Petroleum Science 18 (2021) 1465–1482

Contents lists available at ScienceDirect

Petroleum Science

journal homepage: www.keaipublishing.com/en/journals/petroleum-science

Original Paper

Iterative learning-based many-objective history matching using deep neural network with stacked autoencoder

Jaejun Kim [a], Changhyup Park [b, *], Seongin Ahn [c], Byeongcheol Kang [a], Hyungsik Jung [a], Ilsik Jang [d]

[a] Department of Energy Systems Engineering, Seoul National University, Seoul, 08826, Republic of Korea
[b] Department of Energy and Resources Engineering, Kangwon National University, Chuncheon, Kangwon, 24341, Republic of Korea
[c] Geo-ICT Convergence Research Team, Korea Institute of Geoscience and Mineral Resources, Daejeon, 34132, Republic of Korea
[d] Department of Energy and Resources Engineering, Chosun University, Gwangju, 61452, Republic of Korea

ARTICLE INFO

Article history:
Received 28 April 2020
Accepted 5 January 2021
Available online 17 August 2021

Edited by Yan-Hua Sun

Keywords:
Deep neural network
Stacked autoencoder
History matching
Iterative learning
Clustering
Many-objective

ABSTRACT

This paper presents an innovative data-integration that uses an iterative-learning method, a deep neural network (DNN) coupled with a stacked autoencoder (SAE) to solve issues encountered with many-objective history matching. The proposed method consists of a DNN-based inverse model with SAE-encoded static data and iterative updates of supervised-learning data are based on distance-based clustering schemes. DNN functions as an inverse model and results in encoded flattened data, while SAE, as a pre-trained neural network, successfully reduces dimensionality and reliably reconstructs geo-models. The iterative-learning method can improve the training data for DNN by showing the error reduction achieved with each iteration step. The proposed workflow shows the small mean absolute percentage error below 4% for all objective functions, while a typical multi-objective evolutionary algorithm fails to significantly reduce the initial population uncertainty. Iterative learning-based many-objective history matching estimates the trends in water cuts that are not reliably included in dynamic-data matching. This confirms the proposed workflow constructs more plausible geo-models. The workflow would be a reliable alternative to overcome the less-convergent Pareto-based multi-objective evolutionary algorithm in the presence of geological uncertainty and varying objective functions.

1. Introduction

Data analytics is relevant in reservoir simulation. Data deluge needs robust and reliable tools for interpreting different-scaled data that is related to a broad range of disciplines, such as geology, geophysics, petrophysics, and reservoir engineering. The degree of confidence and granularity assist in the performance evaluation of data-driven models to ensure field applicability. However, data processing with acceptable standard forms is challenging in history matching; the key questions are related to establishing right links among the spatiotemporal values in time and space. It must incorporate all available data, and account for field data observed at wells, which is evaluated by geological and geophysical studies. The challenging tasks of many-objective history matching, i.e., matching independently different histories, can be categorized as uncertainty quantification and dimension reduction (Deb et al., 2002; Guria et al., 2014; Min et al., 2016; Kim et al., 2017; Carneiro et al., 2018).

Geological uncertainty is inevitable due to the lack of accurate data in reservoir characterization (Hegstad and More, 2001; Carneiro et al., 2018; Koneshloo et al., 2018; Scheidt et al., 2018). The spatial uncertainty of reservoir properties creates difficulty in the reliable analysis of fluid transport. Production histories are related to the geological characteristics within drainage areas, while geological interpretations should explain the entire reservoir. The differences in decomposition concepts between the local area and entire area can result in erroneous evaluation. Geological uncertainty increases production performance outliers, and thereby has a negative effect on converging the global minima. The ill-posed nature of history matching hinders utilization of all the available objective functions that are weighted comparably.

The robustness demands to keep parameter dimensionality as

* Corresponding author.
E-mail address: changhyup@kangwon.ac.kr (C. Park).

https://doi.org/10.1016/j.petsci.2021.08.001

small as possible. As the number of objectives increases, divergence or scale-dependency problems occur, which are known as "the curse of dimensionality." Dimensional reductions, such as linear objective reduction, principal component analysis, and clustering, have been introduced to manage implications from "the curse of dimensionality (Scheidt and Caers, 2009; Min et al., 2016; Siena et al., 2016; Esmaeilzadeh et al., 2020; Lim et al., 2020)." However, choosing the most significant components may ignore the local characteristics of unselected objectives. Individual producers have their own production characteristics, but field-averaged data do not sufficiently explain the observations. A higher number of dynamic responses requires additional objective functions to be allocated to each observation, thereby causing the convergence to global minima to be impossible. Pareto-based multi-objective evolutionary algorithms have been applied in solving history matching problems, and Pareto-front solutions with non-dominated relationship have been explored (Guria et al., 2014; Hutahaean et al., 2017; Kim et al., 2017). A typical Pareto-based many-objective history matching limits the number of objective functions that do not comprise the divergence matter; however, it satisfies a limited number of objectives within an acceptable error.

Clustering schemes, such as *k*-mean clustering and *k*-medoids clustering, are additional dimension reductions. Having a predefined distance results in the creation of different distance maps and similarity matrices. It also helps to select reliable geo-models based on flow responses. Clustering relies on distance being defined with respect to flow responses, such as liquid production rates and pressure, and geological characteristics (Scheidt and Caers, 2009; Kim et al., 2020b; Lim et al., 2020). Lim et al. (2020) implemented *k*-medoids clustering separately for observed well-based responses and used their union as initial ensembles for data assimilation. Kim et al. (2020b) proposed geo-model selection based on *k*-medoids clustering with expansion of the searching domain. Despite that the different distances can be defined as the production rates observed at each producer, how to use the selected geo-models in many-objective history matching remains challenging. For example, the distance based on field production can represent field characteristics, but the non-overlap zone among drainage areas is uncertain, which can cause inaccurate forecasting of liquid production rates. However, if the distance is defined as the linear weighted sum of observed data, it follows single-objective history matching and thus, the requirements of many-objective history matching are no longer applicable.

Machine learning that uses big data typically includes training processes, e.g., supervised or unsupervised-learning. Supervised learning is a machine training method that uses specific output values, such as labels, for input data. In other words, it is a data learning process that predicts or classifies output values for new input data through training of a given data set. Unsupervised learning does not aim to classify or predict output data because it does not have specific output values for the input data, but rather learns data features by training the given input data itself. An autoencoder is a typical unsupervised learning algorithm, which is a neural network-based algorithm that is widely used for dimension reduction (Hinton and Salakhutdinov 2006; Vincent et al., 2010; Zhang et al., 2018), anomaly detection (Zhou and Paffenroth, 2017; Ribeiro et al., 2018), and image processing (Zeng et al., 2017; Han et al., 2020). Recently, in addition to its effectiveness in dimension reduction, it has been used as a pre-trained learning method. This method uses bottleneck neurons that learn input data characteristics through autoencoders as input data for a conventional supervised learning algorithm (Erhan et al., 2010; Weston et al., 2012; Ahn et al., 2018; Mo et al., 2019; Liu et al., 2019; Liu and Grana, 2020).

Recent advances in neural networks with normalized datasets provide alternative methods for controlling differently scaled available data (Hinton et al., 2006; Hinton and Salakhutdinov 2006; LeCun et al., 2015; Goodfellow et al., 2016; Ahn et al., 2018; Chan and Elsheikh, 2019; Liu et al., 2019; Mo et al., 2019; Ki et al., 2020; Kim et al. 2020a, 2020b). Neural networks, such as deep neural network (DNN) that has some hidden layers, can input various types or scales of data in a normalized form, and establish an empirical relationship between input and output layers. Modern neural-network research has concentrated on big-data analytics and data-integration by combining network design using deep learning, dimensional reduction, and feature extraction. By controlling neurons and hidden layers, users can design the optimum framework of neural networks to minimize computational costs and achieve accuracy. Several successful case studies have demonstrated the benefits of applying data analytics to transform the traditional reservoir model to data-driven decision support (Mehta, 2016; Cavalcante et al., 2019; Saputelli, 2019; Kim et al. 2020a, 2020b). Cavalcante et al. (2019) proposed a continuous learning algorithm in data assimilation and showed its algorithm could be applied to a history matching problem. Kim et al. (2020a) implemented DNN with the latent features and constructed the prior geo-models to mitigate the overshooting problem of ensemble smoother with multiple data assimilation. Kim et al. (2020b) confirmed the applicability of DNN-based inverse modeling in fluvial channel reservoirs where flow paths are not complex and mainly occur in sandstone channels. Applicability was limited because the production histories were four kinds and 2D geo-models that typical Pareto-based multi-objective evolutionary algorithms could be applicable successfully.

This research examines an iterative-learning scheme to produce more reliable and less error-prone geo-models as a supervised-learning dataset of DNN-based inverse models. By updating the supervised-training dataset iteratively, this study tries to develop a methodology to upgrade the empirical correlations made by DNN. The objective of this research is to develop a robust workflow that can help mitigate many-objective history matching issues. This workflow consists of DNN-based inverse modeling and an iterative-learning method that can update more reliable supervised-learning datasets and produce plausible and heterogeneous geo-models with matching production histories. This can be applied to remedy or prevent issues encountered in many-objective history matching. The applicability to many-objective history matching is examined by comparing typical Pareto-based multi-objective evolutionary algorithms, i.e., Non−dominated Sorting Genetic Algorithm−II (NSGA-II) (Deb et al., 2002; Guria et al., 2014; Min et al., 2016; Kim et al., 2017), in a 3D heterogeneous reservoir.

2. Many-objective history matching problem: PUNQ-S3 benchmarking field and geostatistical construction of initial geo-models

A 3D heterogeneous reservoir with many objective functions, PUNQ-S3 (Production forecasting with uncertainty quantification-S3) reservoir, has been widely used as a benchmark to examine the pros and cons of history matching schemes (Gu and Oliver, 2005; Hutahaean et al., 2017; Lee et al., 2017; Carneiro et al., 2018). PUNQ-S3 has a corner-point grid system with $(x, y, z) = (19, 28, 5)$, and five layers along the z direction. The number of permeable grids is 1761, i.e., 396 in the first and the second layers and 323 in the others. An impermeable fault surrounds the reservoir on the southeast part while a strong aquifer is placed at the northeast zone (Lee et al., 2017). Fig. 1 illustrates the permeability distribution transformed using natural logarithm and the locations of production wells. Table 1 summarizes the ranges of absolute permeability and porosity with z layers; the first, third, and fifth

