We present OpenFWI, a collection of large-scale open-source benchmark datasets for seismic full waveform inversion (FWI). OpenFWI is the first-of-its-kind in the geoscience and machine learning community to facilitate diversified, rigorous, and reproducible research on machine learning-based FWI. OpenFWI (as shown in figure 1) includes datasets of multiple scales, encompasses diverse domains, and covers various levels of model complexity. Along with the dataset, we also perform an empirical study on each dataset with a fully-convolutional deep learning model. OpenFWI has been meticulously maintained and will be regularly updated with new data and experimental results. We appreciate the inputs from the community to help us further improve OpenFWI. At the current version, we publish seven datasets in OpenFWI, of which one is specified for 3D FWI and the rest are for 2D scenarios. All datasets and related information\footnote{Codes will be released upon approval by Los Alamos National Laboratory and U.S. Department of Energy.} can be accessed through our website at https://openfwi-lanl.github.io/.

1 Introduction

The subsurface geophysical properties (velocity, density, bulk modulus, etc.) are critical to a myriad of subsurface applications, such as carbon sequestration, earthquake detection and early-alarming, etc [1]. These important properties can be inferred by solving seismic full waveform inversion (FWI), the process of which is illustrated in figure 2. With the target of minimizing the residual between predicted and observed seismic signals, FWI falls into the family of non-convex optimization problems with PDE constraints. Such problems have been intensively studied in the paradigm of physics-driven approaches [2, 3, 4, 5, 6, 7]. Notorious complications of these approaches include extravagant computational power consumption, cycle-skipping, and ill-posedness issues. In spite of substantial efforts by imposing various regularization terms [8, 9, 10, 11, 12] to mitigate the aforementioned problems, the issue of expensive computation is still inevitable in the physics-driven framework.

With the advance of deep learning techniques, researchers have been actively exploring end-to-end solutions for complicated scientific computational imaging problems [13, 14, 15, 16, 17]. These methods are usually regarded as...
Data-driven approaches, since they yield a hefty data dependency, just to name a few: quantity, diversity, authenticity, etc. In recent years, promising results have been demonstrated for data-driven FWI. The early attempt by [18] introduced an 8-layer fully neural network model to obtain velocity maps from shot gathers. Inspired by the renowned results of using convolutional neural networks in Computer Vision, prolific works have been proposed based on encoder-decoder fully convolutional networks [17, 19, 20, 21]. These models focus on 2D solutions. Very recently, the first solution for 3D imaging has been demonstrated by [22], which employs group convolution in the encoder and invertible layers in the decoder for high efficiency and scalability. For all the aforementioned methods, the models are supervised, meaning that both velocity maps and seismic data are required for the training purpose. But in practice, it is unrealistic to obtain such a large volume of labeled data in advance. To alleviate the heavy reliance on labels, [23] recently developed an unsupervised inversion model incorporating both InversionNet model and the governing wave equation. For a thorough view of the research line of data-driven FWI, please refer to a survey paper [24].

Undeterred by the delights on this celebrated progress, we have a crucial observation: All experiments are implemented on individual datasets scarcely published! This immediately leads to several defective implications on (1) Unified Evaluation: How do we confirm the empirical superiority of one machine learning solution to another? The experimental protocol has to be identical, which is not possible with data not being published. A unified evaluation also helps our community with a global perspective of what we have achieved and what lies ahead. (2) Further Improvement: What if motivated researchers would like to further improve the proposed algorithm but have no data to test with? The data access denial only strangles valuable inspirations in the cradle. (3) Re-producibility and Integrity: Although it sounds a pathological scenario, people may still preserve the skeptics on the integrity of such a blossom of results, as implementing deep neural networks becomes more trivial. We also remark that the lack of open-sourced data is reasonable due to the difficulty of data acquisition. We either collect data from a real-world field study, which requires arduous human labor, or synthesize velocity maps and then generate seismic data by forward modeling. Because of these concerns and conditions, we believe our community has arrived at a complete and sound junction to establish an open-source data and benchmark platform.

We present OpenFWI, the first collection of large-scale seismic FWI datasets. OpenFWI is dedicated to facilitating diversified, rigorous, and reproducible research on data-driven FWI. Seven benchmark datasets will be released in OpenFWI, together with detailed descriptions on the background, synthesis process and file formats, etc. The benchmark datasets in OpenFWI demonstrate the following favorable characteristics:

1. Multi-scale: OpenFWI covers multiple scales of data samples. The smallest one has around 20K data samples and can fit into the memory of a single GPU, which supports training without massive computation power. The large datasets contain about 70K data samples, which are usually trained in the distributed setting, further expediting the development of scalable algorithms for deep learning-driven FWI.

2. Diverse Domains: OpenFWI empowers both 2D and 3D scenarios of FWI. The 3D Kimberlina dataset is the first large-scale 3D dataset generated by multiple institutions [25] and supported under the US Department of Energy-SMART Initiative [26]. The 2D datasets include velocity models that are representative of realistic subsurface applications (such as time-lapse imaging, subsurface carbon sequestration, general purpose, etc.).
As an example, the Kimberlina CO\textsubscript{2} datasets describe the spatial and temporal migration of the supercritical CO\textsubscript{2} within the reservoir, which are accompanied by timestamps and can be used for time-lapse imaging.

