

A novel hybrid recurrent convolutional network for surrogate modeling of history matching and uncertainty quantification

Xiaopeng Ma^a, Kai Zhang^{a,b,*}, Jinding Zhang^a, Yanzhong Wang^a, Liming Zhang^a, Piyang Liu^b, Yongfei Yang^a, Jian Wang^c

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

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

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



ARTICLE INFO

Keywords:

History matching
Surrogate modeling
Convolutional neural network
Recurrent neural network

ABSTRACT

Automatic history matching (AHM) has been widely studied in petroleum engineering due to it can provide reliable numerical models for reservoir development and management. However, AHM is still a challenging problem because it usually involves running a great deal of time-consuming numerical simulations during the solving process. To address this issue, this article studies a hybrid recurrent convolutional network (HRCN) model for surrogate modeling of numerical simulation used in AHM. The HRCN model is end-to-end trainable for predicting the well production data of high-dimensional parameter fields. In HRCN, a convolutional neural network (CNN) is first developed to learn the high-level spatial feature representations of the input parameter fields. Following that, a recurrent neural network (RNN) is constructed with the purpose of modeling complex temporal dynamics and predicting the production data. In addition, given that the fluctuations of production data are influenced by well control measures, the well control parameters are used as auxiliary inputs of RNN. Moreover, the proposed surrogate model is incorporated into a multimodal estimation of distribution algorithm (MEDA) to formulate a novel surrogate-based AHM workflow. The numerical studies performed on a 2D and a 3D reservoir model illustrate the performance of the proposed surrogate model and history matching workflow. Compared with the MEDA using only numerical simulations, the surrogate-based AHM workflow significantly reduces the computational cost.

1. Introduction

In the development of oil and gas, it is necessary to establish accurate and reliable numerical models of the underground reservoir for management and optimization (Liu and Oliver 2005, Oliver and Reynolds, 2008). However, geological properties such as permeability and porosity bring nonnegligible uncertainty to the numerical model (Liao and Zhang, 2015). AHM is an effective and efficient approach to reduce the uncertainty of model parameters by correcting numerical simulation results to match historical observations (Oliver and Chen, 2011). Under the Bayesian framework, AHM can be transformed into a posterior probability sampling problem (Stuart, 2010). In the last few decades, various numerical methods have been developed to solve the history matching problem, such as ensemble-based data assimilation methods and optimization algorithms (Emerick and Reynolds, 2013; Liao et al., 2019; Ma et al., 2020). However, these numerical methods usually need

a lot of numerical simulations, and a single simulation run may spend minutes to hours of CPU time (Tang et al., 2020, Xue et al., 2022). Therefore, it is still an open question how to conduct AHM at a reasonable computational cost.

Because of non-invasive features and ease of implementation, integrating the data-driven surrogate models into AHM methods has been widely studied to tackle this problem (Saad et al., 2009, Zeng and Zhang, 2010; Li et al., 2011; Elsheikh et al., 2014). According to the construction method of surrogate models, the surrogate-based AHM methods can be divided into online and offline. The online method is to continuously sample and dynamically build the surrogate models during the solving process. Li and Lin (2015) developed an adaptive importance sampling algorithm for history matching using a mixture of polynomial chaos expansions (PCE). Chang et al. (2017) proposed a surrogate-assisted iterative ensemble smoother algorithm combining PCE and interpolation models. To gradually reduce the surrogate error, Zhang et al. (2018)

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

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

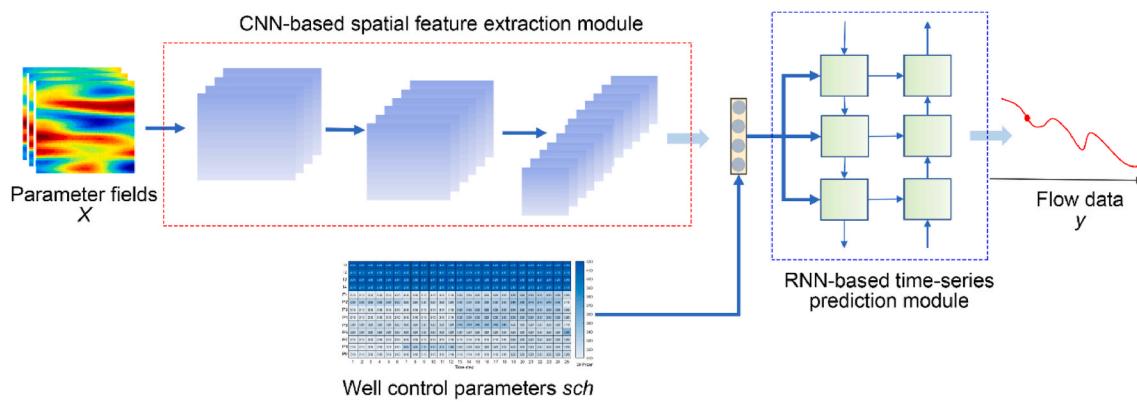


Fig. 1. Architecture of the proposed HRCN surrogate model.

designed an adaptive multi-fidelity Gaussian process (GP) model and applied it to Markov chain Monte Carlo (MCMC). Ma et al. (2021) introduced a multimodal differential evolution algorithm for history matching and used a radial basis function (RBF) model as the proxy. The surrogate models used in online methods are usually traditional machine learning models, such as PCE, GP, RBF, etc. These surrogate models have explicit mathematical representation and are easy to train. However, as the variables dimension and the nonlinear degree of the problem increase, the surrogate performance will decrease significantly. Therefore, it is often suitable for approximating local regions in the search space, which requires dynamic sampling and sophisticated management.

The offline method is to construct a surrogate model that is accurate enough before the AHM. This surrogate model can completely replace the role of numerical simulation and approximate the entire search space. Most of the previous studies (Bhark and Dehghani, 2014; Dachanuwattana et al., 2018; Li et al., 2019) used sensitivity analysis methods to select a few key parameters, then used experimental design methods to sample, and with the help of traditional machine learning methods to build the surrogate model. However, for history matching problems involving high-dimensional spatially distributed parameters, it is difficult or even impossible to build such a global surrogate model until the advent of deep-learning-based methods. Recently, deep-learning-based surrogate models have been made remarkable achievements in inverse modeling. Especially, the grid-based high-dimensional spatially varying parameter fields can be treated as image type data, and the convolutional neural networks (CNN) can directly process these data. Based on the encoder-decoder structure, Zhu and Zabaras (2018) proposed a CNN surrogate model for the underground flow problem. The generative adversarial network (GAN) (Goodfellow et al., 2014) also has been studied for surrogate modeling the reservoir pressure and fluid saturation (Zhong et al. 2020, 2021). Zhang et al. (2022) predicted saturation and pressure fields by taking vector type parameters such as relative permeability as input parameters while considering high-dimensional parameter fields. Compared with the traditional surrogate modeling methods, the deep learning-based methods can represent more complex mapping relations and describe spatial-temporal characteristics of the reservoir flow problems.

However, most of the recently proposed deep-learning-based surrogate modeling methods are based on the image-to-image regression framework, for example, the permeability field is used as the input to predict the pressure and saturation field. These image-to-image-based models need to store, read, and process the saturation and pressure field of each time step, which will increase the cost of extra computation, especially for the long time-series prediction of large-scale 3D reservoir models. In most cases, the observations of AHM are time-series data, which can be obtained from the measurements of production and injection wells, such as oil production rate (OPR), water production rate (WPR), and bottom hole pressure (BHP). Therefore, the previous works

need to use the Paceman model (Peaceman, 1978) to calculate the production data based on the predicted pressure and saturation fields (Tang et al., 2020; Zhong et al., 2021; Zhang et al., 2022), which may introduce extra errors to the calculation of the flow data at production wells.