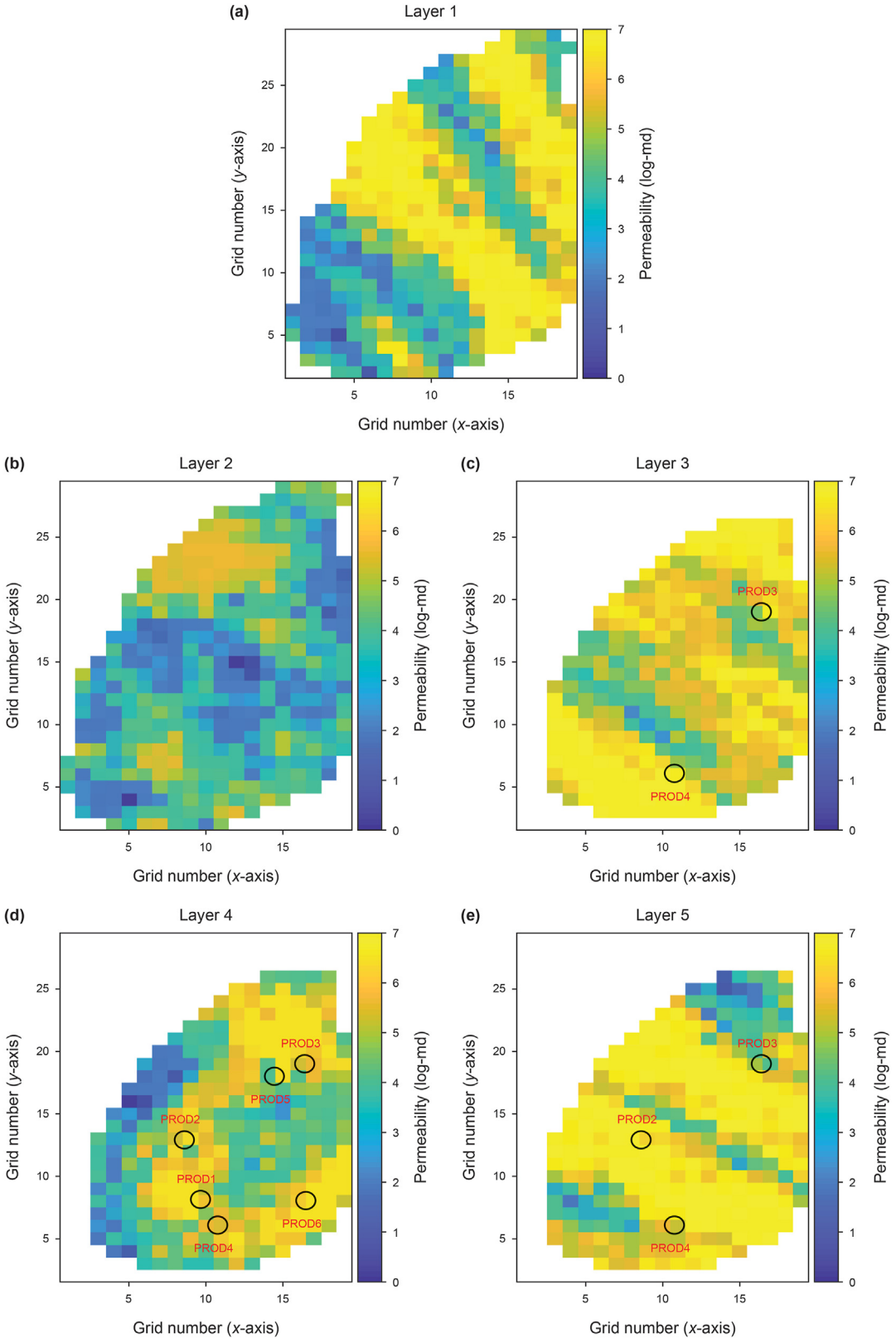

Fig. 1. Spatial distribution of permeability converted by natural logarithm in the PUNQ-S3 field: (**a**) layer 1; (**b**) layer 2; (**c**) layer 3; (**d**) layer 4; and (**e**) layer 5. Layers 1 and 2 do not contain production wells, while the other layers have various perforated zones with six producers.

Table 1
Porosity and permeability ranges at each layer in the PUNQ-S3 field.

Layers	1, 3, 5	2	4
Porosity (fraction)	0.01–0.30	0.01–0.17	0.01–0.22
Horizontal permeability (*xy*-plane), mD	0.1–1000	0.1–200	0.1–500
Vertical permeability (*z* direction), mD	0.1–500	0.1–50	0.1–100

layers have similar properties with high productivity, while the second layer has less productivity, and the fourth layer has the intermediate values. Six producers are perforated at the fourth, the fifth, and third layers (Fig. 1). The observed data include oil production rates, well bottom-hole pressure (well flowing pressure observed at the well bottom), gas-oil ratios (surface volume ratio of gas to oil production rates), and water cut (the volume ratio of water production to liquid production), which accounts for the availability of 24 different categories of dynamic data. Tables 2 and 3 describe well operations including shut-in processes, and show that the reservoir experiences irregular shut-in operations, which results in nonlinear production performances. Table 2 summarizes well operations during the history matching period (until 2936 days) and Table 3 shows them during the prediction period of 2937–6025 days.

The many-objective history matching problem that needs solving is to match 18 different production histories, except for water cut, until 2936 days. The water cut is implemented as a blind test to evaluate the reliability of the proposed workflow since it is not included in history matching. Most values of water cut during the history matching period (0–2936 days; 20 time steps; see Table 2) are zero, and therefore the water cut trends represent whether it is feasible for the workflow to construct the plausible geo-models. The 18 types of objective functions are defined in Eqs. (1) and (2):

$$\mathrm{argmin}F(y) = \left\{f_{1,1}, \cdots, f_{i,j}, \cdots, f_{o,W}\right\} \tag{1}$$

$$f_{i,j}(y) = \frac{1}{t_n}\sum_{k=1}^{t_n}\left|\frac{y_{i,j}^{\mathrm{true}} - \widehat{y}_{i,j}^{\mathrm{sim}}}{y_{i,j}^{\mathrm{true}}}\right| \tag{2}$$

where $F(y)$ is the set of objective functions ($f_{i,j}$); subscripts i, j, o, and W represent the production response ($i = 3$; oil rates, bottom-hole pressure, and gas–oil ratio), the name of production well ($j = 6$; PROD1, PROD2, PROD3, PROD4, PROD5, and PROD6), the total number of observed response used in history matching problem ($o = 3$), and the total number of production well ($W = 6$), respectively; the superscript 'true' means the ith observed response (y) at jth production well in the target field; the superscript 'sim' denotes that ($\widehat{y}$) in the geo-models resulted from inverse modeling; and t_n is the number of time steps until the end of the history matching period ($t_n = 20$ as shown in Table 2). The many-objective history matching minimizes $F(y)$ by calibrating absolute permeability assigned in each grid.

Well operations influence the production performances and thereby further complicate production performances forecasting without changing the operation conditions. The initial geo-models, made using Stanford University's SGeMS software (Remy et al., 2009), are implemented to train the DNN and the SAE at the first stage. The size of the initial population is 100. ECL100

Table 2
Time steps of the PUNQ-S3 field during the history matching period.

Time step	1	2	3	4	5	6	7	8	9	10
Time, days	1.01	91	182	274	366*	1461*	1642	1826	1840*	1841
Time step	11	12	13	14	15	16	17	18	19	20
Time, days	2008	2192	2206*	2373	2557	2571*	2572	2738	2922	2936*

Notes: * shut-in wells.

Table 3
Time steps of the PUNQ-S3 field during the prediction period.

Time step	21	22	23	24	25	26	27	28	29	30
Time, days	2937	3103	3287	3288*	3301*	3302	3469	3653	3654*	3667*
Time step	31	32	33	34	35	36	37	38	39	40
Time, days	3668	3834	4018	4019*	4032*	4033	4199	4383	4384*	4397*
Time step	41	42	43	44	45	46	47	48	49	50
Time, days	4398	4564	4748	4749*	4762*	4763	4930	5114	5115*	5128*
Time step	51	52	53	54	55	56	57	58	59	60
Time, days	5129	5295	5479	5480*	5493*	5494	5660	5844	5845*	5858*
Time step	61	62								
Time, days	5859	6025								

Notes: * shut-in wells. Time step and time are continued in Table 2.

(Schlumberger, 2018), a commercial black oil simulator, outputs production performance results. Fig. 2 depicts the production histories observed at the producers including oil rates (Fig. 2(a)), bottom-hole pressure (Fig. 2(b)), gas-oil ratio (Fig. 2(c)), and water cut (Fig. 2(d)), respectively. Fig. 2 presents production well histories making it conducive to the conflicted-objectives problems.

3. Materials and methods

Fig. 3 depicts the workflow of the iterative learning-based many-objective history matching, and is divided into two parts: (1) DNN-based inverse modeling with SAE encoding/decoding process (DNN-SAE); and (2) the iterative-learning process with training dataset updating of DNN for the next stage.

SAE is established using the initial models as the pre-trained process so that the training set updates are applied to upgrade the DNN structure. The DNN-SAE process results in encoded static data and the geo-models are constructed again through decoding. SAE is implemented not only to encode the static data as the output neurons of DNN, but also to decode the output values estimated by DNN to grid properties for reservoir simulations. The iterative-learning process is to update the supervised-training dataset for next stage; it applies distance-based clustering schemes. The update continues until the given stage (20 stages in this work) and stops early when the most of the error values tend to converge to the minimum.

3.1. DNN-SAE: DNN-based inverse modeling with SAE encoding/decoding process

DNN-based inverse modeling places the observed histories at the input layer and the static properties at the output layer. DNN-SAE framework replaces the original static data with SAE-encoded data; the encoded values of permeability. Fig. 4 divides DNN-SAE into SAE encoding (Fig. 4(a)), a DNN-based inverse model (Fig. 4(b)), and SAE decoding (Fig. 4(c)), respectively. To train DNN, the input layer has the observed production data (Y) that are normalized, and the output neurons are encoded data (Z) of the static property (grid permeability) in training dataset (100 initial geo-models at the first stage). The trained DNN assigns the input neurons as the true responses until history matching (2936 days) is completed and produces the output neurons ($\widehat{Z}$) that are the same format of encoded values (Fig. 4(b)). The output results ($\widehat{Z}$) are decoded, converted to permeability values ($\widehat{X}$), and the geo-model is constructed by history matching (Fig. 4(c)).

Training neural networks (DNN and SAE) is a process that adjusts weights to find a nonlinear function to minimize the prediction error between actual values and neural network predictions. A learning algorithm, which is typically used in neural networks, is a gradient descent method that calculates the actual slope using a backpropagation method. The slope descent method obtains the slope by differentiating the function, and then finds the optimal solution by incrementally changing the factor the direction of the slope (Reed and Marks, 1999). The sum of the weights in the neuron is calculated through the operator, and the calculated values are finally implemented using the activation function. This study implements a continuous log-sigmoid function as an activation function of SAE and a tan-sigmoid function for DNN. The optimizer for SAE, the training (learning) algorithm, is a scaled conjugate gradient backpropagation (*trainscg* in MATLAB; MathWorks, 2018), while for the optimizer for DNN is a gradient descent with momentum and adaptive learning rate backpropagation (*traingdx* in MATLAB; MathWorks, 2018). In short, DNN-SAE estimates the encoded values honor to the observed data, and the pre-trained SAE reconstructs the geo-model. For training DNN-SAE, the input uses the simulated responses of the geo-model, while the output is set with encoded values of permeability in the model.

3.1.1. Training SAE encoding/decoding system: geo-model reconstruction

SAE can stack multiple hidden layers for learning complex data. The purpose of training SAE is to reconstruct the input data as it is, so that the neural network structure has the same number of neurons in the input and the output layers (Fig. 5). The dimension of the input data can be modified by the number of neurons in the hidden layer, the bottleneck, which is located in the middle of the SAE structure. If the number of neurons in the bottleneck is set smaller than the number of input data, the input data can be compressed (dimensionally reduced). In SAE training, learning from the input layer to the bottleneck is called the encoder, while the learning stage of reconstructing the compressed data in the bottleneck to the input data is called the decoder. Training the SAE framework aims to minimize the mean squared error (MSE), which is a function of loss or cost (Eq. (3)):

$$L\left(\boldsymbol{X},\widehat{\boldsymbol{X}}\right)=\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\widehat{x}_i\right)^2 \tag{3}$$

where $L(\boldsymbol{X},\widehat{\boldsymbol{X}})$ is the MSE; n is the number of input data; x_i is the ith input data; and $\widehat{x}_i$ is the ith reconstructed data. The dimensions of input and output layers (the superscript D of $\mathbb{R}^D$ in Fig. 5) are the same as the calibrating permeabilities in the geo−model with a grid number of 2660 (the total grid number is $19\times28\times5=2660$). The calibrating positions for history matching include all grids in PUNQ-S3. The bottleneck dimension, or encoded data (the superscript E of $\mathbb{R}^E$ in Fig. 5), is set as 100. The supervised-training dataset of SAE consists of 100 geo-models generated as the initial population except for the true model; 80% of which is for the training set and the remaining 20% for the validation set.

3.1.2. Training the DNN-based inverse model

The DNN-based inverse model in this study is a multiple-hidden-layered neural network consisting of two hidden layers, one input, and one output layer. The DNN training follows the training process of SAE; however, input and output neurons are different with SAE's. The input neurons are the production histories, while the output is the encoded values related to permeability in the geo-models. Because three kinds of observed histories are used as the input neurons, the values should be normalized in the range of −1 and 1 (Eq. (4)):

$$I_{i,j}(t)=2\times\frac{\psi_{i,j}(t)-\psi_{\min}(t)}{\psi_{\max}(t)-\psi_{\min}(t)}-1, \tag{4}$$

where $I_{i,j}(t)$ is the observed histories as input neurons; for inverse modeling, $\psi_{i,j}(t)=y_{i,j}^{\text{true}}$ is the response observed at the PUNQ-S3 field that is the ith observed response from jth production well at time t; and for training DNN, $\psi_{i,j}(t)=\widehat{y}_{i,j}^{\text{sim}}$ is the ith response from jth production well at time t, resulted from the flow simulation of the geo−model. The subscripts 'min' and 'max' represent the minimum value and the maximum value of $\psi_{i,j}(t)$ at the given time, t. The total number of input neurons ($I_{i,j}(t)$) is 360; the multiplication of the number of objective functions and that of time steps ($18\times20=360$; refer to Table 2), while that of output neuron sets as the data encoded by SAE are 100, instead of the full set of grid permeability (2660 data).

