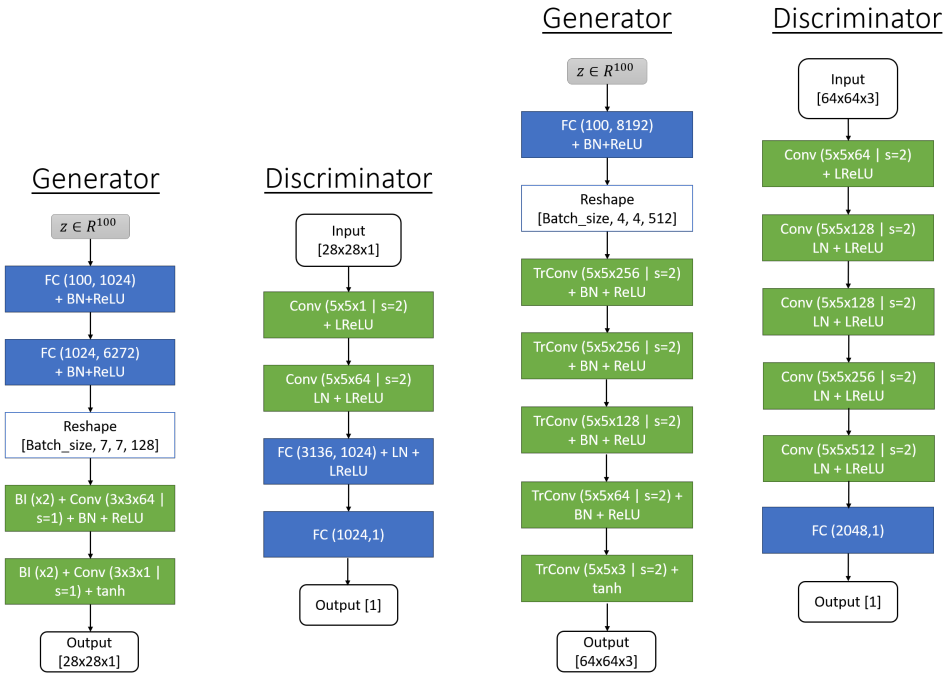


Figure 3.19: Generator and discriminator architecture used for image classification task for both MNIST and NotMNIST datasaet.



(a) Architecture for MNIST dataset (b) Architecture for CelebA dataset

Figure 3.20: Generator and discriminator architectures for (a) MNIST dataset and (b) CelebA dataset used in image inpainting and active learning tasks. Note that the same architecture as CelabA was used for the physics-driven inference problems described in section 3.3.3 except instead of three channels at the output of the generator and the input of the discriminator two channels were used (corresponding to permeability and pressure field)

Chapter 4

Inferring Visco-Elastic Properties from Interior Time Harmonic Data

> Plurality must never be posited
> without necessity.
>
> William of Ockham

4.1 Introduction

Inverse problem involving identification of material parameters (such as elastic modulus) are of ubiquitous in nature. For example, this type of inverse problem arises in bio-mechanical imaging, non-destructive testing, structural health monitoring, seismic exploration and so on. Due to its widespread applications and practical importance significant research effort has been put in recent years into developing efficient algorithms for material identification problem. In this chapter we focus on this deterministic inverse problem which involves inferring visco-elastic properties of tissue from time harmonic displacement data. Such type of inverse problem arises when we are interested in creating quantitative maps of mechanical properties of tissue and we can create it by deforming tissue in a time-harmonic fashion and measuring its response via advanced imaging modalities such as MRI. We develop general framework to solve such problem in this chapter. Further, we propose a novel domain decomposition

technique which can allow solving such problem in parallel significantly reducing total computational time. In the next section we begin with providing brief overview of Magnetic Resonance Elastography (MRE) and some of the open challenges in it. Then we provide mathematical formulation of inverse problem in MRE and in the final section we provide numerical results as well as explanation of proposed domain decomposition algorithm.

4.2 Magnetic Resonance Elastography

One of the important application of inverse problems involving identification of material parameters is Magnetic Resonance Elastography (MRE). MRE is a member of a family of techniques called elastography or elasticity imaging. Elastography is concerned with developing mechanical property map of tissue from tissue deformation data. It is well known that the progression of several diseases such as cancer, fibrosis and sclerosis, is marked by significant changes in tissue micro-structure, which eventually lead to changes in mechanical properties of tissues [134–137]. Elastography aims to map this mechanical properties of tissue *non-invasively*, which could be helpful in detection, diagnosis, and treatment monitoring of different diseases.

MRE is a dynamic elasticity imaging technique that uses mechanical waves to quantitatively assess the mechanical properties of tissue. MRE technique has three basic steps:

1. Low frequency shear waves are generated inside the tissue through external excitation.

2. Induced shear waves are imaged using Magnetic Resonance Imaging (MRI) depicting propogation of shear waves.

3. Solving a time-harmonic inverse problem using shear wave propagation image to generate qualitative map of tissue stiffness.

In this chapter we focus on the last step described above. Specifically, given a displacement image of tissue deformation inside the tissue obtained via processing of MRI, we are

interested in developing algorithm capable of producing mechanical property (shear modulus, for example) map of the tissue.

Many existing works studying this problem typically employ direct inversion method where measured data is directly used inside the model [138, 139]. While these direct methods are simpler to implement, computationally inexpensive, and less susceptible to the error in modeling assumptions, they are based on local homogeneity assumption. This significantly reduces their accuracy as one of the purpose of MRE (specially in the context of bio-mechanical imaging) is to find tissues with elevated stiffness. And so the boundary of such tissue inherently has heterogeneous shear modulus distribution, invalidating the local homogeneity assumption and hence the shear modulus estimates obtained via such methods are not too reliable. Iterative methods for MRE on the other hand do not face this issue as they do not make any homogeneity assumption. These methods however are quite susceptible to the error in modeling assumption. One such important assumption is the value of boundary condition, which is often difficult to characterize in MRE. In this chapter we leverage the Coupled Adjoint State Equation (CASE) formulation [54] (which allows for simultaneous solution of forward and adjoint problem (and hence calculation of gradient) without boundary condition data) to solve the MRE problem using iterative method. In next section we provide the problem formulation and then demonstrate the effectiveness of the proposed method by applying it to *in-sillico* data in the last section.

4.3 Problem Formulation

4.3.1 Strong formulation

We model the tissue as linear, incompressible solid material with 2D plane stress assumption. The strong formulation of the forward problem for MRE application (wave equation) for such material model is given as follow: given ...

1. the shear modulus $\mu(\boldsymbol{x})$ and density $\rho(\boldsymbol{x})$, in the entire spatial domain Ω,

2. the traction vector $\boldsymbol{h}(x)$ on a part of the boundary denoted by Γ_h and

3. the prescribed displacement vector $\boldsymbol{g}(x)$ on a part of boundary denoted by Γ_g

Find $\boldsymbol{u}(\boldsymbol{x}) \in \mathcal{S} \coloneqq \{\boldsymbol{u} \mid u_i \in H^1(\Omega), \boldsymbol{u} = \boldsymbol{g} \text{ on } \Gamma_g\}$ such that

$$\nabla \cdot \boldsymbol{\sigma} - \rho \ddot{\boldsymbol{u}} = \boldsymbol{0}, \qquad \text{in } \Omega \tag{4.1}$$

$$\boldsymbol{u} = \boldsymbol{g}, \qquad \text{on } \Gamma_g \tag{4.2}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{h}, \qquad \text{on } \Gamma_h \tag{4.3}$$

where,

$$\boldsymbol{\sigma} = \mu \boldsymbol{A}(\boldsymbol{u}) \tag{4.4}$$

$$\boldsymbol{A}(\boldsymbol{u}) = 2(\nabla^s \boldsymbol{u} + (\nabla \cdot \boldsymbol{u})\mathbb{1}) \tag{4.5}$$

In the above equations $\Omega \subset \mathbb{R}^2$ represents interior of the domain whose boundary is $\partial\Omega = \Gamma_g \cup \Gamma_h$, $\boldsymbol{n}$ is unit outward normal vector on $\partial\Omega$, $\nabla^s \boldsymbol{u} \coloneqq (\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)/2 = (\partial u_i/\partial x_j + \partial u_j/\partial x_i)/2$.

Let's take Fourier transform of Eq. (4.1)

$$\nabla \cdot (\mu \boldsymbol{A}(\hat{\boldsymbol{u}})) + \rho\omega^2 \hat{\boldsymbol{u}} = \boldsymbol{0} \tag{4.6}$$

where,

$$\hat{f}(\omega) = \mathcal{F}(f(t)) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \tag{4.7}$$

Eq.(4.6) represents frequency domain representation of Eq.(4.1) with natural frequency ω.

4.3.2 Weak formulation

Multiplying eq.(4.6) by weighting function $\hat{\boldsymbol{w}} \in \mathcal{V} \ := \{\hat{\boldsymbol{w}} \mid \hat{w}_i \in H^1(\Omega), \hat{\boldsymbol{w}} = \mathbf{0} \ on \ \Gamma_g\}$ and integrating by parts,

$$-\int_\Omega \nabla \hat{\boldsymbol{w}} \colon \mu \boldsymbol{A}(\hat{\boldsymbol{u}}) d\Omega + \rho\omega^2 \int_\Omega \hat{\boldsymbol{w}} \cdot \hat{\boldsymbol{u}} d\Omega + \cancelto{0}{\int_{\Gamma_h} \hat{\boldsymbol{w}} \cdot \mu \boldsymbol{A}(\hat{u}) \boldsymbol{n} d\Gamma_h} = \mathbf{0} \tag{4.8}$$

Now, let's define a Hermitian inner product as,

$$(a, b) = a^\star b \tag{4.9}$$

Then an equivalent weak form of Eq.(4.1-4.3) can be defined as: find $\hat{\boldsymbol{u}} \in \mathcal{S}$ such that

$$B(\hat{\boldsymbol{w}}, \hat{\boldsymbol{u}}; \mu) = 0 \qquad \forall \hat{\boldsymbol{w}} \in \mathcal{V} \tag{4.10}$$

where,

$$B(\hat{\boldsymbol{w}}, \hat{\boldsymbol{u}}; \mu) = \int_\Omega \nabla \hat{\boldsymbol{w}}^\star \colon \mu \boldsymbol{A}(\hat{\boldsymbol{u}}) d\Omega - \rho\omega^2 \int_\Omega \hat{\boldsymbol{w}}^\star \cdot \hat{\boldsymbol{u}} d\Omega \tag{4.11}$$

Expressing Eq. (4.10) as complex quantity,

$$-\rho\omega^2 \int_\Omega (\hat{\boldsymbol{w}}_{\boldsymbol{R}} - i\hat{\boldsymbol{w}}_{\boldsymbol{I}}) \cdot (\hat{\boldsymbol{u}}_{\boldsymbol{R}} + i\hat{\boldsymbol{u}}_{\boldsymbol{I}}) d\Omega + \tag{4.12}$$

$$\int_\Omega [(\nabla \hat{\boldsymbol{w}}_{\boldsymbol{R}} - i\nabla \hat{\boldsymbol{w}}_{\boldsymbol{I}}) : (\mu_R + i\mu_I)(\boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{R}}) + i\boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{I}})) d\Omega = 0 + i0 \tag{4.13}$$

and comparing real and imaginary parts

Real part :

$$-\rho\omega^2 \int_\Omega (\hat{\boldsymbol{w}}_{\boldsymbol{R}} \hat{\boldsymbol{u}}_{\boldsymbol{R}} + \hat{\boldsymbol{w}}_{\boldsymbol{I}} \hat{\boldsymbol{u}}_I) d\Omega +$$
$$\int_\Omega [\nabla \hat{\boldsymbol{w}}_{\boldsymbol{R}} : \mu_R \boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{R}})] + [\nabla \hat{\boldsymbol{w}}_{\boldsymbol{I}} : \mu_I \boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{R}})] + [\nabla \hat{\boldsymbol{w}}_{\boldsymbol{I}} : \mu_R \boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{I}})] - [\nabla \hat{\boldsymbol{w}}_{\boldsymbol{R}} : \mu_I \boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{I}})] = 0 \tag{4.14}$$

Imaginary part:

$$
\begin{aligned}
&-\rho\omega^2 \int_\Omega (\hat{\boldsymbol{w}}_{\boldsymbol{R}}\hat{\boldsymbol{u}}_{\boldsymbol{I}} - \hat{\boldsymbol{w}}_{\boldsymbol{I}}\hat{\boldsymbol{u}}_{\boldsymbol{R}}) d\Omega + \\
&\int_\Omega [\nabla \hat{\boldsymbol{w}}_{\boldsymbol{R}} \colon \mu_I \boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{R}})] - [\nabla \hat{\boldsymbol{w}}_{\boldsymbol{I}} \colon \mu_R \boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{R}})] + [\nabla \hat{\boldsymbol{w}}_{\boldsymbol{R}} \colon \mu_R \boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{I}})] + [\nabla \hat{\boldsymbol{w}}_{\boldsymbol{I}} \colon \mu_I \boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{I}})] = 0
\end{aligned}
\tag{4.15}
$$

4.4 CASE formulation

As discussed in Section 4.1, in this chapter we tackle the problem of using iterative algorithms for inverse problems arising in MRE with incomplete boundary information by using CASE formulation [54]. In this formulation a strong natural boundary condition is enforced by redefining the test function space.

$$
\mathcal{V}_{CASE} \coloneqq \{\mathbf{w} \mid \mathbf{w} = \mathbf{w_R} + i\mathbf{w_I}; w_{j_R} \in \mathbf{H^1}(\boldsymbol{\Omega}), w_{j_I} \in \mathbf{H^1}(\boldsymbol{\Omega}), w_{j_R} = w_{j_I} = 0 \text{ on } \Gamma\} \tag{4.16}
$$

$$
\mathcal{S}_{CASE} \coloneqq \{\mathbf{u} \mid \mathbf{u} = \mathbf{u_R} + i\mathbf{u_I}; u_{j_R} \in \mathbf{H^1}(\boldsymbol{\Omega}), u_{j_I} \in \mathbf{H^1}(\boldsymbol{\Omega})\} \tag{4.17}
$$

4.4.1 Forward problem

The weak form of forward problem given in Eq. (4.10) reduces to the following form after we incorporate the $\mathcal{V}_{CASE}$

$$
B(\hat{\boldsymbol{w}}, \hat{\boldsymbol{u}}; \mu) = \mathbf{0} \qquad \hat{\boldsymbol{w}} \in \mathcal{V}_{CASE} \tag{4.18}
$$

whose real and imaginary parts are given in Eq. (4.14) and (4.15) respectively.

4.4.2 Inverse problem

Now, we focus on the central part of this chapter concerning solving the inverse problem arising in MRE with potentially missing boundary information. To solve this inverse problem

using iterative methods, we need to compute gradient and for that we rely on adjoint based techniques. We follow the general guidelines [56] of computing gradient using adjoint method by first defining Lagrangian $\mathcal{L}$ and taking its directional derivative with respect to it to first compute primal (solution of forward problem) and dual (solution of adjoint problem) variables. Then we use this information to compute gradient.

Let's introduce the Lagrangian $\mathcal{L}$ that contains a plane stress elasticity equilibrium constraint and a weighted least square data mismatch term along with regularization term. The frequency domain measured displacement $\hat{\boldsymbol{u}}^{\boldsymbol{m}}$ and shear modulus μ are assumed to be known.

$$\mathcal{L}(\hat{\boldsymbol{w}}, \hat{\boldsymbol{u}}; \mu) = \frac{1}{2} \int_{\Omega} \boldsymbol{T}(\hat{\boldsymbol{u}} - \hat{\boldsymbol{u}}^{\boldsymbol{m}}) \cdot \boldsymbol{T}(\hat{\boldsymbol{u}} - \hat{\boldsymbol{u}}^{\boldsymbol{m}}) d\Omega + \frac{\alpha}{2} R(\mu) + Re[B(\hat{\mathbf{w}}, \hat{\mathbf{u}}; \mu)] \tag{4.19}$$

Here, tensor $\boldsymbol{T}$ is introduced for the fact that the measurement could be a linear function of displacement. In other words $\boldsymbol{T}$ could account for relative weightage of axial and lateral component of displacement to account for different resolution/sensitivity of imaging device in this two perpendicular directions. We note that we only considered here the real part of constraint B since both real and imaginary parts give rise to the same constraint equation.