In this paper, we propose an end-to-end trainable surrogate model for history matching problems, termed as HRCN. In HRCN, a CNN model first encodes the input parameter fields into a feature vector, and then a recurrent neural network (RNN) decodes the features to predict the production data. In addition, the fluctuations of production data are influenced by well control parameters. Therefore, the well control parameters are used as auxiliary inputs to make accurate predictions. This deep-learning model structure, which combines CNN and RNN, is also widely used in the field of image captioning (Kinghorn et al., 2018; Tan and Chan, 2019). Based on the proposed surrogate, we develop a novel history matching workflow by integrating the MEDA algorithm. History matching is a typical inverse problem with multiple solutions. Existing studies (Chen et al., 2018; Ma et al., 2021) have shown that history matching can be treated as a multimodal optimization problem. It is necessary to explore the posterior probability space as fully as possible to obtain multiple best matched solutions. Estimation of distribution algorithm (EDA) is a population-based algorithm that searches for new solutions by sampling from an estimated probability distribution of previous promising solutions. MEDAs can find multiple optimal solutions by integrating niching techniques into conventional EDAs. Therefore, in this work, a MEDA (Yang et al., 2017) is utilized to solve the history matching problem with multiple solutions.

The remainder work is organized as follows. Section 2 briefly introduces the long short-term memory (LSTM) unit which is used in the RNN model. In section 3, the proposed surrogate model HRCN is presented in detail. The surrogate-based MEDA history-matching workflow is presented in section 4. Then in section 5, the proposed surrogate model and history matching workflow are evaluated on a 2D and a 3D waterflooding reservoir model. Finally, the conclusions and some potential future work are summarized in section 6.

2. Related work

2.1. Long short-term memory unit

LSTM has been widely used in sequence modeling problems because it can capture and maintain important features over a long range of sequences. A basic LSTM unit (Hochreiter and Schmidhuber, 1997) consists of a cell state and three gates. The cell state is used to convey information throughout the timeline, while three gates are used to control the amount of information. Specifically, the forget gate is used to determine whether the input information enters the cell state, the input gate is used to adjust the entry of new information into the cell, and the output gate calculates the output of the LSTM unit. Eqs. (1)–(4)

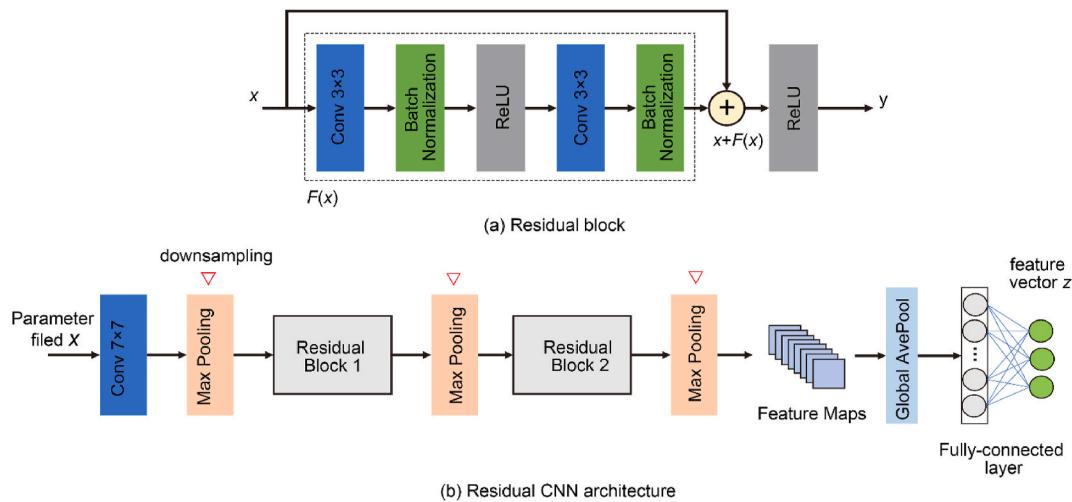


Fig. 2. Architecture of the residual convolutional network.

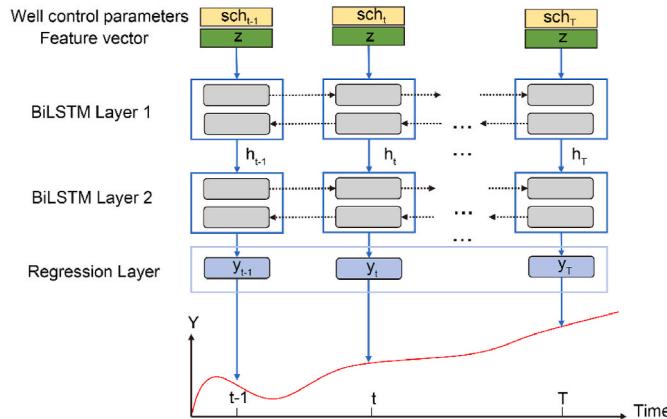


Fig. 3. RNN-based time-series module.

summarize the calculation of forget gate f_t , input gate i_t , output gate o_t , and cell state c_t in a basic LSTM unit:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (4)$$

where x_t is the input at time t , h_{t-1} is hidden state at time $t-1$. W_α and b_α represent the weight parameters and bias term of each gate α . And σ is the sigmoid activation function, c_{t-1} and c_t are the cell state at time $t-1$ and time t , $\tanh(\cdot)$ is the arctangent function. The output hidden state h_t of current LSTM unit can be computed as $h_t = o_t \cdot \tanh(c_t)$.

3. Proposed hybrid recurrent convolutional network surrogate model

In AHM, spatially varying parameter fields, such as permeability, need to be adjusted by observed data $d_{obs} \in R^{T \times N_d}$, such as OPR and WPR of different production wells. It is necessary to input the estimating parameter fields into the numerical simulator to obtain the corresponding production data $y^{sim} \in R^{T \times N_d}$, which is usually a computationally time-consuming process. Therefore, we propose the HRCN surrogate model to approximate this process:

$$F : R^{H \times W \times N_x} \rightarrow R^{T \times N_d} \quad (5)$$

where $H \times W$ is the spatial discretization resolution, N_x is the number of the parameter fields, T is the number of time-step of production data, and N_d is the number of measurement types.

As shown in Fig. 1, the proposed HRCN surrogate model consists of a spatial feature extraction module and a time series prediction module. In the following, we describe the proposed model in detail.

3.1. Extracting spatial features using the residual CNN

For the spatially varying parameter fields, such as permeability and porosity, deep CNN can extract high-level feature representations (Zhao and Kumar 2019). However, deep CNN is difficult to train due to the possibility of gradient disappearance. In this work, we apply the residual CNN structure to extract the spatial features of uncertain parameter fields. Residual CNN uses residual learning with a short connection structure (He et al., 2016), which can effectively solve the problem of gradient disappearance during training the deep network.