3. **Various levels of model complexity**: Depending on the subsurface modeling, OpenFWI encompasses a wide range of ‘model complexity’. We provide several metrics on the model complexity based on the spatial information, which further can be regarded as an estimation on the difficulty of learning. As a result, OpenFWI supports researchers to start with moderate datasets and refine for more challenging ones.

In addition to publishing the datasets, OpenFWI illustrates the experimental results of each dataset, obtained by InversionNet\cite{19,22}, an end-to-end encoder-decoder based deep neural networks. These results can be regarded as a baseline to empower potential advancements on ML-based FWI. With OpenFWI datasets, we envisage three challenges to meet in the near future: (1) 2D FWI with highly-complicated velocity maps. The style-transfer dataset presented by OpenFWI is the most challenging dataset, yielding some space for empirical improvement. (2) 3D FWI. The 3D Kimberlina-V1 datasets is the first one for 3D models, and the InversionNet3D\cite{22} is the “solitary” work so far tackling this task. (3) Unsupervised learning for FWI. Currently, all the datasets support full-supervised learning. Two datasets, in particular, embark the study of unsupervised learning for data-driven FWI. OpenFWI is anticipated to inspire and more importantly, support the development including all the challenges.

The rest of the paper is organized as follows: Section 2 introduces the background of the governing equation, and the method used for data synthesis. Section 3 provides a detailed description for each dataset with illustrations. Section 4 includes experimental results as baselines for each dataset. In section 5, we initiate a discussion on the model complexity and the performance of our deep learning model. Finally, section 6 concludes the paper.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure2.png}
\caption{Illustration of Seismic Full Waveform Inversion (FWI) and Forward Modelling.}
\end{figure}

## 2 Backgrounds

### 2.1 Acoustic Forward Modelling

The governing equation of the acoustic wave forward modeling is the wave equation,

\[
\left[ \frac{1}{\kappa(r)} \frac{\partial^2}{\partial t^2} - \nabla \cdot \left( \frac{1}{\rho(r)} \nabla \right) \right] p(r, t) = s(r, t),
\]

where \( r = (x, y, z) \) represents the spatial location in Cartesian coordinates, \( t \) denotes time, \( \kappa(r) \) and \( \rho(r) \) are the material bulk modulus and density, respectively. \( p(r, t) \) represents the pressure wavefield, and \( s(r, t) \) is the source term that specifies the location and time history of the source. Then, the compressional-(P-)velocity can be represented as

\[
c(r) = \sqrt{\frac{\kappa(r)}{\rho(r)}}.
\]

Forward wave propagation modeling entails calculating equation 1 given the boundary conditions and source, as well as subsurface geophysical parameters, including velocity \( c \) and density \( \rho \). For simplify, We can denote the forward modeling problems expression as

\[
p = f(m),
\]

where \( f \) represents the highly nonlinear forward mapping and \( m \) is the subsurface geophysical parameter vector.

In the simulation of the following datasets, we use the finite-difference method\cite{27} with 2nd-order accurate in time and 8th-order accurate in space. The absorbing boundary conditions\cite{28} are applied to all the boundaries.
3 OpenFWI Datasets

In this section, we start by giving an overview of OpenFWI datasets, each of which is then followed by a section of introduction on the background, synthesis process, formats, etc.

3.1 Dataset Summary

We summarize the basic information of all datasets presented by OpenFWI in table 1 below.

| Dataset          | Size    | Training Set | Testing Set | Seismic Data Size | Velocity Map Size |
|-----------------|---------|--------------|-------------|-------------------|-------------------|
| FlatVEL         | 247GB   | 45K          | 5K          | $3 \times 2000 \times 150$ | $100 \times 150$  |
|                 | Note    | Simple situation with flat layers. |
| CurvedVEL       | 247GB   | 45K          | 5K          | $3 \times 2000 \times 150$ | $100 \times 150$  |
|                 | Note    | Simple situation with curved layers. |
| Kimberlina-CO$_2$ | 96GB    | 15K          | 4430        | $9 \times 1251 \times 101$ | $401 \times 141$  |
|                 | Note    | Simulation of co2 leakage in 991 scenarios over a duration of 200 years. |
| Style-transfer  | 172GB   | 65K          | 2K          | $10 \times 200 \times 200$ | $200 \times 200$  |
|                 | Note    | Velocity maps transfer from real-life images, and contains two resolutions. |
| FlatFault       | 96GB    | 24K + 24K    | 6K          | $5 \times 1000 \times 70$ | $70 \times 70$    |
|                 | Note    | Except for 30K labeled data pairs, It contains 24K unlabeled seismic data. |
| CurvedFault     | 96GB    | 24K + 24K    | 6K          | $5 \times 1000 \times 70$ | $70 \times 70$    |
|                 | Note    | Except for 30K labeled data pairs, It contains 24K unlabeled seismic data. |
| 3D Kimberlina-V1 | 1.4TB   | 1664         | 163         | $25 \times 5001 \times 40 \times 40$ | $350 \times 400 \times 400$ |
|                 | Note    | Experimental version generated based on limited number of full-size 351 × 601 × 601 velocity models. Simulations non-uniformly cover a time range of 200 years. |

3.2 The FlatVEL & CurvedVEL Dataset

3.2.1 Data Description

FlatVEL and CurvedVEL are two large-scale geophysical dataset, which each consists of 50K pairs of seismic data and velocity maps. FlatVEL are velocity maps with flat layers and CurvedVEL are velocity maps with curved layers. The seismic data and velocity maps are splitted as 45K/5K for training, validation and testing respectively. Samples are shown in figure 3 and figure 4.

