



(51) International Patent Classification:

E21B 7/04 (2006.01)

(21) International Application Number:

PCT/US2020/038108

(22) International Filing Date:

17 June 2020 (17.06.2020)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/862,886

18 June 2019 (18.06.2019)

US

(71) Applicants: UNIVERSITY OF HOUSTON SYSTEM

[US/US]; 4800 Calhoun Road, Houston, Texas 77004 (US).

CYENTECH CONSULTING LLC [US/US]; 14330

Hazeldale Dr., Cypress, Texas 77429-5574 (US).

(72) Inventors: HUANG, Yueqin; 14330 Hazeldale Dr., Cy-

press, Texas 77429 (US). WU, Xuqing; 6919 Dunbarton

Dr., Sugar Land, Texas 77479 (US). CHEN, Jiefu; 14330

Hazeldale Dr., Cypress, Texas 77429 (US).

(74) Agent: LIKOUREZOS, George; CARTER, DELUCA &

FARRELL LLP, 576 Broadhollow Road, Melville, New

York 11747 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ,

CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO,

DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN,

HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP,

KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME,

MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ,

OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,

SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR,

TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH,

GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ,

UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ,

TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK,

EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV,

MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,

(54) Title: SYSTEMS AND METHODS FOR SOLVING GEOSTEERING INVERSE PROBLEMS IN DOWNHOLE ENVIRONMENTS USING A DEEP NEURAL NETWORK

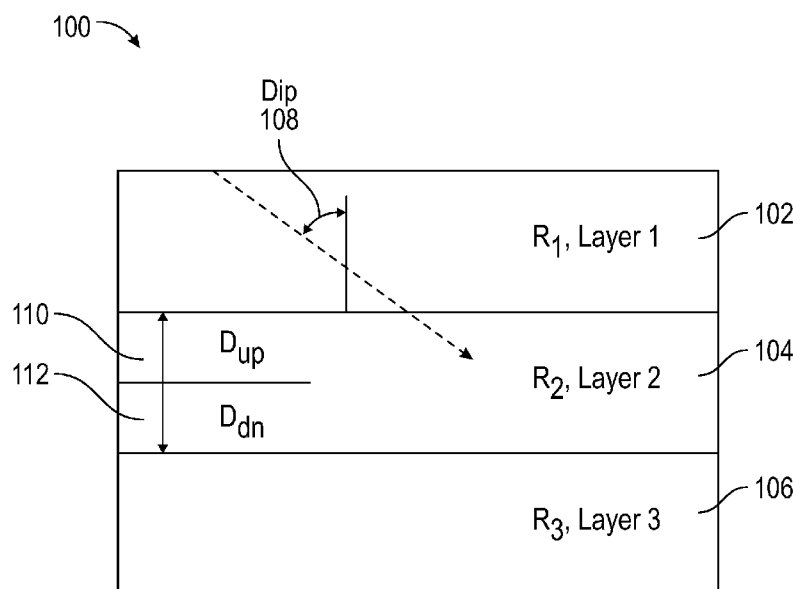


FIG. 1

(57) Abstract: An aspect of the present disclosure provides a system for directional drilling in downhole environments, the system includes a processor and a memory. The memory includes instructions, which, when executed by the processor, cause the system to receive logging data and predict, by a neural network, an earth model based on the received logging data.



TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,
KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

SYSTEMS AND METHODS FOR SOLVING GEOSTEERING INVERSE PROBLEMS IN DOWNHOLE ENVIRONMENTS USING A DEEP NEURAL NETWORK

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of and priority to U.S. Provisional Patent Application Serial No. 62/862,886, filed on June 18, 2019, the entire contents of which is incorporated by reference herein.

TECHNICAL FIELD

[0002] The present disclosure relates to directional drilling in industrial practice. More particularly, the present disclosure relates to a system and method for solving geosteering inverse problems in downhole environments using an algorithm. The algorithm includes a physics-driven deep neural network implemented by combining a deep neural network with a predefined forward model function which follows physical rules.

SUMMARY

[0003] Embodiments of the present disclosure are described in detail with reference to the drawings wherein like reference numerals identify similar or identical elements.

[0004] An aspect of the present disclosure provides a method for directional drilling in downhole environments. The method includes receiving logging data and predicting an earth model based on the received logging data.

[0005] In an aspect of the present disclosure, the predicting may further include inputting the logging response data into a neural network and predicting, by the neural network, parameters of the predicted earth model. The neural network configured to

predict parameters of a predicted earth model based on the logging response.

[0006] In another aspect of the present disclosure, the neural network may include a deep neural network.

[0007] In yet another aspect of the present disclosure, the neural network may be trained. The training the neural network may include receiving a set of earth models and corresponding logging data as a training dataset, initializing weighting parameters of the neural network, inputting the logging data into the neural network, inputting the predicted physical subsurface model into the forward model, generating a synthetic logging response, and determining a physics-driven loss function.

[0008] In a further aspect of the present disclosure, determining the physics-driven loss function may include determining a first difference between the synthetic logging response and the received logging data and determining a second difference between the predicted earth model and the received earth model.

[0009] In yet a further aspect of the present disclosure, training may further include backpropagating the physics-driven loss function through the neural network and updating the weighting parameters of the neural network in a case where the weighted sum of the first difference and the second difference are larger than a predetermined threshold.

[0010] In an aspect of the present disclosure, the physics-driven loss function includes model misfit information and data misfit information.

[0011] In another aspect of the present disclosure, receiving logging response data may further include collected logging response data by a logging tool.

[0012] In yet another aspect of the present disclosure, the logging tool may include at

least one transmitter/receiver pair.

[0013] In a further aspect of the present disclosure, the at least one transmitter/receiver pair may be deployed in different directions and frequency bands from a second transmitter/receiver pair.

[0014] An aspect of the present disclosure provides a method for directional drilling in downhole environments. The system includes at least one processor and a memory. The memory includes instructions, which, when executed by the processor, cause the system to receive logging data and predict an earth model based on the received logging data.

[0015] In an aspect of the present disclosure, when predicting, the instructions, when executed, may further cause the system to input the logging data into a neural network, the neural network configured to predict parameters of a predicted physical subsurface earth model based on the logging data and predict, by the neural network, parameters of the predicted earth model.

[0016] In an aspect of the present disclosure, the neural network may include a deep neural network.

[0017] In another aspect of the present disclosure, when the neural network is being trained, the instructions, when executed, may further cause the system to receive a set of earth models and corresponding logging data as a training dataset, initialize weighting parameters of the neural network, input the logging data into the neural network, input the predicted physical subsurface model into the forward model, generate a synthetic logging response, and determine a physics-driven loss function.

[0018] In yet another aspect of the present disclosure, determining the physics-driven

loss function may include determining a first difference between the synthetic logging response and the received logging data and determining a second difference between the predicted earth model and the received earth model.

[0019] In a further aspect of the present disclosure, training further includes backpropagating the physics-driven loss function through the neural network and updating the weighting parameters of the neural network in a case where the weighted sum of the first difference and the second difference are larger than a predetermined threshold.

[0020] In yet a further aspect of the present disclosure, the physics-driven loss function may include model misfit information and data misfit information.

[0021] In yet another aspect of the present disclosure, when receiving logging response data, the instructions, when executed, may further cause the system to collect logging response data by a logging tool.

[0022] In a further aspect of the present disclosure, the logging tool may include at least one transmitter/receiver pair.

[0023] An aspect of the present disclosure provides a non-transitory computer-readable medium storing instructions which, when executed by a processor, cause the processor to perform a method including receiving logging data and predicting an earth model based on the logging data.

[0024] Further details and aspects of exemplary embodiments of the present disclosure are described in more detail below with reference to the appended figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0025] A better understanding of the features and advantages of the disclosed technology will be obtained by reference to the following detailed description that sets forth illustrative embodiments, in which the principles of the technology are utilized, and the accompanying figures of which:

[0026] FIG. 1 illustrates parameters for an exemplary 3-layer geosteering model, in accordance with the present disclosure;

[0027] FIG. 2 is a diagram of an exemplary system for training a deep neural network for solving geosteering inversion problems, in accordance with the present disclosure;

[0028] FIG. 3 is a diagram of an exemplary system for training a deep neural network using both data misfits and model misfits, in accordance with the present disclosure;

[0029] FIG. 4 illustrates an exemplary integrated training loss for an exemplary training set, in accordance with the present disclosure;

[0030] FIGS. 5A-C illustrate a comparison of an exemplary inversion result, in accordance with the present disclosure;

[0031] FIGS. 6A-B illustrate an exemplary comparison of observed and synthetic values of selected curves, in accordance with the present disclosure;

[0032] FIG. 7 is a diagram of an azimuthal resistivity tool, in accordance with the present disclosure;

[0033] FIG. 8 is a diagram of an exemplary 3-layer model, in accordance with the present disclosure;

[0034] FIG. 9 is a diagram of the network architecture for the physics-driven deep neural network, in accordance with the present disclosure;

[0035] FIGS. 10A-B illustrates a visualization of the training loss, in accordance with the present disclosure;

[0036] FIGS. 11A-D illustrates an example of predicted earth model $\hat{m}$ from different methods, in accordance with the present disclosure;

[0037] FIGS. 12A-C illustrates exemplary predicted measurements $\hat{d}$ from different methods for an example, in accordance with the present disclosure;

[0038] FIG. 13 illustrates exemplary numerical testing results of different exemplary methods, in accordance with the present disclosure;

[0039] FIGS. 14A-B illustrate exemplary misfits during the training phase for an exemplary 3-layer artificial neural network, in accordance with the present disclosure;

[0040] FIGS. 15A-B illustrates an exemplary predicted earth model sample from two exemplary artificial neural networks, in accordance with the present disclosure;

[0041] FIG. 16 is a table which illustrates an exemplary numeric evaluation for the model misfit and data misfit, in accordance with the present disclosure;

[0042] FIG. 17 is a table which illustrates two exemplary geosteering inversion scenarios, in accordance with the present disclosure;

[0043] FIG. 18 is a table that illustrates exemplary time consumptions for predicting one sample, in accordance with the present disclosure;

[0044] FIG. 19 is a table that illustrates exemplary memory consumption for predicting one sample, in accordance with the present disclosure;

[0045] FIG. 20 is a table that illustrates exemplary estimated misfits for predicting one sample, in accordance with the present disclosure;

[0046] FIG. 21 is a table that illustrates exemplary estimated misfits for predicting 80 points with an artificial neural network, in accordance with the present disclosure;

[0047] FIG. 22A is a flowchart that illustrates an exemplary method for training a Physics-Driven Deep Neural Network, in accordance with the present disclosure;

[0048] FIG. 22B is a flowchart that illustrates an exemplary method for directional drilling in downhole environments using the Physics-Driven Deep Neural Network, in accordance with the present disclosure; and

[0049] FIG. 23 is a high-level block diagram of an exemplary computing system that may be used with directional drilling in downhole systems, in accordance with aspects of the present disclosure.

[0050] Further details and aspects of various embodiments of the present disclosure are described in more detail below with reference to the appended figures.

DETAILED DESCRIPTION

[0051] This disclosure relates to systems and methods for directional drilling in downhole environments. More specifically, an aspect of the present disclosure provides a method for solving geosteering inverse problems in downhole environments using a physics-driven deep neural network.

[0052] Although the present disclosure will be described in terms of specific embodiments, it will be readily apparent to those skilled in this art that various modifications, rearrangements, and substitutions may be made without departing from the spirit of the present disclosure. The scope of the present disclosure is defined by the claims appended hereto.

[0053] For purposes of promoting an understanding of the principles of the present disclosure, reference will now be made to exemplary embodiments illustrated in the drawings, and specific language will be used to describe the same. It will nevertheless be understood that no limitation of the scope of the present disclosure is thereby intended. Any alterations and further modifications of the inventive features illustrated herein, and any additional applications of the principles of the present disclosure as illustrated herein, which would occur to one skilled in the relevant art and having possession of this disclosure, are to be considered within the scope of the present disclosure.

[0054] Geosteering is the act of adjusting a well trajectory on the fly. A highly accurate inversion technique may be useful for conducting a successful geosteering service. The traditional lookup table approach is unable to provide high inversion accuracy due to the hardware limitation on the size of the table. The disclosed method includes replacing the lookup table with a deep neural network (DNN). Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher level features from the raw input. Typically, a DNN has multiple hidden layers. Compared to the lookup table, the method provided in the present disclosure is faster and requires much less memory space. Further, various embodiments include a physics-driven loss function that includes both model misfit and data misfit information, which significantly improves inversion performance. Moreover, the inversion accuracy and the training loss of the DNN can be improved by introducing a data misfit that measures the disagreement between the observed logging measurement and the output of a forward model. The network may be trained under the loss function provided in the present disclosure, which balances the model misfit and the data misfit.

[0055] With reference to FIG. 1, an exemplary 3-layer geosteering model 100 assuming the drilling tool is in the center layer and unknown parameters for a 3-layer geosteering model 100 as described in the present disclosure. The unknown parameters may include resistivities (R_1, R_2, R_3) 102, 104, 106, distance-to-boundary (D_{up}, D_{dn}) 110, 112, and relative dip angle (Dip) 108. For example, supposing that one parameter (Dip) 108 is known, various embodiments include inverting the remaining unknown parameters (R_1, R_2, R_3, D_{up} , and D_{dn}) 102, 104, 106, 110, 112, that characterize the structure of underground formation from azimuthal resistivity measurements, and provide guidance to keep the drilling trajectory in the desired bed.

[0056] Given a geosteering forward model $y = f(u)$, where u is the vector of model parameters (*i.e.*, resistivity, distance to boundary, and relative dip angle), the inverse relationship can be written as $u = f^{-1}(\tilde{y})$, where f^{-1} defines the inverse mapping and $\tilde{y}$ is the logging tool responses. As most inverse problems are fundamentally undetermined (ill-posed) when the parameter space is large, measurements are sparse, and local minima are unavoidable. The problem may be solved by minimizing $\Omega(u) + \lambda R(u)$, where Ω is the objective function, and R serves the role of a regularizer with regularization constant $\lambda > 0$. Under the least-square criterion, the objective function may be written as $\|\tilde{y} - f(u)\|^2$, which minimizes the data misfit between measurement curves and forward responses.

