

A Physics-Driven Deep-Learning Network for Solving Nonlinear Inverse Problems¹

Yuchen Jin², Qiuyang Shen², Xuqing Wu², Jiefu Chen², and Yueqin Huang³

ABSTRACT

Geosteering inversion, which can be viewed as a nonlinear inverse problem, is an important technique used by directional drilling. Traditional methods that rely on iterative procedures and regularization are sensitive to the selection of initial values and can be slow due to convergence issues. In industrial applications, a lookup table is used to produce fast predictions. However, performance is not guaranteed by this approach due to the limitation of the hardware. In this paper, we propose a novel physics-driven deep-learning framework for

providing a fast, accurate surrogate to solve the inverse problem. Particularly, leveraged by the forward physical model and 1D convolutional neural network (1D-CNN), the proposed method provides more reliable solutions to the inverse problem with improved performance. In addition, a new physics-driven loss function is introduced to accommodate both the model misfit and the data misfit. Our experiments demonstrate the effectiveness of our method.

INTRODUCTION

Accurate interpretation of geosteering data obtained by an electromagnetic (EM) logging-while-drilling (LWD) tool in high-angle and horizontal wells is critical for formation evaluation and well placement. Among all common sensing technologies used for geosteering, which include gamma-ray, neutron density, and EM resistivity, resistivity measurement is preferred since it provides large depth of investigation (DoI). An EM resistivity tool takes the measurement of earth's electrical properties by propagating the electromagnetic wave into the formation. Equipped with tilted or transverse antennas, the tool can distinguish the direction of the boundary-crossing as well as the formation anisotropy (Edwards, 2000; Li et al., 2005; Omeragic et al., 2006). In recent years, the DoI of azimuthal resistivity tools can be extended to over 100 feet from the borehole owing to the advent of ultradeep azimuthal resistivity tools (Seydoux et al., 2014, Ezioba and Denichou, 2014; Wang et al., 2018). The new tools have similar designs to the conventional ones, which are configured with multiple transmitter-receiver pairs working at multiple frequencies. Figure 1 demonstrates a schematic diagram of an azimuthal resistivity tool. It expands the application of geosteering to the reservoir-scale.

Interpretation of the azimuthal resistivity measurements, however, is not straightforward. The induced receiver captures the EM wave and records the signal attenuation

and phase shift. To reconstruct the resistivity profile, the interpretation of multichannel measurements is indispensable. Since the EM wave can be synthesized by simulation using a physical forward model $\mathcal{F}(\cdot)$, the interpretation can be viewed as an inverse problem, $\mathcal{F}^{-1}(\cdot)$, where we need to reconstruct the formation properties from the observed measurements.

Conventional Methods

The inverse problem has been well studied by both academia and the oil and gas industry. The lookup table is one of the methods used in industrial practices. The lookup table reflects a one-to-one mapping relationship between the measurements and the resistivity profile calculated via the forward model $\mathcal{F}(\cdot)$. When new measurements are observed, an exhaustive search of the table would be performed to locate the best matching profile (Tchakarov et al., 2016; Itskovich and Nikitenko, 2017). A significant drawback of the lookup table method is its lack of scalability. The size of the lookup table grows exponentially as the dimension of parameters of the formation resistivity profile increases. Limited by the storage space, the lookup table method is unable to provide accurate inversion results for complicated formation structures.

Iterative methods have also been used to solve the inverse problem. The so-called deterministic inversion iteratively solves an optimization problem by minimizing

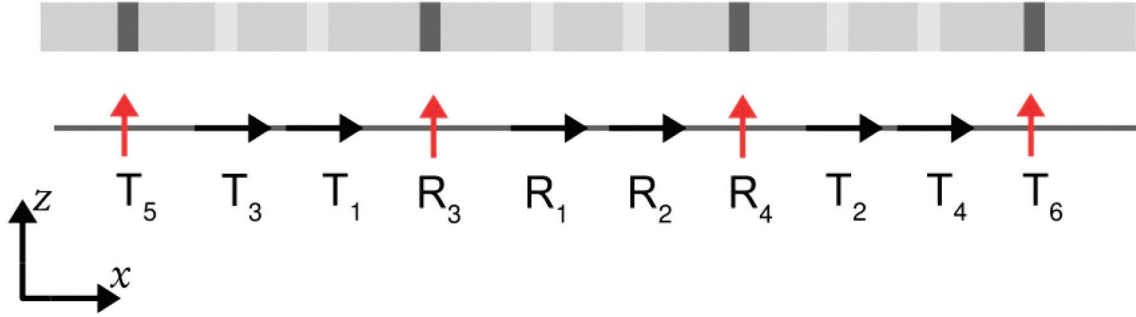


Fig. 1— Schematic of an azimuthal resistivity tool. T_1 , T_2 , T_3 and T_4 are z-direction transmitting antennas, T_5 and T_6 are x-direction transmitting antennas. R_1 and R_2 are z-direction receiving antennas, and R_3 and R_4 are x-direction receiving antennas

the difference between the observed measurements and synthesized responses obtained via a forward model $\mathcal{F}(\cdot)$ built upon rigorous physics. A variety of iterative algorithms is available, including the gradient descent method, Gauss-Newton method, and the Levenberg-Marquardt algorithm (LMA) (Levenberg, 1944; Marquardt, 1963). The deterministic methods are usually gradient-based and searching for the best earth model is efficient. However, due to the high nonlinearity of the forward model, deterministic methods suffer from the local minima problem and are very sensitive to the initial values. In addition, additional regularizing terms are required since inverse problems are typically ill-posed (Key, 2009; Thiel et al., 2018).

Stochastic inversion is another popular approach for solving inverse problems. The stochastic global optimization is governed by the Bayesian theorem. Under the Bayesian framework, the observation model or the so-called likelihood is usually built upon the forward model and some knowledge about the errors (e.g. measurement noise). The Markov chain Monte Carlo (MCMC) sampling method is used to sample the posterior distribution of model parameters given the measurements. Stochastic inversion is computationally expensive. Although some efforts have been made to improve the sampling efficiency through the multichain sampling (Shen et al., 2017), hybrid MCMC (Shen et al., 2018b), and nonparametric sampling (Shen et al., 2018a), the overall efficiency is much lower than deterministic methods.

Deep-Learning Approaches

In recent years, deep-learning approaches have been used to solve inverse problems. The basic idea is to train a deep neural network (DNN) to map measurements into the model parameters. In this case, the network serves as a surrogate for the inverse mapping $\mathcal{F}^{-1}(\cdot)$. Examples of applications include geoacoustic model inversion (Benson et al., 1998), EM nondestructive evaluation (NDE) signal

inversion (Ramuhalli et al., 2002), eddy current inversion (Behun et al., 2018), radar backscattering model inversion (Farah et al., 2006), spectral optical model inversion (Malkiel et al., 2018) and geosteering earth model inversion (Xu et al., 2018). Instead of finding a numerical solution through the physical modeling, these implementations try to learn an inverse mapping driven by the historical data. The feasibility of learning a physical model via the data-driven approach has been supported by Lerer et al. (2016), who propose a segmentation network to predict the movement of block towers. In Fragkiadaki et al. (2016), the visualization of a billboard game and the agent's applied forces are input to a recurrent neural network for predicting the movement of the objects.

Different from the aforementioned approaches, the inverse mapping can be learned by combining the deep-learning framework with conventional iterative algorithms. Adler and Oktem (2017) proposed that gradients produced by the gradient descent algorithm could be adjusted by a well-trained artificial neural network. Huang et al. (2018) proposed a similar workflow for full waveform inversion (FWI). In this work, the DNN predicts the update step for the seismic velocity model from the gradient calculated from the forward modeling. This work aims at improving the updating direction and avoiding the local minima during the inversion. Multiple works also explored the idea of applying physical constraints to the DNN by using analytical and differential models. For example, the inverse discrete cosine transform is used to construct the loss function by Johnston et al. (2017). Jaderberg et al. (2015) formulated the affine transform as an analytical function and constructed a spatial transformer network. Whitening and coloring feature transforms are used to transfer the style features extracted by the auto-encoder in Li et al. (2017). Wu et al. (2017) proposed a novel data misfit function for training an image segmentation network by including a differential physical model. Zhang et al. (2019) built a physics-constrained CNN for predicting seismic responses. In this work, the network is

guided by a data loss and a physical loss simultaneously. The physical loss is formulated by the physical constraints of a dynamic system. The introduction of the physical constraint mitigates the overfitting problem and reduces the dependency on a large training set. Jiang and Fan (2019) constructed a generative network for configuring metasurfaces, which are subwavelength-structured artificial media that can shape and localize EM waves in unique ways. The network is trained to learn the relationship between device geometry and optical response. The training process is driven by the data misfit, which measures the difference between the ground-truth and the synthetic output obtained through the forward simulation. The back-propagation of the forward simulated is estimated by the adjoint state method.