Fig. 2a shows the basic architecture of a residual block, in which, the desired underlying mapping $H(x)$ is fitted using $F(x) + x$, where x denoting the inputs. Therefore, the stacked layers in a residual block mainly approximate a residual function $F(x) = H(x) - x$, which is a composite function consists of convolution (Conv), batch normalization (BN), and the rectified linear units (ReLU). The operation $F(x) + x$ is a shortcut connection structure by using the element-wise addition. Compared with conventional CNN direct learning $H(x)$, learning the residual function $F(x)$ is easier. Based on the residual block, we can obtain the residual CNN. As shown in Fig. 2b, the model first deals with the input parameter fields using a convolution layer and a down sampling (Max-pooling) layer, then multiple stacked residual blocks and down sampling layers are adopted to extract the feature maps. In the down sampling, the size of the feature map is halved, meanwhile, the number of feature maps in the subsequent residual block is doubled to compress and extract abstract feature representations of the input parameter fields. In this work, we use three residual blocks to build the residual CNN. Afterward, a global average pooling (GAP) operation is used to calculate the average output of each channel in the final feature maps (Christlein et al., 2019). The residual CNN ends with a fully-connected layer to obtain the final feature vector z of the input parameter fields X .

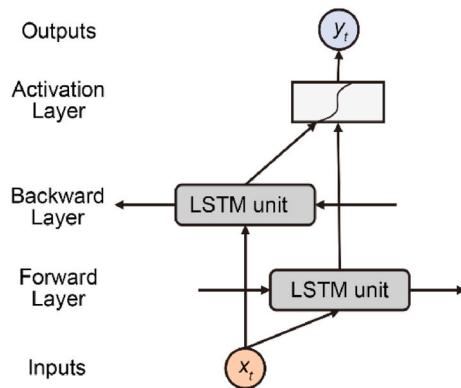


Fig. 4. The schematic diagram of BiLSTM layer.

3.2. Deep RNN for production data forecast

Generally, the simulation data in AHM include multiple time series such as oil/water production rates and bottom-hole pressure. The temporal dynamics of these data are difficult to model and predict because they relate not only to reservoir parameters but also to surface well control measures. To tackle this problem, we construct a deep RNN model with LSTM units for production data prediction using the extracted spatial features and well control parameters.

As shown in Fig. 3, in this RNN-based time-series module, the combination of extracted spatial feature vectors z and well control vectors sch is used as input parameters for each time step. Then two-layer bidirectional LSTM (BiLSTM) is adopted to model the temporal features from the input layer. Compared with the single-layer BiLSTM model, the two-layer BiLSTM can predict the sequence with stronger non-linearity (Graves et al., 2013; Sagheer and Kotb, 2019). Fig. 4 depicts the schematic diagram of BiLSTM which mainly consists of the forward and the backward layer (Graves and Schmidhuber, 2005). In the forward layer, LSTM processes sequence data forward in the timeline. Conversely, LSTM processes sequence data in reverse in the backward layer. This bidirectional structure can enhance the feature representation and promote temporal dependability. Finally, a time-distributed fully-connection structure is used in the module as the regression layer to obtain the prediction data.

3.3. Loss function and training strategy

The proposed HRCN model is end-to-end trainable and uses the mean absolute error (MAE) function as the loss function. MAE is one of the most widely used loss functions in regression tasks. It calculates the average of the absolute difference between the actual and predicted values and is more robust to outlier data points in the training samples:

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

where θ is the trainable parameters, y_i^{sim} and y_i^{pred} is the simulation data and prediction data of i th sample.

Then the Adam algorithm (Kingma et al., 2014) is adopted to optimize the loss function to adjust and update the model parameters according to the back-propagation method. The learning rate is very important for the performance of the Adam algorithm. In this work, the initial learning rate is set as 0.001 and then divided by 10 when the error stagnates. To prevent overfitting, we utilize the early-stopping strategies (Raskutti et al., 2011) to train the model. The above process and our model are implemented using Keras and TensorFlow (Abadi et al., 2016).

3.4. Evaluation matrix

As shown in Eqs. (7) and (8), two commonly used performance metrics, coefficient of determination R^2 and root-mean-square error $RMSE$, are used to evaluate the predictive performance of the surrogate model. R^2 is a statistical measure used to assess the extent to the regression predictions approximate real data. The closer R^2 is to 1, the closer the predicted value is to the real data. Meanwhile, the $RMSE$ is used to evaluate the accuracy of the forecast.

$$R^2 = 1 - \frac{\sum_{i=1}^N \|y_i^{sim} - y_i^{pred}\|_2^2}{\sum_{i=1}^N \|y_i^{sim} - y_m^{sim}\|_2^2} \quad (7)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i^{sim} - y_i^{pred})^2} \quad (8)$$

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

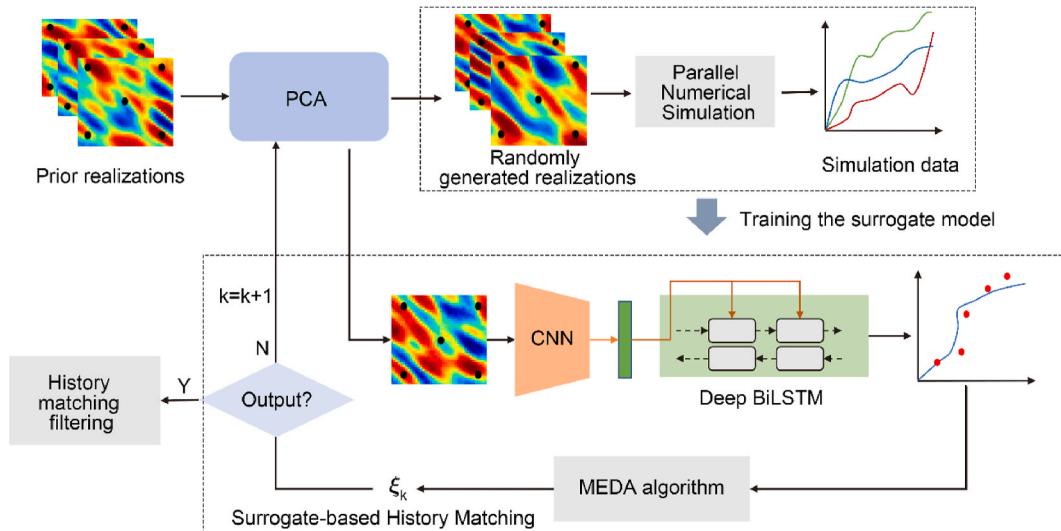


Fig. 5. The proposed surrogate-based history matching workflow.

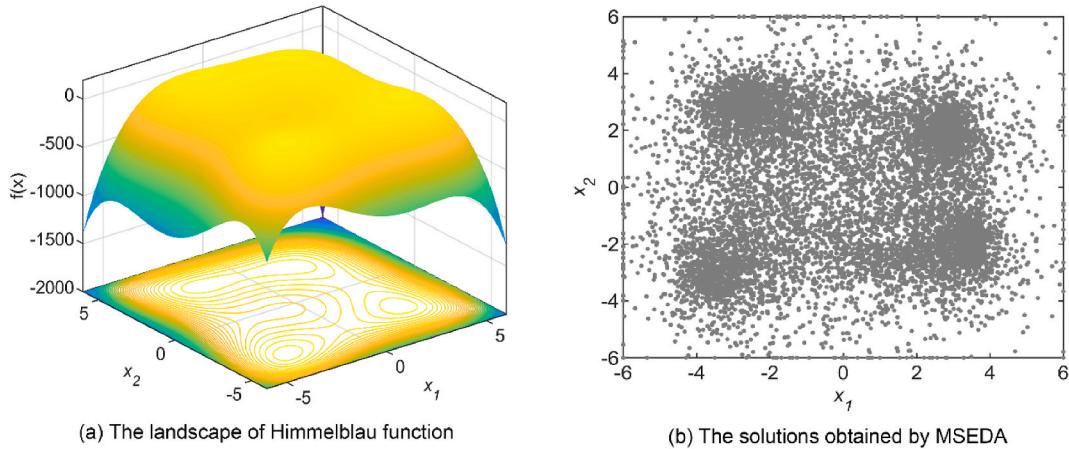


Fig. 6. Illustration of distributed convergence capability of MEDA algorithm.

