On the Robustness and Generalization of Deep Learning Driven Full Waveform Inversion

A Preprint

Chengyuan Deng
Los Alamos National Laboratory
charles.deng@lanl.gov

Youzuo Lin
Los Alamos National Laboratory
ylin@lanl.gov

November, 2021

ABSTRACT

The data-driven approach has been demonstrated as a promising technique to solve complicated scientific problems. Full Waveform Inversion (FWI) is commonly epitomized as an image-to-image translation task, which motivates the use of deep neural networks as an end-to-end solution. Despite being trained with synthetic data, the deep learning-driven FWI is expected to perform well when evaluated with sufficient real-world data. In this paper, we study such properties by asking: how robust are these deep neural networks and how do they generalize? For robustness, we prove the upper bounds of the deviation between the predictions from clean and noisy data. Moreover, we demonstrate an interplay between the noise level and the additional gain of loss. For generalization, we prove a norm-based generalization error upper bound via a stability-generalization framework. Experimental results on seismic FWI datasets corroborate with the theoretical results, shedding light on a better understanding of utilizing Deep Learning for complicated scientific applications.

1 Introduction

The surge of interest in exploiting Deep Learning for difficult scientific problems has been witnessed with remarkable success in physics [1, 2], geoscience [3, 4], and neuroscience [5, 6], etc. Usually combined with physics-informed constraints, the data-driven approaches typically solve the problems via an end-to-end manner, as a circumvention of the expensive computation and ill-posedness issue inherited from those problems. Full-waveform Inversion (FWI), emerges as a classical inverse problem as described [7, 8, 9]. The FWI explores geophysical properties such as site geology, stratigraphy, and rock quality employing reconstructing subsurface velocity models from seismic waveform signals. It aims at minimizing the misfit between the predicted and recorded seismic waveforms, thus lies in the family of optimization problems constrained by partial differential equations (PDEs). Figure 1 presents an example of the FWI and its corresponding forward modeling.

En route to mitigating the cycle-skipping and ill-posedness issues of FWI, researchers have revolutionized the methodology from “physics-driven”, which usually uses gradient-based optimization and incorporates physics constraints as a regularization term[10, 11], to “data-driven” approaches[3, 12, 13, 14]. The latter has demonstrated significant empirical superiority by utilizing deep neural networks. As demonstrated in figure 1, FWI is modelled as an image-to-image translation task, therefore sharing similarity with a myriad of domains in Computer Vision: style transfer[15, 16, 17], image super-resolution [18, 19], image restoration [20, 21], etc. Inspired by such observations, most current works introduce celebrated deep learning models such as Convolutional Neural Networks [3, 4], Generative Adversarial Networks [22, 23], etc. as the backbone, which learns the target velocity map in an end-to-end manner. This practice, however, evokes two concerns to be evaluated more discreetly: robustness and generalization behavior of the proposed deep learning models.

The issue of robustness and generalization of learning algorithms has been studied in theoretical machine learning regimes for decades [24, 25, 26, 27], followed by empirical techniques to ameliorate both properties [28, 29, 30, 31]. In recent years, people have extended the analysis to deep neural networks [32, 33, 34, 35], shedding light on understanding the obstacles for deep learning to perform well in more general scenarios. For deep learning-driven FWI, such difficulties
Figure 1: Seismic FWI (2D & 3D) and Forward Modeling

Figure 1 shows an example of seismic FWI with the relationship between velocity maps and seismic data. Forward modelling takes velocity maps as input and calculates the seismic signals with PDEs. While in practice, the seismic signals can be recorded by geological equipment, the FWI is expected to produce the velocity maps so that people can understand the subsurface geologic structures.

are exacerbated due to the following reasons. (1) Acquiring and processing real-world data requires arduous human efforts and an excessively long time (sometimes hundreds of years), therefore most DNNs for FWI are learned and evaluated on synthetic datasets. (2) The recorded seismic data for testing is usually contaminated by noise, however, the seismic data is synthesized from clean signals. (3) The training data is often synthesized with fixed geophysical features, such as source signature, wave frequency, however, there might be a slight shift on those features during testing, thus raising the issue of generalization with potential distribution drift. Furthermore, as elaborated in section 2, most of the proposed DNN models [3, 13, 4, 12] are based on an encoder-decoder architecture, thus sharing a possibility to perform similarly. Because of the above evidence, the desiderata grows to understand the robustness and generalization of DNNs for FWI problems, thus motivating the following questions:

*How robust are DNN models for FWI when tested with noisy data?*

*How well do these DNN models generalize?*

In this paper, we would like to answer the above questions. In the first question, the robustness of DNNs is defined as the performance degradation when they are tested with perturbed noisy data. We analyze deep neural networks trained with MAE loss and MSE loss, respectively. An upper bound of the loss gain is proved based on Lipschitz continuity and Hölder’s inequality for MAE loss. The bound indicates that the performance degradation is associated with the product of weight matrices at each layer and the noise itself. A corollary implies that the bound of MSE loss is looser than that of MAE loss as the noise level raises. The analysis of generalization is established upon the framework proposed by [27], which bridges a stability property of an algorithm to its generalization error. An upper bound of the generalization error is obtained, taking the norm of weights and the covering number as important parameters.
To provide intuitions on the theoretical results and connect them to practice, we conduct a meticulous empirical study on both topics by training an encoder-decoder deep neural network on standard seismic FWI datasets. In the robustness test, we illustrate the interplay between the perturbation intensity (SNR) and the prediction performance change for DNNs with MAE loss and MSE loss. The conclusion is that DNNs with MSE loss give worse predictions as the noise increases.

The experiments on generalization are developed in two settings: (1) standard-setting, where the generalization gap is defined as the difference between train loss and test loss, provided the data samples are i.i.d in both datasets, and (2) FWI-domain-setting, where the test data has a slight distribution drift caused by instability of the geological features in real-world data. We show that our generalization bounds have a positive relationship with the empirical generalization gap in both settings.

We summarize the contributions of this paper as follows:

1. Upper bounds of gain of loss for Deep Neural Networks tested with noisy data.
2. An upper bound of generalization error for Deep Neural Networks solving FWI in an end-to-end manner.
3. In addition to the context of FWI, our results and analysis can be extended to other inverse problems and computational imaging tasks with some adjustment, thus may potentially inspires broad interest in the inverse problems community.

The rest of this paper is arranged as follows: In Section 3, we introduce the necessary backgrounds and the notations; Section 4 focuses on robustness, and presents the upper bound of the difference when testing on noisy and clean data; Section 5 moves forward to generalization, which is demonstrated with the generalization error bound; Section 2 provides an overview of the research lines on deep learning-driven FWI and theoretical analysis of DNNs; In Section 6, we show empirical results implied by our theoretical results; Section 7 concludes the paper and proposes open problems.

2 Related Work

2.1 Deep Learning Driven FWI

With the recent progress of deep learning in image generation tasks, researchers have adopted deep neural networks as the first-of-choice as the data-driven solution for FWI. GeoDNN [36] proposed an 8-layer fully-connected neural network. Other attempts include InversionNet [3], modifiedFCN [13] and FCNVMB [4], all of which are encoder-decoder networks with CNN as the backbone. SeisInvNet [14] enhanced each seismic trace with auxiliary knowledge from neighborhood traces for better spatial correspondence. NNFWI [37] used deep models to generate a physical velocity model, which is then fed to a PDE solver to simulate seismic waveforms. All the mentioned work focused on 2D modeling, and we refer to a thorough survey [38] for complete references. Recently, Zeng et al. proposed InversionNet3D [12] based on an encoder-decoder network, as the first deep learning solution for high-resolution 3D FWI. InversionNet3D employs group convolution in the encoder and invertible layers in the decoder to achieve high efficiency and scalability. A fewer other works based on recurrent neural networks(RNNs) [39] are proposed as a contrast to CNNs. Unsupervised learning techniques are also exploited for representation learning and data augmentation for FWI [40, 41, 22, 42].