The size of the velocity maps in FlatVEL and CurvedVEL are both 100×150 grid points and the grid size is 15 meter in both x and z direction. Each velocity map contains 2 to 5 layers and the thickness of each layer ranges from 5 to 80 grids. The layers in the CurvedVEL follows a sine function. Compared to the maps with flat layers in FlatVEL, curved velocity maps yield much more irregular geological structures, making inversion more challenging. The velocity in each layer is randomly sampled from a uniform distribution between 1.5 $km/s$ and 3.5 $km/s$. The velocity is designed to increase with depth to be more physically realistic. We also add geological faults to every velocity map. The faults shift from 0 grids to 80 grids, and the tilting angle ranges from 25° to 165°.
To synthesize the seismic data, three sources are distributed on the surface at 0.975km, 1.125km and 1.275km. The seismic traces are recorded by 150 receivers positioned at each grid with an interval of 15 meter. The source is a Ricker wavelet (Wang, 2015) with a central frequency of 15 Hz. Each receiver records time-series data for 2 second, and we use a 1 millisecond sample rate to generate 2,000 timesteps. Therefore, the dimensions of seismic data become 3x2000x150.

### 3.2.2 Data Generation

The generation of the synthetic FlatVEL and CurveVEL is based on the iterative deformation of randomly generated flatten subsurface structures. There are four major steps to generate a new subsurface sample:

1. Generate random flatten subsurface structures by randomly setting each layer’s width and velocity. The velocity of each layer is sorted in descending order from the bottom to the top.
2. Deform the flatten image by $\sin(\cdot)$ function by setting the period and the magnitude of the $\sin(\cdot)$ randomly to mimic the extrusion effect.
3. Divide the image into two parts by generating a line randomly and shifting one of the parts to mimic the fault. Make sure there is no intersection with previous faults in the current iteration.
4. Discard the part without shifting and go back to step 2 to fill the empty region until the whole image is fulfilled.

### 3.2.3 File Format and naming

**Format:** All samples of FlatVEL and CurvedVEL datasets are stored in .npz format. Velocity maps and seismic data are stored as sub-directories in each dataset. Each file contains a single NumPy array of one sample. The shapes of the arrays in velocity map files and seismic data files are (1, 3, 150, 100) and (1, 150, 100) respectively. It is worth mentioning that .npz is nothing more than a compressed format of .npy, we converted the data only for the sake of space consumption reduction.

**Naming:** The naming of files follows the format of `{data}_{n}.npz` for seismic data and `{model}_{n}.npz` for velocity maps, n denotes the index of a file (starting from 0). Notice that for the same n, data and model becomes a pair. Here are some examples:

- `data_1.npz` is the file that contains seismic data, is the first file among all.
- `model_99.npz` is the file containing the 99-th velocity map.

### 3.3 The Kimberlina-CO$_2$ Dataset

#### 3.3.1 Data Description

In geologic carbon sequestration (GCS), developing effective monitoring methods is urgently needed to detect and respond to CO$_2$ leakage. However, we are unaware of any available field seismic data that fits the scope of the problem of leakage detection. Given the importance of the problem, the U.S. Department of Energy (DOE), through the National Risk Assessment Partnership (NRAP) project, has generated a set of high fidelity simulations, the Kimberlina dataset, with the aim of providing a standard baseline dataset to understand and assess the effectiveness of various geophysical monitoring techniques for detecting CO$_2$ leakage [29]. Specifically, the Kimberlina dataset contains 991 CO$_2$ leakage scenarios, each simulated over a duration of 200 years, with 20 leakage velocity maps provided (i.e., at every 10 years) for each scenario [30]. Excluding some missing velocity maps, in total, it has 19,430 pairs of seismic data and velocity maps. The data are split as 15K/4,430 for training and testing, respectively. Samples are shown in figure 5.

The size of the velocity maps in Kimberlina-CO$_2$ dataset are 401x141 grid points and the grid size is 10 meter in both x and z direction. To synthesize the seismic data, 9 sources are evenly distributed along the top of the model, with depths of 5 m. The seismic traces are recorded by 141 receivers positioned at each grid with an interval of 10 meter. The source is a Ricker wavelet with a central frequency of 10 Hz. Each receiver records time-series data for 2 second, and we use a 0.5 ms sample rate to generate 2,000 timesteps, then the seismic data is downsampled to 1,251x101 to save the memory and the computation cost for data-driven FWI. Therefore, the dimensions of seismic data become 9x1,251x101.

#### 3.3.2 File Format and naming

All samples of Kimberlina-CO$_2$ datasets are stored in .npz format. Velocity maps and seismic data are stored as sub-directories in each dataset. ach file contains a single NumPy array of one sample. The shapes of the arrays in
velocity map files and seismic data files are (9, 1251, 101) and (401, 141) respectively. It’s worth mentioning that .npz is nothing more than a compressed format of .npy, we converted the data only for the sake of space consumption reduction.

**Naming:** The naming of files follows the format of \{data\}_sim[n]_t[m] for seismic data and \{label\}_sim[n]_t[m] for velocity maps, \texttt{n} denotes the 4 digital index of a simulation (starting from 0), and \texttt{m} represents the timesteps from 0 to 190 (at every 10 years). Notice that for the same \texttt{n} and \texttt{m}, data and model becomes a pair. Here are some examples:

- \texttt{data_sim0000_t0.npz} is the file that contains seismic data, and it is the first file among all.
- \texttt{label_sim0991_t160.npz} is the file containing the velocity map of the 991th scenarios for the 160th year.