Similarly, we have included the data misfit as a part of the loss function. Our forward model is differentiable but requires numerical estimation of the Jacobian matrix. The technique developed in this work can be applied to an arbitrary differentiable forward model.

METHODOLOGY

Geosteering modeling and inversion requires a time-sensitive workflow to satisfy the real-time data interpretation and decision-making. Since 3D modeling is computationally expensive, the formation structure is parameterized as a 1D

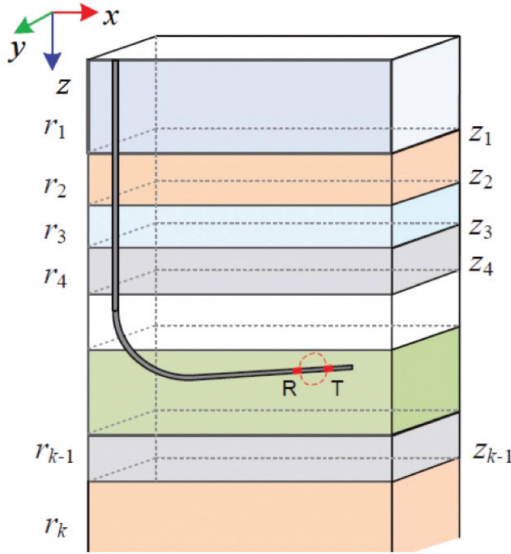


Fig. 2—An illustration of the underground formation. Divided by several homogeneous layers, the underground formation could be described by resistivities of each layer and boundary depth between each two layers.

model at every single logging station. Figure 2 presents a schematic diagram of a 1D layered earth model, which assumes that the formation property remains constant within each horizontal layer and all layers are stacked vertically.

The 1D layered model is defined by the electrical resistivity, r_i , of each layer, and the true vertical depth (TVD), z_i , of the boundary between adjacent $(i - 1)^{\text{th}}$ and i^{th} layers. We hereby parameterize the earth model of k layers by a vector as follows:

$$\mathbf{m} = \{r_1, r_2, \dots, r_k, z_1, z_2, \dots, z_{k-1}\}. \quad (1)$$

Given a forward model, the synthetic measurements \mathbf{d} can be obtained by:

$$\mathbf{d} = \mathcal{F}(\mathbf{m}). \quad (2)$$

Where $\mathcal{F}(\cdot)$ indicates the forward model. The inverse modeling can be written as:

$$\hat{\mathbf{m}} = \mathcal{F}^{-1}(\hat{\mathbf{d}}). \quad (3)$$

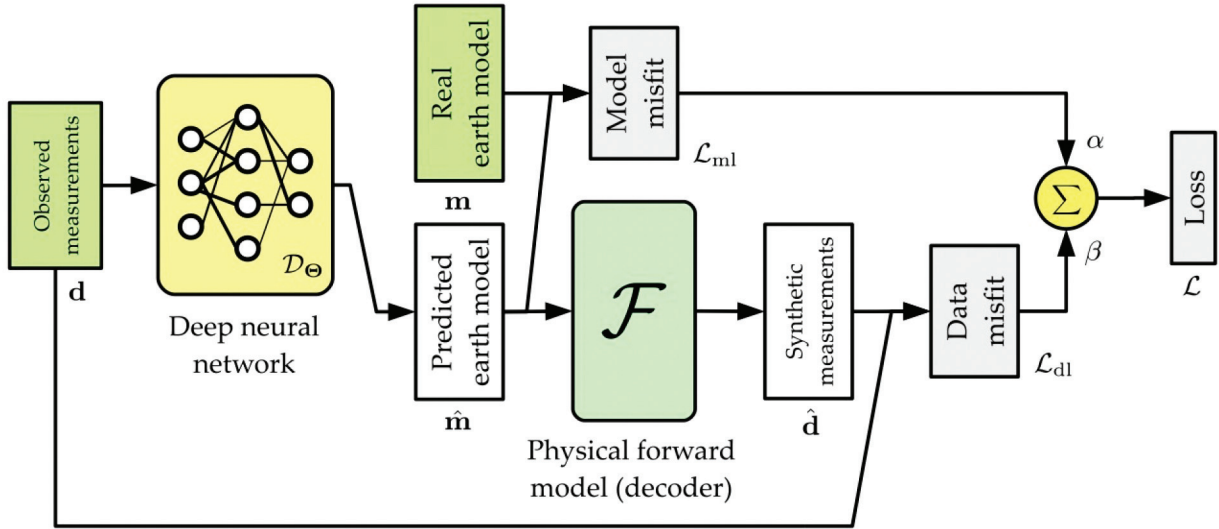
Where $\mathcal{F}^{-1}(\cdot)$ defines the inverse mapping and $\hat{\mathbf{d}}$ is the observed measurements.

In this article, the azimuthal resistivity LWD tool we synthesize adopts the design shown in Fig. 1. The receivers capture the induced EM field and record the voltage change. A full mutual inductance tensor with nine components can be calculated. Hence, different types of measurements can be further derived through the captured response. We simulate the EM response in 1D layered space by using multiple pairs of triaxial antennas. In detail, the channels are divided by different frequencies and T-R spacings. The lowest frequency is 1 kHz with the longest spacing of 160 ft. Given a 1D formation model, in our synthetic case, 92 measurements are generated.

The relationship among the measurement, the source, and the medium is essentially governed by physics. An analytical inversion process relies on the forward model to guide its search for solutions. However, the solution could be suboptimal due to local minima problems, noise, and other unknown interference. In this paper, we propose a novel physics-driven DNN (PhyDNN) framework for solving the geosteering inverse problem. The notations used in the following part are outlined in Table 1. Figure 3 demonstrates the framework diagram of the PhyDNN.

Table 1—List of Notations for the Proposed Methods

Symbol	Representation
L	Total number of layers of the 1D earth model
$\mathbf{m} \in \mathbb{R}^{2L-1}$	A parameterized earth model m
$\mathbf{d} \in \mathbb{R}^n$	A measurement.
\mathcal{D}_{Θ}	A DNN with tunable parameter Θ
\mathcal{A}_j	The j^{th} layer of the DNN
\mathcal{F}	A physical forward model
$\hat{\mathbf{m}} \in \mathbb{R}^{2L-1}$	A predicted earth model \hat{m}
$\hat{\mathbf{d}} \in \mathbb{R}^n$	A synthetic measurement produced by the forward model
$\mathbf{J} \in \mathbb{R}^{n \times (2L-1)}$	The Jacobian matrix $\frac{\partial \mathcal{F}}{\partial \hat{\mathbf{m}}}$
$\mathcal{A} \circ \mathcal{B}$	A composition of two functions. It is equivalent to $\mathcal{A}(\mathcal{B}(\cdot))$
$\mathbf{w} \otimes \mathbf{x}$	Convolve \mathbf{x} by with kernel \mathbf{w}


Fig. 3—An overview of the framework for the proposed PhyDNN.

