Solving Forward and Inverse Problems for Seismic Imaging using Invertible Neural Networks

Naveen Gupta

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Anuj Karpatne, Chair
Youzuo Lin
Adrian Sandu

May 9, 2023
Blacksburg, Virginia

Keywords: Full Waveform Inversion, Inverse Problems, Invertible Neural Networks, AI for Science, Machine Learning

Solving Forward and Inverse Problems for Seismic Imaging using Invertible Neural Networks

Naveen Gupta

(ABSTRACT)

Full Waveform Inversion (FWI) is a widely used optimization technique for subsurface imaging where the goal is to estimate the seismic wave velocity beneath the Earth's surface from the observed seismic data at the surface. The problem is primarily governed by the wave equation, which is a non-linear second-order partial differential equation. A number of approaches have been developed for FWI including physics-based iterative numerical solvers as well as data-driven machine learning (ML) methods. Existing numerical solutions to FWI suffer from three major challenges: (1) sensitivity to initial velocity guess (2) non-convex loss landscape, and (3) sensitivity to noise. Additionally, they suffer from high computational cost, making them infeasible to apply in complex real-world applications. Existing ML solutions for FWI only solve for the inverse and are prone to yield non-unique solutions. In this work, we propose to solve both forward and inverse problems jointly to alleviate the issue of non-unique solutions for an inverse problem. We study the FWI problem from a new perspective and propose a novel approach based on Invertible Neural Networks. This type of neural network is designed to learn bijective mappings between the input and target distributions and hence they present a potential solution to solve forward and inverse problems jointly. In this thesis, we developed a data-driven framework that can be used to learn forward and inverse mappings between any arbitrary input and output space. Our model, Invertible X-net, can be used to solve FWI to obtain high-quality velocity images and also

predict the seismic waveforms data. We compare our model with the existing baseline models and show that our model outperforms them in velocity reconstruction on the OpenFWI dataset. Additionally, we also compare the predicted waveforms with a baseline and ground truth and show that our model is capable of predicting highly accurate seismic waveforms simultaneously.

Solving Forward and Inverse Problems for Seismic Imaging using Invertible Neural Networks

Naveen Gupta

(GENERAL AUDIENCE ABSTRACT)

Recent advancements in deep learning have led to the development of sophisticated methods that can be used to solve scientific problems in many disciplines including medical imaging, geophysics, and signal processing. For example, in geophysics, we study the internal structure of the Earth from indirect physical measurements. Often, these kind of problems are challenging due to existence of non-unique and unstable solutions. In this thesis, we look at one such problem called Full Waveform Inversion which aims to estimate velocity of mechanical wave inside the Earth from wave amplitude observations on the surface. For this problem, we explore a special class of neural networks that allows to uniquely map the input and output space and thus alleviate the non-uniqueness and instability in performing Full Waveform Inversion for seismic imaging.

Dedication

Dedicated to my loving parents, siblings, and my lovely niece Krishu for their constant support, inspiration, and encouragement.

Acknowledgments

First, I would like to express my deepest gratitude to my advisor Dr. Anuj Karpatne for his unwavering support, guidance, and immense motivation for this dissertation. He provided me with the freedom to explore topic of my interest and provided all the needful resources to become a better researcher. Despite my background in geophysics, he believed in me and encouraged me to learn computer science through my interests and research.

Next, I would like to beholden my sincere thanks to Dr. Youzuo Lin, Scientist at Los Alamos National Laboratory, for sharing his valuable guidance through the course of this research. His insights during our biweekly meetings were quite helpful and interesting. Since the beginning of our collaboration, he has been encouraging and providing resources for shaping this research in right direction.

I would like to especially thank to Arka Daw and Jie Bu for sparing their valuable time and mentoring me through this journey. They are always willing to listen, provide guidance and discuss new ideas which made this thesis possible.

I would also like to thank the group members of Knowledge Guided Machine Learning at Virginia Tech. The weekly group meetings and exchange of ideas have been a huge factor in my learning.

My two years at Blacksburg would not be possible without the support from friends around. They have helped me through many obstacles and shared cherished, long-lasting memories.

Last but not the least, I would also like to give my special thank to my parents, elder brothers, and family for their constant support during my masters. I cannot thank them enough for encouraging me to pursue this goal two years back during the pandemic. This would not have been possible without their confidence and steadfast determination.

Contents

List of Figures

List of Tables

List of Abbreviations

CNN Convolutional Neural Network

FWI Full Waveform Inversion

GAN Generative Adversarial Network

INN Invertible Neural Network

MAE Mean Absolute Error

ML Machine Learning

MSE Mean Square Error

SSIM Structural Similarity Index Measure

Chapter 1

Introduction

In this chapter, we start our discussion with the inverse problem in section 1.1. We further discuss the problem of seismic imaging, the motivation behind the full waveform inversion (FWI) problem, and its importance in imaging internal structures of the Earth in section 1.2. We conclude the chapter by describing the contribution of this thesis in section 1.3.

1.1 Background on Inverse Problems

Inverse problems are a class of mathematical problems that involve making inference about an unknown cause or input from observed outputs. This type of problem is encountered in various fields, including medical imaging, geophysics, and signal processing. For most inverse problems, the forward problem is to predict an observed output when inputs are known. Of note, a problem is said to be well-posed only if it satisfies all three mathematical conditions: (1) a solution must exists for the problem, (2) the solution must be unique, and (3) the solution must also be stable. A solution is said to be stable if small perturbations in inputs result into small variation in the output. In contrast to forward problems, inverse problems are mostly ill-posed due to the fact that it may violate one or more of the three strict conditions of a well-posed problem.

Due to the ill-posed nature of an inverse problem, it remains challenging to solve in many fields. In geophysics, there is a need to study the internal structure of the Earth without

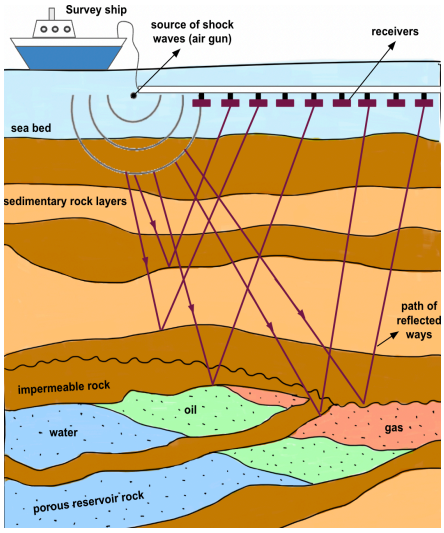
direct observations. A prime example comes from energy exploration where we are interested in finding oil and gas prospects beneath Earth surface. Throughout the history, invasive drilling based on geological observations was a common technique, but it is highly expensive that could only be employed when the likelihood of success is high.

A variety of techniques have been proposed to solve subsurface imaging problem including seismic, gravity, geomagnetism, and electrical resistivity. Nonetheless, the inverse problem is prevalent across all these methods as the forward problem solves for predicting observable outputs when the subsurface structure and the governing physics equations are known. In this thesis, we focus on solving the inverse problem for seismic waves, as this method has been proven to be effective in delineating highly detailed images of the subsurface.

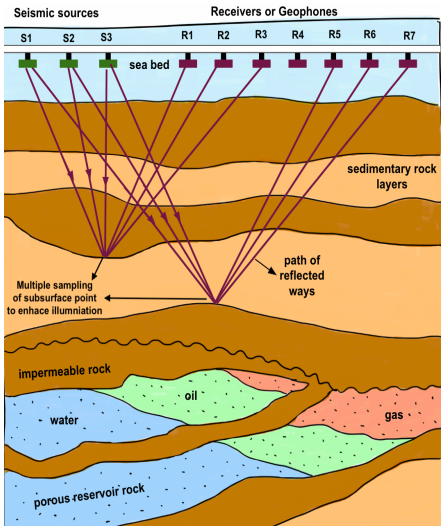
1.2 Seismic Imaging

Seismic waves are mechanical waves generated by an artificial source that sample the mechanical properties of rocks inside the Earth. As described in Figure 1.1a, we have a source that generates seismic waves that traverse inside the Earth and are recorded back at the surface by an array of receivers. To enhance the subsurface illumination, it is a common practice to sample same point multiple times with different source and receiver configurations (see Figure 1.1b). The propagation of seismic waves inside the Earth is well-understood through a physics based model, also known as forward model, which can predict the waveform response of a subsurface setting when its velocity distribution is known.

In contrast, Full Waveform Inversion (FWI) is a method that estimates velocity distribution given an observed seismic waveform response. FWI is an ill-posed inverse optimization problem governed by the second-order partial differential equation i.e. wave equation and due to its ill-posed nature, many seismic waveforms can be mapped to a velocity configuration.



(a) Cartoon image depicting the seismic reflection survey for subsurface imaging



(b) Cartoon image depicting sampling of a subsurface point by different source-receiver configuration.

Figure 1.1: Schematic figures demonstrating the propagation of seismic wave inside the Earth.

Similar to other inverse problems, iterative optimization algorithms can be used to solve FWI. However, the iterative evaluation of the non-linear wave equation for fitting seismic waveforms present three major challenges: (1) sensitivity to initial velocity guess (2) non-convex loss landscape, and (3) sensitivity to noise [26].

To address the challenges in existing solutions for performing FWI, we explore a special class of neural networks called Invertible Neural Networks (INN) that make computation of analytical inverse feasible. Originally, INNs were designed for image generative tasks and have some unique properties for efficiently calculating Jacobians. Inspired by this idea, we adopted INNs for performing FWI and designed a novel architecture for learning both forward and inverse problems using a data-driven framework. A major reason behind solving forward and inverse problems jointly using INNs is to alleviate the issue of non-unique solutions for an inverse problem. Moreover, the training of our architecture does not suffer

from vanishing or exploding gradient issues and therefore the prediction yields a stable solution. In other words, our proposed method helps us to alleviate the two major challenges in solving the inverse problem for performing FWI: (1) non-unique solutions, and (2) unstable solutions. To the best of our knowledge, we are the first to solve forward and inverse FWI problems jointly and predicting both seismic velocity and waveforms without any external supervision. Our approach outperforms existing baselines for velocity prediction on the OpenFWI dataset and also predicts seismic waveforms under multiple source and variable velocity conditions with high accuracy.

1.3 Thesis Contributions

The primary goal of this thesis is to develop a novel deep learning approach for Full Waveform Inversion (FWI) problem by solving both the forward and inverse problems jointly. In Chapter 2, we review relevant literature and deep learning based methods for FWI. In Chapter 3, we give a background on the Invertible Neural Network (INN) which serves as a foundation for the development of our proposed method. In Chapter 4, we review the U-net like architectures for domain-to-domain translation problem and then we present our own contribution that is the proposed architecture to solve FWI. Finally, in Chapter 5, we discuss about the OpenFWI dataset and present our results on different geological settings with comparisons to baseline models.

Chapter 2

Literature Review

In this chapter, we present a comprehensive overview of existing literature on Full Waveform Inversion, including various methods and their shortcomings. The ultimate objective is to emphasize the effectiveness of deep learning methods in solving this problem. In section 2.1, we discuss conventional techniques that are primarily governed by the physics of the problem. We then provide an overview of data-driven methods in section 2.2, with a particular focus on deep learning methods that have been proposed for estimating seismic velocity. Finally, we discuss physics integrated data-driven methods in section 2.3 to improve the accuracy and reliability of results.

2.1 Physics Based Full Waveform Inversion

Physics-based methods utilize the fundamental principles and governing equations to estimate the subsurface properties. Seismic waves are essentially mechanical perturbations that propagate through a medium at a speed determined by the acoustic impedance of the material. The acoustic wave equation, which describes the behavior of the waves in the time domain, is expressed as:

$$\nabla^2 p(r,t) - \frac{1}{v^2(r)} \frac{\partial^2}{\partial t^2} p(r,t) = s(r,t) \tag{2.1}$$

where r defines the spatial domain in (x, z) coordinate space (x is the horizontal direction and z is the depth from the surface), t denotes time, $v(r)$ is the velocity of the wave at the point r and time t in space, ∇^2 is the Laplacian operator, $s(r, t)$, describes the source input signal as a function space and time, and $p(r, t)$ refers to the pressure wavefield, also known as wave amplitude.

The above equation can be viewed as a deterministic relationship between amplitude and velocity in the following form, also known as forward model:

$$d = f(v) \tag{2.2}$$

where d represents waveform response recorded at the sensors or receivers at the surface and v repesents velocity distribution of the wave inside the Earth.

The goal of the inversion problem here is to invert velocity distribution of wave from the given amplitude response and this could be formulated as minimization of the following loss function:

$$E(m) = min_v\{||d - f(v)||_2^2 + \lambda R(v)\} \tag{2.3}$$

where R is the regularization term with parameter λ and $||d - f(v)||_2^2$ is the content or data loss term with $||.||_2$ representing the l_2 norm.

Traditionally, equation (2.3) has been typically solved as an iterative optimization problem, as shown in Figure 2.1 and the gradient of the loss function is computed iteratively using Gradient Descent or L-BFGS algorithm to update the velocity model until the loss is converged [17]. Nonetheless, these methods are sensitive to initial velocity guess and are known to suffer from local-minima problem due to cycle skipping [13]. Cycle skipping occurs when

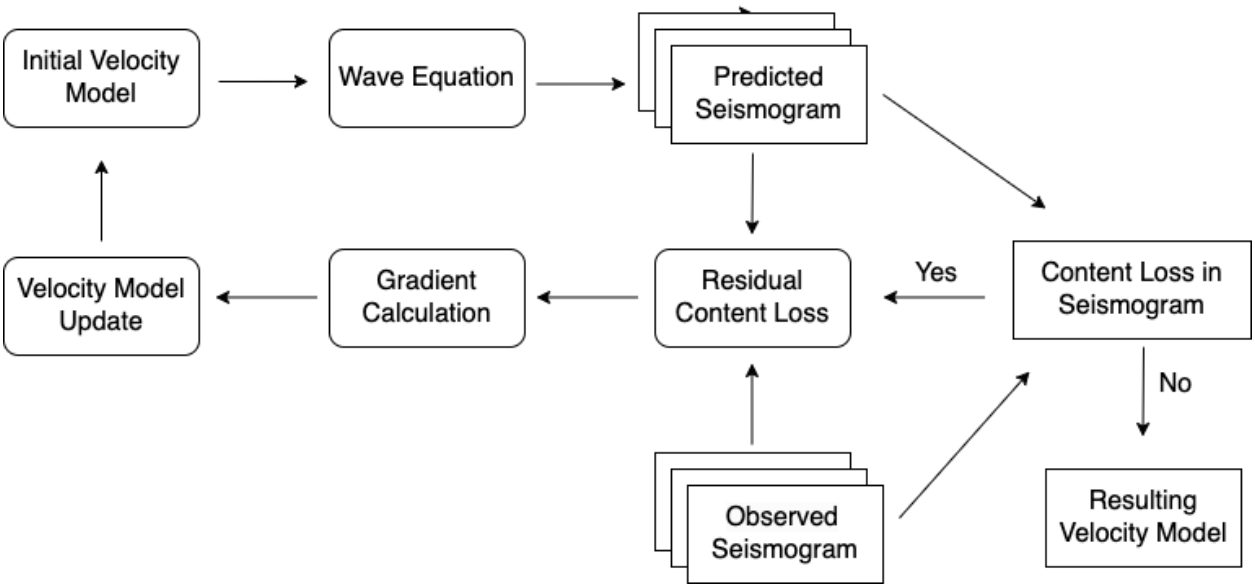


Figure 2.1: Iterative Full Waveform Inversion workflow.

the phase difference between the predicted and observed waveforms is greater than half a wavelength, resulting in a sudden jump of the misfit function. The jump creates a local minima in the objective function, making it difficult to find global minimum. To overcome the challenge, researchers have proposed different schemes to obtain a good initial velocity guess based on data components (tomography), adding regularization schemes, and human interpretation [1, 21, 28, 31].