4. Surrogate-based MEDA history matching workflow

As shown in Fig. 5, the proposed workflow mainly consists of four parts. Firstly, based on prior knowledge, the prior realizations of parameter fields are constructed using the geostatistical modeling method. And the parameterization model is trained based on these prior realizations. Secondly, the parameterization model is used to randomly generate multiple parameter fields, and the corresponding production data are obtained through numerical simulations. Based on these data, the HRCN surrogate model can be trained. Thirdly, the trained surrogate model is combined with the MEDA algorithm to estimate parameter fields, and numerical simulation is not necessary for this process. Finally, the solutions obtained by MEDA are filtered to obtain the final history-matched models. This step requires a small number of numerical simulations. In the following, we present the details of these parts.

4.1. PCA-based parameterization and objective function

For the estimation of spatially varying parameters, it is difficult for existing algorithms to update parameters directly involving hundreds of thousands or even millions of grid blocks. Therefore, it is necessary to reduce the dimension of parameters by means of the parameterization technique. In this work, we apply the proposed methods to the estimation of continuous parameter fields, so the principal component analysis (PCA) is used as the parameterization model. In PCA, N_p prior realizations are combined into a centered matrix:

$$X_c = \frac{1}{\sqrt{N_p - 1}} [x_1 - x_m \quad x_2 - x_m \quad \dots \quad x_{N_p} - x_m] \quad (9)$$

where $X_c \in R^{N_x \times N_p}$, $x_i \in R^{N_x \times 1}$ represents the i th realization, and $x_m \in R^{N_x \times 1}$ is the mean of all N_p realizations. Then the singular-value decomposition method is performed on the centered matrix X_c , which gives $X_c = U\Sigma V^T$, where $U \in R^{N_x \times N_x}$ and $V \in R^{N_p \times N_p}$ are the left and right singular matrix respectively, $\Sigma \in R^{N_x \times N_p}$ is a diagonal matrix whose elements are the square root of the corresponding eigenvalues. On this basis, the new PCA realizations can be generated by

$$x_{pca} = U_l \Sigma_l \xi + x_m \quad (10)$$

where $U_l \in R^{N_x \times l}$ contains the first l columns in U , $\Sigma_l \in R^{l \times l}$ is a diagonal matrix containing the first l largest singular-values, and $\xi \in R^{l \times 1}$ is the low-dimensional vector drawn from the standard normal distribution. In PCA, the $l \ll N_x$ which can be determined by using an energy criterion. Based on the PCA parameterization, the objective function of history matching can be defined as follows:

$$O(\xi) = \frac{1}{2} [y_{(\xi)} - d_{obs}]^T C_D^{-1} [y_{(\xi)} - d_{obs}] + \frac{1}{2} \xi^T \xi \quad (11)$$

where C_D is the covariance matrix of observation error, $y_{(\xi)}$ is the corresponding production data of the parameter fields generated by ξ . The first term of the objective function can be derived from the likelihood function of Bayesian inference, and the second term represents a regularization constraint on ξ , such that ξ obey the normal distribution as much as possible.

4.2. Multimodal estimation of distribution algorithm

As shown in Algorithm 1, the MEDA algorithm consists of three procedures: speciation clustering with dynamic cluster size, distribution estimation and offspring generation, and crowding selection. By speciation clustering, the population is divided into multiple sub-populations, also known as niches, and the corresponding probability distribution parameters for each niche are estimated, usually using the Gaussian distribution as follows:

$$\mu_i^d = \frac{1}{M} \sum_{j=1}^M x_j^d \quad (12)$$

$$\sigma_i^d = \sqrt{\frac{1}{M} \sum_{j=1}^M (x_j^d - \mu_i^d)^2} \quad (13)$$

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 niche, $x_j = [x_j^1, \dots, x_j^d, \dots, x_j^D]$ is the j th individual in the i th niche, M is the cluster size and D is the dimension size of the problem. In each iteration, the cluster size M of speciation clustering is determined dynamically, that is, an integer is randomly selected from a predefined range. In this work, we set the minimum and maximum value of M are 5 and 10 as suggested in (Yang et al., 2017). Then the next step is the generation of offspring using the Gaussian distribution as follows:

$$U_i^t = \text{Gaussian}(\mu_i^t, \sigma_i^t, N_c M) \quad (14)$$

To increase the search capability for each niche, we use a scaling factor N_c to control the number of generated offspring solutions. In surrogate-based history matching, the optimization process is entirely surrogate-based, N_c can select a larger value to generate more solutions. After generating the offspring solutions, the population is updated by crowding selection (lines 13–15 in Algorithm 1).

As shown in Fig. 6a, a multimodal problem with four peaks, Himmelblau function (Himmelblau, 1972), is used to illustrate the

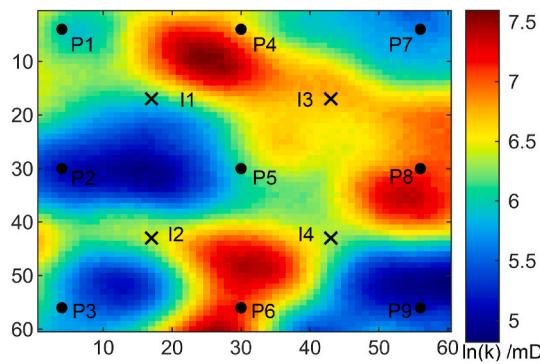


Fig. 7. The true log-permeability and well pattern for the 2D Case.

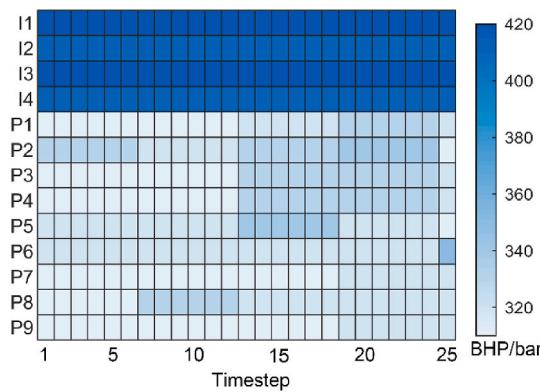


Fig. 8. Heatmap of well control parameters over time-steps (Case 1).

Table 1

Results of cross check of hyperparameters. N_z : the dimension of extracted spatial feature, N_{unit} : the number of neurons in BiLSTM layer, R^2 : the coefficient of determination on training set, $ValR^2$: the coefficient of determination on validation set. The best value is indicated in bold.

No.	N_z	N_{unit}	R^2	$ValR^2$
1	100	100	0.9645	0.8903
2	100	200	0.9631	0.8742
3	500	100	0.9717	0.8800
4	500	200	0.9744	0.8704
5	1000	100	0.9714	0.8660
6	1000	200	0.9869	0.8445

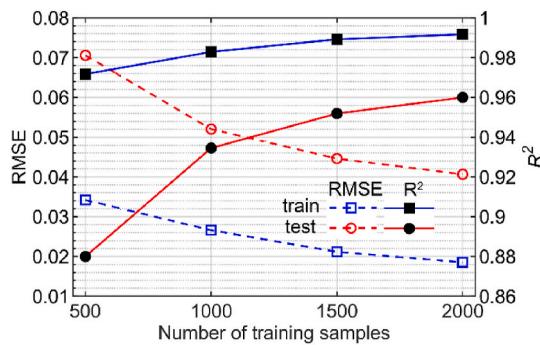


Fig. 9. Comparison of the $RMSE$ and R^2 score evaluated on the test samples of the 2D case.