[0057] Solving a forward problem includes calculating the synthetic measurements ($d \in \mathbb{R}^N$) of a physical model ($\mathcal{F}$) given its earth model parameter set m , *i.e.*, $d = \mathcal{F}(m)$, where $\mathcal{F}$ is derived based on Maxwell's equations. The inverse relationship may be written as $\hat{m} = \mathcal{F}^{-1}(\tilde{d})$, where $\mathcal{F}^{-1}$ defines the inverse mapping and $\tilde{d}$ is the observed

measurements. Hence, solving geosteering inverse problems is aimed at solving $\mathcal{F}^{-1}$. In a case where no additional information is available, the solution for $\hat{m}$ is either highly unstable, or highly undetermined, or both. A regularization method may be employed in order to obtain a stable solution. A differential method may be used to solve the complex forward function ($\mathcal{F}(\cdot)$). The gradients of the forward model may be computed by estimating the Jacobian matrix.

[0058] With reference to FIG. 2, a diagram of an exemplary system 201 for training a DNN 204 to solve geosteering inversion problems is shown in accordance with the present disclosure. In methods, an approach that follows the physics and leverages the regression power of the DNN 204 may be used. The first component of the system may be a DNN 204 for predicting physical subsurface earth model parameters 206 based on a logging response 202, and the second component of the system may be a physics engine driven by a forward model 208 to recover the logging response given a subsurface model. During the training stage, the system 201 may sequentially compute the model parameters based on logging responses and generates corresponding responses 209 through the forward model. The system may adapt to both the neural network and the differentiable forward model 208, which can be jointly trained.

[0059] In machine learning, a DNN may include a convolutional neural network (CNN), which is a class of artificial neural network (ANN), most commonly applied to analyzing data. The convolutional aspect of a CNN relates to applying matrix processing operations to localized portions of the data, and the results of those operations (which can involve dozens of different parallel and serial calculations) are sets of many features that are used to train neural networks. A CNN typically includes convolution layers,

activation function layers, and pooling (typically max pooling) layers to reduce dimensionality without losing too many features. Additional information may be included in the operations that generate these features. Providing unique information that yields features that give the neural networks information can be used to ultimately provide an aggregate way to differentiate between different data input to the neural networks.

[0060] With reference to FIG. 3, an exemplary diagram of two neural networks is shown in accordance with the present disclosure. Unlike the conventional supervised learning method that only considers the model misfit inaccuracy of the inverted result, various embodiments include data misfit (logging responses) driven by a physical forward model (the physics engine). In order to meet the physical constraints of the model, the present disclosure includes a scaling layer to regularize the outputs of the network as follows:

$$\text{Scale}(x, x_{\min}, x_{\max}) = \frac{(x_{\max} - x_{\min})}{2} \tanh(\gamma x + \beta) + \frac{(x_{\max} + x_{\min})}{2},$$

where γ, β are trainable. The present disclosure denotes $x_{\min}$ and $x_{\max}$ as the lower bound and the upper bound of the scaled vector x , respectively. To avoid gradient vanishing, generally a more flexible boundary than the actual one may be used. By adding trainable parameters, γ, β , the gradients back-propagated from the loss function can be re-calibrated so that they match with those of the boundary positions.

[0061] With reference to FIG. 4, an exemplary comparison of the training loss with and without a physics engine is shown in accordance with the present disclosure. The training loss decreases significantly after explicitly incorporating the data misfit driven by the forward model. The loss function $\mathcal{L}$ may include two parts, $\mathcal{L}_{\text{ml}}$ and $\mathcal{L}_{\text{dl}}$. These two loss functions may be measured by mean square errors (MSE). The first part ($\mathcal{L}_{\text{ml}}$) is the

model misfit compared to the ground truth of the geosteering model. Ground truth is a term used to refer to information provided by direct observation (i.e., empirical evidence) as opposed to information provided by inference. The second part ($\mathcal{L}_{dl}$) is the data misfit by comparing observed logging responses to the output of the forward model. The integrated loss function is defined: $\mathcal{L} = \frac{1}{2}(\mathcal{L}_{ml} + \mathcal{L}_{dl})$. For example, an Adam optimizer with a learning rate of 1e-3 to train the network for 80,000 steps may be deployed. An Adam (adaptive moment estimation) optimizer is an adaptive learning rate method, which means, it computes individual learning rates for different parameters. Adam optimizers use estimations of first and second moments of gradient to adapt the learning rate for each weight of the neural network. For example, the batch size may be set to 64. A 3-layer earth model may include five parameters to be decided. The exemplary training set may be created by uniformly sampling 16 values for each parameter, and there are a total of 2.36×10^6 .

[0062] With reference to FIGS. 5A-C, a geosteering model, and its inversion results are shown in accordance with the present disclosure. With reference to FIGS. 6A and B, data fit on selected logging response curves is shown in accordance with the present disclosure. In all cases, the method provided in the present disclosure achieved more accurate inversion results by leveraging the physics-driven DNN.

[0063] With reference to FIG. 7, a diagram of an azimuthal resistivity tool is shown in accordance with the present disclosure. An azimuthal resistivity logging tool 700 may be used to collect observed measurements. The logging tool 700 may be equipped with both transmitting antennas and receiving antennas. Both the transmitting antennas and the receiving antennas may be deployed in different directions and frequency bands. 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. In each specific position, the collected results from the logging tool may be referred to as a group of observed measurements, or curves ($\tilde{d} \in \mathbb{R}^N$, where N is the number of curves). In the process of geosteering using logging-while-drilling (LWD) azimuthal resistivity measurements, experiments, and observations may include physical theories, which in turn may be used to predict the outcome of experiments.

[0064] With reference to FIG. 8, an illustration of an exemplary 3-layer model is shown, in accordance with the present disclosure. For example, the underground formation may be viewed as a 3-layer model, and five parameters may be used to describe the underground formation. Apart from three resistivities, (R_1 , R_2 , and R_3) the boundaries between two adjacent layers (for example, if the tool is in the middle layer, there would be an upper boundary D_{up} and a lower boundary D_{dn}), and the relative dip angle of the logging tool, may also be used. Each group of the six parameters may be referred to as an “earth model” ($m \in \mathbb{R}^6$).

[0065] With reference to FIG. 9, a diagram of the network architecture for the disclosed physics-driven deep neural network (PhyDNN) 900 is shown. The PhyDNN 900 may be used for solving geosteering inversion. A differentiable forward model may be used to regulate the backpropagation process. The PhyDNN 900 generally uses observed measurements 902 (e.g., a logging response), as an input to DNN 204. The DNN 204 predicts parameters of a predicted physical subsurface earth model 206. The parameters of the predicted physical subsurface earth model 206 are used by a forward model 208 to calculate synthetic measurements 914 (e.g., $d \in \mathbb{R}^N$) of the predicted

physical subsurface earth model 206. A physics-driven loss function 918 $\mathcal{L}$ includes both the model misfit 910 $\mathcal{L}_{ml}$ and data misfit 916 $\mathcal{L}_{dl}$ information. These two loss functions may include mean square errors (MSE). The first part ($\mathcal{L}_{ml}$) is the model misfit 910 by comparing the predicted physical subsurface earth model 206 to the ground truth (e.g., a real earth model 908). The second part ($\mathcal{L}_{dl}$) is the data misfit 916 by comparing observed measurements 902 (e.g., observed logging responses) to the output of the forward model 208 (e.g., the synthetic measurements 914).

[0066] With reference to FIGS. 10A-B, visualization of the training loss is shown, in accordance with the present disclosure. The training metrics for both model misfit and data misfit are visualized in the FIG 10A and FIG. 10B, respectively. FIG 10A shows that model misfit is not influenced by whether or a physical forward model may be used during the backpropagation. FIG. 10B shows that data misfit may be significantly reduced by using a physical forward model during the backpropagation. Comparing PhyDNN to the data-driven only network, data misfit may be significantly reduced by incorporating a physical model to drive the backpropagation during the training.

[0067] With reference to FIGS. 11A-D, diagrams that illustrate an example of predicted earth model $\hat{m}$ from different methods is shown in accordance with the present disclosure. The x-axis represents the distance along the horizontal direction. The z-axis is along the vertical direction, the green line represents the trajectory of the wellbore, and the different resistivities from different layers in the three-layer model are illustrated by different colors.

[0068] With reference to FIGS. 12A-C, diagrams that illustrate predicted measurements $\hat{d}$ from different exemplary methods are shown, in accordance with the

present disclosure. The prediction $\hat{d}$ is compared with the ground truth d . The x-axis represents the distance along the horizontal direction. Since $d, \hat{d} \in \mathbb{R}^N$, five representative measurements are chosen to draw five curves. The number of the measurement is notated near the y-axis, which represents the signal value.

[0069] The training phase of the PhyDNN may be formulated as follows,

$$\arg \min_{\Theta} \mathbb{E}_{(m,d)} [\mathcal{L}_{ml}(m, d, \Theta)],$$

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

where (m, d) is a sample that is randomly selected from the training set, $\mathcal{D}_{\Theta}$ represents a DNN with tunable parameters Θ . Hence, $\mathcal{D}_{\Theta}(d)$ is the prediction for an earth model, and $\mathcal{L}_{ml}$ is the model misfit. Training the PhyDNN may include learning the network parameters Θ to minimize the error between the predictions and true earth models. After the network gets well-trained, given a group of measurements d , the prediction could be given by $\hat{m} = \mathcal{D}_{\Theta}(d)$, *i.e.* $\mathcal{D}_{\Theta}$ serves as $\mathcal{F}^{-1}$. $\mathcal{F}^{-1}$ is implicitly learned from the statistical features of the dataset, since the train set follows the distribution such that $m \sim \mathcal{F}(d)$.

[0070] A physical forward model may be introduced as a constraint into the network explicitly. The training phase of the PhyDNN may be formulated by

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

$$\mathcal{L}_{dl}(d, \Theta) = \|\mathcal{F}(\mathcal{D}_{\Theta}(d)) - d\|_2^2.$$

To optimize the loss function defined in the above equation, calculation of the gradient of the data misfit $\mathcal{L}_{dl}$ is required. A Jacobian matrix for $\mathcal{F}(\hat{m})$, may be denoted as $J(d) := \frac{\partial \mathcal{F}}{\partial \hat{m}} \Big|_{\hat{m}=\mathcal{D}_{\Theta}(d)}$.

The derivative of $\mathcal{L}_{dl}$ can be written as

$$\nabla_{\text{dl}}\Theta = -2 \left(d - \mathcal{F}(\mathcal{D}_{\Theta}(d)) \right) J(d) \frac{\partial \hat{m}}{\partial \Theta} \Big|_{\hat{m}=\mathcal{D}_{\Theta}(d)}.$$

This equation shows that the gradient of a differential physical model $\mathcal{F}$ will affect the back-propagation of the network during the training to minimize both model misfit.

[0071] The predefined forward model function $\mathcal{F}(\cdot)$ may be used to generate a lookup table as the training set, where each sample (m, d) is called a “point.” m is the model parameter and $m := (R_1, R_2, R_3, D_{\text{up}}, D_{\text{dn}})$, where $R_1, R_2, R_3 \in (10^{-1}, 10^2)[\Omega \cdot \text{m}]$, $D_{\text{up}} \in (-25.0, 0)[\text{ft}]$ and $D_{\text{dn}} \in (0, 25.0)[\text{ft}]$. Dip is fixed at 90 degree in this case. Such a configuration ensures that $D_{\text{up}} < D_{\text{dn}}$, which indicates that the logging tool is kept in the middle layer and the two boundaries are restricted in the sensing range of the logging tool. For each resistivity, for example, the present disclosure provides dividing the logarithmic range into 16 intervals. For each boundary, for example, the present disclosure provides dividing the depth range into 24 intervals. For example, $16^3 \times 24^2$, *i.e.*, $2.3,6 \times 10^6$ points may be used in the training set. In practice, if the points are stored in the ascending order of m , it will be unnecessary to store m in the table. For example, each d , if it has 92 values and each value takes 8 bytes, the total size of the lookup table is approximately 1.61 GB.

[0072] For example, there may be, for example, 100 samples in the testing set, and each sample may have 80 points, *i.e.*, 80 pairs of (m, d) . The 80 points in the same sample may share the same resistivities, while the boundaries may change continuously. It is contemplated that, all parameters of m may be generated randomly.

[0073] The network may be trained and tested to demonstrate its effectiveness in solving geosteering inverse problems. For example, the network may be deployed on

Python® 3.5+ and TensorFlow® r1.4+. For example, the forward model function may be implemented in C++ with a Python® -C-API wrapper. OpenMP® may be used to enable parallel computing for the forward model to accelerate the CPU computation. During the training and testing phases, the program, except the implementation of the forward model function, may be run on both a CPU and a GPU. The Jacobian matrix of the forward model may be estimated in a routine function implemented in C++ with a Python® -C-API wrapper. Further, the Jacobian matrix of the forward model may subsequently be registered to the TensorFlow® via `tf.py_func`. Further, a tensor may then be defined and it may serve as a part of the network architecture.

[0074] For example, the training may be performed on an NVIDIA ® DGX workstation, equipped with GPUs. Further, parallel computing may be used for the forward model function, in which case all CPUs may be utilized. One or more GPUs may be used to train the DNN. It is contemplated that other computer systems may be used to train the above-mentioned methods, and the present disclosure should not be construed to be limited to the above-mentioned computer system.

[0075] For example, during each step of the training, 64 pairs of (m, d) may be randomly selected from the training as a batch during the batch-wise optimization. For example, the network may be trained for a total of 80K steps. The training process for the network may be divided into two phases. For example, in the first phase, for the first 0~20K steps, the DNN may be pre-trained solely by $\mathcal{L}_{ml}$, and in the second phase, during the 20K~80K steps, the DNN may be trained by both $\mathcal{L}_{ml}$ and $\mathcal{L}_{dl}$, *i.e.*, the gradient from data misfit would be back-propagated to DNN. In addition, as an exemplary comparison, a data-driven DNN may be trained. The data-driven DNN may be trained, for example,

with only the model misfit loss function $\mathcal{L}_{\text{ml}}$ for 80K steps. Except for the absence of the physical forward model, all configurations of this data-driven DNN (including the network configurations) may be the same with the PhyDNN.