Two regularization schemes that have been commonly used with content loss to alleviate problems with gradient computations: are (1) L2 regularization and (2) Total Variation (TV) regularization. L2 regularization, also known as Tikhonov regularization [7], penalizes the loss proportional to the square of weights, and TV regularization [11], typically penalizes the loss proportional to the total variation of the solution. While the L2 regularization prevents overfitting and penalizes roughness, the TV regularization encourages piecewise smoother solutions. Yang *et al.* and Engquist *et al.* [5, 27] showed the applicability of optimal transport or Earth mover's distance in solving FWI and its advantages in preventing cycle

skipping issue, avoiding local minima, and robustness to noise.

Nonetheless, the challenges in solving FWI in time-domain requires large number of forward simulations of wave equation and this increases the computational cost for large-scale velocity configurations. Additionally, time-domain inversion exhibits more sensitivity to high-frequency content, leading to the smoothing of small-scale features and decreased resolution.

2.2 Data Driven Approaches for Full Waveform Inversion

In seismic community, data-driven methods have emerged as a powerful tool that has been applied to a variety of tasks such as fault detection [22, 24], seismic data interpolation, and interpretation [18, 19]. The recent availability of large-scale labeled seismic datasets has further facilitated the adoption of supervised deep learning approaches, which have demonstrated comparable performance to traditional techniques with faster processing times. These frameworks possess the ability to learn intricate non-linear relationships between input data and target output, making them particularly effective in solving complex inversion problems.

The key idea behind the data-driven approach is to train a neural network to capture a non-linear complex relationship between the input and ground truth. In context of FWI, Convolutional Neural Network (CNN) have been successful in mapping non-linear seismic data to velocity model. This could also be treated as an image-to-image translation problem and CNNs have been widely used in computer vision community to solve this kind of problem.

The problem can be mathematically formulated as follows:

$$\begin{aligned} \hat{\boldsymbol{v}} &= g_{\theta^*}(\boldsymbol{d}) \\ \text{s.t.} \quad \boldsymbol{\theta^*} &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{(m_i,d_i)} L(g_\theta(\boldsymbol{d}), \boldsymbol{v}) \end{aligned} \tag{2.4}$$

where $g(.)$ represents neural network parameterized with the weights θ^* to be trained on paired sample of (v_i, d_i).

2.2.1 InversionNet

Wu *et al.* [23] proposed an encoder-decoder like architecture that is trained on the OpenFWI dataset [2] and showed that their approach is able to learn the complex mapping between seismic waveforms and velocity. The encoder is used to extract high level features from the waveforms into a latent vector which is later translated to a velocity field using a decoder. The encoder network consists of 8 convolutional block and each block consists of two convolution sub-blocks, batch normalization, and ReLU activation. Similarly, the decoder constitutes 5 transposed convolution (deconvolution) blocks and 6 convolution blocks. Each deconvolution block is followed by an convolution block with LeakyReLU non-linearity except the last convolutional block which uses tanh activation. They showed that their encoder-decoder model is able to handle different geological complexity when provided with enough training examples and predict realistic velocity with faster inferences to new instances.

2.2.2 VelocityGAN

Inspired by the success of Generative Adversarial Networks (GAN), Zhang *et al.* [29] proposed another data-driven approach to pose seismic inversion as a min-max optimization problem. In this model, the generator is an encoder-decoder model and is guided by the

CNN as the discriminator. The input to the generator model is seismic waveforms, the generator outputs fake velocity labels, and the role of discriminator is to distinguish fake and true velocity labels. To stabilize the GAN training and alleviate mode collapse issues, they have used Wasserstein loss with gradient penalty, asymmetric loss weighting for generator and discriminator, and asymmetric training paradigm as well. For every 5 training gradient descent steps for discriminator, generator is updated once. They showed their framework is able to generate clearer and sharper velocity labels compared to fully convolutional networks [16, 20].

2.2.3 Shortcomings

Unlike physics-based approaches, these methods train a neural network to directly map the seismic amplitude to velocity space. This eliminates the need for iterative evaluations of wave equations, thereby reducing the computational cost of the solution. In [20, 23, 25], the authors have used different CNN architecture to predict velocity models from the seismic amplitude data.

However, it is important to note that the data-driven approach requires a large representative training dataset and therefore purely data-driven techniques are unlikely to generalize or adapt well to a new unseen velocity configuration because there is not enough data [12]. Additionally, deep learning networks, such as CNNs, are often considered black-box models, meaning that it is difficult to understand how the network arrives at its predictions. In the context of FWI, the lack of interpretability can be problematic, as it is essential to understand the underlying geological structures accurately.

2.3 Incorporating Physics Guidance in Data Driven Full Waveform Inversion

Fundamentally, seismic inversion or FWI is a physics problem, and purely data-driven techniques have limited capability in capturing the complete underlying system characteristics [12]. Hence, integrating physics knowledge with data-driven techniques can provide a new approach to enhance the generalizability and explainability of the results. In this section, we would discuss several methods that incorporate physics knowledge with available data to address data paucity issues in seismic inversion.

2.3.1 Neural-Network-Based Full Waveform Inversion (NNFWI)

In contrast to data-driven method that treat FWI as a supervised problem, physics guided methods do not use ground truth or velocity labels and treat is as an unsupervised problem. The most common approach to incorporate physics supervision is to integrate the governing wave equation in the training loop and it has been shown to perform better and more robust to noise. The residual term from the wave equation helps to regularize the solution space and therefore improves the generalizability of the model.

Further, the addition of priori knowledge about velocity distribution also helps the model to constraints the solution space. He *et al.* and Ren *et al.* [9, 14] showed that a good initial velocity guess, as a priori knowledge, that captures low frequency content of the true velocity distribution, can be useful in circumventing local minima during training.

Zhu *et al.* [30] proposed neural network based full waveform inversion (NNFWI) that utilizes prior knwloedge about velocity distribution in conjunction with the wave equation to re-parameterize the neural network for seismic inversion as shown in Figure 2.2

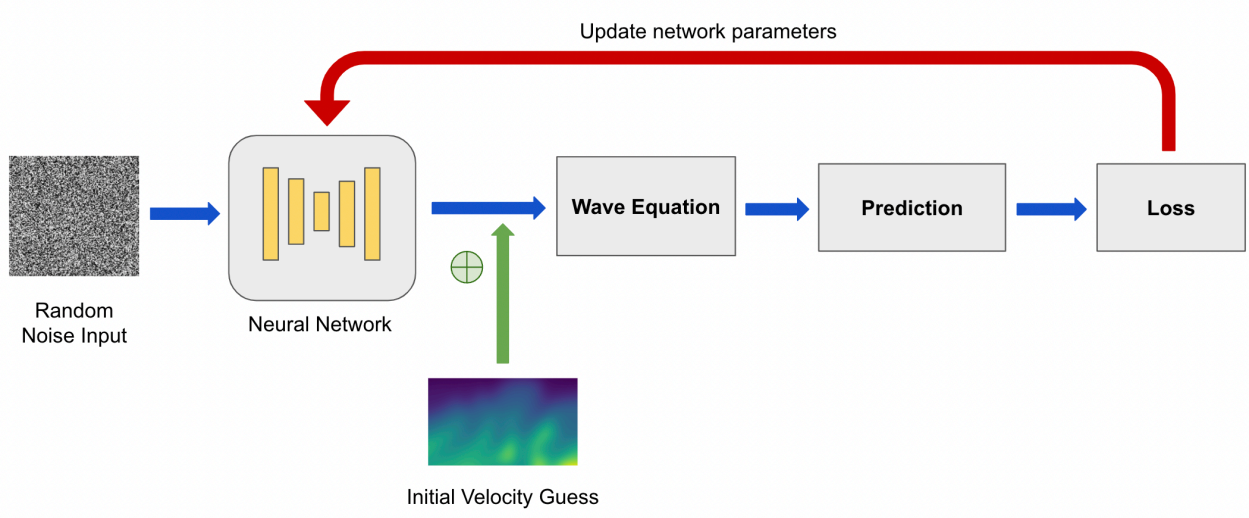


Figure 2.2: Neural-Network-Based Full Waveform Inversion (NNFWI) architecture.

The authors of NNFWI have formulated the FWI problem as :

$$\begin{aligned} \hat{\boldsymbol{v}} &= g_{\theta^*(d)}(\boldsymbol{\gamma}) \\ \text{s.t.} \quad \boldsymbol{\theta^*(d)} &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum L(f(g_\theta(\boldsymbol{\gamma})), \boldsymbol{d}) \end{aligned} \tag{2.5}$$

where $g(.)$ represents neural network with the weights θ^*, $f(.)$ represents wave equation, γ represents an arbitrary random vector.

The regularization effect of a neural network structure in FWI has been studied by exploring different network architectures [9, 14]. In NNFWI, a good initial velocity guess is required, and the network is first pre-trained to map a noise vector to the initial velocity guess. This enables the network to learn the mapping from noise to velocity distribution. Subsequently, the output of the network is fed as an input to the wave equation to predict amplitude, and the gradient of the loss in the amplitude domain is utilized to update network parameters until convergence is reached. In the inference step, noise is the input to the network, and the output is the predicted seismic velocity distribution.

2.3.2 Full Waveform Inversion using Wasserstein GAN

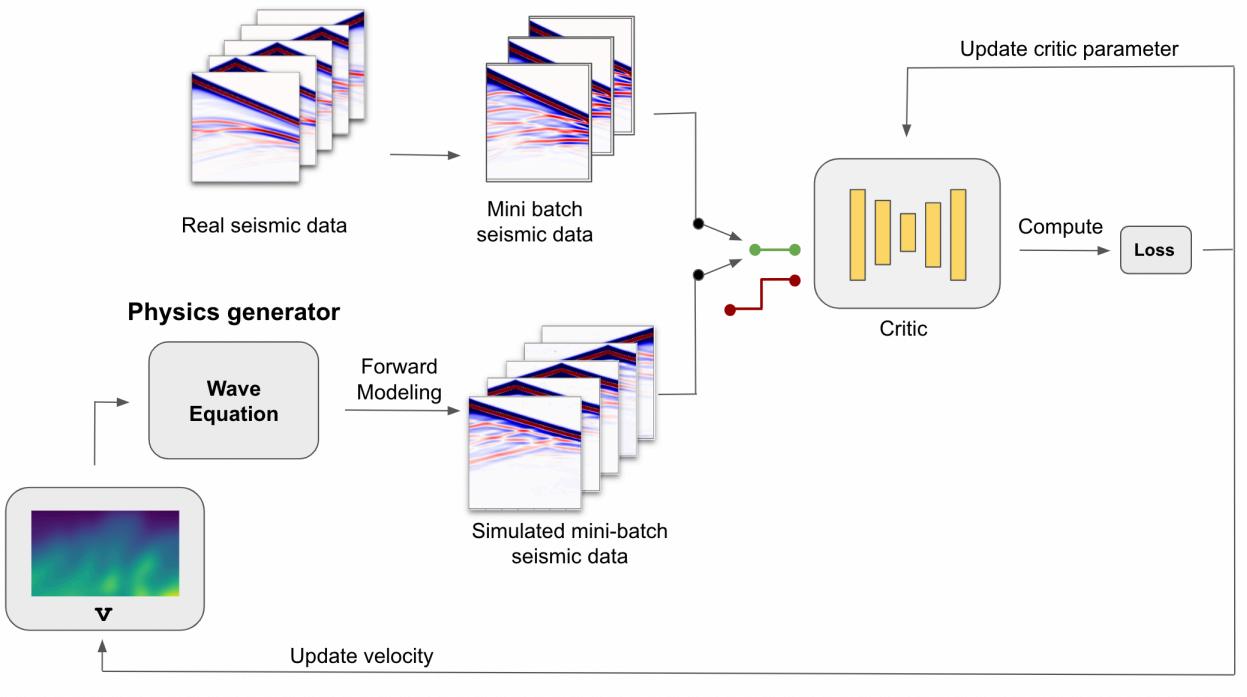


Figure 2.3: Wasserstein-GAN FWI architecture.

Another approach by Yang *et al.* [26] showed the applicability of the Generative Adversarial Networks (GAN) model in the context of seismic inversion. The authors introduced an unsupervised physics-guided Wasserstein GAN framework to invert seismic velocity using the min-max learning strategy. In this model, the wave equation is used in place of the generator and the discriminator is an arbitary neural network. In contrast to the vanilla GAN, the discriminator in wasserstein GAN act as a critic that measures the dissimilarity between the real and fake distribution as shown in Figure 2.3. The objective function optimizes the Wasserstein-1 distance, which is also called the Earth-Mover distance, and showed that this framework is more robust to noise and independent of the choice of initial velocity guess. The framework optimizes the following cost function described in equation 2.6 for velocity estimation.

$$\hat{v} = \arg\min_{v} \max_{D_\theta} \mathbb{E}_{d\sim P_{true}}[D_\theta(P(d))] - \mathbb{E}_{u\sim P\text{rec}}[D_\theta(P(u))] - \lambda \mathbb{E}_{\hat{u}\sim P_{\text{int}}}[(||\nabla_{\hat{u}} D_\theta(\hat{u})||_2 - 1)^2] \quad (2.6)$$

where P_{true} is the real data distribution, P_{rec} denote fake data distribution, P_{int} is the data uniformly sampled from the real P_{true} and fake P_{rec} distributions, D_θ represents discriminator neural network parameterized with θ.