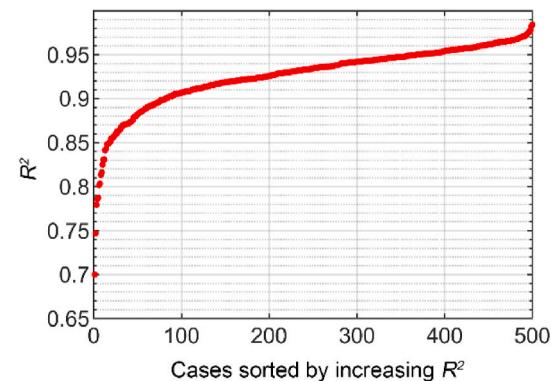


Fig. 10. Overall R^2 for all test samples, sorted in increasing order (2D Case).

distributed convergence property of MEDA. The expressions of Himmelblau function is as follows:

$$f(x_1, x_2) = 200 - (x_1^2 + x_2 - 11)^2 - (x_1 + x_2^2 - 7)^2 \quad (15)$$

The search range of this function is $[-6, 6]$ for all dimensions. Fig. 6b shows the solutions collected by the MEDA. It can be seen that the obtained solutions are distributed in the corresponding regions of the four peaks. This shows that the MAEDA method can locate multiple solutions and maintain the diversity of solutions.

Algorithm 1. Multimodal Estimation of Distribution Algorithm.

```

1: Randomly initialize the population  $P$  and evaluate the fitness of  $P$ ;
//Speciation clustering with dynamic cluster size// 
2: Randomly generate the cluster size  $M = (M_{max} - M_{min}) \times \text{rand}(0, 1) + M_{min}$ 
3: Sort  $P$  according to fitness;
4: While  $P$  is no empty:
5:   Select the best individual  $P_{best}$  in  $P$  as a new seed;
6:   Combine  $M-1$  individuals nearest to  $P_{best}$  and  $P_{best}$  as a species  $S_i$ ;
7:   Eliminate these  $M$  individuals from  $P$ ;
8: End While
//Generate the offspring solutions//
9: For each species  $S_i$  in  $S$ :
10: Produce the offspring solutions  $U_i$  using Eq. (14);
11: End For
12: Evaluate the fitness of offspring  $U$ ;
//Crowding selection//
13: For each new individual  $u_j$  in  $U$ :
14:   Compare the fitness of  $u_j$  with that of the most similar individual in  $P$  and replace
       it if  $u_j$  is better;
15: End For
16: Stop if a termination criterion is satisfied. Otherwise go to Step 2.

```

4.3. History-matching filtering

Due to the existence of surrogate error, samples obtained by using the surrogate-based MEDA need to be filtered according to the numerical simulation results to select a set of history-matched models. In this work, we treat the final population as the updated samples and perform the numerical simulation to obtain the corresponding true objective function values. Then an objective function value can be selected according to the fitting quality of production curves to determine the filtering criteria. A more detailed description of the history-matching filtering can be found in (Li et al., 2019).

5. Case study

In this section, we assess the performance of the HRCN surrogate model for time-varying production data predictions. We then use the surrogate for history matching problems. The results are presented for two different cases: the estimation of a 2D Gaussian permeability field and a 3D Brugge model. Additionally, all neural networks in this work

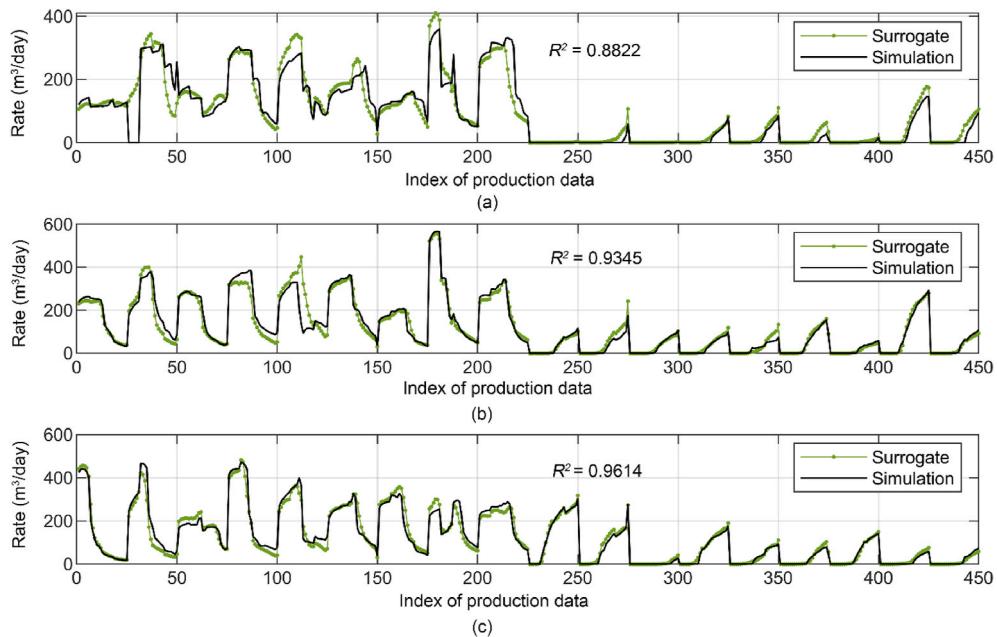


Fig. 11. Comparison of well rates from the numerical simulator (black lines) and surrogate model (green dots and lines) for the (a) P10, (b) P50 and (c) P90- R^2 test samples (2D Case). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

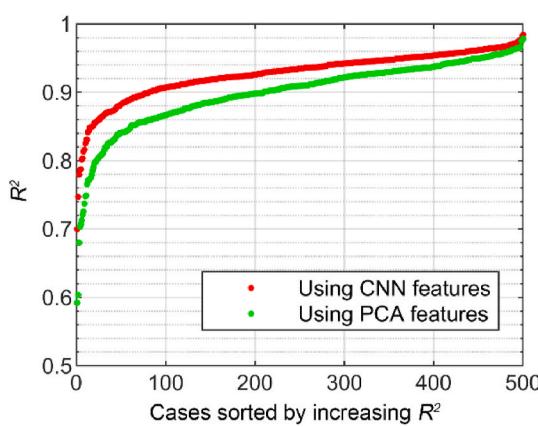


Fig. 12. Comparison of prediction performance using PCA feature and CNN feature as input.

are trained on a single Nvidia Tesla V100 GPU with 32 GB memory.

5.1. Case 1: permeability inversion of a 2D Gaussian model

Fig. 7 shows the true log-permeability field and the well pattern for this 2D case. The reservoir model is discretized uniformly into 60×60 grid blocks, and the dimension of each grid block is $10\text{m} \times 10\text{ m}$. The thickness and porosity are 4 m and 0.2 respectively for all grids. There are four injection wells and nine production wells in the reservoir domain, all of which are BHP-controlled. In this case, we set the interval for each time step to 30 days. **Fig. 8** depicts the BHP values for all wells at 25 time-steps for a history-matching period of 750 days. The observed data consist of OPR and WPR of all production wells. Therefore, the observation data and simulation data during history matching have 25 time-steps and 18 observation types. The observed data are generated by adding Gaussian noise with zero means to the numerical simulation data of the true permeability field. The standard deviation of the noise is set as 5% of the simulation data.

