

A vector-to-sequence based multilayer recurrent network surrogate model for history matching of large-scale reservoir



Xiaopeng Ma ^{a,b}, Kai Zhang ^{a,b,*}, Hanjun Zhao ^c, Liming Zhang ^b, Jian Wang ^d, Huaqing Zhang ^d, Piyang Liu ^a, Xia Yan ^b, Yongfei Yang ^b

^a School of Civil Engineering, Qingdao University of Technology, Qingdao, China

^b School of Petroleum Engineering, China University of Petroleum, Qingdao, China

^c Petrochina Exploration and Production Company, Beijing, China

^d School of Science, China University of Petroleum, Qingdao, China

ARTICLE INFO

Keywords:

History matching
Surrogate modeling
Recurrent neural network
Normalization method

ABSTRACT

History matching can estimate the parameter of spatially varying geological properties and provide reliable numerical models for reservoir development and management. However, in practice, high-dimension, multiple-solutions and computational cost are key issues that restrict the application of history matching methods. Recently, the combination of deep-learning-based surrogate model and sampling algorithm has been widely studied in history matching to overcome the limitations. Considering that real-world large-scale reservoirs often have hundreds of thousands or even millions of grid-based uncertain parameters, extracting spatial features using convolutional neural networks requires a lot of computational cost and storage requirements. Therefore, in this work, we mainly study how to use the recurrent neural network (RNN) to construct the surrogate model for history matching. Specifically, we propose a multilayer RNN surrogate model based on a vector-to-sequence modeling framework. The multilayer RNN surrogate model with gated recurrent unit (GRU), termed MLGRU, is developed to approximate the mapping from feature vector of geological realizations to the production data. The feature vector is the low-dimensional representation of geological parameter fields after using the re-parameterization method, while production data are the simulation results of historical period. In addition, we design a log-transformation-based windowed normalization (LTWN) method for the production data, which can enhance the learnability and features of production data. The MLGRU model is incorporated into a multi-modal estimation of distribution algorithm (MEDA) to formulate a history matching workflow. The hyper-parameters and performance of the proposed MLGRU model are analyzed by numerical experiments on a 2D reservoir model. Furthermore, numerical experiments performed on the Brugge benchmark model, a large-scale 3D reservoir model, demonstrated the performance of the proposed surrogate model and history matching method.

1. Introduction

Spatially varying geological properties, such as permeability and porosity, are the main source of uncertainty in reservoir numerical modeling. This uncertainty makes it difficult for the numerical model to effectively simulate the underground flow. In the last decades, history matching or data assimilation methods have been widely studied to reduce the uncertainty of geological properties. Based on the Bayesian probability inference, the history matching method (Oliver and Chen, 2011; Li et al., 2019) can estimate the parameters by integrating static

reservoir information with dynamic production observation data. Generally, history matching is an ill-posed inverse problem with multiple solutions (Chen et al., 2018; Zhang et al., 2018a; Ma et al., 2021) due to the available data being sparse to the subsurface structure. In addition, the estimation of subsurface parameters is also a high-dimensional sampling problem, which needs a great deal of time-consuming numerical simulations. Therefore, how to effectively solve such a computationally expensive and multi-solution inverse problem remains a challenging issue in science and engineering areas.

Recently, the combination of deep learning (DL) based surrogate

* Corresponding author. School of Petroleum Engineering, China University of Petroleum, Qingdao, China.

E-mail address: zhangkai@upc.edu.cn (K. Zhang).

model and sampling algorithm has attracted extensive attention in the field of history matching (Mo et al., 2019; Tang et al., 2020; Zhong et al., 2020; Zhang et al., 2021; Xiao et al., 2022). The DL-based surrogate model can approximate the numerical simulation with high accuracy and be cheap to evaluate, which can replace the simulation model during the search of sampling algorithm (Mo et al., 2019). Therefore, some sampling algorithms require a lot of numerical evaluations that can be used for solving the history matching problems. For example, in (Mo et al., 2019), a deep residual convolutional neural network (CNN) is utilized as the surrogate and combined with a local updating ensemble smoother algorithm (Zhang et al., 2018a). The generative adversarial network (GAN) model also has been introduced into this surrogate modeling task (Zhong et al., 2021). Compared with conventional CNN, adversarial learning enables GAN to produce higher-resolution predictions. Zhang et al. (2018b) used the relative permeability curve as an auxiliary input to predict pressure field and saturation field together with geological properties. In order to reduce the data requirements for training the surrogate, Wang et al. (2021) introduced the physical information and domain knowledge to the surrogate modeling process. In general, the above DL-based surrogate models are based on an image-to-image modeling framework (Zhu and Zabaras, 2018). In the image-to-image surrogate modeling framework, the surrogate model is mainly used to construct the mapping of geological parameter fields to flow field, such as the prediction of saturation field from permeability field. These spatially varying fields can be regarded as image data, so CNN models can process these data and extract high-level features. For history matching, the image-to-image based surrogate model needs to be combined with Paceman equations to calculate and predict production data of wells. Another DL-based surrogate model for history matching is constructed based on an image-to-sequence modeling framework (Ma et al., 2021b), in which, the CNN model is used to extract the spatial features of geological properties while the recurrent neural network (RNN) model is adopted to prediction the production data. In their following work (Ma et al., 2022), well control parameters are used as auxiliary inputs to further improve the accuracy of the surrogate model.

The above surrogate models can improve the computational efficiency of history matching. However, these methods always need to deal with geological parameter fields, whereas actual large-scale reservoir models often have hundreds of thousands or even millions of grids. Feature extraction using CNN requires a lot of computing requirements. In this work, we investigate the prediction of production data directly from low-dimensional features under a vector-to-sequence (vec2seq) modeling framework. In the last few years, based on the RNN model (Hochreiter and Schmidhuber, 1997; Graves et al., 2013; Yuan-yuan et al., 2016; Sun et al., 2017), the vec2seq and sequence-to-sequence (seq2seq) modeling framework have made a significant breakthrough in the field of machine translation and speech recognition (Cho et al., 2014; Makin et al., 2020). On the other hand, the RNN model has also been widely studied and applied in the field of petroleum engineering. Chen and Zhang (2020) utilized the long short-term memory (LSTM) network for the generation of well-log. Song et al. (2020) studied the single well production prediction based on the LSTM model. Fan et al. (2021) combined the autoregressive integrated moving average model with LSTM to predict the well production. Li et al. (2022) developed a bidirectional gated recurrent unit (GRU) model for the well production forecasting. Basically, these well production forecast studies are under the seq2seq modeling framework (Chung et al., 2015), in which the input and output are time-series data.

Vec2seq is another RNN-based modeling framework (Donahue et al., 2017; He and Deng, 2017; Gao et al., 2021), in which the input and output are the feature vector and time-series data, respectively. However, there are two main difficulties in applying the vec2seq learning framework to surrogate modeling for history matching problems. Firstly, the mapping from low-dimensional feature vectors to production data is very nonlinear, which includes the reconstruction of parameter fields and numerical simulation. Secondly, in history matching,

production data corresponding to different realizations may be very similar and easy to confuse. In this work, we propose a multi-layer GRU surrogate model and a log-transformation-based windowed normalization (LTWN) method to tackle these two issues. The MLGRU model is incorporated into a multimodal estimation of distribution algorithm (MEDA) (Yang et al., 2017; Ma et al., 2022) to formulate a history matching framework. The remainder work is organized as follows: we first introduce the details of MLGRU surrogate model and LTWN method, then introduce the surrogate-based history matching workflow. Afterward, we discuss and analyze the experimental results of case studies. Finally, we make a summary of this research and prospects for future work.

2. Proposed MLGRU surrogate model

Geological properties, such as permeability and porosity, are the main source of uncertainty in reservoir numerical modeling. In history matching, the uncertainty variable m of geological properties can be adjusted by using the production observation data $d_{obs} \in R^{N_{obs} \times T}$, such as oil or water production rate of different wells. T is the number of time-steps in the historical observation period, and N_{obs} is the number of observation indexes at each time-step. However, the uncertainty variable m of geological properties are difficult to adjust directly because they are spatially varying and represented by grid-based parameters. Generally, the m is high-dimensional and needed to be reparametrized to a low-dimensional vector $\xi \in R^{N_q}$ by reparameterization method. N_q is the dimensional of ξ , which is far less than the dimension of m . Based on the reparameterization method, the geological uncertainty variable m can be generated by $m = \Phi(\xi)$, where $\Phi(\cdot)$ represents the geological forward modeling. Therefore, a common operation in history matching is generating a new realization of m from the updated low-dimensional feature vector ξ and input to the numerical simulator to obtain the corresponding production data $y \in R^{N_{obs} \times T}$, which can be expressed as $y = g(\Phi(\xi))$. The $g(\cdot)$ represents the forward numerical simulation. It is worth noting that this operation leads to the history matching is computationally time-consuming. As we all know, the numerical simulation process is expensive in CPU time. Additionally, the geological forward modeling of high-dimensional parameter fields requires extra computational requirements, especially for large-scale reservoir models. Based on these findings, in this study, we propose a surrogate model to replace the expensive process to improve the computation efficiency. Specifically, the mapping of this surrogate modeling task is $f: R^{N_q} \rightarrow R^{N_{obs} \times T}$. In this work, a basic gated recurrent neural network model, GRU, is adopted to formulate our MLGRU surrogate model.