Modern Deep Neural Network (DNN) Structure

In deep-learning approaches, a DNN serves as a surrogate for the whole iterative optimization. Generally, a DNN \mathcal{D} could be viewed as a stack of non-linear feed-forward operators $\{\mathcal{A}_n\}$, i.e. “layers”. A DNN with n layers could be formulated as:

$$\mathcal{D}_{\Theta} = \phi \circ \mathcal{A}_n \circ \mathcal{A}_{n-1} \circ \dots \circ \mathcal{A}_2 \circ \mathcal{A}_1, \quad (4)$$

In our work, the DNN is adapted from the VGG16 network (Simonyan and Zisserman, 2015). The detailed network layout is shown in Fig. 4. Our network consists of

nine 1D convolutional layers, five average pooling layers, and one fully-connected layer. The observed measurements are normalized before fed into the convolutional layers. The output of the fully connected layer is sliced into two parts and scaled into the resistivity domain and the TVD boundary domain respectively. The whole network contains 1.83×10^6 trainable parameters. Each network layer is a composition of convolution, normalization, and activation and can be formulated as:

$$\mathcal{A}_j(\mathbf{x}) = \text{PReLU} \left(\gamma_j \frac{\mathbf{w}_j \otimes \mathbf{x} - \mu}{\sigma} + \beta_j, \alpha_j \right), \quad (5)$$

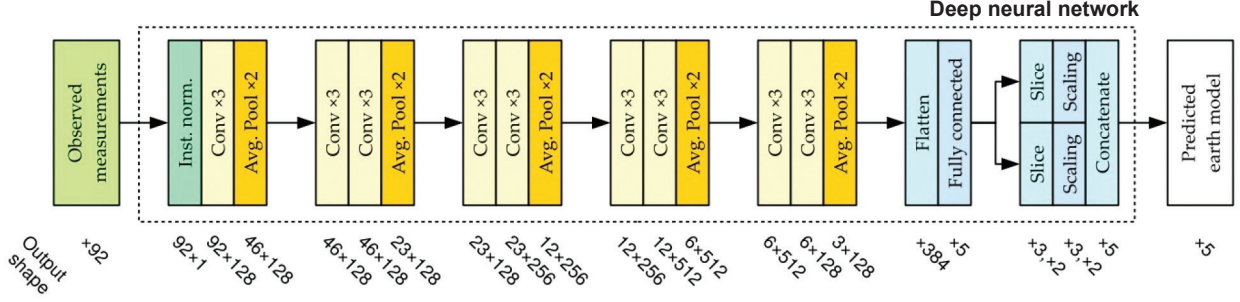


Fig. 4—Illustration of the architecture details of the DNN adapted from the VGG16.

where \mathbf{w}_j is a learnable 1D convolutional kernel. μ , σ are mean and standard deviation estimated from $\mathbf{w}_j \otimes \mathbf{x}$. In the following, all learnable network parameters are represented by $\Theta = \{\mathbf{w}_j, \gamma_j, \beta_j, \alpha_j, \dots\}$. Unique features included in this network design include:

- 1D convolutional layers for 1D inputs (Van Den Oord et al., 2016; Ince et al., 2016).
- Instance normalization (Ulyanov et al., 2016; He et al. 2016; Zhang et al., 2018).
- PReLU (He et al., 2015) as the activation function.

The last layer ϕ of this network is a scaling layer, which is defined as

$$\phi(\mathbf{h}_n, m_{\min}, m_{\max}) = (m_{\max} - m_{\min}) \text{sigmoid}(\gamma \mathbf{h}_n + \beta) + m_{\min}, \quad (6)$$

where \mathbf{h}_n is the output of the hidden layer \mathcal{A}_n . m_{\max} and m_{\min} are the upper bound and lower bound of the prediction respectively. Both γ and β are learnable vectors. The scaling layer serves as a constraint to enforce the prediction's value range, i.e. $\hat{\mathbf{m}} = \phi(\mathbf{h}_n) \in (m_{\min}, m_{\max})$.

The DNN is designed to learn the mapping function between the observed measurements \mathbf{d} (input of the network) and predicted earth model $\hat{\mathbf{m}}$ (output of the network).

Data-Driven Method and Model Misfit

Most deep-learning methods for solving an inverse problem are data driven. In those methods, the deep network learns the inverse mapping $\mathcal{F}^{-1} : \mathbf{d} \mapsto \mathbf{m}$ from the statistical distribution of the dataset, which is generated by following physical constraints. During the training process, the network learns these constraints from the dataset empirically. The objective function for training used by a data-driven network is as follows:

$$\arg \min_{\Theta} \mathbb{E}_{(\mathbf{m}, \mathbf{d}) \in (\mathbb{M}, \mathbb{D})} [\mathcal{L}_{\text{ml}}(\mathbf{m}, \mathbf{d}, \Theta)], \quad (7)$$

$$\mathcal{L}_{\text{ml}}(\mathbf{m}, \mathbf{d}, \Theta) = \|\mathcal{D}_{\Theta}(\mathbf{d}) - \mathbf{m}\|_2^2,$$

where (\mathbf{m}, \mathbf{d}) is a sample that is randomly selected from the training set. The “model misfit” \mathcal{L}_{ml} in Eq. 7 measures the difference between the predicted earth model $\hat{\mathbf{m}}$ and the true earth model \mathbf{m} . Once the model is well trained, we can use the feed-forward network prediction $\hat{\mathbf{m}} = \mathcal{D}_{\Theta}(\mathbf{d})$ to estimate the unknown earth model for a new group of measurements \mathbf{d} .

In a data-driven DNN, the convergence of the network is guided by the distribution of the parameter of the earth model alone. Due to the randomness in selecting the data batch and the nonlinearity of the governing physics, the descending path will be long and fluctuate. Therefore, a data-driven DNN could either oversmooth or overfit the inverse mapping function between the measurement and the earth model. A differentiable forward model, on the other hand, can be used to regulate the back-propagation process.

Physics-Driven Method and Data Misfit

The idea of the PhyDNN is inspired by the auto-encoder structure, which has been applied for unsupervised learning (Baldi, 2012). Generally, the auto-encoder comprises two parts, an encoder and a decoder. The encoder maps the input into the encoded space and the decoder would recover the input data of the encoder. In most cases, the encoder and the decoder are designed in a symmetric structure. In Fig. 3, the 1D convolutional DNN serves as an encoder, which can be viewed as a surrogate for the inverse mapping \mathcal{F}^{-1} . The decoder is replaced by the predefined physical forward-modeling function $\mathcal{F} : \mathbf{m} \mapsto \mathbf{d}$. The objective function of the PhyDNN training is defined as:

$$\arg \min_{\Theta} \mathbb{E}_{(\mathbf{m}, \mathbf{d}) \in (\mathbb{M}, \mathbb{D})} [\alpha \mathcal{L}_{\text{ml}}(\mathbf{m}, \mathbf{d}, \Theta) + \beta \mathcal{L}_{\text{dl}}(\mathbf{d}, \Theta)],$$

$$\mathcal{L}_{\text{dl}}(\mathbf{d}, \Theta) = \|\mathcal{F} \circ \mathcal{D}_{\Theta}(\mathbf{d}) - \mathbf{d}\|_2^2, \quad (8)$$

According to Eq. 8, the loss function consists of two terms, model misfit \mathcal{L}_{ml} and data misfit \mathcal{L}_{dl} . Similar to the traditional iterative optimization, the data misfit, \mathcal{L}_{dl} , quantifies the difference between observed measurements

and synthetic tool responses obtained via a forward function for a given earth model. Unlike a data-driven DNN, whose convergence is guided by the model misfit alone, a differentiable forward model is used by PhyDNN to regulate the back-propagation process by minimizing the data misfit. A weighting factor, $\{\alpha, \beta\}$, is added to balance the two misfit terms. Compared to the data-driven DNN and iterative optimization, PhyDNN will deliver an accurate inversion result with high measurement compliance and more success in avoiding local minima problems.

