Data-Driven Seismic Waveform Inversion: A Study on the Robustness and Generalization

Zhongping Zhang and Youzuo Lin

***Abstract*— Full-waveform inversion is an important and widely used method to reconstruct subsurface velocity images. Waveform inversion is a typical nonlinear and ill-posed inverse problem. Existing physics-driven computational methods for solving waveform inversion suffer from the cycle-skipping and local-minima issues, and do not mention that solving waveform inversion is computationally expensive. In recent years, data-driven methods become a promising way to solve the waveform-inversion problem. However, most deep-learning frameworks suffer from the generalization and overfitting issue. In this article, we developed a real-time data-driven technique and we call it VelocityGAN, to reconstruct accurately the subsurface velocities. Our VelocityGAN is built on a generative adversarial network (GAN) and trained end to end to learn a mapping function from the raw seismic waveform data to the velocity image. Different from other encoder–decoder-based data-driven seismic waveform-inversion approaches, our VelocityGAN learns regularization from data and further imposes the regularization to the generator so that inversion accuracy is improved. We further develop a transfer-learning strategy based on VelocityGAN to alleviate the generalization issue. A series of experiments is conducted on the synthetic seismic reflection data to evaluate the effectiveness, efficiency, and generalization of VelocityGAN. We not only compare it with the existing physics-driven approaches and data-driven frameworks but also conduct several transfer-learning experiments. The experimental results show that VelocityGAN achieves the state-of-the-art performance among the baselines and can improve the generalization results to some extent.**

***Index Terms*— Condition adversarial networks, data-driven method, full-waveform inversion (FWI), transfer learning.**

I. Introduction

SEISMIC full-waveform-inversion (FWI) techniques are commonly used in geophysical exploration to determine site geology, stratigraphy, and rock quality. These techniques provide information about subsurface layering and rock geomechanical properties. In particular, seismic FWI infers a 2-D/3-D map of seismic velocity from observations (see Fig. 1). The seismic velocity depends on and, therefore, predicts the subsurface material properties. There are two primary ways of solving this problem depending on the complexity of the forward model that is used. The simpler approach is by travel-time inversion [43], which has a linear forward operator, but provides the results of inferior accuracy and resolution [29]. The FWI techniques [9], [13], [17], [46] provide superior solutions by modeling the wave propagation in the subsurface, but the forward operator is nonlinear and computationally expensive to simulate, and the problem is ill-posed, without a unique solution [46].

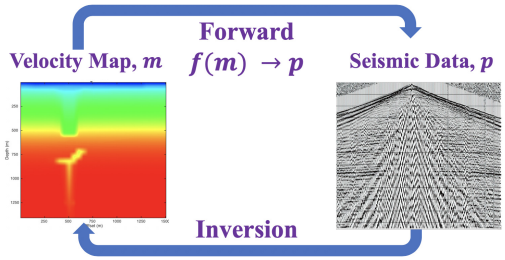


Fig. 1. Schematic of the FWI problem, which is to infer the velocity map from a given set of seismic data.

FWI can be solved in either time-domain or frequency domain [12], [17], [45]. The major challenges of solving FWI mostly come from threefolds: ill-posedness, cycle skipping, and high computational cost. Similar to other geophysical exploration methods, FWI suffers from limited data coverage, which results in extremely underconstrained inverse problems. Due to the fact that FWI is highly nonlinear and sensitive to the initial guess, a naive approach to the FWI problem typically converges to a local minima. When the starting model is far away from the global minimum (common in field applications), a deterministic algorithm is unable to move the events in seismic data to the correct cycle. The mismatch between the calculated and observed wavefield phases is also called cycle skipping [9]. Having low-frequency components in inversion is critical to alleviating this cycle-skipping issue. High computational cost is another challenge for solving the FWI problems. Most of the existing approaches to solve FWI rely on the iterative nonlinear optimization techniques. Given a 2-D $n \times n$ subsurface model, and assuming k line search steps are needed to determine the best step size at each iteration, computing the update has $\mathcal{O}(k\,n^2)$ cost.

To mitigate those aforementioned issues, many regularization approaches have been proposed and developed in recent years, which includes Tikhonov-like regularization [6], [17], [36], total-variation regularization [3], [12], [26]–[28], high-order regularization techniques [44], and prior-based methods [32], [55]. Most of those existing regularization and prior-based techniques are handcrafted, meaning that they are loosely (if at all) related to the physical problem at hand.

Manuscript received May 12, 2019; revised October 8, 2019, December 26, 2019, and February 12, 2020; accepted February 26, 2020. Date of publication March 23, 2020; date of current version September 25, 2020. 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 of LANL under project number 20200061DR. *(Corresponding author: Youzuo Lin.)*

The authors are with Earth and Environmental Sciences, Los Alamos National Laboratory, Los Alamos, NM 87544 USA (e-mail: ylin@lanl.gov).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2020.2977635

Furthermore, all these solutions are developed under the physics-driven FWI framework. Hence, the expensive computational costs will be inherited and unavoidable.

More recently, with the successes of deep learning in computer vision community [18], [56], [58], researchers have developed various data-driven seismic FWI techniques [4], [34], [37], [40], [48], [49]. Data-driven frameworks take the waveform data as the input and directly output their corresponding velocity images. In this article, we study the generative adversarial network (GAN)-based method [11]. A GAN has been proven to be effective in the areas of photograph inpainting [18], [58], image denoising [51], super-resolution [24], image deblurring [23], and so forth. Motivated by these successes, we solve the FWI problem using a GAN. Specifically, our model consists of two parts: generator and discriminator. The generator is an encoder–decoder structure that maps the raw seismic waveform data to the velocity image. The discriminator is a convolutional neural network (CNN) designed to classify the real velocity image and fake velocity image. There are two major benefits using a GAN to solve our seismic-waveform-inversion problems. First, our model learns the regularization term directly from data through the discriminator and further imposes the learned regularization term to the generator. The regularization term is used to differentiate between the ground-truth velocity map and the generated velocity map. This type of GAN-based regularization has recently been discussed in [25] and yields supreme results for the computer-vision tasks. Second, our GAN-based inverse problem model is an end-to-end framework, which is similar to the image-to-image translation problem from computer vision [10], [19], which means that the velocity map can be output in real time once the training is completed.

Compared with the physics-driven methods, the major disadvantage of the data-driven methods is that they suffer from robustness and generalization issues. The deep neural network, which is trained on a specific data set, tends to perform worse on another data set. To alleviate the issue, we incorporate our data-driven method into the network-based deep transfer learning. Network-based deep transfer learning means the reuse of the network that is pretrained in the source domain and then transfer the network parameters and structures to the target domain. In our project, we apply a fine-tuning strategy to update the model's parameters, which means we first pretrain the neural network on a large data set and then employ the weights of the pretrained neural network as initialization when the new smaller data set becomes available.

To summarize, the main features of our work are as follows.

1) Our model transfers the inverse process of the physics-driven methods into an image-mapping problem. As a result, it can alleviate the local-minima and low computational efficiency issues.
2) Our model is built on a GAN structure, which consists of a generator and a discriminator.
3) The structure of the generator consists of an encoder and a decoder.
4) We combine the mean-absolute-error (MAE) loss with the mean-square-error (MSE) loss to improve the quality of the velocity images.

To validate the performance of our data-driven inversion, we tested our method on a couple of experimental test data sets. Through comparison, we show that our method yields better reconstruction accuracy than the encoder–decoder-based inversion method. We perform a series of experiments to demonstrate the robustness of our model, validating that our model does not just "memorize" the training data while it learns the intrinsic physics law from the training set. We also conduct additional experiments to demonstrate that the data-driven methods plus transfer learning are a feasible way to alleviate the generalization issue. We further test our data-driven method on the Society of Exploration Geophysicists (SEG)/EAGE salt model and show encouraging inversion results.

In the following sections, we first briefly provide the related work in Section II. We also describe the fundamentals of the physics-driven versus data-driven methods and deep neural networks (see Section III). We then develop and discuss our novel inversion method - VelocityGAN. Section IV describes the data we tested on, experimental setup, and experimental results we obtained. Finally, concluding remarks are presented in Section VI.

II. Related Work

A. Data-Driven Inverse Problems

FWI is a specific type of inverse problem. We first provide relevant literature in solving inverse problems from other domains. In particular, we focus on deep neural network-related techniques [1], [15], [21], [52], [53], [57]. In general, those different deep-learning-based methods for solving inverse problems can be categorized into four types [31]: 1) to learn an end-to-end regression with vanilla CNN; 2) to learn higher level representation; 3) to refine gradually the inversion procedure; and 4) to incorporate with analytical methods and to learn a denoiser. An interesting work under the first category is AUTOMAP, which was recently developed by Zhu *et al.* [57]. The authors developed an end-to-end reconstruction algorithm for MRI imaging, where the encoder consists of three fully connected networks to read in the sensor-domain data and the decoder consists of three additional convolutional and deconvolutional layers to yield the image-domain output. A common use of CNNs is to learn a compressed representation prior to constructing an output image. Several existing works use the effectiveness of autoencoders to learn relevant features to solve inverse problems in imaging. As an example, Zeng *et al.* [53] employed the autoencoder's representation-learning capability to learn useful representations of low-resolution and high-resolution images. A shallow neural network is then trained to learn the correspondence between the learned low-resolution representation and the high-resolution representation. In the third category, CNNs are used to learn a residual between two or more layers by the skip connection from the input of the residual block to its output. This network structure is particularly well suited to inverse problems such as image restorations when the input and output images share similar content. The work of Yao *et al.* [52] and Kim *et al.* [21] both belong to this category. Another type of research effort to solve the inverse problems using neural networks is to incorporate analytical

solutions. The work developed in Hammernik *et al.* [15] falls into this category. Hammernik *et al.* [15] reformulated a generalized compressed sensing reconstruction as a variational model, which is embedded in an unrolled gradient descent iterative scheme. Key parameters such as those used in the activation functions are learned through offline training procedures. In the inference stage, the previously learned model will be applied online to unseen data. Another example under this category is the one developed in Adler and Öktem [1]. They unrolled a proximal primal–dual optimization method and replaced the proximal operators using CNNs, and successfully applied to the computed tomography (CT) image-reconstruction problem.