To alleviate the challenges associated with GAN models, the discriminator should be 1-Lipschitz continuous for stable training. This can be achieved by clipping weights of discriminator in a confined space, however, it fails to converge in some cases. In this case, the authors used a gradient penalty strategy proposed by [8] to stabilize the training process and showed the generalizability of their framework for seismic inversion.

2.3.3 Unsupervised Physics-informed Full-Waveform Inversion

While these physics-based models provide better generalization, they are also computationally expensive as the entire architecture is trained from scratch for a new instance. Therfore, Jin *et al.* [10] proposed an Unsupervised Physics-informed Full Waveform Inversion (UP-FWI) architecture, described in Figure 2.4, which is trained in batch-mode on unlabelled waveform data on the OpenFWI dataset [2]. There are two advantages for this architecture (1) the architecture is physics informed and (2) the training requires less data with better generalization and robustness than purely data-driven techniques.

Mathematically, the problem can be posed as:

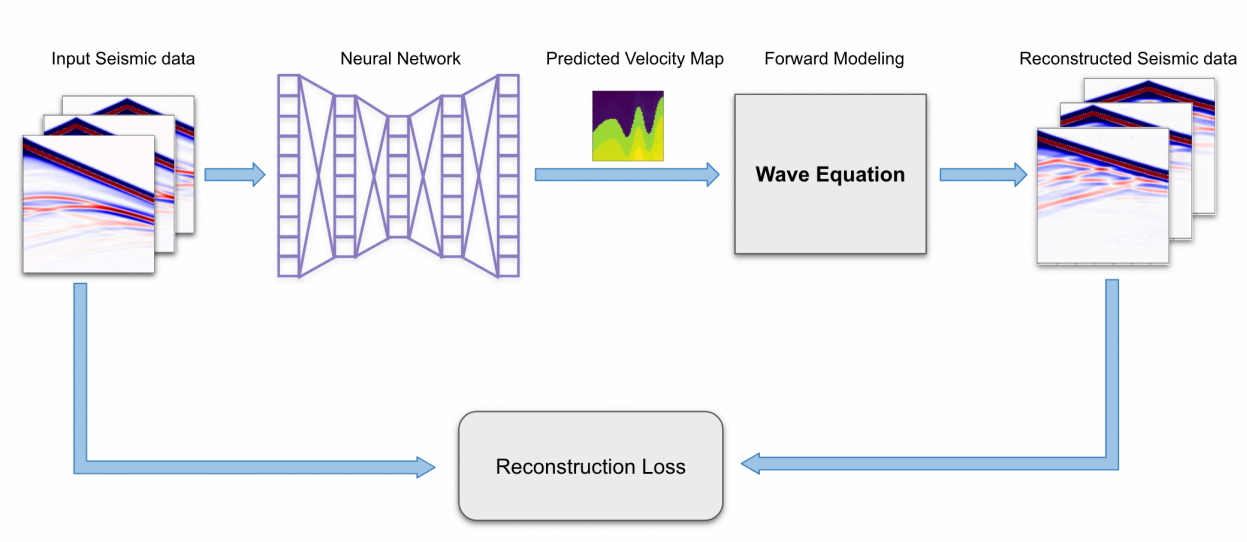


Figure 2.4: UPFWI architecture.

$$\begin{aligned} \hat{\boldsymbol{v}} &= g_{\theta^*}(\boldsymbol{d}) \\ \text{s.t.} \quad \boldsymbol{\theta^*(d)} &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum L(f(g_\theta(\boldsymbol{d_i})), \boldsymbol{d_i}) \end{aligned} \tag{2.7}$$

where d_i is the seismic waveform data, $g(.)$ is the neural network parameterized with θ_*, $L(.)$ denotes the loss function, $f(.)$ is the forward wave equation in the architecture.

The reconstruction loss for the UPFWI architecture is content loss and perceptual loss as described in equation 2.8.

$$\begin{aligned} L(d, \hat{d}) &= L_{pixel}(d, \hat{d}) + L_{perceptual}(d, \hat{d}) \\ L_{pixel}(d, \hat{d}) &= \lambda_1 L_1(d, \hat{d}) + \lambda_2 L_2(d, \hat{d}) \\ L_{perceptual}(d, \hat{d}) &= \lambda_3 L_1(f(d), f(\hat{d})) + \lambda_4 L_2(f(d), f(\hat{d})) \end{aligned} \tag{2.8}$$

where $L(.)$ is the loss function between original d and the predicted $\hat{d}$ waveform, L_{pixel} is the weighted L1 (L_1) and L2 loss (L_2) on waveforms, $L_{perceptual}$ is the weighted L1 and L2 loss between the perceptual features of original and predicted waveforms, and $f(.)$ is a pretrained

network to extract visual features like VGG-16 and ResNet.

2.3.4 Shortcomings

Although, the physics knowledge with data-driven techniques provides robust solution, it is non-trivial to incorporate physics guidance with deep neural network architecture for three main reasons: (1) these methods often requires multiple evaluations of the forward physics model, which can add significant computational overhead, (2) the magnitude of gradients backpropagated through physics model are very small and therefore it either requires some hand-crafted training schemes for gradient scaling for faster convergence or large number of epochs before loss converges, and (3) the training parameters are also sensitive to hyperparameter selection due to non-convex nature of physics loss and may need exhaustive hyperparameter search for network training.

Chapter 3

Background of Invertible Neural Networks

In this chapter, we present a brief background on the Invertible Neural Networks (INNs). Firstly, in section 3.1, we provide our motivation behind using INNs for FWI problem. Moving forward, in section 3.2, we introduce INNs and provide a high level glance on the invertibility. Lastly, in section 3.3, we delve into the mathematical formulation of INNs, providing a clear understanding of the concepts and techniques involved.

3.1 Motivation

In recent years, Invertible Neural Networks (INNs) have gained considerable attention due to their unique properties of being able to invert their own computations and preserve the input data. Additionally, the formulation of INNs provides an efficient way to compute Jacobian and guarantees the computation of exact log-likelihood, facilitating the learning of a one-to-one mapping between input and target distributions. As discussed in Chapter 1, FWI is a powerful technique for seismic imaging when the observed waveforms are known, however, it suffers from the issue of being an ill-posed problem, where the forward problem is well-defined but the inverse problem yields non-unique solutions. Given the non-unique nature of FWI, INNs present a promising solution as the inverse computation is analytically

feasible and the forward and inverse computations can be uniquely mapped.

3.2 Introduction

An Invertible Neural Network (INN) is a generative model that aims to learn a bijective mapping between input and output spaces. Unlike traditional neural networks, which typically learns "many-to-one" mapping, INNs are specifically designed to be fully invertible i.e. learn "one-to-one" mapping between input and target domains.

Given an input data x and a function $f(x)$ that maps x to a transformed space, the forward pass can be expressed as:

$$y = f(x) \tag{3.1}$$

where y is the transformed output and $f(x)$ is the invertible transformations applied to input data x. Similarly, the inverse pass of the INN, which reconstructs the original input from the transformed output, can be represented as:

$$x = g(y) \tag{3.2}$$

where x is the original input and $g(y)$ is the invertible transformations applied to y. A key property of an INN is that both the forward pass and the inverse pass are fully invertible, meaning that the original input x can be exactly reconstructed from the transformed output y and vice versa.

3.3 Mathematical Formulation

The design of invertible neural network is based on change of variable formula as described in equation 3.3.

$$p_X(x) = p_Y(f(x)) \left| \det \frac{\partial f(x)}{\partial x} \right| \tag{3.3}$$

where $p_X(x)$ is probability distribution of input data x, $p_Y(f(x))$ is probability distribution of the transformed output y, and $\left|\det \frac{\partial f(x)}{\partial x}\right|$ is the determinant of the Jacobian matrix of function f at x.

In general, the computation of the Jacobian matrix for high dimensional data is very expensive, however, Dinh *et al.* [3] showed that transformation $f(x)$ can be decomposed into a series of invertible transformations (see Figure 3.1) such that jacobian can be efficiently and accurately computed. The main idea behind building an invertible transformation is to partitioned input data x into two blocks (x_1, x_2) such that the after transformation, the resulting outputs (y_1, y_2) are also invertible.

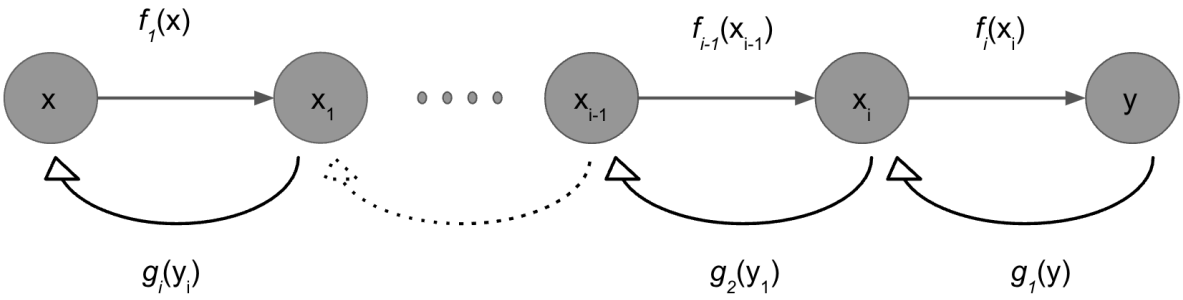


Figure 3.1: Normalizing flows for Invertible Neural Networks.

Dinh *et al.* [4] proposed simple bijective affine coupling block that can be stacked sequentially

to map any arbitrary complex distribution. Given an input data $x \in \mathbb{R}^D$ and N < D, the output $y \in \mathbb{R}^D$ can be obtained using the equation 3.4 and the original input can be constructed using equation 3.5.

$$\begin{aligned} y_{1:N} &= x_{1:N} \\ y_{N+1:D} &= x_{N+1:D} \odot \exp(s(x_{1:N})) + t(x_{1:N}) \end{aligned} \tag{3.4}$$

$$\begin{aligned} x_{1:N} &= y_{1:N} \\ x_{N+1:D} &= (y_{N+1:D} - t(x_{1:N})) \odot \exp(-s(y_{1:N})) \end{aligned} \tag{3.5}$$

where s and t can be arbitrary non-invertible complex functions or neural networks. The Jacobian of this transformation is given by equation 3.6.

$$\frac{\partial f(x)}{\partial x^T} = \begin{bmatrix} \mathbb{I}_{\mathbb{N}} & 0 \\ \frac{\partial f(x)_{N+1:D}}{\partial x_{1:N}^T} & diag(\exp[s(x_{1:N})]) \end{bmatrix} \tag{3.6}$$

The Jacobian matrix is a lower triangular matrix and therefore its determinant is equal to the product of the diagonal elements, and this could be efficiently computed for high dimensional data as well. This design allows us to compute the inverse like a forward propagation in the opposite direction. Though these coupling blocks are capable of capturing non-linear distribution, some part of forward propagation remains unchanged because the input data is naively copied to the output space (see Equation 3.4). To overcome this challenge, Dinh *et al.* [4] suggested to compose multiple coupling blocks with alternating patterns to utilize the full capability of the invertible networks.

Chapter 4

Invertible Neural Networks for Full Waveform Inversion

In the last chapter, we discussed the architecture of invertible neural networks, however, the design of the coupling block prohibits the computation of analytical inverse when input and output are of different dimensions. Another challenge is that pure coupling blocks cannot be used for designing an encoder-decoder like architecture which is useful for many domain-to-domain translation problems. Therefore, in the present chapter, we first begin by discussing the U-net architecture in section 4.1 and then describe the invertible U-net architecture in section 4.2 with learnable upsampling and downsampling blocks. Finally, we propose our Invertible X-net architecture in section 4.3 for the arbitrary input and output space.

4.1 U-net architecture

U-Net is a popular convolutional neural network architecture that was originally developed for image segmentation tasks [15]. The architecture was specifically designed for biomedical image segmentation tasks, where the goal is to identify and segment specific structures within an image. However, the architecture has since been used for a variety of other applications, including semantic segmentation and object detection.

The U-Net architecture consists of an encoder and a decoder network, with a contracting path

and an expanding path, respectively (see Figure 4.1). The contracting path is responsible for capturing context and reducing the spatial resolution of the input, while the expanding path is responsible for recovering spatial resolution and generating the final segmentation output.

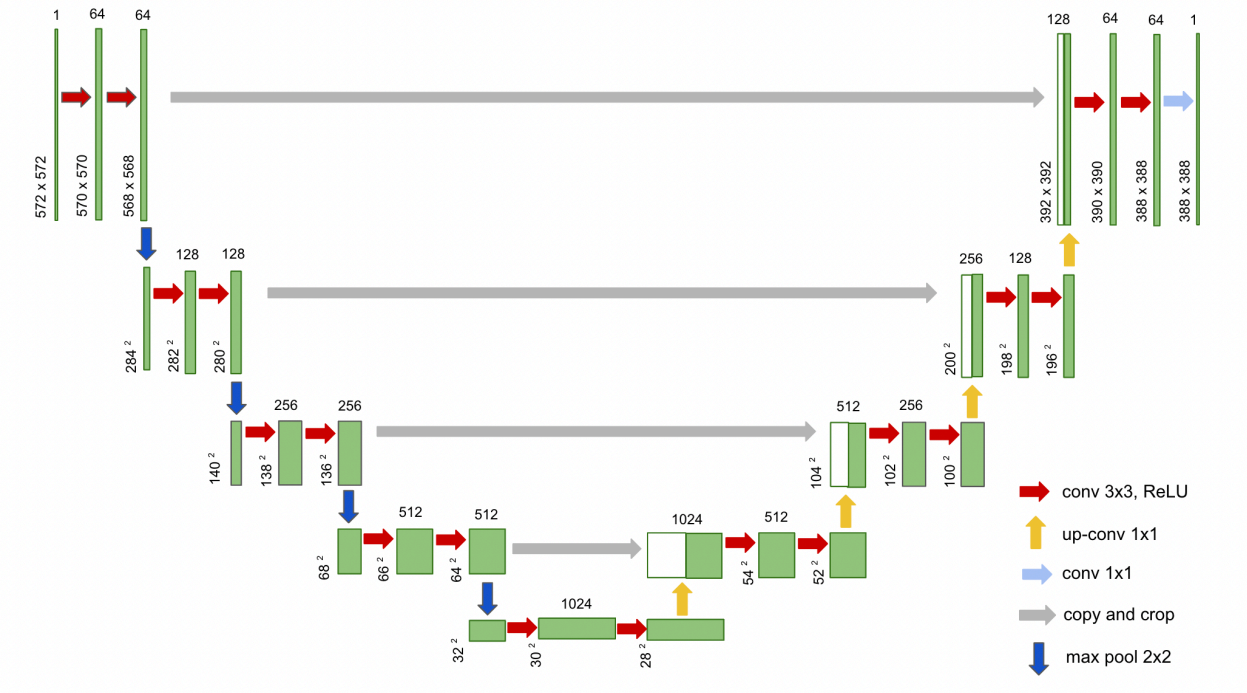


Figure 4.1: U-net architecture.