2.2 Generalization of Deep Neural Networks

There has been substantial progress on characterizing the generalization behavior of machine learning algorithms, both empirically and theoretically. The earliest works were propelled from a theoretical perspective of understanding the uniform convergence of empirical quantities to their mean. Several complexity measure frameworks were established then, and maintain as essential ingredients even in modern analysis: VC dimension [43, 44, 45]; Rademacher complexity [46, 47], fat-shattering dimension [26, 48], and PAC-Bayes bound [24]. Recently, initiated by [49], a family of norm-based bounds has emerged [49, 50, 51, 52, 53, 54]. This family employs the product of Frobenius norm or spectral norm of each layer as a crucial factor of the generalization bound. Notice that the norm-based bounds can be generally applied to both fully-connected neural networks (FNNs) and convolutional neural networks (CNNs). table 1 demonstrates a comparison of the norm-based bounds for both FNNs and CNNs. There are also independent works from various views. Long et al.[32] proved a bound for deep CNNs related to the total number of parameters and the distance from trained weights to initial weights. Ledent et al.[55] incorporated weight-sharing in CNNs and proved a bound for multi-classification setting.

Another perspective of understanding the generalization behavior of DNNs is empirical study. Arora et al.[34] observed the noise-stability of weight matrices as the networks go deeper via massive experiments, and proposed generalization bounds via a compression framework. Also by proposing the remarkable double descent phenomena, Nakirran et al.[56]
We consider three variants of DNNs: fully-connected neural networks (FNNs), general convolutional neural network (CNNs), and encoder-decoder convolutional neural networks (usually with all convolutional layers), as they are the first-of-choice in FWI problem and other computational imaging tasks [3, 4, 14, 13]. To simplify the analysis, we will show in this section that the three variants share a unified formulation.

### 3 Backgrounds

We firstly introduce definitions of the essential ingredients and formalizing the deep learning models in the regime of FWI and, along the way, bringing in the notations. Throughout this paper, we denote $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ and $\mathcal{Y} \subseteq \mathbb{R}^{n_y}$ as the real spaces, the inverse problem is to estimate $x \in \mathcal{X}$ from $y \in \mathcal{Y}$ with $\mathcal{F}^{-1}$ such that:

$$y = \mathcal{F}(x) + n,$$

where $n$ is usually assumed as Gaussian noise.

#### 3.1 Full Waveform Inversion

The Full Waveform Inversion (FWI) targets the optimal velocity map $m$ to minimize the difference between predicted and observed seismic data $u = u(x,t)$, where $x$ is the location and $t$ refers to a timestamp. Typically, the acoustic FWI follows the following PDE constraint:

$$\frac{\partial^2 u}{\partial t^2} = \nabla \cdot (m^2 \nabla) + f,$$

where $f$ is the source term.

Focusing on the partial differential equation, the forward modeling can be formulated in the form of: $\mathcal{F}(m) = u$, naturally the inverse problem is given by $m = \mathcal{F}^{-1}(u)$. The optimization task for such an inverse problem can be expressed as:

$$\min_m \mathcal{L}(\hat{u}(x,t,m), u) \text{ s.t. } F_{\text{acoustic}}(u,m) = 0.$$  \hspace{1cm} (2)

where $\mathcal{L}(\cdot)$ denotes the loss function, $\hat{u}$ is the estimated seismic data, and $F_{\text{acoustic}}(u,m)$ stands for the PDE in eq (1)

#### 3.2 Deep Neural Networks

We consider three variants of DNNs: fully-connected neural networks (FNNs), general convolutional neural network (CNNs), and encoder-decoder convolutional neural networks (usually with all convolutional layers), as they are the first-of-choice in FWI problem and other computational imaging tasks [3, 4, 14, 13]. To simplify the analysis, we will show in this section that the three variants share a unified formulation.

| Work | Simplified Bounds | Remark |
|------|-------------------|--------|
| [49] | $\tilde{O}(2^d \prod_{i=1}^d ||W_i||_F / \sqrt{n})$ | CNN: $W_i \rightarrow \text{op}_i$ |
| [50] | $\tilde{O}(\prod_{i=1}^d ||W_i||_\sigma (\sum_{i=1}^d ||W_i||_{L_2})^{3/2})^{3/2} / \sqrt{n}$ | CNN: $W_i \rightarrow \text{op}_i$ |
| [51] | $\tilde{O}(\prod_{i=1}^d ||W_i||_\sigma \sqrt{d^d w \sum_{i=1}^d ||W_i||_{L_2}^2} / \sqrt{n})$ | CNN: $W_i \rightarrow \text{op}_i, w \rightarrow cm$ |
| [52] | $\tilde{O}(\prod_{i=1}^d ||W_i||_\sigma \sqrt{d^d w} / \sqrt{n})$ | $d \gg w$, CNN: $W_i \rightarrow \text{op}_i, w \rightarrow cm$ |
| [53] | $\tilde{O}(\prod_{i=1}^d ||W_i||_F \cdot \min(1 / \sqrt{n}, \sqrt{d/dn})$ | CNN: $W_i \rightarrow \text{op}_i$ |

Table 1: A Comparison of Norm-based bounds

Notations: $n$ is the number of training samples, $d$ is the number of layers, $W_i$ is either the weight matrix for FNNs or the linear operator ($\text{op}_i$) for CNNs of the $i$-th layer and $w$ is the largest layer width; $|| \cdot ||_F$ denotes the Frobenius norm and $|| \cdot ||_\sigma$ denotes the spectral norm. For CNNs, $c$ denotes number of channels, $r$ represents the size of filter, $m$ is the number of outputs generated by the network.

suggested that larger model and size of training data may not be helpful. Notably, Jiang et al. [57] did a thorough empirical study on three most representative types of generalization error bounds.
For a neural network $G$ with $d$ layers, denote $(\sigma_1, \ldots, \sigma_d)$ as the activation function (e.g. ReLU) or pooling function after each layer, with the assumption that $\sigma_i$ is $l_i$-Lipschitz continuous. Recall that an inverse problem takes $y \in \mathcal{Y}$ as the input, for a fully-connected neural network, we obtain:

$$G_\theta(y) := \sigma_d \left( W_d \sigma_{d-1} \left( W_{d-1} \ldots \sigma_1 \left( W_1 \cdot y + b_1 \right) \ldots \right) + b_d \right),$$

where $\theta = \{ (W_i, b_i) : |W_i| \in \mathbb{R}^{n_i \times n_{i-1}}, b \in \mathbb{R} \}$. 

It has been demonstrated in [54, 58] that convolution is a linear operation, thus can be represented as matrix multiplication. Formally, we have the following:

**Fact 1.** Suppose a convolutional filter with dimension $d_c$ is imposed on input $x \in \mathbb{R}^{d_{in}}$, let $W \in \mathbb{R}^{1 \times d_c}$ be the weight matrix, then there exists a unique matrix $\text{op} \in \mathbb{R}^{d_{out} \times d_{in}}$ such that $\text{conv}(x, W) = \text{op} \cdot x$ and $\forall \text{op}_{i,j}$ is either 0 or $w_k \in W$.

In an encoder-decoder-based CNN, the down-sampling process is accomplished by a stack of convolution layers, while the up-sampling hinges on the decoder, which is generally composed of transposed convolution layers [59]. Because of the above facts, we denote $\text{op}_{i,j}$ as the matrix generated by the convolution operator imposed on $X_{i-1}$, so that Eq. eq (3) can be applied to encoder-decoder convolutional neural networks by setting $b_i = 0$. Specifically, denote $(W_1, \ldots, W_d)$ as the weights of convolutional kernels at each layer, $W_i \in \mathbb{R}^{c_i \times r_i}$ has $c_i$ convolutional kernels, each with kernel size $r_i$. Letting $y_i$ be the output of layer $i$, the following holds:

$$y_{i+1} = \sigma_{i+1} \left( \text{op}_{i+1} \left( W_{i+1} y_i \right) \right).$$

Therefore, the class of encoder-decoder convolutional neural networks follows:

$$G_\theta(y) := \sigma_d \left( \text{op}_d W_d \sigma_{d-1} \left( \text{op}_{d-1} W_{d-1} \ldots \sigma_1 \left( \text{op}_1 W_1 \cdot y \right) \ldots \right) \right).$$

The family of general convolutional neural networks is mostly a combination of fully-connected layers and convolutional layers, together with activation and pooling functions. Hence we use $(W_1, \ldots, W_d)$ as either the weight matrices of layers in fully-connected neural networks, or the convolution operator with weights of convolutional layers, and obtain a unified formulation of DNNs in the regime of FWI:

$$G_\theta(y) := \sigma_d \left( W_d \sigma_{d-1} \left( W_{d-1} \ldots \sigma_1 \left( W_1 \cdot y \right) \ldots \right) \right).$$

### 3.3 Definitions

**Definition 3.1.** (Lipschitz Continuity) A real-valued function $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$ is $L$-lipshitz if there exists a real constant $L > 0$, s.t. $\forall x_1, x_2 \in \mathbb{R}^m$,

$$|f(x_1) - f(x_2)| \leq L|x_1 - x_2|.$$

**Definition 3.2.** (Covering number) Let $(X, \rho)$ be a metric space and $H \subseteq X$, we say that $G$ is $\epsilon$ – cover of $H$ with respect to $\rho$ if every $h \in H$ has a $g \in G$ such that $\rho(g, h) \leq \epsilon$. Then $\mathcal{N}_\rho(H, \epsilon)$ denotes the size of smallest $\epsilon$ – cover of $H$ w.r.t. $\rho$.