### 3.4 The Style-transfer Dataset

#### 3.4.1 Data Description

Style-transfer dataset is a large-scale geophysical dataset built by style-transfer method[21], which contains seismic data for two types of velocity maps: high-resolution and low-resolution maps. Each type has 67K pairs of seismic data and velocity maps. The data are split as 65K/2K for training and testing. Samples are shown in figure 6.

The COCO dataset [31] is set as the content images and The Marmousi model is set as the style image. An image transfer network is trained to transfer the COCO dataset to velocity perturbations. A 1D velocity model is added to the velocity perturbations to construct the velocity maps. The velocity maps are then smoothed by a Gaussian filter with random standard derivation from 0 to 5 to build the high-resolution velocity maps and from 5 to 10 to build the low-resolution velocity maps. This dataset can be applied to the data-driven FWI and traveltime inversion. More details about the velocity building and the applications of this dataset can be found in [32] and [21].

The size of the velocity maps in Style-transfer dataset are 200×200 grid points and the grid size is 5 meter in both x and z directions. To synthesize the seismic data, 10 sources are evenly distributed on the surface with a spacing of 100 m. The seismic traces are recorded by 200 receivers positioned at each grid with an interval of 5 meter. The source is a Ricker wavelet with a central frequency of 15 Hz. Each receiver records time-series data for 2 second, and we use a 0.0002s sample rate to generate 5,000 timesteps, then the seismic data is downsampled from 5000×200 to 200×200 to save the memory and the computation cost for data-driven FWI. Therefore, the dimensions of seismic data become 10×200×200.

#### 3.4.2 File Format and naming

All samples of Style-transfer datasets are stored in .npz format. Velocity maps and seismic data are stored as sub-directories in each dataset. model_low and data_low indicate the low resolution velocity maps and their seismic data while model_high and data_high indicate the high resolution velocity maps and their seismic data. Similar to other datasets, we save the data in .npz format to save some memory space.

**Naming:** The naming of files follows the format of \{data\}_n for seismic data and \{model\}_n for velocity maps, \texttt{n} denotes the index of a file (starting from 0). Notice that for the same \texttt{n}, data and model becomes a pair. Here are some examples:

- \texttt{data_1.npz} is the file that contains seismic data and it is the first file among all.
- \texttt{model_99.npz} is the file containing the 99th velocity map.

### 3.5 The FlatFault & CurvedFault Dataset

#### 3.5.1 Data Description

FlatFault and CurvedFault are two large-scale geophysics dataset for FWI, each of which consists of 54K seismic data including 30K with paired velocity maps (labeled) and 24K unlabeled. Velocity maps in FlatFault and CurvedFault contain flat layers and curved layers, respectively. The 30K labeled pairs of seismic data and velocity maps are split into training set (24K), validation set (3K) and testing set (3K). Samples are shown in figure 7.

The aim of CurvedFault dataset is to better validate effectiveness of FWI methods on curved topography, and the shape of those curves follows a sine function. All velocity maps in FlatFault and CurvedFault contain 2 to 4 layers, and the velocity in each layer is uniformly sampled between 3,000 meter/second and 6,000 meter/second. The dimensions of each velocity map are 70×70 grid points, and the grid size is 15 meter in both directions. The layer thickness ranges from 15 to 35 grids. The velocity is designed to increase with depth to be more physically realistic. Geological faults
are also added to every velocity map. The faults shift from 10 to 20 grids, and the tilting angle ranges from -123° to 123°.

Five Ricker [33] sources with a central frequency of 25 Hz are used for seismic data synthesis, and they are evenly

distributed on the surface with a spacing of 255 meter. The seismic traces are recorded by 70 receivers positioned at

each grid with an interval of 15 meter. Each receiver records time-series data for 1 second, and we use a 1 millisecond

time rate to generate 1,000 timesteps. Therefore, the dimensions of seismic data become 5\times1000\times70.

3.5.2 File Format and naming

Format: All samples in OpenFWI are stored in .npy format. Velocity maps and seismic data are stored in separate files.

Each file contains a single NumPy array of 500 samples. The shapes of the arrays in velocity map files and seismic data

files are (500, 1, 70, 70) and (500, 5, 1000, 70), respectively.

Naming: The naming of files can be described as \{vel|seis\}_{n}_{i}.npy, where vel and seis specify whether

a file stores velocity maps or seismic data, n denotes the number of layers in (corresponding) velocity maps and \(i\) is the

index of a file (start from 0) among the ones with the same \(n\). Here are several examples:

- vel_2_1_3.npy is the file that contains velocity maps with two layers, and it is the fourth file among all the files with

two-layer velocity maps.

- vel_4_1_0.npy is the file that contains velocity maps with four layers, and it is the first file among all the files with

four-layer velocity maps.

- seis_4_1_0.npy is the file that contains the seismic data corresponding to the velocity maps in vel_4_1_0.npy.