Now, let us take the differential of $\mathcal{L}$

$$\delta\mathcal{L} = \mathcal{D}_{\hat{\boldsymbol{w}}}\mathcal{L} \cdot \hat{\boldsymbol{\delta w}} + \mathcal{D}_{\hat{\boldsymbol{u}}}\mathcal{L} \cdot \hat{\boldsymbol{\delta u}} + \mathcal{D}_{\mu}\mathcal{L} \cdot \delta\mu \tag{4.20}$$

Now, setting individual parts of above equation to zero. First consider the variation with respect to weight $\hat{\boldsymbol{w}}$

$$\mathcal{D}_{\hat{\boldsymbol{w}}}\mathcal{L} \cdot \hat{\boldsymbol{\delta w}} = Re[B(\hat{\boldsymbol{\delta w}}, \hat{\boldsymbol{u}}; \mu)] = 0 \tag{4.21}$$

which implies

$$-\rho\omega^2 \int_\Omega (\hat{\boldsymbol{\delta w}}_{\boldsymbol{R}} \hat{\boldsymbol{u}}_{\boldsymbol{R}} + \hat{\boldsymbol{\delta w}}_{\boldsymbol{I}} \hat{\boldsymbol{u}}_I) d\Omega +$$
$$\int_\Omega [\nabla \hat{\boldsymbol{\delta w}}_{\boldsymbol{R}} : \mu_R \boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{R}})] + [\nabla \hat{\boldsymbol{\delta w}}_{\boldsymbol{I}} : \mu_I \boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{R}})] + [\nabla \hat{\boldsymbol{\delta w}}_{\boldsymbol{I}} : \mu_R \boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{I}})] - [\nabla \hat{\boldsymbol{\delta w}}_{\boldsymbol{R}} : \mu_I \boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{I}})] = 0 \tag{4.22}$$

Next, consider the variation with respect to displacement $\hat{\boldsymbol{u}}$

$$\mathcal{D}_{\hat{\boldsymbol{u}}} \mathcal{L} \cdot \hat{\boldsymbol{\delta u}} = Re[(\hat{\boldsymbol{\delta u}}, \boldsymbol{D}(\hat{\boldsymbol{u}} - \hat{\boldsymbol{u}}^{\boldsymbol{m}}))] + Re[B(\hat{\boldsymbol{w}}, \hat{\boldsymbol{\delta u}}; \mu)] = 0 \tag{4.23}$$

where, $\boldsymbol{D} = \boldsymbol{T}^{\boldsymbol{T}} \boldsymbol{T}$

So,

$$\mathcal{D}_{\hat{\boldsymbol{u}}} \mathcal{L} \cdot \hat{\boldsymbol{\delta u}} = \int_\Omega [\hat{\boldsymbol{\delta u}}_{\boldsymbol{R}}^{\ T} \boldsymbol{D}(\hat{\boldsymbol{u}}_{\boldsymbol{R}} - \hat{\boldsymbol{u}}_{\boldsymbol{R}}^{\boldsymbol{m}})] + [\hat{\boldsymbol{\delta u}}_{\boldsymbol{I}}^{\ T} \boldsymbol{D}(\hat{\boldsymbol{u}}_{\boldsymbol{I}} - \hat{\boldsymbol{u}}_{\boldsymbol{I}}^{\boldsymbol{m}})] d\Omega + -\rho\omega^2 \int_\Omega (\hat{\boldsymbol{w}}_{\boldsymbol{R}} \hat{\boldsymbol{\delta u}}_R + \hat{\boldsymbol{w}}_{\boldsymbol{I}} \hat{\boldsymbol{\delta u}}_I) d\Omega$$
$$+ \int_\Omega [\nabla \hat{\boldsymbol{w}}_{\boldsymbol{R}} : \mu_R \boldsymbol{A}(\hat{\boldsymbol{\delta u}}_{\boldsymbol{R}})] + [\nabla \hat{\boldsymbol{w}}_{\boldsymbol{I}} : \mu_I \boldsymbol{A}(\hat{\boldsymbol{\delta u}}_{\boldsymbol{R}})] + [\nabla \hat{\boldsymbol{w}}_{\boldsymbol{I}} : \mu_R \boldsymbol{A}(\hat{\boldsymbol{\delta u}}_{\boldsymbol{I}})] - [\nabla \hat{\boldsymbol{w}}_{\boldsymbol{R}} : \mu_I \boldsymbol{A}(\hat{\boldsymbol{\delta u}}_{\boldsymbol{I}})] d\Omega = 0 \tag{4.24}$$

Similarly variation with respect to shear modulus μ when H^1 regularization is used (i.e. $\mathcal{R}(\mu) \coloneqq \int_\Omega \nabla \mu^T \nabla \mu d\Omega$) is given by,

$$\mathcal{D}_{\hat{\mu}} \mathcal{L} \cdot \hat{\delta\mu} = Re[B(\hat{\boldsymbol{w}}, \hat{\boldsymbol{u}}; \delta\mu)] + \alpha Re[(\nabla \delta\mu, \nabla \mu)] \tag{4.25}$$

which implies,

$$\mathcal{D}_{\hat{\mu}} \mathcal{L} \cdot \hat{\delta\mu} = \int_\Omega [\nabla \hat{\boldsymbol{w}}_{\boldsymbol{R}} : \delta\mu_R \boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{R}})] + [\nabla \hat{\boldsymbol{w}}_{\boldsymbol{I}} : \delta\mu_I \boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{R}})] + [\nabla \hat{\boldsymbol{w}}_{\boldsymbol{I}} : \delta\mu_R \boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{I}})] -$$
$$[\nabla \hat{\boldsymbol{w}}_{\boldsymbol{R}} : \delta\mu_I \boldsymbol{A}(\hat{\boldsymbol{u}}_{\boldsymbol{I}})] d\Omega + \alpha \int_\Omega [\nabla \hat{\delta\mu}_R \nabla \hat{\mu}_R + \nabla \hat{\delta\mu}_I \nabla \hat{\mu}_I] d\Omega \tag{4.26}$$

4.4.3 Iterative inversion

Now we have all the necessary ingredients to perform iterative update in gradient-based algorithms (like BFGS or conjugate gradient) to find optimal shear modulus distribution μ^*, such that it matches with the measured displacement under regularization term.

Iterative inversion begins with an initial guess of the material parameter distribution $\boldsymbol{\mu}$, observed/measured displacement field $\boldsymbol{u^m}$, and appropriate choice of regularization parameter α and tolerance parameter ϵ. Once we have this information we follow the following steps:

1. Solve Eq.(4.22) to obtain predicted displacement $\boldsymbol{u}$.
2. Compute the objective function; if it is smaller than a set tolerance ϵ, stop and take the current material properties ($\boldsymbol{\mu}^* \leftarrow \boldsymbol{\mu}$) to be optimal. If not go to next step.
3. Use the predicted displacement $\boldsymbol{u}$ and shear modulus distribution $\boldsymbol{\mu}$ in Eq. (4.24) to obtain adjoint/dual variable $\boldsymbol{w}$.
4. Use $\boldsymbol{u}$ and $\boldsymbol{w}$ in Eq. (4.26) to compute gradient of objective function with respect to $\boldsymbol{\mu}$.
5. Update the previous guess of $\boldsymbol{\mu}$.
6. Go back to step 2.

4.5 Numerical Results

We demonstrate the effectiveness of proposed method in solving inverse problem in MRE using simulated data. For this we consider the shear modulus distribution shown in the left panel of Figure 4.1. The shape of the inclusion in this shear modulus distribution and the modulus value was chosen such that it mimics a malignant tumor. The exact mathematical description of generating such elliptical inclusions in soft matrix is shown in Section 5.4 of

Chapter 5. Using the technique described in that section we had generated a library of 10,000 such shear modulus samples. We randomly selected one sample from that library as real component of the complex-valued shear modulus. Then we computed the imaginary component of shear modulus by multiplying its real counterpart by 0.7. The resulting real and imaginary components of shear modulus are shown in the left panel of Figure 4.1. This will act as our ground truth. The physical size of this modulus sample is $0.02 \times 0.02m^2$ and it was discretized in 51 equispaced nodes in each direction. Given this complex-values shear modulus distribution we solve the forward problem (Eq. (4.22) in frequency domain with value of ρ=1000 kg/m^3 and $\omega = 601.18$ Hz to obtain frequency domain displacement field (shown in the middle panel of Figure 4.1). We note that since in CASE formulation we do not need any boundary condition data we can solve this forward problem without making any additional assumption or considerations about domain like Perfectly Matched Layer (PML). We consider the output of this forward problems as our observed/measured displacement field and solve the inverse problem of recovering shear modulus distribution by following the steps described in subsection 4.4.3. The shear modulus reconstruction is shown in the right panel of Figure 4.1. As can be observed from the figure our algorithm does excellent job at reconstructing the complex-valued modulus distribution and it matches almost perfectly with the ground truth (Figure 4.1 left panel)

Next, to mimic the real life situation (and to avoid the *inverse crime* committed in the previous step), where typically observed displacement field is corrupted with noise, we add an additive Gaussian noise of 3% to the displacement field obtained by solving the forward problem (Figure 4.1 middle panel). The resulting noisy displacement fields are shown in Figure 4.2(middle panel). We note that with MRI we can get displacement measurements which are much less noisy than the ones obtained from ultrasound and the 3% noise we have added here is at a relatively higher end of the spectrum of noise observed in MRI. We then again followed the same steps as described in subsection 4.4.3 and solve the inverse

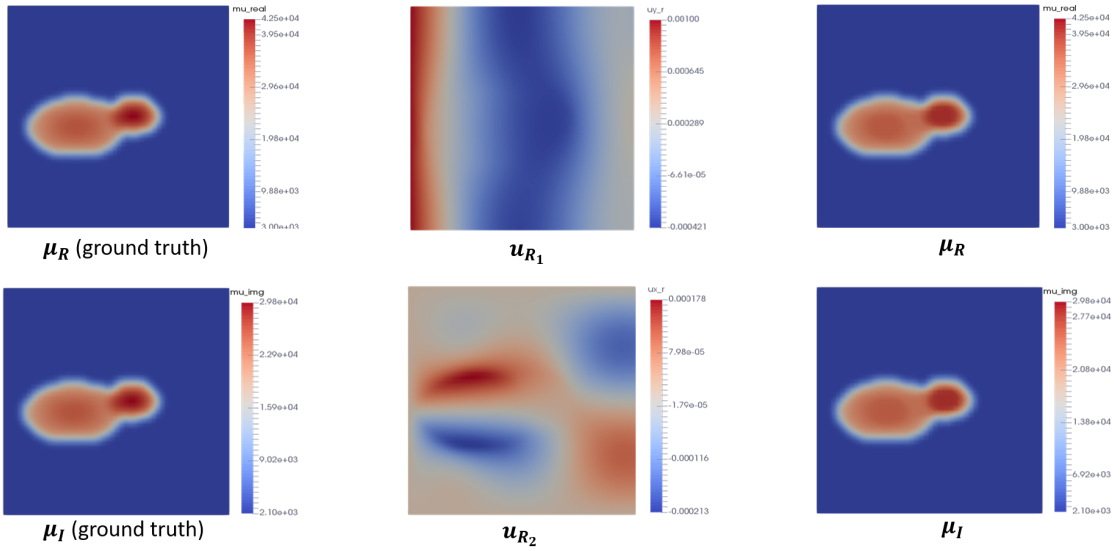


Figure 4.1: Reconstruction results with elliptical inclusion. *Left panel:* real (top row) and imaginary (bottom row) components of shear modulus distribution for ground truth. *Middle panel:* real and imaginary components of displacement field corresponding to ground truth shear modulus shown on the left. *Right panel:* Real (top row) and imaginary (bottom) components of reconstructed shear modulus.

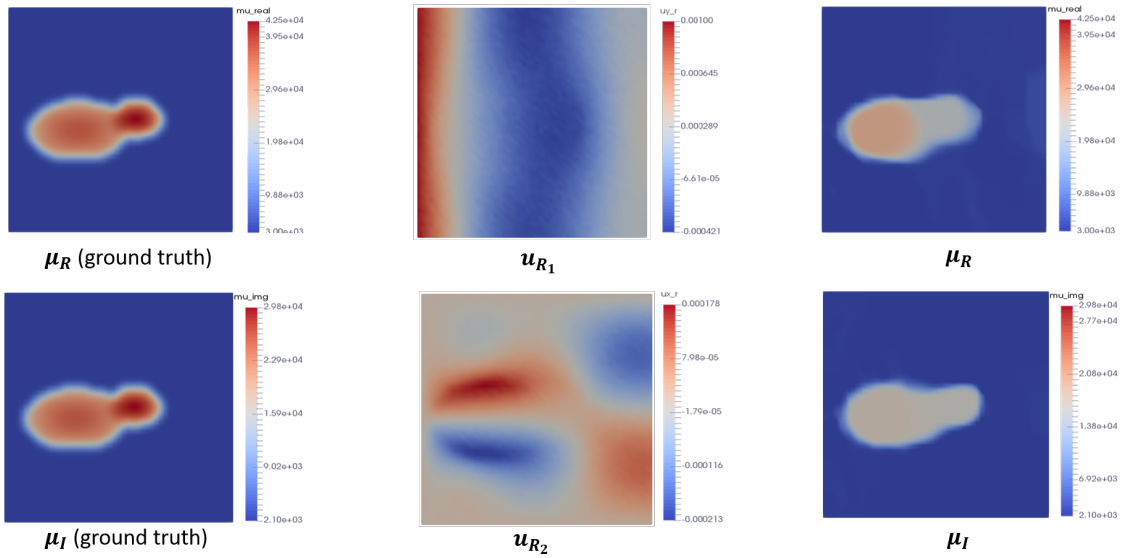


Figure 4.2: Reconstruction results from noisy measured displacement. *Left panel:* real (top row) and imaginary (bottom row) components of shear modulus distribution - ground truth. *Middle panel:* noisy measured/observed displacement field with 3% additive Gaussian noise. *Right panel:* Real (top row) and imaginary (bottom) components of reconstructed shear modulus.

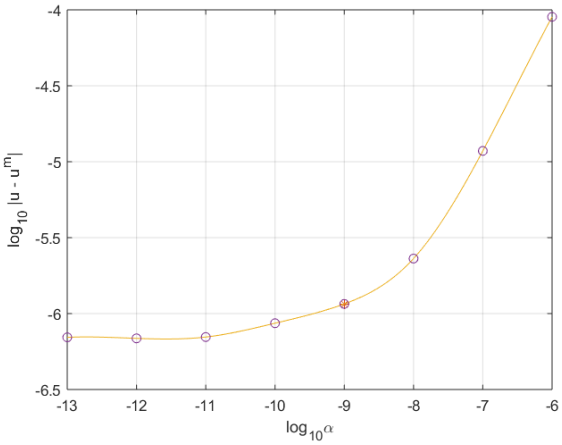


Figure 4.3: L-curve for reconstruction results with 3% additive Gaussian noise. Note that optimal regularization parameter value (10^{-9}) is indicated by asterisk.

problem by appropriately selecting regularization parameter using L-curve (shown in Figure 4.3). The final reconstruction result is shown in the right panel of Figure 4.2.

4.5.1 Domain Decomposition

In this section we propose a novel and extremely useful application of our algorithm. It is concerned with solving a large-scale inverse problems. Many practical science and engineering problems involve solving time-harmonic inverse problems on a very fine grid where the discrete representation of field we are interested in inverting could be very large. In such scenarios, solving such a large-scale inverse problem could be extremely slow and time consuming.

If we have sufficient computing resources then it is sensible to divide the domain on which the inverse problem is being solved into number of smaller sub-domains and solving individual inverse problems on each of these sub-domains using separate computing node. Such an approach is usually refered to as domain decomposition in the numerical analysis and scientific computing community. One key challenge, however, of employing such technique to time-harmonic inverse problems considered in this chapter is that it requires sharing common information across different compute nodes. For example, if we divide our domain

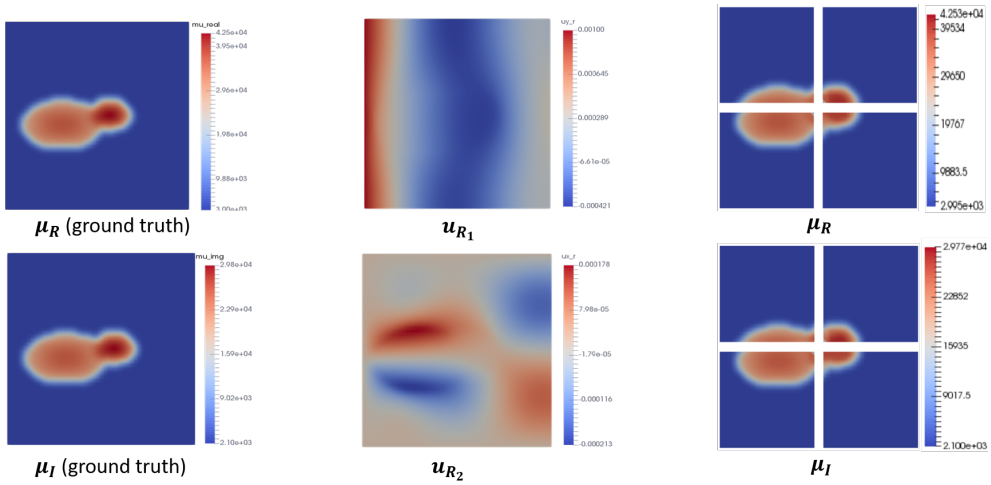


Figure 4.4: Domain Decomposition: Inverse problem in each sub-domain is solved by different processor in parallel.

into two equally sized sub-domains and hope to solve individual inverse problems in each of them by assigning it to different compute nodes, then it might not work as both these sub-domains share a common edge (which will act as boundary) for both sub-domains and any change in the value of parameters on any nodal point on this boundary during optimization process of inverse problem solution would make the overall solution inconsistent as the value of same parameter at same nodal point might be different in opposite sub-domain. Hence, it is desirable to develop domain decomposition technique which does not require sharing information across common boundary.