**Definition 3.3.** (Jacobian matrix) Consider a $d$-layer fully-connected neural network parameterized by $\theta$ with 1-lipshitz activation functions $\phi(\cdot) : \hat{x} = G_\theta(y) = W_d \phi(\ldots \phi(W_1^T y + b_1) \ldots) + b_d$, the Jacobian matrix of the neural network $G_\theta$ is given as:

$$J = \frac{d \hat{x}}{dy} = \begin{bmatrix}
\frac{\partial \hat{x}_1}{\partial y_1} & \cdots & \frac{\partial \hat{x}_1}{\partial y_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial \hat{x}_N}{\partial y_1} & \cdots & \frac{\partial \hat{x}_N}{\partial y_n}
\end{bmatrix}. $$
Owing to the chain rule of derivatives, $J$ can be decomposed into the product of layer-wise Jacobian matrices \( i.e. J = \prod_{i=1}^{d} J_i \).

**Definition 3.4.** (K – ε(T) robustness) Let \( T \) be the training set of \( N \) entries and \( D \) be the sample space. A learning algorithm \( G \) is said to be \( K – \varepsilon(T) \) robustness if \( D \) can be partitioned into \( K \) disjoint sets \( K_k, (k = 1, \ldots, K) \), such that for any \((x_i, y_i) \in T \) and \((x, y) \in D\):

\[
(x_i, y_i), (x, y) \in K_k \implies |L(G(y_i), x_i) - L(G(y), x)| \leq \varepsilon(T).
\]  

\( 6 \)

## 4 Robustness Against Perturbation

In this section, we study how the performance of deep neural networks might be impacted when tested with noisy data. The gravity of this issue increases for FWI as the test data collected from the real world is inevitably contaminated with noise. We consider DNNs trained with MAE loss and MSE loss and analyze how the model robustness differs from each other. By the end of this section, we conclude that as the noise level increases, DNNs trained with MAE loss show better robustness than DNNs trained with MSE loss.

### 4.1 Setup

Throughout the analysis, we make merely one assumption about the level of noise is bounded. In other words, the random noise \( n \) satisfies: \( ||n||_2 \leq \eta \).

Now let’s consider a DNN \( G_\theta \) which learns a mapping: \( \mathbb{R}^p \rightarrow \mathbb{R}^q \), its robustness indicates how significant the performance drop will be if the test data is imposed with random noise \( n \). Denoting \( L(\cdot) : \mathbb{R}^q \rightarrow \mathbb{R} \) as the loss function adopted, we are interested in the upper bound of \( |L(G_\theta(x+n)) - L(G_\theta(x))| \).

### 4.2 Upper Bound of DNNs Robustness with MAE

As an essential ingredient of the optimization process, the loss functions commonly adopted for image-to-image analysis include Mean Absolute Error (MAE) or (Rooted) Mean Square Loss (MSE/RMSE) [64], with the exponential term, MSE implicitly gives a higher weight on large errors, thus becoming more sensitive to marginal predictions caused by perturbations on the input data. Inspired by such an inductive bias, we first consider MAE loss as the objective function of the DNN and prove that the gain of the loss could be upper-bounded, thus indicating certain level of robustness can be guaranteed. The following theorem states our main result:

**Theorem 4.1.** (Robustness) For a objective function \( g = f \circ L : \mathbb{R}^p \rightarrow \mathbb{R} \), where \( f \) denotes a neural network with \( d \) layers that learns a mapping \( f : \mathbb{R}^p \rightarrow \mathbb{R}^q \) and \( L : \mathbb{R}^q \rightarrow \mathbb{R} \) is loss function. If the input \( x \in \mathbb{R}^p \) is perturbed data with an arbitrary noise \( n \ (||n||_2 \leq \eta) \), then the following holds if \( L \) is MAE loss:

\[
RB_{MAE} = |g(x+n) - g(x)| \leq \prod_{i=1}^{d} ||W_i||_F \cdot \eta,
\]

where \( W_i \) is the weight matrix of the \( i \)-th layer in the DNNs as defined in section 3.

Before proving theorem 4.1, we introduce the following lemma:

**Lemma 4.2.** Consider a function \( f : \mathbb{R}^d \rightarrow \mathbb{R} \) that satisfies the Lipschitz continuity, we have:

\[
|f(x) - f(y)| \leq L_p ||x - y||_q,
\]

where \( \frac{1}{p} + \frac{1}{q} = 1, (1 < p, q < \infty), L_p = \sup \{ ||f(x)||_p : x \in \mathbb{R}^d \} \).

**Proof.**

\[
|f(x) - f(y)| = |f(x_1, x_2, \ldots, x_n) - f(y_1, y_2, \ldots, y_n)|,
\]

\[
= |f(x_1, x_2, \ldots, x_n) - f(y_1, x_2, \ldots, x_n) + f(y_1, x_2, \ldots, x_n) - f(y_1, y_2, \ldots, y_n)|,
\]

\[
\leq |f(x_1, x_2, \ldots, x_n) - f(y_1, x_2, \ldots, x_n)| + |f(y_1, x_2, \ldots, x_n) - f(y_1, y_2, \ldots, y_n)|,
\]

\[
= |f(x_1, x_2, \ldots, x_n) - f(y_1, x_2, \ldots, x_n)| \cdot |x_1 - y_1| + |f(y_1, x_2, \ldots, x_n) - f(y_1, y_2, \ldots, y_n)|.
\]
From the middle values theorem, there exists a constant $c_1$ such that the first term equals: $|f'_{x_i}(c_1)| \cdot |x_1 - y_1|$. Now we focus on the second term, with triangular inequality and middle values theorem again:

$$|f(y_1, x_2, \ldots, x_n) - f(y_1, y_2, \ldots, y_n)| \leq |f(y_1, x_2, \ldots, x_n) - f(y_1, y_2, x_3, \ldots, x_n)| + |f(y_1, y_2, x_3, \ldots, x_n) - f(y_1, y_2, \ldots, y_n)|,$$

$$= \left|\frac{f(y_1, x_2, \ldots, x_n) - f(y_1, y_2, x_3, \ldots, x_n)}{|x_2 - y_2|}\right| \cdot |x_2 - y_2| + |f(y_1, y_2, x_3, \ldots, x_n) - f(y_1, y_2, \ldots, y_n)|,$$

$$= |f'_{x_2}(c_2)| \cdot |x_2 - y_2| + |f(y_1, y_2, x_3, \ldots, x_n) - f(y_1, y_2, \ldots, y_n)|.$$

So on and so forth, by substituting $x_i$ with $y_i$ on the second term iteratively, and summing all the terms by middle values theorem, we obtain:

$$|f(x) - f(y)| \leq \sum_{i=1}^{n} |f'_{x_i}(c_i)| \cdot |x_i - y_i|. \quad (7)$$

Now recall the Hölder’s inequality:

**Fact 3. (Hölder’s Inequality)**

$$\sum_{i=1}^{n} |a_i b_i| \leq \left(\sum_{i=1}^{n} |a_i|^p\right)^{\frac{1}{p}} \cdot \left(\sum_{i=1}^{n} |b_i|^q\right)^{\frac{1}{q}},$$

where $p, q > 1$ and $\frac{1}{p} + \frac{1}{q} = 1$, equality holds when $|b_i| = c|a_i|^{p-1}$.