The implementation for PhyDNN requires backpropagating the gradients from the data misfit. Similar to deterministic methods, we could formulate the first-order gradient of data misfit by using the chain rule:

$$\begin{aligned} \nabla_{\text{dl}} \Theta &= \frac{\partial \mathcal{L}_{\text{dl}}}{\partial \Theta} = \frac{\partial \mathcal{L}_{\text{dl}}}{\partial \hat{\mathbf{d}}} \frac{\partial \hat{\mathbf{d}}}{\partial \Theta}, \\ \frac{\partial \mathcal{L}_{\text{dl}}}{\partial \hat{\mathbf{d}}} &= -2(\mathbf{d} - \mathcal{F} \circ \mathcal{D}_{\Theta}(\mathbf{d})), \\ \frac{\partial \hat{\mathbf{d}}}{\partial \mathbf{m}} &:= \frac{\partial \mathcal{F}}{\partial \mathbf{m}} = \mathbf{J}, \\ \frac{\partial \mathbf{m}}{\partial \Theta} &:= \frac{\partial \mathcal{D}_{\Theta}}{\partial \Theta}. \end{aligned} \quad (9)$$

For forward models without analytical form of derivatives, the Jacobian matrix in Eq. 9 is estimated numerically as:

$$\mathbf{J}[:, i] = \frac{\mathcal{F}(m_i + \Delta m, \mathbf{m} \setminus m_i) - \mathcal{F}(m_i, \mathbf{m} \setminus m_i)}{\Delta m}. \quad (10)$$

EXPERIMENTS

In this part, we train and test the network to demonstrate its effectiveness in solving the geosteering inverse problem. The experiment is deployed on Python 3.5+ and TensorFlow r1.13+. The forward-model function is implemented in C++ with a Python-C-API wrapper. We use OpenMP to enable the parallel computing for the forward model to accelerate the CPU computation. During the training and testing phases, the whole program, except the implementation of the forward-model function, can be run on both CPU and GPU.

To accelerate the implementation cycle and reduce coding efforts, we provide an open source library: Modern Deep Network Toolkits (MDNT) (Jin, 2018). MDNT can wrap an arbitrary external forward-model function as a TensorFlow operator. The modern convolutional layers and the scaling layer are also implemented in MDNT.

Earth Model Description and Dataset

In the experiment, we assume that the underground formation can be represented by a three-layer model of five parameters, three resistivity values (R_1, R_2, R_3) and the distance to the upper and lower boundaries (D_{up} and D_{dn}). To simplify the problem, the relative dip angle of the logging tool is assumed to be fixed. Figure 5 shows the schematic graph of the three-layer model.

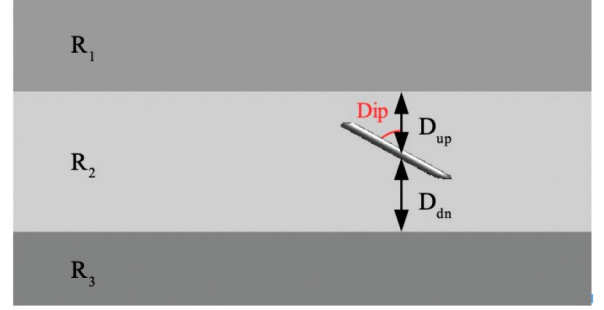


Fig. 5—A schematic diagram for a 3-layer earth model.

The training data set is sampled from a lookup table generated by using the forward model function $\mathcal{F}(\cdot)$. Each entry of the lookup table (\mathbf{m}, \mathbf{d}) is called a “point”, where $\mathbf{m} = \{R_1, R_2, R_3, D_{\text{up}}, D_{\text{dn}}\}$ is the model parameter and \mathbf{d} the corresponding tool responses (measurements). The range of the model parameters is defined as: $R_1, R_2, R_3 \in (10^{-1}, 10^2)$ [$\Omega \cdot \text{m}$], $D_{\text{up}} \in (-25.0, 0)$ [ft] and $D_{\text{dn}} \in (0, 25.0)$ [ft]. We fixed the dip angle at 90° in all cases. Such a configuration ensures that (1) $D_{\text{up}} < D_{\text{dn}}$, (2) the logging tool is kept horizontally in the middle layer, and (3) two boundaries are in the sensing range of the logging tool. To sample the resistivity and boundary values, we divide the logarithmic resistivity range into 16 intervals and the depth range into 24 intervals. Therefore, we have $16^3 \times 24^2$, i.e., 2.36×10^6 , points in the lookup table. For each measurement, the forward model outputs 92 values. The total size of the lookup table for training is approximately 1.61 GB.

To minimize the training workload and demonstrate the effectiveness of the model, we selected a three-layer model and keep the tool horizontal in the middle layer. During the experiment, 90% of the dataset is used for training and the rest 10% is reserved for validation.

Training

The training is performed on a NVIDIA DGX workstation, equipped with Intel Xeon E5-2698 v4 CPU (20-core) and 4 Tesla V100 GPUs. Only one GPU is used

for training the model. The GPU has 5,120 CUDA cores, 640 tensor cores, 32 GB GPU RAM and 900 GB/s RAM bandwidth.

The network is trained for a total of 5 epochs (Goodfellow et al., 2016). At each step, we randomly select 64 pairs of (\mathbf{m}, \mathbf{d}) from the training set as a minibatch during the batchwise optimization. We train the network with ADAM optimizer (Kingma and Ba, 2015). During the first 2 epochs, the DNN is trained exclusively by minimizing \mathcal{L}_{ml} alone.

The DNN is then further optimized by minimizing both \mathcal{L}_{ml} and \mathcal{L}_{dl} in the next 3 epochs. As a comparison, we also trained a data-driven DNN by minimizing the model misfit loss \mathcal{L}_{ml} for 5 epochs. The training loss is measured by the mean squared error (MSE).

Figure 6a shows that the training losses with regard to the model misfit are statistically the same for both data-driven DNN and PhyDNN. However, compared to the data-driven DNN, PhyDNN has a much smaller training error with regard to the data misfit. Figure 6 also compares training loss with validation errors, which indicates that our network training has converged.

Testing

Once the network is well trained, only the encoder part

is needed for inversion. The end-to-end pretrained feed-forward network takes measurements as the input and outputs the predicted model parameters as the inversion result. In order to validate the efficiency of the proposed approach, we compare the inversion performance by using PhyDNN, data-driven DNN, and the lookup-table methods. We use the same lookup table as the training and no interpolation is performed for the table search.

Figure 7 shows the inversion result by applying different methods to a case where the logging tool locates in a layer with high resistivity. The green line represents the path of the LWD tool. Figure 8 compares the measurement discrepancy between observed and synthetic values for different inversion approaches. Figure 9 shows two more examples with different resistivity distributions. In all cases, PhyDNN delivers the best inversion performance with regard to both model accuracy and measurement compliance. Figure 10 compares model misfit and data misfit by averaging testing results for all three approaches and the result shows that (1) the lookup-table method has the highest model and data misfits on average, (2) the prediction from the data-driven network returns small model misfit but high data misfit, and (3) PhyDNN has the lowest overall misfit.