CASE formulation discussed in this chapter naturally provides this advantage as it does not require any boundary condition data. Figure 4.4 shows application of this novel domain decomposition technique to the inverse problem in MRE. Here we use the same measurement (as in noiseless case described above). However, rather than solving the whole inverse problem on a single machine, we divided the physical domain into four equally sized sub-domains and assign each of the sub-domain to different processors/computing nodes. Despite sharing no boundary information our algorithm is capable of performing excellent reconstruction even in the regions where common edge is shared between two sub-domains. Although in the example we divide teh physical domain into four sub-domains, in principle, it can very easily

be divided into many more sub-domains. Thus by allowing simultaneous solution of inverse problems in multiple sub-domains without compromising accuracy, our method opens up new avenue for studying/analyzing and solving large-scale time harmonic inverse problems.

Chapter 5

Circumventing the Solution of Inverse Problems in Mechanics through Deep Learning

> Simplicity is the ultimate sophistication.
>
> Leonardo da Vinci

5.1 Introduction

The ability to identify elastic heterogeneity and nonlinearity from the response of a specimen is important in areas like non-destructive testing and evaluation and, more recently, in medical imaging. In medical imaging, conventional imaging techniques like ultrasound, optical coherence tomography (OCT) and magnetic resonance imaging (MRI) can be used to measure displacement data within tissue as it is deformed [140–143]. This data is then used in conjunction with the laws of mechanics to infer the spatial distribution of elastic properties [144–147]. Once these properties have been determined, they are examined to identify features that are useful in making a diagnosis. For example, in the context of breast cancer, recent results have shown that the degree of heterogeneity in the shear or Young's

Portions of this chapter previously appeared as: D. Patel, R. Tibrewala, A. Vega, L. Dong, N. Hugenberg, and A. A. Oberai, "Circumventing the solution of inverse problems in mechanics through deep learning: Application to elasticity imaging," Comput. Methods Appl. Mech. Eng., May 2019.

modulus and the extent of nonlinear elastic response are both linked to a malignant diagnosis [148–150]. In particular, malignant lesions tend to have a spatially heterogeneous Young's modulus distribution and also tend to stiffen at a faster rate with increasing overall strain than their benign counterparts.

The process of determining whether a given specimen is elastically heterogeneous, or has significant elastic nonlinearity, includes the following steps:

1. Image the specimen using a conventional phase-sensitive imaging modality such as ultrasound, MRI or OCT, as it is being deformed.

2. Use image cross-correlation or a phase-tracking approach to determine displacements inside the tissue.

3. Use displacement data and the physical laws of mechanics to infer the spatial distribution of mechanical properties of the specimen by solving an inverse problem.

4. Quantify the value of the target feature within these distributions (heterogeneity/nonlinearity).

5. Use this feature to classify the specimen (malignant or benign, for example).

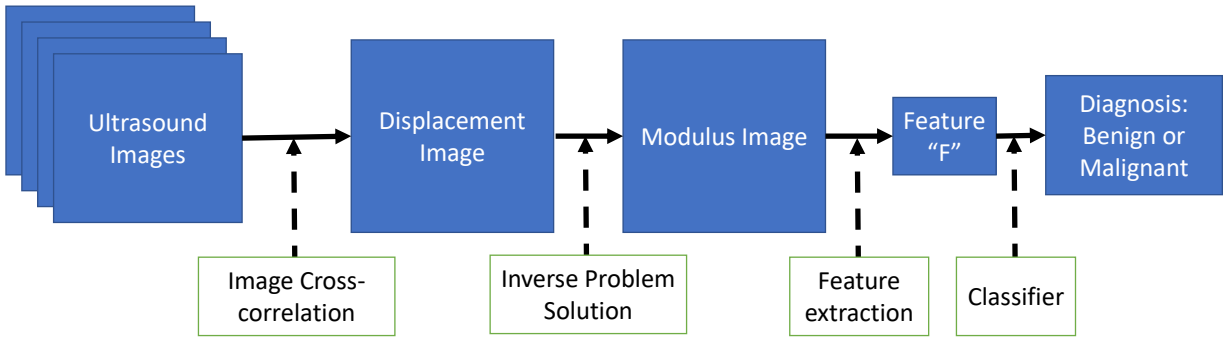


Figure 5.1: Standard workflow for diagnosis based on ultrasound elastography.

This workflow, which represents the current state-of-the-art, is shown in Figure 5.1. Among the steps in the workflow, the step of solving the inverse elasticity problem to determine the spatial distribution of mechanical properties (Step 3) and the step of identifying

and quantifying the appropriate features from these distributions (Step 4) are particularly challenging. Step 3 requires the solution of an ill-posed problem, which is a complex task. Further, once the inverse problem is solved and modulus distribution is obtained, identifying the useful features and relevant measures for quantifying these features for a given problem is also challenging.

Given that the mechanical properties obtained through the solution of inverse problem are interrogated for relevant features, and that these features are used to make a diagnosis, an intriguing question is whether it is necessary to solve the inverse problem at all. Or put another way, is it possible to move directly from measured displacements to the desired diagnosis? Considering that deep learning algorithms are good at extracting useful features from data and then using these for a given task, we ask whether it is possible to leverage the capabilities of these data-based models in the standard physics-based workflow.

Due to their ability to discover intricate structure in high dimensional data, along with improved computational resources and availability of large amount of data, deep learning algorithms have enabled significant advances in various domains that range from image recognition [20, 151], natural language processing [22], reinforcement learning [24] to high energy physics [25], computational chemistry [26] and medical imaging [28]. These algorithms can perform a variety of data analysis tasks that include solving classification problems, where the goal is to bin a collection of input data into two or more groups [152]. The input to these problems can be hierarchical, such as audio, image, text and video, or non-hierarchical, such as general tabular data. Convolutional neural networks (CNNs) are designed specifically to learn different levels of abstraction of hierarchical data in an efficient manner and are now widely used in a variety of computer vision tasks like object detection, image recognition and image segmentation [20, 153, 154]. These networks are comprised of several layers of spatially local convolution operators that operate on the image data to extract relevant features from the images. The convolution layers are typically followed by fully connected layers where

the extracted features are manipulated in a high dimensional feature space and are supplied to the final layer which performs the classification task.

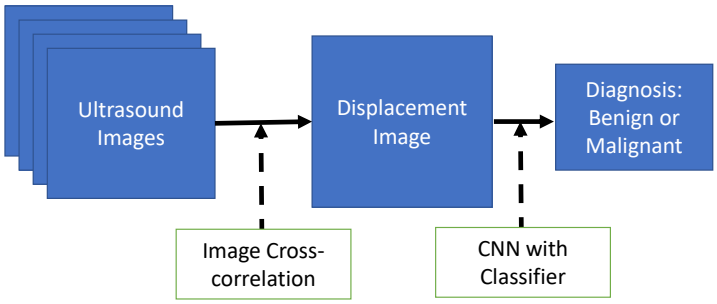


Figure 5.2: Deep-learning based workflow for classifying malignant lesions.

In this chapter, we consider the following question: can we train a CNN so that it can successfully identify mechanical features using displacement data as input? In particular, can it be used to classify specimens as being elastically heterogeneous or elastically nonlinear based on their displacement response? The utility of a network that can accomplish this task is clear when considering the workflow depicted in Figure 5.1. If successful, it would circumvent the steps of (a) solving a complex inverse problem to determine mechanical property distributions , (b) identifying and extracting the relevant features, and (c) classifying based on these features. Instead of the workflow shown in Figure 5.1, it would lead to a much simpler learning-based workflow depicted in Figure 5.2, where three three steps are combined into the single step of training the CNN.

In order to answer this question, we generate a large set of material property distributions that belong to "malignant" and "benign" classes. When compared to the benign class, the Young's modulus distribution in the malignant class is more heterogeneous and the nonlinear elastic parameter is elevated. Thereafter we solve forward elasticity problems to determine the displacement field for these specimens in response to a compression load. This displacement data and the corresponding label (benign or malignant) are used to train a simple CNN, whose performance is then tested on a different set of displacement data. It

is found that the trained CNN is able to correctly classify a lesion as malignant or benign with remarkable accuracy (≈99%) even in the presence of significant image noise.

Once the CNN is trained, we examine the weights of its convolution layers to better understand why it is successful, and what it has learned from the data. Since the convolutions are spatially local operators, they offer a straightforward interpretation as discrete local operators [155]. Generally speaking, we observe that the convolutions in the first layer behave as low-pass filters that smooth the input displacement data, while the convolutions in the following layers behave like finite difference operators that extract features in the displacement images by computing discrete spatial derivatives. This is not surprising given that several methods that are currently in use in elastography compute different measures of strain (derivatives of displacements) to infer something about the underlying elastic property distributions [140, 156].

Finally, we apply the trained nonlinear elasticity classifier to displacement data obtained from breast lesions of ten human subjects. We note that the classifier is trained using only simulation data, and is tested on real data. Remarkably, even under this circumstance, it is able to correctly classify eight out of ten lesions. This idea illustrates how one may use physics-based models to generate data to train machine learning (ML) algorithms which can then be applied to tasks involving "real" data. In the context of ML nomenclature, this is a form of transfer learning [157–159] where one trains the model for one task and uses the knowledge learned from that task to perform another task. What is remarkable about our approach is that we use a physics-based model to generate the training data which encodes valuable knowledge in our model. Thereafter, we apply this model to real-world data. Similar ideas have recently been explored in [**peng2017sim2real**] in the context of training a robot in simulated environments to perform certain tasks, and then testing its performance in a real-world scenario. One may also think of this approach as a form of data augmentation that makes use of the underlying physics to augment training data [152, 160, 161]. This data augmentation is physically consistent and is also "extreme" in the sense that while training

the net, only augmented data is used. We believe that ideas like these could prove useful in data-sparse applications like medical diagnostics.

The format of the remainder of this paper is as follows. Section 5.2 is focused on the computational methods used in this study and includes a description of the current and the proposed workflows for solving the classification problem, the process of generating training and testing displacement data, and the architecture of the convolutional neural network. Section 5.3 is focused on results and analysis. In particular, we quantify the accuracy of the CNN on noisy data, analyze some of the learned convolution filters, and also test the performance of the CNN on a small set of experimental/patient data. We end with concluding remarks in Section 5.4.

5.2 Computational Methods

5.2.1 Workflow

In a typical elasticity imaging problem, the workflow proceeds as follows (see Figure 5.1). First, the tissue is gently compressed using some external force. During this time, it is imaged with a conventional imaging modality such as ultrasound, MRI or OCT. The sequence of images thus obtained are used to determine the displacement field within the tissue using a technique like image cross-correlation. This displacement field is used in an inverse problem to determine the spatial distribution of mechanical properties, typically the material's linear elastic shear modulus (μ). In most cases, the inverse problem is solved as a constrained minimization problem, seeking the spatial map of the shear modulus which generates a displacement field that is closest to the measured displacement field [162, 163]. The shear modulus and the predicted displacements together are required to satisfy the constraints of the equations of equilibrium. The shear modulus images are then analyzed for features that may be useful in discerning the physiological state of the tissue. For example, in the case

of breast cancer, it is observed that the shear modulus distribution in malignant tumors is more heterogeneous and this feature can be used to improve diagnosis [149, 150].

When working with the nonlinear elastic response of tissue as the target feature, the workflow is more complex. In addition to a deformation field at small values of overall strain, another one at a large value of strain is measured. Using the steps outlined above, a map of the linear shear modulus is determined from the small-strain displacement data. Thereafter, using this map and the measured displacement field at finite strain, another inverse problem is solved and a map of the nonlinear elastic properties of tissue is generated [148, 164]. The precise nonlinear property determined depends on the constitutive model used for the tissue. For one such model, the uniaxial stress is approximately given by,

$$\sigma = \frac{\mu}{\gamma}(e^{\gamma\epsilon} - 1), \tag{5.1}$$

where μ is the shear modulus at small strains, ϵ is a measure of strain, and γ is the nonlinear parameter. Large values of γ imply large tangent modulus for the material at a given value of strain. Several ex-vivo and in-vivo studies have reported that the value of γ is elevated in malignant tumors, and that the average value of this parameter within the tumor is a good marker in distinguishing benign tumors from malignant tumors [148, 165].

In this chapter, we propose a simplified workflow for both linear and nonlinear elastic cases. We eliminate the complicated step of solving the inverse problem for determining elastic properties from measured displacements and automate the steps of identifying features and using these features to classify benign and malignant lesions. This alternative workflow is shown in Figure 5.2.

Within this alternative workflow, when working with linear elastic properties, we proceed as follows. We use the displacement images acquired at low overall strain as input to a convolutional neural network (CNN). In the training phase, we supply to the network an image and its corresponding label (benign or malignant). Once the network is trained, we test its performance on a different set of testing data.

When working with nonlinear elastic properties, we use a displacement images at a small strain ($\sim 1\%$) and at large strain ($\sim 20\%$). We normalize these images by the average value of displacement in each image, and compute another image that is the difference between the two. We term this image the "difference in displacement image." When the response of the material is linear elastic, we expect the normalized images at the two values of strains to be similar, and therefore the magnitudes in the difference image to be small. On the other hand, when the response is nonlinear, the normalized displacement images will be different and the magnitudes in the difference image will be large. The difference in displacement image is used as input to the CNN, whose output is once again the classification into benign and malignant cases.

In all cases presented in this chapter, the displacement images used to train the CNNs are generated synthetically. That is, we assume a certain form of material property distribution for benign and malignant lesions, which broadly resembles the material distribution in actual lesions, and generate multiple samples of each class. For each of these, we generate the corresponding displacement images by "compressing" the tissue virtually using a finite element method. The displacement images thus obtained are the input to the CNN. Our objectives are:

1. To demonstrate that CNNs can be trained to discern differences in elastic features (heterogeneity and nonlinearity) from displacement images without having to solve a complicated inverse problem.

2. To analyze the learned convolution filters in order to understand how the CNNs are performing the classification task.

3. To apply the CNN-based classifier to a small set of real human subject data to discover whether a net trained on simulated data can be used to classify real data.

5.2.2 Generation of training and testing data sets

The displacement data used to train and test the CNN is obtained by solving a forward elasticity problem using the finite element method. We model the tissue as an incompressible isotropic hyperelastic solid. The stress-strain response for this material is completely specified by a strain energy density function, and for this we use a function with an exponential dependence on strain. In particular, the strain energy density is given by [166],

$$W = \frac{\mu}{2\gamma}\big(e^{\gamma(J^{-2/3}I_1-3)} - 1\big). \tag{5.2}$$

Here I_1 is the first invariant of the Cauchy-Green strain tensor, J is the determinant of the deformation gradient, μ is the linear shear modulus for the material, and γ is the nonlinear elastic parameter. For large values of γ the material tangent modulus increases more rapidly with applied strain. It has been reported that malignant lesions in the breast are characterized by a heterogeneous shear modulus distribution μ and by larger values of the nonlinear parameter γ [148–150, 165]. We are guided by these observations in generating virtual samples of benign and malignant tumors.

For the heterogeneity study, the shear modulus for both benign and malignant lesions is represented as the sum of two elliptical distributions whose boundaries are smoothed by a Gaussian filter. The variation of the modulus within these inclusions is quadratic and the parameters for this variation are selected from a uniform distribution (see Appendix). The coordinates for the centers of the inclusions are also selected from a uniform distribution. For benign lesions, this range is such that the two centers are close and the lesion appears to be a single large inclusion. For malignant lesions the range is selected so that the centers are further apart and the lesion appears to have two distinct foci giving it a more heterogeneous appearance. This is seen in the modulus distributions for two typical samples from each class shown in Figure 5.3. The precise formula used to generate the shear modulus distributions is described in the Appendix.

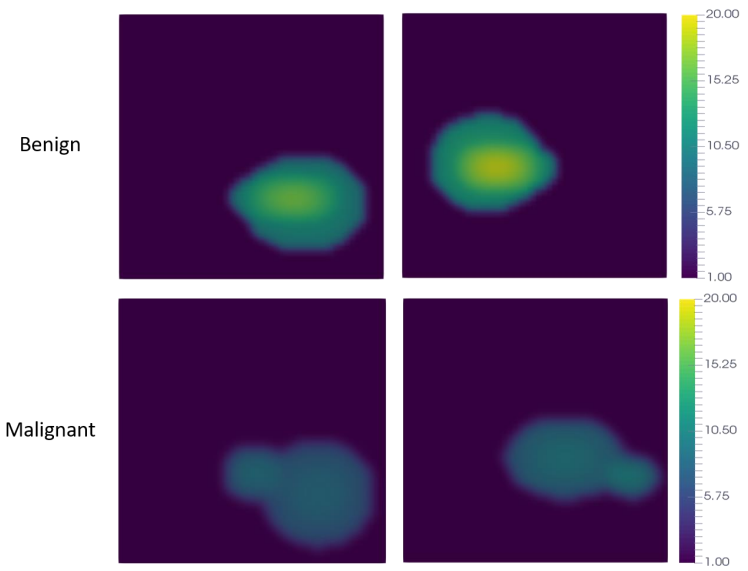


Figure 5.3: Two typical shear modulus distributions (μ) for benign and malignant classes.