The contracting path consists of several convolutional and max-pooling layers, which progressively reduce the spatial resolution of the input image. This allows the network to capture context and extract high-level features from the input image, while reducing the number of parameters in the network. The expanding path consists of several upsampling and convolutional layers, which progressively recover the spatial resolution of the input image and generate the final segmentation output.

One of the key design principles of the U-Net architecture is the use of skip connections, which connect the encoder and decoder networks at multiple levels. These skip connections

allow the network to reuse low-level features from the input image, which are important for accurately segmenting small structures and preserving fine details in the output.

While the U-net architecture is quite successful in segmentation task, it has also been applied to domain-to-domain translation problems [23]. In domain translation problem, the primary goal is to map an input data from one domain to a target domain, an encoder-decoder architecture is widely used because of their ability to translate high-level features. Nonetheless, the design of U-net is not invertible i.e. it only learns to translate data from one domain to another and not vice versa.

4.2 Invertible U-net

In a traditional U-Net, the encoding path involves convolutions, max-pooling, and downsampling, while the decoding path involves transposed convolutions, upsampling, and concatenation with the corresponding encoding path feature maps. However, these operations are not invertible, making it difficult to use the U-net for learning bijective mappings.

On the other hand, we saw in the previous chapter that the architecture of invertible neural networks using a coupling block allow us to learn both forward and inverse mapping, given the dimensionality of input and the target must be the same. Therefore, Etmann *et al.* [6] proposed an invertible U-net architecture using the idea of invertible coupling blocks and learnable upsampling and downsampling filters. The general idea behind invertible U-net is to replace convolutional layers with the coupling blocks, max pooling operation with an orthogonal convolutional filter for downsampling, and upsampling operation with an orthogonal deconvolution filter for upsampling.

4.2.1 Invertible Downsampling and Upsampling Operator

The purpose of eliminating the max pooling layer is to avoid a non-invertible operation and downsampling can also be achieved through a convolutional filter with an appropriate stride. In this section, we would discuss designing a learnable orthogonal convolutional and deconvolution filter for downsampling and upsampling an image respectively. The orthogonal convolution is a special case of convolution when the filter size is equal to the stride.

Given an input image $I \in \mathbb{R}^{C \times N_1 \times \ldots \times N_d}$ and stride $s \in \mathbb{R}^{s_1 \times \ldots \times s_d}$ such that the dimensions of the image I is perfectly divisible by the stride s_i and dimensions of downsampling filter $K \in \mathbb{R}^{\tilde{C} \times C \times s_1 \times \ldots \times s_d}$ is equal to the stride, then the convolution of the image with this filter would be:

$$Conv(K, I) : \mathbb{R}^{C \times N_1 \times \ldots \times N_d} \rightarrow \mathbb{R}^{\tilde{C} \times \tilde{N_1} \times \ldots \times \tilde{N_d}} \tag{4.1}$$

Similarly, the upsampling filter $K^* \in \mathbb{R}^{\tilde{C} \times C \times s_1 \times \ldots \times s_d}$ where deconvolution of the input image I with upsampling filter would be:

$$DeConv(K^*, I) : \mathbb{R}^{C \times N_1 \times \ldots \times N_d} \rightarrow \mathbb{R}^{\tilde{C} \times \tilde{N_1} \times \ldots \times \tilde{N_d}} \tag{4.2}$$

where C is the number of channels, $\tilde{C}$ is the number of filters, N_i and $\tilde{N_i}$ is the input image size and output image size along dimension d_i.

Since the downsampling and upsampling are performed using convolutional operator, the weights of filters can be learned during the training. The primary reason behind using orthogonal convolution for downsampling are (1) determinant of the filter is either +1 or -1, which gives some mathematical advantage in computing exact log likelihood like INNs, and

(2) the inverse operation i.e. upsampling can be known simply by computing adjoint of the filter.

4.2.2 Architecture

Similar to the traditional U-net, the architecture of invertible U-net calculates image features on multiple scales by a series of invertible coupling blocks followed by a learnable downsampling filter with an increase in feature dimensions. The invertible coupling blocks are also called resolution preserving layers as they do not alter the dimensionality of the input image. Unlike the traditional U-net, the dimensionality of the input and target domains have to be the same as the architecture is symmetrical (see Figure 4.2).

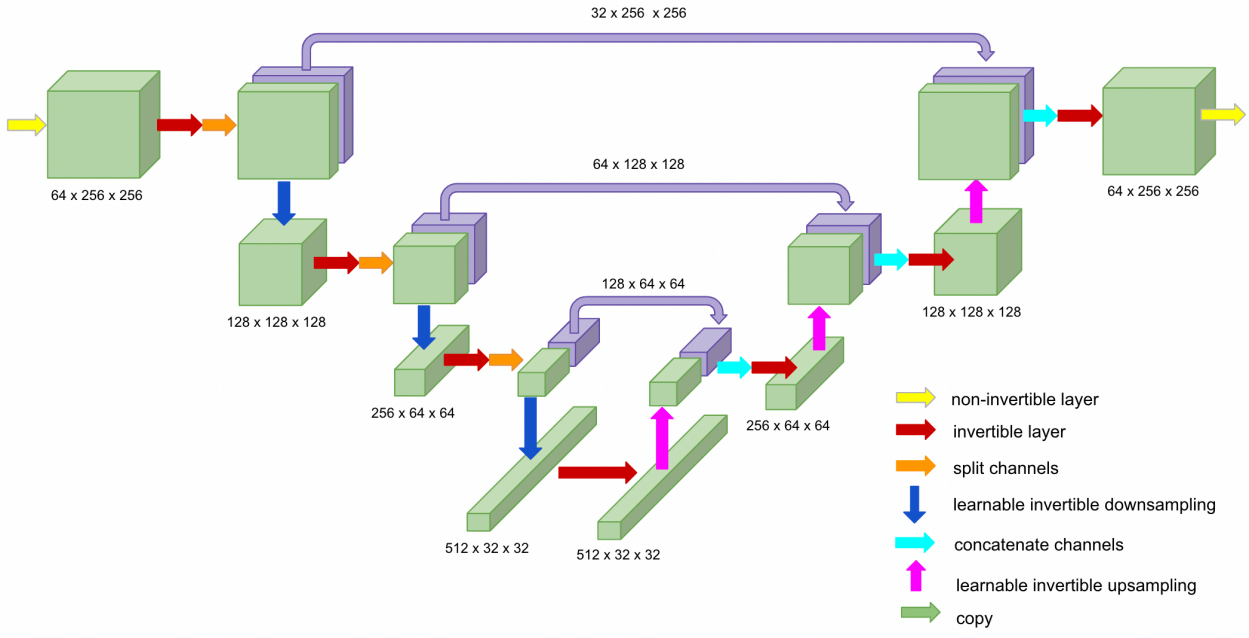


Figure 4.2: Invertible U-net architecture.

Like the traditional U-net, the architecture consists of skip connections from encoder block to decoder block, however, to maintain constant dimensionality across coupling blocks, only

partial channel blocks from encoder side is concatenated to the decoder. Since splitting and concatenating channels are invertible operation, therefore, this does not affect the invertibility of the overall architecture.

Mathematically, the downsampling and upsampling operations in the invertible U-net can be visualized using equations 4.3 and 4.4 respectively.

$$\begin{aligned} Y_i^e &= \phi_e(X_i^e) \\ [Y_{i1}^e, Y_{i2}^e] &= split(Y_i^e) \\ X_{i+1}^e &= Downsample(Y_{i1}^e) \end{aligned} \tag{4.3}$$

$$\begin{aligned} X_{i1}^d &= Upsample(Y_{i-1}^d) \\ X_i^d &= Concatenate(X_{i1}^d, Y_{i2}^e) \\ Y_i^d &= \phi_d(X_i^d) \end{aligned} \tag{4.4}$$

where Y_i^e, is the output of coupling block ϕ_e at i^{th} depth on encoder side, X_i^e is the input to the coupling block, Y_{i1}^e and Y_{i2}^e are the splitted channel at i^{th} depth and only Y_{i1}^e is further processed at encoder side and the other half is concatenated with decoder output at that depth, X_{i+1}^e is the input to the coupling block at $i + 1^{th}$ depth. Similarly, Y_{i-1}^d is the input to upsampling filter at $i - 1^{th}$ depth, X_{i1}^d is the upsampled image at i^{th} depth at decoder side, X_i^d is the concatenated output of the remaining encoder channel Y_{i2}^e with upsampled image, and Y_i^d is the output of coupling block ϕ_d at i^{th} depth on decoder side.

The downsampling operations are repeated to extract input data context into a feature vector, which is later recovered using upsampling operations. The advantage of invertible U-net architecture is that, unlike traditional U-net, it can be used to learn bijective mapping i.e. both forward and inverse translation simultaneously.

4.3 Invertible X-net

In the last section, we saw the advantages of invertible U-net over the traditional U-net, however, the symmetrical design of invertible U-net also prohibits it to be used only when input and target domains are of same size strictly. To overcome this challenge, we propose our novel architecture that builds upon the invertible U-net and allows it to be used for any arbitrary input and target domain dimensions (see Figure 4.3).

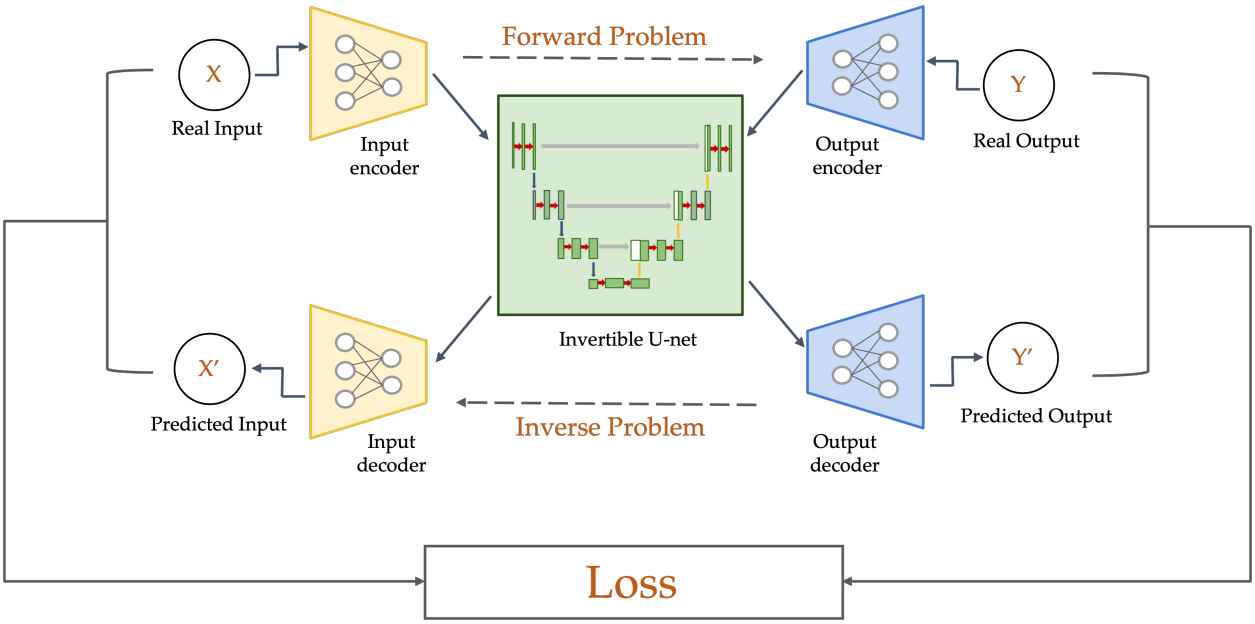


Figure 4.3: Invertible X-net architecture.

Mathematically, the visualization of the forward and inverse pass using this architecture is straightforward as described in following equations:

$$\begin{aligned} X_l &= X_{enc}(x) \\ Y_l &= f(X_l) \\ y &= Y_{dec}(Y_l) \end{aligned} \tag{4.5}$$

$$\begin{aligned} Y_l &= Y_{enc}(y) \\ X_l &= f(Y_l) \\ x &= X_{dec}(X_l) \end{aligned} \tag{4.6}$$

where X_l and Y_l is the latent space representation of the input data x and target label y, X_{enc}, X_{dec} is the encoder and decoder for input data respectively, Y_{enc}, Y_{dec} is the encoder and decoder for the target label respectively, and f is the invertible U-net as discussed in the previous section.

The main idea behind this architecture is to leverage invertible U-net to learn invertibility in the latent space where the input and output dimensions are strictly equal. Therefore, we add encoder and decoder blocks on both side of the input and target domains to learn a meaningful latent representations. First, the input data would be passed to a latent encoder, the output of this encoder would be further fed to the invertible U-net, and finally, the output from the invertible network would be decoded to make a prediction in the target domain. The advantage of using this architecture are: (1) it gives us flexibility in learning both forward and inverse transformations with a single architecture, (2) the input and target domains can be of any arbitrary dimensions, and (3) encoder-decoder blocks are trained end-to-end from scratch.

Chapter 5

Experiments and Results

In this chapter, we first briefly discuss the OpenFWI benchmark dataset used for the present study in section 5.1. Further, we detail our Invertible X-net architecture used for the experiments in section 5.2. Finally, we discuss the results of our experiments for different OpenFWI datasets and compare the prediction of seismic velocity and wavefields with baseline models in section 5.3.

5.1 OpenFWI Dataset

The OpenFWI comprises multi-structural benchmark datasets of significant size that can be used for solving full waveform inversion (FWI) using machine learning techniques [2]. In particular, the repository contains 4 major groups of data: (1) Vel Family, (2) Fault Family, (3) Style Family, and (4) Kimberlina Family. These groups represent simple to complex sub-surface geological settings with seismic velocity and waveforms information. The Vel family is the simplest geological patterns including four dataset - (1) FlatVel-A, (2) FlatVel-B, (3) CurveVel-A, and (4) CurveVel-B. The difference between FlatVel and CurveVel is that the former represents low-energy geological environment where the rock layers are deposited horizontally and the latter consists of curve layers which is formed due to structural deformation of flat layers. Further, the Fault family also has four dataset - (1) FlatFault-A, (2) FlatFault-B, (3) CurveFault-A, and (4) CurveFault-B. Unlike Vel datasets, the Fault

family contains fault-like deformation, which is fracturing of rocks under certain pressure conditions. Due to presence of fault, the geological setting becomes more complicated and challenging to model. In summary, the detail about the Vel and Fault datasets is described in table 5.1. The waveform data is represented as (# source × recording time × receiver length) whereas the velocity follow (1 × depth × receiver length) shape. In seismic, the wave arrival time is recorded at the surface and thus the waveform data for a single source is represented as a function of time and receiver length. In total, there are 5 seismic sources uniformly spaced along the surface and wavefields are recorded by 70 receivers (uniformly spaced) along the surface for 1000 milliseconds. Therefore, the seismic wavefield are of the shape (5 × 1000 × 70). On the other hand, velocity maps are represented as function of spatial dimensions that is depth and horizontal coverage and has the shape (1 × 70 × 70).