Then 100 random prior realizations for the log-permeability are generated independently using the Stanford Geostatistical Modeling Software (SGeMS) (Remy, 2005) without using hard data. And the log-permeability is parameterized using PCA with a reduced dimension of 100. This value is determined based on the number of prior realizations. Afterward, we generated 2500 random PCA realizations and perform flow simulations of the history-matching period as the data set. We use the first 2000 simulation results as training data and the remaining 500 simulation results as test data. And the epoch and batch size are set as 200 and 16, respectively.

Firstly, we present the assessment of surrogate performance for various hyperparameter values. Here, we only consider the dimension of extracted spatial feature N_z and the number of neurons in the BiLSTM layer N_{unit} . In this comparison, we set the $N_{train} = 500$, the coefficient of determination on training set R^2 and testing set Val_R^2 after training is shown in **Table 1** for 6 different combinations of hyperparameters. We can see that the sixth parameter combination achieved the best training performance, but the worst test performance. The first set of parameters performed the best test performance. Generalization performance is more important for prediction. Therefore, the first settings are used in the following experiments.

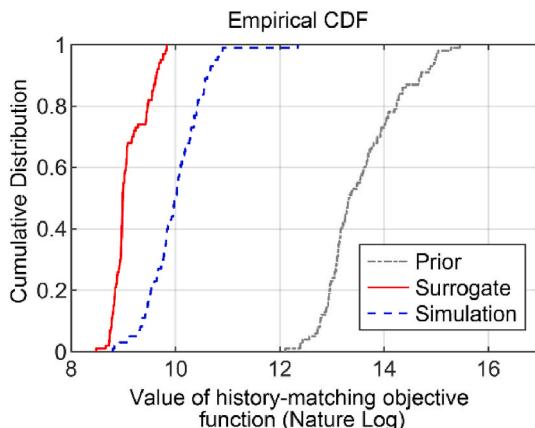


Fig. 13. Comparison of objective function values of prior models, surrogate-based history-matched models and simulation-based history-matched model.

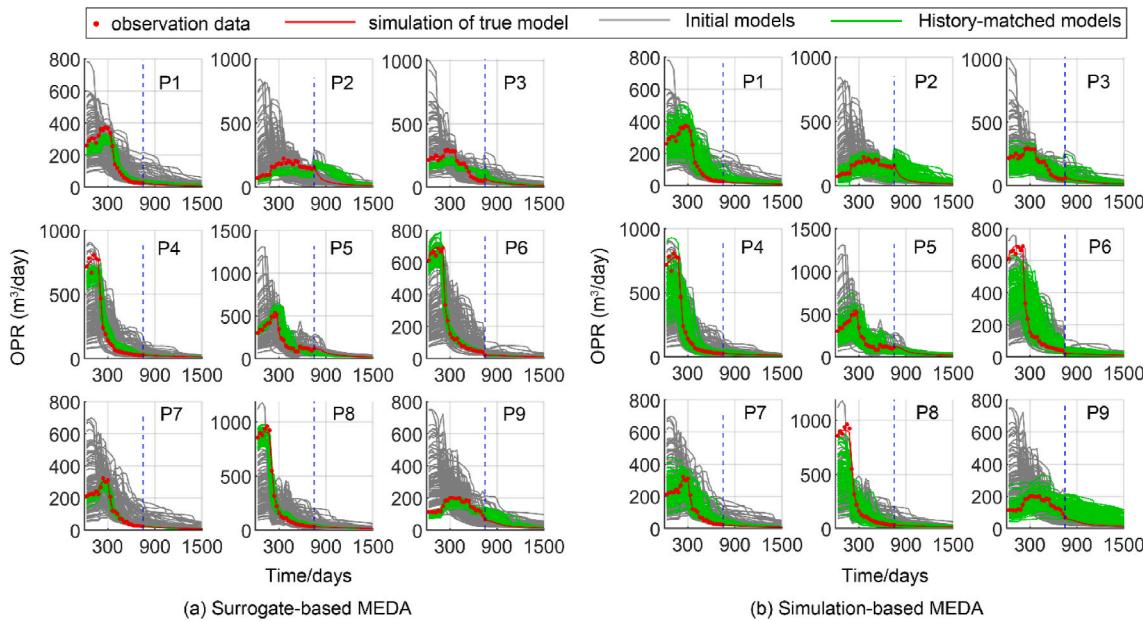


Fig. 14. Comparison of the matching and prediction of the OPR of the posterior parameter fields obtained by (a) the surrogate-based MEDA and (b) the simulation-based MEDA.

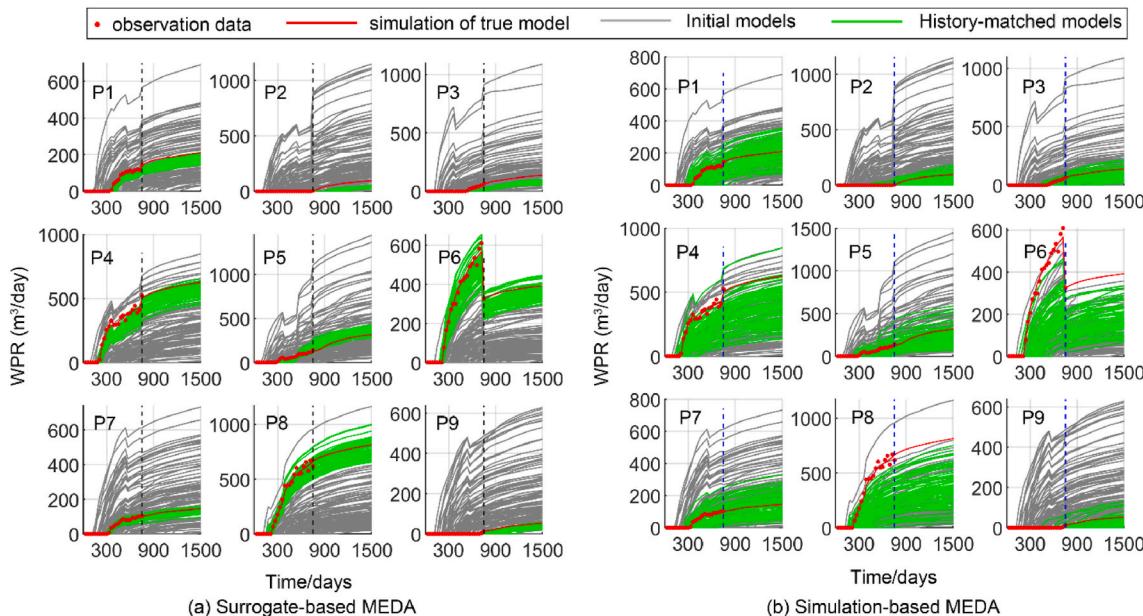


Fig. 15. Comparison of the matching and prediction of the WPR of the posterior parameter fields obtained by (a) the surrogate-based MEDA and (b) the simulation-based MEDA.

We next evaluate the impact of training sample size N_{train} . Fig. 9 displays the RMSE and R^2 results of using different N_{train} . The blue squares and red circles indicate R^2 for the training and test sets, respectively. It can be seen that the performance of the model is significantly improved when the N_{train} increases from 500 to 1000. However, the test performance of the surrogate is slightly improved when the N_{train} increases from 1000 to 2000. Therefore, we specify $N_{train} = 1000$ for training the surrogate in the following studies.