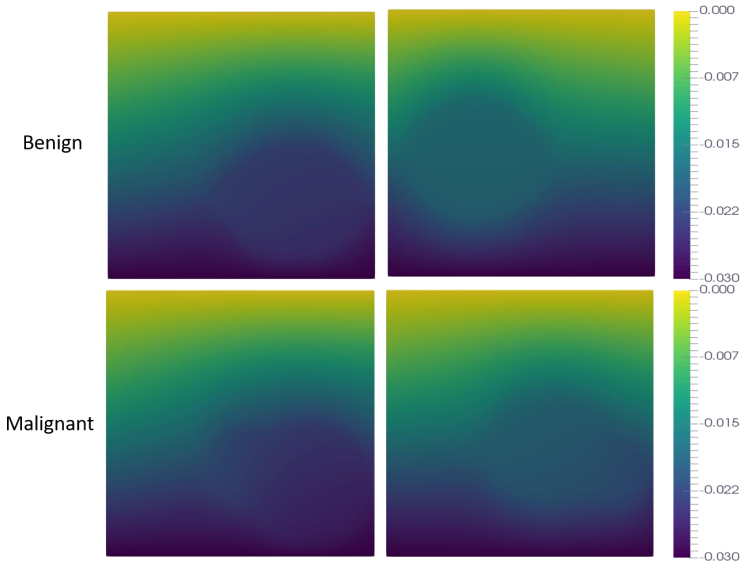


Figure 5.4: Two typical displacement distribution images for benign and malignant classes.

We would like to emphasize that these modulus distributions are not the input to the CNN. Rather, samples with these modulus distributions are compressed to a small strain and the resulting axial component of the displacement (shown in Figure 5.4) is used as input to the CNN. From examining this figure we conclude that classifying samples through a visual inspection of the displacement images would not be an easy task. Our goal is to assess the ability of a CNN to perform this task.

In the nonlinear study the shape of the lesion among malignant and benign classes is kept similar. That is, the range of parameters used to generate the shapes of the shear modulus and nonlinear parameter distributions is identical for the two classes. They differ in the mean value for the nonlinear parameter, which is larger for the malignant class (see Figure 5.5), as is typically observed in the real world. The precise formulae used to generate these distributions are reported in the Appendix.

Once again we emphasize that we do not use these material property distributions as input to the CNN. Instead we use the response of these lesions to compression, as it would be measured in a typical elastography setup, as the input. In particular, we compress each lesion to 1% overall strain and compute the displacement field. Then we compress it to 20% overall strain and once again compute the deformation field within the sample. We then normalize axial component of both displacements and compute the difference between the large-strain displacement and the small-strain displacement fields. This difference image is used as input to the CNN. The difference images for two typical cases selected from the malignant and benign classes are shown in Figure 5.6. From this figure it is difficult to discern obvious differences between the two classes.

5.2.3 Convolutional neural networkworkworks

The architecture of the CNN used in the heterogeneity study is shown in Figure 5.7. The input is a 50×50 image. The CNN contains four convolution stages, where each stage is comprised of several convolution filters and a Re-LU activation. The first two stages also

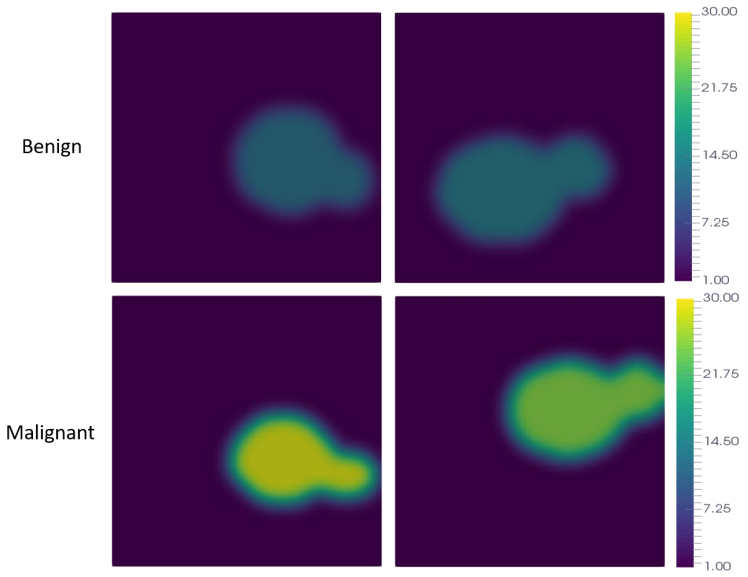


Figure 5.5: Two typical nonlinear modulus (γ) distributions for benign and malignant classes.

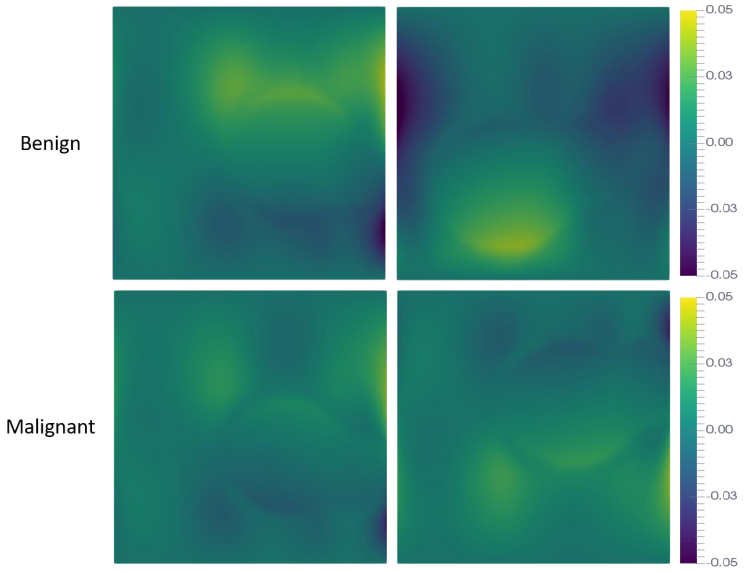


Figure 5.6: Two typical "difference in displacement" images for benign and malignant classes.

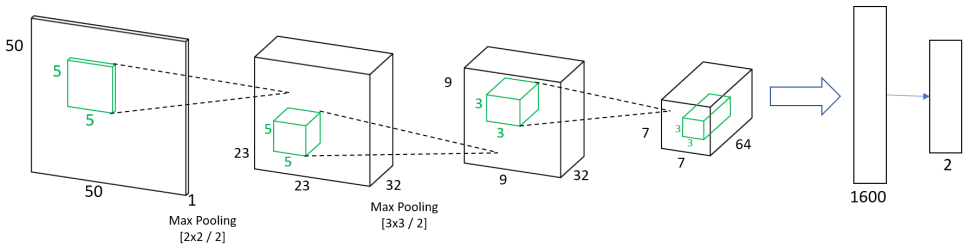


Figure 5.7: CNN architecture for the shear modulus heterogeneity study.

contain a max-pooling operation which reduces the dimension of the image in the hidden layers and allows the algorithm to step through a hierarchy of coarse spatial scales. It ensures that the size of image at the end of the convolution layers is small, thereby constraining the total number of weights in the network.It also introduces translation invariance to the network. As the image is processed through the convolution stages, its size in the vertical and horizontal directions decreases; however its size in the feature space increases (number of channels). After the fourth stage the image is flattened into a $1 \times 1 \times 1600$ vector. This stage is followed by a fully connected layer which leads to a simple soft-max classifier.

In this study we have attempted to keep the architecture of the CNN as simple as possible. For the heterogeneity study, we started with three convolution stages and found that these could only provide around 90% accuracy at high noise levels. Increasing the number of layers to four dramatically improved the performance.

The architecture of the CNN used in the nonlinear study is shown in Figure 5.8. The input is once again a 50×50 image. This is followed by three convolution stages that lead to a $1 \times 1 \times 3136$ vector. This stage is followed by a fully connected layer, which leads to a simple soft-max classifier. For the nonlinear study, we found that three convolution stages were sufficient in delivering very good accuracy.

5.3 Results and Analysis

In this section we describe the process of training the neural networks and quantify their performance on test data. We also analyze the convolution layers for the heterogeneity study and then test the performance of the nonlinear elasticity classifier on real data.

5.3.1 Training and performance

For the heterogeneity study we use 4,000 displacement images and the corresponding labels for each class to train the CNN. Two typical displacement images from each class are shown

in Figure 5.4. In order to test the robustness of the resulting net to noise in measurement, we consider cases with 0, 1, 3, & 10% additive Gaussian white noise.

We train the net on a single GPU using the Adam optimizer [167] with initial learning rate of 0.001 and standard value of parameters β_1=0.9 and β_2=0.999. We select a batch size of 100 with an l_1-regularization parameter of 1e-6 and train the net for 500 epochs in Tensorflow [168]. The performance of the net in correctly classifying a set of testing data (1,000 images of each class) as a function of the number of epochs is shown in Figure 5.9. In the jargon of ML, an epoch is completed when the stochastic gradient algorithm has cycled through the entire training data once. From the figure we observe that the training process proceeds smoothly for all levels of noise (except for a blip for the 1% case) as the testing accuracy increases almost monotonically in all cases.

Table 5.1: Performance of the CNN at different levels of noise (heterogeneity study).

Noise level (%)	Accuracy	Specificity	Sensitivity
0	99.95%	100%	99.9%
1	99.90%	100%	99.8%
3	99.95%	100%	99.9%
10	99.75%	99.7%	99.8%

The performance of the fully-trained CNN (at the end of 500 epochs) is reported in Table 5.1. Here we observe that for all levels of noise, the CNN achieves very good accuracy (99.7-99.9%). It also yields very good values of sensitivity and specificity. We remind the reader that sensitivity is the ratio of the number of correctly classified malignant lesions to the total number of malignant lesions, and specificity is the ratio of the number of correctly classified benign lesions to the actual number of benign lesions. These results makes it clear that the CNN has learned to analyze the input displacement data to infer the heterogeneity in the spatial distribution of the shear modulus, thus by-passing some of the complex steps shown in the workflow in Figure 5.1.

For the nonlinearity study we use 2,000 difference in displacement images and the corresponding labels for each class to train the CNN (see Figure 5.6 for typical images). Once

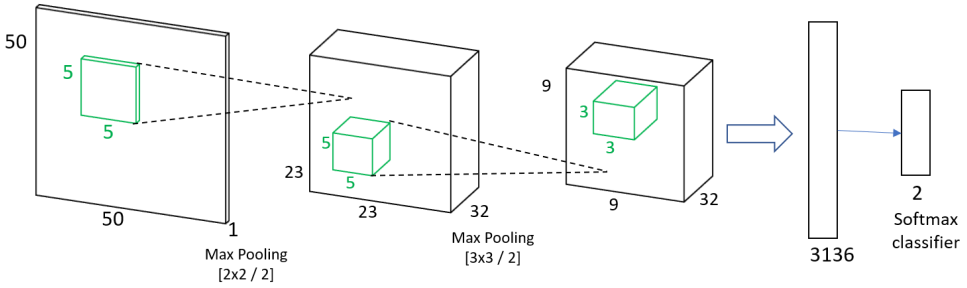


Figure 5.8: CNN architecture for the nonlinearity study.

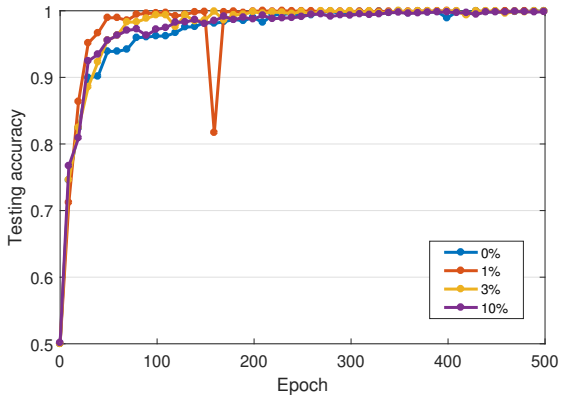


Figure 5.9: Test accuracy at different noise levels for heterogeneity study.

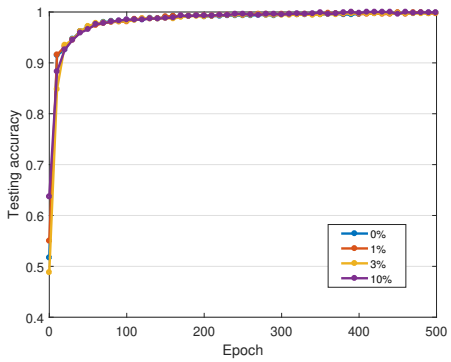


Figure 5.10: Test accuracy at different noise levels for nonlinearity study.

again we consider 0, 1, 3, & 10% Gaussian noise cases. The regularization parameter and the learning rate are the same as in the heterogeneity study. The performance of the network in correctly classifying a set of testing data (500 images of each class) as a function of the number of epochs is shown in Figure 5.10, where we observe that accuracy improves smoothly with increasing number of epochs.

Table 5.2: Performance of the CNN at different levels of Gaussian noise (nonlinearity study).

Noise level (%)	Accuracy	Specificity	Sensitivity
0	99.8%	100%	99.6%
1	99.7%	99.6%	99.8%
3	99.8%	99.8%	99.8%
10	99.9%	99.8%	100%

The performance of the fully-trained nonlinear CNN at the completion of 500 epochs is quantified in Table 5.2. Once again we observe that the trained network accurately classifies lesions (99.6-100% accuracy) with good sensitivity and specificity, even at high levels of noise. These results indicate that CNN has learnt to use the input displacement data to infer whether some region within the domain is displaying a nonlinear elastic response.

5.3.2 Analysis of convolution layers

The convolution filters employed in a CNN are local. For example, for the heterogeneity study, the filters are 5×5 in the first two stages and 3×3 thereafter. Further, each stage contains multiple filters whose components are learned by the CNN during the training process. Once the training process is complete, analyzing these filters can provide insight into how the CNN operates on input data in order to generate features that are relevant for its task (classification or regression). With this in mind, in this section we analyze the filters learned by the CNN for the heterogeneity and nonlinear elasticity studies. We examine a given convolution filter both in physical space by mapping its components as a matrix, and in the Fourier space by performing a fast Fourier transform (FFT) on these components and mapping the magnitude of the transform. In order to understand these filters in physical

space, we look for the closeness of a given filter to other well-understood local operators like finite difference stencils; in the Fourier space, where the convolution operation transforms to multiplication, we view these filters as spectral transforms.

In Figure 5.11, we have displayed all the active convolution filters from the first stage of the heterogeneity-based CNN classifier. For each filter, we have also plotted the corresponding absolute value of its Fourier transform. In the Fourier transform map the $k_x = k_y = 0$ component is at the center of image and the horizontal and vertical axes represent wavenumbers (k_x and k_y, respectively) along those directions. From the maps of the filters in physical space we observe that they are dominated by values that are all of the same sign (uniformly negative). Thus the main effect of these convolution filters is to smooth out a given image. Essentially, in the output image of the first stage, every pixel will be an average of the input image computed over a 5×5 pixel region centered about the same location. This is also seen by examining the Fourier transform of these filters where we observe that the zero wavenumber component is dominant and there is a decay in the spectrum at high wavenumers. Thus, the first convolution stage can be understood as a smoothing stage where the CNN has learned to average out the noisy displacement image to mitigate the effect of noise.

In Figure 5.12, we examine some of the active filters from the second convolution stage. From the map of the filters in physical space we observe that they are mostly centered around zero and contain positive and negative values in (more-or-less) equal measures. Further, the variation along a given direction is monotonic. For example in the first image, the components appear to vary most significantly at 45° to the horizontal and this variation is mostly monotonic, going from negative in the bottom-left corner to positive in the top-right corner. Thus, this convolution represents a first order derivative along $x = y$. This can also be seen by examining the Fourier transform of the filter where we observe that it has a small magnitude at origin and rises symmetrically along the line $k_x = k_y$. Further it has small values at large wave numbers. This indicates that the filter is likely a first-order derivative

along the line $x = y$ convolved with a low pass filter that suppresses high wavenumbers. One choice for the spectrum of a filter with these properties is

$$F(k_x, k_y) = (k_x \cos\theta + k_y \sin\theta) \times \exp(\frac{-k^2}{2\sigma^2}), \tag{5.3}$$

with $\theta = 45°$. Here the linear part accounts for the first order derivative and the exponential part represents the low-pass filter. The magnitude of this spectrum is shown the Figure 5.13 (image on the right). We can clearly see similarities between this plot and the first image in Figure 5.12. A quantitative comparison between these filters is shown in Figure 5.14, where we have plotted their spectrum along $k_x = k_y$. From this figure we conclude that assumed form described above is a good approximation to the actual filter. Similarly, all the other filters in Figure 5.12 predominantly represent a smoothed version of a first order derivative along different directions.