B. Data-Driven FWI

In particular, in seismic waveform inversion, there are some recent developments in the data-driven techniques, which can be categorized into two groups: an end-to-end learning [4], [37], [48], [49] and low-wavenumber learning [34], [40]. The end-to-end strategy directly learns a mapping correspondence from seismic data to the velocity model. The low-wavenumber strategy learns low-wavenumber from data followed by traditional FWI iteration. Comparing these two strategies, the end-to-end-learning strategy is more aggressive, which usually requires much more complex network structures to account for the nonlinear nature of the FWI. Encouraging results have been recently reported in [49] due to significant amount of training sets that are used.

C. Deep Transfer Learning

A great number of deep transfer-learning methods are developed in recent years. There are mainly four types of deep transfer-learning approaches, which are instance-based deep transfer learning, mapping-based deep transfer learning, network-based deep transfer learning, and adversarial-based deep transfer learning [41]. Our work belongs to network-based deep transfer learning. Two types of network-based deep-transfer-learning methods are widely used in practical applications: fine-tuning and feature extraction. Feature extraction refers to the reuse of a pretrained model and only updates few-layer weights for the target domain. For example, Oquab *et al.* [33] reused the front layer trained on ImageNet to compute intermediate image representation for images in other data sets. Fine-tuning means all the model parameters for a new task are updated.

III. Inversion Models

We first present some overview of the governing physics equation (acoustic-wave equation), physics-driven FWI method, and data-driven inversion method in Section III-A. In Section III-B, we provide details on our VelocityGAN and its network structure. In Section III-C, we provide the loss functions of our VelocityGAN. In Section III-D, we discuss the connections to inverse and regularization theory.

A. Acoustic-Waveform Inversion: Physics-Driven Approach

1) Governing Physics—Wave Equation: Mathematically, the seismic acoustic-wave equation is

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

where $\rho(\mathbf{r})$ is the density at spatial location $\mathbf{r}$, $K(\mathbf{r})$ is the bulk modulus, $s(\mathbf{r}, t)$ is the source term, $p(\mathbf{r}, t)$ is the pressure wavefield, and t represents the time.

To simplify the expression, the forward modeling problems in (1) can be written as

$$P = f(\mathbf{m}) \tag{2}$$

where P is the pressure wavefield for the acoustic case, f is the forward acoustic-wave modeling operator, and $\mathbf{m}$ is the velocity model parameter vector. We use a time-domain stagger-grid finite-difference scheme to solve the acoustic-wave equation. Inference of unknown subsurface properties relies on indirect and limited geophysical measurements taken at or near the surface. Therefore, seismic FWI is extremely underconstrained and can be severely ill-posed. Various explicit regularization techniques have been developed to stabilize the computation. This regularized physics-driven seismic inversion can be posed as

$$E(\mathbf{m}) = \min_{\mathbf{m}}\left\{\|\mathbf{d} - f(\mathbf{m})\|_2^2 + \lambda\, R(\mathbf{m})\right\} \tag{3}$$

where $\mathbf{d}$ represents a recorded/field waveform data set, $f(\mathbf{m})$ is the corresponding forward modeling result, $\|\mathbf{d} - f(\mathbf{m})\|_2^2$ is the data misfit, $\|\cdot\|_2$ stands for the L_2 norm, λ is a regularization parameter, and $R(\mathbf{m})$ is the regularization term. The regularization term measures the "complexity" of the model $f(\mathbf{m})$ so that the minimization in (3) favors a simple solution that is consistent with the data. Explicit regularization techniques such L_1-norm [3], [12], [26], [27] or L_2-norm [9], [36], [44] have been developed for seismic inversion, but these regularizers do not fully express an expert's prior knowledge.

2) Data-Driven Approach: Different from the physics-driven methods, end-to-end data-driven methods transfer the minimization process into a mapping problem. The parameters of $\mathbf{m}$ are directly learned from

$$\mathbf{m} = g(\mathbf{d}) = f^{-1}(\mathbf{d}) \tag{4}$$

where $g = f^{-1}(\cdot)$ is the inverse operator of $f(\cdot)$. We can, therefore, obtain the loss function as follows:

$$g = \underset{g}{\operatorname{argmin}}\left\{\sum_{i=1}^{N}\|\mathbf{m}_i - g(\mathbf{d}_i)\|_2^2\right\} \tag{5}$$

where $(\mathbf{m}_i, \mathbf{d}_i)$ are N pairs of velocity map and the corresponding seismic data. Most of the existing end-to-end data-driven FWI methods use the encoder–encoder structure to learn the mapping function of g in (5) [4], [37], [48], [49].

B. VelocityGAN: Data-Driven Approach

The overall architecture of VelocityGAN is shown in Fig. 2. It consists of an image-to-image generator and a CNN discriminator. We discuss below the structure of the generator and the discriminator.

1) Generator: To better understand the network structure, we first recall the governing physics of our input seismic shot-gather (see Fig. 3) and output velocity map (see Fig. 4). In this article, the input seismic shot-gather is the combination of the acoustic waves received by different receivers and the visualization of a shot-gather is shown in Fig. 3. Specifically, there

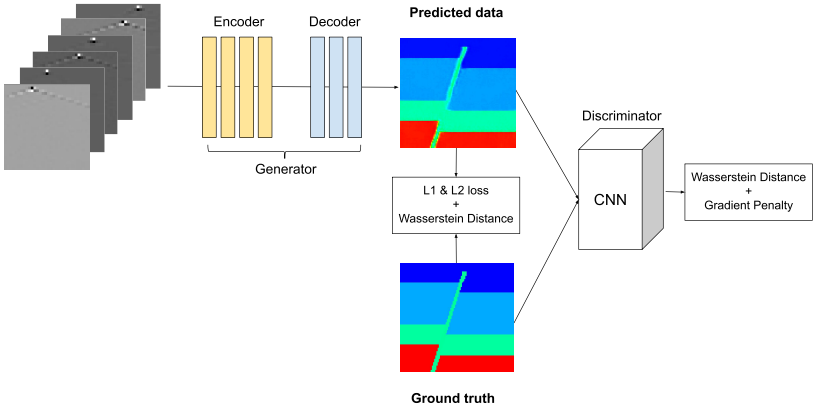


Fig. 2. Overall architecture of VelocityGAN. We apply an encoder–decoder structure as the generator ("Inversion Generator") and a CNN as the discriminator.

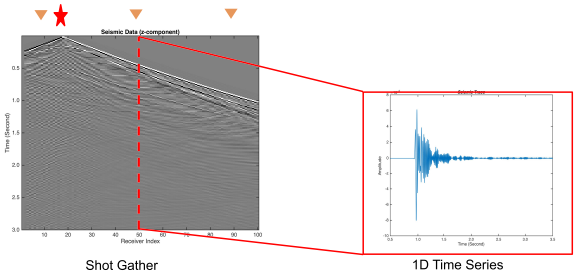


Fig. 3. Visualization of shot-gathers. The value in the horizontal direction represents different receivers. The value in the vertical direction represents the 1-D time-series signal. We pick out a 1-D time-series signal and present it in the red bounding box.

are three source functions and 32 receivers, which correspond to $s(\mathbf{r}, t)$ and $P(\mathbf{r}, t)$ in (1), respectively. The source function contains only P wave. Correspondingly, the seismic waveform data (in displacement) collected are a tensor with a dimension of $32 \times 1000 \times 6$, where the first dimension is 32 receivers, the second dimension of 1000 is the time sequence length of the waveform trace, and the third dimension of 6 represents the total two channels of the three source functions. As shown in Fig. 3, the shot-gather consists of 32 1-D time-series signals, and each signal contains a pulse that contains the information of a subsurface structure. Correspondingly, we also present several velocity images in Fig. 4, which is the output of our model. The dimension of the output velocity map is (m, n), where the first dimension of m stands for depth and the second dimension of n stands for horizontal offset. The value of each pixel in the velocity image stands for the absolute velocity value at each location. The grid spacing between the pixels is 5 m. Therefore, the total size of the velocity map in the real world is $5m \times 5n$ m. The linear geologic feature shown in the velocity map in Fig. 4 is the geologic fault.

As we discussed above, there are no direct spatial similarities between the input seismic shot-gather imagery and the output subsurface velocity map (see Figs. 3 and 4). Therefore, we do not penalize the mismatch between the input and the output like that in [38], [58]. In addition, since the height (1000) and width (32) of the input is unbalanced, we apply several convolutional layers with $k \times 1$ kernels to get the same height (32) and width (32). k means the length of the convolutional kernel in the height dimension. For the first convolutional layer, k is equal to 7. For convolutional blocks 1, 2, and 3, k is equal to 3. In these layers, a stride with a size of 2×1 will decrease the height. The particular structure of the generator is shown in Table I. To extract the waveform features of each receiver, nine convolutional layers with $k \times 1$ kernels are first deployed. Each convolutional layer is followed by a batch normalization layer and a Leaky ReLU layer. After the dimension of height is reduced to 32, 3×3 convolutional kernels with stride 2 are then added to encode the whole extracted features. In the last layer of the encoder, 8×8 convolutional kernels are used to eliminate the influence of spatial information. With regard to the decoder, it consists of five upconv blocks, a center cropping layer, and a convolutional layer. Each upconv block consists of a transposed convolutional layer, a batch normalization layer, and an activation layer. The transposed convolutional layers are applied to increase the height and width dimensions of the image and decode the extracted features. The convolutional layer is designed to map the features into the same dimension with the ground-truth labels. The center cropping layer is used to crop the feature maps into the desired dimension. To limit the value of the output into a specific range, the center cropping layer is followed by a Tanh layer.