We retrain the surrogate model using the hyperparameters and training sample size chosen in the above assessment. Fig. 10 displays the R^2 , sorted in increasing order, for all the test cases. The P10, P50, and P90 of test R^2 here are 0.8822, 0.9345, and 0.9614, and the corresponding prediction of production data are presented in Fig. 11. In all

displays, the black lines denote the reference numerical simulation results, and the green lines with dots indicate the surrogate estimations. In this figure, every 25 data points correspond to an observation type. The first 225 data points are the OPR from wells P1 to P9, and the last 225 data points are the WPR from wells P1 to P9. It is apparent that the surrogate provides accurate predictions for all quantities.

Additionally, we compared the prediction performance of production data using CNN features with using PCA features directly. As shown in Fig. 12, it can be seen that the prediction performance of using CNN features is significantly better than that of using PCA features. In general, using the CNN features is more robust and can realize end-to-end training from high-dimensional parameter fields to production data. And using parameter fields as input can hold more information.

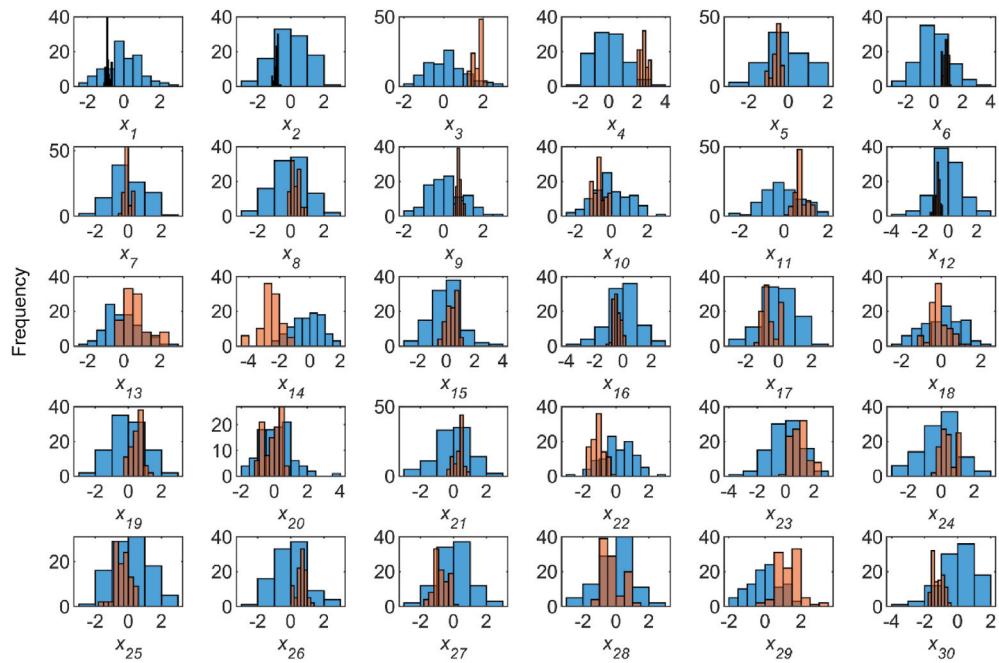


Fig. 16. The frequency histogram of the first 30 latent variables of prior samples (blue) and surrogate-based MEDA results (orange). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

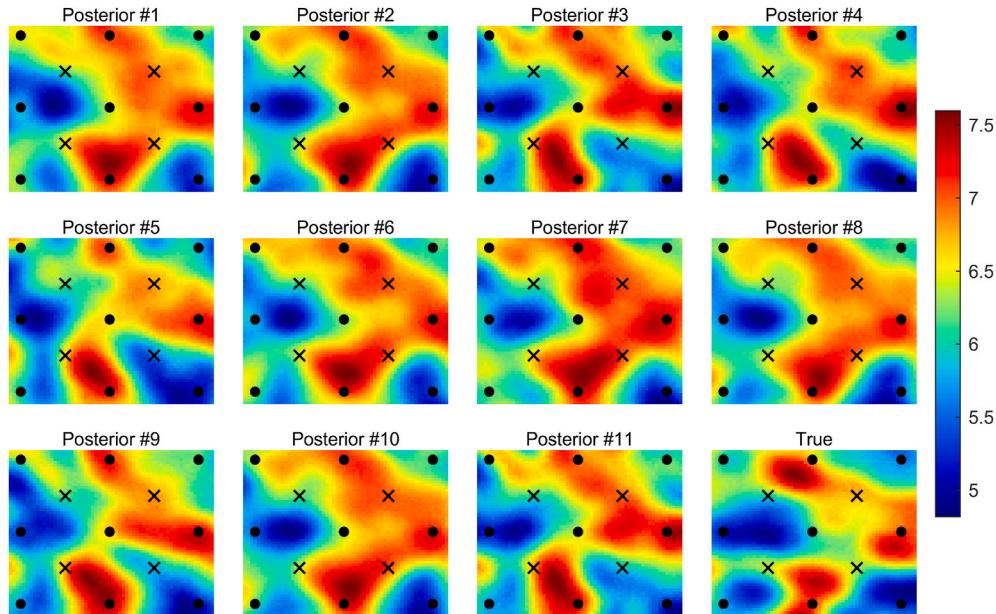


Fig. 17. Results of permeability field using the surrogate-based MEDA.

After the analyze of the surrogate performance, we now use it for history matching. For the surrogate-based MEDA, the population size is set to $NP = 100$ and the coefficient of generated offspring solutions $N_c = 5$. The termination condition of the algorithm is that the number of iterations reaches 100, which means that the number of function evaluations required in surrogate-based MEDA is 50,000. To compare and verify the surrogate-based history-matched results, we also used the simulation-based MEDA. The NP and N_c are set as same as in surrogate-based MEDA. However, considering the simulation is computational time-consuming, the termination condition of the algorithm is set as the maximum function evaluations reaches 10,000. The surrogate-based MEDA do not require any simulations during optimization, but 1000 full-order training runs are required. GPU-based training is also

required, but this is a one-time cost and only need above 10 min. After history-matching, the final updated population needs run simulations. Therefore, the total number of simulations required for surrogate-based MEDA is 1100 in the history matching, which is about 10% of simulations used in the simulation-based MEDA.

The log values of objective function of prior models, the final population of surrogate-based MEDA and simulation-based MEDA are shown in Fig. 13. The log objective function values of the prior model are concentrated between 12 and 16. The values of objective function of the surrogate-based results are mostly around 9. The wide range of the objective function values of simulation-based results means poor convergence. For this case, all the solutions in the final population obtained by surrogate-based MEDA have a good fit quality.

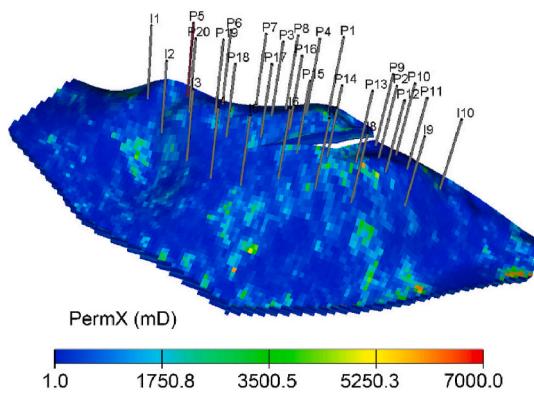


Fig. 18. The true log-permeability and well pattern for the 3D Case.