[0076] With reference to FIG. 13, a diagram that compares three methods is shown in accordance with the present disclosure. The three methods compared in FIG. 13 include: (1) lookup table (2) data-driven network (3) PhyDNN. Each method was tested on 100 earth models. Among these three exemplary methods, the lookup table has the highest model and data misfits on average. The prediction from the data-driven network returns small model misfit but high data misfit. The PhyDNN has the lowest overall misfit. Further, as an example, each test result was evaluated by average model misfit and data misfit among 80 observations in the same sample. The x -axis represents the average model misfit and the y -axis represents the average data misfit. Each point represents the estimation for a sample with 80 observations. A comparison between the predicted measurements and the observed ones for the same sample are shown. The results show that the predicted earth model is very inaccurate using the lookup table method even if there is a good match for the measurements. Further, the data-driven network may predict the boundary positions well. The PhyDNN obtained the most accurate earth model with low measurement misfit.

[0077] With reference to FIGS. 14A-B, exemplary misfits during the training phase for an exemplary 3-layer artificial neural network (ANN) are shown in accordance with the present disclosure. The misfits are measured by mean squared error (MSE).

[0078] Referring to FIGS. 15A-B, the predicted earth model sample from two exemplary ANNs, which have the same network structure, is shown in accordance with

the present disclosure. This sample containing 80 points is the same sample shown in FIG. 11.

[0079] With reference to FIG. 16, a table that illustrates the mean squared error (MSE) numeric evaluation results for the model misfit and data misfit is shown in accordance with the present disclosure. The proposed model in the present disclosure was tested on a set of randomly generated earth models. Numerical evaluation results show that both data fit and model fit were reduced by using the physics-driven approach.

[0080] Referring to FIG. 17, a table which illustrates the advantages and disadvantages of two geosteering inversion methods is shown in accordance with the present disclosure. There are two possible scenarios of calling geosteering inversion: one is performed on the surface and the other is conducted downhole. For geosteering inversion on the surface, the logging tool sends collected data back to the ground, and the data can be analyzed by computers on site or on the cloud. The bottleneck of this method is the low data transmission rate. As a result, not all the collected data can be transmitted back in time. Thus, the inversion could only be applied on incomplete data. In downhole computing scenarios, the logging tool which is equipped with the microprocessors could process data directly. However, due to the extreme underground environment, the computational resources deployed downhole are very limited. The lack of data for geosteering data processing on the surface would cause the inverse problem to become more undetermined. Since it may be difficult to overcome the problem brought by inadequate data, various embodiments include a method with low computational costs for directly solving the inverse problem in downhole environments using a PhyDNN.

[0081] With reference to FIGS. 18 and 19, the time consumptions and the memory consumptions of different exemplary methods are shown, respectively. The computational efficiency of different methods including the lookup table, PhyDNN, and an iterative algorithm (LMA) are estimated, in which both the time consumption and memory consumption are taken into consideration. Referring again to FIG. 18, a table that illustrates exemplary time consumption for predicting one sample consisting of 80 points is shown. 10K and 2M indicate different lookup table size. Referring again to FIG. 19, a table that illustrates exemplary memory consumption for predicting one sample of 80 points is shown. In this table, only memory space used to save the parameters for the prediction model are considered.

[0082] With reference to FIG. 20, a table that illustrates the average inversion accuracy for different methods is shown in accordance with the present disclosure. In this table, estimated misfits for predicting one exemplary sample of 80 points is shown. In this table, both the data and model misfits are considered. The results indicate that the overall inversion accuracy of the PhyDNN is comparable to LMA. The computing speed, however, is approximately 500 times faster. The PhyDNN result may be used as the initial guess for the deterministic method, which will help the system converge much faster.

[0083] With reference to FIG. 21, a table that illustrates estimated misfits for predicting 80 points with an Artificial Neural Network is shown in accordance with the present disclosure. In this table, both the data misfit and model misfit are considered.

[0084] With reference to FIG. 22A, a flowchart that illustrates an exemplary method for training a Physics-Driven Deep Neural Network for directional drilling in downhole

environments using, is shown in accordance with the present disclosure. Initially, at step 2206, the method collects a set of physical subsurface models (e.g., earth models) and corresponding logging response data as a training dataset.

[0085] At step 2208, the method initializes the weighting parameters of the neural network. The neural network may include pretrained weight parameters.

[0086] At step 2210, the method inputs the logging response data into a neural network. The neural network predicts a physical subsurface model based on the logging response data. The physics-driven deep neural network (PhyDNN) may capture the physics for solving geosteering inversion problems. For example, a DNN may be used.

[0087] At step 2212, the method inputs the predicted physical subsurface model into the forward model and generates a synthetic logging response.

[0088] At step 2214, the method calculates the difference between the synthetic logging responses and the collected logging response data and further calculates the difference between the predicted subsurface model and the collected physical subsurface model.

[0089] At step 2216, if the differences are larger than a predetermined threshold, the method backpropagates the difference through the neural network and updates the weighting parameters of the neural network. If the differences are not larger than a predetermined threshold, end the training.

[0090] In various embodiments, the system may adapt to both the neural network and the differentiable forward model, which can be jointly trained.

[0091] With reference to FIG. 22B, a flowchart that illustrates an exemplary method for directional drilling in downhole environments using a Physics-Driven Deep Neural Network, is shown in accordance with the present disclosure.

[0092] Initially, at step 2202, the method receives logging response data based on a physical logging response. Logging response data may be collected by a logging tool, for example, by an azimuthal response tool. The logging tool may be equipped with both transmitting antennas and receiving antennas. Both the two kinds of antennas may be deployed in different directions and frequency bands.

[0093] Next, at step 2204, the method inputs the logging response data into a neural network. The neural network predicts a physical subsurface model (e.g., an earth model) based on the logging response data. The physics-driven deep neural network (PhyDNN) may capture the physics for solving geosteering inversion problems. For example, a DNN may be used.

[0094] Referring now to FIG. 23, high-level block diagram of an exemplary computing system 200 that may be used with directional drilling in downhole systems of the present disclosure. The computing system 200 may include a processor or controller 220 that may be or include, for example, one or more central processing unit processor(s) (CPU), one or more Graphics Processing Unit(s) (GPU or GPGPU), a chip or any suitable computing or computational device, an operating system, a memory 230, a storage 210, input devices and output devices. Modules or equipment for collecting or receiving or transmitting logging data collected by the logging device 212 (FIG. 2) may be or include, or may be executed by, the computing system 200 shown in FIG. 3. A communication component 240 of the computing system 200 may allow communications with remote or

external devices, e.g., via the Internet or another network, via radio, or via a suitable network protocol such as File Transfer Protocol (FTP), etc. The neural networks of the disclosed method may be trained on the computing system or on a remote computing system (e.g., a remote server).

[0095] A database can be located in the storage 210. The term “storage” may refer to any device or material from which information may be capable of being accessed, reproduced, and/or held in an electromagnetic or optical form for access by a computer processor. A storage may be, for example, volatile memory such as RAM, non-volatile memory, which permanently hold digital data until purposely erased, such as flash memory, magnetic devices such as hard disk drives, and optical media such as a CD, DVD, Blu-ray disc, or the like.

[0096] Certain embodiments of the present disclosure may include some, all, or none of the above advantages and/or one or more other advantages readily apparent to those skilled in the art from the drawings, descriptions, and claims included herein. Moreover, while specific advantages have been enumerated above, the various embodiments of the present disclosure may include all, some, or none of the enumerated advantages and/or other advantages not specifically enumerated above.

[0097] The embodiments disclosed herein are examples of the disclosure and may be embodied in various forms. For instance, although certain embodiments herein are described as separate embodiments, each of the embodiments herein may be combined with one or more of the other embodiments herein. Specific structural and functional details disclosed herein are not to be interpreted as limiting, but as a basis for the claims and as a representative basis for teaching one skilled in the art to variously employ the present disclosure in virtually any

appropriately detailed structure. Like reference numerals may refer to similar or identical elements throughout the description of the figures.

[0098] The phrases “in an embodiment,” “in embodiments,” “in various embodiments,” “in some embodiments,” or “in other embodiments” may each refer to one or more of the same or different embodiments in accordance with the present disclosure. A phrase in the form “A or B” means “(A), (B), or (A and B).” A phrase in the form “at least one of A, B, or C” means “(A); (B); (C); (A and B); (A and C); (B and C); or (A, B, and C).”

[0099] Any of the herein described methods, programs, algorithms, or codes may be converted to, or expressed in, a programming language or computer program. The terms “programming language” and “computer program,” as used herein, each include any language used to specify instructions to a computer, and include (but is not limited to) the following languages and their derivatives: Assembler, Basic, Batch files, BCPL, C, C+, C++, Delphi, Fortran, Java, JavaScript, machine code, operating system command languages, Pascal, Perl, PL1, scripting languages, Visual Basic, metalanguages which themselves specify programs, and all first, second, third, fourth, fifth, or further generation computer languages. Also included are database and other data schemas, and any other meta-languages. No distinction is made between languages which are interpreted, compiled, or use both compiled and interpreted approaches. No distinction is made between compiled and source versions of a program. Thus, reference to a program, where the programming language could exist in more than one state (such as source, compiled, object, or linked) is a reference to any and all such states. Reference to a program may encompass the actual instructions and/or the intent of those instructions.

[0100] It should be understood that the foregoing description is only illustrative of the present disclosure. Various alternatives and modifications can be devised by those skilled in the art without departing from the disclosure. Accordingly, the present disclosure is intended to embrace all such alternatives, modifications and variances. The embodiments described with reference to the attached drawing figures are presented only to demonstrate certain examples of the disclosure. Other elements, steps, methods, and techniques that are insubstantially different from those described above and/or in the appended claims are also intended to be within the scope of the disclosure.

CLAIMS

WHAT IS CLAIMED IS:

1. A method for directional drilling in downhole environments, the method comprising:
 - receiving logging data; and
 - predicting an earth model based on the received logging data.
2. The method of claim 1, wherein the predicting further includes:
 - inputting the logging data into a neural network, the neural network configured to predict parameters of a predicted physical subsurface earth model based on the logging data; and
 - predicting, by the neural network, parameters of the predicted earth model.
3. The method of claim 2, wherein the neural network includes a deep neural network.
4. The method of claim 2, wherein the neural network is trained, and
 - wherein the training the neural network includes:
 - receiving a set of earth models and corresponding logging data as a training dataset;
 - initializing weighting parameters of the neural network;
 - inputting the logging data into the neural network;
 - inputting the predicted physical subsurface model into a forward model;
 - generating a synthetic logging response; and
 - determining a physics-driven loss function.

5. The method of claim 4, wherein determining the physics-driven loss function comprises:
 - determining a first difference between the predicted logging response and the received logging data; and
 - determining a second difference between the predicted earth model and the received earth model.
6. The method of claim 5, wherein training further comprises:
 - backpropagating the physics-driven loss function through the neural network and
 - updating the weighting parameters of the neural network in a case where a weighted sum of the first difference and the second difference are larger than a predetermined threshold.
7. The method of claim 6, wherein the physics-driven loss function includes model misfit information and data misfit information.
8. The method of claim 1, wherein receiving logging response data further comprises collected logging response data by a logging tool.
9. The method of claim 8, wherein the logging tool includes at least one transmitter/receiver pair.
10. The method of claim 9, wherein the at least one transmitter/receiver pair is deployed in a different direction and a different frequency band from a second transmitter/receiver pair.
11. A system for directional drilling in downhole environments, the system comprises:

at least one processor; and

a memory, including instructions, which when executed by the processor, cause the system to:

- receive logging data; and
- predict an earth model based on the received logging data.

12. The system of claim 11, wherein when predicting, the instructions, when executed, further cause the system to:

- input the logging data into a neural network, the neural network configured to predict parameters of a predicted physical subsurface earth model based on the logging data; and
- predict, by the neural network, parameters of the predicted earth model.

13. The system of claim 12, wherein the neural network includes a deep neural network.

14. The system of claim 12, wherein when the neural network is being trained, the instructions, when executed, further cause the system to:

- receive a set of earth models and corresponding logging data as a training dataset;
- initialize weighting parameters of the neural network;
- input the logging data into the neural network;
- input the predicted physical subsurface model into a forward model;
- generate a synthetic logging response; and
- determine a physics-driven loss function.

15. The system of claim 14, wherein determining the physics-driven loss function comprises:

determining a first difference between the synthetic logging response and the received logging data; and

determining a second difference between the predicted earth model and the received earth model.

16. The system of claim 15, wherein training further comprises:

backpropagating the physics-driven loss function through the neural network and updating the weighting parameters of the neural network in a case where a weighted sum of the first difference and the second difference are larger than a predetermined threshold.

17. The system of claim 16, wherein the physics-driven loss function includes model misfit information and data misfit information.

18. The system of claim 11, wherein when receiving logging response data, the instructions, when executed, further cause the system to:

collect logging response data by a logging tool.

19. The system of claim 18, wherein the logging tool includes at least one transmitter/receiver pair.

20. A non-transitory computer-readable medium storing instructions which, when executed by a processor, cause the processor to perform a method comprising:

receiving logging data; and

predicting an earth model based on the logging data.