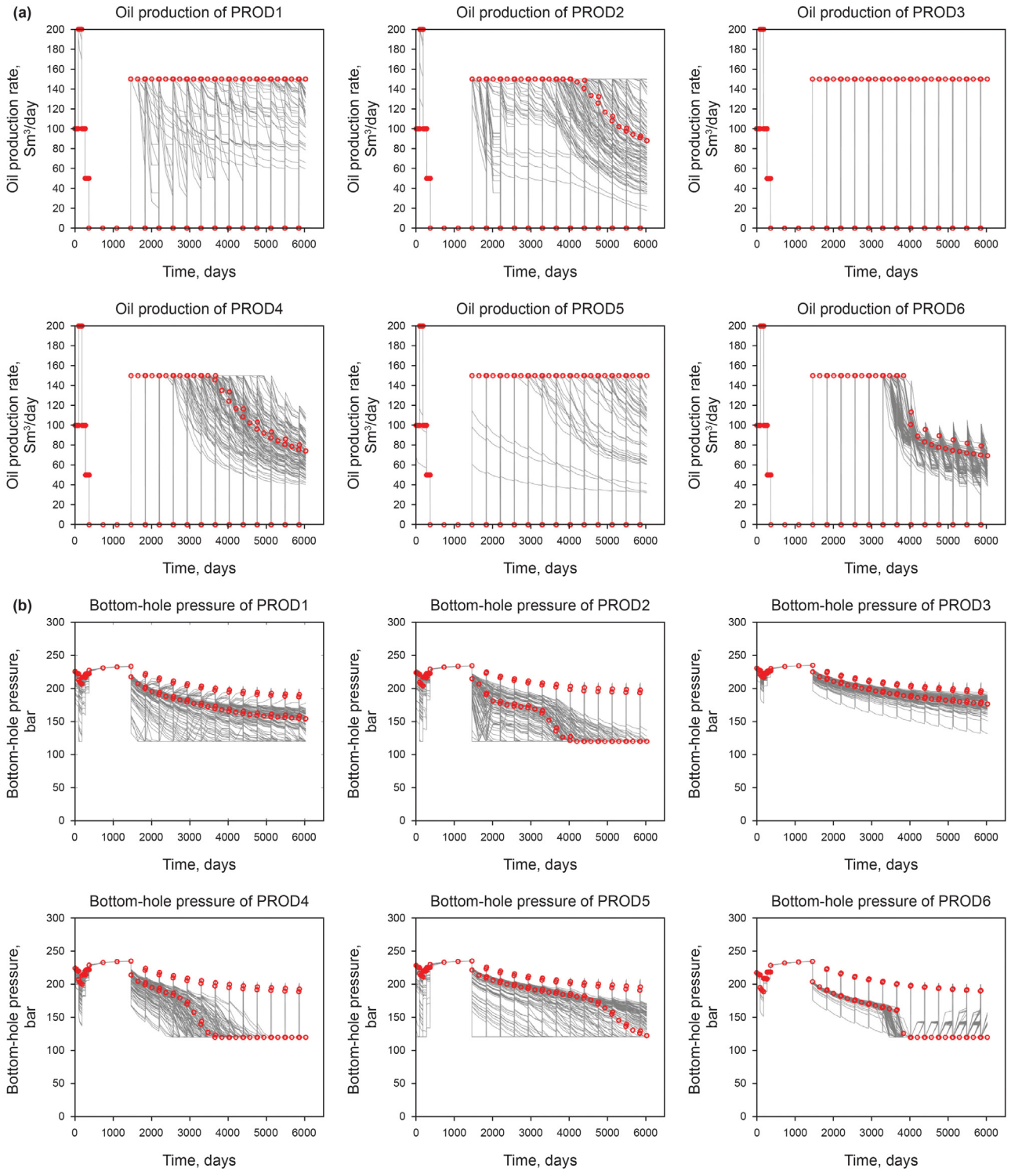

Fig. 2. Production histories of the PUNQ–S3 field and the initial geo-models: (**a**) oil production rate, (**b**) bottom-hole pressure, (**c**) gas-oil ratio, and (**d**) water cut.

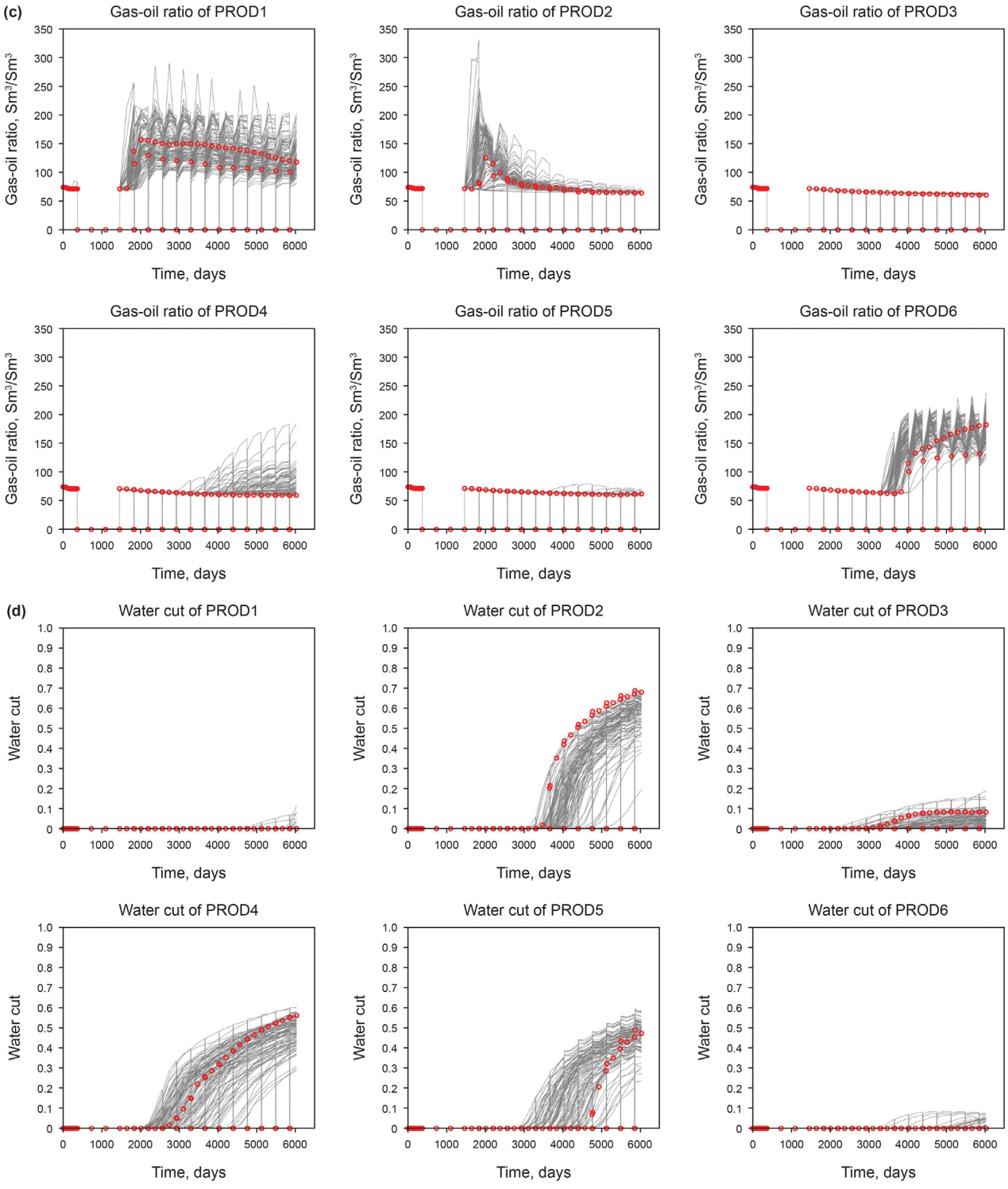

Fig. 2. (*continued*).

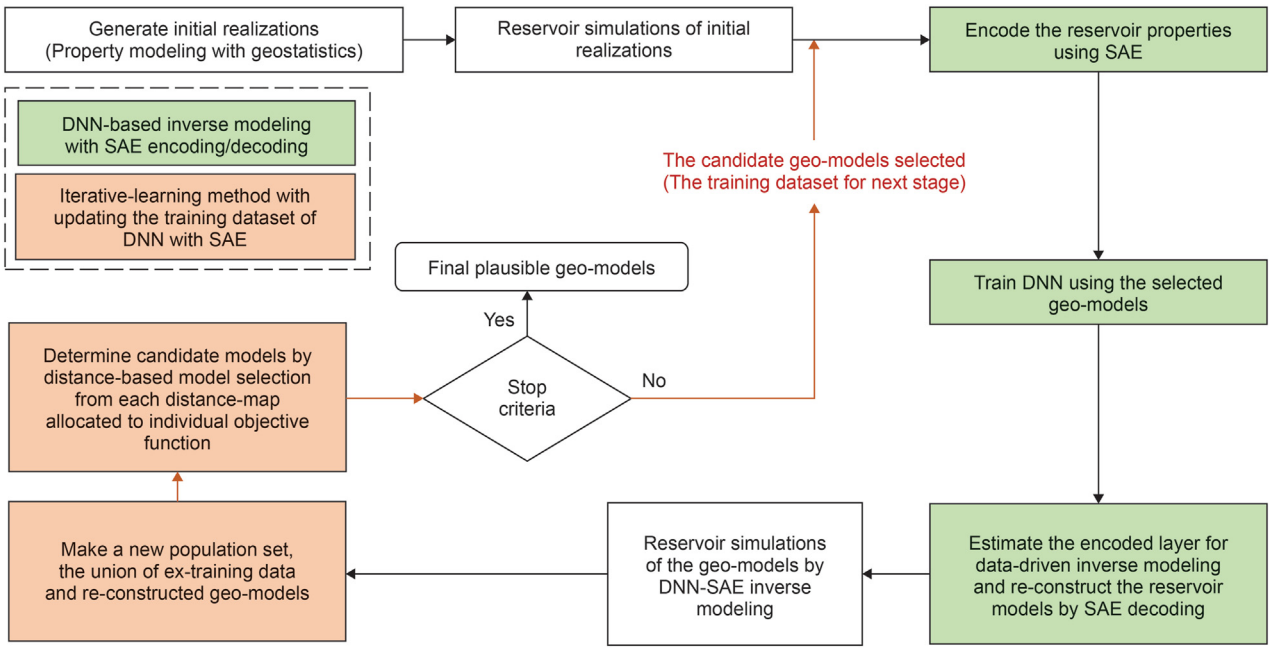

Fig. 3. Workflow of the iterative learning-based many-objective history matching.

3.2. Iterative-learning method with supervised-training dataset updates

The iterative-learning method iteratively updates DNN. A key component of the process is to determine the DNN supervised-training dataset at the next stage. The first consideration applies a method to include the reliable geo-models that can represent or show the production performances similar to those of true field conditions (PUNQ-S3 field). The geo-models selected should contain geological characteristics that can show all production performances similar to the observed histories (refer to Section 3.2.1). The other consideration is how to maintain the adequate population size or sufficient training size to train the DNN system. If the dataset size is small, the training performances would be poor. However, as computing time increases, the convergence activity might be poor, and could cause the DNN-SAE to experience a divergence problem in which it would not converge to the optimal solutions. This study intends to maintain the dataset size for training DNN similar to the initial population of 100 geo-models (refer to Section 3.2.2).