2) Discriminator: Similar to Radford *et al.* [35], we adapt our discriminator from a CNN architecture. In particular, it consists of five convolution blocks, a global average pooling layer, and fully connected layers. Each convolutional block involves a combination of Convolutional, BatchNormalization, LeakyReLU, and MaxPooling layers. We apply the "PatchGAN" classifier [18] in the discriminator to capture local style statistics. We set the patch size as 4 and calculate the mean loss value of all patches in an image. In our data set, velocity maps can be a rather different one from the others (such as different tilting angles, layer thicknesses, and layer velocities). However, if we focus only on the local information, their geological faults and interfaces share some similarities (the geological faults and interfaces always have a drastic change in velocity). Therefore, "PatchGAN" is more suitable than "GlobalGAN" for our task.

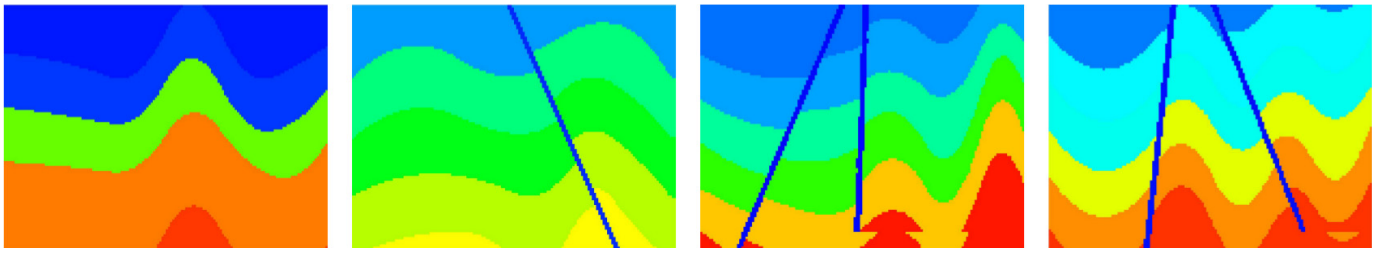


Fig. 4. Visualization of velocity images, CurvedData. The number of geologic faults (the linear features shown in the velocity map) varies with different subsurface structures.

TABLE I
GENERATOR ARCHITECTURES OF VELOCITYGAN

Layers	Output Size	The Generator of VelocityGAN
conv1	500×32	7×1 conv, channel 32, stride 2×1
conv block1	125×32	$\begin{bmatrix} 3\times1, 64, 2\times1 \\ 3\times1, 64, 1\times1 \end{bmatrix}\times 2$
conv block2	32×32	$\begin{bmatrix} 3\times1, 128, 2\times1 \\ 3\times1, 128, 1\times1 \end{bmatrix}\times 2$
conv block3	8×8	$\begin{bmatrix} 3\times3, 256, 2\times2 \\ 3\times3, 256, 1\times1 \end{bmatrix}\times 2$
conv2	1×1	8×8, 512, 1×1
upconv block1	10×7	$\begin{bmatrix} 10\times7 & \text{deconv}, 512, 2\times2 \\ 3\times3 & \text{conv}, 512, 1\times1 \end{bmatrix}$
upconv block2	20×14	$\begin{bmatrix} 4\times4, 256, 2\times2 \\ 3\times3, 256, 1\times1 \end{bmatrix}\times 2$
upconv block3	40×28	$\begin{bmatrix} 4\times4, 128, 2\times2 \\ 3\times3, 128, 1\times1 \end{bmatrix}\times 2$
upconv block4	80×56	$\begin{bmatrix} 4\times4, 64, 2\times2 \\ 3\times3, 64, 1\times1 \end{bmatrix}\times 2$
upconv block5	160×112	$\begin{bmatrix} 4\times4, 32, 2\times2 \\ 3\times3, 32, 1\times1 \end{bmatrix}\times 2$
conv3	160×112	3×3, 1, 1×1
crop layer	150×100	—

C. Loss Function

A Wasserstein GAN (WGAN) with gradient penalty [14] has been proven to be robust of a wide variety of generator architectures. Considering the modified structure in our generator, we use Wasserstein loss with a gradient penalty to distinguish the real and generated velocity maps. The loss function of the discriminator is formulated as

$$L_d = \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g} D(\tilde{x}) - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r} D(x) + \lambda \mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}} [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \quad (6)$$

where $\mathbb{P}_g$ means the distribution of the velocity map, which is predicted by the generator of VelocityGAN, $\mathbb{P}_r$ is the distribution of the ground-truth velocity map, and $\mathbb{P}_{\hat{x}}$ is the random samples from both $\mathbb{P}_g$ and $\mathbb{P}_r$. $D(\cdot)$ represents the discriminator of VelocityGAN. Correspondingly, $\mathbb{E}D(\cdot)$ represents the expected value of the output, which is generated by the discriminator. In practical implementation, we use the average value of output to approximate the expectation.

For the generator, we want the predicted velocity map to not only yield a discriminator loss but also reveal the accurate information of the geological structure. Therefore, the loss function is a combination of adversarial loss and content loss. Consistent with (6), the adversarial loss is $-\mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g} D(\tilde{x})$. The content loss is set as a combination of MAE and MSE. In our experiments, we observe that MSE loss is good at capturing the geological faults, while the MAE loss performs better on revealing the geological interfaces. Therefore, the loss function of the generator is formulated as

$$L_g = -\mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g} D(\tilde{x}) + \frac{\lambda_1}{w\cdot h}\sum_{i=1}^{w}\sum_{j=1}^{h}|\tilde{v}(i,j) - v(i,j)| + \frac{\lambda_2}{w\cdot h}\sum_{i=1}^{w}\sum_{j=1}^{h}(\tilde{v}(i,j) - v(i,j))^2 \quad (7)$$

where w and h are the width and height of the velocity map, respectively, $v(\cdot)$ represents the real pixel value of the velocity map, and $\tilde{v}(\cdot)$ means the predicted pixel value. λ_1 and λ_2 are the hyperparameters to control the relative importance of the two loss terms. In our experiments, we pick λ_1 and λ_2 by balancing the MAE and MSE losses during the training process. The specific values of λ_1 and λ_2 are discussed in Section IV-A2.

D. Connection to Regularization Theory

There is a close connection of the GAN with the regularization techniques used in the inverse problems [25]. To see the connection, we can rewrite (7) as

$$g = \mathop{\text{argmin}}_{g}\left\{\lambda_1\sum_{i=1}^{N}\|\mathbf{m}_i - g(\mathbf{d}_i)\|_1 + \lambda_2\sum_{i=1}^{N}\|\mathbf{m}_i - g(\mathbf{d}_i)\|_2^2 - \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g} D(\tilde{x})\right\} \quad (8)$$

where the target mapping, g, in (7) can be interpreted as either the inversion operator according to (8) or the generator

according to (7). Similarly, the term of $-\mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g} D(\tilde{x})$ not only is an adversarial loss term but also plays the role of regularization, which is learned from the training data. The content-loss terms (MAE and MSE) in (7) or (8) promote the velocity-map consistency. In particular, we will use the GANs to learn a classifier to discriminate between the distribution of the ground-truth velocity maps and the generated velocity maps. This discriminator effectively penalizes the velocity models that do not "look like" the velocity models that are used for training. The usual approach to alleviate the ill-posedness of inverse problems is to incorporate prior knowledge with a regularization term that penalizes solutions that are inconsistent with this prior knowledge. Most of the existing regularization techniques employ generic functions (e.g., L_1-norm or L_2-norm penalties on coefficients) that are loosely (if at all) related to the physical problem at hand. On the other hand, regularization learned from data can be more effective and customized for the problem.

IV. Experiments

We introduce the data sets and the training details in Section IV-A. We discuss the experimental settings in Section IV-B. Following that, we compare and analyze the results of different methods. Finally, we present the generalization experiments and provide a feasible way to solve the generalization issue.

A. Data Sets and Training Details

1) Data Sets: In the practical applications, velocity models are estimated by the physics-driven methods (usually an optimization algorithm). It can be unrealistic and expensive to obtain a large-scale data set, which consists of seismic waveforms and velocity models. To verify the efficacy of our VelocityGAN, we, therefore, generate a data set including velocity images and corresponding seismic waveform data generated using (1). The velocity images that we generated are varied with different tilting angles, layer thicknesses, and layer velocities. They can be a good representation of the real velocity images [30]. Although our VelocityGAN is validated using the seismic acoustic-wave equation, the method developed in this article can be directly adapted to the elastic scenario as well.

Our experimental data set contains 50 000 velocity models with 150×100 dimensions, along with their corresponding seismic waveform. This data set contains complicated geological layers in a velocity image. Furthermore, most geological layer interfaces are curved. We name this data set "CurvedData." For this data set, three common-shot-gathers of the synthetic seismic data with 32 receivers are posed at the top surface. We use a Ricker wavelet with a center frequency of 50 Hz as the source time function and a staggered-grid finite-difference scheme with a perfectly matched layered absorbing boundary condition to generate 2-D synthetic seismic reflection data [42], [54]. The synthetic trace at each receiver is a collection of time-series data of length 1000. Hence, the input size is (32, 1000, 6), where 1000 is the time sequence length, 32 is the number of receivers, and 6 comes from three shots and two channels.

We also create two small data sets for the generalization experiments. The velocity images of these two data sets are similar to the images in CurvedData except for the number of faults. We adjust the number of geological faults to zero or two in order to evaluate our VelocityGAN in a more generalized condition. We use 2-Fault CurvedData and 0-Fault CurvedData to represent these two data sets in the following parts. There are 2000 pairs of velocity images in 2-Fault CurvedData and 1000 pairs of velocity images in 0-Fault CurvedData.