Equation 7 can be extended as:

$$|f(x) - f(y)| \leq \sum_{i=1}^{n} |f'_{x_i}(c_i)| \cdot |x_i - y_i|,$$

$$\leq \left(\sum_{i=1}^{n} |f'_{x_i}(c_i)|^p\right)^{\frac{1}{p}} \cdot \left(\sum_{i=1}^{n} |x_i - y_i|^q\right)^{\frac{1}{q}},$$

$$= \|\nabla f(c)\|_p \cdot ||x - y||_q = L_p \cdot ||x - y||_q,$$

where $c = (c_1, c_2, \ldots, c_n) \in \mathbb{R}^d$.

Lemma 4.2 requires that $f$ must be Lipschitz-continuous, which characterized the predictions of our deep neural networks are bounded with respect to the input. Recent progress [65, 66] has shown that DNNs with Relu activations are indeed Lipchitz-continuous, we employ this evidence as the last piece to complete the puzzle, and now we are ready to present the proof of theorem 4.1.

**Proof.** Recall that the loss function $\mathcal{L}$ is specified as MAE loss, which indicates $\mathcal{L}(\cdot) = ||\cdot||_1$.

$$|g(x + n) - g(x)| = ||f(x + n)|_1 - ||f(x)||_1|$$

$$\leq |f(x + n) - f(x)|,$$

(triangular inequality)

$$\leq L_2 \cdot ||x - y||_2,$$

(lemma 4.2)

$$\leq \prod_{i=1}^{d} |J_i| \cdot \eta,$$

$$\leq \prod_{i=1}^{d} ||W_i||_{F} \cdot \eta,$$

which completes the proof.

An immediate observation on theorem 4.1 is that the performance degradation is bounded by a constant. If a pre-trained model is tested with noisy data, $||J_i||$ or $||W_i||_{F}$ is fixed and the performance depends entirely on the noise level. While as argued in [12], the additive noise is considered as Gaussian noise at a low level in most scenarios, thus leading to a significant performance degradation with low probability.
4.3 Upper Bound of DNNs Robustness with MSE

It can be observed that the technique above is not applicable to the proof of MSE loss. Let \( \mathcal{L}(x, \hat{x}) \) denote the loss of the predicted values and the ground truth, it is trivial that \( |\mathcal{L}(x_1, \hat{x}) - \mathcal{L}(x_2, \hat{x})| \leq \mathcal{L}(x_1, x_2) \) when \( \mathcal{L} \) stands for MAE loss, namely \(|·|_1\). Equipped with such a fact, we can apply lemma 4.2. However this requirement cannot be satisfied by MSE loss. On the contrary, we can show that \( |\mathcal{L}(x_1, \hat{x}) - \mathcal{L}(x_2, \hat{x})| > \mathcal{L}(x_1, x_2) \) for certain values of \( x \). We formalize the fact for MSE loss as fact 4 and provide a proof in the appendix.

Fact 4. For \( \forall x_1, x_2 \in \mathbb{R}^d \), and \( \mathcal{L}(x, \hat{x}) = ||x - \hat{x}||_2 \), \( \mathcal{L}(x_1, \hat{x}) - \mathcal{L}(x_2, \hat{x})| \leq \mathcal{L}(x_1, x_2) \) does not hold for \( \forall x \in \mathbb{R}^d \)

However, there is a simple approach to upper bound the gain of MSE loss based on theorem 4.1, whilst implying that \( R^{\text{MSE}} \) is looser than \( R^{\text{MAE}} \). An ingredient here is that a neural network is lipschitz continuous with any common activation function (ReLU, tanh, sigmoid, etc.) [65, 66]. The bound is given in the following corollary 4.2.1

Corollary 4.2.1. For a objective function \( g = f \circ \mathcal{L} : \mathbb{R}^p \rightarrow \mathbb{R} \), where \( f \) denotes a neural network with \( d \) layers that learns a mapping \( f : \mathbb{R}^p \rightarrow \mathbb{R}^q \) and \( \mathcal{L} : \mathbb{R}^q \rightarrow \mathbb{R} \) is loss function. If the input \( x \in \mathbb{R}^p \) is perturbed with an arbitrary noise \( n (||n||_2 \leq \eta) \), then the following holds if \( \mathcal{L} \) is MSE loss:

\[
R^{\text{MSE}} \leq L \cdot \frac{\eta}{\sqrt{a_{in}}} \cdot (R^{\text{MAE}} + 2\alpha),
\]

where \( L \) is the Lipschitz constant of the neural network, \( a_{in} \) is the dimension of input data, and \( \alpha \) denotes that maximum of test loss on clean data, all of which are considered as a constant.

Proof.

\[
R^{\text{MSE}} = |g(x + n) - g(x)|,
\]

\[
= \left| ||f(x + n) - y||_2 - ||f(x) - y||_2 \right|,
\]

\[
\leq |f(x + n) + f(x)| \cdot |f(x + n) - f(x)| - 2\eta \cdot |f(x + n) - f(x)|,
\]

\[
= |f(x + n) - f(x)| \cdot |f(x + n) + f(x) - 2\eta|.
\]

Suppose \( L \) is the Lipschitz constant of the neural network, following the definition we have: \( |f(x + n) - f(x)| \leq L \cdot |n| \), let \( \alpha \) be the max test loss with clean data, namely \( \max (|f(x) - y|) \)

\[
R^{\text{MSE}} \leq L \cdot |n| \cdot (|f(x + n) - y| + |f(x) - y|),
\]

\[
\leq L \cdot |n| \cdot (R^{\text{MSE}} + 2|f(x) - y|),
\]

\[
\leq L \cdot \frac{\eta}{\sqrt{a_{in}}} \cdot (R^{\text{MSE}} + 2\alpha),
\]

(Cauchy-schwartz Inequality)

concluding the proof.

We would like to remark that the Lipschitz constant \( L \), although intractable, has been shown to yield a close estimation for both Fully-connected Neural networks [66, 65, 67, 68] and Convolutional Neural Networks [69, 70, 50]. Notably, [65] proposed an estimation \( L \approx \sqrt{\rho} \) where \( \rho \) can be obtained via semi-definite programming.

4.4 DNNs with MAE Are More Robust with Significant Noise

Notice that although the bound of \( R^{\text{MAE}} \) is tighter than that of \( R^{\text{MSE}} \), we are still not close to the conclusion that DNNs with MAE is more robust than DNNs with MSE. First of all, the bounds are deterministic, so that no claim on an instance can be made. This is trivial if we consider an example that given \( x \leq 100, y \leq 1000 \), we may not conclude \( x \leq y \). Secondly, our proof of theorem 5.1 and corollary 4.2.1 is the worst-case analysis, therefore obstructing probabilistic approaches.

To address this issue, we adopt the strategy of lower-bounding, \( R^{\text{MSE}} \). The insight is that it can only be concluded a bound induced by \( R^{\text{MAE}} \) is tighter than a bound induced by \( R^{\text{MSE}} \) if \( LB(R^{\text{MSE}}) \geq UB(R^{\text{MAE}}) \) where \( UB \) stands for upper bound and \( LB \) for lower bound. In the following corollary we show that \( LB(R^{\text{MSE}}) \geq UB(R^{\text{MAE}}) \) holds under certain conditions.

Corollary 4.2.2. \( R^{\text{MSE}} \geq R^{\text{MAE}} \), if \( |f(x + n) + f(x) - 2y| \geq 1 \).

Proof. For \( R^{\text{MAE}} \), we have \( |g(x + n) - g(x)| \leq |f(x + n) - f(x)| \) from the proof of theorem 4.1; For \( R^{\text{MSE}} \), let \( g' \) denote the function of \( f \circ \mathcal{L} \) when \( \mathcal{L} = ||·||_2 \), then \( |g'(x + n) - g'(x)| \geq |g'(x + n) - g'(x)| = (f(x + n) + f(x) - 2y) \cdot (f(x + n) - f(x)) \). Now it is clear to see that \( |f(x + n) + f(x) - 2y| \geq 1 \) must hold for \( R^{\text{MSE}} \geq R^{\text{MAE}} \).
From corollary 4.2.2, it is straightforward that \(|f(x + n) - y| + |f(x) - y| \geq 1\) must hold for \(RB_{\text{MSE}} \geq RB_{\text{MAE}}\). There can be two scenarios: (1) The learning process has not converged, hence the predicted results will impose considerably large value of loss, namely both \(|f(x + n) - y|\) and \(|f(x) - y|\) can be large; (2) The training has already converged, then the test loss on clean data \(|f(x) - y|\) is relatively small, as the additive noise \(n\) increases, \(|f(x + n) - y|\) also becomes incremental. Apparently, scenario (2) fits our context. Therefore, we are able to make a rigorous claim that DNNs with MSE become less robust than DNNs with MAE as the level of additive noise becomes more significant.