In summary, most convolution filters in this stage represent first order derivative operators. Considering that these filters operate on the smoothed displacement field, which is the input to the second stage, the output of this stage is different components of strain in the specimen. The fact the the CNN has learned to use some measure of strain as an important feature for solving the classification problem is consistent with common practice in elastography, where practitioners often use different components of strain to visualize the underlying mechanical properties.

In Figure 5.15, we have plotted some of the active filters and their FFT from the third convolution stage. Note that these filters have a smaller width (3×3). From examining these filters it is clear that like the filters in the second stage they are are essentially discrete first-order operators along different directions. By comparing their transforms with the analytical transforms shown in Figure 5.14, we conclude that most of the these directions are along $\theta = 90°$. Given that these filters act on the output of the second stage, which in turn is a obtained after a first-order derivative operation on the original image, the output of the third

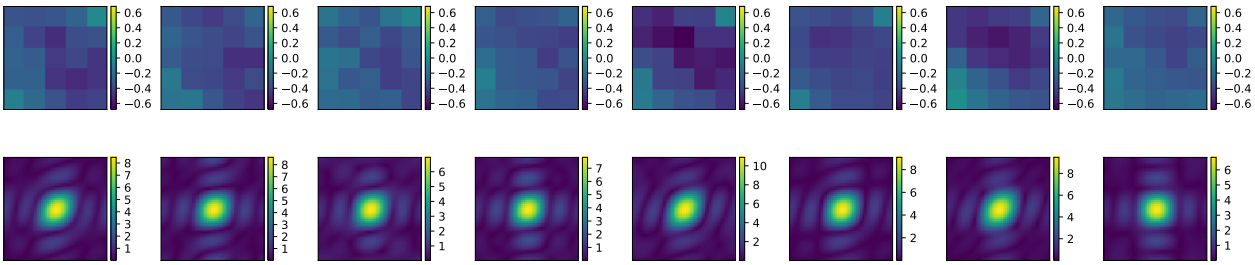


Figure 5.11: Learned weights (top row) and corresponding Fourier transform (bottom row) of active convolution filters in the first layer/stage.

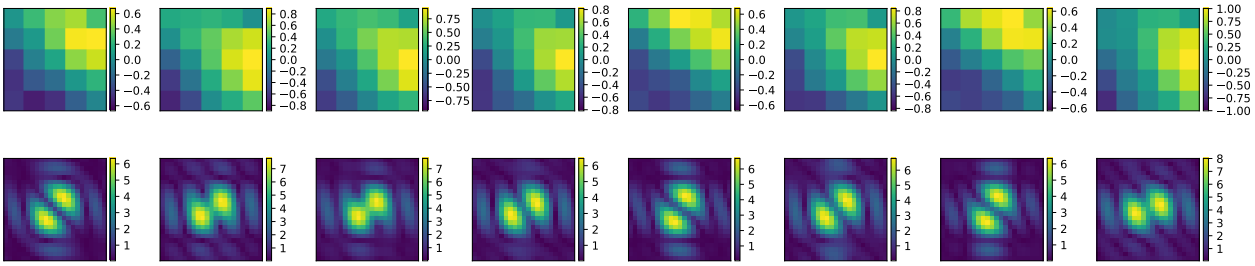


Figure 5.12: Learned weights and corresponding Fourier transform of some typical active convolution filters of second layer/stage.

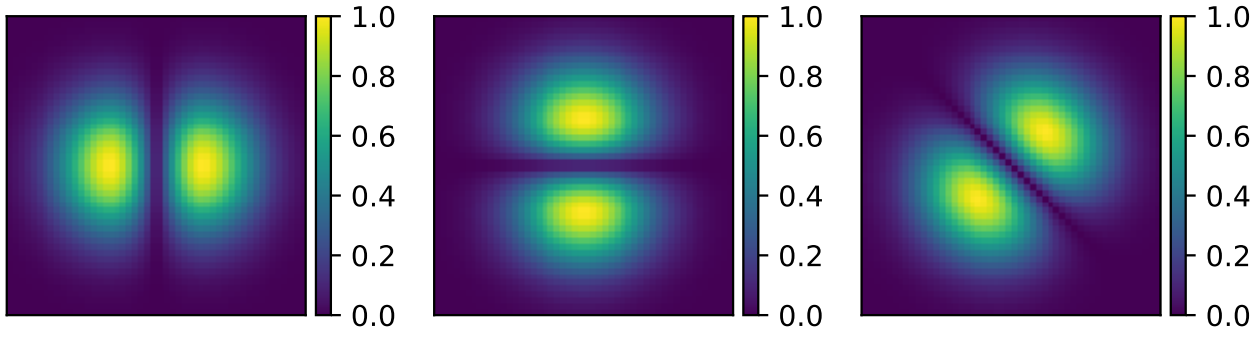


Figure 5.13: Spectrum of filters that represent a first order derivative with a low-pass Gaussian filter (from left to right: $\theta = 0, 90, 45°$).

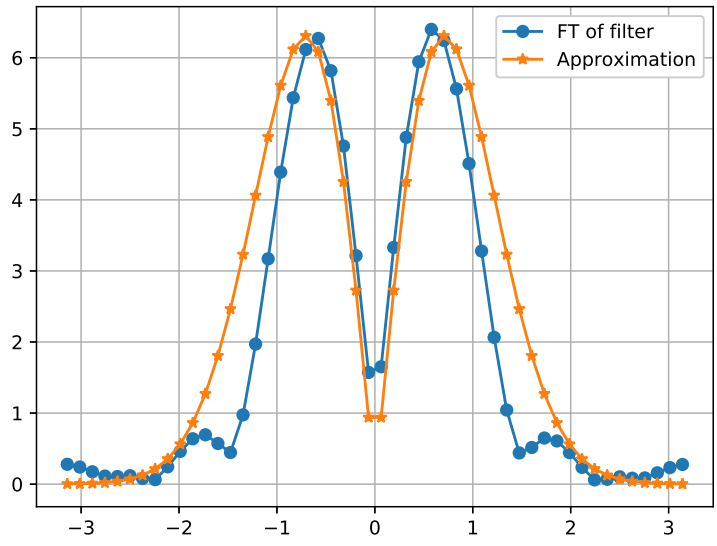


Figure 5.14: Spectrum of an actual filter and an approximation plotted along $k_x = k_y$.

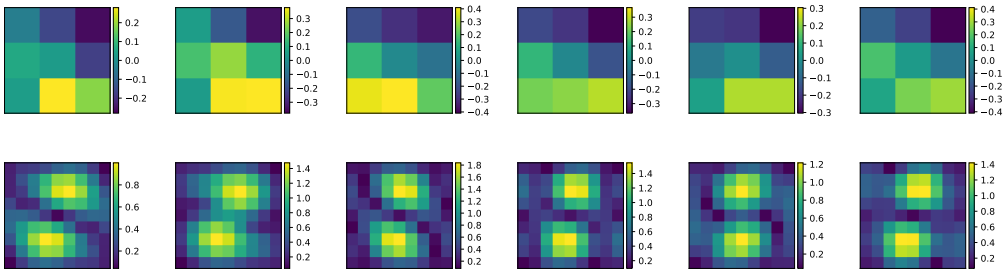


Figure 5.15: Learned weights and corresponding Fourier transform of some typical active convolution filters of third layer/stage.

layer contains a mixture of smoothed second-order derivatives of the original displacement image.

From the analysis above we begin to develop an understanding of the working of the heterogeneity-based CNN classifier and the possible connection of this learning process with some of the traditional elastography techniques. In the first convolution stage, it smooths and coarsens the input displacement image, while in subsequent stages it computes strains and their derivatives (with coarsening resolution) to look for features that are useful in solving the classification problem. We believe that the knowledge gained from this analysis could be used to limit the number of channels to be equal to the number of filters that are active in each layer. This could reduce the number of weights in the network. A similar analysis done for a large network could also be used to eliminate some layers that are performing operations that are close to an identity operator. However, this is not applicable to our network which is small to begin with.

The convolution filters for the nonlinear elasticity-based CNN (not shown here) are similar.

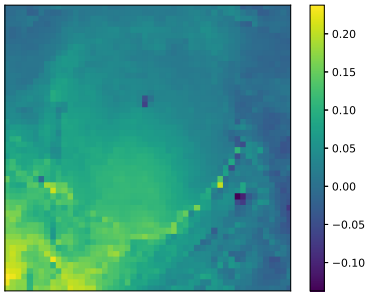
5.3.3 Performance on real data

In the previous section, we have trained two CNN-based classifiers solely using simulated data. In this section we apply one of these (the nonlinear elasticity classifier) to real-world patient data and assess its performance. The fact that this classifier has been trained on simulated data, which is easy to come by, and will be applied to real data (which may be scarce) points to how deep learning techniques can be applied in domains where real data is scarce. It also describes how physics-based modeling can complement data-based techniques in data-scarce fields by providing additional training data. This approach can be understood in the context of two very useful and widely researched topics in machine learning: transfer learning and data augmentation.

In transfer learning, the objective is to reuse the knowledge gained from one task to improve the performance on another task. This is often accomplished by training the network for one task and using the weights of the learned model for performing a different task, while re-learning only a small subset of the weights during the second task. In the approach described in this section, the first task is the classification of simulated data and the second task is the classification of real data. However, in this case all the learning happens during the first task and there is no fine-tuning of weights in the second task. Recently, this approach of learning purely from simulation-based data has been applied to machine learning applications that include self-driving cars and robotics [169–171], and is gaining in popularity. It is particularly useful in domains where real data is scarce or is risky and/or expensive to gather. Here we present, for the first time, its application to a problem in medical diagnosis.

One can also interpret our approach as a form of data augmentation, where a given set of data (say images) is transformed to generate a larger set and this larger set is used to train the net. In the case of images this transformation may involve rotating, translating, cropping or scaling the input images to create many more copies to train the network. Applying these types of transformations to data that is governed by physical principles can clearly lead to improbable data, which would likely hurt the performance of the network. For example, in our case if we were to rotate the displacement data and use that to train the network, we would be including displacement fields that are physically inconsistent with the process of compression in the vertical direction. Thus, in augmenting data it is desirable to respect the underlying physical principles. When viewed from this context, our approach of using only simulated data in the training set can be viewed as physically-consistent and extreme data augmentation. The approach is physically-consistent because the augmented data is consistent with the underlying physical principles and it is extreme because the training comprises only of augmented data.

We consider displacement data acquired from ten human subjects that presented with breast lesions. It was determined through biopsies, that five were benign (fibroadenoma)



[Benign] [Malignant]

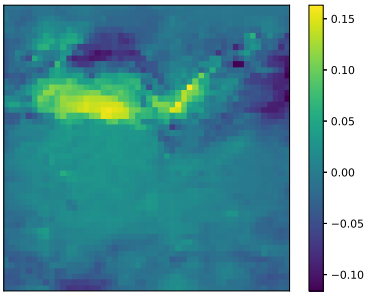


Figure 5.16: Typical 'difference in displacement' image of each class for real data

and five were malignant (invasive ductal carcinoma) lesions. Displacement data at around 1% and 12% overall strain was acquired using ultrasound elastography as described in [148]. The displacement images were then downsampled to a 50×50 pixels range and normalized as described in Section 5.2. The difference in displacement image was generated by subtracting the normalized small-strain displacement image from the normalized large-strain displacement image. The resulting displacement images for one benign and one malignant lesion are shown in Figure 5.16. These images, along with eight other such images, were used as input to the trained nonlinear-elasticity-based CNN classifier (trained on 3 % noise), and the classification results were compared with the determination made through biopsy.

Table 5.3: Confusion matrix for physics based transfer learning

Patient data (N = 10)		Actual Benign	Actual Malignant
Predicted	Benign	4	1
	Malignant	1	4

The resulting confusion matrix is shown in Table 5.3. We note that the classifier was correctly able to classify 4 out of 5 benign and malignant lesions leading to an accuracy of 80%. While this level of accuracy is comparable with other analysis techniques used in conjunction with elastography, what is remarkable here is that in training the CNN no real data was used. It was completely trained with simulated data and then this knowledge was transferred to solve a real-world problem. It is likely if the net were re-trained while accounting for real data, perhaps with higher weights, it would perform better. However, this requires a larger set of real data and will be considered in the future. We also note that the number of patients considered here is too small to draw definitive conclusions about the robustness of the approach. In order to do that a larger set of testing data will be required.

5.3.4 Comparison of ML-based and inverse problem-based approaches

In this section we return to comparing an ML-based approach for solving a given classification problem with an inverse problem-based (IP-based) approach. In order to make this discussion concrete, we first describe some pre-requisites for this comparison. We assume

1. The existence of a forward model that maps a material fields, μ and/or γ, to an observed response field u. That is the existence of a forward map.

2. The ability to solve for the inverse map, that is to determine (μ, γ) from the observed field u, albeit with some computational effort.

3. The desire to solve a classification problem (benign or malignant, for example) using either u or (μ, γ) as input.

We consider two scenarios for our problem, both of which are quite common in the types of classification problems we have in mind. These are:

1. Knowledge-rich scenario where we assume that significant domain-specific knowledge exists (mostly for the material property distribution) which can be used to directly

classify a sample material distribution. For example, knowing that the nonlinear elastic parameter is elevated in malignant lesions might be sufficient to classify a sample once the nonlinear elasticity parameter distribution is determined.

2. Data-rich scenario where lots of labeled data is available. In our example this would mean that we have a library of measurements of the displacement field u and the corresponding biopsy-proven labels that determine whether a lesion is malignant or benign.

5.3.4.1 ML-based approach

Within the data-rich scenario the application of the ML-based approach is straightforward and described in the workflow shown in Figure 5.2. It involves training a CNN-based classifier using the available measured displacement data and labels, and then using this classifier in a forward mode to make a classification. This approach can also be implemented in a knowledge-rich scenario where the measured training data may not be available. In this case one could create synthetic training data (displacement fields) by first generating synthetic samples of material property distributions that are consistent with the prior knowledge, and then solving the forward problem to produce the corresponding displacement fields. Indeed, this is the approach we have used in Section 5.3.3.

Computational costs The costs associated with implementing the ML-based approach can be split into two categories : set-up cost and run-time cost. Set-up costs are one-time costs that are incurred while training the classifier and run-time costs are incurred whenever the classifier is used to make a prediction. Both these costs are strongly influenced by parameters like size of the input data, number of layers in the CNN, and the size of the fully connected layers. The training costs are also influenced by the size of the training set, the regularization parameter and the learning rate, the epoch size, and other such variables.

The time spent in performing these tasks is ~~heavily~~ significantly influenced by the hardware (GPU accelerators) and software libraries used for training and prediction. In order provide the reader with a ball-park estimate of this time, we report the wall-clock time spent in performing these tasks while solving the non-linear elastic problem described in Section 5.3.3 (see Table 5.4). We note that we have made very little effort in trying to optimize our algorithm or its execution in order to reduce these times. From this table we observe that the time spent in training the network is significant, and that very little time is spent in running it to make a prediction. Since an end-user will only use the fully-trained network in a predictive mode, an attractive feature of the ML-based approach is that it demands very little computational effort from the end user.

Approach	Set-up time (minutes)	Prediction time (minutes)
ML-based	51.45 (training)	0.003 (classifying)
IP-based	-	68.05 (classifying)

Table 5.4: Wall-clock time for solving the nonlinear elasticity classification problem on a AMD Phenom II, 6 core processor.

Accuracy Assessment of the accuracy of the ML-based algorithm in a data-rich scenario would require the measurement of displacement data on a large number of subjects with biopsy-proven diagnosis. To our knowledge this type of dataset does not exist as of now, and therefore this quantification is not possible. On the other hand, in this chapter we have applied this ML-based approach in the knowledge-rich scenario where we have used the forward map to generate the data to train the classifier. In this case, when running the the classifier on synthetic data we achieve nearly perfect accuracy (99.7 % -99.9%) and on real data achieved an accuracy of 8/10. Both these results are promising. A couple of remarks are in order about the results with real data. First, a sample size of 10 is too small to draw definitive conclusions. Second, the accuracy of this approach can be improved further by

training the CNN using hybrid data-set (containing both real and simulated samples) once more samples of real data is available.

5.3.4.2 IP-based approach

The workflow for the IP-based approach is described in Figure 5.1 within a knowledge-rich scenario. Here the measured displacement data is transformed to material properties, and these are used to determine simple quantities of interest, which then drive ~~s~~ a simple classifier. There is no need to train the classifier as it is determined from prior knowledge; for example, knowing that malignant lesions are more elastically nonlinear.