Dataset	**#Examples**	**Velocity shape**	**Waveform shape**
FlatVel-A	30,000	(1, 70, 70)	(5, 1000, 70)
FlatVel-B	30,000	(1, 70, 70)	(5, 1000, 70)
CurveVel-A	30,000	(1, 70, 70)	(5, 1000, 70)
CurveVel-B	30,000	(1, 70, 70)	(5, 1000, 70)
FlatFault-A	60,000	(1, 70, 70)	(5, 1000, 70)
CurveFault-A	60,000	(1, 70, 70)	(5, 1000, 70)

Table 5.1: Description of OpenFWI dataset.

5.2 Model Architecture

For OpenFWI dataset, we design an invertible X-net model such that encoder blocks of velocity and waveforms project them to a 128 × 70 × 70 latent space. The forward problem solves for the seismic waveform given the velocity, while the inverse problem involves solving for seismic velocity given an input seismic waveform.

The velocity encoder consists of 5 convolution blocks, each comprising a 1 × 1 convolution,

batch normalization, and LeakyReLU activation. In contrast, the velocity decoder comprises 5 deconvolution blocks, each comprising a 1×1 transposed convolution, batch normalization, and tanh activation (see Table 5.2).

The waveform encoder-decoder also have 5 convolutional and deconvolution blocks respectively. As the waveform inputs are rectangular, the kernel size, strides, and padding are rectangular in size (see Table 5.3).

Layers	**Convolution**	**Stride**
Encoder blocks		
Layer 1+BN+Leaky ReLu	(1,1)	(1,1)
Layer 2+BN+Leaky ReLu	(1,1)	(1,1)
Layer 3+BN+Leaky ReLu	(1,1)	(1,1)
Layer 4+BN+Leaky ReLu	(1,1)	(1,1)
Layer 5+BN+Leaky ReLu	(1,1)	(1,1)
Upsample	-	-
Decoder blocks		
Layers	**Convolution**	**Stride**
Layer 6+BN+Tanh	(1,1)	(1,1)
Layer 7+BN+Tanh	(1,1)	(1,1)
Layer 8+BN+Tanh	(1,1)	(1,1)
Layer 9+BN+Tanh	(1,1)	(1,1)
Layer 10+BN+Tanh	(1,1)	(1,1)
Convolution Layer	(1,1)	(1,1)
Upsample	-	-

Table 5.2: Architecture detail for velocity encoder and decoder used for designing Invertible X-net for the OpenFWI dataset.

The Invertible U-net consists of a learnable downsampler and upsampler, each consisting of 4 invertible blocks, and each invertible block contains 4 invertible layers. Throughout our experiments, we utilize the same model architecture and perform end-to-end training of encoders, invertible U-net, and decoders.

Layers	Convolution	Stride	Padding
Encoder blocks			
Layer 1+BN+LeakyReLU	(7,1)	(3,1)	(3,0)
Layer 2+BN+LeakyReLU	(7,1)	(2,1)	(1,0)
Layer 3+BN+LeakyReLU	(5,1)	(2,1)	(1,0)
Layer 4+BN+LeakyReLU	(5,1)	(1,1)	(0,0)
Layer 5+BN+LeakyReLU	(5,1)	(1,1)	(0,0)
Upsample	-	-	-
Decoder blocks			
Layers	**Convolution**	**Stride**	**Padding**
Layer 6+BN+Tanh	(7,1)	(3,1)	(0,0)
Layer 7+BN+Tanh	(7,1)	(2,1)	(0,0)
Layer 8+BN+Tanh	(5,1)	(2,1)	(0,0)
Layer 9+BN+Tanh	(5,1)	(1,1)	(0,0)
Layer 10+BN+Tanh	(5,1)	(1,1)	(0,0)
Convolution Layer	(7,3)	(1,1)	(3,1)
Upsample	-	-	-

Table 5.3: Architecture detail for waveform encoder and decoder used for designing Invertible X-net for the OpenFWI dataset.

5.3 Experiments

In this study, we examine five datasets from the Vel and Fault family, namely FlatVel-A, FlatVel-B, CurveVel-A, CurveVel-B, and FlatFault-A. Since the Invertible X-net model solves both forward and inverse problems jointly, we use two baseline models for velocity and one for waveform to compare with our model. Our velocity results are compared with the pretrained InversionNet and VelocityGAN models, while our seismic waveform results are compared with our own formulated baseline WaveformNet model (an encoder-decoder model for waveform prediction). Additionally, we also compare the seismic trace of predicted waveforms with observed waveforms to visualize the predicted wave amplitude and phase. We train and validate our model (80:20 split) using a learning rate of 1e-3 and optimized using Adam and StepLR scheduler. Due to the asymmetry in the minimization of the forward and inverse problems, we employ a weighted loss function for training.

Dataset	InversionNet			VelocityGAN			Invertible X-net		
	MAE	MSE	SSIM	MAE	MSE	SSIM	MAE	MSE	SSIM
FlatVel-A	19.65	1000.44	0.9895	17.76	712.63	0.9916	**8.64**	**251.82**	**0.9949**
FlatVel-B	52.77	17271.53	0.9461	49.36	14639.84	0.9521	**32.66**	**11923.30**	**0.9679**
CurveVel-A	102.77	36441.60	0.8074	72.38	24071.10	0.8624	**58.92**	**17107.70**	**0.9003**
CurveVel-B	224.57	188047.62	0.6727	190.17	154236.04	0.7111	**162.26**	**125230.40**	**0.7570**
FlatFault-A	25.85	4084.73	0.9766	130.21	49621.83	0.9313	**17.15**	**1513.48**	**0.9867**

Table 5.4: Experimental results for the velocity prediction on the OpenFWI dataset using InversionNet, VelocityGAN, and Invertible X-net (proposed method) models.

Dataset	WaveformNet			Invertible X-net		
	MAE	MSE	SSIM	MAE	MSE	SSIM
FlatVel-A	0.0686	0.1375	0.7899	**0.0211**	**0.0047**	**0.9192**
FlatVel-B	0.1053	0.1419	0.6257	**0.0432**	**0.0125**	**0.7945**
CurveVel-A	0.1051	0.1561	0.5826	**0.0580**	**0.0216**	**0.6667**
CurveVel-B	0.1667	0.1931	0.4131	**0.1075**	**0.0504**	**0.5121**
FlatFault-A	0.0766	0.1362	0.7761	**0.0396**	**0.0144**	**0.8147**

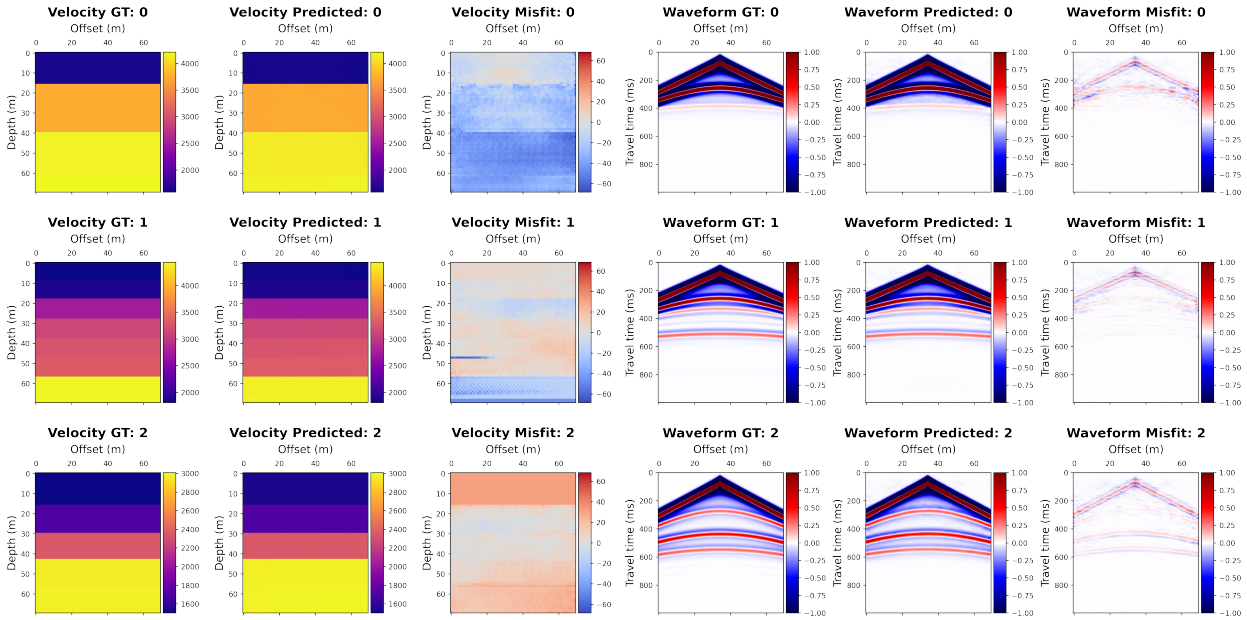
Table 5.5: Experimental results for the waveform prediction on the OpenFWI dataset using WaveformNet and Invertible X-net (proposed method) models.

5.3.1 FlatVel Dataset

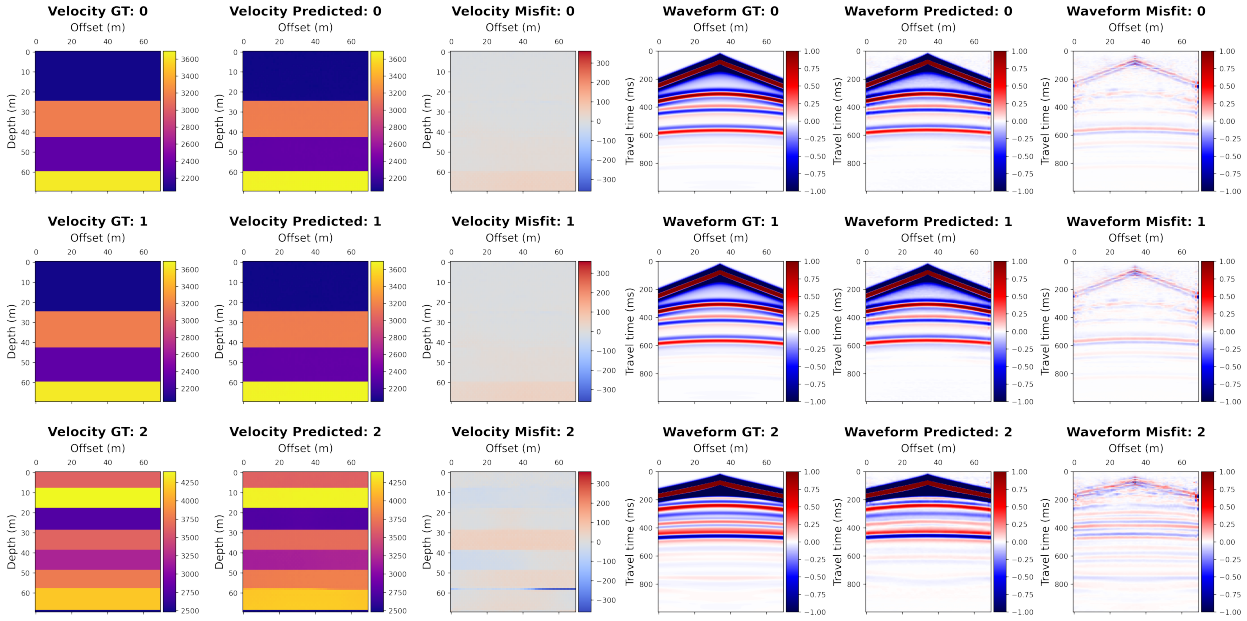
The FlatVel family consists of two different datasets - FlatVel-A and FlatVel-B. These datasets cover simpler geological settings where sub-surface layerings are horizontal with constant velocity and varying layer thickness. Both FlatVel-A and FlatVel-B dataset contain 30,000 paired velocity and waveform data, out of which 24,000 samples are used for training and the remaining are used for validation.

While the two datasets capture rapid variation in velocity, FlatVel-B is relatively more challenging than the FlatVel-A dataset. The FlatVel-B is created to simulate complicated geological settings with locally increasing and decreasing trend in velocity values. Due to non-monotonic trend in velocity, the seismic waveforms become more complicated with multiple wave arrivals coming from sub-surface interfering with each other.