5 Generalization Error Bound

Our analysis of generalization is established on the robustness and generalization framework [27]. Notice that the concept of robustness here deviates from section 4, as the robustness in [27] is a property of learning algorithms applied to the entire data space and ours considers the testing phase.

5.1 Setup

In the context of supervised deep learning, training data \(S_{\text{train}} = \{(x, y)|x \sim X, y \sim Y\}\) contains a number of \(N\) i.i.d samples from a sample space \(S = X \times Y\) where \(X \subset \mathbb{R}^p\) and \(Y \subset \mathbb{R}^q\). Hence, in the training phase, an average training error can be obtained as \(E_{(x,y) \sim S_{\text{train}}} [\tilde{L}(G_{\theta})]\) where \(\tilde{L}(\cdot) = \frac{1}{N} \sum_{i=1}^N l(G(y_i), x_i)\). Similarly, by denoting the test error as \(E_{(x,y) \sim S}[\tilde{L}(G_{\theta})]\), the generalization error can be formally defined as \(Err_g = |E_{(x,y) \sim S_{\text{train}}} [\tilde{L}(G_{\theta})] - E_{(x,y) \sim S}[\tilde{L}(G_{\theta})]|\). Next, we will provide and further demonstrate an upper bound on \(Err_g\).

5.2 Main Result and Proof

To begin with, we present the main result by theorem 5.1.

**Theorem 5.1. (Generalization)** For spaces \(X\) and \(Y\) equipped with \(l_2\) norm, and sample space \(D = X \times Y\), with the assumption that the forward operator \(F(\cdot)\) is \(L\)-Lipschitz, a \(d\)-layer DNN. \(G_{\theta}(\cdot)\) learns a mapping: \(Y \rightarrow X\) trained on the training set \(T\) containing \(N\) i.i.d samples. Then with probability \(1 - \varepsilon\), the generalization error of \(G_{\theta}\) is upper-bounded by:

\[
Err_{G_{\theta}} \leq (1 + \prod_{i=1}^d ||W_i||_{\ell_2})(L\delta + 2\eta) + M \sqrt{\frac{2N(\frac{\delta}{2}; X, l_2) \ln 2 + \ln(\frac{1}{\varepsilon})}{N}},
\]

where \(L\) is the Lipschitz constant of the forward modelling, \(\delta, \varepsilon, \eta\) are small constants, \(M\) is also a constant representing the maximum training loss.

Recall definition 3.2 and definition 3.4, our first step is to prove that DNNs satisfy the \(K - \epsilon(T)\) robustness with respect to the covering number of the data sample space as follows:

**Theorem 5.2.** Denote \((S, \rho)\) as a metric space where \(S = X \times Y\) is the sample space of the dataset, where \(X \subset \mathbb{R}^p\) and \(Y \subset \mathbb{R}^q\). Suppose a deep neural network which learns a mapping \(f : Y \rightarrow X\) is trained on a subset \(S_{\text{train}} \subset S\), it follows that

the deep neural network is \(\mathcal{N}(\frac{\delta}{2}; S, \rho), (1 + \prod_{i=1}^d ||J_i||_{2,2})\delta\)-robust,

where \(|| \cdot ||_{2,2}\) denotes spectral norm, and \(\mathcal{N}(\frac{\delta}{2}; S, \rho)\) denotes the covering number of the metric space \((S, \rho)\) with \(\frac{\delta}{2}\) as the radius of the metric ball.

**Proof.** The proof of theorem 5.2 is an immediate result of the lemma below.

**Lemma 5.3.** Suppose a deep neural network learns a inverse mapping \(f : Y \rightarrow X\), and \(y_1, y_2 \in Y\), the following holds:

\[
||G_{\theta}(y_1) - G_{\theta}(y_2)||_2 \leq \prod_{i=1}^d ||J_i||_{2,2} ||y_1 - y_2||_2.
\]

To keep the consistency, we provide the proof of lemma 5.3 in appendix A. We continue with the proof of theorem 5.2 provided that lemma 5.3 is correct.
Supposing that \( s_1 = (x_1, y_1), s_2 = (x_2, y_2) \) are two samples from \( S \) and \( s_1 \in S_{train} \), we notice that it suffices to prove the following:

\[
|\mathcal{L}(G_\theta(y_1), x_1) - \mathcal{L}(G_\theta(y_2), x_2)| \leq (1 + \prod_{i=1}^{d} ||J_i||_{2,2}) \cdot \delta,
\]

for a \( \frac{\delta}{2} \)-cover of \( S \) and \( \rho(s_1, s_2) \leq \delta \).

We adopt \( l_2 \) metric, and it follows that:

\[
\begin{align*}
|l(G_\theta(y_1), x_1) - l(G_\theta(y_2), x_2)| &= ||x_1 - G_\theta(y_1)||_2 - ||x_2 - G_\theta(y_2)||_2, \\
&\leq ||x_1 - G_\theta(y_1) - x_2 + G_\theta(y_2)||_2, \quad \text{(triangular inequality)} \\
&\leq ||x_1 - x_2||_2 + ||G_\theta(y_1) - G_\theta(y_2)||_2, \quad \text{(Minkowski inequality)} \\
&\leq ||x_1 - x_2||_2 + \prod_{i=1}^{d} ||J_i||_{2,2} ||y_1 - y_2||_2, \quad \text{(lemma 5.3)} \\
&\leq \left(1 + \prod_{i=1}^{d} ||J_i||_{2,2}\right) \cdot \rho(s_1, s_2), \\
&\leq \left(1 + \prod_{i=1}^{d} ||J_i||_{2,2}\right) \cdot \delta,
\end{align*}
\]

which completes the proof. \( \square \)

The robustness and generalization framework proposed by [27] provides an approach to quantify the generalization error given that the algorithm satisfies the robustness defined by 3.4.

**Lemma 5.4.** (Adopted from Theorem 3 in [27]) If \( S \) consists of \( n \) i.i.d. samples, and \( A \) is \((K, \epsilon(S))\)-robust, then for any \( \delta > 0 \), with probability at least \( 1 - \delta \),

\[
|\hat{l}(A_s) - l(A_s)| \leq \epsilon(S) + M \sqrt{\frac{2K \ln 2 + 2 \ln(1/\delta)}{n}},
\]

where \( M \) is the maximum training loss.

Since we already show that \( G_\theta \) is \((N(\frac{\delta}{2}; S, \rho), (1 + \prod_{i=1}^{d} ||J_i||_{2,2})\delta)\)-robust, by following lemma 5.4, we conclude that:

\[
|\hat{l}(G_\theta) - l(G_\theta)| \leq \left(1 + \prod_{i=1}^{d} ||J_i||_{2,2}\right) \delta + M \sqrt{\frac{2N(\frac{\delta}{2}; S, \rho) \ln 2 + 2 \ln(1/\delta)}{n}}.
\]

By incorporating the forward modeling of FWI, specifically, we embed the \( L - \) lipschitz continuity and the upper bound of the noise imposed on the observed data.

**Lemma 5.5.** Suppose \( X \subset \mathbb{R}^p \) and \( Y \subset \mathbb{R}^q \), the forward mapping \( f : X \to Y \) is \( L \) - lipschitz. Denoting \( S = X \times Y \) as the sample space equipped with metric \( \rho \), we have:

\[
N\left(\frac{L \delta + 2\eta}{2}; S, \rho\right) \leq N\left(\frac{\delta}{2}; X, l_2\right).
\]

**Proof.** Let \( X' \) be a \( \delta \)-cover of \( X \), then \( \forall x \in X, \exists x' \in X' \) s.t. \( ||x - x'||_2 \leq \delta \).

Let \( Y' := \{ y' = f(x') + n, x' \in X', ||n'||_2  \leq \eta \} \), and similarly \( Y := \{ y = f(x) + n, x \in X, ||n||_2  \leq \eta \} \). It suffices to show that \( S' = X' \times Y' \) is a \((L \delta + 2\eta)\)-cover of \( S \).