3.2.1. Geo-model selection using a distance-based map

A key idea of an iterative-learning method for many-objective history matching is to draw a distance–map in compliance with the individual production performance given at each producer (Lim et al., 2020; Kim et al., 2020b). Fig. 6 demonstrates a way to select geo–models in distance-maps. Fig. 6(a) is the result of typical *k*-medoid clustering and Fig. 6(b) shows the addition of geo-models

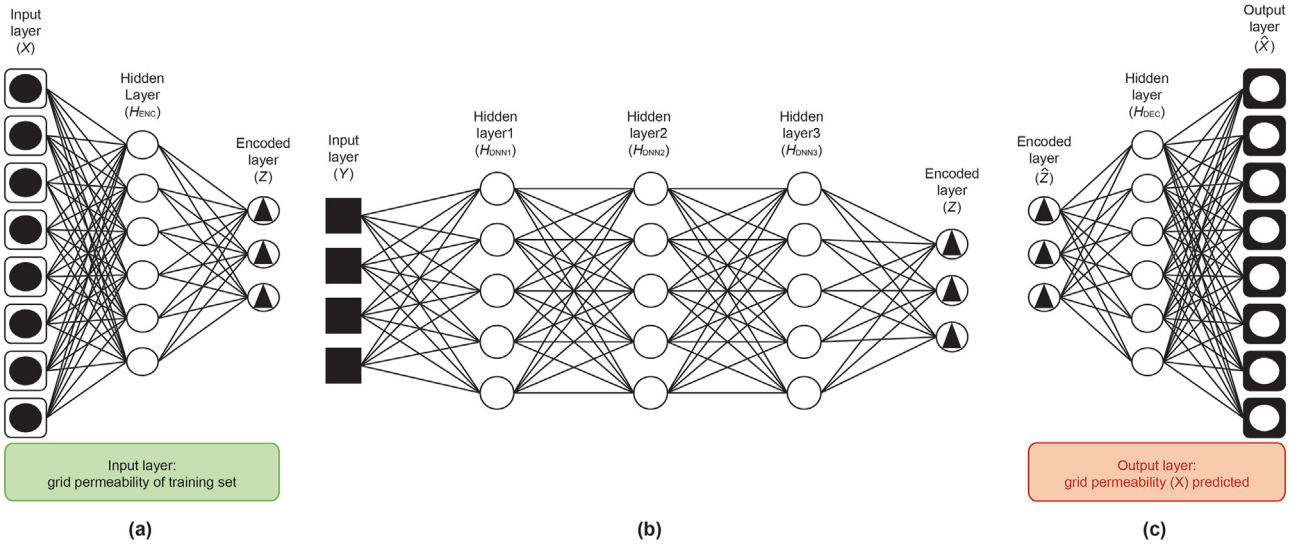

Fig. 4. DNN-SAE workflow: (**a**) SAE encoding process that encodes grid permeability (X) of individual geo-models; (**b**) DNN-based inverse model consisting of normalized production-related dynamic data (Y) as input neurons and the encoded data (Z) by SAE; and (**c**) SAE decoding process to reconstruct the grid permeabilities from the output values ($\hat{Z}$) estimated by DNN.

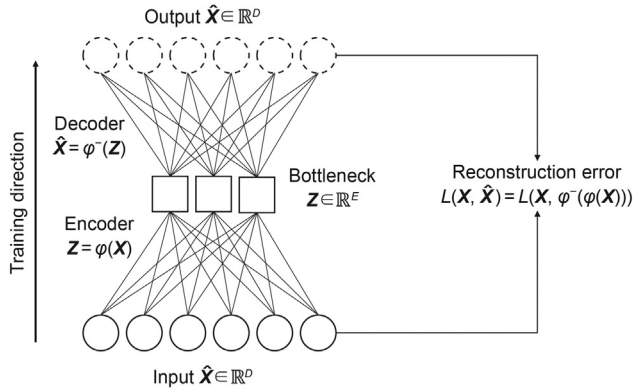

Fig. 5. Schematic diagram of SAE. X and Z denote neurons in the input layer and bottleneck neurons in the autoencoder, respectively. $\hat{X}$ represents the reconstructed output neurons. The superscripts D and E denote the dimensionality numbers of each data space. φ is the activation function for encoder and φ^- is for decoder in the fully connected layers.

nearer to the true response. Because we have three types of responses and six producers, the number of the distance maps is 18. The distance ($d_{i,j}^{\alpha,\beta}$) between the αth and the βth response at the ith observed response from jth production well is determined using (Eq. (5)):

$$d_{i,j}^{\alpha,\beta} = \sqrt{\left(I_{i,j}^{\alpha} - I_{i,j}^{\beta}\right)\left(I_{i,j}^{\alpha} - I_{i,j}^{\beta}\right)^T}, \quad \begin{matrix} \forall \alpha = 1, 2, \cdots, n,\ n+1 \\ \forall \beta = 1, 2, \cdots, n,\ n+1 \end{matrix} \tag{5}$$

where $I_{i,j}^{\alpha}$ is the normalized input value related to the ith observed response from the jth production well of the αth geo–model. There are ($n+1$) points, the number of geo–models (n) plus one true response, that are placed in individual distance map. This equation assumes that we do not know the true reservoir, but know the production response in PUNQ–S3. The user knows the geo–models (property distribution) that were generated, as well as the simulated production data. The geo–models selected in each distance map are defined as $S_{i,j}$ in Eq. (6):

$$S_{i,j} = C_{i,j} \cup E_{i,j} \tag{6}$$

where $S_{i,j}$ is the set of geo–models selected at the ith observed response from jth production well; $C_{i,j}$ is the set of geo–models within the cluster containing the ith observed response from jth production well (Fig. 6(a)); and $E_{i,j}$ is the geo–model within the

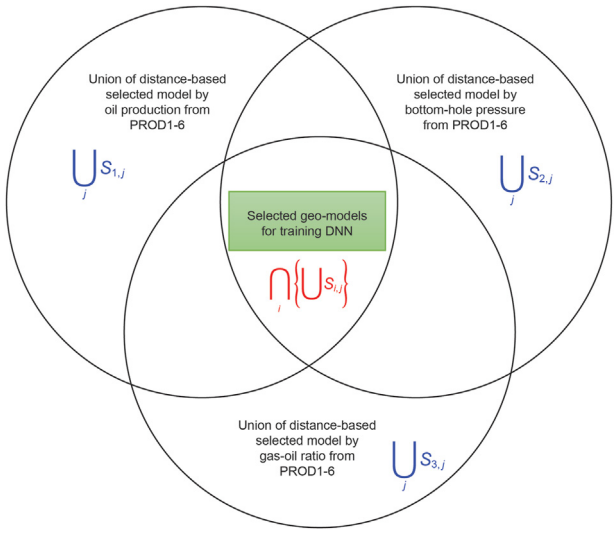

Fig. 7. Schematic diagram to determine the training set of DNN stages. The SAE encoding/decoding system was constructed using the initial population.

circle of which the radius is the distance between the true response and the medoid of selected cluster (Fig. 6(b)). This model–selection is repeated in the population (P_k) at the kth stage, and each distance map has the same number of points ($n+1$) but that of the geo-models selected ($S_{i,j}$) in each distance map would be different. A reason that this study does not assign the true response as one medoid is not only to conserve the random data selection, but to also examine whether the suggested methodology (the iterative update of geo-models) can choose the more reliable dataset as the iteration steps.

3.2.2. Iterative-learning of DNN with generating new training datasets

The first training set, (T_1), of DNN is the same as the initial population generated by the geostatistical method, (the 100 geo-model), which preserves uncertain geological scenarios to the greatest extent possible. Inverse modeling with DNN-SAE trained by T_k results in plausible geo-models, which is the set of H_k at the kth stage. DNN-SAE can reconstruct the geo-models as many times as the user wants, and certain important features that can improve the convergence to the optimal front can be indirectly transferred to the next stage. U_k is the union set of T_k and H_k at the kth stage before determining $S_{i,j}^k$ (Eq. (8) and (9)). The term $S_{i,j}^k$ for the ith

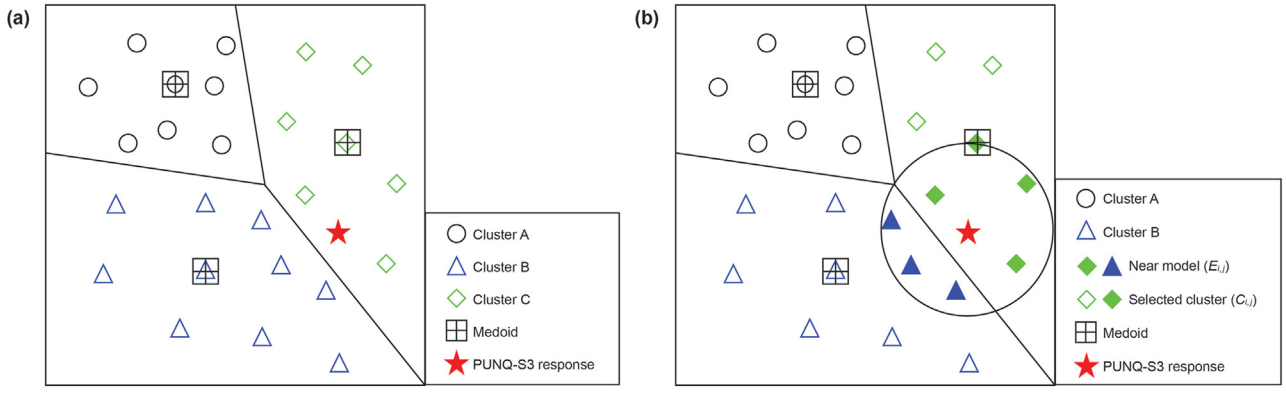

Fig. 6. Schematics to determine the set of geo–models, ($S_{i,j}$), on the distance map of the ith observed response from the jth production well (**a**) k-medoid clustering to determine $C_{i,j}$ (after k-medoids clustering) and (**b**) to include the geo–models near the true response ($E_{i,j}$) (after completing the geo–model selection) (modified from Kim et al., 2020b).

observed response from the jth production well at kth stage is determined; 18 types of $S^k_{i,j}$ sets are obtained. The training set of DNN for the $(k+1)$th stage (T_{k+1}) is determined at the intersection of union of all $S^k_{i,j}$ results (Eq. (9); Fig. 7). The purpose of this process is to prevent the divergence of training performances and to preserve the effective traits that are similar to the true responses. However, as the step progresses, the number of T_k values is at risk of decreasing and thereby causes the training ability of DNN to decrease. To maintain sufficient T_{k+1} values for improving DNN, the number of models in H_k is set to 500.

$$U_k = H_k \cup T_k, \ \forall k = 1,\ 2,\ 3,\ \cdots \tag{8}$$

$$T_{k+1} = \bigcap_{i,j}\left\{\cup S^k_{i,j}\right\}, \quad \begin{matrix}\forall k = 1,\ 2,\ 3, \cdots \\ \forall i = 1,\ 2,\ 3 \\ \forall j = 1,\ 2,\ \cdots,\ 6\end{matrix} \tag{9}$$

4. Results and discussion

4.1. Design of DNN-SAE system

The SAE encoding/decoding extracts some important features of permeability (encoding) and reconstructs the permeabilities (decoding the output neurons of DNN). SAE system does not use iterative-learning but only permeability values. Therefore, SAE is a pre−training process based on static datasets from geo-model encoding and decoding workflow. It uses the permeability tensors of the initial population set (100 geo-models; $P_0 = T_0$). SAE encoder/decoder had three hidden layers (H_{ENC}, Z, and H_{DEC} in Fig. 4(a) and (c)). The width of bottleneck (Z) was fixed as 100 neurons, and the remaining hidden layers (H_{ENC} and H_{DEC}) had the same number of neurons.

As the number of encoding/decoding hidden neurons ($H_{ENC} = H_{DEC}$) changed, the mean absolute percentage error (MAPE in Eq. (9)) and its arithmetic mean as the reconstruction error of SAE (ε_R in Eq. (10)) was introduced:

$$\varepsilon\left(\boldsymbol{X}, \widehat{\boldsymbol{X}}\right) = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{x_i - \widehat{x}_i}{x_i}\right| \tag{9}$$

$$\varepsilon_R = \mathbb{E}_{P_0}\left\{\varepsilon\left(\boldsymbol{X}, \widehat{\boldsymbol{X}}\right)\right\} \tag{10}$$

where $\varepsilon(\boldsymbol{X}, \widehat{\boldsymbol{X}})$ is the MAPE; n is the number of permeabilities as same as the total grid number; x_i represents the permeability at the ith grid that assumes the value before the encoding process; and $\widehat{x}_i$ is the permeability at the ith grid reproduced from the encoded bottleneck layer (Z). Each geo−model has its own MAPE and thus the reconstruction error is the averaged MAPEs (Eq. (10)). The case study was performed by changing the number of neurons of the first hidden layer; 200, 300, 500, and 1000. Table 4 summarizes the reconstruction errors and computing time, and Fig. 8 depicts their trends in relation to the number of the initial hidden neurons in the SAE system. As the number of neurons was increased, both computation time and reconstruction error increased. The SAE capacity was obtained at 200 as the width of the hidden layer that is sufficient for ensuring computational efficiency and training performance. The reconstruction error (ε_R) in the case of 200 neurons was smallest up to 1.6% and the SAE training required 374 s to be completed.