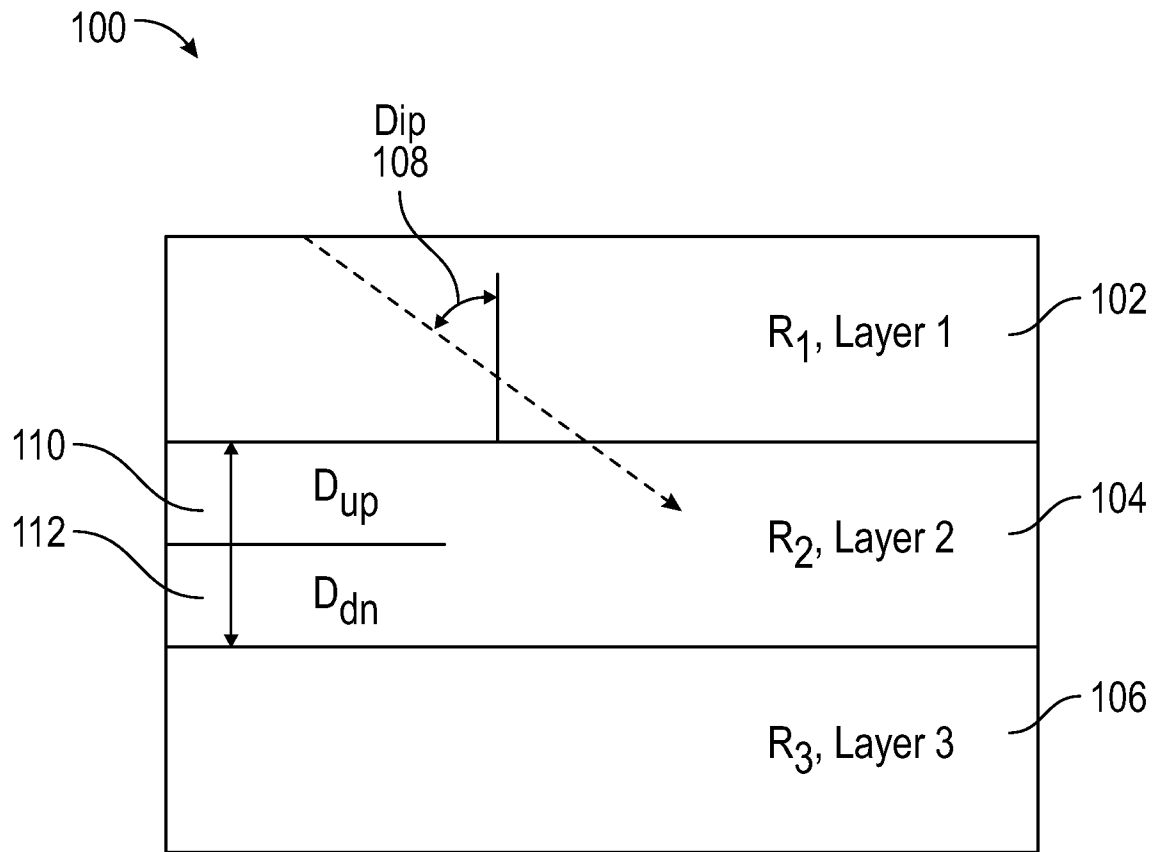


FIG. 1

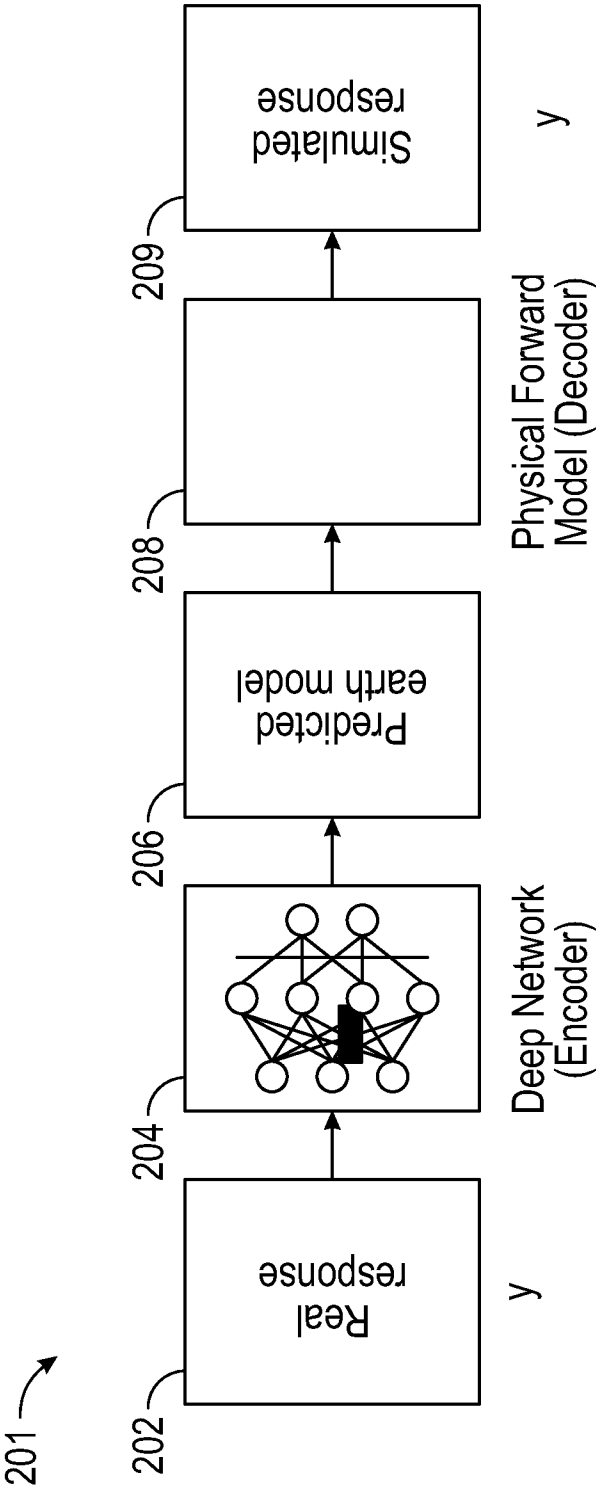


FIG. 2

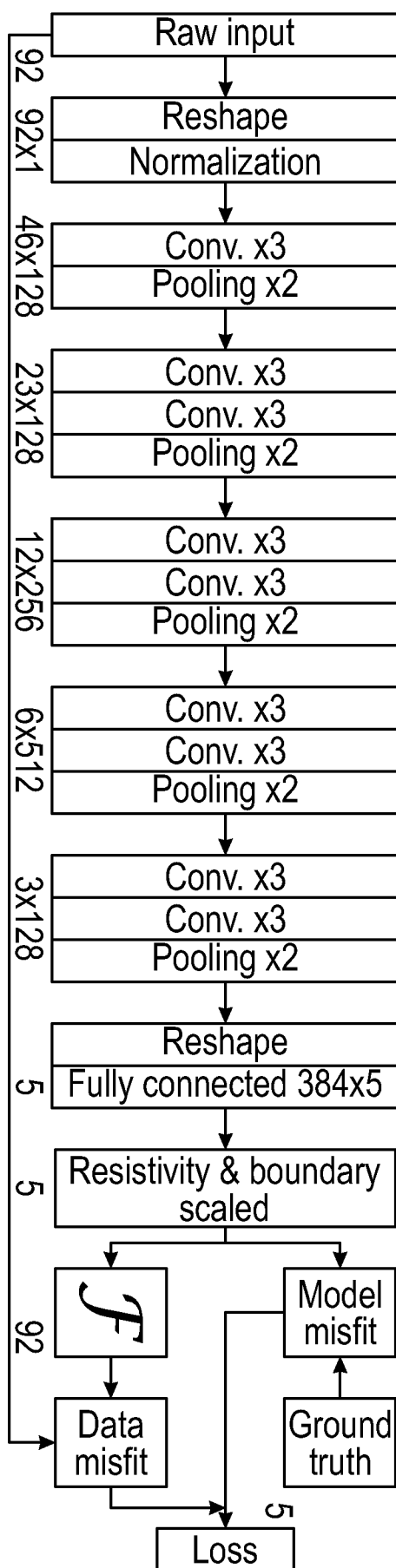


FIG. 3

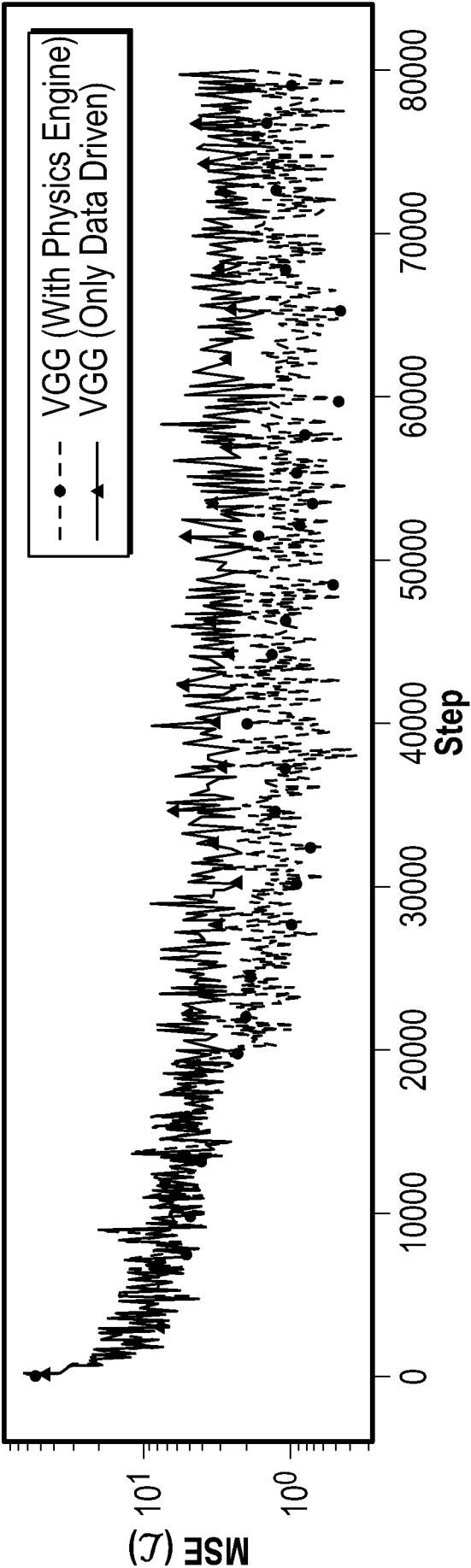
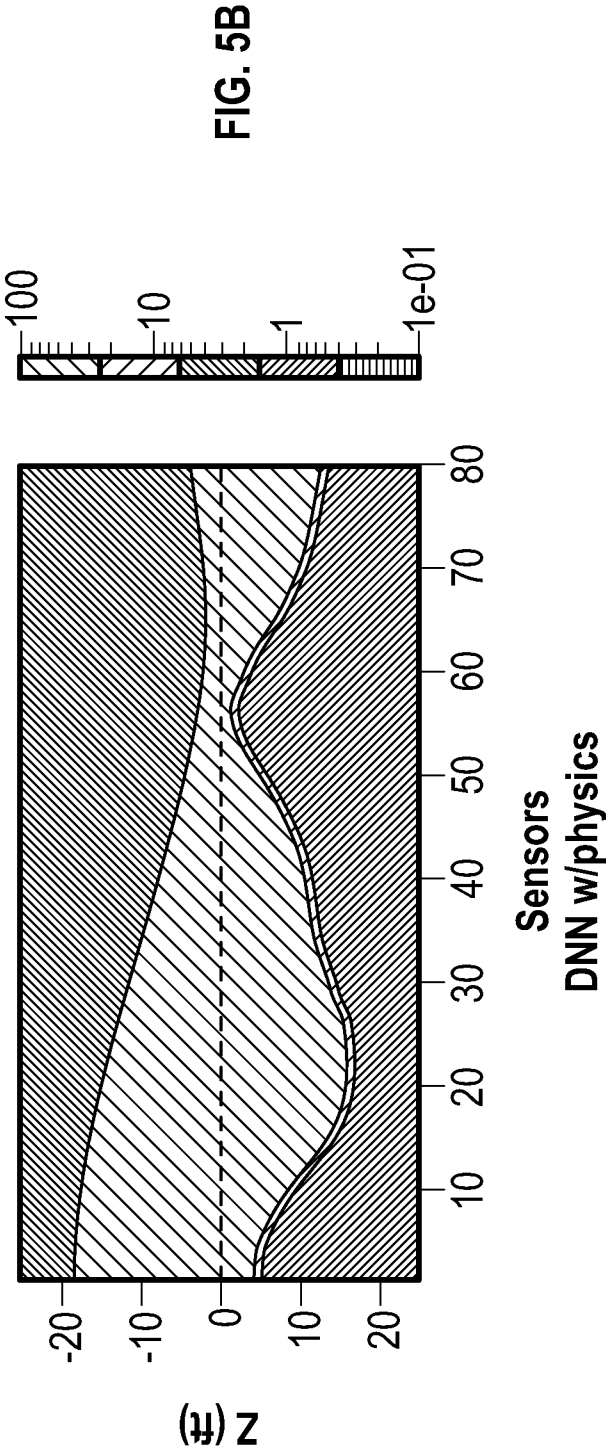
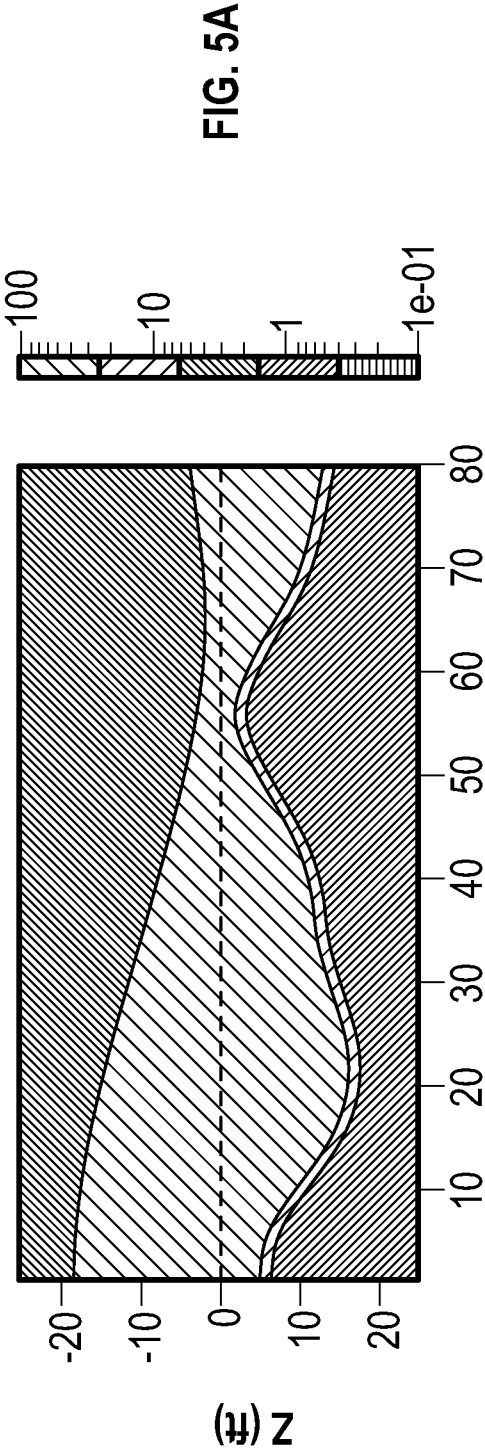
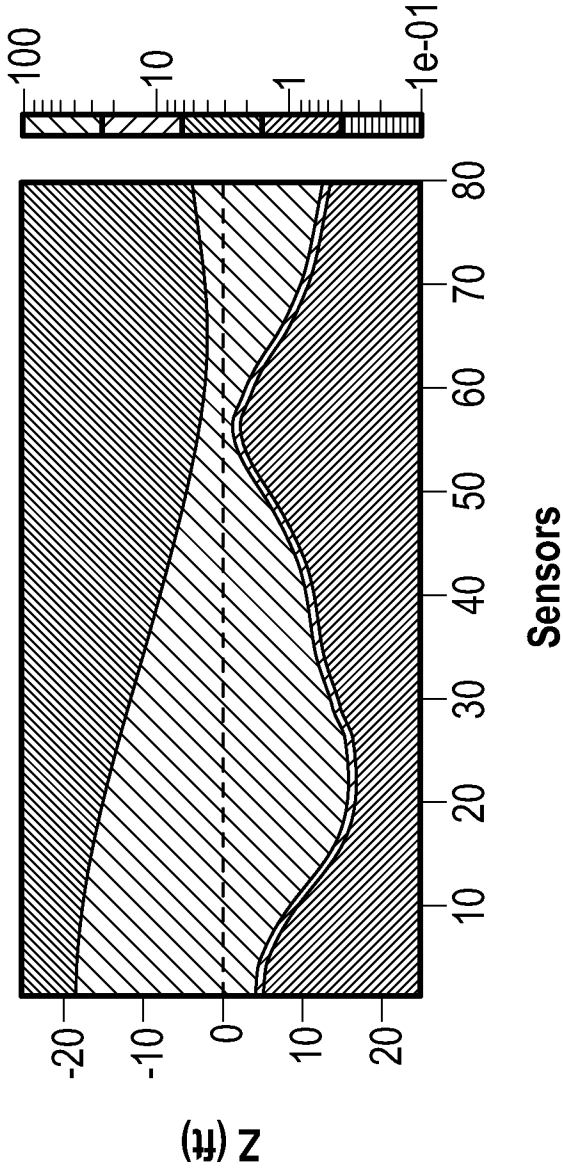