Computational costs There are no one-time or set-up costs for this approach. The prediction costs are dominated by the costs associated with solving an inverse problem. These are challenging problems to solve, and over the year significant progress has been made to make their solution tractable. This includes the use of adjoint-based methods, Newton-Krylov-Schur methods and other such approaches [172, 173]. In the end these costs are determined by the number of inverse parameters and the number of directions along which data is informative.

In Table 5.4, we report the time spent in solving this problem for the nonlinear elasticity problem considered in this paper. The inverse problem was solved using a quasi-Newton method in conjunction with adjoint equations to compute the gradient [174]. No effort was made to reduce this time through parallelization or through other techniques. We note that this time is comparable to the time spent in training the CNN (which is a one-time cost) and is significantly more than the time incurred by the ML-based approach. On this basis we may conclude that the ML and IP-based approaches have comparable computational costs when only a few predictions are required. However, when the intent is to make many predictions, the ML-based approach incurs less costs.

Accuracy Based on the results presented in this chapter it is possible to compare the accuracy of the IP- and ML-based approaches in solving the nonlinear elasticity classification problem. With real data, the ML-based approach classified 8/10 samples correctly. The IP-based approach achieved an accuracy of 9/10 on the same data set [175]. Based on these observations we conclude that for this specific problem, the IP-based approach is slightly more accurate. However, we would like to note that number of samples is too small to draw definitive conclusions.

5.3.4.3 Hybrid approach

Next we consider how we might apply the IP-based approach, which involves mapping the measured displacements to material properties, in a data-rich scenario. In this scenario, the IP-based approach also requires a classifier that is applied to the QoI obtained from solving the inverse problem. This leads to the lower half of the workflow shown in Figure 5.17, where we begin with the measured displacement, transform this to a material property distribution, compute QoIs, and provide these as input to a ML-based classifier. An interesting extension of this approach would be to also use the measured displacements as input to a CNN-based algorithm to generate a small-sized feature vector, and then use this and the QoI obtained from solving the inverse problem as input to a ML-based classifier (all of Figure 5.17). Note that this hybrid approach combines domain-specific knowledge (by targeting a given QoI) with knowledge learned from data, it is likely to be more accurate than either the ML or IP-based approach. However, it also includes steps from both the ML and IP-based approaches, and thus its computational costs will exceed that of either one.

5.4 Conclusions

In this paper, we have demonstrated the use of CNNs in classifying, with a high degree of accuracy, different specimens based on their elastic heterogeneity and nonlinearity. The

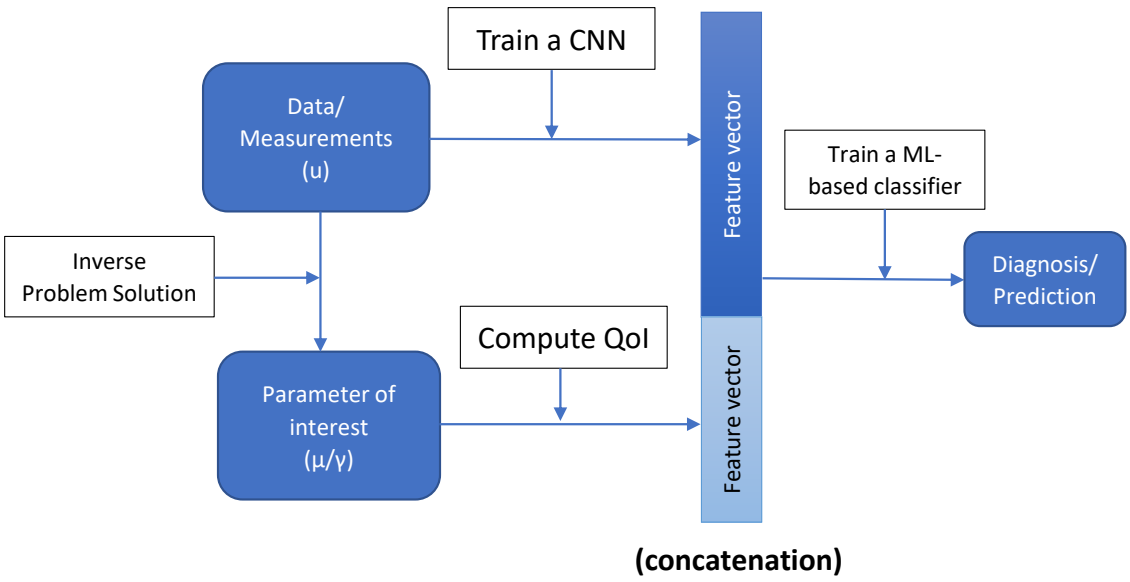


Figure 5.17: Hybrid ML/IP approach for solving a classification problem.

input data to the CNNs is the measured displacement field and they are able to solve the classification problem while circumventing the need to solve a complex inverse elasticity problem. It is likely that the same approach can also be extended to other physics-based classification problems such as those involving thermal or electromagnetic characterization.

We have analyzed the convolution filters of the trained network to better understand how it performs the classification task. We found that it does so by learning two types of filters. These include (a) smoothing filters that help in ameliorating the effect of noise and in coarse-graining data and (b) finite-difference-stencil like filters that enhance features through successive differentiation. It is remarkable, though not surprising, that these types of filters (strain filters) are currently used by practitioners in elastography.

Finally, by applying a network that is trained solely using simulated data to a real-world classification problem, we have demonstrated how physics-based modeling can facilitate transfer learning in data-scarce applications.

Appendix

Here we describe how the material parameters distributions for the shear modulus (μ) and the nonlinear elastic parameter (γ) were generated. Maps for both these parameters were created by superposing two elliptical inclusions as shown in Figure(5.18). Note that (X_1,Y_1) represents the center of a large ellipse with semi-major axis a_1 and semi-minor axis b_1. Similarly, (X_2,Y_2) represents the center of a smaller ellipse with semi-major axis a_2 and semi-minor axis b_2.

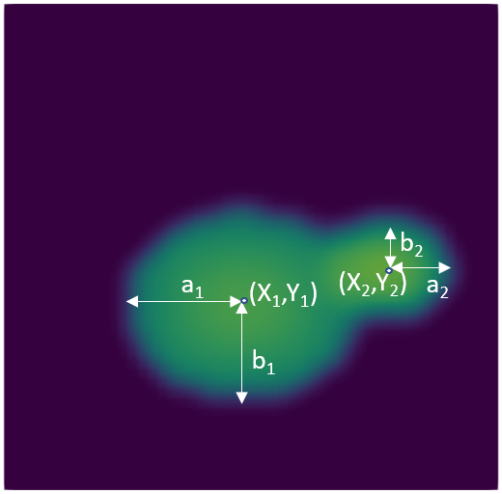


Figure 5.18: A typical material parameter distribution.

The material parameter distribution was given by,

$$\beta(x,y) = G\left(\sum_{i=1}^{2}\chi_i\left[\beta_0 + \hat{\beta}_i\left\{1-\left(\frac{x-X_i}{a_i}\right)^2\right\}\left\{1-\left(\frac{y-Y_i}{b_i}\right)^2\right\}\right] + 1 - [\chi_1+\chi_2-\chi_1\chi_2]\right) \tag{5.4}$$

Here, $\beta(x,y)$ represents the value of the shear modulus (μ) or the nonlinear parameter (γ) at any point (x,y) inside the domain, G is the two-dimensional Gaussian filter with standard deviation of 0.5 and $\chi_i(x,y)$ is the indicator function defined as,

$$\chi_i(x,y) = \begin{cases} 1, & \left(\frac{x-X_i}{a_i}\right)^2 + \left(\frac{y-Y_i}{b_i}\right)^2 \leq 1 \\ 0, & \text{, else} \end{cases}$$

The parameters that appear in (5.4) are sampled from a uniform distribution ($\mathcal{U}$) as shown below. In the heterogeneity study for both classes

$$\begin{aligned}
\mu_0 &\sim \mathcal{U}(25, 35) \text{ kPa} \\
\hat{\mu}_1 &\sim \mu_0 \times \mathcal{U}(0.45, 0.6) \text{ kPa} \\
\hat{\mu}_2 &\sim \mu_0 \times \mathcal{U}(0.6, 0.8) \text{ kPa} \\
a_1 &\sim l \times \mathcal{U}(0.08, 0.15) \text{ mm} \\
b_1 &\sim l \times \mathcal{U}(0.06, 0.1) \text{ mm} \\
a_2 &\sim a_1 \times \mathcal{U}(0.45, 0.65) \text{ mm} \\
b_2 &\sim b_1 \times \mathcal{U}(0.50, 0.65) \text{ mm} \\
X_1 &\sim l \times \mathcal{U}(0.25, 0.75) \text{ mm} \\
Y_1 &\sim l \times \mathcal{U}(0.25, 0.75) \text{ mm}.
\end{aligned}$$

Here, $l = 29.4$ mm is the length of the square domain.

In addition, for the benign class the centers of the ellipses were chosen so that they were close

$$\begin{aligned}
X_2 &= X_1 + \phi(X_1) e_x \\
Y_2 &= Y_1 + \phi(Y_1) e_y \\
e_x &\sim a_1 \times \mathcal{U}(0.20, 0.30) \text{ mm} \\
e_y &\sim b_1 \times \mathcal{U}(0.05, 0.15) \text{ mm},
\end{aligned}$$

where $\phi(\xi)$ is a switch given by

$$\phi(\xi) = \begin{cases} 1, & 0 \leq \xi < \frac{l}{2} \\ -1, & \frac{l}{2} \leq \xi \leq l. \end{cases}$$

Whereas in the malignant class they were further apart,

$$\begin{aligned} X_2 &= X_1 + \phi(X_1)(a_1 + a_2 - e_x) \\ Y_2 &= Y_1 + \phi(Y_1)e_y \\ e_x &\sim a_1 \times \mathcal{U}(0.35, 0.45) \text{ mm} \\ e_y &\sim b_1 \times \mathcal{U}(0.35, 0.45) \text{ mm}, \end{aligned}$$

leading to a more heterogeneous distribution.

For the elastic nonlinearity study the parameters were sampled from

$$\begin{aligned} \mu_0 &\sim \mathcal{U}(25, 35) \text{ kPa} \\ \hat{\mu}_1 &\sim \mu_0 \times \mathcal{U}(0.45, 0.6) \text{ kPa} \\ \hat{\mu}_2 &\sim \mu_0 \times \mathcal{U}(0.6, 0.8) \text{ kPa} \\ \hat{\gamma}_1 &\sim \gamma_0 \times \mathcal{U}(0.45, 0.6) \\ \hat{\gamma}_2 &\sim \gamma_0 \times \mathcal{U}(0.6, 0.8) \\ a_1 &\sim l \times \mathcal{U}(0.08, 0.15) \text{ mm} \\ b_1 &\sim l \times \mathcal{U}(0.06, 0.1) \text{ mm} \\ a_2 &\sim a_1 \times \mathcal{U}(0.45, 0.65) \text{ mm} \\ b_2 &\sim b_1 \times \mathcal{U}(0.50, 0.65) \text{ mm} \end{aligned}$$

for both classes. The only difference was in the magnitude of the nonlinear parameter γ_0. For the benign class it was small,

$$\gamma_0 \sim \mathcal{U}(5, 15).$$

Whereas for the malignant class it was large,

$$\gamma_0 \sim \mathcal{U}(25, 45).$$

In this study the shape of the tumor for both classes is kept similar, and so X_1, X_2, Y_1 and Y_2 are chosen in the same as they are calculated for the malignant class in the heterogeneity study.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In summary, the goal of this work was to tackle the ill-posedness of the inverse problems and advance the state-of-the-art of the current deterministic as well as stochastic inversion algorithms. This was achieved over the course of several projects with the development of novel algorithms and their applications to various domains ranging from thermal and medical imaging to computer vision and machine learning.

One of the major roadblocks of applying stochastic inverse problems to practical science and engineering problems is the inability of existing algorithms to do full posterior characterization and uncertainty quantification (UQ) in high dimensions. Further, there are certain situations where we have qualitative information or domain knowledge that is difficult to quantify in an analytical form useful for solving stochastic inverse problems. We addressed both this issue by developing novel algorithm which used the Generative Adversarial Network (GAN) as a prior in Bayesian update and leveraged the low-dimensionality of the latent space of the GAN to perform efficient posterior inference. We applied this algorithm on a series of inverse problems arising in thermal imaging, elasticity imaging, and material microstructure identification.

Further, many of the existing deep learning models only output a point estimate and lack the ability of providing error/confidence intervals in their outputs. We addressed this issue by

making minor modifications to our algorithm which allowed it to learn the joint distribution of input $\boldsymbol{x}$ and output $\boldsymbol{y}$. By the virtue of learning this joint density, we are now able to perform classical supervised learning tasks such as image classification, inpainting, etc. with an added quantified uncertainty estimates. In most cases, superior results are observed compared to existing similar Bayesian deep learning methods which rely on learning joint density of image ($\boldsymbol{x}$) and label ($\boldsymbol{y}$). We also demonstrated how learning this joint density could be handy for the simultaneous solution of forward and inverse UQ problems with application to subsurface flow modeling problems.

In the case of deterministic inverse problems, we focused on elasticity imaging application and implemented an adjoint-based optimization routine to infer the visco-elastic properties of tissue from time-harmonic data. This could be useful in the detection, diagnosis and therapeutic monitoring of diseases as tissue mechanical properties (including visco-elastic properties) are tightly correlated to pathological developments. One of the appealing aspects of our proposed method is that it can work even *without* any boundary data making it an attractive choice for Magnetic Resonance Elastography where typically it is difficult to accurately measure boundary data. In order to reduce the total computational time of this inverse problem, we proposed a novel domain decomposition technique the allows the solution of inverse problems in parallel without any message passing between the compute nodes. In order to reduce the total computational cost of elastography, we proposed a deep learning assisted simplified workflow which can perform final classification while circumventing the need of solving a complex and expensive inverse problem.

6.2 Future Work

The work presented in this thesis has opened up numerous avenues for future study.

Applications: From the successful application of our algorithm of using GAN-based priors to solve physics-based inversion to thermal imaging, elasticity imaging, subsurface flow

modeling (Chapter 2 and 3), it is not hard to see that our algorithm is independent of the underlying physical model. As long as we have access to a sufficient number of data samples, our algorithm is able of solving any Bayesian inverse problems. Therefore, we are actively considering applying our algorithm to various inverse problems arising in different domains ranging from medical imaging to physical science. A closely related application involve inverse design or inferring optimal micro/nano-structure of meta-materials for desired quantity of interest. Since GAN provides a differentiable map from parameter to measurements and in these inverse design tasks quantity of interest is typically a known function of measurements, we could have an end-to-end differentiable map from parameter to the quantity of interest and our proposed algorithm could directly be applied there. Similarly, it can also be applied to a topology optimization problem which poses very similar mathematical formulation.

Analyzing complex systems of multiple temporal and length and time scale requires computationally expensive simulations and it is desirable to use computationally inexpensive surrogates for many such applications. As shown in Chapter 3 the model developed in this thesis could act as a surrogate for both forward as well as inverse problems while providing uncertainty estimates. Another important area of such complex physical simulations is to do model order reduction and perform multi-fidelity analysis. The methods developed in this thesis can perform both tasks with minor extension. GAN provides a natural way of dimensionality reduction while ideas from our active learning experiment (Chapter 3) could be extended to do multi-fidelity simulations.

Algorithmic developments: Apart from an application point of view, there are many interesting modeling/algorithmic extensions one can do to the algorithms described in this thesis. A major shortcoming of our proposed algorithm is its massive data requirement, which severely restricts its application to many domains. There are multiple ways one can address this issue. One way is to incorporate known domain knowledge into the GAN model. An approach we are currently actively pursuing for this is by adding an extra penalty term in the loss function. Another way of achieving this is directly by incorporating domain

knowledge in the architecture of the model. Ideas based on equivariant networks could be promising for this.

Another important challenge with the current algorithm is the lack of theoretical guarantees for convergence of posterior statistics and it provides an excellent avenue for further research. The stability of the GAN training dynamics and selection of hyper-parameter also poses a practical challenge and require some careful consideration. We are currently actively investigating different GAN architectures with better stability properties. The use of alternative generative models like Variational Auto-Encoder (VAE) [176] or Normalizing Flows [50] for uncertainty quantification is an interesting and actively pursued research direction by many groups.

In the case of methods developed for deterministic inverse problems, it is currently applicable for time-harmonic cases. Extending it to the transient case is an interesting and practically important research direction. A possible approach for solving it is by posing the time-dependent inverse problem in the frequency domain and solving it at multiple frequencies separately.