This study assumed a fixed DNN structure (Fig. 4(b)) because it was expected that the iterative-learning process can improve DNN; three hidden layers with 500 units. Table 5 summarizes the DNN-SAE structure. The DNN consisted of normalized dynamic histories (PUNQ-S3 has 360 production data within the matching period) minus three hidden layers (500-500-500) and minus the output layer, $\widehat{Z}$, (100). The SAE structure had three hidden layers, which consisted of 200 neurons for the encoding layer, 100 for the bottleneck layer, and 200 for the decoding layer, respectively.

4.2. DNN-SAE with iterative-learning for many-objective history matching

The reliability of DNN-SAE with iterative-learning process was examined by comparing with the typical Pareto-optimal algorithm (NSGA-II). The comparison method consisted of non-dominated sorting and crowding-distance sorting for diversity preservation in such a way that it generated Pareto-optimal solutions with stages. The stop criterion of NSGA-II was 20 stages, and was the

Table 4
Summary of reconstruction errors and computing time for encoding/decoding processes depending on the number of hidden neurons.

Number of hidden neurons ($H_{ENC} = H_{DEC}$) of SAE	Reconstruction error (MAPE), %	Computing time, s
200	1.60	374
300	2.10	538.2
500	3.70	861.2
1000	4.00	1770.2

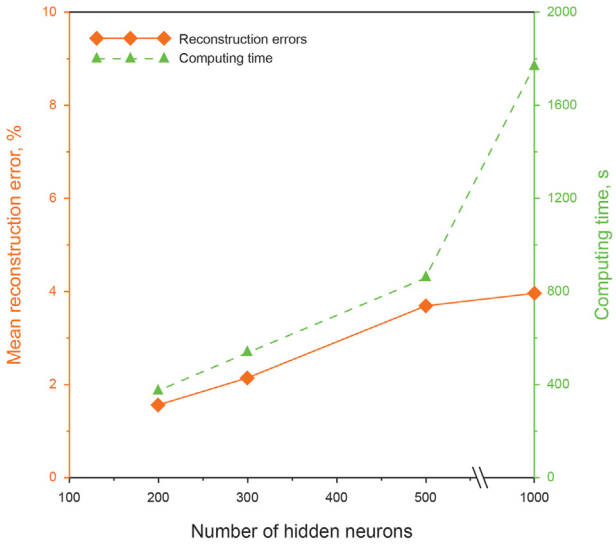

Fig. 8. Trends of the reconstruction error and computing time depending on the number of hidden neurons for SAE encoding/decoding processes.

Table 5
Design of DNN-SAE structure.

	Type of layer	Number of neurons
SAE	Input layer X	2660
	Output layer $\widehat{X}$	
	Hidden layer H_{ENC}	200
	Encoded layer Z	100
DNN	Input layer Y	360
	Hidden layer 1 H_{DNN1}	500
	Hidden layer 2 H_{DNN2}	
	Hidden layer 3 H_{DNN3}	
	Output layer $\widehat{Z}$	100

same as the maximum number of stages of the proposed workflow. Three tests were performed; dynamic-data matching ability, prediction accuracy of unknown production profiles, and the plausible geo-models. Eq. (11) defines the error (ε_W), which is the averaged MAPE calculated at the given producer to effectively demonstrate the error trends during the history matching:

$$\varepsilon_W = \mathbb{E}_{T_k}\mathbb{E}_{0\sim t_n}\left\{\varepsilon\left(\boldsymbol{Y}, \widehat{\boldsymbol{Y}}\right)\right\} \tag{11}$$

where $\varepsilon(\boldsymbol{Y}, \widehat{\boldsymbol{Y}})$ is MAPE with the true production value ($\boldsymbol{Y}$) and the estimated value ($\widehat{\boldsymbol{Y}}$); the subscripts W, T_k, and t_n are the production well, the sample set, and the time step at the end of history matching, respectively (t_n=20; see Eq. (1)). Fig. 9 describes the distribution of ε_W with relation to the stages. Errors with small values converge up to the 20th stage, and early stopping is reached at the 18th stage. All errors decrease with the stages for the three objective functions. Therefore, these decreased trajectories validate the effectiveness of the iterative-learning process on matching the production histories. As a result of examining the data matching, the iterative-learning workflow can improve data matching of

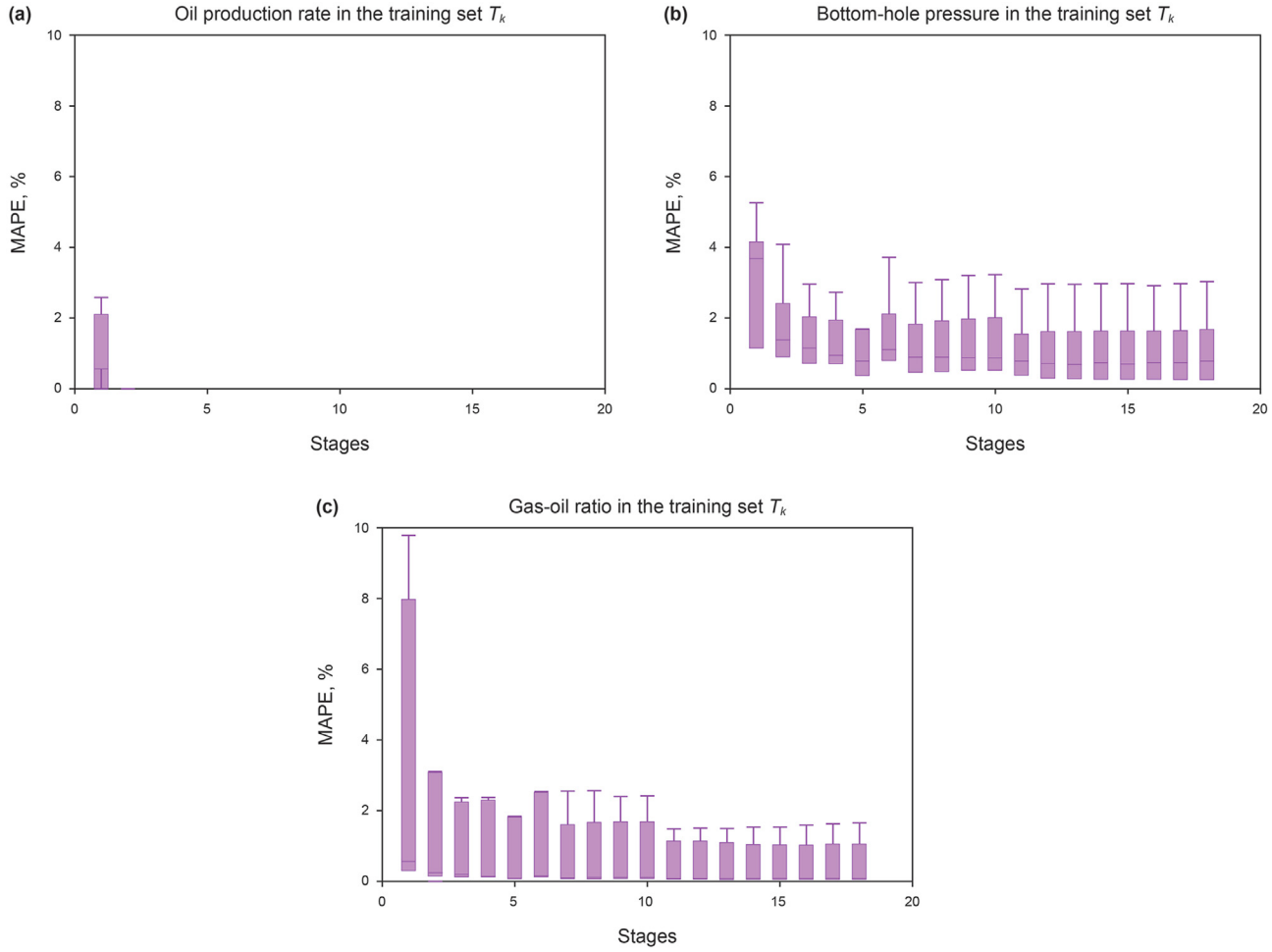

Fig. 9. Error bars for MAPE (ε_W) with relation to the stages: (**a**) oil production rate; (**b**) bottom-hole pressure; and (**c**) gas-oil ratio.

Table 6
Summary of the MAPE (ε_W) during the history matching period.

		MAPE ε_W, %						
		PROD1	PROD2	PROD3	PROD4	PROD5	PROD6	Average
Oil production rate	Initial models	1.4	1.8	0.0	0.1	0.7	0.0	0.6
	NSGA-II	1.3	1.8	0.0	0.1	1.2	0.0	0.7
	DNN-SAE	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bottom-hole pressure	Initial models	3.8	5.3	1.4	3.7	3.8	0.5	1.4
	NSGA-II	3.0	5.4	0.9	3.3	3.4	0.5	1.4
	DNN-SAE	0.3	0.9	1.2	0.6	3.0	0.0	0.7
Gas-oil ratio	Initial models	5.2	6.9	0.2	0.4	0.4	0.2	2.6
	NSGA-II	5.5	6.8	0.2	0.3	0.3	0.1	2.6
	DNN-SAE	1.2	0.6	0.1	0.0	0.1	0.0	0.4

DNN-SAE.

Table 6 compares ε_W with initial models (100 models; P_0), the final geo-models of NSGA-II (100 models at the 20th generation), and the final geo-models of DNN-SAE with iterative-learning (the proposed method; 251 models at the 18th generation). The averaged values of ε_W, $\mathbb{E}_W\{\varepsilon_W\}$, prove that the proposed workflow more effectively reduced the error ranges of the initial geo-models. Reducing all errors of 18 objective functions is difficult, and is referred to as "the curse of dimensionality". This results in some error reduction by the producers; however, other errors are not decreased by either method. A notable comparison is that NSGA-II fails to reduce overall errors in the initial models; however, the proposed method is effective in reducing individual error and improving overall data-matching. The averaged errors do not

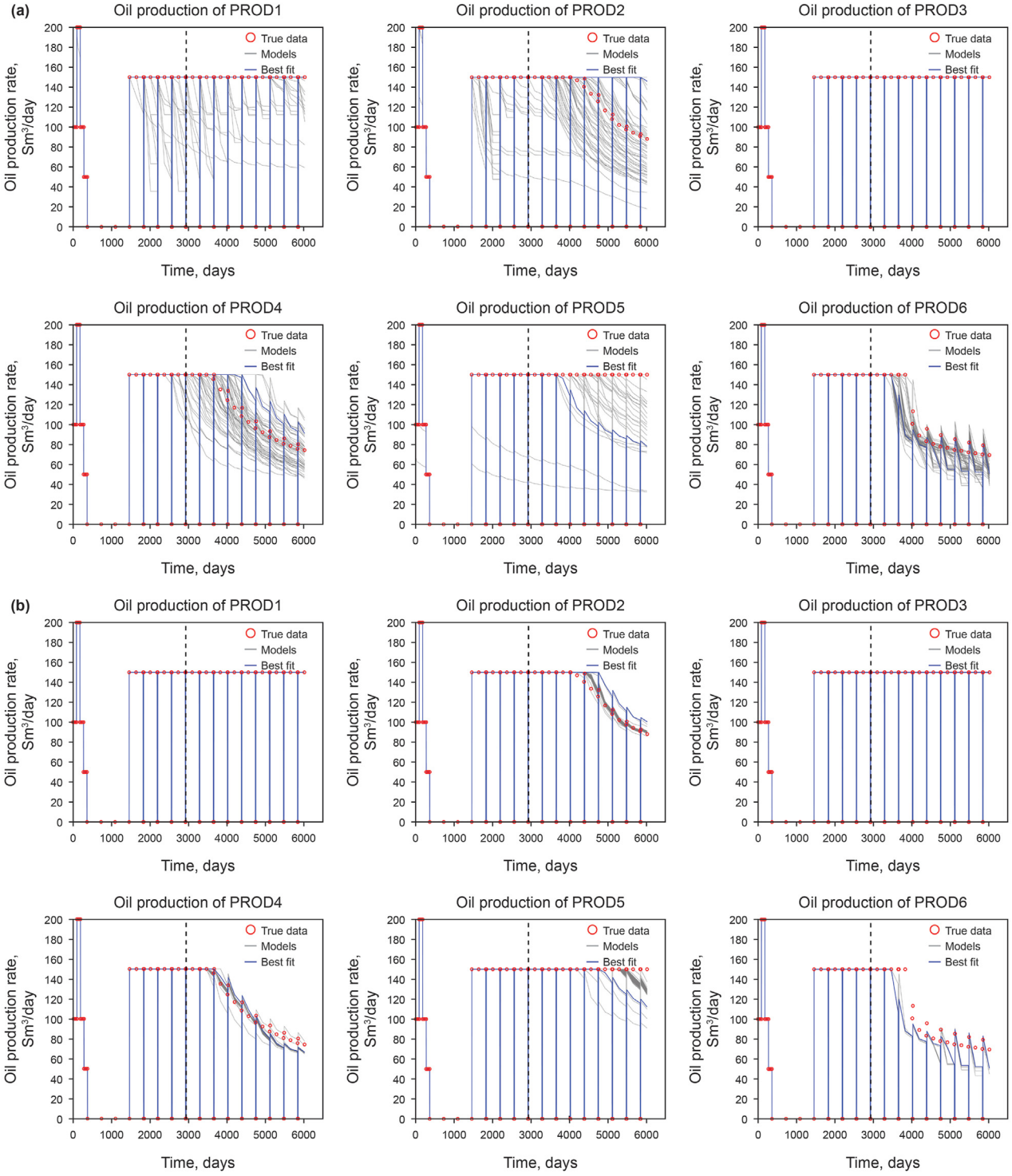

Fig. 10. Oil production profiles: (**a**) NSGA-II and (**b**) the study. The vertical dashed line indicates the end of the history matching period. The number of trajectories is 100 by NSGA-II and 251 in this study.

necessarily represent successfully satisfying many objectives because the outlier can govern the average value. Based on this problem, consideration of both the number of geo-models selected in distance maps (T_k) and the error bars is a valid indicator to estimate the robustness of DNN-SAE history matching at each stage.