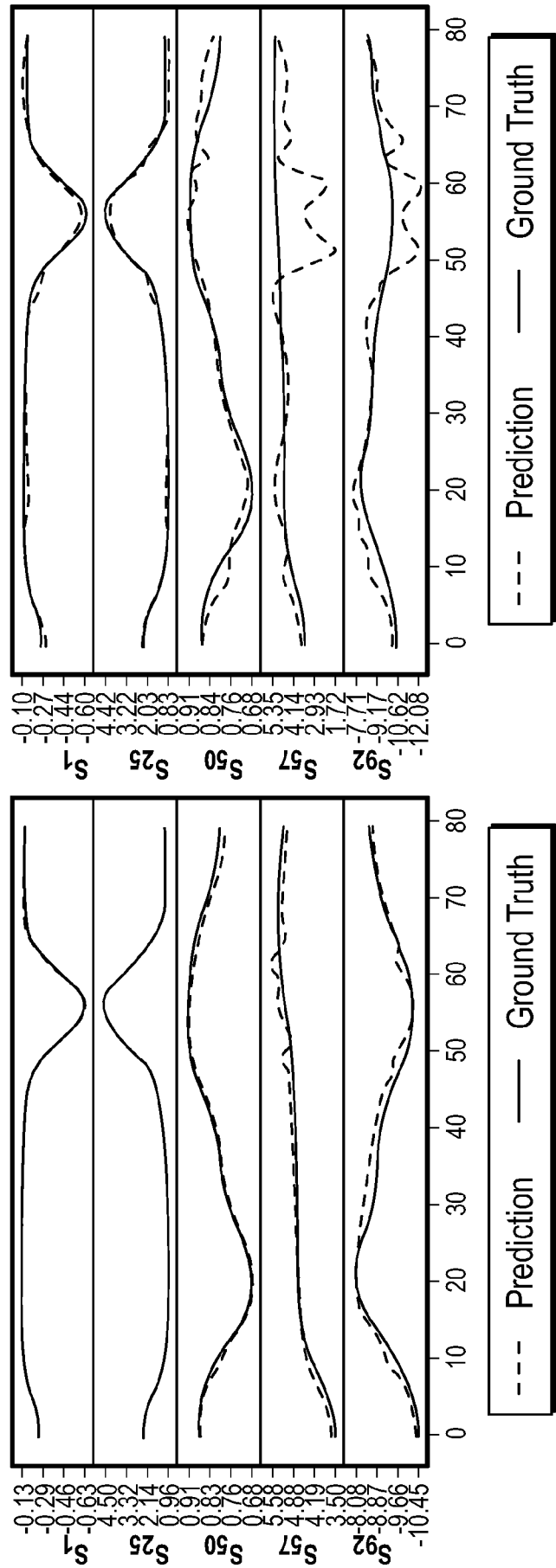
FIG. 4





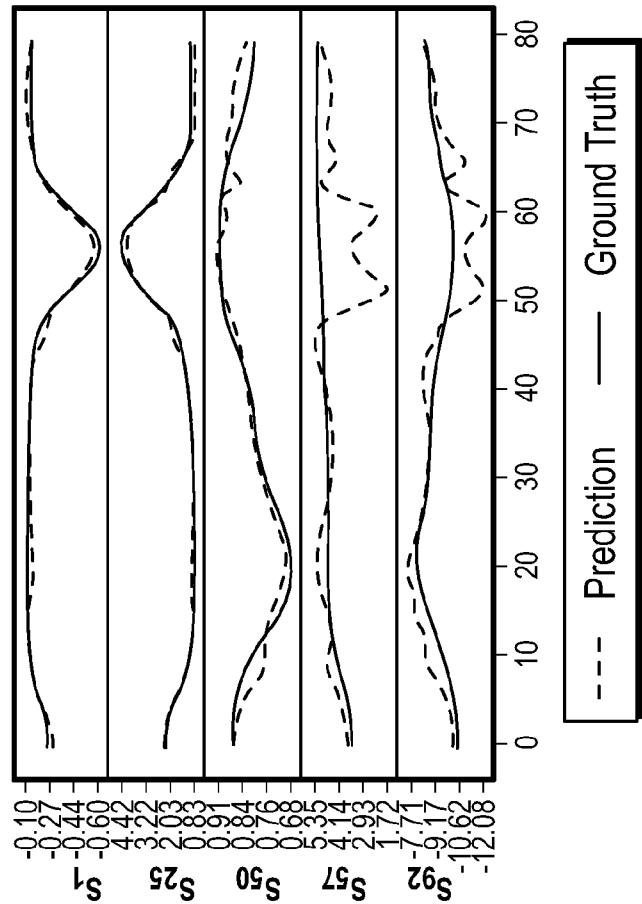
DNN w/o physics

FIG.5C



(a) With Physics Engine

FIG. 6A



(b) With Physics Engine

FIG. 6B

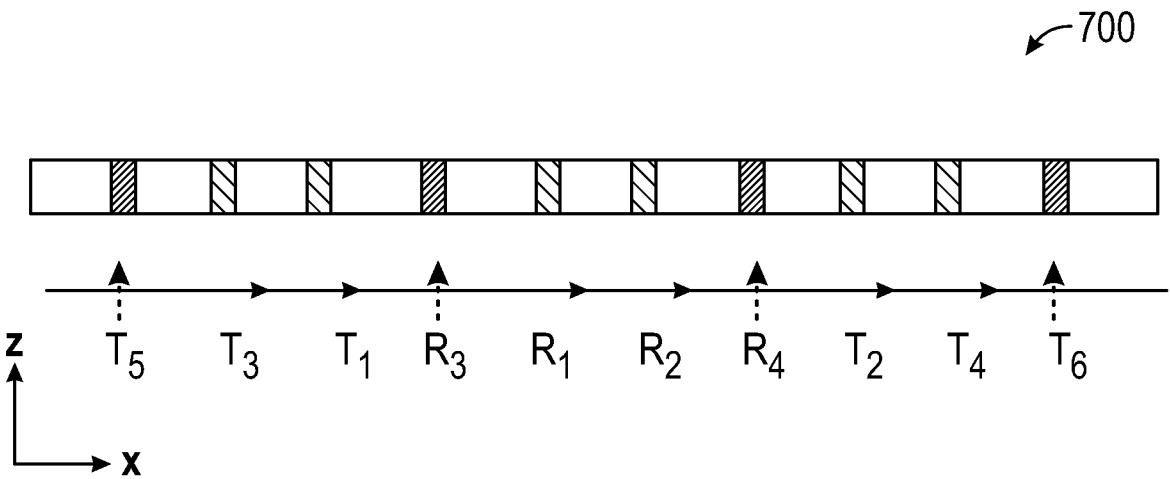


FIG. 7

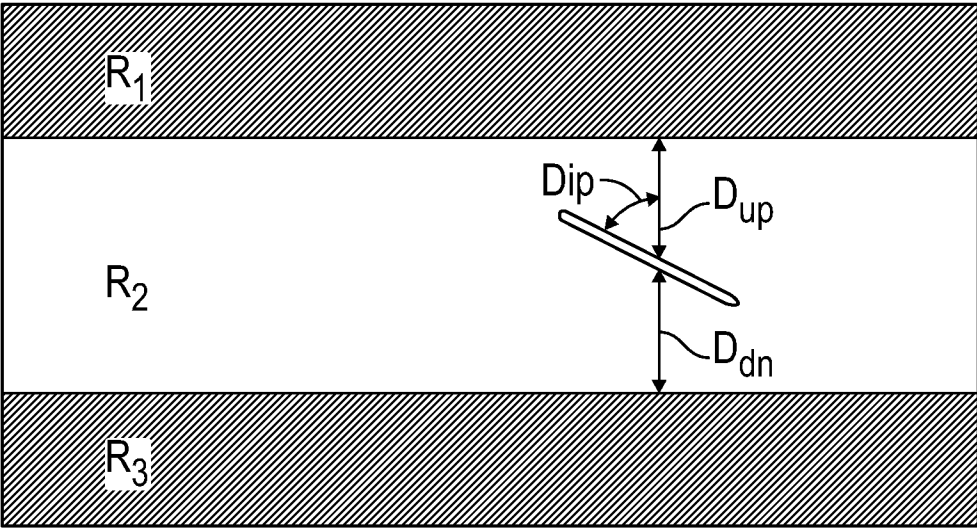


FIG. 8

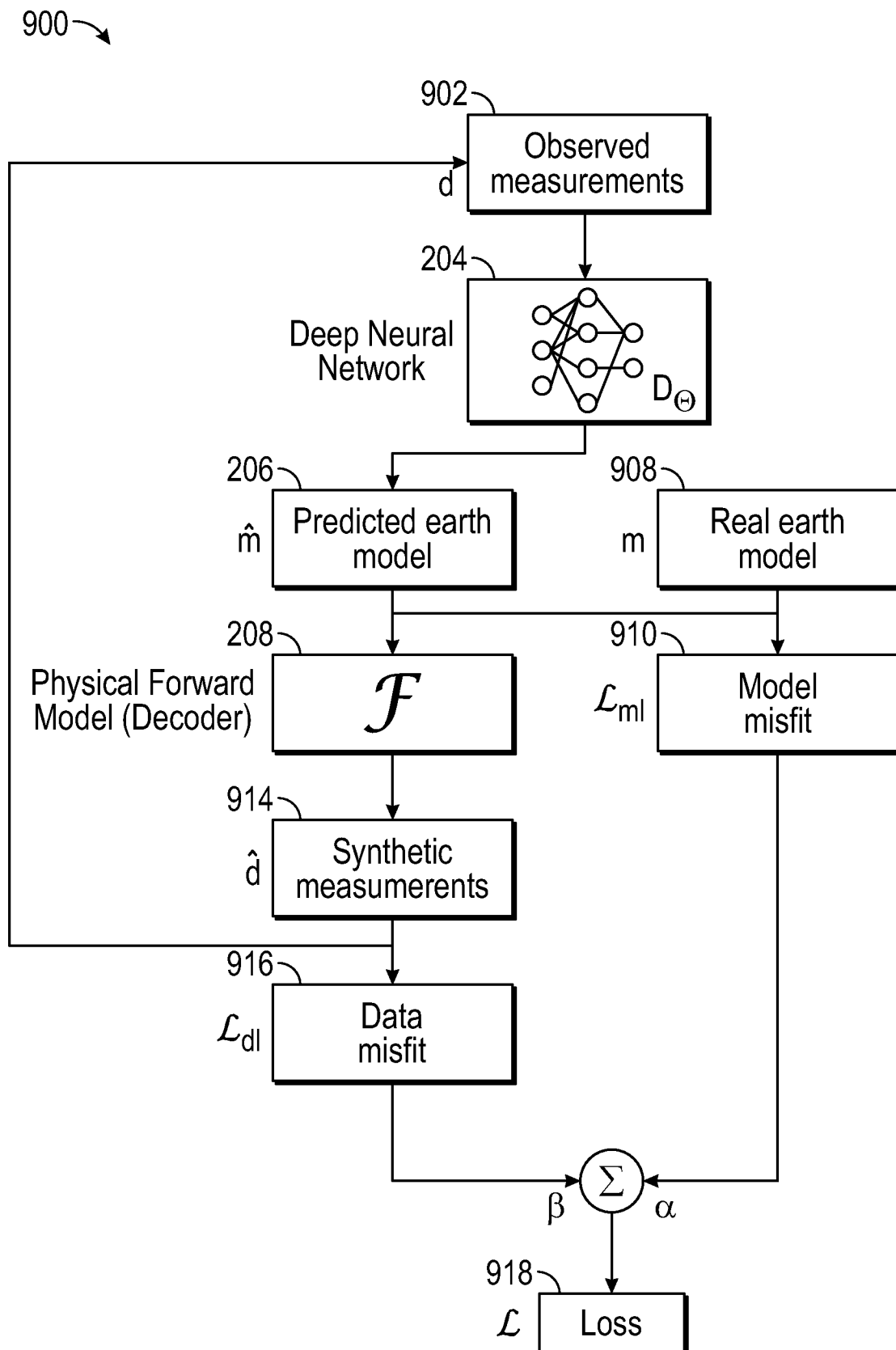


FIG. 9

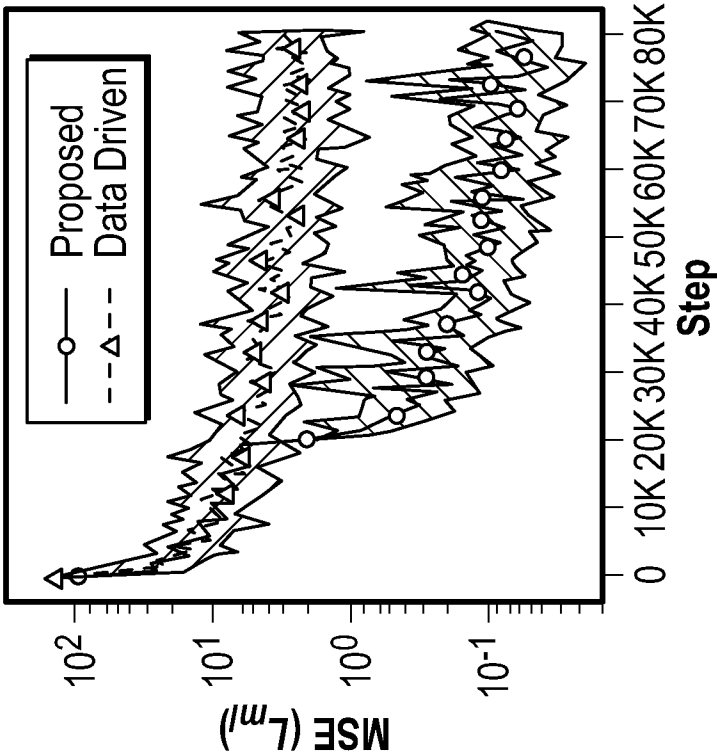


FIG. 10B

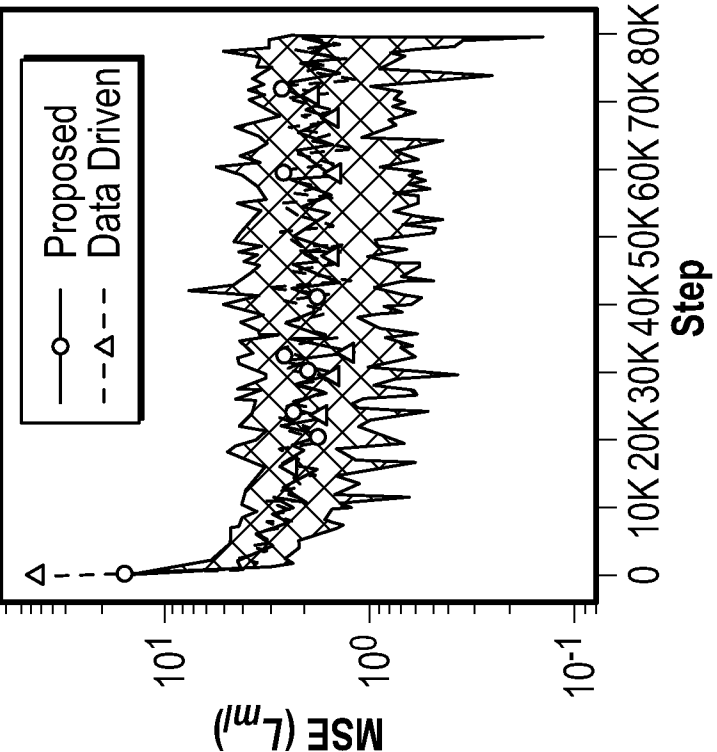
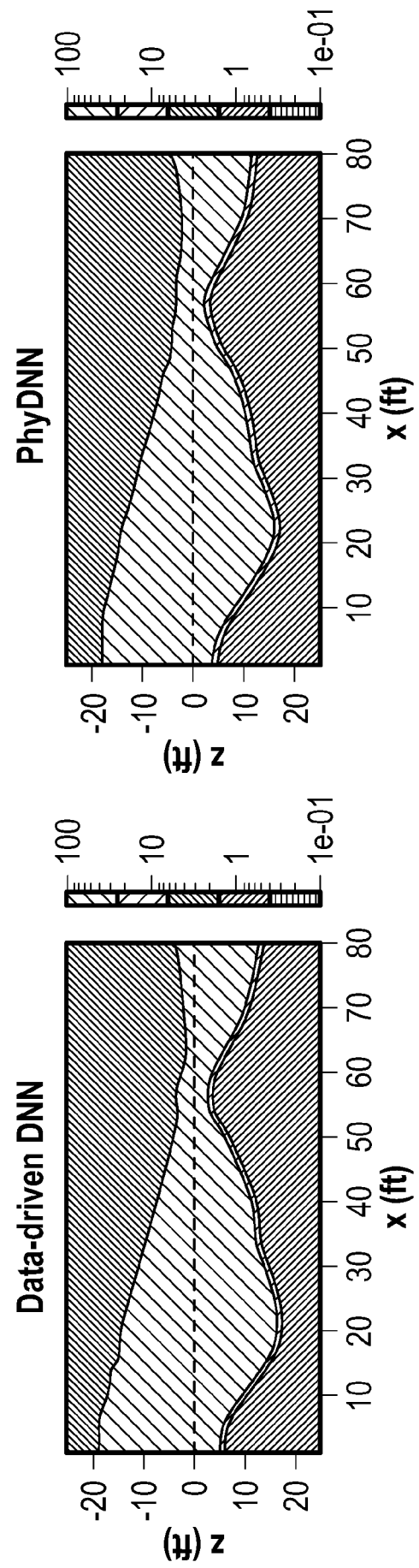
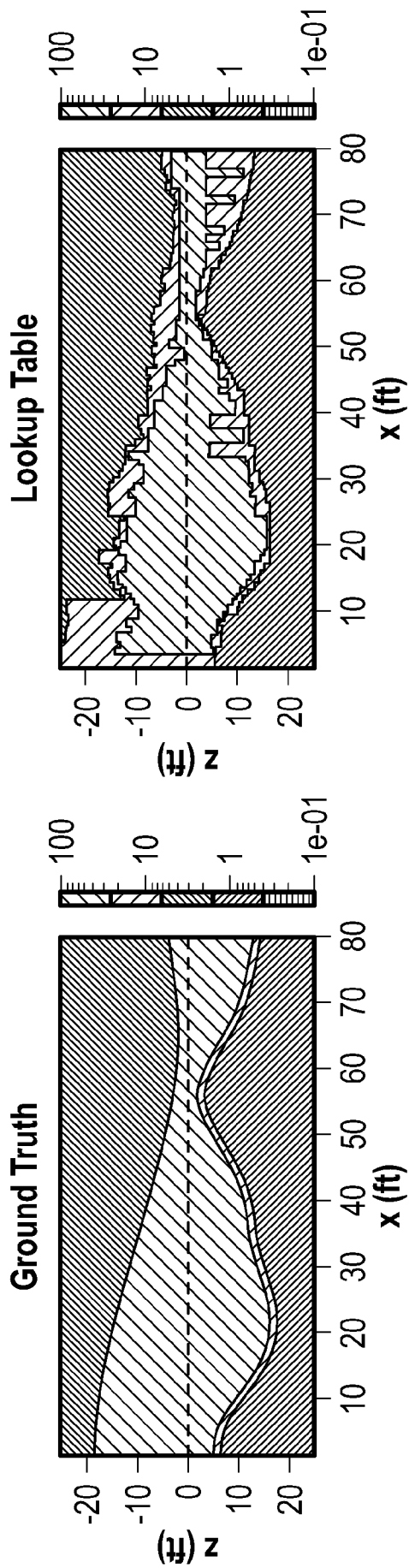