2.1. General architecture of MLGRU

Fig. 1 depicts the architecture of the proposed MLGRU model which

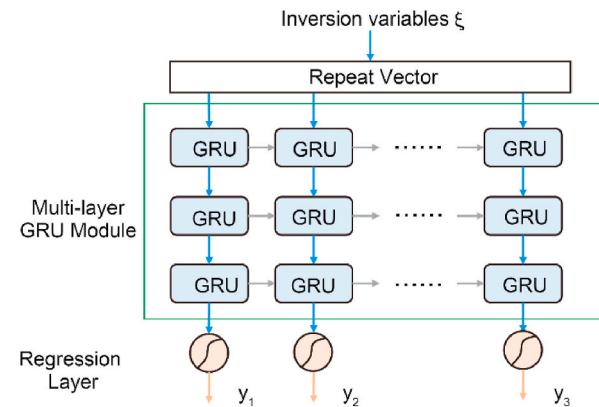


Fig. 1. The architecture of the MLGRU surrogate model.

mainly consists of stacked GRU layers. The inversion vector ξ is expanded along the timeline as input for each time step. Production data, as the output, are typical time-series data with T time steps and each time step has N_{obs} features. Such a regression task is from vector-to-sequence. Sequence is a machine learning term for a class of data which is an arrangement of any objects, such as sentences in natural language processing, voice data in speech recognition, etc. The time-series data also is a subclass of sequential data. The vector-to-sequence modeling task is as opposed to the conventional time-series task. Conventional timeseries prediction is sequence-to-sequence, for example, using the sequence data of first sever time-steps to predict the sequence of the next sever time-steps. Previous studies (Li et al., 2022; Fan et al., 2021) have shown that single-layer RNN model can achieve good results for this conventional prediction. However, the surrogate modeling in this work is to predict time series data from low-dimensional spatial feature vector, so it is necessary to use a deep stacked structure of RNN to learn the complex nonlinear mapping between inputs and outputs. Existing studies have proved that multilayer structure can enhance the representative learning ability.

The GRU structure was originally designed for the machine translation problems (Cho et al., 2014b) and then has been widely used in many time series tasks. Fig. 2 depicts the schematic diagram of the GRU. It adopts update gate and reset gate to regulate the flow of information and adaptively capture the dependency of time series data. Compared with the LSTM unit which needs to calculate three gate vectors, GRU is much simpler to implement and compute. Given the time-series input feature $\{x_1, \dots, x_{t-1}, x_t, x_{t+1}, \dots, x_T\}$, the structure and parameters of GRU are shared on the timeline for the calculation of hidden state $\{h_1, \dots, h_{t-1}, h_t, h_{t+1}, \dots, h_T\}$. The $h_t = \text{GRU}(h_{t-1}, x_t)$ is computed recursively, Fig. 2 depicts the detailed calculation process of h_t . As shown in Fig. 2, the current input feature is x_t , and the previous hidden state is h_{t-1} , the GRU first calculates the reset gate r_t and the update gate z_t by

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \quad (1)$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \quad (2)$$

where σ is the logistic sigmoid activation function, W_r , U_r , W_z , and U_z are weight matrices which are learned. Then the temporary current hidden state \tilde{h}_t is computed by

$$\tilde{h}_t = \tanh(W_x x_t + U(r_t \odot h_{t-1})) \quad (3)$$

where \tanh is the hyperbolic tangent activation function. Finally, the current hidden state h_t is a weighted average of the previous hidden state h_{t-1} and the temporary hidden state \tilde{h}_t :

$$h_t = z_t h_{t-1} + (1 - z_t) \tilde{h}_t \quad (4)$$

The calculation process of $h_{t-1} = \text{GRU}(h_{t-2}, x_{t-1})$ is same as h_t , and h_0

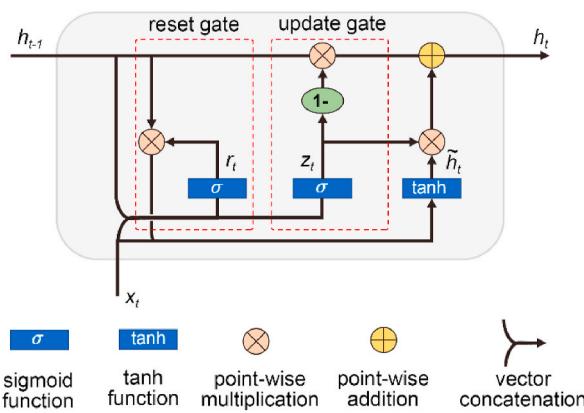


Fig. 2. An illustration of the GRU.

is a hyperparameter which usually is initialized to be zero. We can see that the reset gate is used to force adaptively forgetting the previous hidden state and resetting it with the current input, meanwhile the update gate controls how much previous information is retained to the current state. The GRU provides a compact representation of the dependence of time series data. After using multi-layer stacked GRUs, the final regression layer (output layer) is constructed using fully-connected units, and the tanh activation function is used. For each time step, the number of neurons is equal to the number of features N_{obs} .

2.2. Loss function and training strategy

Loss function is crucial for the training and learning of neural network. We optimize the MLGRU by minimizing the mean absolute error (MAE) loss. The simplified form of the loss function is defined as follows:

$$J(\Theta) = \frac{1}{N} \sum_{i=1}^N |y_i^{sim} - y_i^{pre}| \quad (5)$$

where Θ is the trainable parameters in the model, y_i^{sim} and y_i^{pre} are the simulation data and prediction data of i th sample, respectively.

The Adam optimizer (Kingma et al., 2014) with a start learning rate of 0.0001 is used to train the surrogate model. To prevent overfitting, the epoch is determined to be 100 after trial and error. Given that the sample size of the surrogate modeling is small, the batch size is set as 16.

2.3. Performance evaluation

To monitor and evaluate the performance of the surrogate model, the coefficient of determination R^2 and the root-mean-square error $RMSE$ are used as the evaluation metric in this study. As shown in Eq. (6), the R^2 measures the extent to the regression predictions approximate real data. The prediction is closer to the real data when the R^2 is closer to 1. The $RMSE$ metric can evaluate the prediction accuracy, as shown in Eq. (7).

$$R^2 = 1 - \frac{\sum_{i=1}^N \|y_i^{sim} - y_i^{pred}\|_2^2}{\sum_{i=1}^N \|y_i^{sim}\|_2^2} \quad (6)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i^{sim} - y_i^{pred})^2} \quad (7)$$

where N is the number of samples, y_i^{sim} and y_i^{pred} are the simulation data and prediction data, respectively. y_m^{sim} is the mean value of the simulation data.

2.4. Log-transformation based windowed normalization for production data

It is critical to process the data effectively in modeling using machine learning methods. In this study, we design a normalization method according to the characteristics of production data in history matching task. As shown in Fig. 3a, we randomly selected 10 samples of the production data (from training samples of case 1 in section 4.1) for illustration. On the one hand, the production data $y \in R^{N_{obs} \times T}$ is a time-series data. The changing trend of production data on different time needs to be considered. On the other hand, the production data are the simulation results of different geological realizations with similarity, so the features of different production data are easily confused. Using the conventional normalization methods, such as max-min normalization in the global scope, will lead to distortion of some data features because of the wide gap in production data at different timesteps.

In this section, we propose the LTWN method for production data.

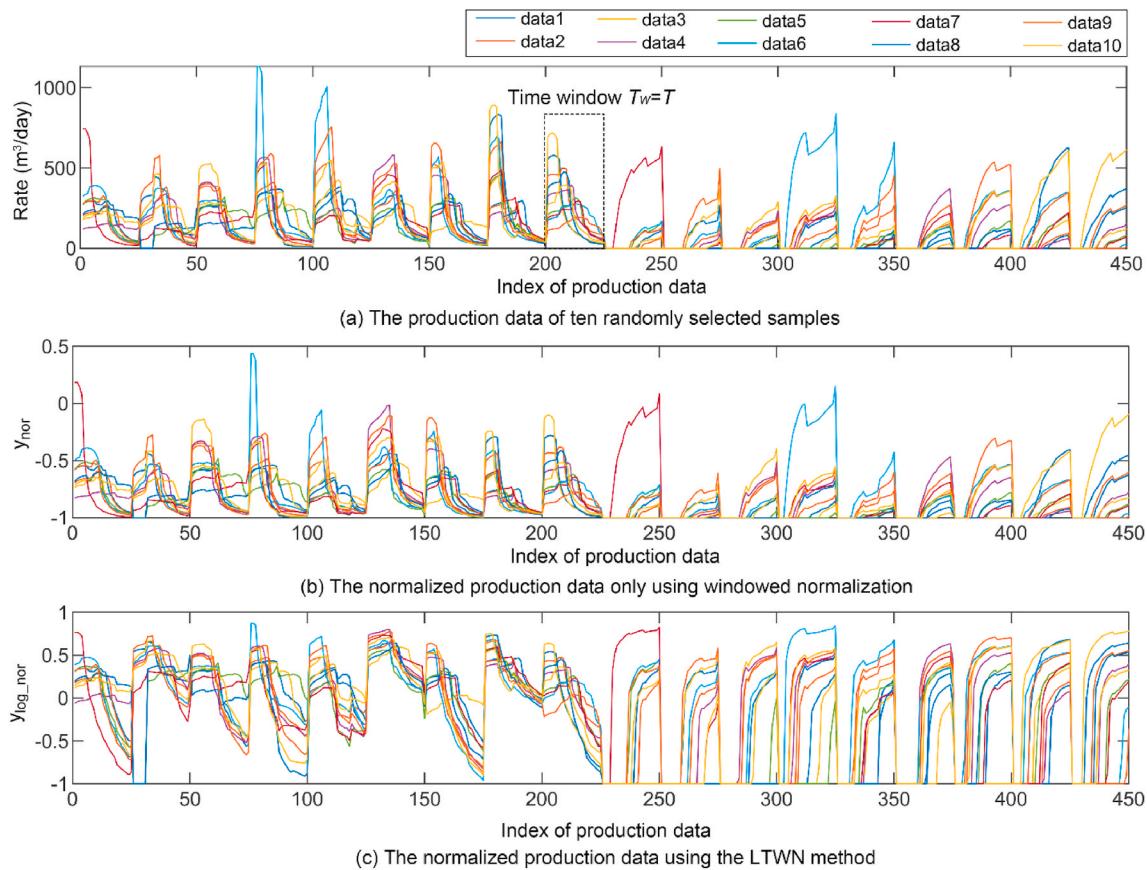


Fig. 3. The illustration of the characteristics of production data and the effect of the LTWN method.