Figs. 10–12 show production performances that allow examination of prediction accuracy for results of NSGA-II and the proposed method. The best-fit model denotes one geo-model that has the smallest error during the matching period, i.e., a plausible model with the minimum error, which is the $\min[\mathbb{E}_W\{\varepsilon_W\}]$. The number of trajectories is 100 models for NSGA-II, while 251 models were used in this study. Notwithstanding the proposed workflow, that accomplished early stopping before 20 stages, i.e., it was stopped at the 18th generation. The proposed method significantly

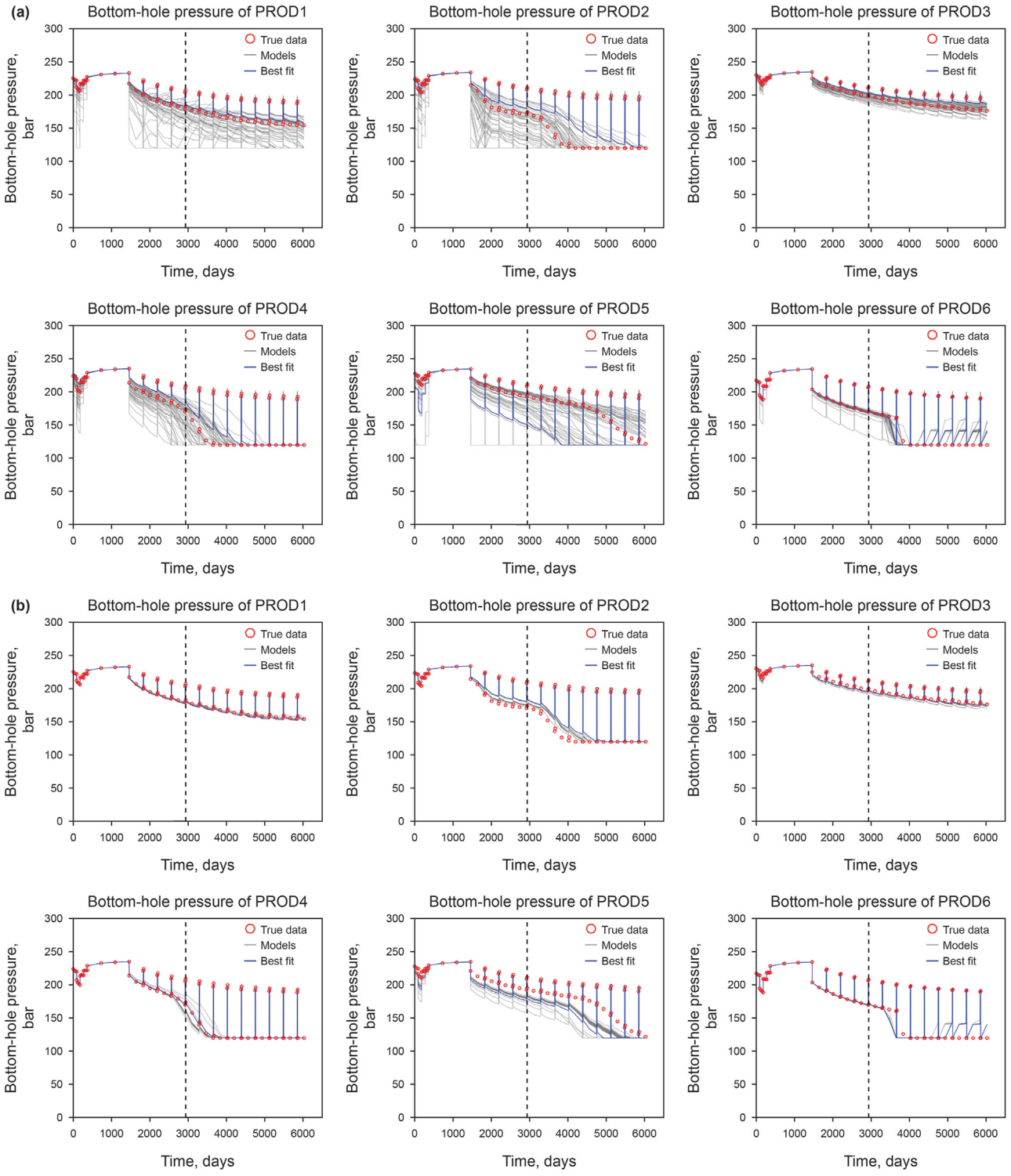

Fig. 11. Bottom-hole pressure profiles: (**a**) NSGA-II and (**b**) this study. The vertical dashed line indicates the end of the history matching period. The number of trajectories is 100 by NSGA-II and 251 in this study.

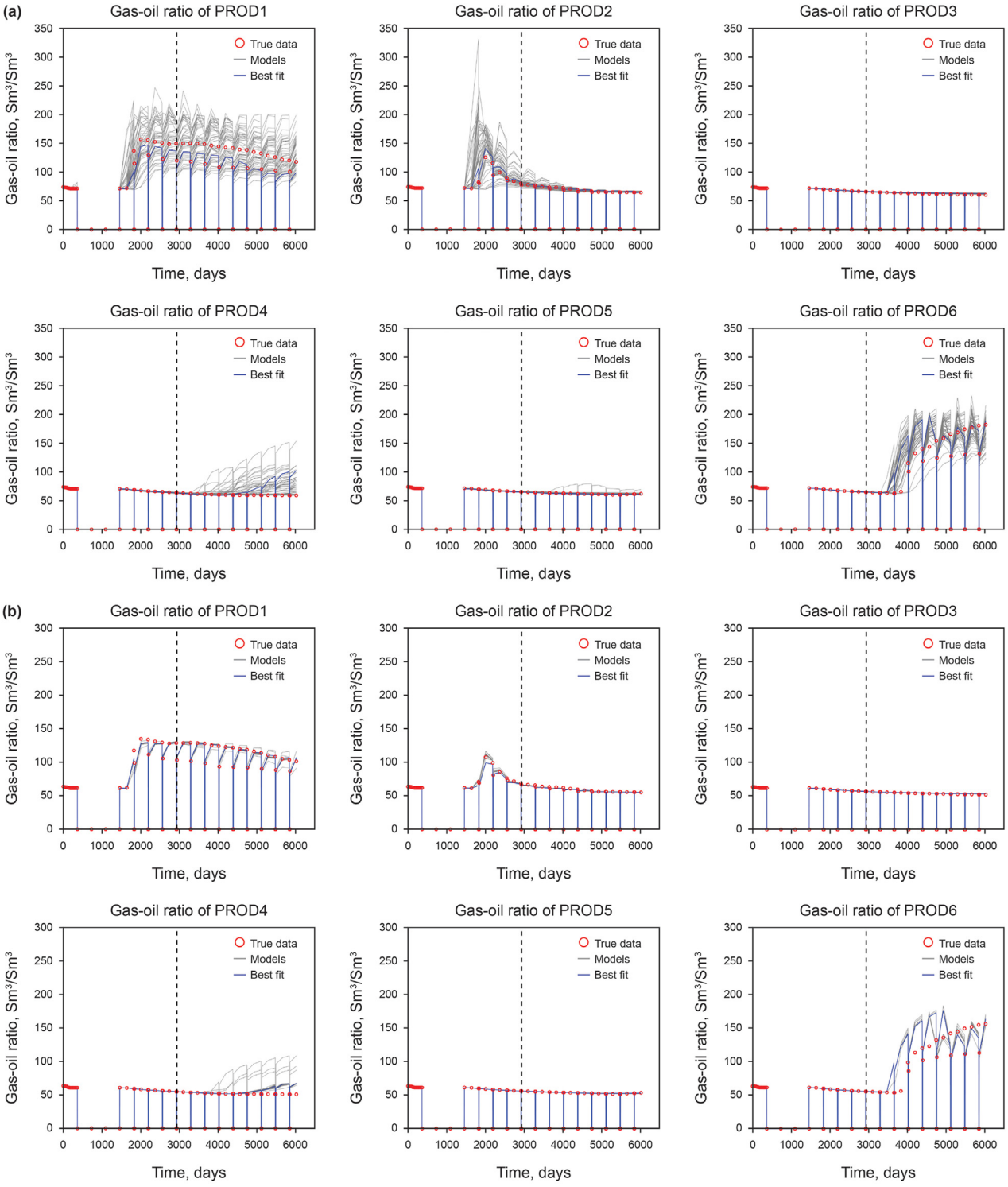

Fig. 12. Gas-oil ratio profiles: (**a**) NSGA-II and (**b**) this study. The vertical dashed line indicates the end of the history matching period. The number of trajectories is 100 by NSGA-II and 251 in this study.

reduced the trajectory ranges as compared with NSGA-II. Another notable result is that the iterative learning-based method generates more geo-models with higher accuracy. A representative objective of many-objective history matching is to construct as many plausible and consistent geo-models as possible. Thus, the proposed workflow can successfully perform many-objective history matching with less divergence. In addition, the forecasting ability of the proposed method is shown to be sufficiently reliable by satisfying 18 objectives. The best-fit model can explain the well-based production performances even though some cases are greater than the others, e.g., oil rates from PROD5 and PROD6. However, the predictability of the compared method (NSGA-II) is not generally insufficient, but the predictability of oil rates is relatively poor.