References

1. Gouveia, W. P. & Scales, J. A. Resolution of seismic waveform inversion: Bayes versus Occam. *Inverse Problems* **13,** 323–349. doi:10.1088/0266-5611/13/2/009 (1997).
2. Malinverno, A. Parsimonious Bayesian Markov chain Monte Carlo inversion in a nonlinear geophysical problem. *Geophysical Journal International* **151,** 675–688. doi:10.1046/j.1365-246X.2002.01847.x (2002).
3. Martin, J., Wilcox, L. C., Burstedde, C. & Ghattas, O. A Stochastic Newton MCMC Method for Large-Scale Statistical Inverse Problems with Application to Seismic Inversion. *SIAM Journal on Scientific Computing* **34,** A1460–A1487. doi:10.1137/110845598 (2012).
4. Isaac, T., Petra, N., Stadler, G. & Ghattas, O. Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale problems, with application to flow of the Antarctic ice sheet. *Journal of Computational Physics* **296,** 348–368. doi:10.1016/J.JCP.2015.04.047 (2015).
5. Jackson, C., Sen, M. K., Stoffa, P. L., Jackson, C., Sen, M. K. & Stoffa, P. L. An Efficient Stochastic Bayesian Approach to Optimal Parameter and Uncertainty Estimation for Climate Model Predictions. *Journal of Climate* **17,** 2828–2841. doi:10.1175/1520-0442(2004)017<2828:AESBAT>2.0.CO;2 (2004).
6. Yang, B., Qian, Y., Lin, G., Leung, R. & Zhang, Y. Some issues in uncertainty quantification and parameter tuning: a case study of convective parameterization scheme in the WRF regional climate model. *Atmospheric Chemistry and Physics* **12,** 2409–2427. doi:10.5194/acp-12-2409-2012 (2012).
7. Loredo, T. J. in *Maximum Entropy and Bayesian Methods* 81–142 (Springer Netherlands, Dordrecht, 1990). doi:10.1007/978-94-009-0683-9_6.
8. Asensio Ramos, A., Martínez González, M. J. & Rubiño-Martín, J. A. Bayesian inversion of Stokes profiles. *Astronomy & Astrophysics* **476,** 959–970. doi:10.1051/0004-6361:20078107 (2007).
9. Wang, J. & Zabaras, N. Hierarchical Bayesian models for inverse problems in heat conduction. *Inverse Problems* **21,** 183–206. doi:10.1088/0266-5611/21/1/012 (2004).
10. Beck, J. V. Nonlinear estimation applied to the nonlinear inverse heat conduction problem. *International Journal of Heat and Mass Transfer* **13,** 703–716. doi:https://doi.org/10.1016/0017-9310(70)90044-X (1970).
11. Natterer, F. & Wang, G. The Mathematics of Computerized Tomography. *Medical Physics* **29,** 107–108. doi:10.1118/1.1429631 (2002).
12. Hiriyannaiah, H. P. X-ray computed tomography for medical imaging. *IEEE Signal Processing Magazine* **14,** 42–59. doi:10.1109/79.581370 (1997).

13. *SIAM Style Manual: For journals and books* 2013.
14. Sabin, T. J., Bailer-Jones, C. A. L. & Withers, P. J. Accelerated learning using Gaussian process models to predict static recrystallization in an Al-Mg alloy. *Modelling and Simulation in Materials Science and Engineering* **8,** 687–706. doi:10.1088/0965-0393/8/5/304 (2000).
15. Vito, E. D., Rosasco, L., Caponnetto, A., Giovannini, U. D. & Odone, F. Learning from Examples as an Inverse Problem. *Journal of Machine Learning Research* **6,** 883–904 (2005).
16. Siltanen, S., Kolehmainen, V., J rvenp, S., Kaipio, J. P., Koistinen, P., Lassas, M., *et al.* Statistical inversion for medical x-ray tomography with few radiographs: I. General theory. *Physics in Medicine and Biology* **48,** 1437–1463. doi:10.1088/0031-9155/48/10/314 (2003).
17. Kolehmainen, V., Vanne, A., Siltanen, S., Jarvenpaa, S., Kaipio, J., Lassas, M., *et al.* Parallelized Bayesian inversion for three-dimensional dental X-ray imaging. *IEEE Transactions on Medical Imaging* **25,** 218–228. doi:10.1109/TMI.2005.862662 (2006).
18. Keller, J. B. Inverse Problems. *The American Mathematical Monthly* **83,** 107–118 (1976).
19. HADAMARD, J. Sur les Problemes Aux Derivees Partielles et Leur Signification Physique. *Princeton university bulletin,* 49–52 (1902).
20. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems,* 1–9. doi:http://dx.doi.org/10.1016/j.protcy.2014.09.007 (2012).
21. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521,** 436–444. doi:10.1038/nature14539 (2015).
22. Sutskever, I., Vinyals, O. & Le, Q. V. *Sequence to Sequence Learning with Neural Networks* in *In Proc. Advances in Neural Information Processing Systems 27* (2014), 3104–3112.
23. Patel, D., Tibrewala, R., Vega, A., Dong, L., Hugenberg, N. & Oberai, A. A. Circumventing the solution of inverse problems in mechanics through deep learning: Application to elasticity imaging. *Computer Methods in Applied Mechanics and Engineering* **353,** 448–466 (2019).
24. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., *et al.* Mastering the game of Go without human knowledge. *Nature* **550,** 354–359. doi:10.1038/nature24270 (2017).
25. Baldi, P., Sadowski, P. & Whiteson, D. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications* **5,** 1–9. doi:10.1038/ncomms5308 (2014).
26. Goh, G. B., Hodas, N. O. & Vishnu, A. Deep learning for computational chemistry. *Journal of Computational Chemistry* **38,** 1291–1307. doi:10.1002/jcc.24764 (2017).
27. Patel, D. V., Bonam, R. & Oberai, A. A. Deep learning-based detection, classification, and localization of defects in semiconductor processes. *Journal of Micro/Nanolithography, MEMS, and MOEMS* **19,** 1. doi:10.1117/1.jmm.19.2.024801 (2020).
28. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., *et al.* A survey on deep learning in medical image analysis. *Medical Image Analysis* **42,** 60–88. doi:10.1016/J.MEDIA.2017.07.005 (2017).

29. Mousavi, A., Patel, A. B. & Baraniuk, R. A deep learning approach to structured signal recovery. *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton),* 1336–1343 (2015).
30. Jin, K. H., McCann, M. T., Froustey, E. & Unser, M. Deep Convolutional Neural Network for Inverse Problems in Imaging. *IEEE Transactions on Image Processing* **26,** 4509–4522. doi:10.1109/TIP.2017.2713099 (2017).
31. Ye, J. C. & Han, Y. Deep Convolutional Framelets: A General Deep Learning for Inverse Problems. *ArXiv* **abs/1707.00372** (2017).
32. Diamond, S., Sitzmann, V., Heide, F. & Wetzstein, G. Unrolled Optimization with Deep Priors. *ArXiv* **abs/1705.08041** (2017).
33. Adler, J. & Öktem, O. Learned Primal-Dual Reconstruction. *IEEE Transactions on Medical Imaging* **37,** 1322–1332 (2018).
34. Gilton, D., Ongie, G. & Willett, R. Neumann Networks for Linear Inverse Problems in Imaging. *IEEE Transactions on Computational Imaging* **6,** 328–343 (2020).
35. Venkatakrishnan, S. V., Bouman, C. A. & Wohlberg, B. *Plug-and-Play priors for model based reconstruction* in *2013 IEEE Global Conference on Signal and Information Processing* (2013), 945–948. doi:10.1109/GlobalSIP.2013.6737048.
36. Romano, Y., Elad, M. & Milanfar, P. The little Engine that Could: Regularization by Denoising (RED). *ArXiv* (2016).
37. Metzler, C. A., Mousavi, A. & Baraniuk, R. G. *Learned D-AMP: Principled Neural Network Based Compressive Image Recovery* in *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Curran Associates Inc., Long Beach, California, USA, 2017), 1770–1781.
38. Bora, A., Jalal, A., Price, E. & Dimakis, A. G. *Compressed sensing using generative models* in *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (2017), 537–546.
39. Metzler, C. A., Mousavi, A., Heckel, R. & Baraniuk, R. Unsupervised Learning with Stein's Unbiased Risk Estimator. *ArXiv* **abs/1805.10531** (2018).
40. Soltanayev, S. & Chun, S. *Training Deep Learning based Denoisers without Ground Truth Data* in *NeurIPS* (2018).
41. Eldar, Y. C. Generalized SURE for Exponential Families: Applications to Regularization. *IEEE Transactions on Signal Processing* **57,** 471–481 (2009).
42. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., *et al. Noise2Noise: Learning Image Restoration without Clean Data* in *Proceedings of the 35th International Conference on Machine Learning* (eds Dy, J. & Krause, A.) **80** (PMLR, Stockholmsmässan, Stockholm Sweden, 2018), 2965–2974.
43. Bora, A., Price, E. & Dimakis, A. G. AmbientGAN: Generative models from lossy measurements. *ICLR* **2,** 5 (2018).
44. Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks* in *Computer Vision (ICCV), 2017 IEEE International Conference on* (2017).
45. Armanious, K., Jiang, C., Abdulatif, S., Küstner, T., Gatidis, S. & Yang, B. Unsupervised Medical Image Translation Using Cycle-MedGAN. *2019 27th European Signal Processing Conference (EUSIPCO),* 1–5 (2019).

46. Quan, T. M., Nguyen-Duc, T. & Jeong, W. .-K. Compressed Sensing MRI Reconstruction Using a Generative Adversarial Network With a Cyclic Loss. *IEEE Transactions on Medical Imaging* **37,** 1488–1497. doi:10.1109/TMI.2018.2820120 (2018).
47. Kingma, D. P. & Welling, M. Auto-Encoding Variational Bayes. *CoRR* **abs/1312.6114** (2014).
48. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., *et al. Generative adversarial nets* in *Advances in neural information processing systems* (2014), 2672–2680.
49. Rezende, D. & Mohamed, S. *Variational Inference with Normalizing Flows* in *Proceedings of the 32nd International Conference on Machine Learning* (eds Bach, F. & Blei, D.) **37** (PMLR, Lille, France, 2015), 1530–1538.
50. Dinh, L., Krueger, D. & Bengio, Y. *NICE: Non-linear independent components estimation* in *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings* (International Conference on Learning Representations, ICLR, 2015).
51. Hand, P., Leong, O. & Voroninski, V. *Phase Retrieval Under a Generative Prior* in *Advances in Neural Information Processing Systems* (eds Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N. & Garnett, R.) **31** (Curran Associates, Inc., 2018).
52. Mosser, L., Dubrule, O. & Blunt, M. J. Stochastic Seismic Waveform Inversion Using Generative Adversarial Networks as a Geological Prior. *Mathematical Geosciences* **52,** 53–79. doi:10.1007/s11004-019-09832-6 (2020).
53. Asim, M., Shamshad, F. & Ahmed, A. Blind Image Deconvolution Using Deep Generative Priors. *IEEE Transactions on Computational Imaging* **6,** 1493–1506. doi:10.1109/TCI.2020.3032671 (2020).
54. Seidl, D. T., Oberai, A. A. & Barbone, P. E. The Coupled Adjoint-State Equation in forward and inverse linear elasticity: Incompressible plane stress. *Computer Methods in Applied Mechanics and Engineering* **357,** 112588. doi:https://doi.org/10.1016/j.cma.2019.112588 (2019).
55. Stuart, A. M. Inverse problems: A Bayesian perspective. *Acta Numerica* **19,** 451–559. doi:10.1017/S0962492910000061 (2010).
56. Oberai, A. A., Gokhale, N. H. & Feijóo, G. R. Solution of inverse problems in elasticity imaging using the adjoint method. *Inverse Problems* **19,** 297–313. doi:10.1088/0266-5611/19/2/304 (2003).
57. Nowozin, S., Cseke, B. & Tomioka, R. *f-gan: Training generative neural samplers using variational divergence minimization* in *Advances in neural information processing systems* (2016), 271–279.
58. Arjovsky, M., Chintala, S. & Bottou, L. Wasserstein GAN (2017).
59. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. & Courville, A. C. *Improved training of wasserstein gans* in *Advances in neural information processing systems* (2017), 5767–5777.
60. Villani, C. *Optimal transport: old and new* (Springer Science & Business Media, 2008).
61. Brooks, S., Gelman, A., Jones, G., Meng, X.-L. & Neal, R. M. *MCMC using Hamiltonian dynamics* tech. rep. (2012).

62. Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., *et al.* TensorFlow Distributions (2017).
63. Andrieu, C. & Thoms, J. A tutorial on adaptive MCMC. *Statistics and Computing* **18,** 343–373. doi:10.1007/s11222-008-9110-y (2008).
64. Iglesias, M., Lin, K. & Stuart, A. Well-posed Bayesian geometric inverse problems arising in subsurface flow. *Inverse Problems* **30,** 114001 (2014).
65. Kaipio, J. & Somersalo, E. *Statistical and computational inverse problems* (Springer Science & Business Media, 2006).
66. LeCun, Y. & Cortes, C. MNIST handwritten digit database (2010).
67. Barbone, P. & Oberai, A. *A Review of the Mathematical and Computational Foundations of Biomechanical Imaging* in (2010).
68. Pavan, T. Z., Madsen, E. L., Frank, G. R., Jiang, J., Carneiro, A. A. & Hall, T. J. A nonlinear elasticity phantom containing spherical inclusions. *Physics in Medicine and Biology* **57,** 4787–4804. doi:10.1088/0031-9155/57/15/4787 (2012).
69. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., *et al.* Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **542,** 115–118. doi:10.1038/nature21056 (2017).
70. Hannun, A. Y., Rajpurkar, P., Haghpanahi, M., Tison, G. H., Bourn, C., Turakhia, M. P., *et al.* Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature Medicine* **25,** 65 (2019).
71. Gent, P. R., Danabasoglu, G., Donner, L. J., Holland, M. M., Hunke, E. C., Jayne, S. R., *et al.* The Community Climate System Model Version 4. *Journal of Climate* **24,** 4973–4991. doi:10.1175/2011JCLI4083.1 (01 Oct. 2011).
72. Schneider, T., Lan, S., Stuart, A. & Teixeira, J. Earth System Modeling 2.0: A Blueprint for Models That Learn From Observations and Targeted High-Resolution Simulations. *Geophysical Research Letters* **44,** 12, 396–12, 417. doi:10.1002/2017GL076101 (2017).
73. Grigorescu, S., Trasnea, B., Cocias, T. & Macesanu, G. A Survey of Deep Learning Techniques for Autonomous Driving. *Journal of Field Robotics* **37,** 362–386. doi:10.1002/rob.21918 (2019).
74. Rausch, V., Hansen, A., Solowjow, E., Liu, C., Kreuzer, E. & Hedrick, J. K. *Learning a deep neural net policy for end-to-end control of autonomous vehicles* in *Proceedings of the American Control Conference* (Institute of Electrical and Electronics Engineers Inc., 2017), 4914–4919. doi:10.23919/ACC.2017.7963716.
75. Heaton, J. B., Polson, N. G. & Witte, J. H. Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry* **33,** 3–12. doi:10.1002/asmb.2209 (2017).
76. De Spiegeleer, J., Madan, D. B., Reyners, S. & Schoutens, W. Machine learning for quantitative finance: fast derivative pricing, hedging and fitting. *Quantitative Finance* **18,** 1635–1643. doi:10.1080/14697688.2018.1495335 (2018).
77. Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., *et al.* End to End Learning for Self-Driving Cars. *ArXiv* **abs/1604.07316** (2016).
78. Gal, Y. *Uncertainty in deep learning* PhD thesis (PhD thesis, University of Cambridge, 2016).