3.6 The 3D Kimberlina-V1 Dataset

3.6.1 Data Description

3D Kimberlina dataset [34] is generated from a hypothetical model of a commercial-scale geologic carbon sequestration

reservoir at the Kimberlina site in the southern San Joaquin Basin, 30 km northwest of Bakersfield, CA, USA. The

reservoir models are provided in [35]. [36] provides the details to build the 3D velocity models. Simulations can be

downloaded from the DOE-EDX platform [37], which covers a time range of 200 years. The original 3D Kimberlina

velocity models in each time step is with a size of 351 \times 601 \times 601 (Depth \times Width \times Height).

Currently, 3D models in 29 time steps are available for experiments. Considering training deep neural networks requires

a larger number of data samples, for each original model, we manually extracted 63 sub-models with a smaller size of

350 \times 400 \times 400 every 20 grids in the Width and Height direction. The grid size is 10 meters. The sub-model As a

result, we obtained 1,827 3D velocity models, which are split into a training subset and a validation subset with the ratio

of 1664:163. Seismic data are generated accordingly using the finite-difference method with 1,600 receivers uniformly

distributed over the 2D earth surface and each of them captures vibration signals as time-series data of length 5,001

with a time spacing of 0.001s. There are 25 sources placed evenly on the 2D spatial grid over the surface and thus the

shape of raw seismic data is 25 \times 5001 \times 40 \times 40 (Channel \times Time \times Width \times Height). We refer to this dataset as

3D Kimberlina-V1. The complete 3D Kimberlina dataset will be released in future updates as 3D Kimberlina-V2.

Samples of velocity maps along y-axis in 3D Kimberlina-V1 are demonstrated in figure 8, note that each row stands for

one data sample, and each column contains a slice of the 3D cube.

3.6.2 File Format and naming

File Format: All samples of 3D Kimberlina-V1 are stored in .bin format. Such a binary record can be converted in to

a numpay array and reshaped to the original size via

```python
import numpy as np
bin_data = np.fromfile(bin_file, dtype=np.float32)
# for seismic data
sdata = bin_data.reshape((40, 40, 5001)).transpose((2, 1, 0))
# for velocity model
vdata = bin_data.reshape((400, 400, 350)).transpose((2, 1, 0))
```

Naming: Velocity models, stored under /Label folder, are named after their timestamp (\(t\)) and sub-volume serial num-

ber (\(n\)) as year\{t\}_cut\{n\}.bin. The corresponding 25-shot seismic data are stored under /data/year\{t\}_cut\{n\}/

with the name of shot\_{m}.bin, where \(m\) is the serial number of sources (1-indexed).
4 OpenFWI Benchmarks

In this section, we demonstrate experimental results on each dataset with InversionNet3D [22] for the 3D Kimberlina Dataset and InversionNet [19] for the rest. InversionNet is an encoder-decoder based fully-convolutional neural network model, and has shown promising results on data-driven FWI [19, 21, 38, 39]. Due to the distinct data input shape and model complexity of each dataset, we change the network architecture and hyper-parameters slightly, which are briefly summarized for each dataset. To provide an illustrative example, figure 9 demonstrates the backbone architecture of the neural network we use.

4.1 Experimental Settings

We perform extensive empirical study and provide the best results for each dataset. Two most commonly used loss functions are adopted: $l_1$-loss and $l_2$-loss. Notice that the optimization method and other training details vary among datasets, thus are introduced separately for each dataset. To illustrate a comprehensive evaluation, we introduce three metrics: mean absolute error (MAE), mean squared error (MSE) and structural similarity (SSIM) [40] for results obtained by both loss functions. MAE and MSE both capture the numerical difference between the predicted and ground truth velocity maps. In our experiments, we normalize the data entries to 0-1 for the convenience during training. The SSIM, however, measures the perceptual similarity between two images. All the experiments are implemented on NVIDIA Tesla P100 GPUs, we further provide the estimated training time of InversionNet on each dataset in the corresponding subsections.

4.2 Experimental Results

4.2.1 FlatVEL & CurvedVEL Dataset

We optimized the old-version InversionNet in [19] by discarding the conditional random fields and performed experiments on FlatVEL, CurvedVEL dataset. The network has a 10-layer encoder and 11-layer decoder, all the layers are equipped with convolution (or de-convolution) layers, batch normalization and leaky-Relu activation, except the last decoder layer, where we employ the tanh activation. The dimension of the bottleneck latent feature is 512. The decoder finalizes with a center crop layer to obtain the desired output shape. As a common practice, the data is normalized to 0-1 before input. We trained the network 100 epochs with $l_1$ and $l_2$ loss function, respectively. Table 2 demonstrates the quantitative results of the velocity error and SSIM. We trained these two datasets on a single NVIDIA Tesla P100 GPU, which takes around 4 hours to complete the training and testing.

4.2.2 Kimberlina 2D Dataset

We tested the Kimberlina 2D dataset with a 30-layer InversionNet. The network is composed of 15 layers encoder and 15 layers decoder. The encoder is composed of 15 convolutional layers; and the decoder is composed of 7 transposed convolutional layer and 8 convolutional layers alternately. Besides, the center-crop layer in the decoder is also kept. The dimension of the bottleneck latent feature is 512, too. We trained the network 100 epochs with $l_1$ and $l_2$ loss function, respectively. The examples of the predicted velocity maps are given in figure 10 and the quantitative results are given in table 3. The training time is approximately 3.5 hours by one P100 GPU.
### 4.2.3 Style-transfer Dataset