As water cuts are not used in matching, the reliability of plausible geo-models can be analyzed quantitatively through water cut

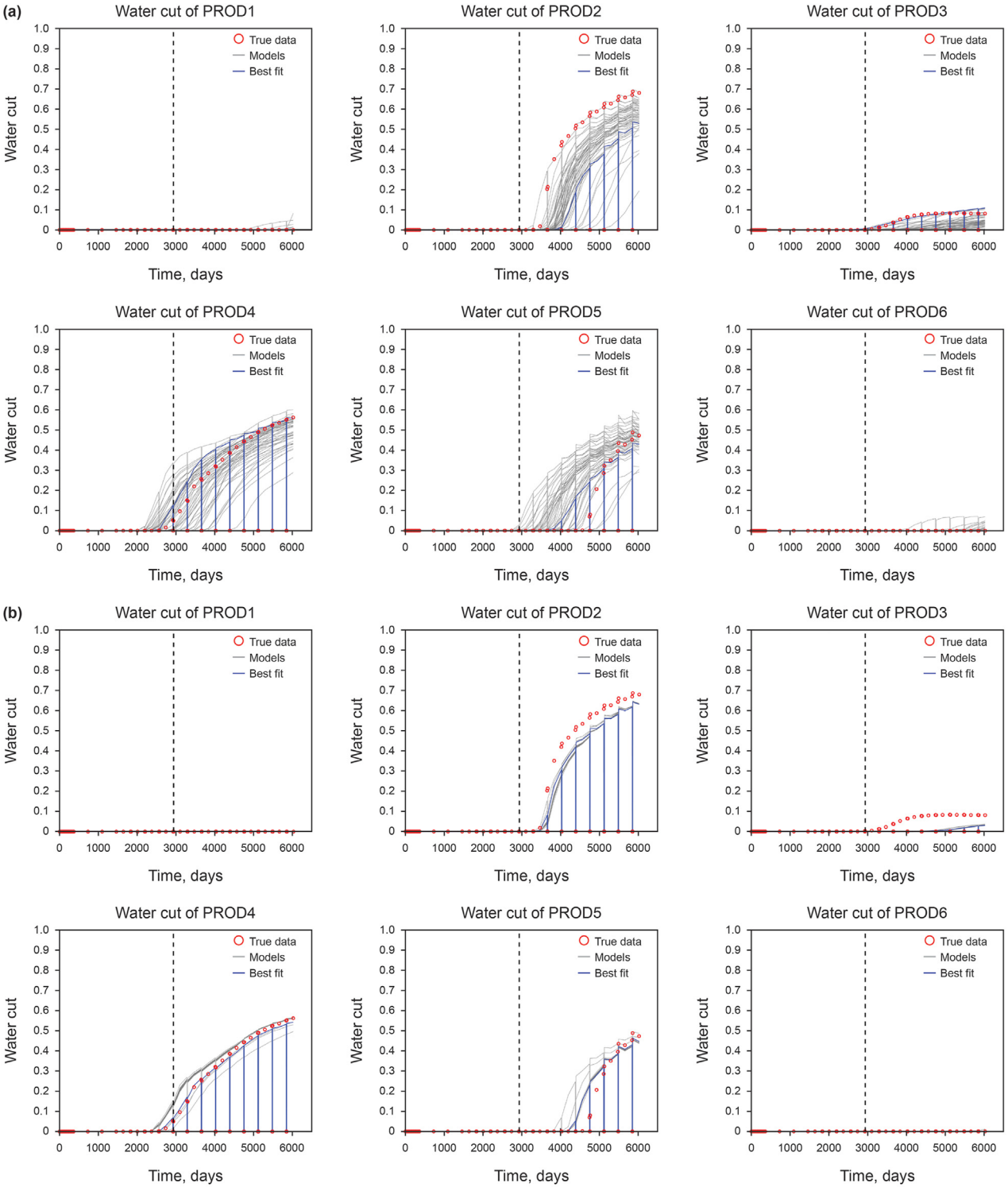

Fig. 13. Comparison of estimating water cut profiles: (**a**) NSGA-II and (**b**) the iterative learning-based many-objective history matching. The number of trajectories is 100 by NSGA-II and 251 in this study.

forecasting performances (Fig. 13). The proposed method forecasts water cuts more accurately than PROD2, 4, and 5, but is less accurate for PROD3. The developed workflow predicts more accurately for high water cut and water breakthrough time, and shows the lower overall error. Fig. 14 compares the field production volumes for oil, gas, and water as decision-making data. As shown in Fig. 14, there are decreases in the error ranges of the initial population, i.e., uncertainty level, and they can provide more reliable predictions. Table 7 summarizes the prediction errors (ε_P; Eq. (12)) of developed method and the comparison:

$$\varepsilon_P = \mathbb{E}_{T_k}\mathbb{E}_{t_n\sim}\left\{\varepsilon\left(\boldsymbol{Y}, \widehat{\boldsymbol{Y}}\right)\right\} \tag{12}$$

where ε_P is the MAPE from the end of history matching until the

end of production. The other variables are previously defined for Eq. (11). This method reduces the errors in initial geo-models and accurately forecasts unknown well-based production performances. The typical Pareto-based evolutionary algorithm does not significantly decrease the errors and the errors are similar to those of the initial population due to the "curse of dimensionality." The typical Pareto-based algorithm is not suitable for attaining global optimality in many-objective problems because it does not effectively reduce the initial errors. The proposed iterative learning-based DNN-SAE method is a promising alternative for mitigating "the curse of dimensionality."

Fig. 15 shows the permeability distribution of the best-fit geo-model and the MAPE between the true permeability and the estimated value, i.e., $\varepsilon(\boldsymbol{X}_{\text{true}}, \widehat{\boldsymbol{X}})$, where $\boldsymbol{X}_{\text{true}}$ is the normalized and natural-log transformed permeabilities in PUNQ-S3 and $\widehat{\boldsymbol{X}}$ is the estimated value of the best-fit model by the developed method. A notable observation is that the errors are minimal near the production wells. The best-fit model results from calibrating the permeabilities to match production histories to successfully decrease the error within the drainage. However, outside the drainage areas, there is no significant improvement as shown by higher errors.

The aforementioned results discuss the pros and cons of iterative learning-based many-objective history matching. There are three main advantages to using this method that are described as follows: (1) it reliably considers the observed dynamic data (well-based production characteristics) with equal weights. This advantage would be helpful in independently matching well-based production performances, and thereby would reduce the occurrence of overall errors; (2) iterative-learning provides the possibility to reduce the errors of all objective functions. The typical Pareto-based many-objective history matching has used some schemes to expand the search domain, e.g., non-dominated sorting and crowd distance; however, they are not always effective in many-objective problems. If the DNN structure is reliable for generating the equi-probable models, the model selection based on unsupervised-learning, e.g., *k*-modoids clustering and the expansion of searching domain for each objective function, would be effective for identifying geological uncertainty; and (3) minimal computing effort is required, despite the dependence on neural network design. The training of neural networks takes much less time than the typical Pareto-based multi-objective evolutionary

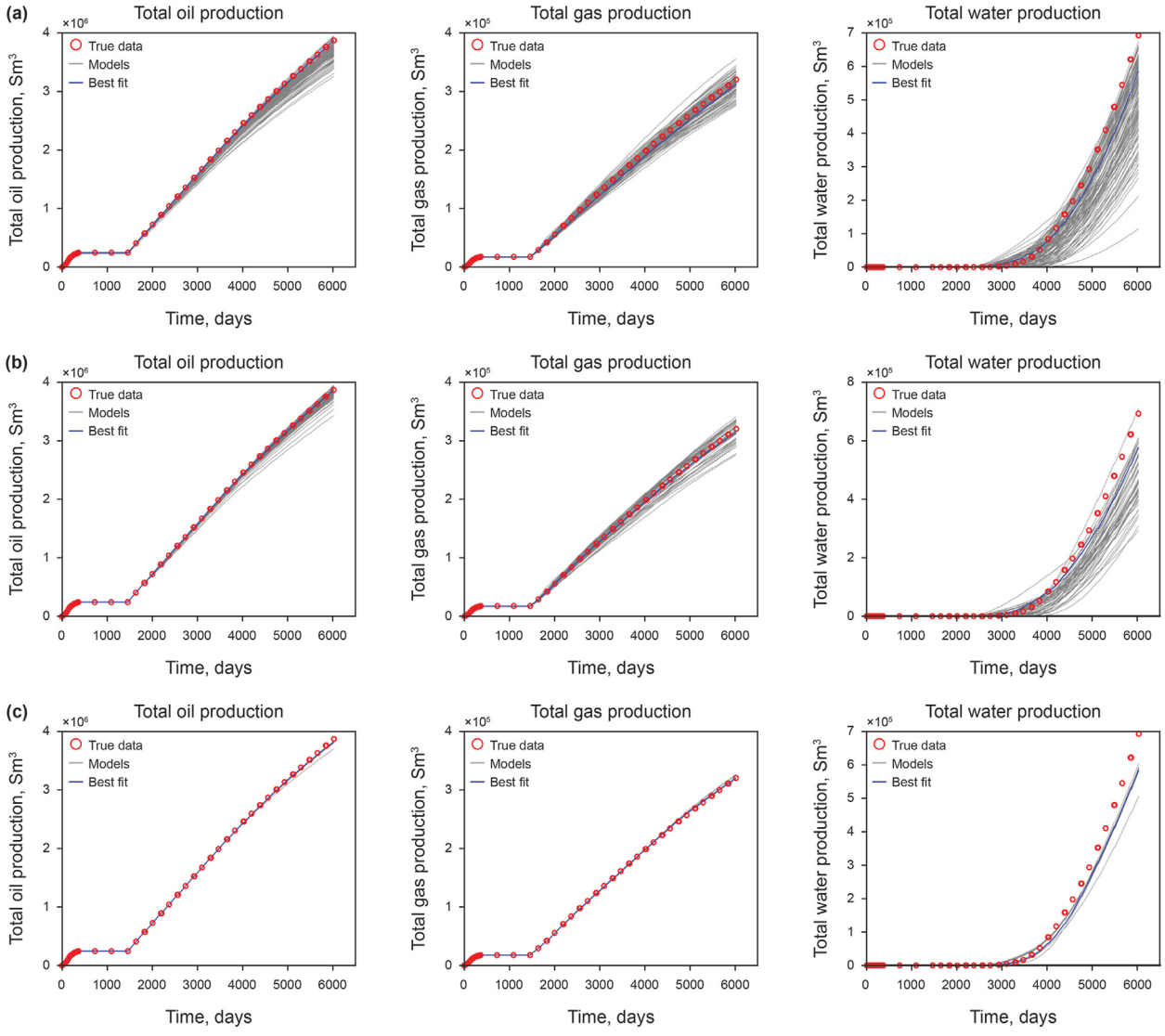

Fig. 14. Total production of oil and gas: (**a**) initial models; (**b**) NSGA-II; and (**c**) the proposed method. The number of trajectories is 100 by initial models, 100 by NSGA-II, and 251 in this study.

Table 7
Summary of prediction accuracy (MAPE; ε_P) for initial models, NSGA-II, and the proposed method (from 2937 to 6025 days).

		Prediction accuracy (MAPE) ε_P, %						
		PROD1	PROD2	PROD3	PROD4	PROD5	PROD6	Average
Oil production rate	Initial models	3.7	11.5	0.0	10.2	4.3	7.6	3.5
	NSGA-II	2.4	11.5	0.0	9.0	4.1	6.8	3.4
	DNN-SAE	0.0	0.9	0.0	3.0	1.1	8.8	2.4
Bottom-hole pressure	Initial models	6.7	4.4	2.6	3.0	7.8	3.8	1.7
	NSGA-II	5.4	4.1	2.0	2.9	8.0%	3.6	1.6
	DNN-SAE	0.5	2.6	1.5	0.9	6.8	4.7	1.9
Gas-oil ratio	Initial models	10.0	2.1	0.9	7.8	1.0	11.4	4.2
	NSGA-II	11.2	1.7	0.9	7.6	1.1	10.5	4.3
	DNN-SAE	1.1	0.6	0.5	5.3	0.2	13.1	3.8
Water cut	Initial models	0.0	22.5	48.9	21.5	352.1	0.0	92.6
	NSGA-II	0.0	24.0	46.2	20.0	352.7	0.0	93.0
	DNN-SAE	0.0	13.7	55.9	13.4	210.2	0.0	56.1

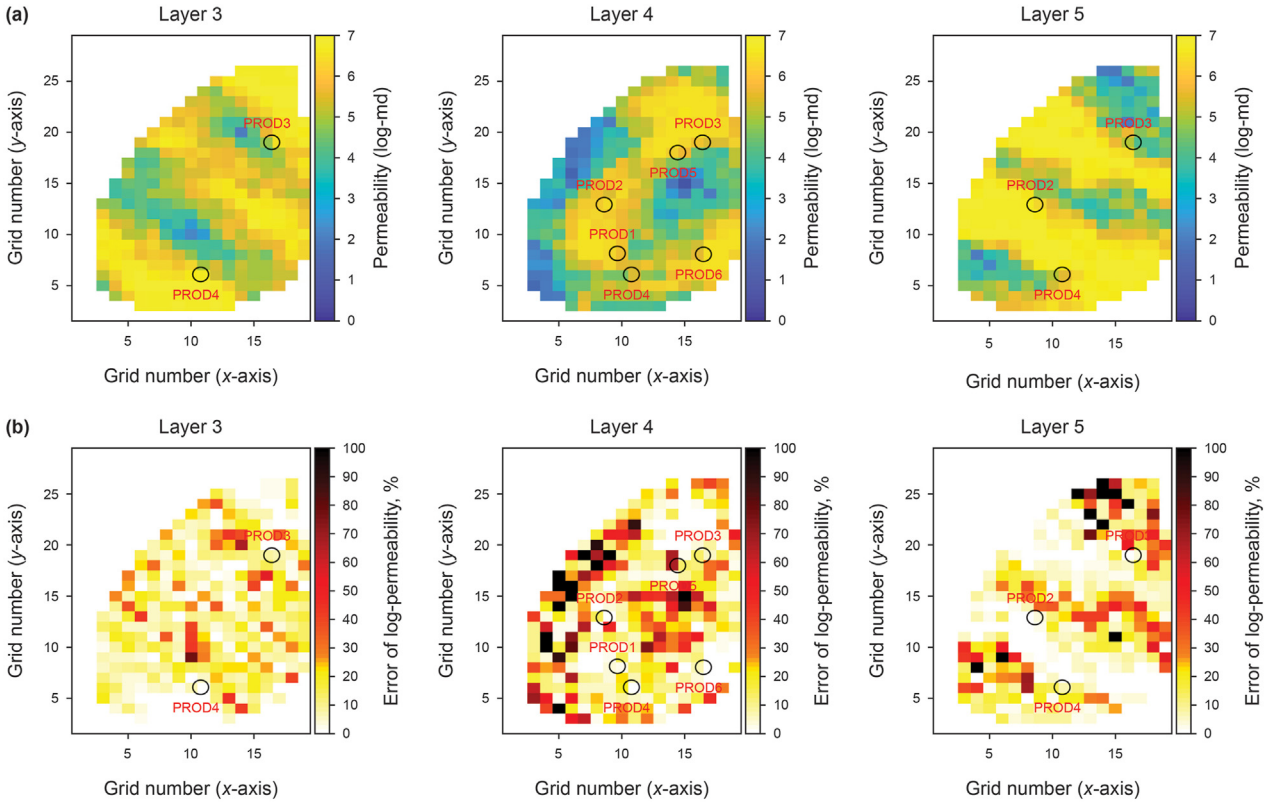

Fig. 15. Best-fit model resulting from the iterative learning-based DNN-SAE: (**a**) natural-log transformed permeabilities and (**b**) their error with the reference field (PUNQ-S3; Fig. 1(c), (d), (e)). The 3rd, 4th, and 5th layers are presented as they contain producers.