2) Training Details: For each data set, we randomly select 20% data as the testing set and 10% data as the validation set to adjust the hyperparameters. We use the remaining images as the training set. The input of our model is normalized to range $(-1$–$1)$. Constrained by the memory of the GPU, we set the size of mini-batch to 50. Following the optimization strategy of [5], we perform five gradient descent steps on the discriminator and then perform one step on the generator. The learning rate of our Adam [22] optimizer is set to 10^{-4} in the first epoch. We linearly decay the learning rate to 0 over the remaining epochs. For the discriminator loss [see (6)], we choose λ as 10. For the generator loss [see (7)], λ_1 and λ_2 are set to 50 and 100 in CurvedData. All our models are implemented on a single GTX 1080-Ti using the PyTorch framework.

B. Experimental Settings

Velocity image-generation experiments are conducted to evaluate the effectiveness of our VelocityGAN. We choose the following algorithms from both physics-driven and data-driven methods as our baselines.

1) *FWI-MTV [26]:* A modified total-variation regularization (MTV) is used as a regularization term in the FWI optimization process. MTV is designed to preserve sharp interfaces in the piecewise constant structures.
2) *FWI-MTV (Plus):* An enhanced version of FWI-MTV. We increase the number of sources to 32. With more information from the receiver, the accuracy of reconstruction results tends to be improved.
3) *Generator:* The generator of VelocityGAN. We apply fully convolutional layers as our encoder and transposed convolutional layers with convolutional layers as the decoder.

Specifically, FWI-MTV [26] and FWI-MTV (plus) are physics-driven methods. A generator with various loss functions and VelocityGAN with various loss functions are data-driven methods.

Similar to the existing works [7], [8] on depth estimation, we adopt the following metrics to evaluate the accuracy of velocity image reconstruction.

1) *Mean Absolute Error (MAE):* $\text{MAE} = (1/n)\sum_i |m_i - m_i^*|$.
2) *Mean Relative Error (rel):* $\text{rel} = (1/n)\sum_i (|m_i - m_i^*|/m_i^*)$.
3) *Mean* log*10 error(*log*10):* $\log10 = (1/n)|\log_{10} m_i - \log_{10} m_i^*|$.
4) *The Percentage of m_i (acc.):* $\text{acc} = \max((m_i^*/m_i), (m_i/m_i^*)) < t$.

TABLE II
QUANTITATIVE RESULTS OF VELOCITY IMAGE RECONSTRUCTION ON CURVEDDATA

	mae	rel (10^{-3})	log10 (10^{-3})	acc. (t=1.01)	acc. (t=1.02)	acc. (t=1.05)	acc. (t=1.10)
FWI-MTV [26]	136.80	56.25	24.44	35.56%	47.30%	67.69%	81.20%
FWI-MTV (plus)	79.01	33.58	14.15	45.18%	60.02%	80.52%	92.28%
Generator-l1	83.03	36.77	14.97	40.05%	59.16%	84.59%	93.41%
Generator-l2	80.93	34.43	14.64	34.29%	56.64%	84.69%	94.16%
Generator	85.69	35.99	15.44	35.99%	56.14%	82.80%	93.35%
VelocityGAN-l1	79.08	34.76	14.46	**42.30**%	**62.70**%	85.70%	93.26%
VelocityGAN-l2	76.19	**32.39**	13.91	32.39%	59.60%	86.17%	**94.46**%
VelocityGAN	**75.85**	32.40	**13.83**	40.64%	62.54%	**86.33**%	93.98%

For qualitative experiments, we present several velocity image samples and vertical velocity profiles to provide an intuitive comparison. It is worthwhile to mention that in CurvedData, we include a small geologic fault in the velocity model. Geologic faults play an important role in siting the wells in subsurface applications because of its high permeability property. However, it can be technically challenging to image a geologic fault zone due to the limited imaging resolution and data coverage. We will compare our method with others not only in the overall reconstruction quality but also in the local region such as fault zone. In addition, we also compare the implementation time between the physics-driven methods and the data-driven methods.

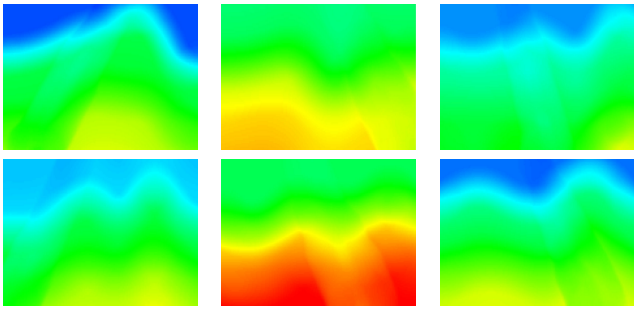


Fig. 5. Starting models of FWI-MTV and FWI-MTV (plus) methods. (Top left to Bottom right) Their ground truth corresponds to the most left column in Fig. 6.

C. CurvedData

In the real world, the geological layers usually yielded irregular shape. To address the curved layer estimation, we create a challenging data set—CurvedData. Using these data, geological faults will disappear with the constraint of MAE loss. MSE loss is good at revealing the geological faults but does not perform well on reconstructing the layer interfaces. Therefore, we use a combination of the MAE and MSE losses to generate more accurate velocity images. In our quantitative experiments, we not only compare VelocityGAN with the physics-driven models but also do ablation study on the data-driven methods.

1) Quantitative Results: Table II shows the quantitative results of our experiments on CurvedData. We can see that our proposed models perform much better than the physics-driven models. Based on FWI-MTV, we increase the number of sources to achieve comparable results with the data-driven methods. The number of sources in FWI-MTV (plus) is 32, while the data-driven methods only use three sources. Under this circumstance, VelocityGAN still performs better than FWI-MTV (plus) in most measurements, except for acc ($t = 1.01$). In addition, we conduct the ablation study on the loss function. We observe that VelocityGAN with a combination of MAE and MSE losses can get better prediction results than a single loss. Though the VelocityGAN with a single loss achieves relatively higher scores in some measurements such as rel and acc ($t = 1.10$), the VelocityGAN with a combination of MAE and MSE losses obtains a better tradeoff under all the metrics. The quantitative experiments on CurvedData validate that the generator structure, the adversarial training strategy, and the combination of losses are all useful improvements and can boost the reconstructed accuracy.

2) Qualitative Results: We illustrate the reconstructions of the velocity images using different baseline methods in Fig. 6. The starting models of the physics-driven methods are shown in Fig. 5. Consistent with our discussion on the loss function, VelocityGAN with the MAE loss is good at handling with the boundary of geological layers; however, it ignores the geological faults and high-velocity areas. VelocityGAN with MSE loss can generate geological faults, but the boundary of geological layers is fuzzy. VelocityGAN with a combination of MAE and MSE can achieve a better tradeoff between the quality of the geological layer interfaces and faults. An interesting finding in our experiments is that under the same loss function and network structure, VelocityGAN with MAE loss reveals correct geological fault, though it is fuzzy in some particular areas. In contrast, generator with MAE loss totally misses the geological faults. This phenomenon further demonstrates the effectiveness of the adversarial training strategy.

We present the vertical velocity profile of CurvedData in Fig. 7. VelocityGAN still outperforms the physics-driven methods on the accuracy of vertical velocity. For VelocityGAN with different loss functions, we observe that the framework with a combination of MAE and MSE performs well in both low-velocity regions and high-velocity regions. VelocityGAN with a single loss sometimes misses the geological fault zones in the low-velocity regions. For example, in the second row and the middle column of Fig. 7, both VelocityGAN-L1

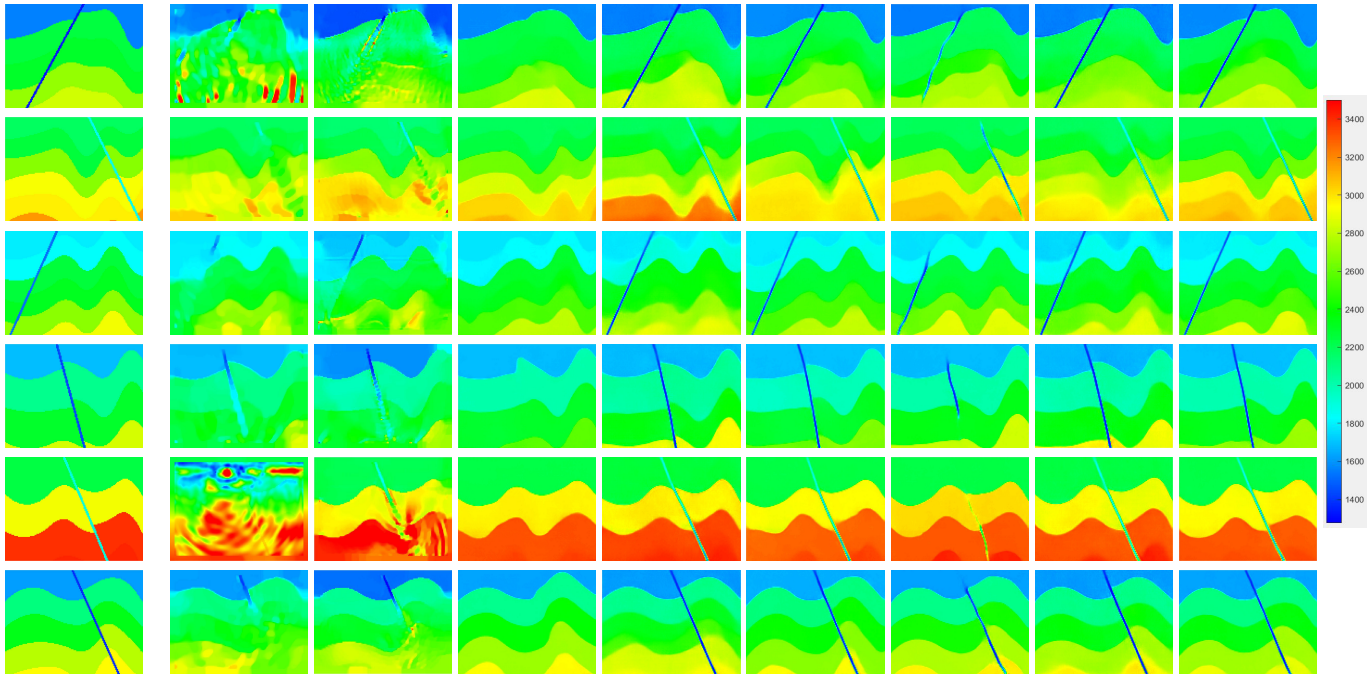


Fig. 6. Examples of different methods on CurvedData. The images from the leftmost columns to right are ground-truth, reconstruction results using FWI-MTV [26], FWI-MTV (plus), Generator-L1, Generator-L2, Generator, VelocityGAN+L1, VelocityGAN+L2, and VelocityGAN. Our VelocityGAN yields the most accurate reconstructed velocity images among both the physics-driven methods and data-driven baselines. The experiment results substantiate the effectiveness of the adversarial training strategy and a combined loss.