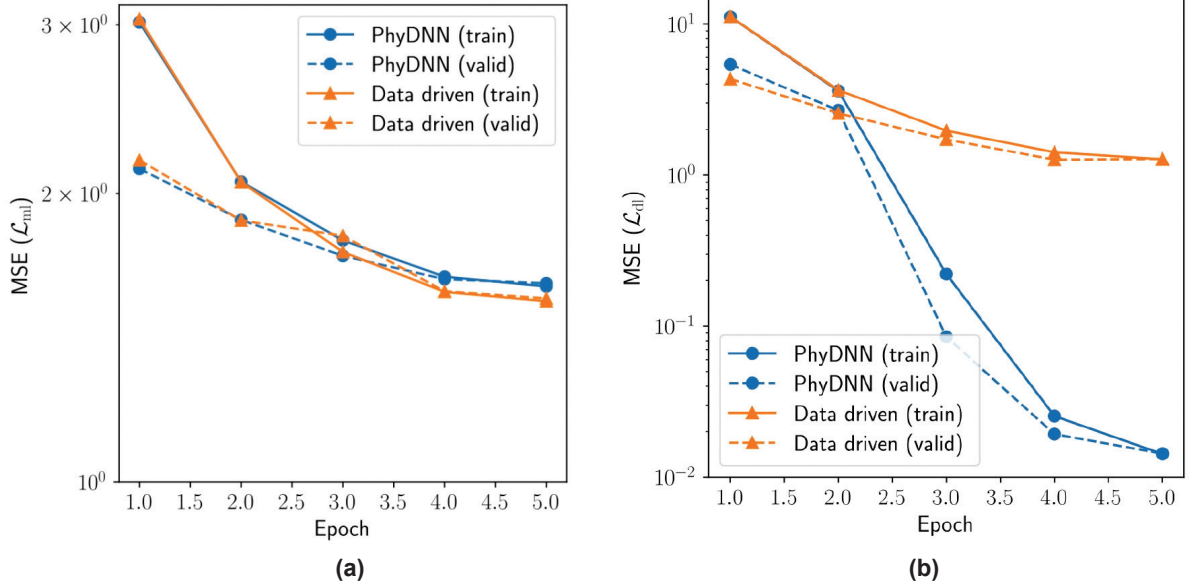


Fig. 6—Training and validation curves for both PhyDNN and data-driven DNN. (a) The curves denoting model misfit (\mathcal{L}_{ml}) for both data-driven DNN and PhyDNN. Both approaches have similar training and validation errors with regard to the model misfit. (b) The curves denoting data misfit (\mathcal{L}_{dl}) for both data-driven DNN, and PhyDNN. Compared to data-driven DNN, PhyDNN has much lower training and validation errors with regard to the data misfit.

A Physics-Driven Deep-Learning Network for Solving Nonlinear Inverse Problems

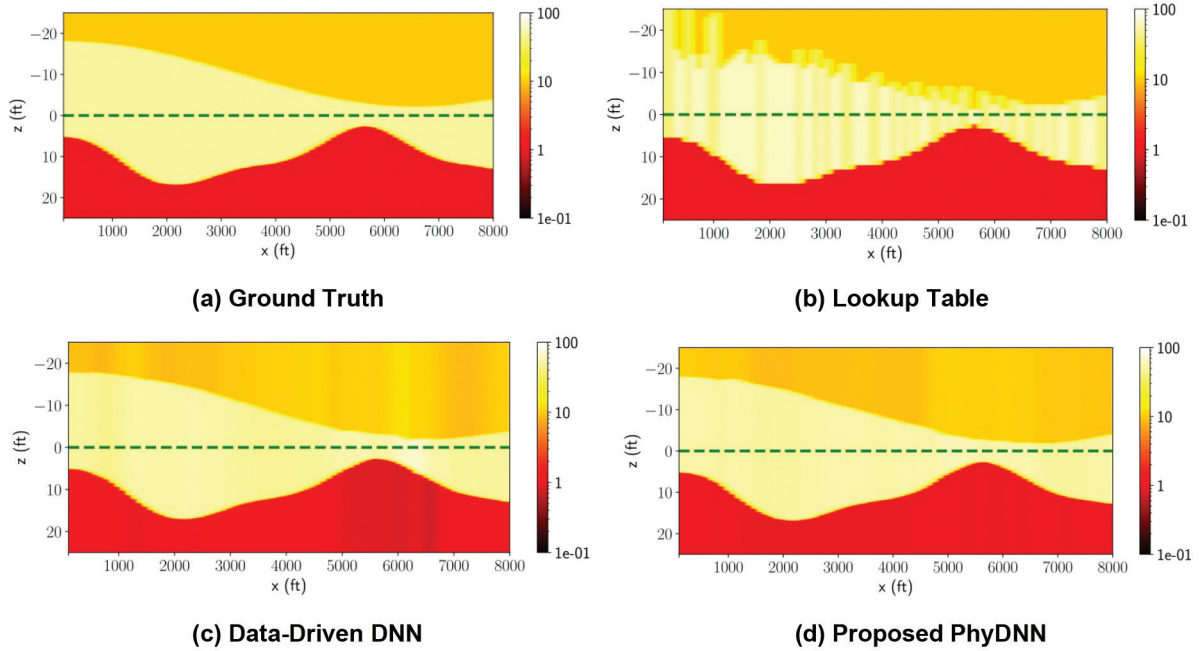


Fig. 7—A three-layer case by using different inversion methods. The x -axis represents the distance along the horizontal direction. z -axis is along the vertical direction. The green line represents the trajectory of the wellbore, and different resistivities are represented by different colors.

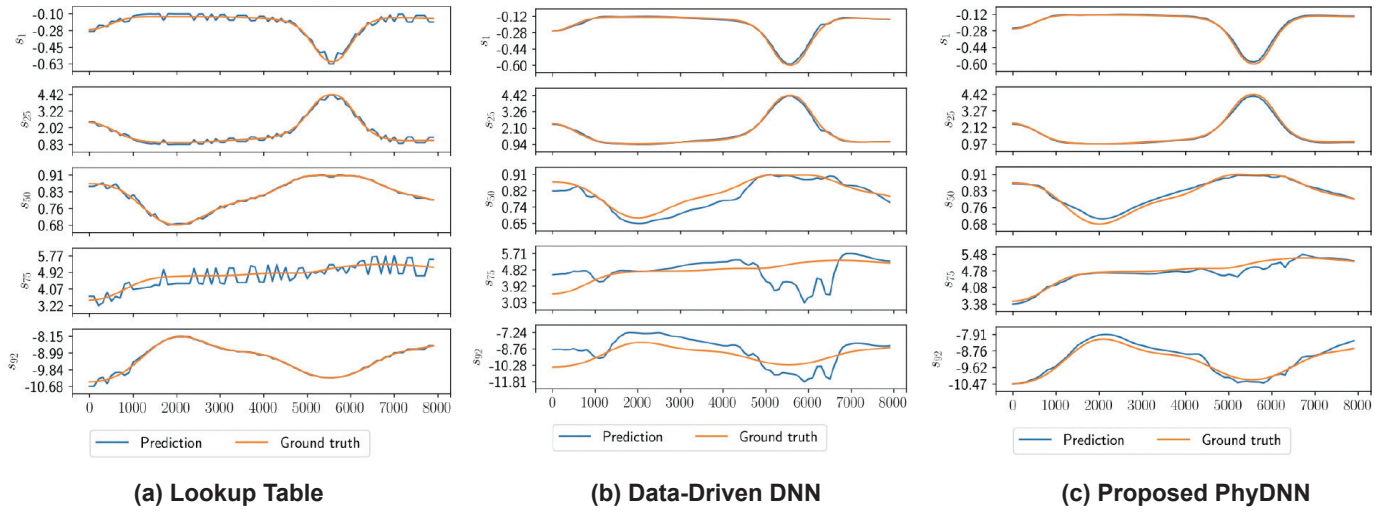


Fig. 8—Comparison between predicted measurements and ground-truth measurements with different methods. In each subfigure, the axis represents the distance along the horizontal direction and we pick five representative measurements.

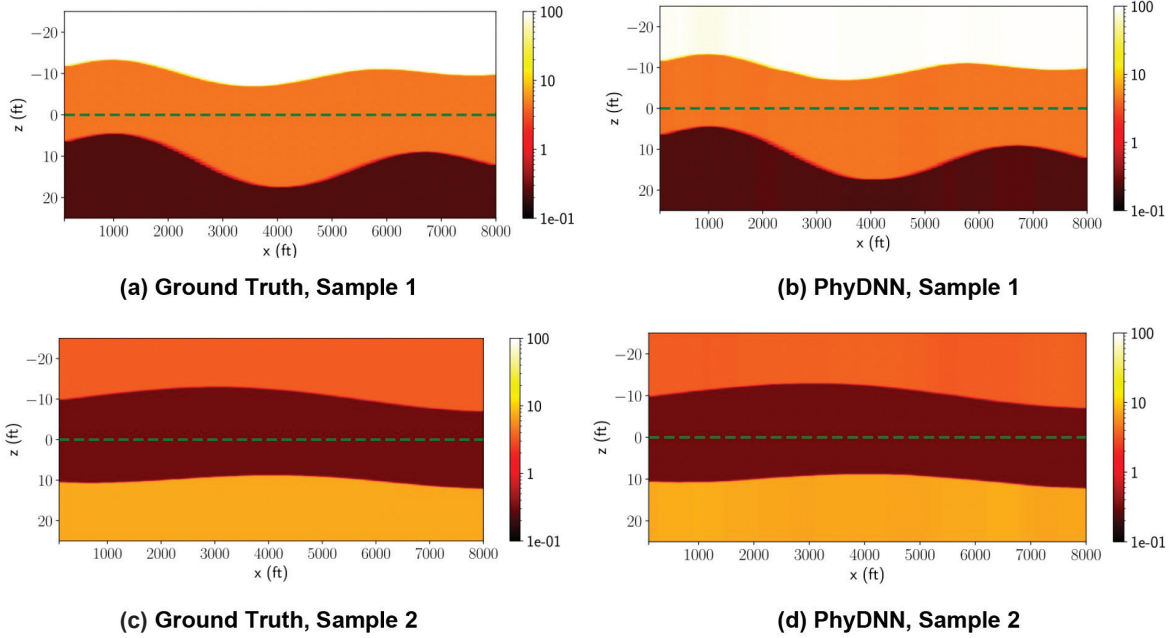


Fig. 9—Two inversion examples using PhyDNN. Numerical tests show that PhyDNN works well for a three-layer model with different resistivity distributions.