For any \( y \in Y \) and \( y' \in Y' \), we have:

\[
||y - y'|| \leq ||f(x) + n - f(x') - n'||_2 \\
\leq ||f(x) - f(x')||_2 + ||n - n'||_2 \\
\leq L ||x - x'||_2 + 2\eta \\
\leq L \delta + 2\eta
\]
| Dataset            | Train Size | Test Size | Data Shape      | Label Shape |
|--------------------|------------|-----------|------------------|-------------|
| Kimberlina CO₂     | 15000      | 4430      | 9×1251×101       | 401×141     |
| Kimberlina Reservoir | 1200      | 337       | 6×1001×200       | 601×351     |
| CurvedVel          | 60000      | 7000      | 6×200×200        | 200×200     |

Table 2: Summary of the Datasets

This indicates that $Y'$ is an $(L\delta + 2\eta)$-cover of Y. Now consider $\forall s \in S$ and $s' \in S'$, and recall the definition of metric $\rho$, it follows that:

$$\rho(s, s') = \max(||x - x'||_2, ||y - y'||_2) \leq \max(L\delta + 2\eta) = L\delta + 2\eta$$

Therefore $S'$ is a $(L\delta + 2\eta)$-cover of $S$, concluding the proof.

6 Experiments

In this section, we show experimental results to provide interpretation on the proved bounds. Following the common experimental settings in the previous work[3, 4], we introduce three standard seismic FWI datasets published in OpenFWI [71]: Kimberlina CO₂, Kimberlina Reservoir and the CurvedVel datasets. Both Kimberlina CO₂ and Reservoir datasets are created by the U.S. Department of Energy (DOE). Specifically, the Kimberlina CO₂ dataset is to study and simulate CO₂ leakage through the well bore in the shallow layers [7], while CO₂ Reservoir dataset focuses on the migration of super-critical CO₂ after injecting into the storage reservoir [8]. We provide a visualization of a few samples and their corresponding shot-gather seismic data (figure 5 and figure 6) in Appendix D. CurvedVel dataset was generated to simulate curved subsurface layers with geologic fault zones and varying central frequencies. We provide a visualization of several seismic shot-gathers generated with different central frequencies (figure 7) and a few velocity maps selected from CurvedVel dataset with different geologic faults (figure 8) in Appendix E. We use the two Kimberlina datasets for robustness test and the CurvedVel dataset for generalization test. The statistics of three datasets are summarized in table 2. “Data” refers to the input seismic data, and “label” stands for the output velocity maps. As for the choice of Deep Neural Network, we implement an encoder-decoder based Convolutional Neural Network following the InversionNet [3]. A general model architecture is given in figure 2. Note that the architecture and parameters vary with different datasets, and we provide a list of all details for each task in the supplementary materials.

6.1 Robustness Test

As theorem 4.1 indicates, we target the upper bound of the deviation between the prediction with clean data and perturbed data, namely the performance drop, as the robustness of the DNN model. Recall that corollary 4.2.2 states that the DNNs with MSE loss has a looser generalization bound than DNNs with MAE loss. Therefore, we trained the InversionNet shown in figure 2 on two Kimberlina Datasets (CO₂ and Reservoir) with MAE loss and MSE loss, respectively. A detailed description and illustration of these two datasets are given in the Appendix. In the test phase, we impose noise to the test data with different levels of signal-to-noise ratios (SNR={inf, 30, 20, 10, 0}, where inf stands for no noise).
Figure 3 illustrates two types of prediction loss for all noise levels on two datasets mentioned above. It can be observed that when SNR = \text{inf}, the value of MSE loss is asymptotically the square of MAE loss, which means that the predicted images show similarly good quality. As the noise level increases, the value of MSE loss surges and surpasses the value of MAE loss when SNR is around 10.

It may be argued that the value of loss does not absolutely capture the real quality of the generated images. To tackle with this issue, we leveraged the well received structural similarity index [72] as a reference, which gives a perceptual evaluation on the similarity between the predicted and ground-truth images. The value of SSIM is normalized into \([0, 1]\), where a larger value indicates a higher level of similarity. Table 3 gives a detailed demonstration of the robustness test with percentage of loss gain and the SSIM index. It can be observed that as the noise level increases, the percentage of loss gain differs vastly between two types of DNNs. The SSIM value of DNNs with MSE quickly drops close to 0.8 and the other remains around 0.9, also implying that the DNNs with MAE loss are able to produce velocity maps with relatively good quality. These experimental results corroborate with our theoretical results in theorem 4.1 and corollary 4.2.2.

**6.2 Generalization Test**

Recall that our results in theorem 5.1 indicates that the generalization bounds can be characterized by: the maximum training loss \(L_m\), training dataset size \(N\), and the product of weight matrices norm, denoted as \(W_0 = \prod_{i=1}^{d} ||W_i||_F\). In this section, we illustrate substantial empirical results of the interplay between the generalization gap and those parameters. Notice that in practice, the number of training samples often yield a direct impact on the model performance, hence we put together \(\frac{L_m}{\sqrt{N}}\) as one parameter of the generalization bounds in theorem 5.1. To accommodate the training dataset size, we choose the CurvedVel dataset which contains 60,000 images for training and 7,000 for testing.

We first study the empirical generalization behavior in a standard setting: data samples are independent and identically distributed (\(i.i.d\)) in both training and testing sets, where the generalization gap \(Err_G\) is the difference of training and testing performances. We train a similar encoder-decoder Convolutional Neural Network as figure 2 with three different optimizers: SGD, AdaGrad and AdamW. The results illustrated in table 4 and table 5 are the average of all trained
models. We can observe from table 4 and table 5 that both of the parameters: $\frac{\ell_w}{\sqrt{N}}$ and $W_0$ show a positive relationship with the generalization gap.

It is necessary to remark that in experiments for table 4, we are not able to fix another important factor $W_0$. Although a number of works have shown that the parameters deviate very little in large-scale neural networks upon convergence [32], it is still not rigorous to ignore the impact of $W_0$. As shown in table 4, we only claim that such a positive relationship can be observed. In experiments for table 5, we fix $N$ and change the neural network architecture so that more dynamics of $W_0$ can be brought in.

### 6.2.1 Generalization Test with Distribution Drift

The issue of distribution drift on testing data emerges as a common hurdle in the field study of FWI, as the model is usually trained on synthetic data which fixes some geological features (such as source location, central frequency, etc). Inspired by such a practical concern, we also conduct an ablation study to show how our generalization bounds work in those scenarios. Specifically, we consider the distribution drift with the change of two geological features: number of geologic faults and the central frequency. We provide samples of the velocity maps and seismic data with the various geological features mentioned above in the Appendix.

The setup here is that we use the models trained with original dataset, which is 1-fault and generated with the central frequency $f = 15$ Hz by default, but test them with data has multiple-fault or generated with different central frequencies. For simplification, we combine the two parameters in table 4 and table 5 as $\frac{\ell_w}{\sqrt{N}} \cdot \prod_{i=1}^{d} ||W_i||_F$ and show the development of generalization gap $Err_G$. table 6 and table 7 demonstrate the experimental results on test data with two geological features, respectively.

An immediate observation from table 6 is that for each model, the generalization gap $Err_G$ increases as the number of fault rises. This is due to the fact that as the increasing of the number of the geological faults in test data indicates a gain of distribution drift. If we focus on each column of multi-fault, as $\frac{\ell_w}{\sqrt{N}} \cdot \prod_{i=1}^{d} ||W_i||_F$ increases from $7.5e+8$ to $2.2e+9$, the generalization gap keeps increasing except for the last two rows, whose results are very close, thus leading to the conclusion as table 5 does for 1-fault, which is the default test data.