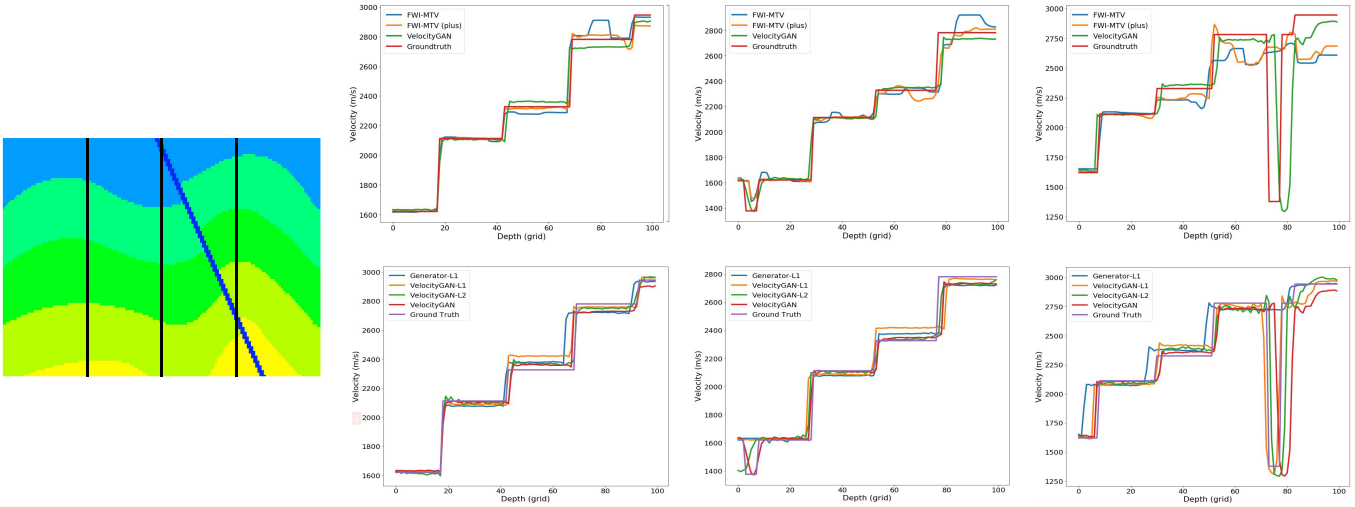


Fig. 7. Vertical velocity profiles of different positions on CurvedData. (Left to right) Positions are 40, 75, and 110, respectively. We compare VelocityGAN with the physics-driven methods in the first row and compare VelocityGAN with different loss functions in the third row.

and VelocityGAN-L2 fail to reconstruct the geological fault zone between position 0 and 10. In contrast, VelocityGAN with a combination of L1 (MAE) and L2 (MSE) loss reveals the geological fault well. In the second row of Fig. 7, we also compare VelocityGAN-L1 with Generator-L1. Though both of them do not perform well in the low-velocity regions, VelocityGAN-L1 is able to reveal a geological fault in the high-velocity regions, which is better than Generator-L1. We attribute this phenomenon as the influence of the adversarial training strategy.

According to the aforementioned comparison, we conclude that our VelocityGAN yields a more accurate reconstruction of the velocity image in obtaining both global and location geological features.

D. Generalization Experiments

In machine learning theory, the test error of a trained model on unseen data is given as [16]

$$\text{Error}_{\text{test}} = \text{Error}_{\text{train}} + \text{Error}_{\text{Generalization}} \tag{9}$$

where $\text{Error}_{\text{train}}$ is the training error and $\text{Error}_{\text{Generalization}}$ is the generalization error. With a large amount of training data and reasonable loss function, we can usually control the training

TABLE III
QUANTITATIVE RESULTS OF VELOCITY IMAGE RECONSTRUCTION ON 0-FAULT CURVEDDATA

	mae	rel (10^{-3})	log10 (10^{-3})	acc. (t=1.01)	acc. (t=1.02)	acc. (t=1.05)	acc. (t=1.10)
FWI-MTV [26]	49.83	19.09	8.52	59.41%	73.99%	88.33%	96.31%
FWI-MTV (plus)	31.20	11.36	5.13	75.60%	83.77%	95.28%	97.89%
VelocityGAN [38]	154.1138	67.19	28.78	11.10%	21.36%	47.19%	76.15%
VelocityGAN-org [39]	82.94	34.06	14.47	31.14%	53.54%	80.24%	92.90%
VelocityGAN-TF [49]	62.94	25.83	11.10	40.74%	63.00%	87.29%	95.85%

TABLE IV
QUANTITATIVE RESULTS OF VELOCITY IMAGE RECONSTRUCTION ON 2-FAULT CURVEDDATA

	mae	rel (10^{-3})	log10 (10^{-3})	acc. (t=1.01)	acc. (t=1.02)	acc. (t=1.05)	acc. (t=1.10)
FWI-MTV [26]	147.02	63.49	27.19	**31.66**%	45.61%	66.01%	79.32%
FWI-MTV (plus)	144.19	62.60	26.68	30.07%	42.68%	63.51%	80.22%
VelocityGAN [38]	242.33	107.07	45.14	7.90%	14.82%	32.83%	55.33%
VelocityGAN-org [39]	128.14	56.26	23.57	29.58%	47.25%	71.69%	86.39%
VelocityGAN-TF [49]	**115.87**	**53.10**	**21.48**	30.56%	**49.32**%	**75.87**%	**88.30**%

error, while the generalization error will then dominate the test error. It is well known that deep neural networks are overparameterized, meaning there are a significantly larger number of parameters than the amount of training data. Minimizing the same loss function might lead to multiple global minima, which all minimize the training error, but some of them might not generalize well. Conventionally, in the machine-learning community, cross-validation techniques are usually used to measure the test error. However, in our problem, cross-validation may be misleading due to the fact that no matter how to split the data, all the training, validation, and test data come from the same distribution. Therefore, we analyze the generalizability of VelocityGAN by studying its performance using specially designed test sets, which are inspired by actual field experiments.

In CurvedData, all velocity images contain one fault. To conduct the generalization experiments, we generate extra velocity models and their corresponding seismic data as our transfer-learning data. Specifically, the transfer-learning data include 0-Fault CurvedData and 2-Fault CurvedData. Based on these two data sets, we compare the reconstruction results of the physics-driven methods, VelocityGAN, which is trained on CurvedData (VelocityGAN-org), VelocityGAN with transfer learning (VelocityGAN-TF), and VelocityGAN without transfer learning (VelocityGAN). The specific procedure of the transfer-learning strategy is that we first train VelocityGAN on CurvedData and save the model weights. We then fine-tune the VelocityGAN weights on 0-Fault CurvedData or 2-Fault CurvedData.

The quantitative results of the generalization experiment are presented in Tables III and IV. In Table III, VelocityGAN-TF outperforms VelocityGAN and VelocityGAN-org. FWI-MTV (plus) gets better results than VelocityGAN-TF in 0-Fault CurvedData. For 2-Fault CurvedData, VelocityGAN-TF achieves the best quantitative results of all these models. We attribute the reason as the size of the training data set. For 0-Fault CurvedData, there are 700 pairs of velocity images and seismic data as the training set. In contrast, 2-Fault CurvedData has 1400 pairs of velocity images and seismic data as the training set, which is two times larger than 0-Fault CurvedData. According to the experiments, we can conclude that the performance of the data-driven methods depends on the size of the training set. A bigger data set can achieve better results. Overall, in our data set, the performance of VelocityGAN-TF is competitive with the FWI-MTV (plus) while outperforming all others. It should be mentioned that FWI-MTV (plus) uses extra information (32 shots) from the seismic data. In contrast, our model only needs three shots to achieve comparable results.

In addition, we compare the visual appearance of our models on 0-Fault CurvedData in Fig. 8. Consistent with the quantitative results, VelocityGAN, which is totally trained on 0-Fault CurvedData, performs worst among these models. It is because the size of 0-Fault CurvedData is not enough to train a good deep neural network. However, the visual appearance of VelocityGAN-org and VelocityGAN-TF is much better than FWI-MTV and FWI-MTV (plus). For the physics-driven approaches, especially for FWI-MTV, there are many oscillations in the deep region and high-velocity areas, which can be observed in the second and third rows, as shown in Fig. 8. The geological interfaces and faults that are generated by the data-driven methods are cleaner and sharper. It may be because we only calculate the average loss of the four selected images for the physics-driven methods; the quantitative results can be misleading. Compared with VelocityGAN-TF, we can clearly observe that VelocityGAN-org still contains parts of geological faults. The comparison further substantiates that the transfer-learning strategy can fine-tune the deep neural network effectively.