Specifically, we first perform the log-transformation of the production data:

$$y_{\log} = \log(y + 1) \quad (8)$$

In general, the production data are greater than or equal to 0, so the constant 1 is added to avoid numeric errors. Log-transformation can make the distribution of data more distinctive, and can make the characteristics of production data more obvious when the inflection point. Then, the production data is segmented in timeline with N_w windows and the window size is T_w . This means that T is equal to $N_w \cdot T_w$.

$$y_{\log} = \left\{ y_{\log}^1, y_{\log}^2, \dots, y_{\log}^{N_{obs} \times N_w \times T_w} \right\} \quad (9)$$

As shown in Fig. 3a, the black dotted box represents a window in timeline. The window size $T_w \leq T$. In this study, we set the size of the time window T_w equal to T . Dividing time-window can keep the trend characteristics of time series data on the timeline. Then the maximum and minimum normalization method is used for each segment of data:

$$y_{\log_nor}^i = 2 \times \left[\frac{y_{\log}^i - \min(y_{\log}^i)}{\max(y_{\log}^i) - \min(y_{\log}^i)} \right] - 1 \quad (10)$$

After the normalization, the production data range from -1 to 1 , which is consistent with the output of the tanh activation function.

By comparing Fig. 3a and b, it can be seen that the change trend of the original production data can be retained by windowing normalization. However, we can also see that most of the production data are squeezed together due to similarities and confusions. In terms of values, most data are concentrated below zero after normalization (Fig. 3b). This will lead to a wrong bias when training the network model. Fig. 3c shows that the data are effectively distinguished by using the log-transformation, which is conducive to the learning of network model.

3. Surrogate-based history matching workflow

Fig. 4 shows the schematic diagram of surrogate-based history matching workflow. It can be seen that the MLGRU surrogate model replaces the process of re-parameterization and numerical simulation in history matching of geological properties. For large-scale reservoirs, this avoids the need to reconstruct parameter fields from hundreds of thousands to tens of millions of grids during the solving of history matching. Meanwhile, there is no need to run the time-consuming numerical simulation during the solving process, which can completely release the search ability of the sampling algorithm. We use the MEDA algorithm for sampling, which has the ability of distributed search to locate multiple solutions in a single run. In the following, we briefly introduce the parameterization method and MEDA algorithm used in this paper.

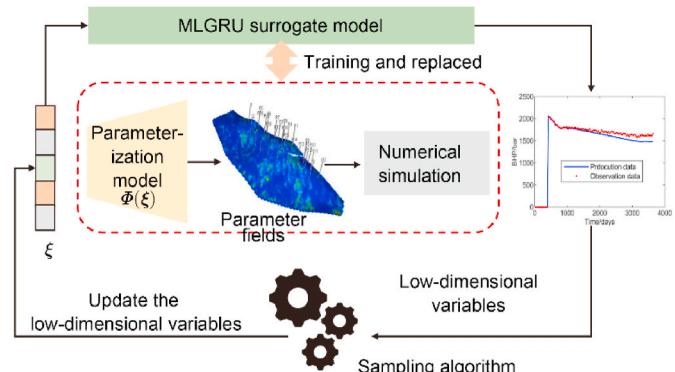


Fig. 4. The proposed surrogate-based history matching method.

3.1. Objective function and parameterization

Spatially varying geological properties are usually discretized into grid-based parameter fields, which are high-dimensional and difficult to update directly. Therefore, the re-parameterization methods have been widely studied and used to reduce the dimension and the parameter fields can be represented as a low-dimensional vector. In this study, principal component analysis (PCA) (Sarma et al., 2008; Ma et al., 2022) method is adopted to reduce the dimension of geological fields.

PCA is a multivariate statistical analysis method. Given N_r prior realizations and assembled into a decentralized matrix:

$$M_c = \frac{1}{\sqrt{N_r - 1}} [m_1 - \bar{m} \quad m_2 - \bar{m} \quad \dots \quad m_{N_r} - \bar{m}] \quad (11)$$

where $M_c \in R^{N_m \times N_r}$, $m_i \in R^{N_m}$ is the i th prior realization, and $\bar{m} \in R^{N_m \times 1}$ is the arithmetic mean of all prior realizations.

To solve the eigenvectors and eigenvalues of the covariance matrix of M_c , the singular-value decomposition method is adopted, which gives $M_c = U\Sigma V^T$, where $U \in R^{N_m \times N_m}$ and $V \in R^{N_r \times N_r}$ are the left and right singular matrix, respectively. The elements of diagonal matrix $\Sigma \in R^{N_m \times N_r}$ are the square root of the eigenvalues. Then the new realizations can be generated by using PCA

$$m^{pca} = U_l \Sigma_l \xi + \bar{m} \quad (12)$$

where $U_l \in R^{N_m \times l}$ is the first l columns of U , the diagonal matrix $\Sigma_l \in R^{l \times l}$ containing the first l largest singular-values. The $\xi \in R^{l \times 1}$ is the low-dimensional representation vector of m^{pca} , which drawn from the Gaussian distribution.

Fig. 5 shows the eigen-image and explained variance of the first 15 principal component feature vectors (PC_1, \dots, PC_{15}) in U_l (the PCA results of case 1 in section 4.1). PC_i is the i th column of U_l . It can be seen that the PC_1 can explain 12.169% variance of the total prior realizations. And each PC represents a corresponding geological feature. The m^{pca} is a linear combination of principal component vectors, ξ can be seen as a weight coefficient. The U_l and Σ_l are both deterministic in PCA, so the low-dimensional vector ξ can represent the feature of the parameter field.

The determination of principal component eigenvectors generally adopts the energy criterion or simply uses the first N_r eigenvectors in U . In addition, the sensitivity analysis methods (Nossent et al., 2011; Park et al., 2016; Yin et al., 2019) can also be used to select a subset of U as

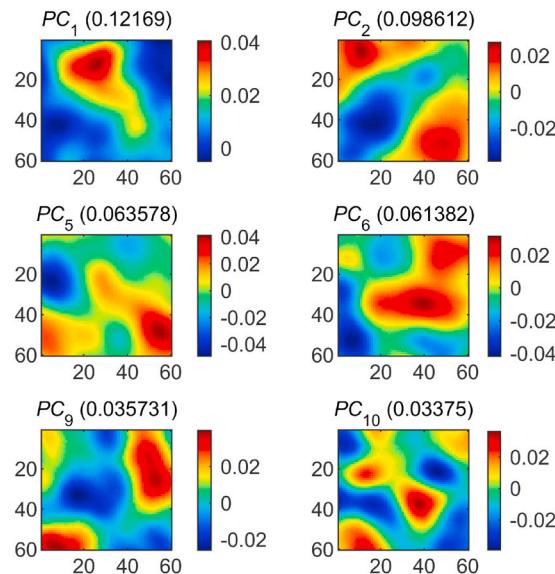


Fig. 5. The illustration of PCA eigen-image and explained variance.

the U_l .

After using the PCA parameterization, the history-matching objective function can be defined as follows:

$$O(\xi) = \frac{1}{2} [y_{(\xi)} - d_{obs}]^T C_D^{-1} [y_{(\xi)} - d_{obs}] + \frac{1}{2} \xi^T \xi \quad (13)$$

where $y_{(\xi)}$ is the production data of the corresponding realization generated by ξ , d_{obs} is the observation data, C_D is the covariance matrix of measurement error.

3.2. Generation of training samples for MLGRU

To construct the training samples, N_r prior realizations $\{m_1, m_2, \dots, m_{N_r}\}$ should be firstly generated based on geological uncertainty information and combined with geostatistical modeling method. Generally, the prior realizations are generated by using the data obtained from seismic, logging, core, etc., which can refer to previous works (Pollack et al., 2021; Yin et al., 2020). In this work, the Stanford Geostatistical Modeling Software (SGeMS) (Remy, 2005) is adopted to generate the prior realizations. Afterward, the PCA method is used to reparametrize the uncertain variable based on prior realizations.

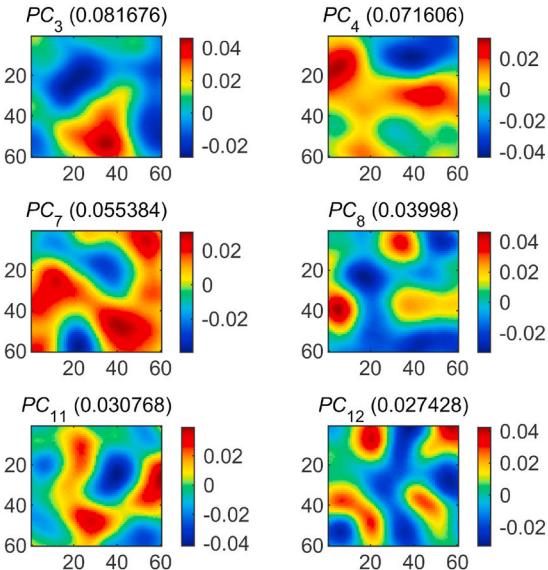
To generate N training samples for MLGRU model, we need sample N random vectors $\{\xi_1, \xi_2, \dots, \xi_N\}$ from the normal distribution, and we can generate the corresponding PCA realizations $\{m_1^{pca}, m_2^{pca}, \dots, m_N^{pca}\}$ according to Eq. (12). For these PCA realizations, we run the numerical simulation and obtain the corresponding production data $\{y_1, y_2, \dots, y_N\}$.

Note that we use the PCA realizations to run the numerical simulation to build the training samples rather than the prior realizations of geostatistical modeling. This is consistent with conventional history matching based on PCA reparameterization.

3.3. MEDA algorithm

MEDA algorithm is a population-based heuristic algorithm, which combines niching technique to solve multimodal optimization problems. The MEDA algorithm includes dynamic crowding clustering (DCC), distribution estimation, offspring generation and crowding selection mechanism. Compared with the work of Yang et al. (2017) and Ma et al. (2022), we make some minor modifications to the MEDA algorithm. In the following, we describe each part of the algorithm in detail.