Table 7 presents the generalization error for test data generated with different central frequencies among $f = \{15, 20, 25\}$ Hz. Similarly, the more deviated from central frequency of 15 Hz, the larger generalization gap can be observed. Again, fixing each column of a specific central frequency, a positive relationship between $\frac{\ell_w}{\sqrt{N}} \cdot \prod_{i=1}^{d} ||W_i||_F$ and the generalization gap still holds as discussed in table 5 and table 6.

| Parameter | 1-fault | 2-fault | 3-fault | 4-fault |
|-----------|---------|---------|---------|---------|
| $\frac{\ell_w}{\sqrt{N}} \cdot \prod_{i=1}^{d} ||W_i||_F = 8.5e+8$ | 0.02856 | 0.03477 | 0.06092 | 0.08848 |
| $\frac{\ell_w}{\sqrt{N}} \cdot \prod_{i=1}^{d} ||W_i||_F = 1.0e+9$ | 0.03225 | 0.04058 | 0.06709 | 0.09362 |
| $\frac{\ell_w}{\sqrt{N}} \cdot \prod_{i=1}^{d} ||W_i||_F = 1.4e+9$ | 0.03417 | 0.05242 | 0.07481 | 0.09701 |
| $\frac{\ell_w}{\sqrt{N}} \cdot \prod_{i=1}^{d} ||W_i||_F = 2.1e+9$ | 0.03538 | 0.05595 | 0.08271 | 0.1076 |
| $\frac{\ell_w}{\sqrt{N}} \cdot \prod_{i=1}^{d} ||W_i||_F = 2.2e+9$ | 0.03469 | 0.05136 | 0.07891 | 0.1059 |

Table 6: Generalization Gap with Different Number of Faults
Table 7: Generalization Gap with Different Wave Frequencies

| Parameter | f=15Hz | f=20Hz | f=25Hz |
|-----------|--------|--------|--------|
| $L_N \cdot \prod_{i=1}^{d} ||W_i||_F = 8.5e + 8$ | 0.02856 | 0.1799 | 0.3197 |
| $L_N \cdot \prod_{i=1}^{d} ||W_i||_F = 1.0e + 9$ | 0.03225 | 0.2020 | 0.3459 |
| $L_N \cdot \prod_{i=1}^{d} ||W_i||_F = 1.4e + 9$ | 0.03417 | 0.2336 | 0.3671 |
| $L_N \cdot \prod_{i=1}^{d} ||W_i||_F = 2.1e + 9$ | 0.03538 | 0.2907 | 0.3852 |
| $L_N \cdot \prod_{i=1}^{d} ||W_i||_F = 2.2e + 9$ | 0.03469 | 0.2818 | 0.3955 |

7 Conclusion

In this paper, motivated by the recent progress of utilizing deep neural networks for Full Waveform Inversion, we study from the theoretical perspectives on how these model perform against perturbation and generalize in practical scenarios. We consider the performance degradation when tested with perturbed data by upper-bounding the gain of loss, and prove that DNNs trained with MAE loss are more robust than DNNs trained with MSE loss as the noise increases. We also prove an upper bound for the generalization gap based on the norm of weight matrices. Experimental results are illustrated with standard FWI datasets on both topics, corroborating with our theoretical results. We further remark that the analysis can be extended to other deep learning driven inverse problems, paving the way of a better understanding of the exploit of deep learning in scientific domains.

Acknowledgment

This work was supported by the Center for Space and Earth Science at Los Alamos National Laboratory (LANL), and by the Laboratory Directed Research and Development program under the project number 20210542MFR at LANL.
A Proof of fact 4

Fact 5. For \( \forall x_1, x_2 \in \mathbb{R}^d \), and \( \mathcal{L}(x, \hat{x}) = ||x - \hat{x}||_2 \), \( |\mathcal{L}(x_1, \hat{x}) - \mathcal{L}(x_2, \hat{x})| \leq \mathcal{L}(x_1, x_2) \) does not hold for \( \forall x \in \mathbb{R}^d \).

Proof. Let \( x = 2x_1 \). We will have

\[
|\mathcal{L}(x_1, \hat{x}) - \mathcal{L}(x_2, \hat{x})| = |||x_1 - x||_2^2 - ||x_2 - x||_2^2|, \\
= |||x_1||_2^2 - 2x_1^T x - ||x_2||_2^2 + 2x_2^T x|, \\
= 3||x_1||_2^2 - 4x_1^T x_2 + ||x_2||_2^2|.
\]

Notice that \( \mathcal{L}(x_1, x_2) = ||x_1 - x_2||_2^2 = ||x_1||_2^2 - 2x_1^T x_2 + ||x_2||_2^2 \). Without loss of generality, we further assume \( ||x_1||_2^2 > ||x_2||_2^2 \), and it follows that:

\[
\mathcal{L}(x_1, x_2) - |\mathcal{L}(x_1, \hat{x}) - \mathcal{L}(x_2, \hat{x})| \\
= 2||x_1||_2^2 - 2x_1^T x_2, \\
\geq ||x_1||_2^2 - 2x_1^T x_2 + ||x_2||_2^2 \geq 0,
\]

which completes the proof. \( \square \)

B Example of fact 1

Supposing the convolution filter \( c \) has dimension \( d_c = 2 \) and the input image \( x \in \mathbb{R}^{3 \times 3} \), we will have:

\[
w = \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \end{bmatrix}, \quad x = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,1} & x_{3,2} & x_{3,3} \end{bmatrix}.
\]

For the sake of simplicity, let padding and stride both be 0, then a standard convolution operation follows:

\[
\text{conv}(x, W) = \begin{bmatrix} \sum_{i=1}^2 \sum_{j=1}^2 2w_{i,j}x_{i,j} \\ \sum_{i=1}^2 \sum_{j=1}^2 2w_{i,j}x_{i,j} \end{bmatrix}.
\]

If we reshape \( x \) to dimension 1 such that \( x \in \mathbb{R}^{9 \times 1} \), apparently \( \text{conv}(x, W) = \text{op} \cdot x \), where \( \text{op} \in \mathbb{R}^{4 \times 9} \), given explicitly as

\[
\text{op} = \begin{bmatrix} w_{1,1} & w_{1,2} & 0 & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\ 0 & w_{1,1} & w_{1,2} & 0 & w_{2,1} & w_{2,2} & 0 & 0 & 0 \\ 0 & 0 & 0 & w_{1,1} & w_{1,2} & 0 & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & w_{1,1} & w_{1,2} & 0 & w_{2,1} & w_{2,2} \end{bmatrix}.
\]

C Example of fact 2

We illustrate the process of convolution and transposed convolution with 1D signals. As a general setting, the convolution filter with dimension 2 down-samples \( X = [x_1, \ldots, x_9] \) to \( Y = [y_1, \ldots, y_5] \), with both stride and padding equals 2. The weight of the filter is \( w_i, i = \{1, 2, 3, 4\} \). The transposed convolution, also termed as de-convolution, takes \( Y \) as input and reconstruct \( X \) through up-sampling. figure 4 provides an explicit illustration of both operations. For fact 2, all we want to argue is that there also exists a matrix \( \text{op} \) such that the output of a transposed convolutional layer is also a linear transformation of the input features. This is obvious from figure 4.

D The Kimberlina CO2 and Reservoir Datasets

We provide illustrations of the Kimberlina CO2 and Reservoir dataset. As shown in figure 5 and figure 6, the first column presents the velocity maps and the rest columns are three channels of the seismic data w.r.t. different sources. It’s worth mentioning that the Kimberlina CO2 simulates the CO2 leakage over a duration of 200 years, and the leakage mass can be categorized into: small, medium and large size, demonstrated as three rows in figure 5.
Recall that the CurvedVel dataset was synthesized with one geological fault and central frequency of 15 Hz, as shown in the first row in figure 7. The rest rows in figure 7 correspond to velocity maps and the generated seismic data with central frequencies of 20Hz and 25Hz, respectively. Similarly, figure 8 illustrates velocity maps with 1, 2, 3 and 4 geologic faults and the corresponding shot-gather seismic data with different sources.
Figure 6: The Kimberlina Reservoir Dataset

Figure 7: Illustration of CurvedVel dataset with different central frequencies
Figure 8: Illustration of CurvedVel dataset with multiple geologic faults

References

[1] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*, 2020.

[2] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre GR Day, Clint Richardson, Charles K Fisher, and David J Schwab. A high-bias, low-variance introduction to machine learning for physicists. *Physics reports*, 810:1–124, 2019.

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

[4] Fangshu Yang and Jianwei Ma. Deep-learning inversion: A next-generation seismic velocity model building method. *Geophysics*, 84(4):R583–R599, 2019.

[5] Blake A Richards, Timothy P Lillicrap, Philippe Beaudoin, Yoshua Bengio, Rafal Bogacz, Amelia Christensen, Claudia Clopath, Rui Ponte Costa, Archy de Berker, Surya Ganguli, et al. A deep learning framework for neuroscience. *Nature neuroscience*, 22(11):1761–1770, 2019.