79. DeGroot, M. H. *et al.* Uncertainty, information, and sequential experiments. *The Annals of Mathematical Statistics* **33,** 404–419 (1962).
80. Tishby, N., Levin, E. & Solla, S. A. *Consistent inference of probabilities in layered networks: Predictions and generalization* in *IJCNN Int Jt Conf Neural Network* (Publ by IEEE, 1989), 403–409. doi:10.1109/ijcnn.1989.118274.
81. MacKay, D. J. A practical Bayesian framework for backpropagation networks. *Neural computation* **4,** 448–472 (1992).
82. Neal, R. M. *Bayesian Learning via Stochastic Dynamics* in *Advances in Neural Information Processing Systems 5, [NIPS Conference]* (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992), 475–482.
83. Welling, M. & Teh, Y. W. *Bayesian Learning via Stochastic Gradient Langevin Dynamics* in *Proceedings of the 28th International Conference on International Conference on Machine Learning* (Omnipress, Bellevue, Washington, USA, 2011), 681–688.
84. Korattikara, A., Rathod, V., Murphy, K. & Welling, M. *Bayesian Dark Knowledge* in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2* (MIT Press, Montreal, Canada, 2015), 3438–3446.
85. Hinton, G. E. & van Camp, D. *Keeping the Neural Networks Simple by Minimizing the Description Length of the Weights* in *Proceedings of the Sixth Annual Conference on Computational Learning Theory* (Association for Computing Machinery, Santa Cruz, California, USA, 1993), 5–13. doi:10.1145/168304.168306.
86. Barber, D. G. & Bishop, C. M. *Ensemble learning in Bayesian neural networks* in (1998).
87. Graves, A. *Practical Variational Inference for Neural Networks* in *Proceedings of the 24th International Conference on Neural Information Processing Systems* (Curran Associates Inc., Granada, Spain, 2011), 2348–2356.
88. Blundell, C., Cornebise, J., Kavukcuoglu, K. & Wierstra, D. *Weight Uncertainty in Neural Networks* in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37* (JMLR.org, Lille, France, 2015), 1613–1622.
89. Louizos, C. & Welling, M. *Structured and Efficient Variational Deep Learning with Matrix Gaussian Posteriors* in *Proceedings of The 33rd International Conference on Machine Learning* (eds Balcan, M. F. & Weinberger, K. Q.) **48** (PMLR, New York, New York, USA, 2016), 1708–1716.
90. Louizos, C. & Welling, M. *Multiplicative Normalizing Flows for Variational Bayesian Neural Networks* in *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (JMLR.org, Sydney, NSW, Australia, 2017), 2218–2227.
91. Gal, Y. & Ghahramani, Z. *Dropout as a bayesian approximation: Representing model uncertainty in deep learning* in *international conference on machine learning* (2016), 1050–1059.
92. Kendall, A. & Gal, Y. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? (2017).
93. Huang, P.-Y., Hsu, W.-T., Chiu, C.-Y., Wu, T.-F. & Sun, M. *Efficient Uncertainty Estimation for Semantic Segmentation in Videos* in *European Conference on Computer Vision (ECCV)* (2018).

94. Lakshminarayanan, B., Pritzel, A. & Blundell, C. *Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles* in *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Curran Associates Inc., Long Beach, California, USA, 2017), 6405–6416.
95. Riquelme, C., Tucker, G. & Snoek, J. Deep Bayesian Bandits Showdown: An Empirical Comparison of Bayesian Deep Networks for Thompson Sampling. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings* (2018).
96. Alemi, A. A., Fischer, I. S. & Dillon, J. V. Uncertainty in the Variational Information Bottleneck. *ArXiv* **abs/1807.00906** (2018).
97. Behrmann, J., Grathwohl, W., Chen, R. T. Q., Duvenaud, D. & Jacobsen, J.-H. *Invertible Residual Networks* in *Proceedings of the 36th International Conference on Machine Learning* (eds Chaudhuri, K. & Salakhutdinov, R.) **97** (PMLR, Long Beach, California, USA, 2019), 573–582.
98. Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D. & Lakshminarayanan, B. Hybrid Models with Deep and Invertible Features. *36th International Conference on Machine Learning, ICML 2019* **2019-June,** 8295–8304 (2019).
99. Chen, R. T. Q., Behrmann, J., Duvenaud, D. & Jacobsen, J. Residual Flows for Invertible Generative Modeling (2019).
100. Shah, V. & Hegde, C. *Solving linear inverse problems using gan priors: An algorithm with provable guarantees* in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), 4609–4613.
101. Yang, L., Zhang, D. & Karniadakis, G. E. Physics-Informed Generative Adversarial Networks for Stochastic Differential Equations. *SIAM J. Scientific Computing* **42,** A292–A317 (2020).
102. Ardizzone, L., Kruse, J., Rother, C. & Köthe, U. *Analyzing Inverse Problems with Invertible Neural Networks* in *International Conference on Learning Representations* (2019).
103. Adler, J. & Öktem, O. Deep bayesian inversion. *arXiv preprint arXiv:1811.05910* (2018).
104. Arridge, S., Maass, P., Öktem, O. & Schönlieb, C.-B. Solving inverse problems using data-driven models. *Acta Numerica* **28,** 1–174 (2019).
105. Kovachki, N., Baptista, R., Hosseini, B. & Marzouk, Y. Conditional Sampling With Monotone GANs. *arXiv preprint arXiv:2006.06755* (2020).
106. Belghazi, M., Oquab, M. & Lopez-Paz, D. *Learning about an exponential amount of conditional distributions* in *Advances in Neural Information Processing Systems* (2019), 13703–13714.
107. Lindgren, E. M., Whang, J. & Dimakis, A. G. Conditional sampling from invertible generative models with applications to inverse problems. *arXiv preprint arXiv:2002.11743* (2020).
108. Winkler, C., Worrall, D., Hoogeboom, E. & Welling, M. *Learning Likelihoods with Conditional Normalizing Flows* 2020.
109. Vauhkonen, M., Kaipio, J. P., Somersalo, E. & Karjalainen, P. A. Electrical impedance tomography with basis constraints. *Inverse Problems* **13,** 523–530. doi:10.1088/0266-5611/13/2/020 (1997).

110. Calvetti, D. & Somersalo, E. Priorconditioners for linear systems. *Inverse Problems* **21,** 1397–1418 (2005).
111. Martin, R. & Walker, S. G. Data-driven priors and their posterior concentration rates. *Electronic Journal of Statistics* **13,** 3049–3081. doi:10.1214/19-EJS1600 (2019).
112. Arora, S., Risteski, A. & Zhang, Y. *Do GANs learn the distribution? some theory and empirics* in *International Conference on Learning Representations* (2018).
113. Uppal, A., Singh, S. & Póczos, B. *Nonparametric density estimation & convergence rates for gans under besov ipm losses* in *Advances in Neural Information Processing Systems* (2019), 9089–9100.
114. Narayanan, H. & Mitter, S. *Sample Complexity of Testing the Manifold Hypothesis* in *Advances in Neural Information Processing Systems* (eds Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R. & Culotta, A.) **23** (Curran Associates, Inc., 2010), 1786–1794.
115. Sagan, H. *Space-filling curves* (Springer Science & Business Media, 2012).
116. Owhadi, H. & Scovel, C. Qualitative robustness in Bayesian inference. *ESAIM: Probability and Statistics* **21,** 251–274 (2017).
117. Owhadi, H., Scovel, C. & Sullivan, T. On the brittleness of Bayesian inference. *SIAM Review* **57,** 566–582 (2015).
118. Kahn, G., Villaflor, A., Pong, V., Abbeel, P. & Levine, S. Uncertainty-Aware Reinforcement Learning for Collision Avoidance (2017).
119. Lakshminarayanan, B., Pritzel, A. & Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *Advances in Neural Information Processing Systems* **2017-December,** 6403–6414 (2016).
120. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J. & Mané, D. Concrete Problems in AI Safety (2016).
121. LeCun, Y., Cortes, C. & Burges, C. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* **2** (2010).
122. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X. & Huang, T. S. *Generative Image Inpainting With Contextual Attention* in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).
123. Yeh, R. A., Chen, C., Lim, T. Y., Schwing, A. G., Hasegawa-Johnson, M. & Do, M. N. *Semantic Image Inpainting with Deep Generative Models* in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 6882–6890.
124. Yan, Z., Li, X., Li, M., Zuo, W. & Shan, S. Shift-Net: Image Inpainting via Deep Feature Rearrangement. *ArXiv* **abs/1801.09392** (2018).
125. Kendall, A., Badrinarayanan, V. & Cipolla, R. *Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding* in (British Machine Vision Association and Society for Pattern Recognition, 2019). doi:10.5244/c.31.57.
126. Kohl, S. A. A., Romera-Paredes, B., Meyer, C., De Fauw, J., Ledsam, J. R., Maier-Hein, K. H., *et al.* A Probabilistic U-Net for Segmentation of Ambiguous Images (2018).
127. Liu, Z., Luo, P., Wang, X. & Tang, X. *Deep Learning Face Attributes in the Wild* in *Proceedings of International Conference on Computer Vision (ICCV)* (2015).

128. Cacuci, D. *Sensitivity and Uncertainty Analysis, Volume I: Theory* doi:10.1201/9780203498798 (2003).
129. Saltelli, A., Chan, K. & Scott, E. *Sensitivity Analysis* (Wiley, 2009).
130. Dashti, M. & Stuart, A. M. The Bayesian approach to inverse problems. *Handbook of Uncertainty Quantification,* 1–118 (2016).
131. Rojas, S. & Koplik, J. Nonlinear flow in porous media. *Phys. Rev. E* **58,** 4776–4782. doi:10.1103/PhysRevE.58.4776 (4 1998).
132. Zhu, Y., Zabaras, N., Koutsourelakis, P.-S. & Perdikaris, P. Physics-Constrained Deep Learning for High-dimensional Surrogate Modeling and Uncertainty Quantification without Labeled Data. *Journal of Computational Physics* **394,** 56–81. doi:https://doi.org/10.1016/j.jcp.2019.05.024 (2019).
133. Alnaes, M. S., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., *et al.* The FEniCS Project Version 1.5. *Archive of Numerical Software* **3.** doi:10.11588/ans.2015.100.20553 (2015).
134. Barr, R. G. Real-time ultrasound elasticity of the breast: Initial clinical results. *Ultrasound Quarterly* **26,** 61–66. doi:10.1097/RUQ.0b013e3181dc7ce4 (2010).
135. Berg, W. A., Gutierrez, L., NessAiver, M. S., Carter, W. B., Bhargavan, M., Lewis, R. S., *et al.* Diagnostic accuracy of mammography, clinical examination, US, and MR imaging in preoperative assessment of breast cancer. *Radiology* **233,** 830–849. doi:10.1148/radiol.2333031484 (2004).
136. Chang, J. M., Moon, W. K., Cho, N., Yi, A., Koo, H. R., Han, W., *et al.* Clinical application of shear wave elastography (SWE) in the diagnosis of benign and malignant breast diseases. *Breast Cancer Research and Treatment* **129,** 89–97. doi:10.1007/s10549-011-1627-7 (2011).
137. Palmeri, M. L., Wang, M. H., Rouze, N. C., Abdelmalek, M. F., Guy, C. D., Moser, B., *et al.* Noninvasive evaluation of hepatic fibrosis using acoustic radiation force-based shear stiffness in patients with nonalcoholic fatty liver disease. *Journal of Hepatology* **55,** 666–672. doi:10.1016/j.jhep.2010.12.019 (2011).
138. Oliphant, T. E., Manduca, A., Ehman, R. L. & Greenleaf, J. F. Complex-valued stiffness reconstruction for magnetic resonance elastography by algebraic inversion of the differential equation. *Magnetic Resonance in Medicine* **45,** 299–310. doi:10.1002/1522-2594(200102)45:2<299::AID-MRM1039>3.0.CO;2-O (2001).
139. Park, E. & Maniatty, A. M. Shear modulus reconstruction in dynamic elastography: Time harmonic case. *Physics in Medicine and Biology* **51,** 3697–3721. doi:10.1088/0031-9155/51/15/007 (2006).
140. Ophir, J., Cespedes, I., Ponnekanti, H., Yazdi, Y. & Li, X. Elastography - A Quantitative Method for Imaging the Elasticity of Biological Tissues. *Ultrasonic Imaging* **13,** 111–134 (1991).
141. Parker, K., Doyley, M. & Rubens, D. Imaging the elastic properties of tissue: the 20 year perspective. *Physics in medicine and biology* **56,** R1 (2011).
142. Greenleaf, J., Fatemi, M. & Insana, M. SELECTED METHODS FOR IMAGING ELASTIC PROPERTIES OF BIOLOGICAL TISSUES. *Annual Reviews in Biomedical Engineering* **5,** 57–78 (2003).
143. Kennedy, B. F., Wijesinghe, P. & Sampson, D. D. The emergence of optical elastography in biomedicine. *Nature Photonics* **11,** 215 (2017).

144. Kallel, F. & Bertrand, M. Tissue elasticity reconstruction using linear perturbation method. *IEEE Transactions on Medical Imaging* **15(3),** 299–313 (1996).
145. Barbone, P. & Oberai, A. A Review of the Mathematical and Computational Foundations of Biomechanical Imaging. *Computational Modeling in Biomechanics,* 375–408 (2010).
146. Barbone, P. E., Oberai, A. A., Bamber, J. C., Berry, G. P., Dord, J.-F., Ferreira, E. R., *et al.* in *Handbook of Imaging in Biological Mechanics* 199–215 (CRC Press, 2014).
147. Dong, L., Wijesinghe, P., Dantuono, J. T., Sampson, D. D., Munro, P. R., Kennedy, B. F., *et al.* Quantitative Compression Optical Coherence Elastography as an Inverse Elasticity Problem. *IEEE Journal of Selected Topics in Quantum Electronics* **22,** 1–11 (2016).
148. Goenezen, S., Dord, J., Sink, Z., Barbone, P., Jiang, J., Hall, T., *et al.* Linear and Nonlinear Elastic Modulus Imaging: An Application to Breast Cancer Diagnosis. *Medical Imaging, IEEE Transactions on* **31,** 1628–1637 (2012).
149. Berg, W. A., Cosgrove, D. O., Doré, C. J., Schäfer, F. K., Svensson, W. E., Hooley, R. J., *et al.* Shear-wave elastography improves the specificity of breast US: the BE1 multinational study of 939 masses. *Radiology* **262,** 435–449 (2012).
150. Liu, T., Babaniyi, O. A., Hall, T. J., Barbone, P. E. & Oberai, A. A. Noninvasive In-Vivo Quantification of Mechanical Heterogeneity of Invasive Breast Carcinomas. *PloS one* **10** (2015).
151. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *nature* **521,** 436 (2015).
152. Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y. *Deep learning* (MIT press Cambridge, 2016).
153. Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39,** 1137–1149. doi:10.1109/TPAMI.2016.2577031 (2017).
154. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition (2015).
155. Haber, E., Ruthotto, L., Holtham, E. & Jun, S.-H. Learning across scales-A multiscale method for Convolution Neural Networks. *arXiv preprint arXiv:1703.02009* (2017).
156. Konofagou, E. & Ophir, J. A new elastographic method for estimation and imaging of lateral displacements, lateral strains, corrected axial strains and Poisson?s ratios in tissues. *Ultrasound in medicine & biology* **24,** 1183–1199 (1998).
157. Pan, S. J. & Yang, Q. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* **22,** 1345–1359. doi:10.1109/TKDE.2009.191 (2010).
158. Torrey, L. & Shavlik, J. *Transfer learning. Handbook of research on machine learning applications and trends : algorithms, methods and techniques* (Information Science Reference, 2010).
159. Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. *How transferable are features in deep neural networks?* in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (MIT Press, 2014), 3320–3328.
160. Chatfield, K., Simonyan, K., Vedaldi, A. & Zisserman, A. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531* (2014).
161. Perez, L. & Wang, J. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621* (2017).

162. Oberai, A., Gokhale, N. & Feijoo, G. Solution of Inverse Problems in Elasticity Imaging Using the Adjoint Method. *Inverse Problems* **19,** 297–313 (2003).
163. Oberai, A., Gokhale, N., Doyley, M. & Bamber, J. Evaluation of the Adjoint Equation Based Algorithm for Elasticity Imaging. *Physics in Medicine and Biology* **49,** 2955–2974 (2004).
164. Oberai, A., Gokhale, N., Goenezen, S., Barbone, P., Hall, T., Sommer, A., *et al.* Linear and nonlinear elasticity imaging of soft tissue in vivo: demonstration of feasibility. *Physics in Medicine and Biology* **54,** 1191–1207 (2009).
165. Wellman, P., Howe, R., Dalton, E. & Kern, K. *Breast Tissue Stiffness in Compression is Correlated to Histological Diagnosis* tech. rep. (Harvard BioRobotics Laboratory, Division of Engineering and Applied Sciences, Harvard University, 1999).
166. Blatz, P., Chu, B. & Wayland, H. On the mechanical behavior of elastic animal tissue. *Journal of Rheology* **13,** 83 (1969).
167. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization (2014).
168. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., *et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems (2016).
169. Peng, X. B., Andrychowicz, M., Zaremba, W. & Abbeel, P. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization (2017).
170. Yoo, J., Hong, Y., Noh, Y. & Yoon, S. Domain Adaptation Using Adversarial Learning for Autonomous Navigation (2017).
171. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W. & Abbeel, P. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World (2017).
172. Biros, G. & Ghattas, O. Parallel Lagrange–Newton–Krylov–Schur Methods for PDE-Constrained Optimization. Part II: The Lagrange–Newton Solver and Its Application to Optimal Control of Steady Viscous Flows. *SIAM Journal on Scientific Computing* **27,** 714 (2005).
173. Bui-Thanh, T., Willcox, K. & Ghattas, O. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing* **30,** 3270–3288 (2008).
174. Goenezen, S., Barbone, P. & Oberai, A. A. Solution of the nonlinear elasticity imaging inverse problem: The incompressible case. *Computer methods in applied mechanics and engineering* **200,** 1406–1420 (2011).
175. Goenezen, S., Sink, Z., Oberai, A. A., Barbone, P. E., Dord, J. F., Jiang, J., *et al. Breast cancer diagnosis using nonlinear elasticity imaging: some initial results* in *Proceeding of the 9th International Conference on the Ultrasonic Measurement and Imaging of Tissue Elasticity* (Snowbird, Utah, 2010).
176. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).