The DCC is a classic niching method, which can divide a population into multiple subpopulations. As shown in Fig. 6, the DCC method firstly



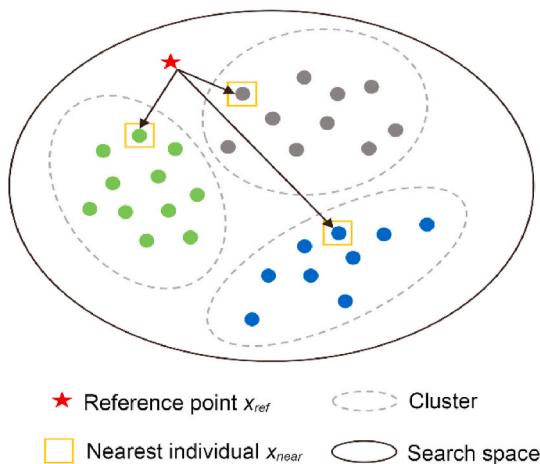


Fig. 6. Schematic diagram of dynamic crowding clustering.

randomly generates a solution in the search space as the reference point x_{ref} , then finding the solution x_{near} which nearest to the x_{ref} in the population, and searching M solutions closest to x_{near} to form a subpopulation. Afterward, removing the founded subpopulation from the population and repeat the process until the population is empty. M is the size of the population, which is dynamically determined during the search. In this work, the M is a randomly integer generated at each iteration from a uniform distribution [5, 10] as suggested in (Yang et al., 2017). To find the nearest solution, Euclidean distance is used in DCC.

After dividing the population into multiple subpopulations by using DCC, a distributed estimate was made for each subpopulation the distribution for each subpopulation can be estimated. In MEDA, the Gaussian distribution is adopted as follows:

$$\mu_i^d = \frac{1}{M} \sum_{j=1}^M x_j^d \quad (14)$$

$$\sigma_i^d = \sqrt{\frac{1}{M} \sum_{j=1}^M (x_j^d - \mu_i^d)^2} \quad (15)$$

where $\mu_i = [\mu_i^1, \dots, \mu_i^d, \dots, \mu_i^D]$ and $\sigma_i = [\sigma_i^1, \dots, \sigma_i^d, \dots, \sigma_i^D]$ are, respectively, the mean and standard deviation of the i th subpopulation, $x_j = [x_j^1, \dots, x_j^d, \dots, x_j^D]$ is the j th individual in the i th niche, D is the dimension of the problem.

Based on the mean and standard deviation determined for each subpopulation, offspring solutions can be generated by random sampling using the Gaussian distribution as follows:

$$U_i^t = \text{Gaussian}(\mu_i^t, \sigma_i^t, M) \quad (16)$$

After the generation of offspring solutions, crowding selection method is used for updating the population. The crowding selection method is to select the nearest solution from the population for each offspring solution, and reserving the solution with better objective function value into the new population. The cosine distance is used in the crowding selection.

4. Case studies

In this section, we conducted two case studies on waterflooding reservoir models to verify the effectiveness of the proposed surrogate model and history matching workflow. The first case is a synthetic 2D model with Gaussian random field, while the second case is tested on the large-scale 3D Brugge benchmark model. All experiments are implemented on a Nvidia Tesla V100 GPU with 32 Gb memory, the epoch and batch size are set as 100 and 16, respectively. The neural network

models in this work are built based on the TensorFlow (Abadi et al., 2016).

4.1. Case 1: 2D Gaussian random field model

Fig. 7 depicts the true log permeability field and well placement of case 1. The reservoir area is discretized into $60 \times 60 \times 1$ grid-blocks, each $10 \times 10 \times 4$ m in size. The model has 4 water injectors and 9 producers. All wells are controlled to achieve the target BHP. In the historical stage, 25 time-steps are simulated and each time-step interval is 30 days. Observation data consist of the oil production rate (OPR) and the water production rate (WPR) of all producers, which are generated by adding Gaussian perturbation with zero mean to simulation data of the true model. The standard deviation of the perturbation is set to 5% of simulation data. Therefore, the production data in history matching has 25 time-steps and 18 features (OPR and WPR of nine producers). The prior realizations consist of 100 random log-permeability fields generated using the SGEMS without hard data. The log-permeability field can be re-parameterized by a 100D feature vector using the PCA method. This dimension is equal to the number of prior realizations.

4.1.1. Performance assessment of MLGRU

The performance of the proposed surrogate model is studied before history matching. In this experiment, the dataset includes 2500 random PCA realizations of permeability and the corresponding simulation data. The dataset is divided into training set, validation set and test set according to the ratio of 0.8: 0.1: 0.1 (2000: 250: 250). First, we use the training set and validation set to select the hyperparameters of the model. In MLGRU, the hyperparameters need to be determined include the number of GRU layers N_{layer} and the number of neurons N_{unit} in each GRU layer. For convenience, we set all GRU layers to use the same number of neurons. Therefore, there are two hyperparameters to be determined, and we set 9 combinations of hyperparameters as shown in Table 1.

The performance indicator R^2 -score of the MLGRU model with different hyperparameter combination is evaluated on the validation samples, as shown in Fig. 8. We can obtain the following observations as for the performance of the model with different hyperparameters combinations. First, it can be seen that the prediction performance of the model increases as the number of GRU layers increases. For example, the combinations 1, 4 and 7 all have the same GRU neurons, but the number of GRU layers increases from 1 to 3. The combination 1 uses single-layer GRU, which has a great difference in the prediction performance on the verification samples, while the combination 7 uses 3-layer GRU, which shows that the prediction performance on the verification samples is more concentrated. These results indicate that the use of multi-layer GRU can improve the robustness of recurrent networks. In addition, for MLGRU with same number of layers but with different numbers of GRU neurons (such as combination 1, 2, and 3), it can be seen that

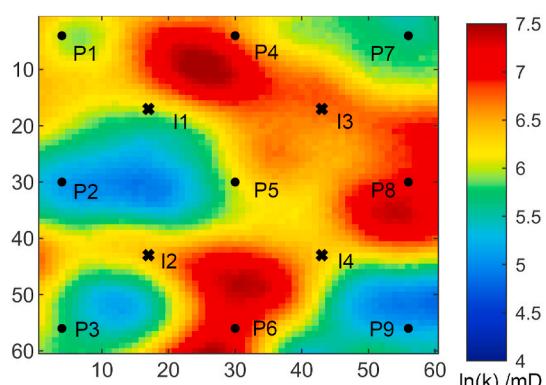
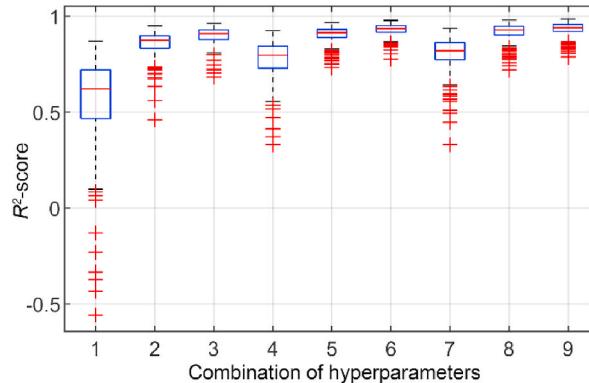


Fig. 7. The true log-permeability field for Case 1.

Table 1

Nine different combination of hyperparameters.

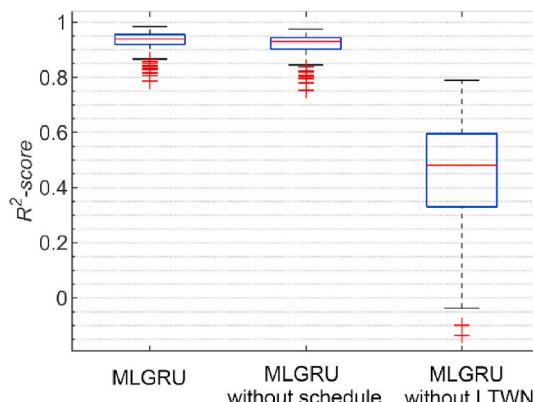
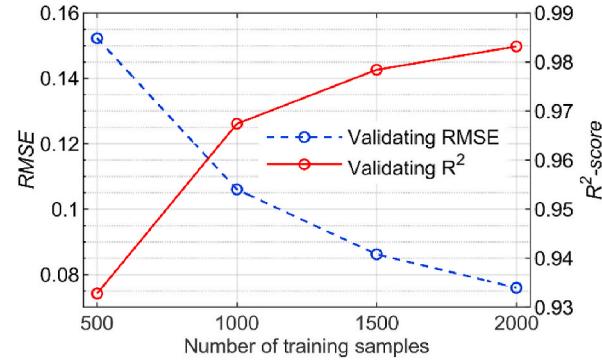
Combination	1	2	3	4	5	6	7	8	9
N_{layer}	1	1	1	2	2	2	3	3	3
N_{unit}	100	500	1000	100	500	1000	100	500	1000

**Fig. 8.** Comparison of the validated performance of different hyperparameters combination.

increasing the number of neurons can significantly improve the prediction performance of the model. Finally, the optimal model hyperparameter combination can be determined as the combination 9, which will be adopted in following experiments. For the MLGRU model with this hyperparameter combination, the total number of trainable parameters is 153,75,018, and the training time is less than 1 s for each epoch.

Fig. 9 shows the results of a comparative experiment to illustrate the effect of using wells schedule as the auxiliary input and the proposed LTWN method on the prediction performance of the model. It can be seen that the use of well control parameters can slightly improve the prediction performance. However, the performance of the model without the proposed LTWN method is significantly reduced. Specifically, we used the windowed normalization but not log-transformation. This result indicate that the log-transformation can significantly distinguish the features of production data corresponding to different realizations.