For Flatvel group, we train two models on FlatVel-A and FlatVel-B datasets separately

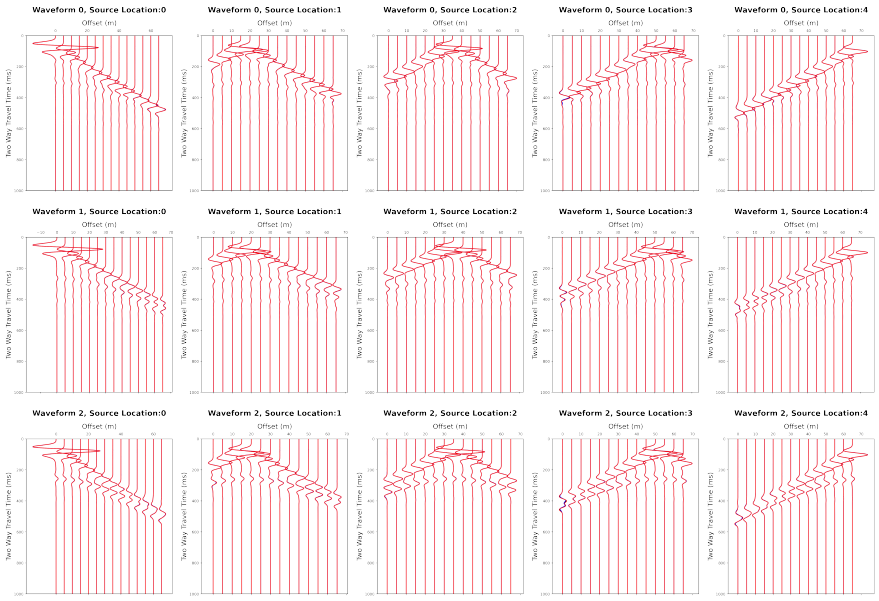


(a) FlatVel-A

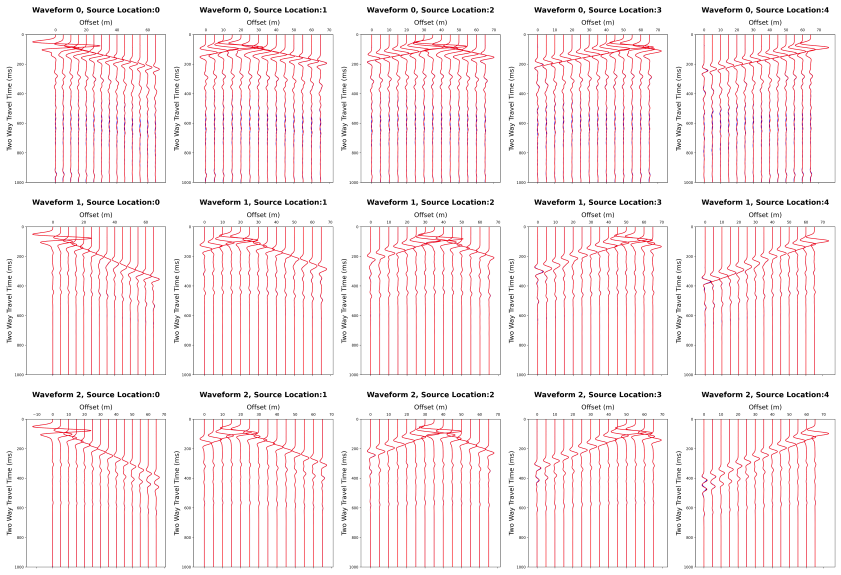


(b) FlatVel-B

Figure 5.1: Result of Invertible X-net model prediction on FlatVel datasets on randomly selected samples from validation datasets. For waveforms, we show the predicted and observed waveform for the 2nd source out of 5 seismic sources.



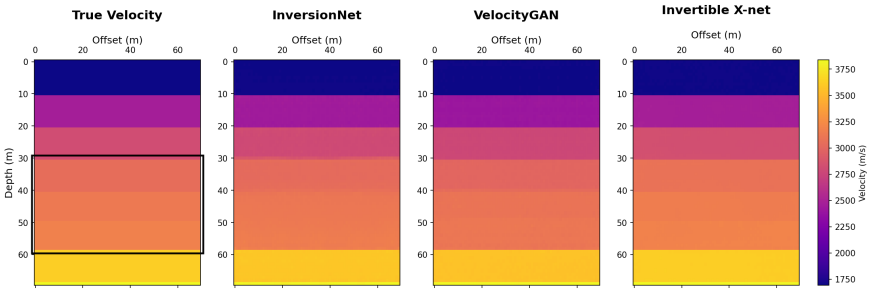
(a) FlatVel-A



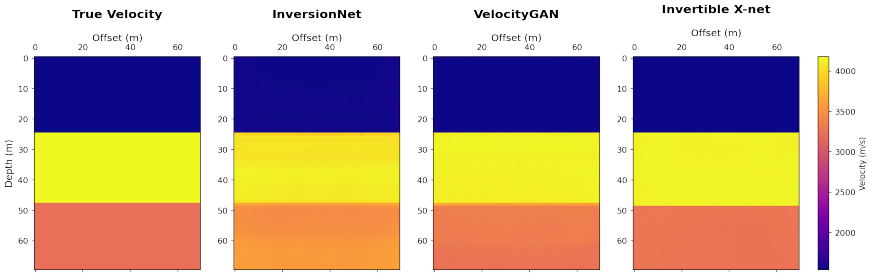
(b) FlatVel-B

Figure 5.2: Plotting seismic traces for FlatVel datasets for every 5th receiver out of 70 receivers to compare the predicted and observed waveforms for all 5 seismic sources. Seismic trace is the recorded travel time of seismic waves at a receiver on the surface. Predicted and observed waveforms are plotted in blue and red respectively.

and assess their performance on their respective validation datasets. In Figure 5.1, we show results of our models on randomly selected samples from validation datasets of FlatVel group. While our models are able to predict velocity accurately, they are also able to predict seismic waveforms, capturing stronger and weaker seismic wave arrivals reliably.



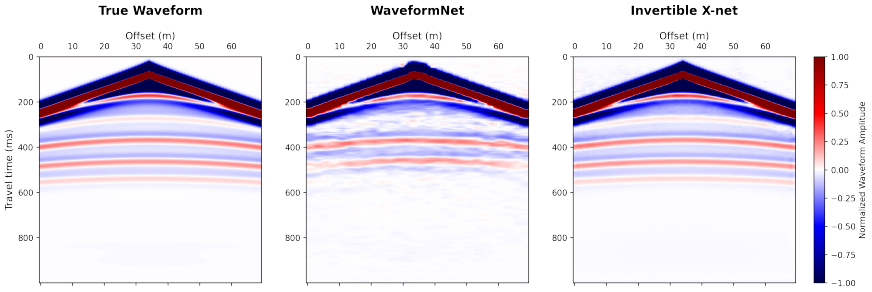
(a) FlatVel-A



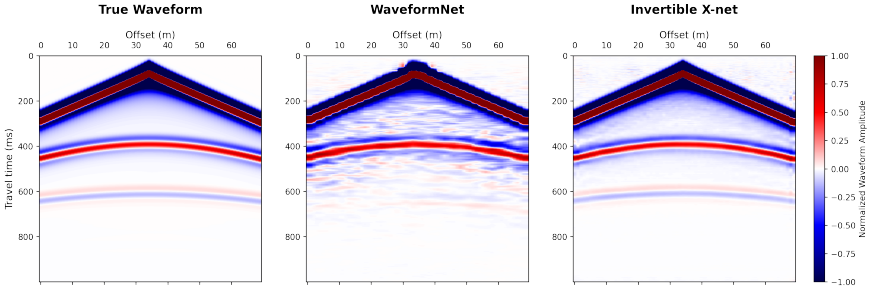
(b) FlatVel-B

Figure 5.3: Comparing the Invertible X-net model velocity prediction with baseline models - InversionNet and VelocityGAN on a randomly selected sample from validation datasets of FlatVel group.

Overall, our velocity predictions are more realistic and sharper than InversionNet and VelocityGAN models. For FlatVel-A dataset, in terms of MAE and MSE, the errors in velocity predictions relative to InverionNet drop by 56.03 % and 74.82 % respectively. We also compare our models against VelocityGAN, which is a stronger baseline than InversionNet, and observe that MAE and MSE for velocity predictions reduce by 51.35 % and 64.66 % respectively. Similarly, for FlatVel-B dataset, the MAE and MSE of our velocity predictions with respect to InversionNet decrease by 38.10 % and 30.96 % respectively. In comparison to VelocityGAN, the MAE and MSE decrease by 33.83 % and 18.55 % respectively. Finally,



(a) FlatVel-A



(b) FlatVel-B

Figure 5.4: Comparing the Invertible X-net model waveform prediction with the WaveformNet baseline on a randomly selected sample from validation datasets of FlatVel group.

we also measure the structural similarity (SSIM) of our velocity predictions with respect to ground truth and show that our models yield better results than the two baseline velocity models (see Table 5.4).

In the visualization (see Figure 5.3), we observe that both InversionNet and VelocityGAN predict slightly blurry results, especially in deep regions, while our Invertible X-net model generates sharper and clearer boundaries with accurate predictions.

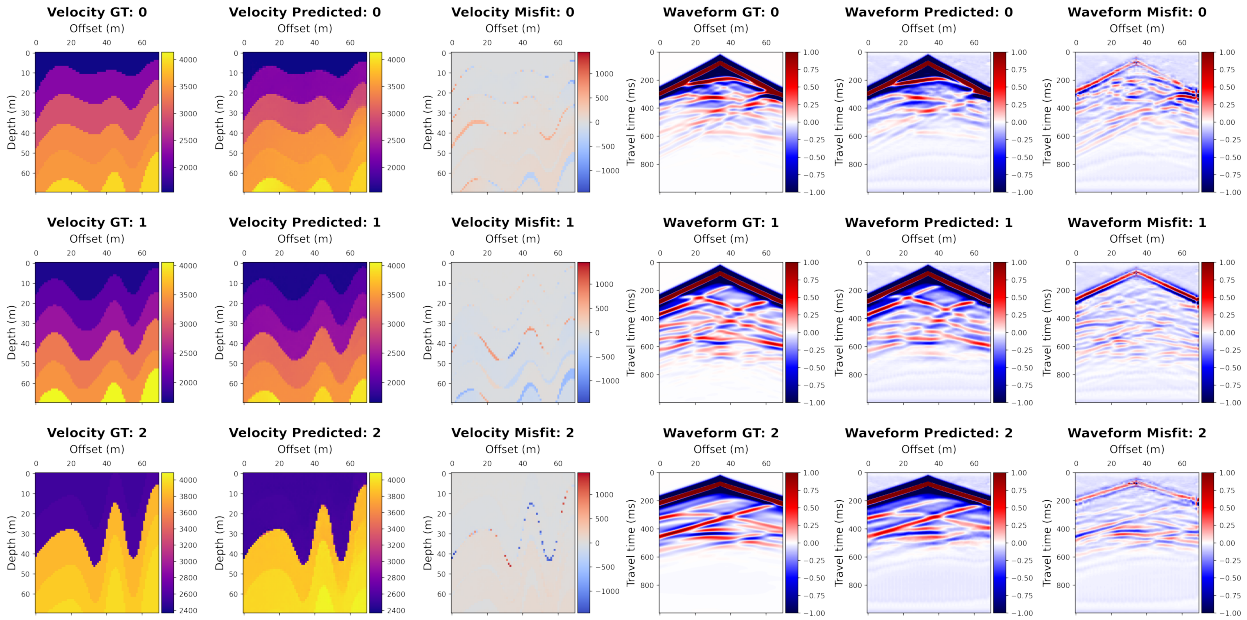
For waveforms, we first compare the predicted seismic trace for every 5^{th} receiver for all seismic sources to the corresponding observed seismic trace to visualize the amplitude and phase misfit (see Figure 5.2). Our waveform predictions are very accurate with minute differences with respect to observed waveforms. Further, we compare our waveform predictions with the WaveformNet baseline (see Figure 5.4) and show that for all metrics our predictions are better than the WaveformNet model by a significant margin (see Table 5.5). For FlatVel-A

dataset, the drop in MAE and MSE of our model prediction compared to WaveformNet is 69.24 % and 96.58 % respectively. However, for FlatVel-B dataset, the MAE and MSE of our model prediction decrease by 58.97 % and 91.19 % respectively.

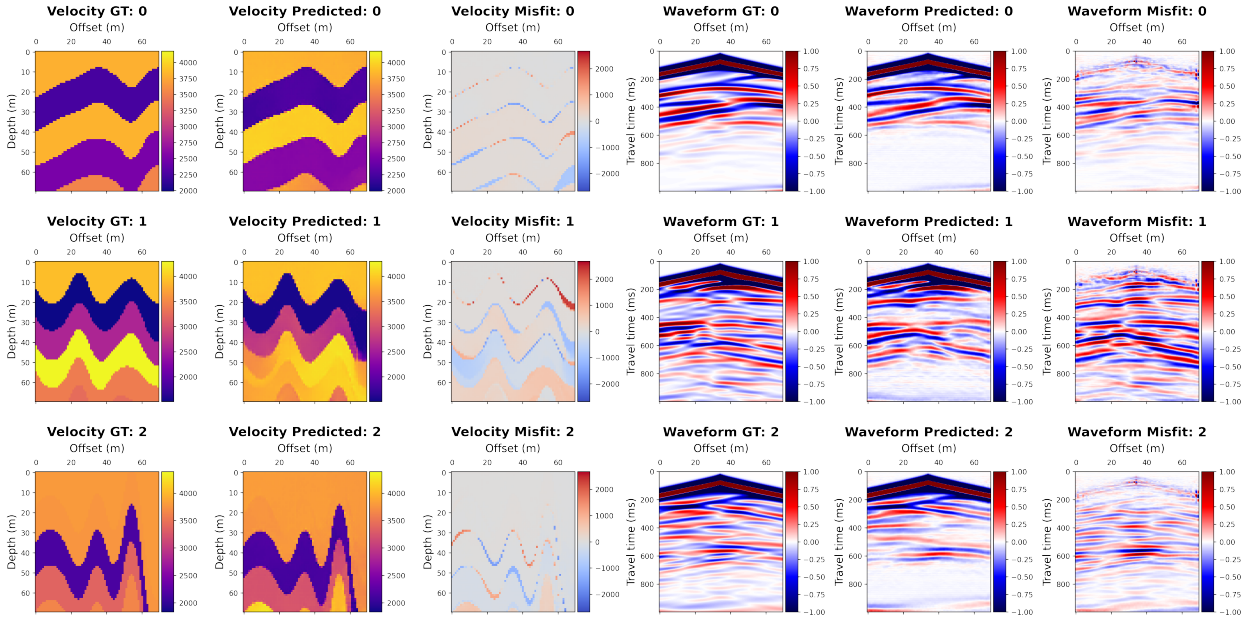
5.3.2 CurveVel Dataset

Similar to the FlatVel group, the CurveVel dataset also consists of two datasets, CurveVel-A and CurveVel-B. The CurveVel dataset contains curve layers representing fold like structure due to slow deformation of rocks. In CurveVel-A, the velocity is monotonically increasing with depth, however, the CurveVel-B has locally increasing and decreasing trend in velocity making it much more challenging than CurveVel-A. In general, the CurveVel datasets are more challenging than FlatVel because of varying thickness and rapid changes in deformed layers. Like FlatVel group, we train two models respectively for the Curvel-A and CurveVel-B datasets.

We demonstrate results of velocity and waveform predictions for CurveVel-A and CurveVel-B datasets in Figure 5.5. Despite complex geological structures, our models are able to outperform baseline models in solving both forward and inverse problems. For CurveVel-A dataset, the MAE and MSE for velocity predictions of our model compared to InversionNet drop by 42.66 % and 53.05 % respectively. However, the MAE and MSE for velocity predictions of our model when compared against VelocityGAN decrease by 18.95 % and 28.92 % respectively. Similarly, for CurveVel-B dataset, our model shows a decrement of 27.74 % and 33.40 % in MAE and MSE of predicted velocity, respectively, compared to the InversionNet model. Moreover, the drop in MAE and MSE of our model for predicted velocity in comparison to the VelocityGAN model is 14.67 % and 18.80 % respectively. Of note, the structural similarity (SSIM) for our models predictions with the ground truth on both



(a) CurveVel-A



(b) CurveVel-B

Figure 5.5: Result of Invertible X-net model prediction on CurveVel datasets on randomly selected samples from validation datasets. For waveforms, we show the predicted and observed waveform for the 2nd source out of 5 seismic sources.