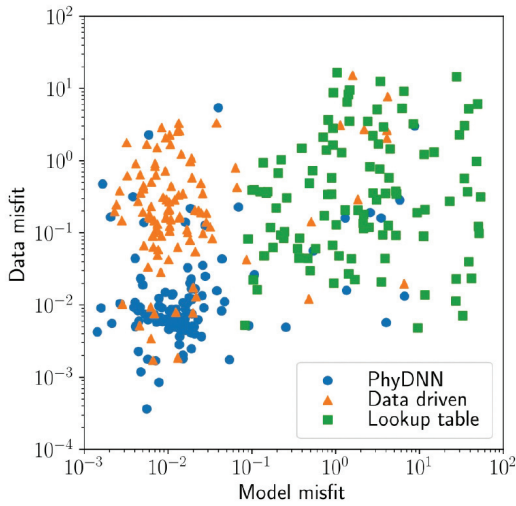


Fig. 10—A comparison of data misfits and model misfits for all testing cases by using different inversion methods.

DISCUSSION

Evaluation of the Computational Efficiency

In this section, we evaluate the computational efficiency of three different inversion methods: lookup table, PhyDNN, and LMA, an iterative deterministic algorithm. Both the time and memory consumption are taken into consideration. The time consumption is the elapsed time consumed by the evaluated method. The memory consumption reflects the required memory for running the method. Table 2 and Table 3 list the time consumption and the memory consumption respectively.

Table 4 shows the average inversion accuracy for the different methods. The overall inversion accuracy of PhyDNN is comparable to LMA. In practice, the PhyDNN result can be used as the initial guess for the deterministic method, which will help the system converge much faster.

Table 2— Time Consumption for Interpreting 80 Logging Points

PhyDNN (sec)	Lookup Table (10K) (sec)	Lookup Table (2M) (sec)	LMA Optimization (sec)
0.011	2.939	62.315	169.754

10K and 2M indicate different lookup-table sizes.

Table 3— Memory Consumption for Interpreting 80 Logging Points

PhyDNN	Lookup Table (10K)	Lookup Table (2M)	LMA Optimization
7 MB	70 MB	1.61 GB	<1K

10K and 2M indicate different lookup-table sizes.

Table 4—Comparison of Average Data and Model Misfits for Using Different Inversion Methods

	PhyDNN	Lookup Table (10K)	Lookup Table (2M)	LMA Optimization
\mathcal{L}_{ml}	0.0123	1.9686	1.4300	0.0517
\mathcal{L}_{dl}	0.0006	0.0439	0.0269	0.0013

Sensitivity Analysis

In order to improve and validate the robustness of the model, we added noise to the input of the network during the training. As Goodfellow et al. (2016) and Loshchilov and Hutter (2019) pointed out, noise injection is an effective data augmentation method and serves as a replacement of the traditional Tikhonov regularization (Bishop, 1995).

In this test, we retrained both PhyDNN and data-driven DNN with the same training set but replaced the network input with $\mathbf{d} + \varepsilon$, where \mathbf{d} is the observed measurements and ε is white noise. ε follows a normal distribution $\mathcal{N}(0, \sigma)$ where the standard deviation (std.) σ is also a random variable and sampled from a uniform distribution $U(0, 0.4)$. The “noise-robust” version of PhyDNN and data-driven DNN is denoted as NR-PhyDNN and NR-data-driven DNN. Figure 11 concludes the test result with noisy measurements. NR-PhyDNN achieved low model misfit and data misfit values on all noise levels. NR-data-driven DNN, however, performed worse than vanilla [Note to author: What does word ‘vanilla’ mean here? ‘basic’?] PhyDNN when the noise level is low.

In short, the above experiments show that PhyDNN is

more robust after augmenting the training processing with the regularization method. NR-PhyDNN is preferred for interpreting noisy field data.

CONCLUSIONS

In this work, we proposed a novel physics-driven DNN to solve the geosteering inverse problem. The proposed PhyDNN is implemented by combining the DNN with a forward-model function that follows the physics. Compared to other methods, we conclude that:

- PhyDNN is an efficient method with high inversion accuracy and low training losses.
- PhyDNN is computationally cheap compared to iterative optimization or lookup-table solutions.
- PhyDNN is robust against noisy measurement after training with data augmentation.
- The deep structure of PhyDNN prevents the underfitting problem encountered by a shallow structure and provides a scalable solution for more challenging inverse problems with complicated formation structures.

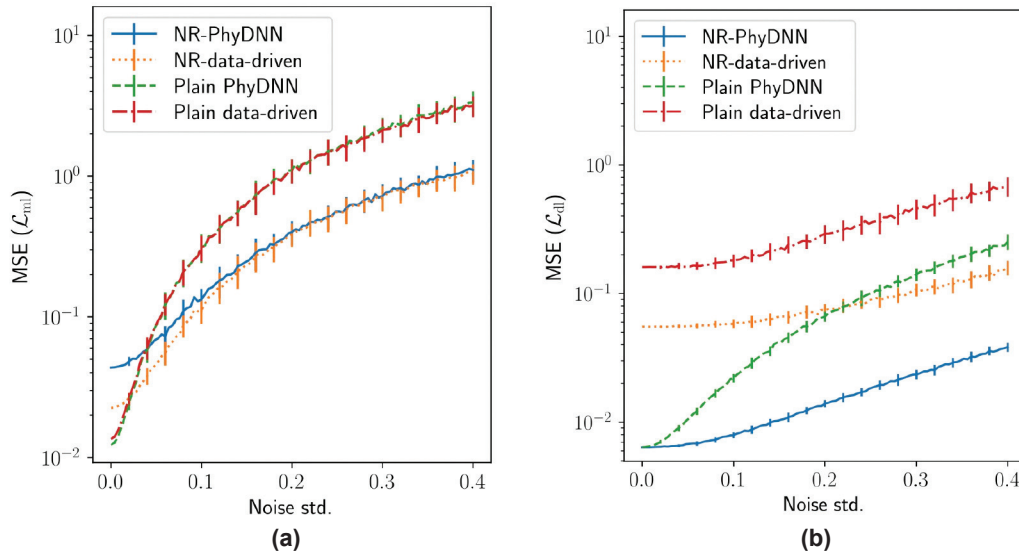


Fig. 11—Comparison of data misfits and model misfits between networks trained with and without noise augmentation. (a) Model misfit vs. noise level (x-axis). As the noise level increases, NR-PhyDNN and NR-data-driven DNN has much lower model misfits. (b) Data misfit vs. noise level (x-axis). As the noise level increases, both NR-PhyDNN and NR-data-driven DNN deliver lower data misfits than corresponding networks trained without noise augmentation. A plain PhyDNN shows good resistance to the noise when the noise level is low.