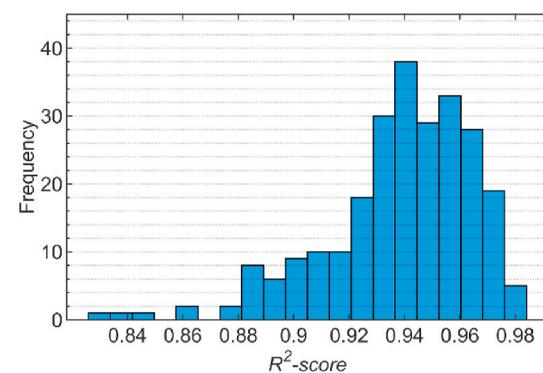
Fig. 10 shows the comparison of prediction performance on validation samples when using different training sample sizes. It can be seen that the prediction performance of the model increases with the increase of the number of training samples. Additionally, the growing rate of model performance decreases with the increase of sample size.

**Fig. 9.** Performance comparison of MLGRU, MLGRU without using schedule and MLGRU without using LTWN.**Fig. 10.** Comparison of MLGRU using different training sample size.

4.1.2. Results of history matching

Based on the above experiments and analysis, we re-train the MLGRU surrogate model using the 9th hyperparameters combination in Table 1 with 500 training samples and apply it to the history matching. Fig. 11 shows the histogram of the R^2 -score of the trained model on the test samples. It can be seen that the R^2 -score of most test samples is between 0.92 and 0.98. Then the test sample (Fig. 12) of P50 of R^2 -score is selected to compare the predicted production data with the numerical simulation results, as shown in Fig. 13. As we can see from this figure, the black line is the simulation results while the red line with circle represents the surrogate prediction. It can be seen that the prediction results can be consistent with the simulation data. In particular, the breakthrough of WPR can be well predicted.

Afterward, we perform the history matching based on the trained MLGRU model. In this experiment, the population size and the number of iterations of MEDA are set to 200 and 100, respectively, which means a total of 20,000 function evaluations are required. Fig. 14 compares the convergence of surrogate-based MEDA with that of simulation-based MEDA. It can be seen that the convergence trend of the surrogate-based MEDA is roughly consistent with that of simulation-based MGRU. In Fig. 15, we compare the distribution of the first 15 variables of the final population obtained by simulation-based MEDA and surrogate-based MEDA, respectively. And we calculated the low-dimensional vector of the true log-permeability as the reference (black dotted line). It can be seen that the distribution of the solutions obtained by these two methods is similar. And the obtained posterior distribution

**Fig. 11.** The distribution histogram of R^2 for all test samples (Case 1).

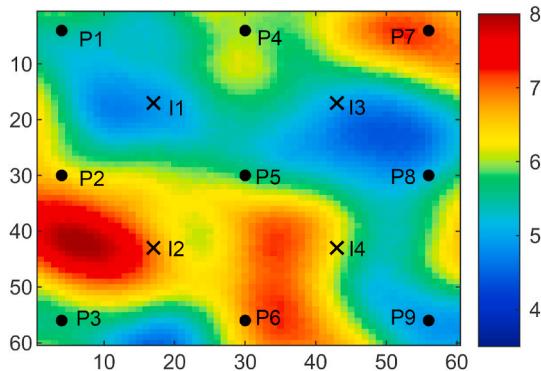


Fig. 12. The log-permeability field corresponding to the P50 test sample.

can basically cover the low-dimensional vector of the true log-permeability. In addition, as shown in Fig. 16, compared with the objective function value of initial solutions, the objective function value of the updated solutions obtained by the two methods decreases significantly. Compared with the solution obtained by simulation-based MEDA, the objective function value of solutions obtained by surrogate-based MEDA is smaller. Combined with Fig. 15, it can also be seen that the solutions obtained by surrogate-based MEDA are more concentrated. On the one hand, the search of MEDA algorithm has randomness, on the other hand, the surrogate model may filter out some poor solutions.

The above results show that the MLGRU model can effectively

replace the numerical simulation during the optimization process. Compared with 20,000 numerical simulations required for simulation-based MEDA, surrogate-based MEDA requires only 500 numerical simulations to train the surrogate model. Training such a surrogate model only takes about 3 min, so using the proposed MLGRU model can significantly reduce the computational cost. In addition, there is no need to reconstruct the parameter fields during the optimization process of surrogate-MEDA, which can also reduce the computational requirements.

Fig. 17 shows the fitting and prediction effect of production data from four producers after history matching. The green lines represent the production data of the history-matched models obtained by surrogate-based MEDA. Compared with the prior models (grey lines), the uncertainty of production data corresponding to the history-matched model decreases significantly. In addition, it can be seen that the simulation data can well cover the real data (red line) in the forecast stage.

Finally, Fig. 18 depicts eight selected posterior realizations obtained by surrogate-based MEDA. It can be seen that the obtained posterior realizations have significant diversity. By comparing the posterior mean field with the reference field, it can be seen that the posterior field can capture the distribution of high-permeability and low-permeability regions in the true model. Compared with the prior standard deviation, it can be seen that the posterior standard deviation decreases significantly after history matching.

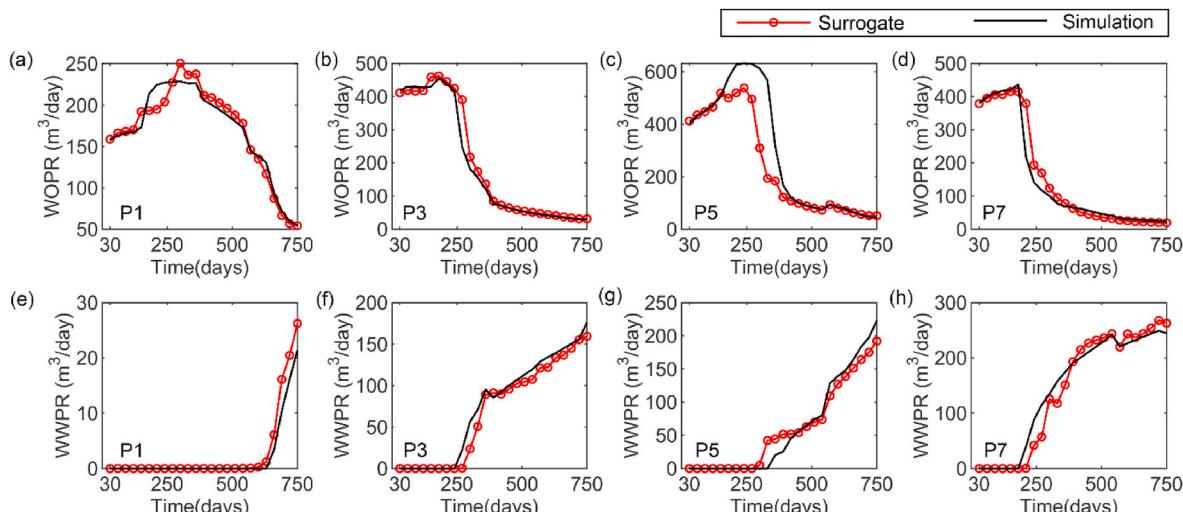


Fig. 13. Comparison of well rates from the numerical simulator (black lines) and surrogate model (red dots and lines) for the P50 test samples (Case 1). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

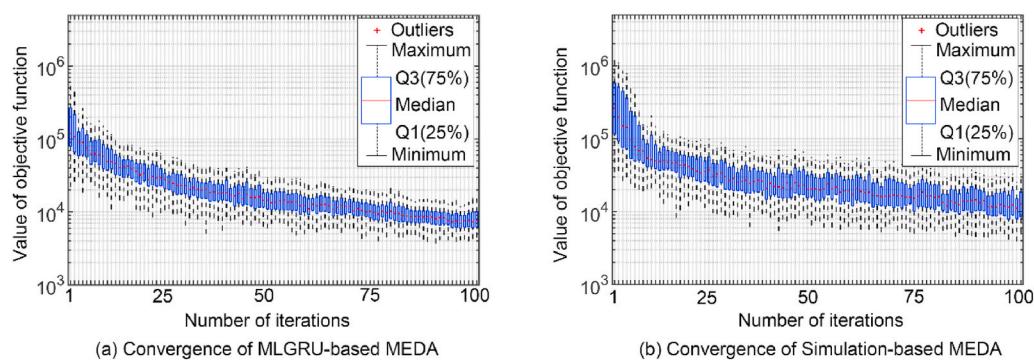


Fig. 14. Comparison of convergence of (a) MLGRU-based MEDA and (b) simulation-based MEDA.

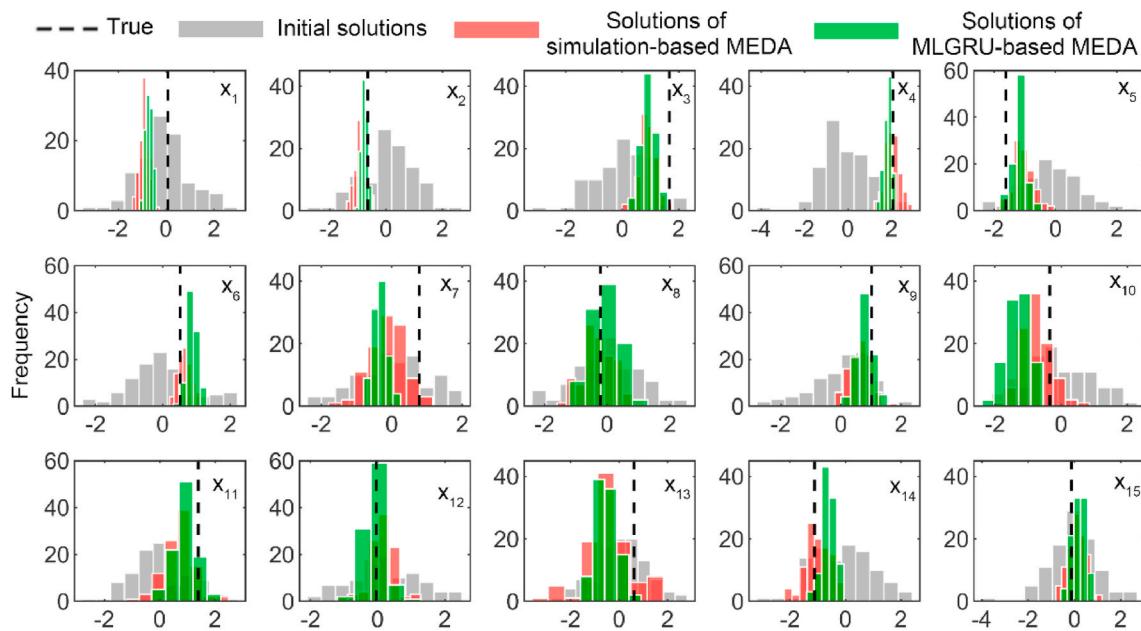


Fig. 15. Comparison of the distribution of the first 15 variables of the low-dimensional vector.