[6] Guangming Zhu, Bin Jiang, Liz Tong, Yuan Xie, Greg Zaharchuk, and Max Wintermark. Applications of deep learning to neuro-imaging techniques. *Frontiers in neurology*, 10:869, 2019.

[7] Jean Virieux and Stéphane Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26, 2009.

[8] Sheng Xu, D Wang, F Chen, Yu Zhang, and G Lambare. Full waveform inversion for reflected seismic data. In *74th EAGE Conference and Exhibition incorporating EUROPEC 2012*, pages cp–293. European Association of Geoscientists & Engineers, 2012.

[9] Albert Tarantola. *Inverse problem theory and methods for model parameter estimation*. SIAM, 2005.
[10] Youzuo Lin and Lianjie Huang. Acoustic-and elastic-waveform inversion using a modified total-variation regularization scheme. *Geophysical Journal International*, 200(1):489–502, 2014.

[11] Youzuo Lin and Lianjie Huang. Quantifying subsurface geophysical properties changes using double-difference seismic-waveform inversion with a modified total-variation regularization scheme. *Geophysical Supplements to the Monthly Notices of the Royal Astronomical Society*, 203(3):2125–2149, 2015.

[12] Qili Zeng, Shihang Feng, Brendt Wohlberg, and Youzuo Lin. Inversionnet3d: Efficient and scalable learning for 3d full waveform inversion. *arXiv preprint arXiv:2103.14158*, 2021.

[13] Wenlong Wang, Fangshu Yang, and Jianwei Ma. Velocity model building with a modified fully convolutional network. In *SEG Technical Program Expanded Abstracts 2018*, pages 2086–2090. Society of Exploration Geophysicists, 2018.

[14] Qili Zeng, Shihang Feng, Brendt Wohlberg, and Youzuo Lin. Inversionnet3d: Efficient and scalable learning for 3d full waveform inversion. *arXiv preprint arXiv:1901.07733*, 2019.

[15] Wenlong Wang, Fangshu Yang, and Jianwei Ma. Velocity model building with a modified fully convolutional network. In *SEG Technical Program Expanded Abstracts 2018*, pages 2086–2090. Society of Exploration Geophysicists, 2018.

[16] Shuai Li, Bin Liu, Yuxiao Ren, Yangkang Chen, Senlin Yang, Yunhai Wang, and Peng Jiang. Deep-learning inversion of seismic data. *arXiv preprint arXiv:1901.07733*, 2019.

[17] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[18] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[19] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[20] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[21] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

[22] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938, 2017.

[23] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189*, 2018.

[24] Zhongping Zhang, Yue Wu, Zheng Zhou, and Youzuo Lin. Velocitygan: Subsurface velocity image estimation using conditional adversarial networks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 705–714. IEEE, 2019.

[25] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[26] David A McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170, 1999.

[27] Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*, pages 11–30. Springer, 2015.

[28] Peter L Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE transactions on Information Theory*, 44(2):525–536, 1998.

[29] Huan Xu and Shie Mannor. Robustness and generalization. *Machine learning*, 86(3):391–423, 2012.

[30] Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

[31] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[32] Philip M Long and Hanie Sedghi. Generalization bounds for deep convolutional neural networks. *arXiv preprint arXiv:1905.12600*, 2019.
[33] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. arXiv preprint arXiv:1706.08947, 2017.

[34] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In International Conference on Machine Learning, pages 254–263. PMLR, 2018.

[35] Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. Advances in Neural Information Processing Systems, 32:10836–10846, 2019.

[36] Mauricio Araya-Polo, Joseph Jennings, Amir Adler, and Taylor Dahlke. Deep-learning tomography. The Leading Edge, 37(1):58–66, 2018.

[37] 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, 2020.

[38] Amir Adler, Mauricio Araya-Polo, and Tomaso Poggio. Deep learning for seismic inverse problems: Toward the acceleration of geophysical analysis workflows. IEEE Signal Processing Magazine, 38(2):89–119, 2021.

[39] Alan Richardson. Seismic full-waveform inversion using deep learning tools and techniques. arXiv preprint arXiv:1801.07232, 2018.

[40] 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. arXiv preprint arXiv:2110.07584, 2021.

[41] Yuxin Yang, Xitong Zhang, Qiang Guan, and Youzuo Lin. Making invisible visible: Data-driven seismic inversion with physics-informed data augmentation, 2021.

[42] Saraiva Marcus, Forechi Avelino, de Oliveira Neto Jorcy, DelRey Antonio, and Rauber Thomas. Data-driven full-waveform inversion surrogate using conditional generative adversarial networks. arXiv preprint arXiv:2105.00100, 2021.

[43] Vladimir Vapnik and Alexey Chervonenkis. Theory of pattern recognition, 1974.

[44] Vladimir Vapnik and Alexey Chervonenkis. The necessary and sufficient conditions for consistency in the empirical risk minimization method. Pattern Recognition and Image Analysis, 1(3):283–305, 1991.

[45] Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. Regularization networks and support vector machines. Advances in computational mathematics, 13(1):1–50, 2000.

[46] Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research, 3(Nov):463–482, 2002.

[47] Peter L. Bartlett, Olivier Bousquet, and Shahar Mendelson. Local rademacher complexities. The Annals of Statistics, 33(4):1497–1537, 2005.

[48] Noga Alon, Shai Ben-David, Nicolo Cesa-Bianchi, and David Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. Journal of the ACM (JACM), 44(4):615–631, 1997.

[49] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In Conference on Learning Theory, pages 1376–1401. PMLR, 2015.

[50] Peter Bartlett, Dylan J Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. arXiv preprint arXiv:1706.08498, 2017.

[51] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. arXiv preprint arXiv:1707.09564, 2017.

[52] Xingguo Li, Junwei Lu, Zhaoran Wang, Jarvis Haupt, and Tuo Zhao. On tighter generalization bound for deep neural networks: Cnns, resnets, and beyond. arXiv preprint arXiv:1806.05159, 2018.

[53] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In Conference On Learning Theory, pages 297–299. PMLR, 2018.

[54] Shan Lin and Jingwei Zhang. Generalization bounds for convolutional neural networks, 2019.

[55] Antoine Ledent, Waleed Mustafa, Yunwen Lei, and Marius Kloft. Norm-based generalisation bounds for multi-class convolutional neural networks, 2021.

[56] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. arXiv preprint arXiv:1912.02292, 2019.

[57] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. arXiv preprint arXiv:1912.02178, 2019.
[58] Hanie Sedghi, Vineet Gupta, and Philip M. Long. The singular values of convolutional layers, 2019.
[59] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285, 2016.
[60] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015.
[61] Wenzhe Shi, Jose Caballero, Lucas Theis, Ferenc Huszar, Andrew Aitken, Christian Ledig, and Zehan Wang. Is the deconvolution layer the same as a convolutional layer? arXiv preprint arXiv:1609.07009, 2016.
[62] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In Proceedings of the IEEE international conference on computer vision, pages 2758–2766, 2015.
[63] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015.
[64] Jun Qi, Jun Du, Sabato Marco Siniscalchi, Xiaoli Ma, and Chin-Hui Lee. On mean absolute error for deep neural network based vector-to-vector regression. IEEE Signal Processing Letters, 27:1485–1489, 2020.
[65] Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George J Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. arXiv preprint arXiv:1906.04893, 2019.
[66] Kevin Scaman and Aladin Virmaux. Lipschitz regularity of deep neural networks: analysis and efficient estimation. arXiv preprint arXiv:1805.10965, 2018.
[67] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013.
[68] Patrick L Combettes and Jean-Christophe Pesquet. Lipschitz certificates for layered network structures driven by averaged activation operators. SIAM Journal on Mathematics of Data Science, 2(2):529–557, 2020.
[69] Dongmian Zou, Radu Balan, and Maneesh Singh. On lipschitz bounds of general convolutional neural networks. IEEE Transactions on Information Theory, 66(3):1738–1759, 2019.
[70] Radu Balan, Maneesh Singh, and Dongmian Zou. Lipschitz properties for deep convolutional networks. Contemporary Mathematics, 706:129–151, 2018.
[71] Chengyuan Deng, Yinan Feng, Shihang Feng, Peng Jin, Xitong Zhang, Qili Zeng, and Youzu Lin. Openfwi: Benchmark seismic datasets for machine learning-based full waveform inversion. arXiv preprint arXiv:2111.02926, 2021.
[72] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4):600–612, 2004.