- PhyDNN has the potential to enable downhole computing for an LWD tool.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Energy, Office of Science, and Office of Advanced Science Computing Research, under Award Numbers DE-SC0017033.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

NOMENCLATURE

Abbreviations

CNN = convolutional neural network
 DNN = deep neural network
 DoI = depth of investigation
 EM = electromagnetic
 FWI = full waveform inversion
 LMA = Levenberg-Marquardt algorithm
 LWD = logging while drilling
 MCMC = Markov chain Monte Carlo
 MDNT = modern deep network toolkits
 MSE = mean squared error
 NDE = nondestructive evaluation
 PhyDNN = physics-driven DNN
 TVD = true vertical depth

Symbols

A_j = the j^{th} layer of the DNN
 \mathbf{d} = a measurement
 $\hat{\mathbf{d}}$ = a synthetic measurement produced by the forward model
 \mathcal{D}_Θ = a DNN with tunable parameter Θ
 $\mathbb{E}[\cdot]$ = an operator for expectation

$\mathcal{F}(\cdot)$ = a physical forward model
 \mathbf{J} = the Jacobian matrix
 $\mathcal{L}_{\text{ml}}, \mathcal{L}_{\text{dl}}$ = the model misfit and the data misfit
 \mathbf{m} = a parameterized earth model
 $\hat{\mathbf{m}}$ = a predicted earth model
 \mathbf{w}_j = the kernel parameter of the j^{th} layer
 ϕ_j = a scaling layer
 μ_j, σ_j = the average and the standard error of the j^{th} layer
 $\{\alpha, \beta\}$ = a weighting factor for misfits
 \otimes = an operator for convolution
 \circ = an operator for function composition
 ∇_{dl} = an operator for the data-based gradient
 γ_j, β_j = the scaling factors for the j^{th} layer

REFERENCES

- Adler, J., and Öktem, O., 2017, Solving Ill-Posed Inverse Problems Using Iterative Deep Neural Networks, *Inverse Problems*, **33**(12), 124007. DOI: 10.1088/1361-6420/aa9581.
- Baldi, P., 2012, Autoencoders, Unsupervised Learning, and Deep Architectures, *Journal of Machine Learning Research, Workshop and Conference Proceedings*, **27**, 37–49. <http://proceedings.mlr.press/v27/baldi12a/baldi12a.pdf>. Accessed January 3, 2020.
- Behun, L., Smetana, M., and Capova, K., 2018, Comparison of Detection Abilities Between Fluxgate and GMR Magnetometer in Inverse ECT of Deep Lying Cracks, *2018 ELEKTRO*, 1–4, DOI: 10.1109/ELEKTRO.2018.8398332.
- Benson, J., Chapman, N.R., and Antoniou, A., 1998, Geoacoustic Model Inversion With Artificial Neural Networks, *Proceedings, 1998 IEEE Symposium on Advances in Digital Filtering and Signal Processing*, , 121–125, DOI: 10.1109/ADFSP.1998.685708.
- Bishop, C.M., 1995, Training With Noise is Equivalent to Tikhonov Regularization, *Neural Computation*, **7**(1), 108–116. DOI: 10.1162/neco.1995.7.1.108.
- Edwards, J., 2000, Geosteering Examples Using Modeled 2-MHz LWD Response in the Presence of Anisotropy, Paper N, *Transactions, SPWLA 41st Annual Logging Symposium*, Dallas, Texas, USA, 4–7 June.
- Ezioba, U., and Denichou, J.-M., 2014, Mapping-While-Drilling system improves well placement and field development, *Journal of Petroleum Technology*, **66**(8), 32–35. DOI: 10.2118/0814-0032-JPT.
- Farah, L.B., Farah, I.R., Bennaceur, R., Belhadj, Z., and Boussema, M.R., 2006, A Neural Network Approach for the Inversion of Multi-Scale Roughness Parameters and Soil Moisture, *Proceedings, 2006 2nd Information and Communication Technologies (ICTTA'06)*, 406–411, DOI: 10.1109/ICTTA.2006.1684404.
- Fragkiadaki, K., Agrawal, P., Levine, S., and Malik, J., 2016, Learning Visual Predictive Models of Physics for Playing Billiards, presented at the 4th International Conference on Learning Representations (ICLR 2016). <https://arxiv.org/pdf/1511.07404.pdf>. Accessed January 5, 2020.

- Goodfellow, I., Bengio, Y., and Courville, A., 2016, *Deep Learning*. MIT Press. ISBN: 978-0262035613. <http://www.deeplearningbook.org/>. Accessed January 5, 2020.
- He, K., Zhang, X., Ren, S., and Sun, J., 2015, Delving Deep Into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification, *Proceedings, 2015 IEEE International Conference On Computer Vision*, 1026–1034, DOI: 10.1109/ICCV.2015.123. [Note to author: This reference is not cited in the text. Add a citation at the appropriate place there or delete here.]
- He, K., Zhang, X., Ren, S., and Sun, J., 2016, Identity Mappings in Deep Residual Networks, *Proceedings, European 14th Conference on Computer Vision (ECCV 2016)*, 4, 630–645. DOI: 10.1007/978-3-319-46493-0_38. <https://arxiv.org/pdf/1603.05027.pdf>. Accessed January 5, 2020. [Note to author: This reference is not cited in the text. Add a citation at the appropriate place there or delete here.]
- Huang, L., Polanco, M., and Clee, E.T., 2018, Initial Experiments on Improving Seismic Data Inversion with Deep Learning, *Proceedings, New York Scientific Data Summit (NYSDS)*, 1–3, DOI: 10.1109/NYSDS.2018.8538956.
- Ince, T., Kiranyaz, S., Eren, L., Askar, M., and Gabbouj, M., 2016, Real-Time Motor Fault Detection By 1-D Convolutional Neural Networks, *IEEE Transactions on Industrial Electronics*, 63(11), 7067–7075. DOI: 10.1109/TIE.2016.2582729.
- Itskovich, G.B., and Nikitenko, M.N., 2017, Fast Inversion of MWD Transient EM Data Excited by a Pulse of an Arbitrary Shape, U.S. Patent No. 9,562,990, granted February 7, 2017.
- Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K., 2015, Spatial Transformer Networks, *Advances in Neural Information Processing Systems*, 28, 2017–2025, <https://papers.nips.cc/paper/5854-spatial-transformer-networks.pdf>. Accessed January 5, 2020.
- Jiang, J., and Fan, J.A., 2019, Global Optimization of Dielectric Metasurfaces Using a Physics-Driven Neural Network, *Nano Letters*, 19(8), 5366–5372. DOI: 10.1021/acs.nanolett.9b01857.
- Jin, Y., 2018, *Modern Deep Network Toolkits for Tensorflow-Keras, MDNT*. <https://github.com/caimagi/MDNT>. Accessed January 5, 2020.
- Johnston, A., Garg, R., Carneiro, G., and Reid, I.D., 2017, Scaling CNNs for High Resolution Volumetric Reconstruction From a Single Image, *Proceedings, 2017 IEEE International Conference on Computer Vision Workshops*, 930–939. DOI: 10.1109/ICCVW.2017.114.
- Key, K., 2009, 1D Inversion of Multicomponent, Multifrequency Marine CSEM Data: Methodology and Synthetic Studies for Resolving Thin Resistive Layers, *Geophysics*, 74(2), F9–F20. DOI: 10.1190/1.3058434.
- Kingma, D.P., and Ba, J., 2015, Adam: A method For Stochastic Optimization, presented at the 3rd International Conference on Learning Representation (ICLR 2015). <https://arxiv.org/pdf/1412.6980.pdf>. Accessed January 5, 2020.
- Lerer, A., Gross, S., and Fergus, R., 2016, Learning Physical Intuition of Block Towers by Example, *Proceedings, 33rd International Conference on Machine Learning*, 48(9), 430–438. <https://arxiv.org/pdf/1603.01312.pdf>. Accessed January 5, 2020.
- Levenberg, K., 1944, A Method for the Solution of Certain Non-Linear Problems in Least Squares, *Quarterly of Applied Mathematics*, 2(2), 164–168. DOI: 10.1090/qam/10666.
- Li, Q., Omeragic, D., Chou, L., Yang, L., Duong, K., Smits, J., Yang, J., Lau, T., Liu, C., Dworak, R., Dreullault, V., and Ye., H., 2005, New Directional Electromagnetic Tool for Proactive Geosteering and Accurate Formation Evaluation While Drilling, Paper UU, *Transactions, SPWLA 46th Annual Logging Symposium*, New Orleans, Louisiana, USA, 26–29 June.
- Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., and Yang, M.-H., 2017, Universal Style Transfer Via Feature Transforms, *Advances in Neural Information Processing Systems*, 386–396. <https://arxiv.org/pdf/1705.08086.pdf>. Accessed January 5, 2020.
- Loshchilov, I., and Hutter, F., 2019, Decoupled Weight Decay Regularization, presented at the 7th International Conference on Learning Representations (ICLR 2019). <https://arxiv.org/pdf/1711.05101.pdf>. Accessed January 5, 2020.
- Malkiel, I., Mrejen, M., Nagler, A., Arieli, U., Wolf, L., and Suchowski, H., 2018, Deep Learning for the Design of Nano-Photonic Structures, presented at the 2018 IEEE International Conference on Computational Photography (ICCP), 1-14, DOI: 10.1109/ICCPHOT.2018.8368462.
- Marquardt, D.W., 1963, An Algorithm for Least-Squares Estimation of Nonlinear Parameters, *Journal of the Society for Industrial and Applied Mathematics*, 11(2), 431–441. DOI: 10.1137/0111030.
- Omeragic, D., Dumont, A., Esmersoy, C., Habashy, T., Li, Q., Minerbo, G., Rosthal, R., Smits, and Tabanou, J., 2006, Sensitivities of Directional Electromagnetic Measurements for Well Placement and Formation Evaluation While Drilling, *SEG Technical Program Expanded Abstracts*, 1630–1634, DOI: 10.1190/1.2369833.
- Ramuhalli, P., Udpa, L., and Udpa, S.S., 2002, Electromagnetic NDE Signal Inversion by Function-Approximation Neural Networks, *IEEE Transactions on Magnetics*, 38(6), 3633–3642. DOI: 10.1109/TMAG.2002.804817.
- Seydoux, J., Legendre, E., Mirto, E., Dupuis, C., Denichou, J.-M., Bennett, N., Kutiev, G., Kuchenbecker, M., Morriss, C., and Yang, L., 2014, Full 3D Deep Directional Resistivity Measurements Optimize Well Placement and Provide Reservoir-Scale Imaging While Drilling, Paper LLL, *Transactions, SPWLA 55th Annual Logging Symposium*, Abu Dhabi, UAE, 18–22 May.
- Shen, Q., Chen, J., and Wang, H., 2018a, Data-Driven Interpretation of Ultradeep Azimuthal Propagation Resistivity Measurements: Transdimensional Stochastic Inversion and Uncertainty Quantification. *Petrophysics*, 59(6), 786–798. DOI: 10.30632/PJV59N6-2018a4.
- Shen, Q., Wu, X., Chen, J., and Han, Z., 2017, Distributed Markov Chain Monte Carlo Method on Big-Data Platform for Large-Scale Geosteering Inversion Using Directional Electromagnetic Well Logging Measurements, *Applied Computational Electromagnetics Society Journal*, 32(5), 405–412. https://aces-society.org/includes/downloadpaper.php?of=ACES_Journal_May_2017_Paper_5&nf=17-5-5.