We randomly select six pairs of velocity images from 2-Fault CurvedData and present their reconstruction results in Fig. 9. As shown in the figure, VelocityGAN-TF per-

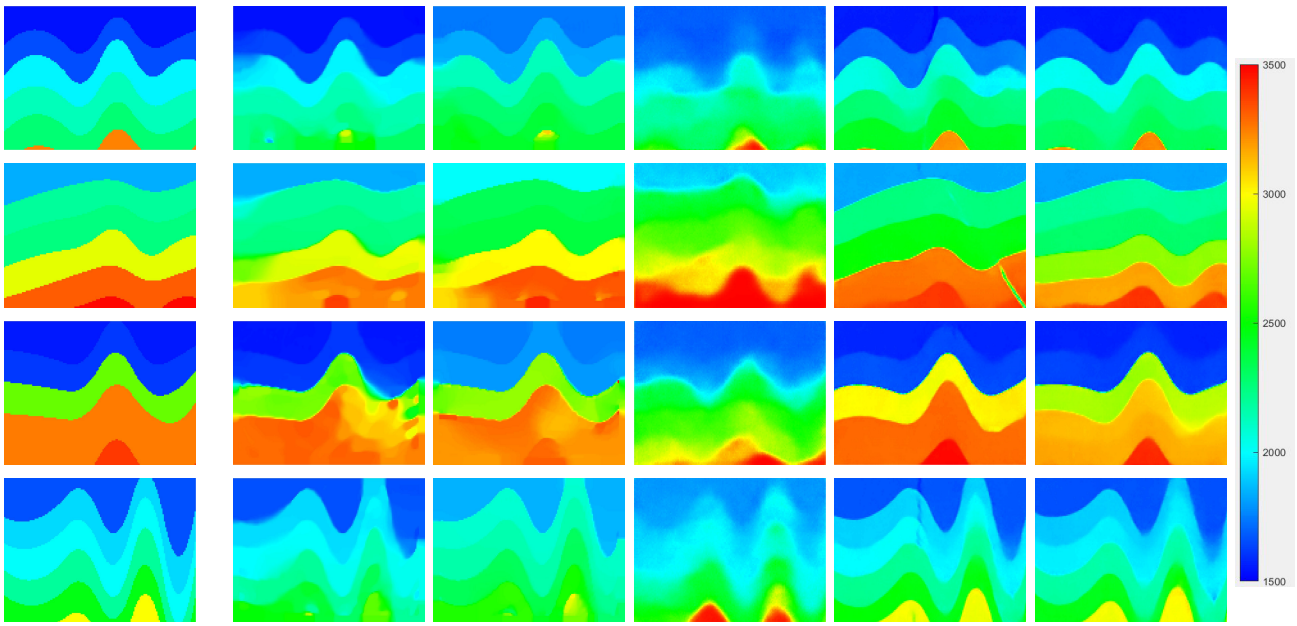


Fig. 8. Examples of different methods on 0-Fault CurvedData. (Left to right) Ground-truth, reconstruction results using FWI-MTV, FWI-MTV (plus), VelocityGAN, VelocityGAN-org, and VelocityGAN-TF.

forms slightly better than VelocityGAN-org, while much better than the other baselines including FWI-MTV (plus). In both 0-Fault CurvedData and 2-Fault CurvedData, VelocityGAN-org demonstrates its generalization ability to some extent. For example, when the distance between two faults is relatively large (the first four rows of Fig. 9), VelocityGAN-org and VelocityGAN-TF are able to locate correctly two faults, though some of them are fuzzy and unclear. When the distance is relatively small (the fifth and sixth rows of Fig. 9), VelocityGAN-org and VelocityGAN-TF cannot reconstruct the correct faults. In this condition, physics-driven approaches, FWI-MTV and FWI-MTV (plus), do not perform well either.

To summarize the experiments on 0-Fault CurvedData and 2-Fault CurvedData, we conclude that our VelocityGAN has a generalization ability to some extent. For instance, VelocityGAN, which is trained only on one fault velocity images, can also output the velocity images that have zero or two faults. With the transfer-learning strategy, VelocityGAN can further improve its generalization effect, which is competitive with the physics-driven methods.

E. More Realistic Application—Test on SEG/EAGE Salt Model

Though VelocityGAN performs well on our synthetic data set, we want to test it on a more realistic test set. In particular, we pick the SEG/EAGE [2] salt data as our test data, since the accuracy of the salt dome velocity can be critical to the subsalt regions (shown in Fig. 10). On the other hand, salt dome inversion can be notoriously challenging for traditional seismic inversion [47]. The SEG/EAGE salt data that we use to test our model are the same as those in [50]. The data set consists of 1700 simulated velocity maps that are generated by Yang and Ma [50] and 140 unique 2-D velocity maps extracted from different locations in the original 3-D SEG/EAGE salt velocity model. The velocity values of the 1700 simulated models range from 2000 to 4500 m/s. In comparison, the values of the 140 salt velocity models range from 1500 to 4482 m/s. We generated seismic data similar to Yang and Ma [50]. For each velocity map, 29 sources and 301 receivers are evenly distributed on the surface. A Ricker wavelet of 25 Hz is used as a source function. Shot-gathers are simulated using the same seismic modeling tool as previous tests.

Unlike 0-Fault or 2-Fault CurvedData, we cannot directly apply the parameters trained on CurvedData as the initial weights. GANs use implicit probability density estimation to generate data, which means it assumes the observed data obey an unobservable probability density function. Since the implicit probability distribution of these two data sets is different, the parameters trained on CurvedData will not perform well on the SEG/EAGE data. Therefore, our model is trained on the 140 salt velocity images. Specifically, we randomly select 130 velocity maps as our training set and ten velocity maps as the testing set. We adapt the convolutional layers to fit the dimension of the SEG salt velocity models, while the main structure is still similar to the generator of VelocityGAN. Following the training details on our CurvedData, we normalize both the seismic data and velocity models to range $(-1 \sim 1)$. The initial learning rate is set to 10^{-4} and is linearly decayed to 0 over 512 training epochs.

We present our results in Fig. 11. As shown in the figure, our VelocityGAN captures the high-velocity salt dome by comparing it with the ground truth. In addition, though some information is missed, our model can still accurately reveal most of the velocity layers. These results validate that it is promising to transfer our data-driven model to the real-world data.

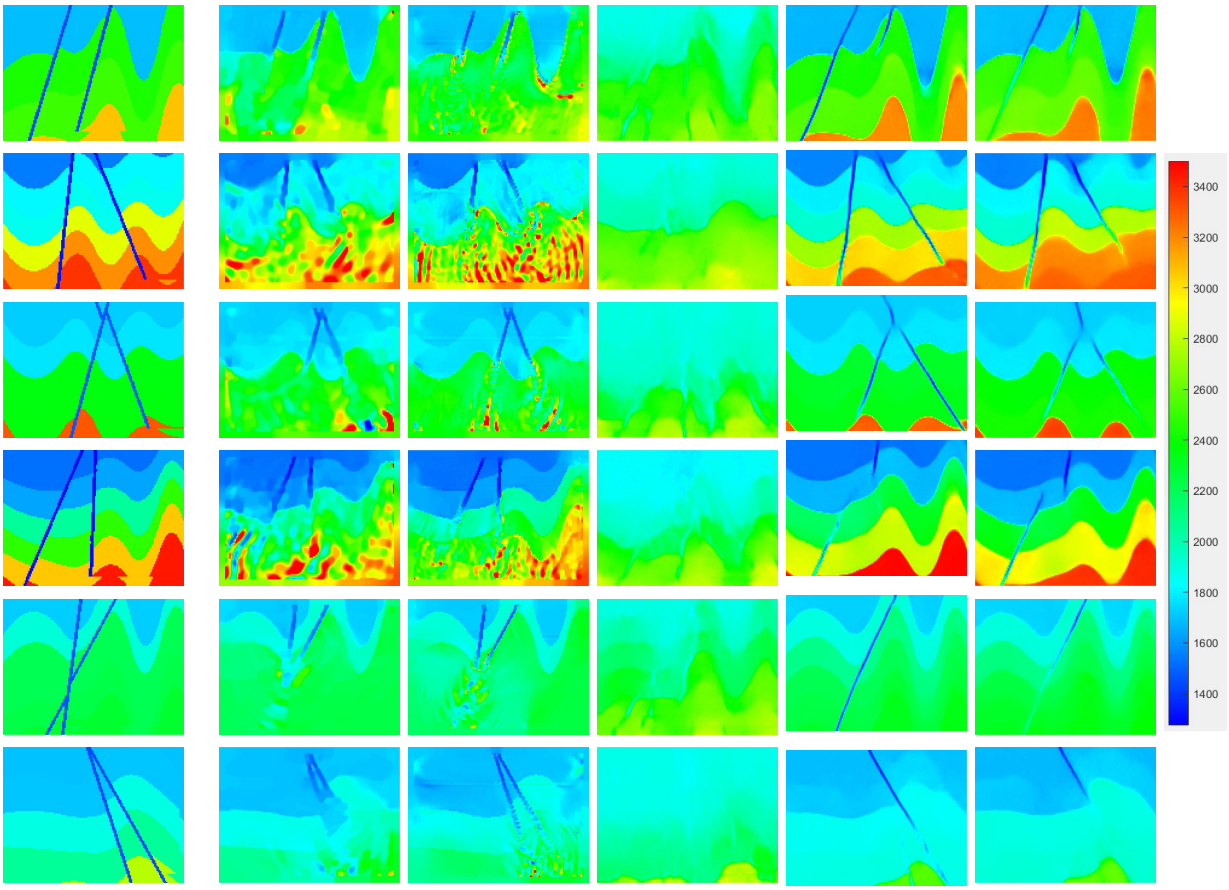


Fig. 9. Examples of different methods on 2-Fault CurvedData. (Left to right) Ground-truth, reconstruction results using FWI-MTV, FWI-MTV (plus), VelocityGAN, VelocityGAN-org, and VelocityGAN-TF.