FIG. 10A



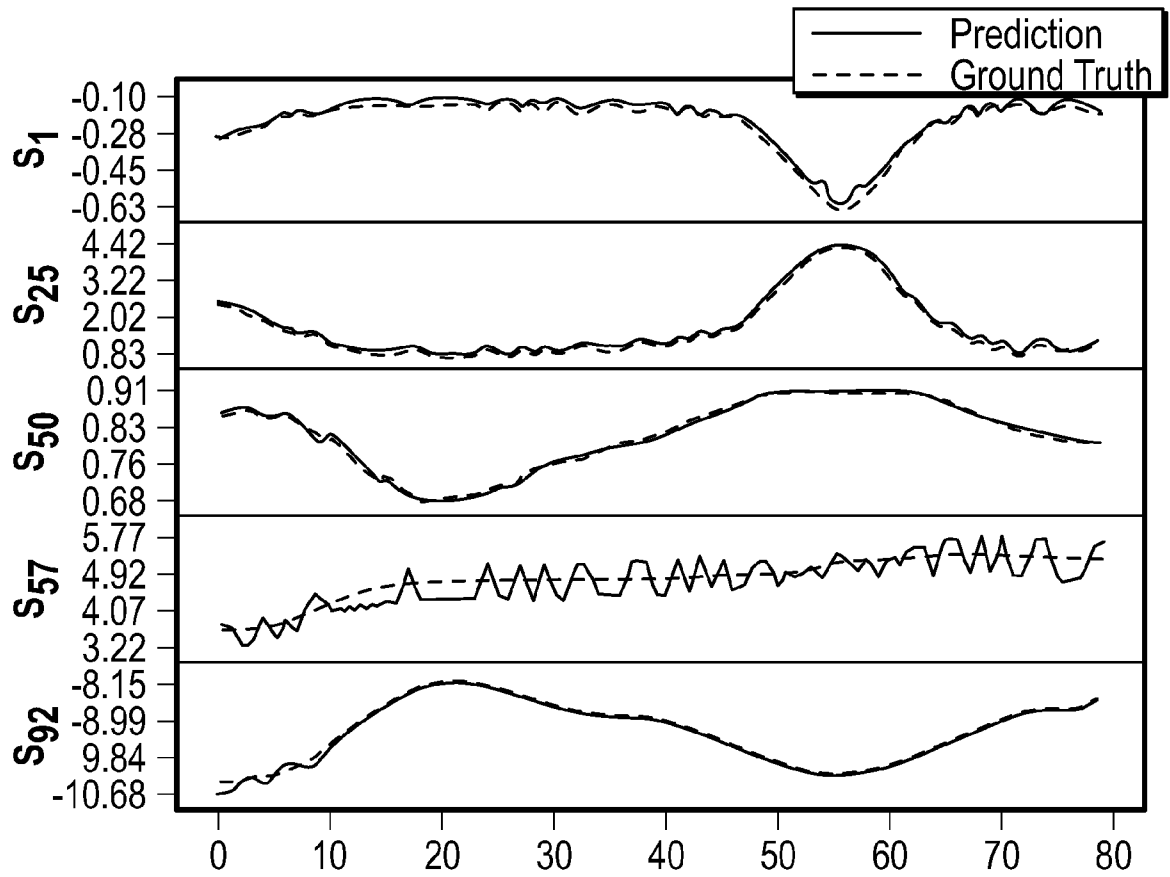


FIG. 12A

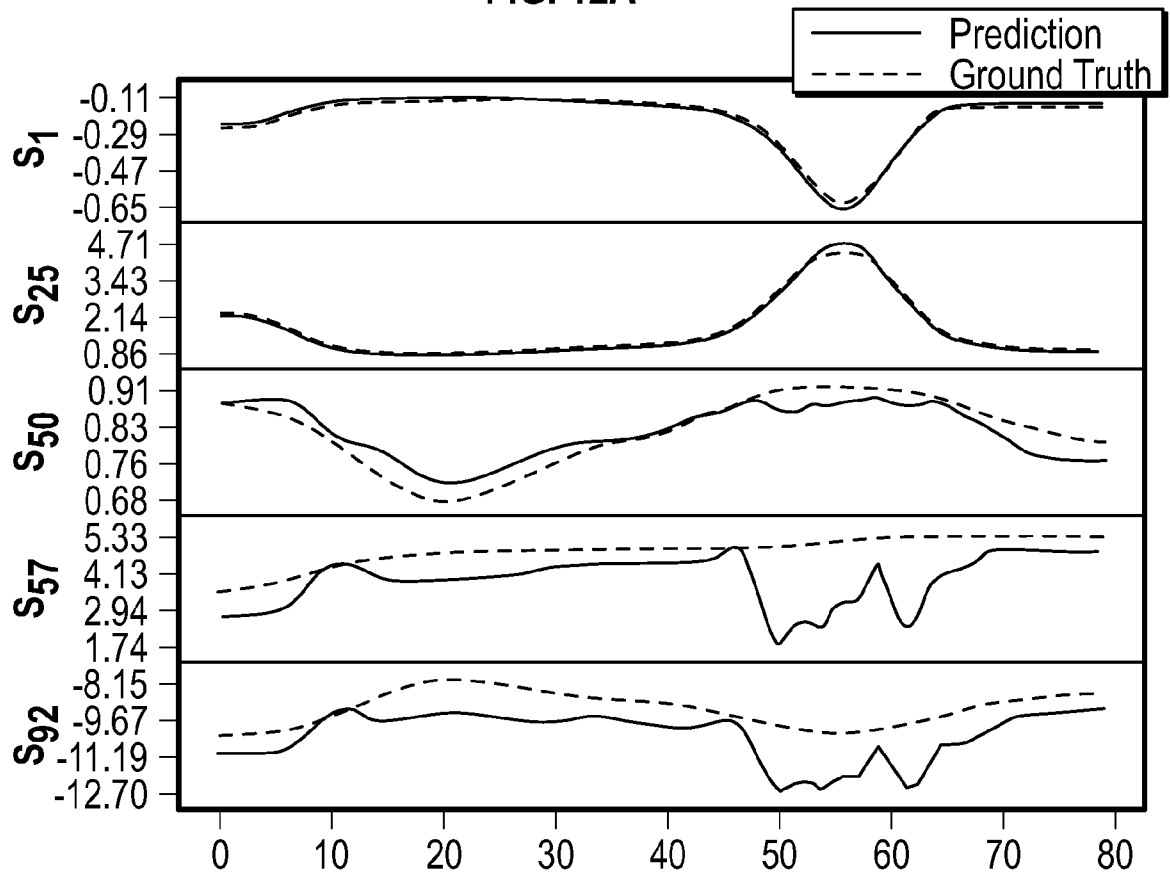


FIG. 12B

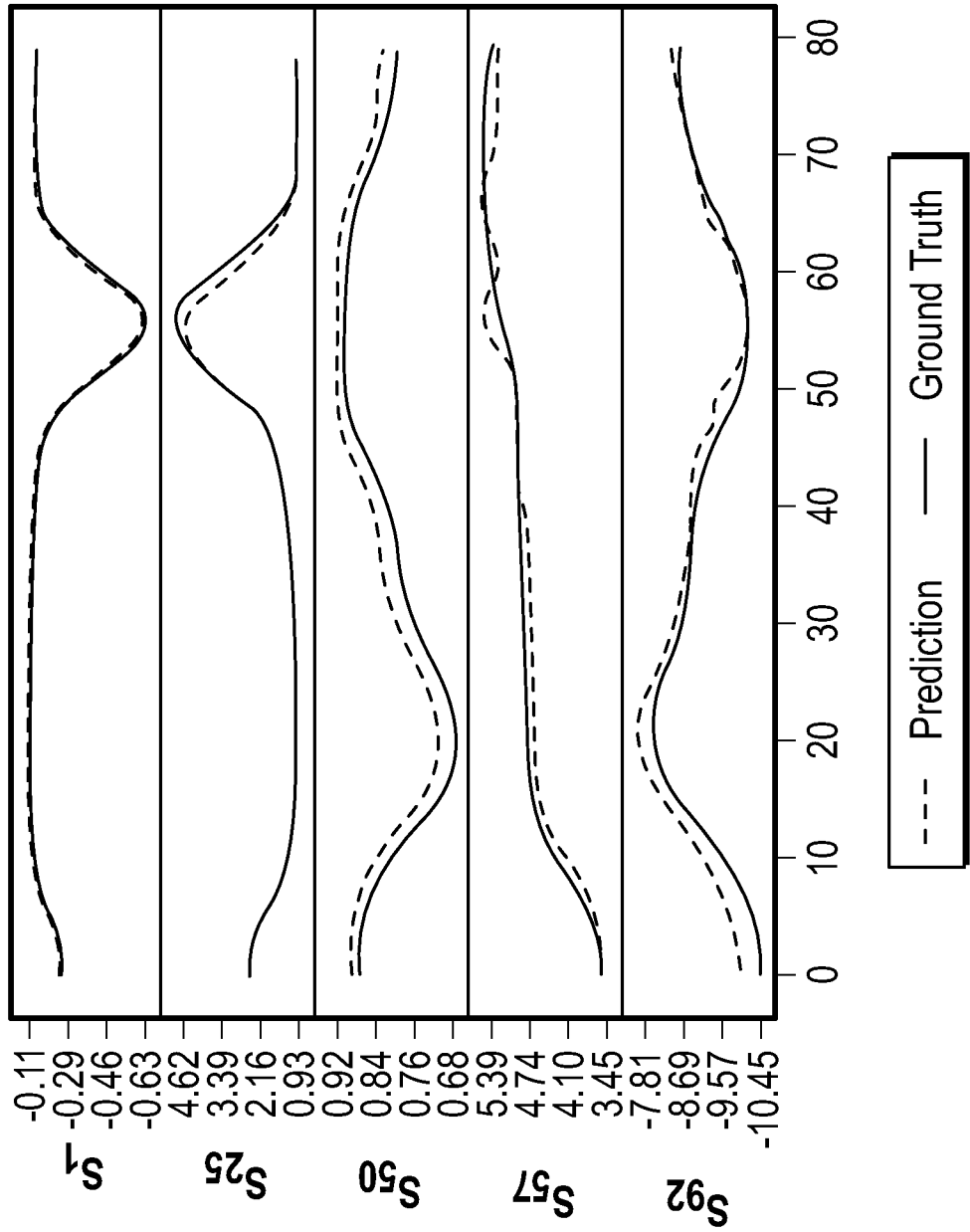


FIG. 12C

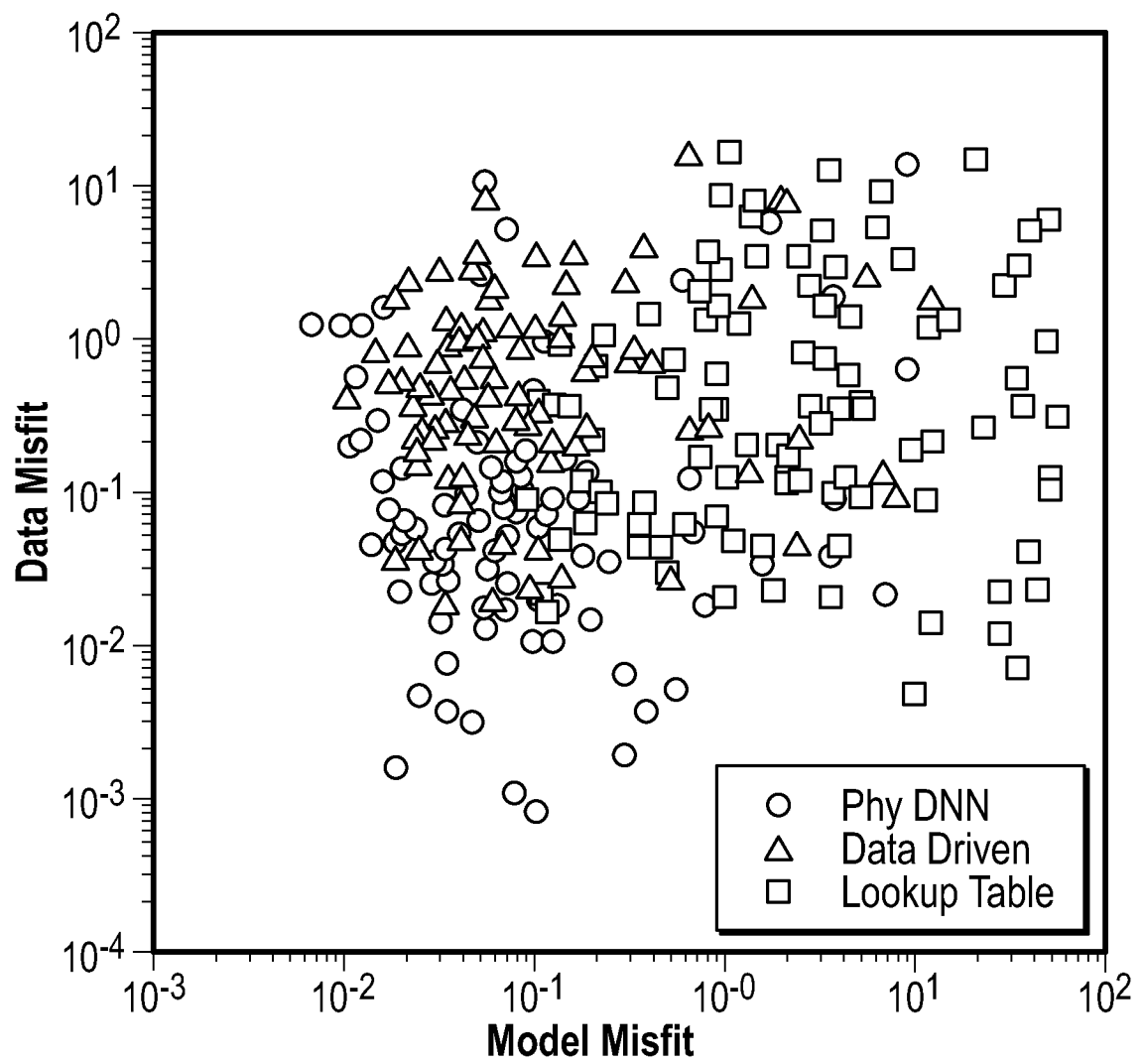


FIG. 13

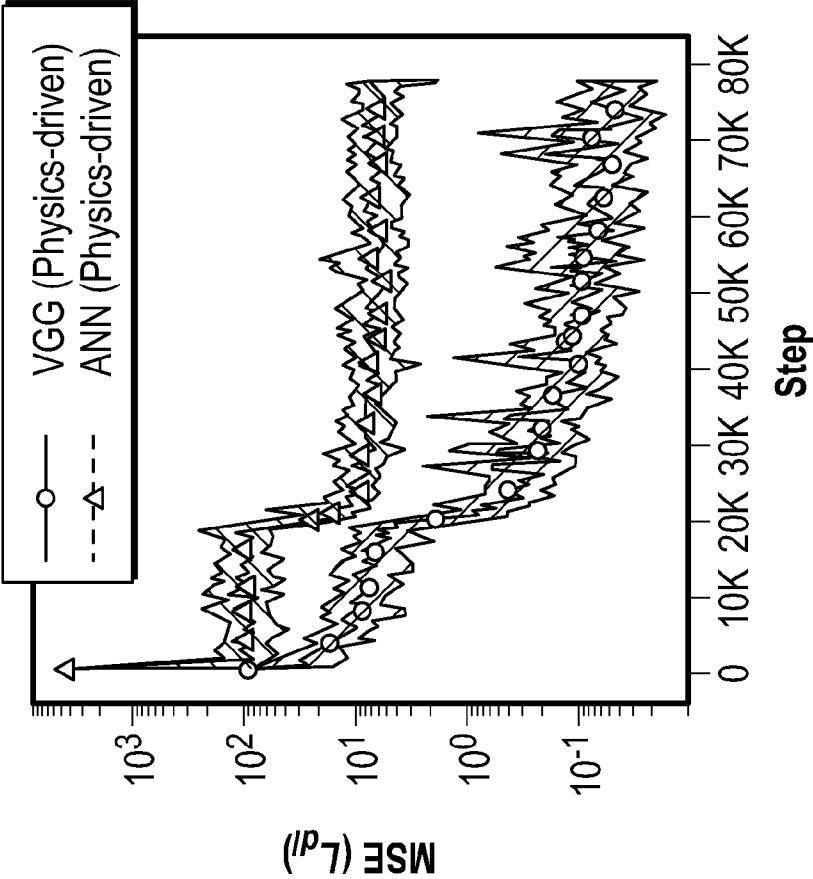


FIG. 14B

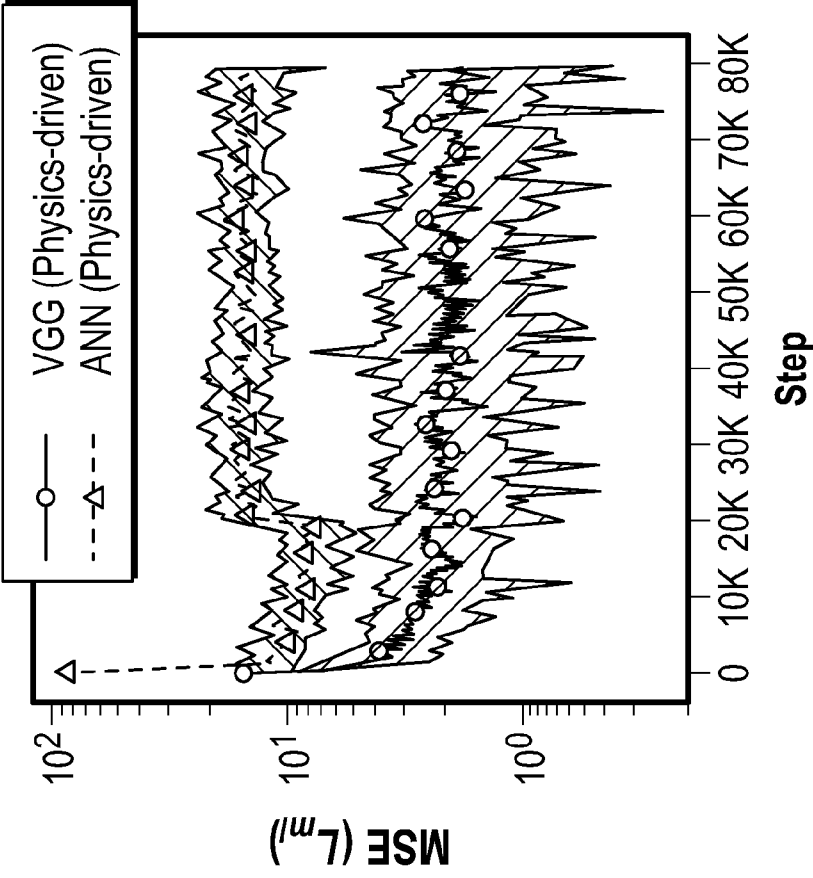


FIG. 14A

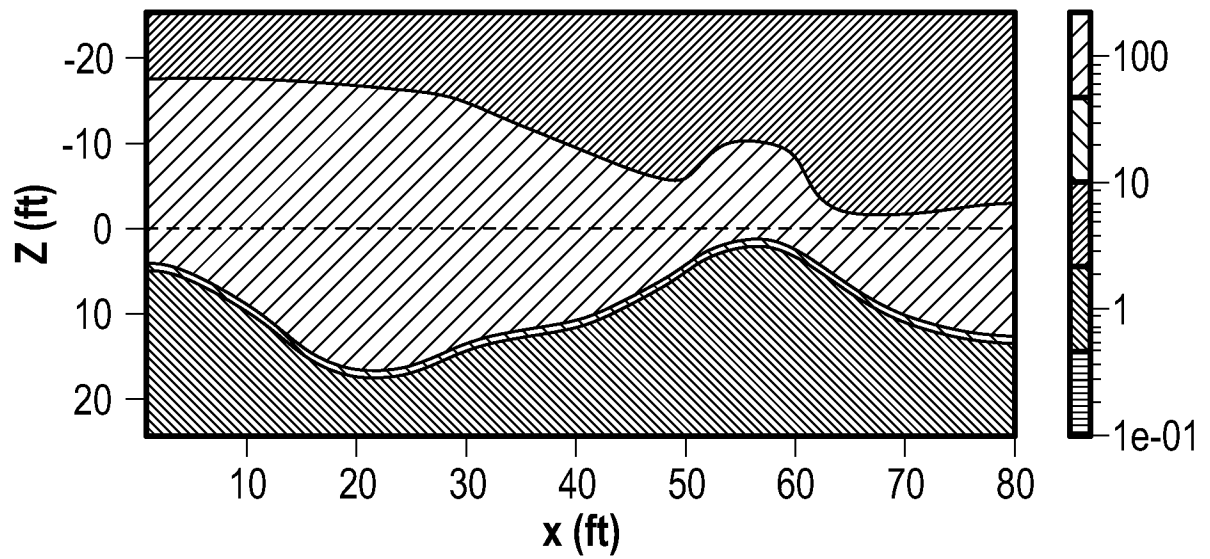


FIG. 15A

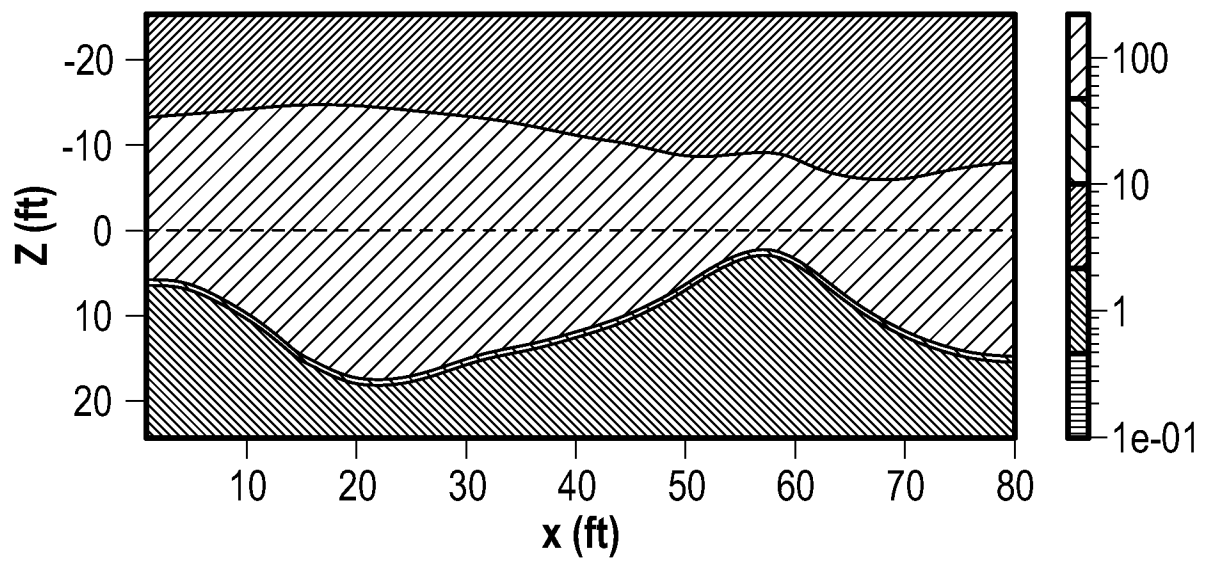


FIG. 15B

Table 1: The Numeric Evaluation (MSE) for the Model Misfit and Data Misfit

Method	Model Misfit ($\mathcal{L}_{ml}$)	Model Misfit ($\mathcal{L}_{dl}$)
DNN (with physics engine)	2.836e-2	1.427e-2
DNN (without physics engine)	6.970e-2	0.2298

FIG. 16

	On Surface	Downhole
Data Amount	Inadequate	Adequate
Computation	Fast	Slow
Memory	Large	Small

FIG. 17

PhyDNN	Lookup Table (10K)	Lookup Table (2M)	LMA Optimization
0.297s	1.805s	68.734s	169.754s

FIG. 18

PhyDNN	Lookup Table (10K)	Lookup Table (2M)	LMA Optimization