algorithm. It can generate plausible geo-models with small computing efforts. The weaknesses or challenges can be addressed to improve the iterative learning-based many-objective history matching by constructing standard forms of neural networks and correlation of the conflicted objectives. As deep learning is a data-driven neural-network method and searches the more accurate empirical relationship between the input and output units, some user interactions are required, e.g., detailed design of neural network, selection of activation functions, and tuning hyper parameters. If the overall errors are addressed, reduction in solving local erroneous objectives is the next challenge. If the initial population is sufficient for explaining the production profiles, the performances of many-objective history matching would be unsatisfied. The indicators, e.g., the stopping criteria, are challengeable that can evaluate the convergence of many-objective optimization. The iterative learning-based neural network implements some assumptions and thereby it is difficult to conclude whether or not the proposed workflow is optimal. Introducing various neural networks and integrating the parameterization or dimensionality reduction may efficiently secure global optimality as future works. The reliability of latent features (dimensionality-reduced parameters) extracted from the field-scaled three-dimensional geo-models is a key factor to secure the applicability of deep-learning-based data analytics.

5. Conclusions

This paper presented iterative learning-based many-objective history matching that is applicable to solving problems that occur when using many-objective methods. This workflow consisted of deep neural network for inverse modeling, stacked autoencoder to reduce dimensionality of static data, and geo-model selection in an individual objective-based distance map. The suggested workflow successfully addressed the divergence issue of many-objective optimizations by showing reliable performances and accurate predictability. Comparison with the Pareto-based multi-objective evolutionary algorithm confirmed that this proposed method improved the learning performance of the neural network by selecting the distance-based candidate group. The proposed method is applicable as a decision-making tool for reservoir development as proven by its ability to estimate dynamic responses with high reliability; even in a reservoir containing many production wells in three-dimensional geo-models.

Acknowledgments

This study was supported by the basic science research program through the National Research Foundation of Korea (NRF) (2020R1F1A1073395), and the basic research project of the Korea Institute of Geoscience and Mineral Resources (KIGAM) (GP2021-011; GP2020-031; 21-3117) funded by the Ministry of Science and ICT, Korea. The research facilities are provided by the Institute of Engineering Research at Seoul National University, Republic of Korea.

References

Ahn, S., Park, C., Kim, J., Kang, J.M., 2018. Data–driven inverse modeling with a pre–trained neural network at heterogeneous channel reservoirs. J. Petrol. Sci. Eng. 170, 785–796. https://doi.org/10.1016/j.petrol.2018.06.084.

Cavalcante, C.C.B., Maschio, C., Santos, A.A.S., Schiozer, D., 2019. A continuous learning algorithm for history matching. Eng. Appl. Artif. Intell. 85, 543–568. https://doi.org/10.1016/j.engappai.2019.07.012.

Carneiro, J., Azevedo, L., Pereira, M., 2018. High-dimensional geostatistical history matching: vectorial multi–objective geostatistical history matching of oil reservoirs and uncertainty assessment. Comput. Geosci. 22, 607–622. https://doi.org/10.1007/s10596-017-9712-6.

Chan, S., Elsheikh, A.H., 2019. Parametric generation of conditional geological realizations using generative neural networks. Comput. Geosci. 23, 925–952. https://doi.org/10.1007/s10596-019-09850-7.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6, 182–197. https://doi.org/10.1109/4235.996017.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P., Bengio, S., 2010. Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. 11, 625–660. https://dl.acm.org/doi/10.5555/1756006.1756025.

Esmaeilzadeh, S., Salehi, A., Hetz, G., Olalotiti-lawal, F., Darabi, H., Castineira, D., 2020. Multiscale modeling of compartmentalized reservoirs using a hybrid clustering-based non-local approach. J. Petrol. Sci. Eng. 184, 106485. https://doi.org/10.1016/j.petrol.2019.106485.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT press, Cambridge, MA, USA.

Gu, Y., Oliver, D.S., 2005. History matching of the PUNQ-S3 reservoir model using the ensemble Kalman filter. SPE J. 10 (2), 217–224. https://doi.org/10.2118/89942-PA.

Guria, C., Gori, K.K., Pathak, A.K., 2014. Multi-objective optimization of oil well drilling using elitist non-dominated sorting genetic algorithm. Petrol. Sci. 11, 97–110. https://doi.org/10.1007/s12182-014-0321-x.

Han, Z., Hossain, M.M., Wang, Y., Li, J., Xu, C., 2020. Combustion stability monitoring through flame imaging and stacked sparse autoencoder based deep neural network. Appl. Energy 259, 114159. https://doi.org/10.1016/j.apenergy.2019.114159.

Hegstad, B.K., More, H., 2001. Uncertainty in production forecasts based on well observations, seismic data, and production history. SPE J. 6 (4), 409–424. https://doi.org/10.2118/74699-PA.

Hinton, G., Osindero, S., Teh, Y., 2006. A fast learning algorithm for deep belief nets. Neural Comput. 18 (7), 1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527.

Hinton, G., Salakhutdinov, R., 2006. Reducing the dimensionality of data with neural networks. Science 313 (5786), 504–507. https://doi.org/10.1126/science.1127647.

Hutahaean, J., Demyanov, V., Christie, M.A., 2017. On optimal selection of objective grouping for multiobjective history matching. SPE J. 22 (4), 1296–1312. https://doi.org/10.2118/185957-PA.

Ki, S., Jang, I., Cha, B., Seo, J., Kwon, O., 2020. Restoration of missing pressures in a gas well using recurrent neural networks with long short-term memory cells. Energies 13, 4696. https://doi.org/10.3390/en13184696.

Kim, J., Kang, J.M., Park, C., Park, Y., Park, J., Lim, S., 2017. Multi-objective history matching with a proxy model for the characterization of production performances at the shale gas reservoir. Energies 10 (4), 579. https://doi.org/10.3390/en10040579.

Kim, J., Kim, S., Park, C., Lee, K., 2020a. Construction of prior models for ES-MDA by a deep neural network with a stacked autoencoder for predicting reservoir production. J. Petrol. Sci. Eng. 187, 106800. https://doi.org/10.1016/j.petrol.2019.106800.

Kim, J., Park, C., Lee, K., Ahn, S., Jang, I., 2020b. Deep neural network coupled with distance-based model selection for efficient history matching. J. Petrol. Sci. Eng. 185, 106658. https://doi.org/10.1016/j.petrol.2019.106658.

Koneshloo, M., Aryana, S.A., Hu, X., 2018. The impact of geological uncertainty on primary production from a fluvial reservoir. Petrol. Sci. 15, 270–288. https://doi.org/10.1007/s12182-018-0229-y.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539.

Lee, K., Jung, S., Lee, T., Choe, J., 2017. Use of clustered covariance and selective measurement data in ensemble smoother for three–dimensional reservoir characterization. J. Energy Resour. Technol. 139, 022905. https://doi.org/10.1115/1.4034443.

Lim, S., Park, C., Kim, J., Jang, I., 2020. Integrated data assimilation and distance-based model selection with ensemble Kalman filter for characterization of uncertain geological scenarios. Nat. Resour. Res. 29, 1063–1085. https://doi.org/10.1007/s11053-019-09489-2.

Liu, M., Grana, D., 2020. Time-lapse seismic history matching with an iterative ensemble smoother and deep convolutional autoencoder. Geophysics 85, M15–M63. https://doi.org/10.1190/geo2019-0019.1.

Liu, Y., Sun, W., Durlofsky, L.J., 2019. A deep-learning-based geological parameterization for history matching complex models. Math. Geosci. 51, 725–766. https://doi.org/10.1007/s11004-019-09794-9.

MathWorks, 2018. MATLAB Programming Fundamentals. Natick. The MathWorks, Inc., USA.

Mehta, A., 2016. Tapping the value from big data analytics. J. Petrol. Technol. 68 (12), 40–41. https://doi.org/10.2118/1216-0040-JPT.

Min, B., Kang, J.M., Lee, H., Jo, S., Park, C., Jang, I., 2016. Development of a robust multi-objective history matching for reliable well-based production forecasts. Energy Explor. Exploit. 34, 795–809. https://doi.org/10.1177/0144598716665008.

Mo, S., Zhu, Y., Zabaras, N.J., Shi, X., Wu, J., 2019. Deep convolutional encoder–decoder networks for uncertainty quantification of dynamic multiphase flow in heterogeneous media. Water Resour. Res. 55, 703–728. https://doi.org/10.1029/2018WR023528.

Remy, N., Boucher, A., Wu, J., 2009. Applied Geostatistics with SGeMS: a User's Guide. Cambridge University Press, New York.

Reed, R., Marks, R.J., 1999. Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks. MIT Press, London, UK.

Ribeiro, M., Lazzaretti, A.E., Lopes, H.S., 2018. A study of deep convolutional auto-encoders for anomaly detection in videos. Pattern Recogn. Lett. 105, 13–22. https://doi.org/10.1016/j.patrec.2017.07.016.

Saputelli, L., 2019. Technology focus: data analytics. J. Petrol. Technol. 71 (10), 61. https://doi.org/10.2118/1019-0061-JPT.

Scheidt, C., Caers, J., 2009. Uncertainty quantification in reservoir performance using distances and kernel methods: application to a West Africa deepwater turbidite reservoir. SPE J. 14 (4), 680–692. https://doi.org/10.2118/118740-PA.

Scheidt, C., Li, L., Caers, J., 2018. Quantifying Uncertainty in Subsurface Systems. John Wiley & Sons Inc., Hoboken, NJ, USA.

Schlumberger, 2018. Eclipse Reference Manual 2018.1. Schlumberger, Houston, USA.

Siena, M., Guadagnini, A., Della Rossa, E., Lamberti, A., Masserani, F., Rotondi, M., 2016. A novel enhanced-oil-recovery screening approach based on Bayesian clustering and principal-component analysis. SPE J. 19 (3), 382–390. https://doi.org/10.2118/174315-PA.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P., 2010. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. 11, 3371–3408. https://dl.acm.org/doi/10.5555/1756006.1953039.

Weston, J., Ratle, F., Mobahi, H., Collobert, R., 2012. Deep learning via semi-supervised embedding. In: Montavon, G., Orr, G.B., Müller, K.R. (Eds.), Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, vol. 7700. Springer, Berlin, Germany, pp. 639–655. https://doi.org/10.1007/978-3-642-35289-8_34.

Zeng, K., Yu, J., Wang, R., Li, C., Tao, D., 2017. Coupled deep autoencoder for single image super-resolution. IEEE T Cybernetics 47 (1), 27–37. https://doi.org/10.1109/TCYB.2015.2501373.

Zhang, D., Shen, A., Jiang, X., Kang, Z., 2018. Efficient history matching with dimensionality reduction methods for reservoir simulations. Simulation 94 (8), 739–751. https://doi.org/10.1177/0037549717742963.

Zhou, C., Paffenroth, R.C., 2017. Anomaly detection with robust deep autoencoders. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 665–674. https://doi.org/10.1145/3097983.3098052.