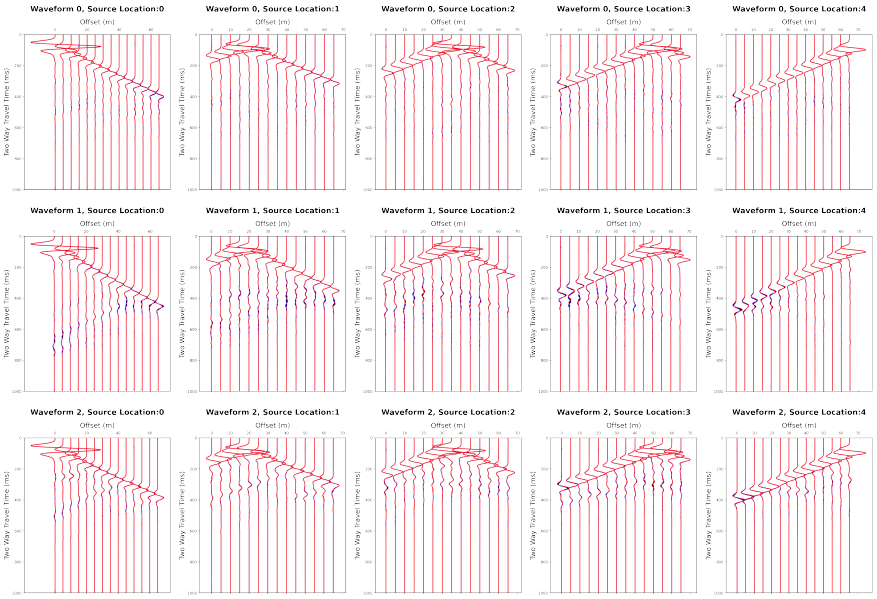
datasets are comparatively better than InversionNet and VelocityGAN models (see Table 5.4).

At last, we show the performance of our models against the baseline models on a randomly selected sample from validation datasets. In Figure 5.7 (a), we can clearly see that the our model is able to predict last layer much better than the other two models. In the same way, the prediction on CurveVel-B dataset using the two baseline models are smooth with some inaccuracies, whereas, our model yield more realistic, sharper, and accurate prediction (see Figure 5.7 (b)).

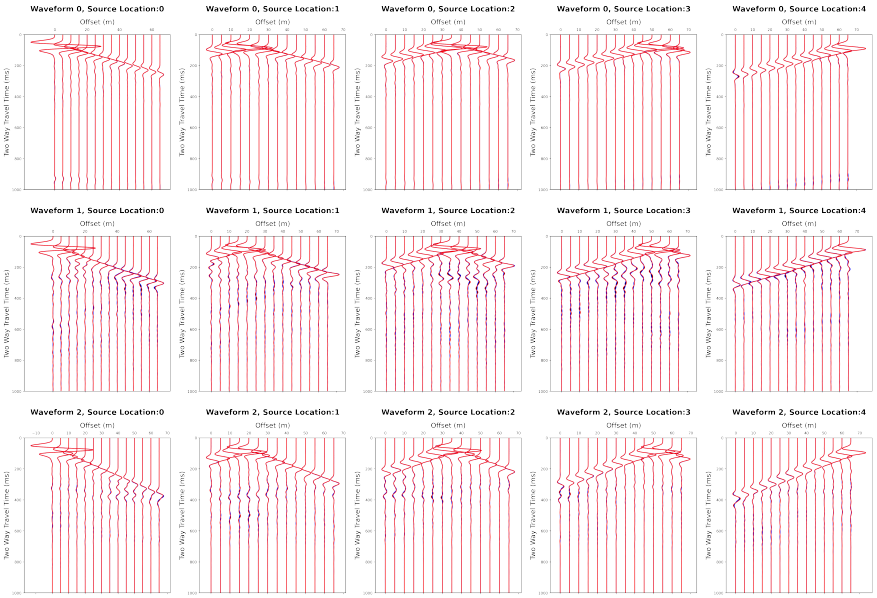
To investigate amplitude and phase of our waveform predictions with respect to observed waveforms, we plot the predicted seismic trace for every 5^{th} receiver out of 70 receivers and project the corresponding observed seismic trace over it. Our waveform predictions match with observed waveforms with minor amplitude and phase mismatch indicating we are able to predict most of seismic wave arrivals in waveforms correctly (see Figure 5.6). Finally, we compare our predicted waveforms to the WaveformNet (baseline) model. Though, seismic waveforms for CurveVel dataset are much more complicated than FlatVel, our models are able to predict most of the prominent reflections and outperform baseline model on both datasets with a significant margin (see Figure 5.8). For CurveVel-A dataset, in terms of MAE and MSE, our model waveform predictions compared to WaveformNet decrease by 44.81 % and 86.16 % respectively. Similarly, for CurveVel-B dataset, the drop in MAE and MSE of our model prediction compared to the WaveformNet is 35.51 % and 73.79 % respectively.

5.3.3 FlatFault Dataset

The FlatFault dataset showcases different geological conditions where underlying layers are shifted relative to each other, causing a linear discontinuity that behaves like a fracture.

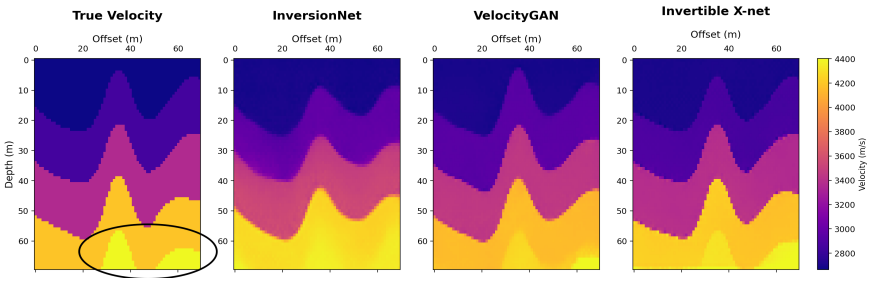


(a) CurveVel-A

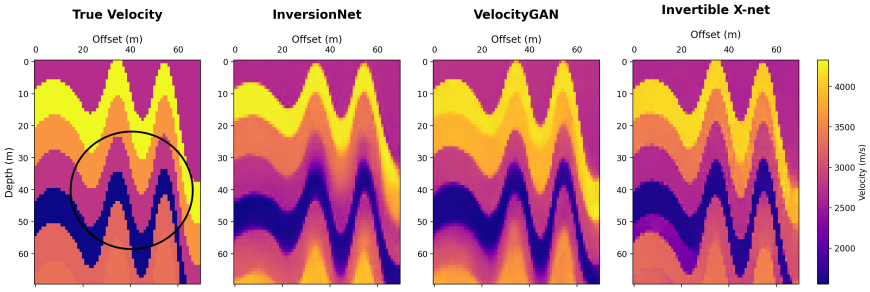


(b) CurveVel-B

Figure 5.6: Plotting seismic traces for CurveVel datasets for every 5th receiver out of 70 receivers to compare the predicted and observed waveforms for all 5 seismic sources. Seismic trace is the recorded travel time of seismic waves at a receiver on the surface. Predicted and observed waveforms are plotted in blue and red respectively.

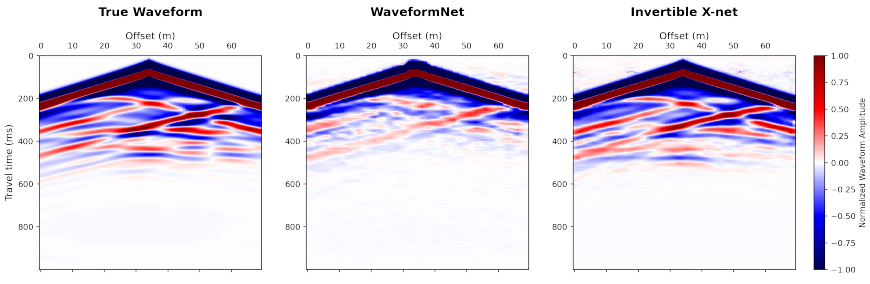


(a) CurveVel-A

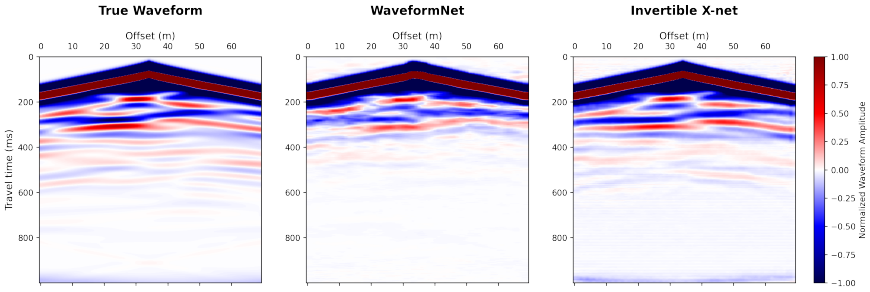


(b) CurveVel-B

Figure 5.7: Comparing the Invertible X-net model velocity prediction with baseline models - InversionNet and VelocityGAN on a randomly selected sample from validation datasets of CurveVel group.



(a) CurveVel-A



(b) CurveVel-B

Figure 5.8: Comparing the Invertible X-net model waveform prediction with the WaveformNet baseline on a randomly selected sample from validation datasets of CurveVel group.

The detection of such faults are of primary importance for velocity inversion as they often have prospects for energy exploration purpose. Unlike Vel family, we only show our model performance and its comparison for FlatFault-A dataset. We train our model on FlatFault-A under similar conditions and predict velocity and seismic waveforms.

In Figure 5.9, we can see the performance of the model on randomly selected samples from the validation dataset. The fault lines are very thin and confined to a local region making it more challenging to detect. Of note, the corresponding waveform response also have complicated wave interference patter that is difficult to map uniquely with the velocity field.

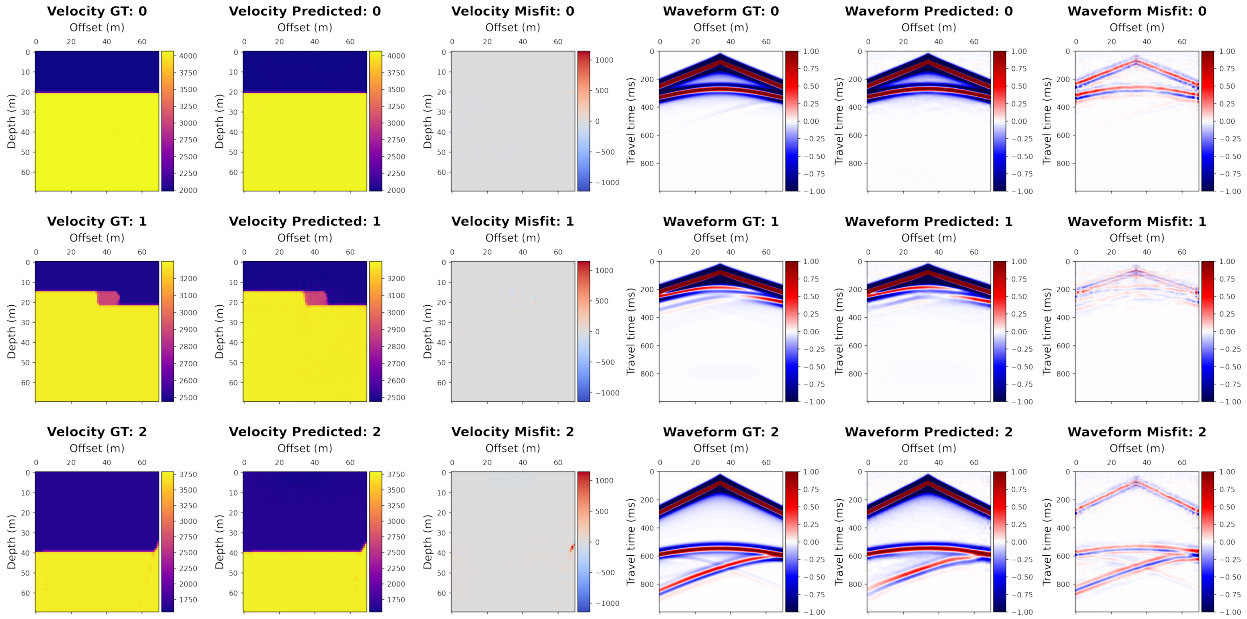


Figure 5.9: Result of Invertible X-net model prediction on FlatFault-A dataset on 3 randomly selected samples from validation dataset. For waveforms, we show the predicted and observed waveform for the 2nd source out of 5 seismic sources.

Nonetheless, the MAE and MSE for velocity predictions for our model compared to InversionNet decrease by 33.65 % and 63.50 % respectively. When we compare our model against VelocityGAN, the MAE and MSE drop by 86.82 % and 96.94 % respectively. Additionally, we compare our model on a randomly selected sample against InversionNet and VelocityGAN

and observe the improvement in our model prediction over baselines (see Figure 5.11). It is evident from Figure 5.11 that the InversionNet and VelocityGAN output blurry and smooth fault deformation whereas our model delineate fault more accurately with sharp boundaries.

On the other hand, the waveform predictions of our model compared to the observed waveforms looks satisfactory as well. Like FlatVel and CurveVel datasets, we compare the predicted seismic traces with the observed trace to examine the amplitude and phase mismatch. From Figure 5.10, we could clearly see that there is some small phase difference between the observed and predicted waveforms. This could be because the presence of these fault lines brings phase shift and that could be difficult to account for the model. Nonetheless, the waveform predictions for our model compared to the WaveformNet baseline model is much better and of high quality (see Figure 5.12). Upon evaluation, we find the MAE and MSE of our model predictions decreases in comparison to the WaveformNet by 48.30 % and 89.42 % respectively.