We have tested the Style-transfer dataset with the InversionNet. The network is composed of 6 encoder layers and 6 decoder layers. Each encoder layer is composed of 2 convolutional layers with stride equal to 1 and 2, respectively. Each decoder layer is composed of a convolutional layer with stride equal to 1 and a transposed convolutional layer with stride equal to 2. There are 32 channels in the first encoder layer and the number of channels is doubled in the latter encoder layers and then halved in the latter decoder layers. In the last decoder layer, a $tanh$ activation function is applied to give the final result. We have run 12 epochs with $l_1$ and $l_2$ loss function. The examples of the predicted velocity maps are given in figure 12 and the quantitative results are given in table 4. Training the model costs about 4 hours, also by a single P100 GPU.

| Dataset      | Loss       | Velocity Error |      |
|--------------|------------|----------------|------|
|              | $l_1$-loss | $l_2$-loss     | MAE↓ | MSE↓ | SSIM↑ |
| Style-transfer | ✓          | 0.1423         | 0.0496 | 0.7786 |
|              | ✓          | 0.1327         | 0.0401 | 0.7898 |

Table 4: Quantitative results of Style-transfer dataset with $l_1$ and $l_2$ loss function settings in terms of MAE, MSE and SSIM.

### 4.2.4 FlatFault & CurvedFault Dataset

Table 5 shows the quantitative results of InversionNet trained with $l_1$ loss function and $l_2$ loss function respectively. Since the number of receivers and the number of timesteps in seismic data are unbalanced, we modify the encoder part in InversionNet to first extract temporal features until the temporal dimension is close to the spatial dimension and then extract spatial-temporal features. The dimension of the bottleneck latent feature is 512. We keep the center-crop layer in the decoder part to transform feature maps into desired dimensions. In addition, for details on results obtained by unsupervised learning, we refer to our recent work in UPFWI[23]. Finally, it takes around 6 hours to train the model with one P100 GPU.

| Dataset             | Loss       | Velocity Error |      |
|---------------------|------------|----------------|------|
|                     | $l_1$-loss | $l_2$-loss     | MAE↓ | MSE↓ | SSIM↑ |
| FlatFault           | ✓          | 0.01049        | 0.00306 | 0.9793 |
|                     | ✓          | 0.01320        | 0.00287 | 0.9715 |
| CurvedFault         | ✓          | 0.01117        | 0.00469 | 0.9680 |
|                     | ✓          | 0.01407        | 0.00342 | 0.9571 |

Table 2: Quantitative results of FlatVEL and CurvedVEL dataset with $l_1$ and $l_2$ loss function settings in terms of MAE, MSE and SSIM.

### 4.3 Kimberlina 3D-V1 Dataset

Kimberlina 3D-V1 is a recently generated experimental dataset, on which only the performance of InversionNet3D [22] are reported. Due to the nature and design of InversionNet3D architecture, where invertible layers and group
| Dataset     | Loss          | Velocity Error |       |
|-------------|---------------|----------------|-------|
|             | $\ell_1$-loss | $\ell_2$-loss  | MAE↓  |
| FlatFault   | ✓             |                | 0.0111|
|             |               |                | 0.0012|
|             |               |                | 0.9799|
| CurvedFault | ✓             |                | 0.0106|
|             |               |                | 0.0007|
|             |               |                | 0.9866|
|             | ✓             |                | 0.0174|
|             |               |                | 0.0029|
|             |               |                | 0.9625|
|             | ✓             |                | 0.0177|
|             |               |                | 0.0021|
|             |               |                | 0.9676|

Table 5: Quantitative results of FlatFault and CurvedFault dataset with $\ell_1$ and $\ell_2$ loss function settings in terms of MAE, MSE and SSIM.

5 Discussion

5.1 Model Complexity Analysis

All OpenFWI datasets are generated in two steps: (1) Synthesize velocity maps; (2) Obtain seismic data via forward modelling. To present various subsurface structures in real-world scenarios, we apply different approaches during the velocity map synthesis. Simple techniques include parallel shifting and curving via trigonometric functions, while the most complicated method would be transfer-learning from large-scale nature images. Therefore, the velocity maps in the six datasets for 2D FWI encompass different levels of model complexity. From the illustrations of each dataset (figure 3, figure 4, figure 5, figure 6, figure 7), we may get simple intuitions such as ’curved’ models should be more complicated than ’flat’ ones. Yet still, a standard metric would be more appropriate for a justification on the model complexity.

We employ several metrics based on the spatial information [41] to provide a qualitative measurement of the model complexity. These metrics can be regarded as an estimator of the edge magnitude, since they are computed by applying a Sober operator, which is used as an edge detection kernel. To be specific, let $G_x$ and $G_y$ denote the gradient magnitude on horizontal and vertical coordinates $(x, y)$ obtained via the Sober filter, let $G$ be the matrix with $G_p$ as the element on pixel $p$, and $P$ be the total number of pixels, we have the following definitions:

$$G_p = \sqrt{(G_x^2 + G_y^2)},$$

$$SI_{mean} = \frac{1}{P} \sum_{p=1}^{P} G_p,$$

$$SI_{std} = \sqrt{\frac{1}{P} \sum_{p=1}^{P} G_p^2 - SI_{mean}^2},$$

$$GSI = \frac{\|\text{vec}(G)\|_0}{P}.$$ 

It follows that $SI_{mean}$ stands for the mean of the magnitude over the image, and $SI_{std}$ for the standard deviation. The GSI (gradient sparsity index) is simply the ratio between the number of non-zero entries in matrix $G$ and $P$. These metrics based on spatial information would be more proper than others (such as compression factor) in our scenarios,
they are able to capture the subsurface structures based on the edge detection. We sample a fraction of velocity maps from each dataset and compute the metrics defined above, the average results are shown in table 7. In particular, the Kimberlina dataset contains three categories based on the leakage mass, and the style-transfer dataset has two levels of resolution, we sample the same number of velocity maps from each category. Note that the numerical values are more like a reference, a qualitative analysis of model complexity suffice in our case. In figure 14, the position of each dataset along the Y axis represents the relative model complexity.