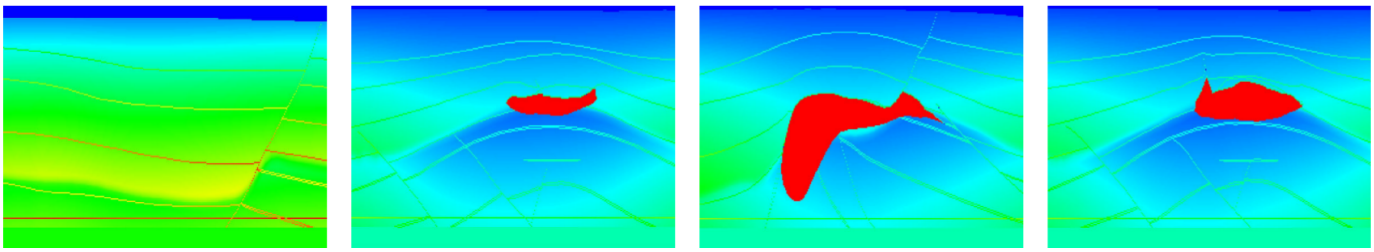


Fig. 10. Visualization of velocity images and synthetic velocity maps from the SEG/EAGE data set (data courtesy: Yang and Ma [50]).

V. Discussion and Future Work

A. Comparison With Global Optimization Techniques

Our VelocityGAN method is fundamentally different from those global optimization techniques (such as Markov chain Monte Carlo (MCMC) or particle swarm optimization approaches) even though they both require a large number of forward simulations at some step.

The motivation behind our VelocityGAN can be summarized as "offline training and online inversion." It means that the training procedure, which is the most expensive step, should be implemented offline. The training stage includes two tasks: 1) generating a large number of simulations as the training data set and 2) training our neural networks until they converge. With the VelocityGAN fully trained, we can then apply it to much other unknown data efficiently without training (or only required some fine-tuned using the transfer-learning technique). According to our tests, the computation in the inversion stage is usually trivial comparing with the training stage and it requires no simulation.

For any global optimization techniques, a large number of simulations are always required when inverting a new unknown data set. This is unavoidable due to the nature of global optimization.

In comparison with our VelocityGAN, the computational cost of these two categories methods may be comparable when inverting only one unknown data set. When inverting many unknown data sets, our VelocityGAN can be significantly efficient than global optimizations.

B. Future Work

1) Dynamic Network Structure: In this article, we mainly focus on discussing generalization on various kinds of velocity images, such as a different number of faults, simula-

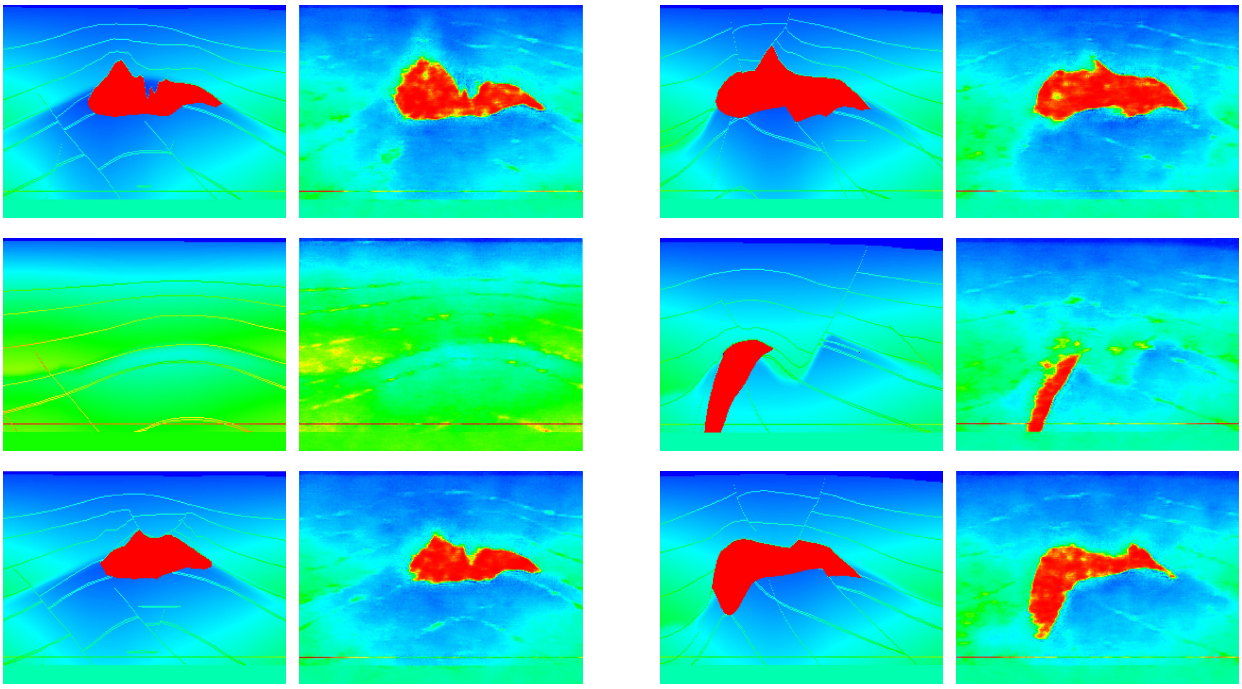


Fig. 11. Illustration of the inversion results using the SEG/EAGE salt data set. We show six inversion results from the testing set to demonstrate that our model can predict velocity maps with varying salt dome shapes. (Left column) Ground truth image. (Right column) Inverted image. The colormap used ranges from the minimum value to the maximum value of each pair of velocity maps.

tion images, real-world images, and so on. A limitation of data-driven inversion methods is that the dimension and size of the input are fixed. Under this circumstance, attributes like the number of shots and the distance between the shots are also fixed. In the future, we would like to explore the possibility of designing a dynamic network structure that can adapt some of these attributes automatically to improve further the robustness and make it applicable to more problems.

2) Application to Field Seismic Data: Our transfer-learning techniques developed in this article would work for the scenario where a sufficient number of labels are available in both the source domain and target domain. However, as for typical inverse problems when applying to field measurements, there is usually no or very few labels existed in the target domain. In the learning theory, such a problem belongs to transductive learning, which is a much more challenging problem to solve. In order to improve the robustness and generalization ability of a transductive learning model, we will need to address two issues: design of the training data set and incorporation of physics into training. Similar to other machine-learning problems, the training data set is critical to prediction accuracy. When applying our data-driven inversion techniques to field data, it is important to reflect some prior information (such as the number of layers, velocity range, and so on) when designing our training data set. That prior information can be usually obtained through other much less expensive geophysical measurements such as logging data. A unique characteristic of our problem is the governing physics, i.e., the intrinsic correspondence between the velocity maps and the seismic measurements. It has been demonstrated in literature [20] that the incorporation of physics into learning procedures will significantly improve the generalization ability of the prediction model. We will investigate how to incorporate wave physics into training to enhance the robustness.

VI. Conclusion

We develop a data-driven method and we call it "VelocityGAN," to solve the seismic FWI problem. We design a modified encoder–decoder structure as the core block of an image-to-image target. Based on the encoder–decoder structure, a conditional adversarial training strategy with an improved loss function is applied to boost further the reconstruction of the velocity images. Compared with the physics-driven methods, VelocityGAN is a more promising tool for subsurface velocity estimation, because it can alleviate the need for starting models (related to cycle-skipping issue) and expensive computational cost issues. Since VelocityGAN is trained on a data set that consists of 50 000 velocity maps, we think that it can also alleviate the local-minima issue. Compared with the physics-driven method, which is iterated based on a single velocity map, the information from various velocity maps can help the neural network avoid the local minima to some extent. We conduct quantitative and qualitative experiments to demonstrate the effectiveness and efficiency of our VelocityGAN from various aspects. The results substantiate that our model outperforms both the physics-driven methods and the selected deep-learning baselines. Furthermore, we also provide extensive experiments to discuss the generalization effectiveness of VelocityGAN. According to the results, we conclude that VelocityGAN has the basic generalization ability and can be improved by the transfer-learning strategy.

ACKNOWLEDGMENT

The computation was performed using the high-performance computing facilities made available by the LANL's Institutional Computing Program. They would also like to thank Dr. J. Ma for sharing his SEG salt data [50] that would allow them to test their algorithm.

REFERENCES

[1] J. Adler and O. Oktem, "Learned primal-dual reconstruction," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1322–1332, Jun. 2018.

[2] F. Aminzadeh, N. Burkhard, L. Nicoletis, F. Rocca, and K. Wyatt, "SEG/EAEG 3-D modeling project: 2nd update," *Lead. Edge*, vol. 13, no. 9, pp. 949–952, Sep. 1994.

[3] A. Y. Anagaw and M. D. Sacchi, "Full waveform inversion with total variation regularization," in *Proc. Recovery-CSPG CSEG CWLS Conv.*, 2011.

[4] M. Araya-Polo, J. Jennings, A. Adler, and T. Dahlke, "Deep-learning tomography," *Lead. Edge*, vol. 37, no. 1, pp. 58–66, Jan. 2018.

[5] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: http://arxiv.org/abs/1701.07875

[6] C. Burstedde and O. Ghattas 2009, "Algorithmic strategies for full waveform inversion 1D experiments," *Geophysics*, vol. 74, no. 6, pp. WCC37–WCC46, 2009.

[7] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2650–2658.

[8] D. Eigen and C. R. P. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2366–2374.

[9] A. Fichtner, *Full Seismic Waveform Modelling and Inversion*. Berlin, Germany: Springer, 2010.

[10] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," 2015, *arXiv:1508.06576*. [Online]. Available: http://arxiv.org/abs/1508.06576

[11] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[12] A. Guitton, "Blocky regularization schemes for full-waveform inversion: Blocky FWI," *Geophys. Prospecting*, vol. 60, no. 5, pp. 870–884, Sep. 2012.