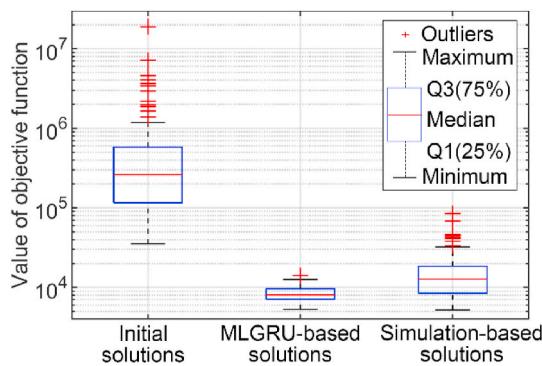


Fig. 16. Comparison of the final updated population of MLGRU-based MEDA and simulation-based MEDA and the objective function value of the initial solutions.

4.2. Case 2: Brugge model

In this case, the surrogate-based MEDA history matching workflow is applied to Brugge benchmark model (Peters et al. 2013), a large-scale 3D waterflooding reservoir model. Brugge model has $138 \times 49 \times 9$ grid blocks (44,550 active cells), 10 injection wells, and 20 production wells, as illustrated in Fig. 19. There are five geological properties, e.g., the permeability in X, Y, and Z directions, the porosity, and the net-to-gross, need to be adjusted. In the historical stage, the ten years of observation data include WOPR and WWPR of all producers, and the BHP of all injectors and producers. And there are 120 time-steps. The injectors are controlled to achieve the target injection rate at 4000 STB/day, while the producers will be shut if the water cut more than 0.9. The standard deviation of Gaussian noise is set to 3% for the production rate and 7.25 psi for BHP.

There are 104 different prior realizations in the Brugge dataset, which can be used for the PCA re-parameterization. The more details about PCA of Brugge case can be referred to Wang et al. After PCA, the five uncertain geological properties can be represented by a 104-D

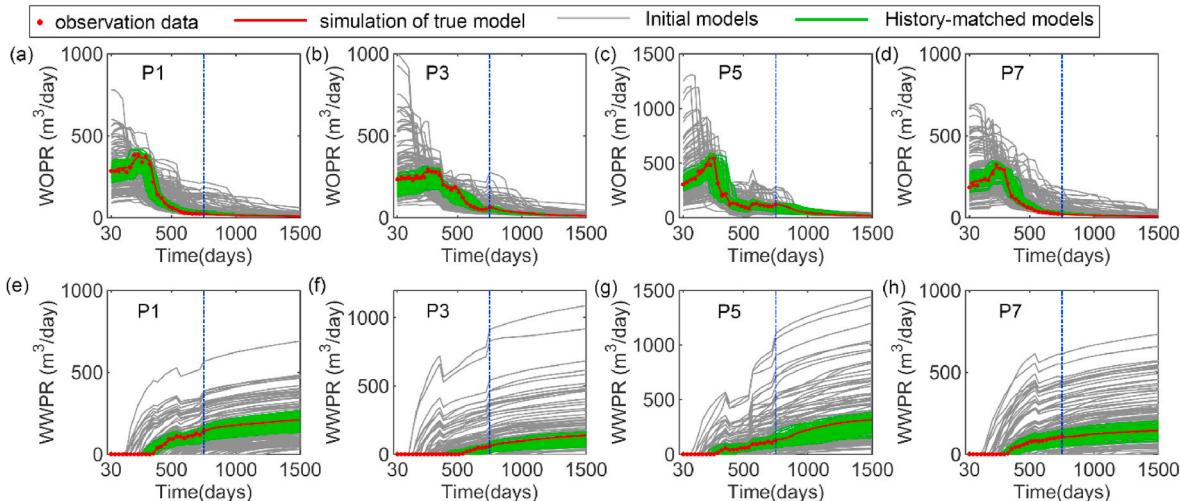


Fig. 17. The matching results of production data of surrogate-based MEDA.

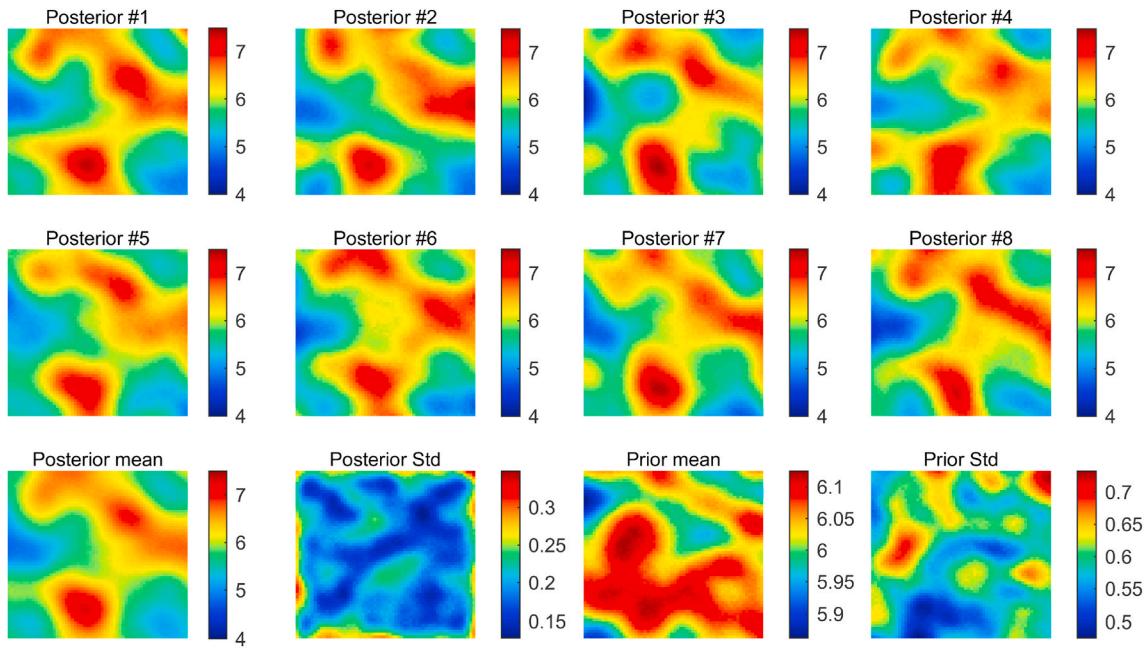


Fig. 18. Reference log-conductivity field, 8 posterior realizations, and mean and variance fields obtained from the MLGRU-based MEDA. (std stands for standard deviation).

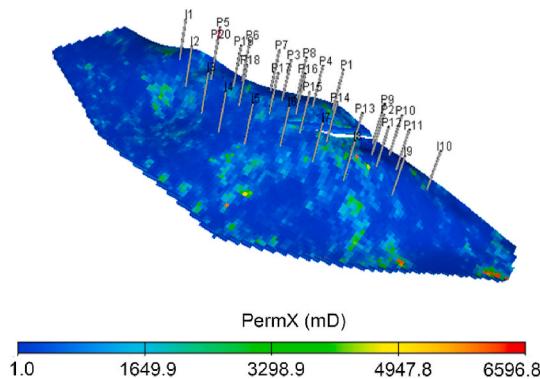


Fig. 19. The true log-permeability of X direction and well pattern for the Brugge case.

vector. Then we generate 1250 random PCA realizations and the corresponding simulation data can be used as the dataset. The dataset can be divided to 1000 training samples and 250 test samples. The hyper-parameters of MLGRU for this case also use the 9th combination in Table 1. For this case, each epoch takes about 4 s, 100 epochs take about 6.7 min. Fig. 20 depicts the convergence of the loss function during the training. It can be seen that the loss function drops rapidly in the early stages and reaches a plateau before 100 epochs.

Fig. 21 shows the histogram of performance distribution of the trained model on the test samples. It can be seen that the R^2 -score of most test samples is around 0.96, and the R^2 -score of a few samples is less than 0.9. The R^2 -score of P50 test sample is 0.9559. Fig. 22 shows the comparison between the prediction data and the numerical simulation result on the P50 test sample. It can be seen that the predicted BHP of all injection wells is almost consistent with the simulation results. The BHP is also well predicted for most production wells except for a few. It is worth noting that the WWPR of all producers obtain a good prediction.

Afterward, we implement the surrogate-based history matching for this case. Fig. 23 compares the objective function value of the 104 prior models with that of the solutions obtained by surrogate-based MEDA. The objective function values of the solutions obtained by surrogate-

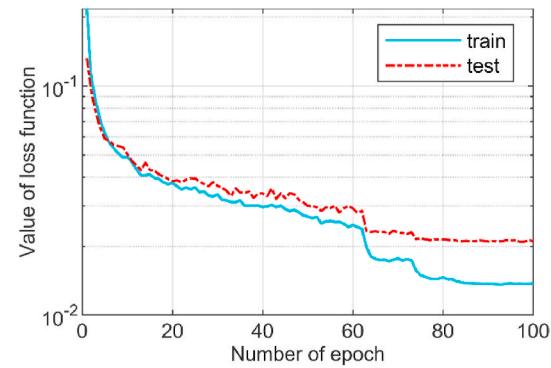


Fig. 20. The change of loss function during the training of MLGRU for Brugge case.