| Complexity Measure       | FlatVEL | CurvedVEL | FlatFault | CurvedFault | Kimberlina | Style-transfer |
|--------------------------|---------|-----------|-----------|-------------|------------|----------------|
| Spatial Information Mean | 5.1271  | 5.4536    | 5.6445    | 6.0857      | 5.6639     | 13.4350        |
| Spatial Information Std  | 25.7746 | 26.6369   | 27.0584   | 28.3049     | 27.4254    | 40.5910        |
| Gradient Sparsity Index  | 0.2269  | 0.2380    | 0.2422    | 0.2626      | 0.2475     | 0.5270         |

Table 7: Summary of the Datasets

5.2 InversionNet Performance

An important thrust of introducing the model complexity, is that OpenFWI is expected to facilitate data-driven FWI research on multiple levels. Beginners may feel comfortable with “easier” datasets while a recently-proposed, complicated algorithm may obtain convincing improvement through challenging datasets.

At the same time, we observe that the InversionNet, though with minor changes due to the shape changes, yield worse performance as the model complexity rises. In particular, the style-transfer dataset leaves a considerable gap with others on all of the MAE, MSE and SSIM evaluation metrics, thus could be considered as de facto the hardest dataset in OpenFWI. Notice that we do not necessarily correlate model complexity and training results. Claims such as “harder datasets lead to worse training results with the same deep learning algorithm”, although sounds intuitively reasonable, would need rigorous theoretical justifications. Here we only summarize the results by InversionNet as a reference to the community to optimize their strategy on selecting a proper dataset.
6 Conclusion

In this paper, we present OpenFWI, the first open-source dataset and benchmark platform for data-driven seismic FWI. OpenFWI aims to facilitate research in both geoscience and machine learning community with plenteous, realistic, and diverse data resources, along with detailed, comparable and reproducible experimental results. The seven benchmark datasets encompass FWI in 2D and 3D scenarios, cover both supervised and unsupervised learning setting, and are distinct in terms of size and model complexity. We expect OpenFWI to contribute to the advancement of the frontier of the data-driven FWI research. Meanwhile, we also realize the OpenFWI is far from being satisfactory. By sharing the current set of data and putting this effort, we would like to hear feedback and comments from the community that would help us to improve this work.

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.

References

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

[2] Andreas Fichtner. *Full seismic waveform modelling and inversion*. Springer Science & Business Media, 2010.

[3] Zhigang Zhang, Lianjie Huang, and Youzuo Lin. A wave-energy-based precondition approach to full-waveform inversion in the time domain. In *SEG Technical Program Expanded Abstracts 2012*, pages 1–5. Society of Exploration Geophysicists, 2012.

[4] Yong Ma, Dave Hale, Bin Gong, and Zhaobo Meng. Image-guided sparse-model full waveform inversion. *Geophysics*, 77(4):R189–R198, 2012.

[5] Zhigang Zhang and Lianjie Huang. Double-difference elastic-waveform inversion with prior information for time-lapse monitoring. *Geophysics*, 78(6):R259–R273, 2013.

[6] Shihang Feng and Gerard T Schuster. Transmission+ reflection anisotropic wave-equation traveltime and waveform inversion. *Geophysical Prospecting*, 67(2):423–442, 2019.

[7] Shihang Feng, Lei Fu, Zongcai Feng, and Gerard T Schuster. Multiscale phase inversion for vertical transverse isotropic media. *Geophysical Prospecting*, 69(8-9):1634–1649, 2021.

[8] 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.

[9] 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.

[10] Wenyi Hu, Aria Abubakar, and Tarek M Habashy. Simultaneous multifrequency inversion of full-waveform seismic data. *Geophysics*, 74(2):R1–R14, 2009.

[11] Antoine Guittton. Blocky regularization schemes for full-waveform inversion. *Geophysical Prospecting*, 60(5):870–884, 2012.

[12] Yuqing Chen, Zongcai Feng, Lei Fu, Abdullah AlTheyab, Shihang Feng, and Gerard Schuster. Multiscale reflection phase inversion with migration deconvolution. *Geophysics*, 85(1):R55–R73, 2020.

[13] 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*, 38:89–119, 2021.

[14] 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*, 1(1):1–34, 2020.

[15] 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.

[16] 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.
Fangshu Yang and Jianwei Ma. Deep-learning inversion: A next-generation seismic velocity model building method. *Geophysics*, 84(4):R583–R599, 2019.

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

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

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.

Shihang Feng, Youzuo Lin, and Brendt Wohlberg. Multiscale data-driven seismic full-waveform inversion with field data study. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–14, 2021.

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.

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.

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.

David Alumbaugh, Michael Commer, Dustin Crandall, Erika Gasperikova, Shihang Feng, William Harbert, Yaoguo Li, Youzuo Lin, Savini Manthila Samarasinghe, and Xianjin Yang. Development of a multi-scale synthetic data set for the testing of subsurface CO_{2} storage monitoring strategies. In *American Geophysical Union (AGU 2021)*, 2021.