[13] A. Guitton and G. E. A. Díaz, "Constrained full-waveform inversion by model reparameterization geologically constrained FWI," *Geophysics*, vol. 77, no. 2, R117–R127, 2012.

[14] Gulrajani, I., F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.

[15] K. Hammernik *et al.*, "Learning a variational network for reconstruction of accelerated MRI data," *Magn. Reson. Med.*, vol. 79, no. 6, pp. 3055–3071, Jun. 2018.

[16] T. Hastie and R. J. T. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Berlin, Germany: Springer, 2016.

[17] W. Hu, A. Abubakar, and T. M. Habashy, "Simultaneous multifrequency inversion of full-waveform seismic data," *Geophysics*, vol. 74, no. 2, pp. R1–R14, Mar. 2009.

[18] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," 2016, *arXiv:1611.07004*. [Online]. Available: http://arxiv.org/abs/1611.07004

[19] J. Johnson and A. L. Alahi Fei-Fei 2016, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 694–711.

[20] A. Karpatne *et al.*, "Theory-guided data science: A new paradigm for scientific discovery from data," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2318–2331, Oct. 2017.

[21] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1646–1654.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[23] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "DeblurGAN: Blind motion deblurring using conditional adversarial networks," 2017, *arXiv:1711.07064*. [Online]. Available: http://arxiv.org/abs/1711.07064

[24] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.

[25] L. Li, J. Pan, W.-S. Lai, C. Gao, N. Sang, and M.-H. Yang, "Learning a discriminative prior for blind image deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6616–6625.

[26] Y. Lin and L. Huang, "Acoustic- and elastic-waveform inversion using a modified total-variation regularization scheme," *Geophys. J. Int.*, vol. 200, no. 1, pp. 489–502, Jan. 2014.

[27] Y. Lin and L. Huang, "Quantifying subsurface geophysical properties changes using double-difference seismic-waveform inversion with a modified total-variation regularization scheme," *Geophys. J. Int.*, vol. 203, no. 3, pp. 2125–2149, Dec. 2015.

[28] Y. Lin and L. Huang, "Building subsurface velocity models with sharp interfaces using interface-guided seismic full-waveform inversion," *Pure Appl. Geophysics*, vol. 174, no. 11, pp. 4035–4055, Nov. 2017.

[29] Y. Lin, E. M. Syracuse, M. Maceira, H. Zhang, and C. Larmat, "Double-difference traveltime tomography with edge-preserving regularization and *a priori* interfaces," *Geophys. J. Int.*, vol. 201, no. 2, pp. 574–594, May 2015.

[30] Y. Lin, S. Wang, J. Thiagarajan, G. Guthrie, and D. Coblentz, "Efficient data-driven geologic feature characterization from pre-stack seismic measurements using randomized machine learning algorithm," *Geophys. J. Int.*, vol. 215, no. 3, 1900–1913, Dec. 2018.

[31] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos, "Using deep neural networks for inverse problems in imaging: Beyond analytical methods," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 20–36, Jan. 2018.

[32] Y. Ma, D. Hale, B. Gong, and Z. Meng, "Image-guided sparse-model full waveform inversion," *Geophysics*, vol. 77, no. 4, pp. R189–R198, Jul. 2012.

[33] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1717–1724.

[34] O. Ovcharenko, V. Kazei, D. Peter, X. Zhang, and T. Alkhalifah, "Low-frequency data extrapolation using a feed-forward ANN," in *Proc. 80th EAGE Conf. Exhibit.*, Jun. 2018, pp. 1–15.

[35] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[36] A. C. Ramírez and W. R. Lewis, "Regularization and full-waveform inversion: A two-step approach," in *Proc. SEG Tech. Program Expanded Abstr., Soc. Explor. Geophys.*, 2010, pp. 2773–2778.

[37] A. Richardson, "Seismic full-waveform inversion using deep learning tools and techniques," 2018, *arXiv:1801.07232*. [Online]. Available: https://arxiv.org/abs/1801.07232

[38] O. Ronneberger and P. T. F. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Munich, Germany: Springer, 2015, pp. 234–241.

[39] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2016, *arXiv:1605.06211*. [Online]. Available: http://arxiv.org/abs/1605.06211

[40] H. Sun and L. Demanet, "Low-frequency extrapolation with deep learning," presented at the SEG Expanded Abstract, 2018, pp. 2011–2015.

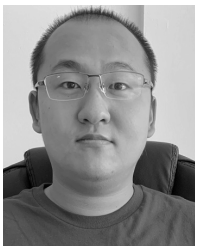
[41] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Proc. Int. Conf. Artif. Neural Netw.* Rhodes, Greece: Springer, 2018, pp. 270–279.

[42] S. Tan and L. Huang, "An efficient finite-difference method with high-order accuracy in both time and space domains for modelling scalar-wave propagation," *Geophys. J. Int.*, vol. 197, no. 2, pp. 1250–1267, May 2014.

[43] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*. Philadelphia, PA, USA: SIAM, 2005.

[44] E. Treister and E. Haber, "Full waveform inversion guided by travel time tomography," *SIAM J. Sci. Comput.*, vol. 39, no. 5, pp. S587–S609, Jan. 2017.

[45] D. Vigh and E. W. Starr, "Comparisons for waveform inversion, time domain or frequency domain," in *Proc. SEG Tech. Program Expanded Abstr., Soc. Explor. Geophys.*, 2008, pp. 1890–1894.
[46] J. Virieux and S. Operto, "An overview of full-waveform inversion in exploration geophysics," *Geophysics*, vol. 74, no. 6, pp. WCC1–WCC26, Nov. 2009.
[47] P. Wang, Z. Zhang, J. Mei, F. Lin, and R. Huang, "Full-waveform inversion for salt: A coming of age," *Lead. Edge*, vol. 38, no. 3, pp. 204–213, Mar. 2019.
[48] W. Wang and F. J. Y. Ma, "Velocity model building with a modified fully convolutional network," in *Proc. SEG Tech. Program Expanded Abstr., Soc. Explor. Geophys.*, 2018, pp. 2086–2090.
[49] Y. Wu and Y. Lin, "InversionNet: A real-time and accurate full waveform inversion with CNNs and continuous CRFs," 2018, *arXiv:1811.07875*. [Online]. Available: http://arxiv.org/abs/1811.07875
[50] F. Yang and J. Ma, "Deep-learning inversion: A next-generation seismic velocity model building method," *Geophysics*, vol. 84, no. 4, pp. R583–R599, Jul. 2019.
[51] Q. Yang *et al.*, "Low-dose CT image denoising using a generative adversarial network with wasserstein distance and perceptual loss," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1348–1357, Jun. 2018.
[52] H. Yao, F. Dai, S. Zhang, Y. Zhang, Q. Tian, and C. Xu, "DR2-Net: Deep residual reconstruction network for image compressive sensing," *Neurocomputing*, vol. 359, pp. 483–493, Sep. 2019.
[53] K. Zeng, J. Yu, R. Wang, C. Li, and D. Tao, "Coupled deep autoencoder for single image super-resolution," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 27–37, Jan. 2017.
[54] W. Zhang and Y. Shen, "Unsplit complex frequency-shifted PML implementation using auxiliary differential equations for seismic wave modeling," *Geophysics*, vol. 75, no. 4, pp. T141–T154, Jul. 2010.
[55] Z. Zhang and L. Huang, "Double-difference elastic-waveform inversion with prior information for time-lapse monitoring," *Geophysics*, vol. 78, no. 6, pp. R259–R273, Nov. 2013.
[56] Z. Zhang, Y. Zhang, Z. Zhou, and J. Luo, "Boundary-based image forgery detection by fast shallow CNN," 2018, *arXiv:1801.06732*. [Online]. Available: http://arxiv.org/abs/1801.06732
[57] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, "Image reconstruction by domain-transform manifold learning," *Nature*, vol. 555, no. 7697, pp. 487–492, Mar. 2018.
[58] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image translation using cycle-consistent adversarial networks," 2017, *arXiv:1703.10593*. [Online]. Available: http://arxiv.org/abs/1703.10593



Zhongping Zhang received the B.S. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2016, and the M.S. degree in electrical and computer engineering from the University of Rochester, Rochester, NY, USA, in 2018. He is pursuing the Ph.D. degree with the Department of Computer Science, Boston University, Boston, MA, USA.

This work was done primarily during his internship at Los Alamos National Laboratory from June 2018 to January 2019. His research interests include computer vision and its relationship with natural language processing, social media data mining, and so on.

Youzuo Lin received the B.S. degree in information and computational science from the Changchun University of Science and Technology, Changchun, China, in 2003, and the Ph.D. degree in computational and applied mathematics from Arizona State University, Tempe, AZ, USA, in 2010.

He is a Staff Scientist with Los Alamos National Laboratory, Los Alamos, NM, USA. Specifically, he has worked at applications from events detection from time series, seismic inversion/imaging, unmanned aerial vehicle (UAV) imagery analysis, hydrology inverse modeling, and ultrasound tomography. He has been a co-inventor on a couple of U.S. patents about ultrasound imaging techniques. He has published more than 60 articles in top journals and conferences, including the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS), the IEEE TRANSACTIONS ON COMPUTATIONAL IMAGING (TCI), *Numerical Linear Algebra with Applications*, *Applied Energy*, *Water Resources Research*, *Cryosphere*, *Geophysical Journal International*, *Signal Processing*, *Seismological Research Letters*, *Pure and Applied Geophysics*, the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), the IEEE International Conference on Image Processing (ICIP), the IEEE International Conference on Data Mining (ICDM), the IEEE Winter Conference on Applications of Computer Vision (WACV), the Society of Exploration Geophysicists (SEG), and so on. His research interests include optimization and numerical methods for ill-posed inverse problems, machine learning, and data-driven methods for physics problems.

Dr. Lin is an Associate Editor of *Machine Learning: Science and Technology (IOPscience)* and co-edited the 2019 Special Issue of *Interpretation: Machine Learning in Seismic Data Analysis*.