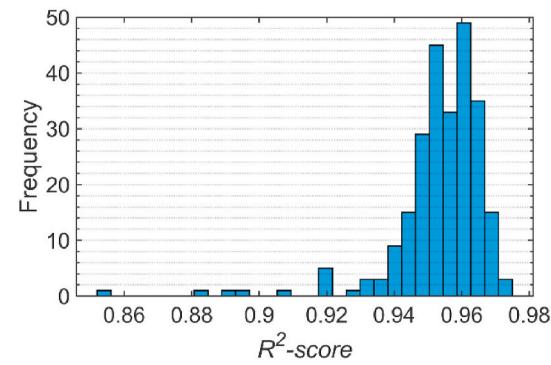


Fig. 21. The histogram of R^2 distribution of test samples.

based MEDA are calculated by using numerical simulations. It can be seen that the objective function value of most posterior solutions is significantly smaller than that of prior models. However, due to the existence of proxy error, there are still some solutions whose objective function value is obviously larger. Therefore, we choose a critical value

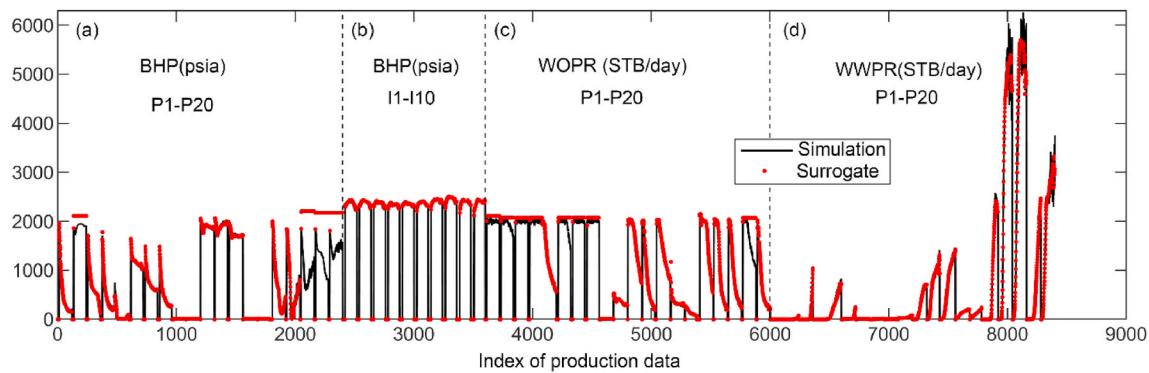


Fig. 22. Comparison of the prediction result and simulation data of the P50 test sample.

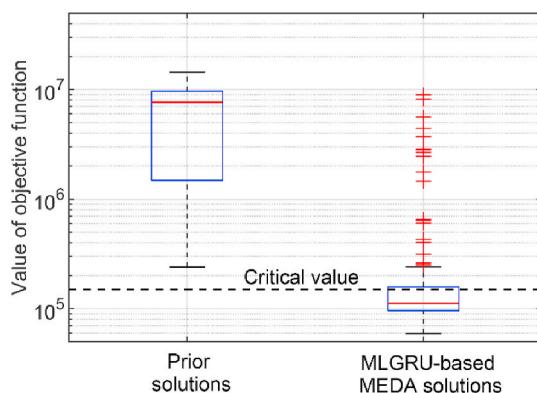


Fig. 23. Comparison of the objective function value of the prior realizations with that of the solutions obtained by MLGRU-based MEDA.

to filter the posterior solutions (the black dashed line in Fig. 23). The critical value of objective function is selected according to the fitting effect of production data, which is a commonly used method in surrogate-based history matching (Li et al., 2019). Finally, 69 posterior solutions are obtained after this filtering.

Fig. 24 compares the production data of the obtained posterior realizations (green lines) and that of prior models (grey lines), while the red dots are observations. It can be seen that the uncertainty of BHP data for all wells decreased significantly after history matching. The uncertainty of WOPR is also reduced for producers. The WWPR of most producers is well fitted, but the fitting of a few wells is poor due to the oscillation in the late production period. Overall, most wells obtain an effective fitting of production data.

Fig. 25 shows the prediction of X-direction permeability in layers 1, 3, 5, and 7. In this figure, from left to right, each column is the reference

field, prior mean, prior standard deviation, posterior mean field, and posterior standard deviation, respectively. It can be seen that the posterior mean field can capture the distribution of high- and low-permeability in area with wells. Compared with the prior standard deviation, it can be seen that the posterior standard deviation decreases significantly after history matching. As can be seen from the standard deviation, the uncertain of posterior realizations is larger in the region without wells.

5. Conclusions

In this study, we propose a deep-learning-based MLGRU surrogate model under a vector-to-sequence modeling framework. A stacked RNN structure is developed to approximate the mapping from feature vector of geological realizations to the production data. The proposed MLGRU model is lightweight for model parameters, which does not need to deal with high-dimensional spatial data. In addition, we design the LTWN normalization method for the production data, which can promote the learning ability of production data. The proposed surrogate model is incorporated into the MEDA algorithm to formulate a history matching framework. In the first 2D Gaussian random field model, numerical experiments verify that the multilayer structure can make the prediction more robust, and increasing the number of neurons in GRU can effectively improve learning ability. Through the numerical experiments on the large-scale 3D Brugge reservoir, it is proved that the proposed method can effectively reduce the computational cost and obtain satisfactory inversion results. In future work, the proposed surrogate model and history matching method need to be studied in reservoirs with more complex geological characteristics.

Author contributions

Xiaopeng Ma: Conceptualization, Methodology, Software, Writing –

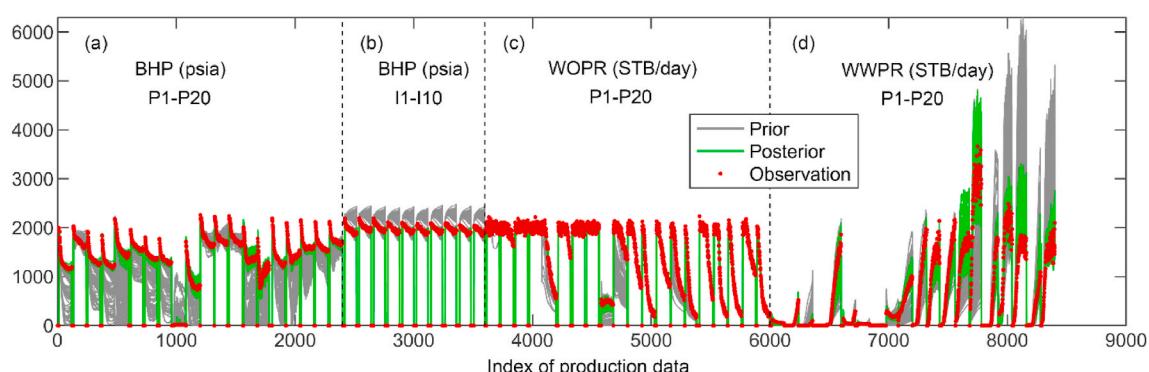


Fig. 24. The history matching results of production data for the Brugge case.

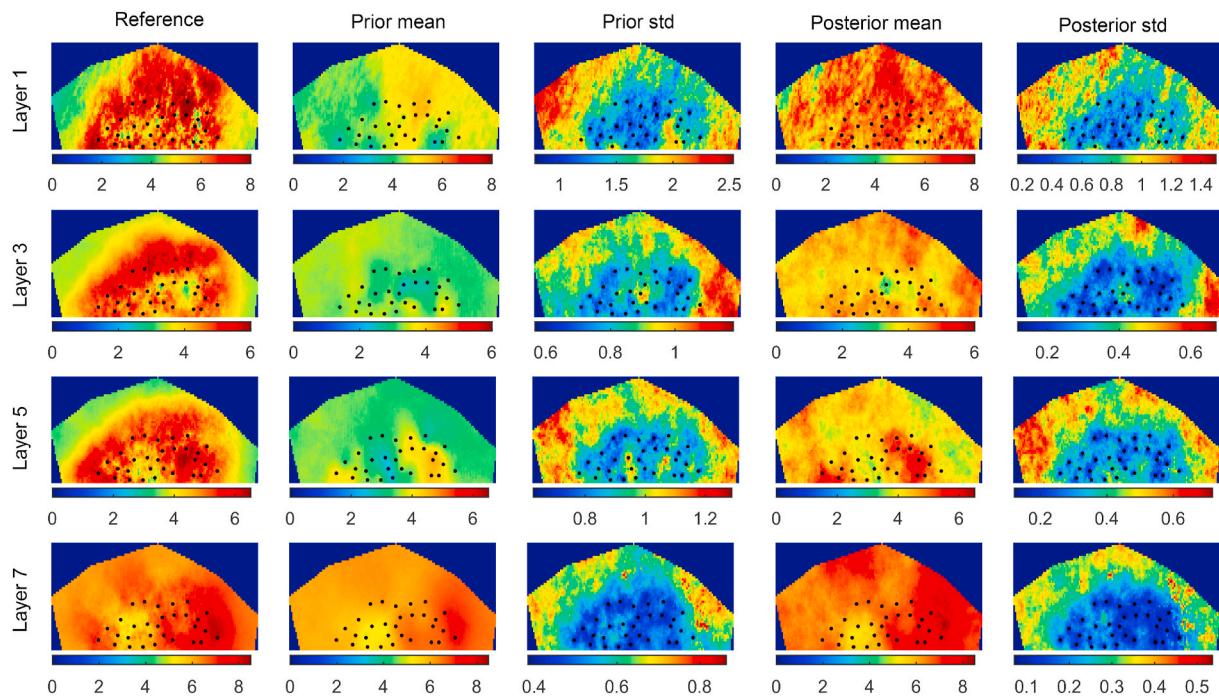


Fig. 25. The history matching results of X-direction log-permeability of 4 selected layers. (std stands for standard deviation).

original draft, Writing – review & editing. Kai Zhang: Methodology, Funding acquisition, Supervision, Project administration. Hanjun Zhao: Validation, Resources. Liming Zhang: Data curation, Resources. Jian Wang: Data curation, Resources. Huqing Zhang: Software, Validation. Piayang Liu: Visualization, Validation. Xia Yan: Visualization, Validation. Yongfei Yang: Software, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is supported by the National Natural Science Foundation of China under Grant 51722406, 52074340, and 51874335, the Shandong Provincial Natural Science Foundation under Grant JQ201808, The Fundamental Research Funds for the Central Universities under Grant 18CX02097A, the Major Scientific and Technological Projects of CNPC under Grant ZD 2019-183-008, the Science and Technology Support Plan for Youth Innovation of University in Shandong Province under Grant 2019KJH002, the National Science and Technology Major Project of China under Grant 2016ZX05025001-006, 111 Project under Grant B08028.