U.S. Department of Energy. Science-informed machine learning for accelerating real-time decisions in subsurface applications (smart initiative), 2019-2029.

Peter Moczo, Johan OA Robertsson, and Leo Eisner. The finite-difference time-domain method for modeling of seismic wave propagation. *Advances in geophysics*, 48:421–516, 2007.

Björn Engquist and Andrew Majda. Absorbing boundary conditions for numerical simulation of waves. *Proceedings of the National Academy of Sciences*, 74(5):1765–1766, 1977.

Preston Jordan and Jeffrey Wagoner. Characterizing construction of existing wells to a CO_{2} storage target: The Kimberlina site, California. Technical report, U.S. Department of Energy - Office of Fossil Energy, March 2017.

PD Jordan and JL Wagoner. Characterizing construction of existing wells to a CO_{2} storage target: The Kimberlina site, California. Technical report, National Energy Technology Laboratory (NETL), Pittsburgh, PA, Morgantown, WV . . . , 2017.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

Shihang Feng, Youzuo Lin, and Brendt Wohlberg. Physically realistic training data construction for data-driven full-waveform inversion and traveltime tomography. In *SEG International Exposition and Annual Meeting*. OnePetro, 2020.

Yanghua Wang. Frequencies of the Ricker wavelet. *Geophysics*, 80(2):A31–A37, 2015.

J. Wagoner. 3D Geologic Modeling of the Southern San Joaquin Basin for the Westcarb Kimberlina Demonstration Project- A Status Report. Technical Report LLNL-TR-412487, Lawrence Livermore National Lab. (LLNL), Livermore, CA (United States), April 2009.

Haruko Wainwright, Stefan Finsterle, Quanlin Zhou, and Jens Birkholzer. Modeling the performance of large-scale CO_{2} storage systems: A comparison of different sensitivity analysis methods. *International Journal of Greenhouse Gas Control*, 17:189–205, 2013.

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, 2018.

NETL. LLNL Kimberlina 1.2 simulations, 2018.

Zhongping Zhang and Youzuo Lin. Data-driven seismic waveform inversion: A study on the robustness and generalization. *IEEE Transactions on Geoscience and Remote sensing*, 58(10):6900–6913, 2020.
[39] Renán Rojas-Gómez, Jihyun Yang, Youzuo Lin, James Theiler, and Brendt Wohlberg. Physics-consistent data-driven waveform inversion with adaptive data augmentation. *IEEE Geoscience and Remote Sensing Letters*, 2020.

[40] 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.

[41] Honghai Yu and Stefan Winkler. Image complexity and spatial information. In *2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*, pages 12–17. IEEE, 2013.
Figure 3: Examples of velocity maps and their corresponding seismic measurements in FlatVEL dataset.
Figure 4: Examples of velocity maps and their corresponding seismic measurements in CurvedVel dataset.
Figure 5: Examples of velocity maps and their corresponding seismic measurements in Kimberlina-CO$_2$ dataset.
Figure 6: Examples of velocity maps and their corresponding seismic measurements in Style-transfer dataset. The top three rows are samples of low-resolution maps, and the bottom three rows are those in high-resolution maps.
Figure 7: Examples of velocity maps and their corresponding seismic measurements in FlatFault (row 1 to 3) dataset and CurvedFault (row 4 to 6).
Figure 8: Examples of velocity maps in 3D Kimberlina-V1 dataset.

Figure 10: The velocity map examples of Kimberlina CO2 dataset with $l_1$ and $l_2$ loss function setting. The first row is the predicted velocity maps with $l_1$ loss function. The second row is the predicted velocity maps with $l_2$ loss function. The third row is the true velocity maps.
Figure 11: The velocity map examples of Style-transfer dataset with $l_1$ and $l_2$ loss function setting. The first row is the predicted velocity maps with $l_1$ loss function. The second row is the predicted velocity maps with $l_2$ loss function. The third row is the true velocity maps.
Figure 12: The velocity map examples of FlatFault and CurvedFault dataset with $l_1$ and $l_2$ loss function in the supervised learning setting and unsupervised learning setting. The first row is the predicted velocity maps with $l_1$ loss function. The second row is the predicted velocity maps with $l_2$ loss function. The third row shows the results of unsupervised learning with 24k unlabeled data, while the fourth row is the result with 48k unlabeled data. The fifth row is the true velocity maps. The left two columns illustrate the results of FlatFault dataset and the right two for the CurvedFault dataset.
Figure 13: Spatial Placement of Sources. Each source is considered as seismic data of one channel in the input. Serial numbers are 0-indexed.

| Model            | Selected Channels | MAE ↓ | RMSE ↓ | SSIM ↑ |
|------------------|-------------------|-------|--------|--------|
| InvNet3Dx1       | [1, 2, 14, 15, 16, 20, 23, 24] | 9.83  | 26.11  | 0.9826 |
|                  | [6, 7, 8, 11, 13, 16, 17, 18] | 10.38 | 26.95  | 0.9819 |
|                  | [0, 2, 4, 10, 14, 20, 22, 24] | 10.19 | 27.27  | 0.9816 |
| five random strategies avg. | **10.41** | **27.57** | **0.9811** |

Table 6: Performance of InversionNet3Dx1 on different channel selection strategies of seismic input.