- Accessed January 5, 2020.
- Shen, Q., Wu, X., Chen, J., Han, Z., and Huang, Y., 2018b, Solving Geosteering Inverse Problems By Stochastic Hybrid Monte Carlo Method, *Journal of Petroleum Science and Engineering*, **161**, 9–16. DOI: 10.1016/j.petrol.2017.11.031.
- Simonyan, K., and Zisserman, A., 2015, Very Deep Convolutional Networks for Large-Scale Image Recognition, presented at the 3rd International Conference on Learning Representations (ICLR 2015). <https://arxiv.org/pdf/1409.1556.pdf>. Accessed January 5, 2020.
- Tchakarov, B.J., Wang, T., Guenther, R.S., Cao, T.H., and Tang, C., 2016, Downhole Closed-Loop Geosteering Methodology, US Patent 9,273,517 granted March 1, 2016.
- Thiel, M., Bower, M., and Omeragic, D., 2018, 2D Reservoir Imaging Using Deep Directional Resistivity Measurements, *Petrophysics*, **59**(2), 218–233. DOI: 10.30632/PJV59N2-2018a7.
- Ulyanov, D., Vedaldi, A., and Lempitsky, V., 2016, Instance Normalization: The Missing Ingredient for Fast Stylization. <https://arxiv.org/pdf/1607.08022.pdf>. Accessed January 5, 2020.
- Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K., 2016, WaveNet: A Generative Model for Raw Audio, presented at the 9th ISCA Speech Synthesis Workshop, 125. <https://arxiv.org/pdf/1609.03499.pdf>. Access January 5, 2020.
- Wang, H., Shen, Q., and Chen, J., 2018, Sensitivity Study and Uncertainty Quantification of Azimuthal Propagation Resistivity Measurements, Paper R, *Transactions, SPWLA 59th Annual Logging Symposium*, London, England, UK, 2–6 June.
- Wu, J., Lu, E., Kohli, P., Freeman, B., and Tenenbaum, J., 2017, Learning to See Physics Via Visual De-Animation, *Advances in Neural Information Processing Systems (NIPS 2017)*, **30**, 153–164. <https://papers.nips.cc/paper/6620-learning-to-see-physics-via-visual-de-animation.pdf>. Accessed January 5, 2020.
- Xu, Y., Sun, K., Xie, H., Zhong, X., Mirto, E., Feng, Y., and Hong, X., 2018, Borehole Resistivity Measurement Modeling Using Machine-Learning Techniques, *Petrophysics*, **59**(6), 778–785. DOI: 10.30632/PJV59N6-2018a3.
- Zhang, R., Liu, Y., and Sun, H., 2019, Physics-guided Convolutional Neural Network (PhyCNN) for Data-Driven Seismic Response Modeling. <https://arxiv.org/pdf/1909.08118.pdf>. Accessed January 5, 2020.
- Zhang, X., Zhou, X., Lin, M., and Sun, J., 2018, Shufflenet: An Extremely Efficient Convolutional Neural Network for Mobile Devices, *Proceedings, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6848–6856. DOI: 10.1109/CVPR.2018.00716.

ABOUT THE AUTHORS

Yuchen Jin is a third-year PhD student at the Department of Electrical and Computer Engineering, University of Houston, co-advised by Dr. Jiefu Chen and Dr. Xuqing Wu.

He received his BS degree in Electronic Information from Huazhong University of Science and Technology in 2017. His research interests include machine learning, inverse problem, optimization, seismic processing, and signal processing.

Qiuyang Shen is a Research Scientist with Cyentech Consulting LLC. He received his PhD degree in Electrical Engineering from the University of Houston in 2019. His current research interests include geophysical inversions, statistical data analysis, and uncertainty quantifications.

Xuqing Wu is an Assistant Professor of the Department of Information and Logistics Technology at the University of Houston. Prior to joining the University of Houston in 2015, he was a data scientist and modeling engineer at Schlumberger. His research interests include machine learning, statistical inference, subsurface sensing, and computer vision. Dr. Wu received his PhD degree in Computer Science from the University of Houston in 2011.

Jiefu Chen is an Assistant Professor at the Department of Electrical and Computer Engineering, the University of Houston. Previously, he worked for four and a half years as a Staff Scientist for Weatherford International. Dr. Chen's research interests include computational electromagnetics, multiphysics modeling and inversion, scientific machine learning, underground and underwater wireless communication, and well logging. He holds a BS degree in engineering mechanics and a MS degree in dynamics and control, both from Dalian University of Technology, and a PhD degree in Electrical Engineering from Duke University.

Yueqin Huang is the Chief Scientist of Cyentech Consulting LLC. Previously she worked as a Geophysicist for DownUnder GeoSolutions, and as an Assistant Professor for Xiamen University. Dr. Huang's research interests include computational acoustics and seismology, ground penetrating radar, and well logging. She received her PhD degree in Electrical Engineering from Xiamen University in 2011.