References

- Abadi, Martín, Agarwal, Ashish, Barham, Paul, et al., 2016. Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv preprint arXiv: 160304467*.
- Chen, Yuntian, Zhang, Dongxiao, 2020. Well log generation via ensemble long short-term memory (EnLSTM) network. *Geophys. Res. Lett.* 47 (23), e2020GL087685 <https://doi.org/10.1029/2020GL087685>.
- Chen, Chaohui, Gao, Guohua, Li, Ruijian, et al., 2018. Global-search distributed-gauss-Newton optimization method and its integration with the randomized-maximum-likelihood method for uncertainty quantification of reservoir performance. *SPE J.* 23 (5), 1496–1517. <https://doi.org/10.2118/182639-PA>.
- Cho, Kyunghyun, van Merriënboer, Bart, Bahdanau, Dzmitry, et al., 2014a. On the properties of neural machine translation: encoder-decoder approaches. In: Proc. Proceedings of Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, pp. 103–111.
- Cho, Kyunghyun, Van Merriënboer, Bart, Gulcehre, Caglar, et al., 2014b. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv preprint arXiv:14061078*.
- Chung, Junyoung, Gulcehre, Caglar, Cho, Kyunghyun, et al., 2015. Gated feedback recurrent neural networks. In: Paper Presented at the Proceedings of the 32nd International Conference on International Conference on Machine Learning, vol. 37 (Lille, France).
- Donahue, J., Hendricks, L.A., Rohrbach, M., et al., 2017. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (4), 677–691.
- Fan, Dongyan, Sun, Hai, Yao, Jun, et al., 2021. Well production forecasting based on ARIMA-LSTM model considering manual operations. *Energy* 220, 119708. <https://www.sciencedirect.com/science/article/pii/S0360544220328152>.
- Gao, Liqiang, Li, Haibo, Liu, Zhijian, et al., 2021. RNN-transducer based Chinese sign language recognition. *Neurocomputing* 434, 45–54. <https://www.sciencedirect.com/science/article/pii/S0925231220318877>.
- Graves, A., Jaitly, N., Mohamed, A., 2013. Hybrid speech recognition with deep bidirectional LSTM. In: Proc., 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, 8–12 Dec. 2013, pp. 273–278.
- He, X., Deng, L., 2017. Deep learning for image-to-text generation: a technical overview. *IEEE Signal Process. Mag.* 34 (6), 109–116.
- Hochreiter, Sepp, Schmidhuber, Jürgen, 1997. Long short-term memory. *Neural Comput.* 9 (8), 1735–1780.
- Kingma, Diederik P., Ba, Jimmy, 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:14126980*.
- Li, Boxiao, Bhark, Eric W., Gross, Stephen, et al., 2019. Best practices of assisted history matching using design of experiments. *SPE J.* 24 (4), 1435–1451. <https://doi.org/10.2118/191699-PA>.
- Li, Xuechen, Ma, Xinfang, Xiao, Fengchao, et al., 2022. Time-series production forecasting method based on the integration of bidirectional gated recurrent unit (Bi-GRU) network and sparrow search algorithm (SSA). *J. Petrol. Sci. Eng.* 208, 109309. <https://www.sciencedirect.com/science/article/pii/S0920410521009621>.
- Ma, Xiaopeng, Zhang, Kai, Wang, Jian, et al., 2021a. An efficient spatial-temporal convolutional recurrent neural network surrogate model for history matching. *SPE J.* 1–16. <https://doi.org/10.2118/208604-PA>.
- Ma, Xiaopeng, Zhang, Kai, Zhang, Liming, et al., 2021b. Data-driven niching differential evolution with adaptive parameters control for history matching and uncertainty quantification. *SPE J.* 1–18. <https://doi.org/10.2118/205014-PA>.
- Ma, Xiaopeng, Zhang, Kai, Zhang, Jinding, et al., 2022. A novel hybrid recurrent convolutional network for surrogate modeling of history matching and uncertainty quantification. *J. Petrol. Sci. Eng.* 210, 110109. <https://www.sciencedirect.com/science/article/pii/S0920410522000067>.
- Makin, Joseph G., Moses, David A., Chang, Edward F., 2020. Machine translation of cortical activity to text with an encoder-decoder framework. *Nat. Neurosci.* 23 (4), 575–582. <https://doi.org/10.1038/s41593-020-0608-8>.
- Mo, Shaoxing, Zabaras, Nicholas, Shi, Xiaoqing, et al., 2019. Deep autoregressive neural networks for high-dimensional inverse problems in groundwater contaminant source

- identification. *Water Resour. Res.* 55 (5), 3856–3881. <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018WR024638>.
- Nosset, J., Elsen, P., Bauwens, W., 2011. Sobol'sensitivity analysis of a complex environmental model. *Environ. Model. Software* 26 (12), 1515–1525.
- Oliver, Dean S., Chen, Yan, 2011. Recent progress on reservoir history matching: a review. *Comput. Geosci.* 15 (1), 185–221. <https://doi.org/10.1007/s10596-010-9194-2>.
- Park, J., Yang, G., Satija, A., et al., 2016. DGSA: a Matlab toolbox for distance-based generalized sensitivity analysis of geoscientific computer experiments. *Comput. Geosci.* 97, 15–29.
- Pollack, A., Cladouhos, T.T., Swyer, M.W., et al., 2021. Stochastic inversion of gravity, magnetic, tracer, lithology, and fault data for geologically realistic structural models: patus Geothermal Field case study. *Geothermics* 95, 102129.
- Remy, Nicolas, 2005. S-GeMS: the Stanford geostatistical modeling software: a tool for new algorithms development. In: Leuangthong, Oy, Deutsch, Clayton V. (Eds.), *Geostatistics Banff 2004*. Springer Netherlands, Dordrecht, pp. 865–871.
- Sarma, Pallav, Durlofsky, Louis J., Aziz, Khalid, 2008. Kernel principal component analysis for efficient, differentiable parameterization of multipoint geostatistics. *Math. Geosci.* 40 (1), 3–32. <https://doi.org/10.1007/s11004-007-9131-7>.
- Song, Xuanyi, Liu, Yuetian, Xue, Liang, et al., 2020. Time-series well performance prediction based on Long Short-Term Memory (LSTM) neural network model. *J. Petrol. Sci. Eng.* 186, 106682. <https://www.sciencedirect.com/science/article/pii/S0920410519311039>.
- Sun, Shiliang, Luo, Chen, Chen, Junyu, 2017. A review of natural language processing techniques for opinion mining systems. *Inf. Fusion* 36, 10–25.
- Tang, Meng, Liu, Yimin, Durlofsky, Louis J., 2020. A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems. *J. Comput. Phys.* 413, 109456. <https://www.sciencedirect.com/science/article/pii/S0021999120302308>.
- Wang, Nanzhe, Chang, Haibin, Zhang, Dongxiao, 2021. Efficient uncertainty quantification and data assimilation via theory-guided convolutional neural network. *SPE J.* 26 (6), 4128–4156. <https://doi.org/10.2118/203904-PA>.
- Xiao, Cong, Lin, Hai-Xiang, Leeuwenburgh, Olwijn, et al., 2022. Surrogate-assisted inversion for large-scale history matching: comparative study between projection-based reduced-order modeling and deep neural network. *J. Petrol. Sci. Eng.* 208, 109287. <https://www.sciencedirect.com/science/article/pii/S0920410521009402>.
- Yang, Q., Chen, W., Li, Y., et al., 2017. Multimodal estimation of distribution algorithms. *IEEE Trans. Cybern.* 47 (3), 636–650.
- Yin, Z., Feng, T., MacBeth, C., 2019. Fast assimilation of frequently acquired 4D seismic data for reservoir history matching. *Comput. Geosci.* 128, 30–40.
- Yin, Z., Sebastien, S., Jef, C., 2020. Automated Monte Carlo-based quantification and updating of geological uncertainty with borehole data (AutoBEL v1. 0). *Geosci. Model Dev. (GMD)* 13 (2), 651–672.
- Yuan-yuan, Chen, Lv, Y., Li, Z., et al., 2016. Long short-term memory model for traffic congestion prediction with online open data. In: Proc., 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 1–4 Nov. 2016, pp. 132–137.
- Zhang, Jiangjiang, Lin, Guang, Li, Weixuan, et al., 2018a. An iterative local updating ensemble smoother for estimation and uncertainty assessment of hydrologic model parameters with multimodal distributions. *Water Resour. Res.* 54 (3), 1716–1733. <https://doi.org/10.1002/2017WR020906> <https://doi.org/10.1002/2017WR020906>.
- Zhang, K., Ma, X., Li, Y., et al., 2018b. Parameter prediction of hydraulic fracture for tight reservoir based on micro-seismic and history matching. *Fractals* 26 (2), 1840009.
- Zhang, Kai, Jinding, Zhang, Xiaopeng, Ma, et al., 2021. History matching of naturally fractured reservoirs using a deep sparse autoencoder. *SPE J.* 26 (4), 1700–1721.
- Zhong, Zhi, Sun, Alexander Y., Wang, Yanyong, et al., 2020. Predicting field production rates for waterflooding using a machine learning-based proxy model. *J. Petrol. Sci. Eng.* 194, 107574. <https://www.sciencedirect.com/science/article/pii/S0920410520306422>.
- Zhu, Yiniao, Zabaras, Nicholas, 2018. Bayesian deep convolutional encoder-decoder networks for surrogate modeling and uncertainty quantification. *J. Comput. Phys.* 366, 415–447. <https://www.sciencedirect.com/science/article/pii/S0021999118302341>.