34.8MB	70.1MB	1.61GB	<1K

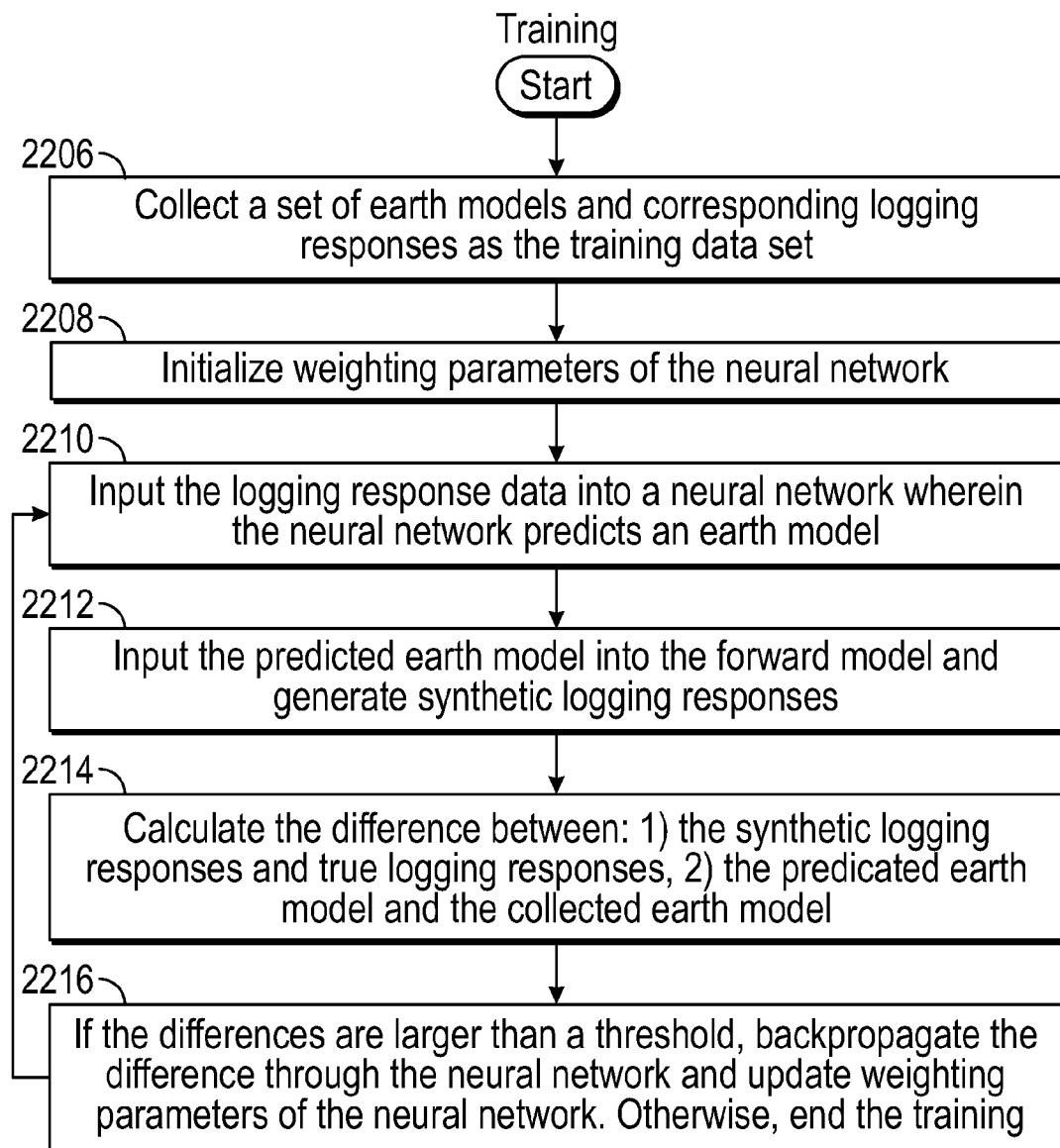
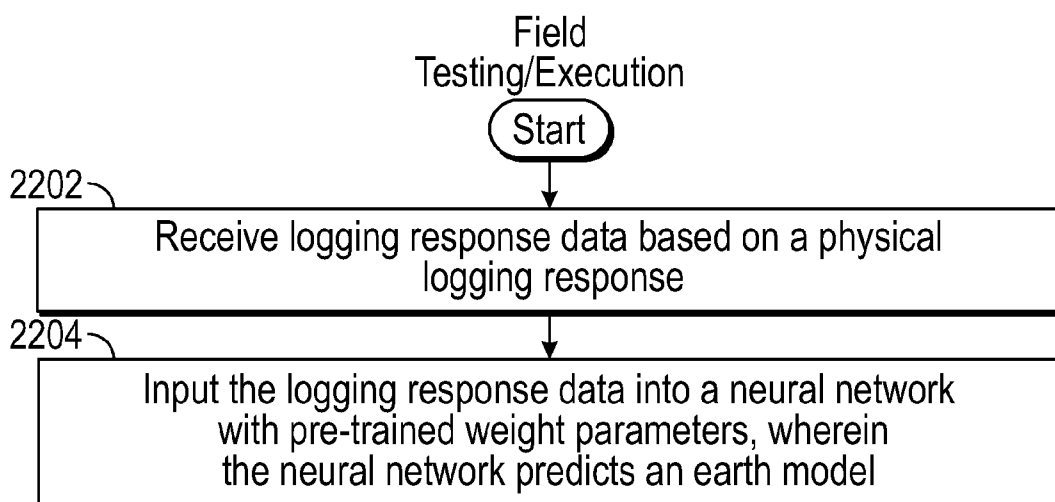
FIG. 19

	PhyDNN	Lookup Table (10K)	Lookup Table (2M)	LMA Optimization
$\mathcal{L}_{\text{ml}}$	0.0392	1.9686	1.4300	0.0517
$\mathcal{L}_{\text{dl}}$	0.0130	0.0439	0.0269	0.0013

FIG. 20

	ANN (Data-driven)	ANN (Physucs-driven)
$\mathcal{L}_{\text{ml}}$	1.1580	2.6692
$\mathcal{L}_{\text{dl}}$	1.1205	0.3155

FIG. 21

**FIG. 22A****FIG. 22B**

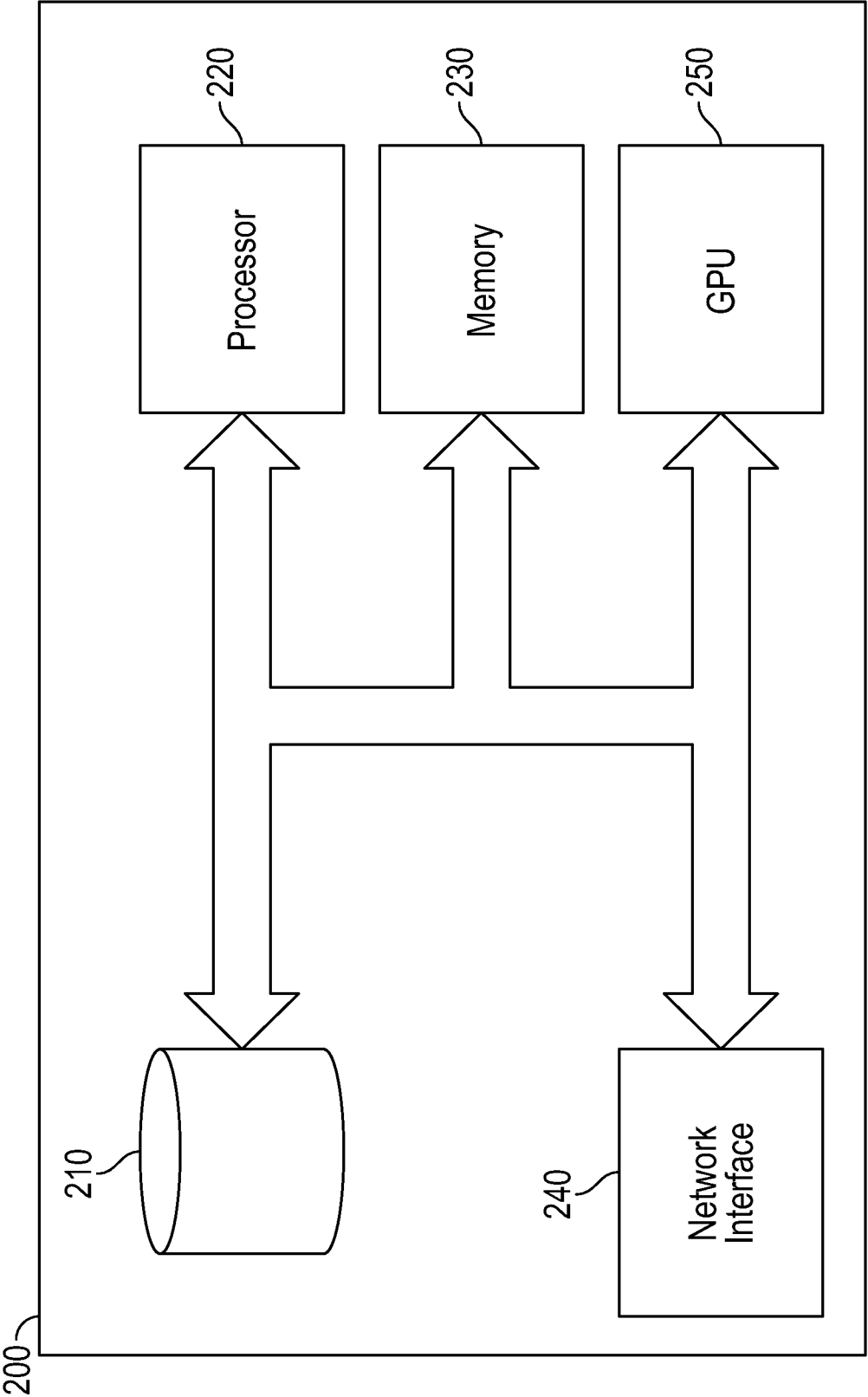


FIG. 23

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 20/38108

A. CLASSIFICATION OF SUBJECT MATTER

IPC - E21B 7/04 (2020.01)

CPC - E21B 47/022, E21B 7/04

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

See Search History document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	JIN et al. "A Physics-Driven Deep Learning Network for Subsurface Inversion". In 2019 United States National Committee of URSI National Radio Science Meeting (USNC-URSI NRSM) [online], pages 1-2, published January 9, 2019 [retrieved 2020.08.09]. Retrieved from the Internet <IIRI> https://ieeexplore.ieee.org/abstract/document/8712940 . entire document, especially Abstract; p 1, col 1; p 1, col 2; p 2, col 1; p 2, col 2; Fig 1; Fig 2	1-3, 8-13, 18-20 ----- 4-7, 14-17
Y	US 5,862,513 A (MEZZATESTA ET AL.) 19 January 1999 (19.01.1999), entire document, especially col 7, ln 32-33; col 7, ln 53-66	4-7, 14-17
A	US 2019/0169986 A1 (QUANTICO ENERGY SOLUTIONS, LLC) 06 June 2019 (06.06.2019), entire document	1-20



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"D" document cited by the applicant in the international application

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

10 August 2020 (10.08.2020)

Date of mailing of the international search report

03 SEP 2020

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450
Facsimile No. 571-273-8300

Authorized officer

Lee Young

Telephone No. PCT Helpdesk: 571-272-4300