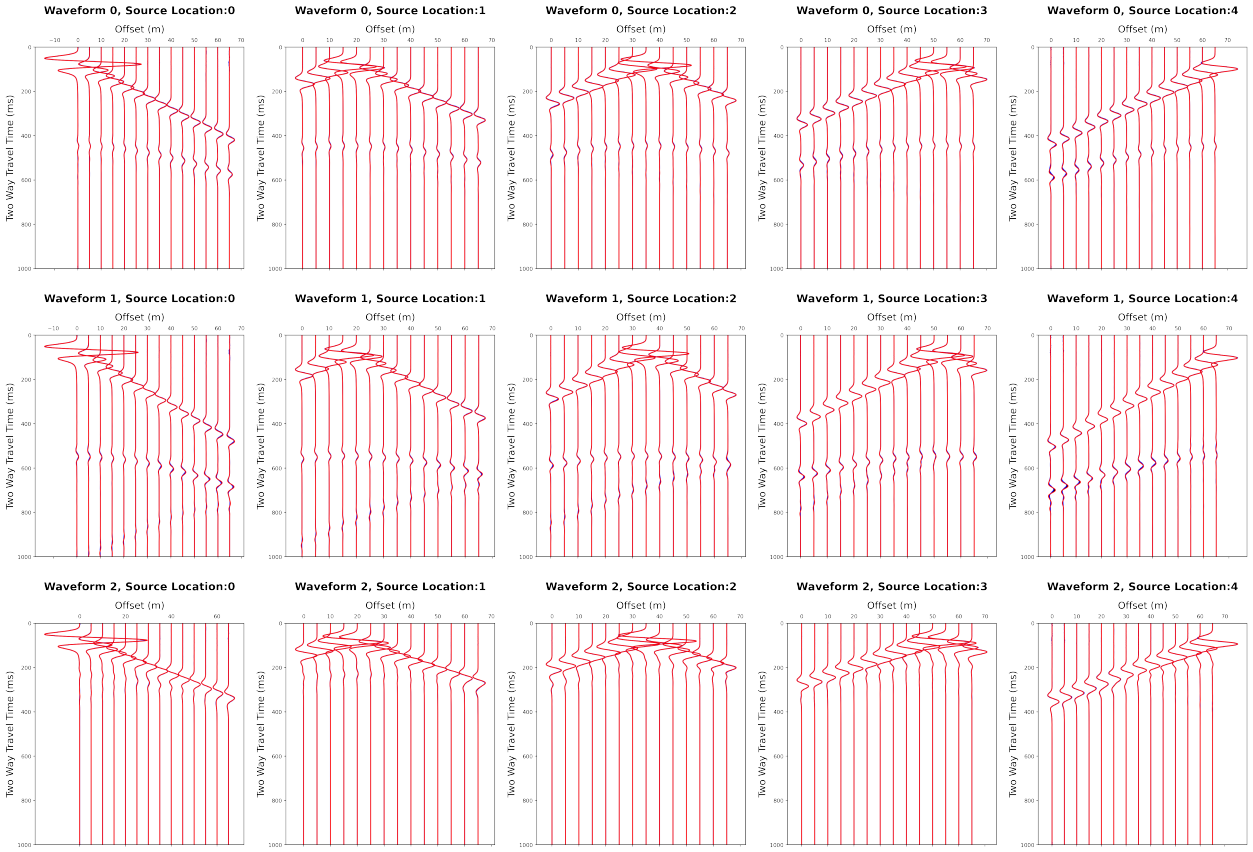


Figure 5.10: Plotting seismic traces for FlatFault-A datasets for every 5th receiver out of 70 receivers to compare the predicted and observed waveforms for all 5 seismic sources. Seismic trace is the recorded travel time of seismic waves at a receiver on the surface. Predicted and observed waveforms are plotted in blue and red respectively.

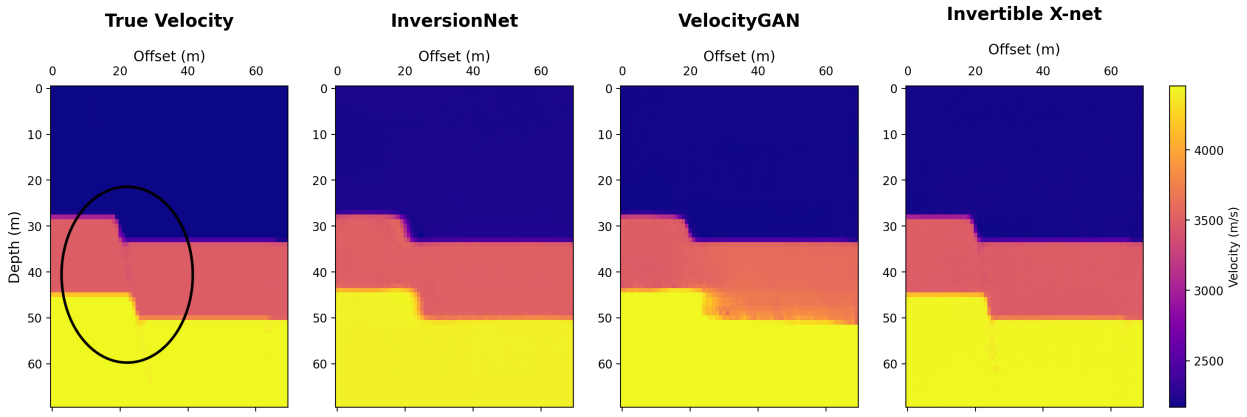


Figure 5.11: Comparing the Invertible X-net model velocity prediction with baseline models - InversionNet and VelocityGAN on a randomly selected sample from validation dataset of FlatFault-A dataset.

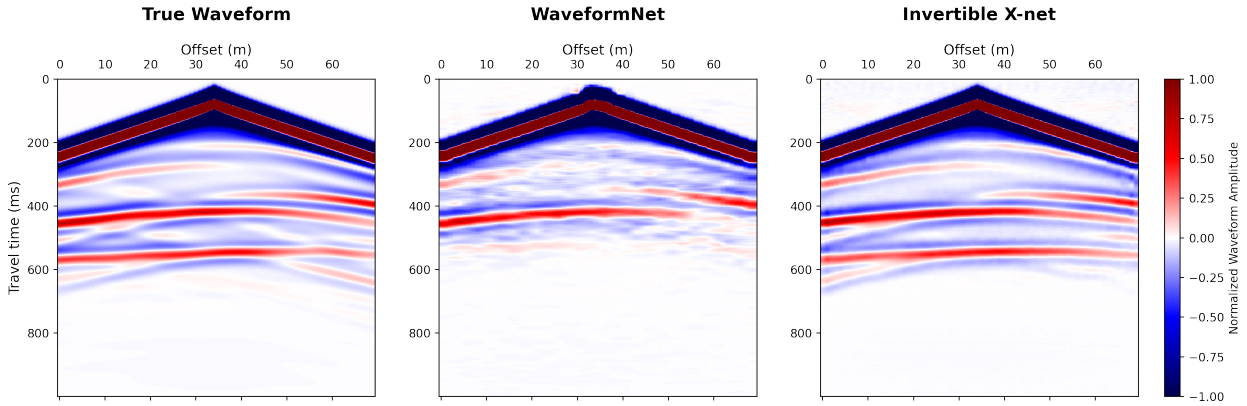


Figure 5.12: Comparing the Invertible X-net model waveform prediction with the WaveformNet baseline on a randomly selected sample from validation dataset of FlatFault-A dataset.

Chapter 6

Conclusions And Future Directions

Inspired by the architecture of Invertible U-net (Section 4.2), we propose a new architecture that allow us to invert computations for arbitrary input and output dimensions (Section 4.3). When the input and output dimensions are different, we proposed to use encoders and decoders to learn a latent representation of the input and output and then learn invertible mapping between the latent representation of the input and output distributions.

We trained our model, Invertible X-net, on the OpenFWI datasets which consists of different multi-structural geological seismic datasets for the FWI problem. We showed that our model not only outperforms the existing baselines on multiple datasets, but also accurately predicts seismic wavefields of high quality under variable velocity models. Traditional methods for predicting seismic wavefields require solving the second order partial differential equation - wave equation, which could be very expensive for variable velocity models. At last, we compared our wavefield prediction with the WaveformNet and show that our model is able to outperform baseline for all datasets.

6.1 Future Directions

The insights presented in the present study may serve as a foundation for some interesting avenues for future work like (1) unpaired FWI, (2) incorporating physics guidance in the current architecture, and (3) probabilistic approach to FWI. We discuss these ideas in brief

in following points.

- **Unpaired FWI:** Existing data-driven FWI employs paired training on the ground truth, while our approach jointly solve both forward and inverse problems. This opens up a new opportunity to explore unpaired FWI and in return, reduces the reliance on paired input and output labels. For unpaired training, we could use some fraction of training data with paired input and output label and the remaining fraction would be used in unpaired mode i.e. an input would predict an output (forward pass) and the predicted output would be used to reconstruct the original input (backward pass). Cycle loss could be used to measure the loss in reconstructing input and output labels for training the model.

- **Physics-guided FWI:** The integration of physics knowledge in the current architecture can be intriguing and useful in solving the forward problem. By incorporating wave equation (governing physics for seismic wave propagation) in the model, we could possibly learn a more generalize model with fewer training examples that is also consistent with physics.

- **Probabilistic approach to FWI:** Another interesting approach to solve FWI problem could be using probabilistic formulations, the current framework is purely deterministic and therefore the model predictions are conditioned on the training data. We could introduce stochasticity in the latent space to learn a probabilistic model that maximizes the likelihood of an observed output instead of minimizing the least square error. Using the stochasticity in latent space, we could also possibly predict multiple solutions of the non-unique inverse problem with their likelihood.

Bibliography

[1] Amir Asnaashari, Romain Brossier, Stéphane Garambois, François Audebert, Pierre Thore, and Jean Virieux. Regularized seismic full waveform inversion with prior model information. *Geophysics*, 78, 2013.

[2] Chengyuan Deng, Shihang Feng, Hanchen Wang, Xitong Zhang, Peng Jin, Yinan Feng, Qili Zeng, Yinpeng Chen, and Youzuo Lin. Openfwi: Large-scale multi-structural benchmark datasets for seismic full waveform inversion. 2022.

[3] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. 2015.

[4] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[5] Björn Engquist, Brittany D. Froese, and Yunan Yang. Optimal transport for seismic full waveform inversion. *Communications in Mathematical Sciences*, 14(8):239–233, 2016.

[6] Christian Etmann, Rihuan Ke, and Carola-Bibiane Schönlieb. iunets: Fully invertible u-nets with learnable up- and downsampling. 2020.

[7] Gene H Golub, Per Christian Hansen, and Dianne P O'Leary. Tikhonov regularization and total least squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1): 185–194, 1999.

[8] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. pages 5769–5779, December 2017.

[9] Qinglong He and Yanfei Wang. Reparameterized full-waveform inversion using deep neural networks. *Geophysics*, 86(1):V1–V13, 2021.

[10] Peng Jin, Xitong Zhang, Yinpeng Chen, Sharon Huang, Zicheng Liu, and Youzuo Lin. Unsupervised learning of full-waveform inversion: Connecting cnn and partial differential equation in a loop. 2022.

[11] Youzuo Lin and Lianjie Huang. Acoustic- and elastic-waveform inversion using a modified total-variation regularization scheme. *Geophysical Journal International*, 200(1): 489–502, 2015.

[12] Youzuo Lin, James Theiler, and Brendt Wohlberg. Physics-guided data-driven seismic inversion: Recent progress and future opportunities in full-waveform inversion. *IEEE Signal Processing Magazine*, 40(1):115–133, 2023.

[13] R.-E. Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167(2): 495–503, 11 2006. ISSN 0956-540X. doi: 10.1111/j.1365-246X.2006.02978.x.

[14] Yuxiao Ren, Xinji Xu, Senlin Yang, Lichao Nie, and Yangkang Chen. A physics-based neural-network way to perform seismic full waveform inversion. *IEEE Access*, 8:112266–112277, 2020.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. 2015.

[16] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1605.06211*, 2016.

[17] Albert Tarantola. Theoretical background for the inversion of seismic waveforms including elasticity and attenuation. *Geophysics*, 49:1259–1266, 1984.

[18] Anders U Waldeland, Are Charles Jensen, Leiv-J Gelius, and Anne H Schistad Solberg. Convolutional neural networks for automated seismic interpretation. *The Leading Edge*, 37(7):529–537, 2018.

[19] Benfeng Wang, Ning Zhang, Wenkai Lu, and Jialin Wang. Deep-learning-based seismic data interpolation: A preliminary result. *Geophysics*, 84(1):V11–V20, 2019.

[20] Wenlong Wang, Fangshu Yang, and Jianwei Ma. Velocity model building with a modified fully convolutional network. *SEG Technical Program Expanded Abstracts*, page 2086–2090, 2018.

[21] Y Wang and Y Rao. Reflection seismic waveform tomography. *Journal of Geophysical Research: Solid Earth*, 114(B3):B03304, 2009.

[22] Xinming Wu, Luming Liang, Yunzhi Shi, and Sergey Fomel. Faultseg3d: Using synthetic data sets to train an end-to-end convolutional neural network for 3d seismic fault segmentation. *Geophysics*, 84(3):IM35–IM45, 2019.

[23] Yue Wu and Youzuo Lin. Inversionnet: An efficient and accurate data-driven full waveform inversion. *IEEE Transactions on Computational Imaging*, 6:419–433, 2019.

[24] Wei Xiong, Xu Ji, Yue Ma, Yuxiang Wang, Nasher M AlBinHassan, Mustafa N Ali, and Yi Luo. Seismic fault detection with convolutional neural network. *Geophysics*, 83 (5):O97–O103, 2018.

[25] Fangshu Yang and Jianwei Ma. Deep-learning inversion: A next-generation seismic velocity model building method. *Geophysics*, 84(4):R583–R589, 2019.

[26] Fangshu Yang and Jianwei Ma. Revisit geophysical imaging in a new view of physics-informed generative adversarial learning. *Computer Vision and Pattern Recognition*, 2021.

[27] Yunan Yang, Björn Engquist, Junzhe Sun, and Brittany F Hamfeldt. Application of optimal transport and the quadratic wasserstein metric to full-waveform inversion. *Geophysics*, 83(6):R43–R62, 2018.

[28] Colin A. Zelt and Penny J. Barton. Three-dimensional seismic refraction tomography: A comparison of two methods applied to data from the faeroe basin. *Journal of Geophysical Research: Solid Earth*, 103(B4):7187–7210, 1998. doi: 10.1029/97JB03316.

[29] Zhongping Zhang, Yue Wu, Zheng Zhou, and Youzuo Lin. Velocitygan: Subsurface velocity image estimation using conditional adversarial networks. pages 705–714, 2019. doi: 10.1109/WACV.2019.00080.

[30] Weiqiang Zhu, Kailai Xu, Eric Darve, Biondo Biondi, and Gregory C. Beroza. Integrating deep neural networks with full-waveform inversion: Reparameterization, regularization, and uncertainty quantification. *GEOPHYSICS*, 87:R93–R109, 2022.

[31] X. Zhu, D.P. Sixta, and B.G. Angstman. Tomostatics: turning-ray tomography+ static corrections. *The Leading Edge*, 11:15–23, 1992.