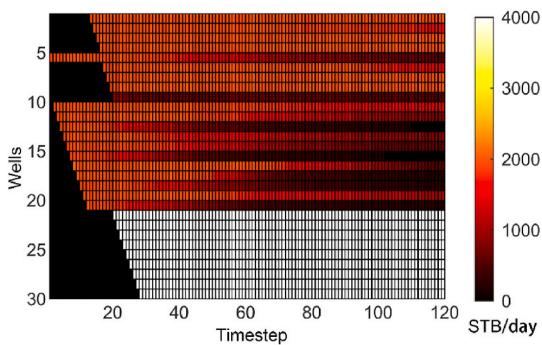


Fig. 19. Heatmap of well control parameters over time-steps. On the Y-axis, 1 to 20 represent production wells P1 to P20, 21 to 30 represent injection wells I1 to I10.

Table 2

Results of cross check of hyperparameters. N_z : the dimension of extracted spatial feature, N_{unit} : the number of neurons in BiLSTM layer, R^2 : the coefficient of determination on training set, $Val R^2$: the coefficient of determination on validation set. The best value is indicated in bold.

No.	N_z	N_{unit}	R^2	$Val R^2$
1	100	100	0.9867	0.9807
2	100	200	0.9849	0.9804
3	500	100	0.9822	0.9782
4	500	200	0.9879	0.9804
5	1000	100	0.9494	0.9476
6	1000	200	0.9861	0.9800

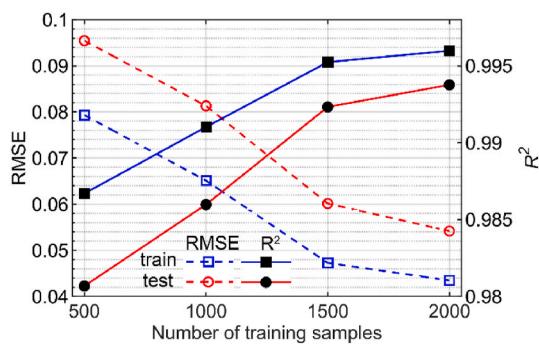


Fig. 20. Comparison of the RMSE and R^2 score evaluated on the test sets of the 3D case.

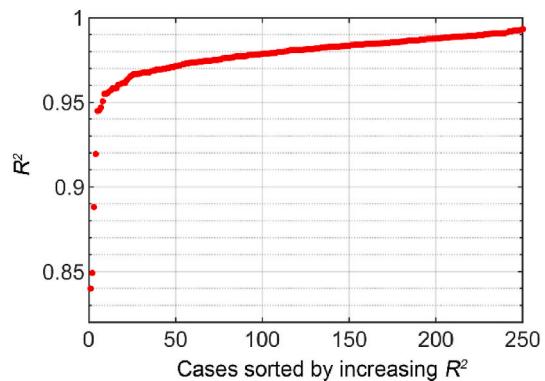


Fig. 21. Overall R^2 for all test samples, sorted in increasing order (3D Case).

Figs. 14 and 15 compare the history-matching and forecast results of OPR and WPR for all production wells, respectively. The dashed blue vertical lines in Figs. 14 and 15 indicate the end of the history-matching period (750 days). Compared with the simulation-based results, posterior models from surrogate-based MEDA provide reasonably close agreement to the observed data in the history-matching period, and much less uncertainty in predictions at later times. In particular, the fitting results of WPR show that simulation-based MEDA has not reached convergence. It is evident that the surrogate-based MEDA uses fewer numerical simulation runs to get a better history matching results.

Fig. 16 shows the histogram of the first 30 low-dimensional variables of the prior and history-matched solutions obtained by surrogate-based MEDA. The first two variables converge to specific values, while the subsequent variables converge while maintaining the diversity. In the PCA-based parameterization, the eigen field (eigenvector) corresponding to each low-dimensional variable is sorted according to their eigenvalues. Therefore, through the history matching, surrogate-based MEDA can identify the main features of permeability field while retaining a variety of secondary features. In general, history matching has multi-solutions, and retaining the diversity of solutions while reaching the acceptable fitting of observation is beneficial to the subsequent production prediction and optimization. Results for log-permeability estimation are presented in Fig. 17. Compared with the true model, in all posterior models, the distribution of high-permeability and low-permeability regions is essentially correct, and the posterior models differ in the local details.

5.2. Case 2: Brugge model

We now apply the proposed HRCN model and surrogate-based MEDA in the Brugge benchmark case (Peters et al., 2013). The Brugge case is designed from a real waterflooding oil field, which has been widely used to test the performance of history matching methods. The reservoir model has 60,048 ($138 \times 49 \times 9$) grid blocks with a total of 44,550 active cells. Fig. 18 displays the true log permeability and well pattern. Inside the model, there is a fault that affects the lateral flow. Brugge field has 20 production wells and 10 injection wells with 3 perforation intervals per well. The noise-corrupted 10-year history production data, including WOPR and WWPR of 20 producers and BHP of all 30 wells, are reported monthly. The noise level is set to 3% for the production rate and 7.25 psi for BHP. As shown in Fig. 19, the injection wells control the injection rate at 4000 STB/day, while the production wells are shut if the water cut exceeds 0.9 and the simulator is controlled to match well oil rates.

The uncertain parameters include porosity, net-to-gross, and permeability (X , Y , and Z directions) at each grid block, and the original Brugge dataset provides 104 prior realizations. Therefore, there are $5 \times 60,048$ uncertain parameters, and the PCA is used as the parameterization method to reduce the dimension to 104. We again generated 2500

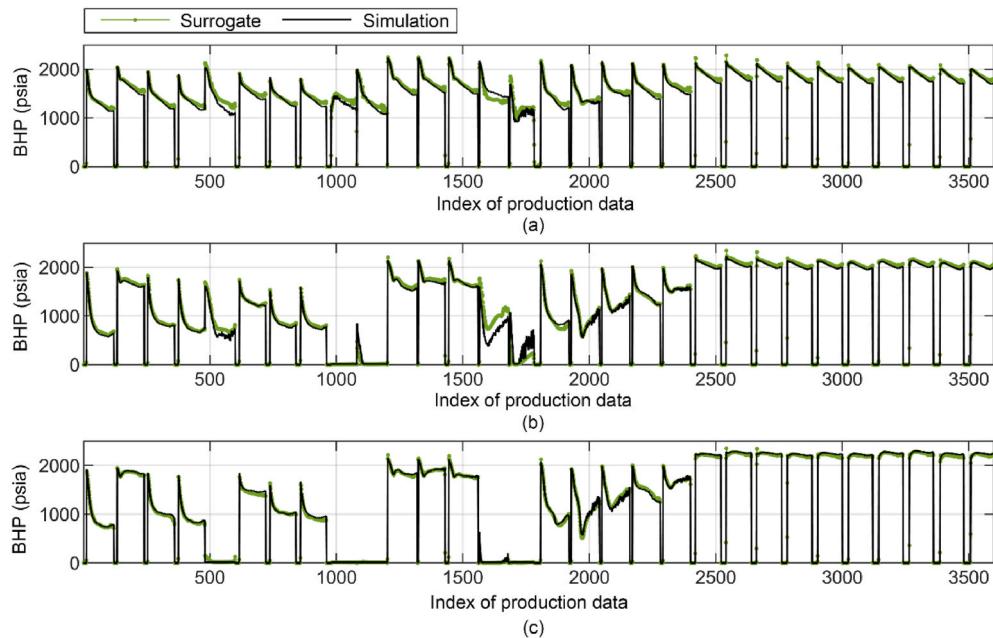


Fig. 22. Comparison of well bottom-hole-pressure from the numerical simulator (black lines) and surrogate model (green dots and lines) for the (a) P10, (b) P50 and (c) P90- R^2 test samples. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

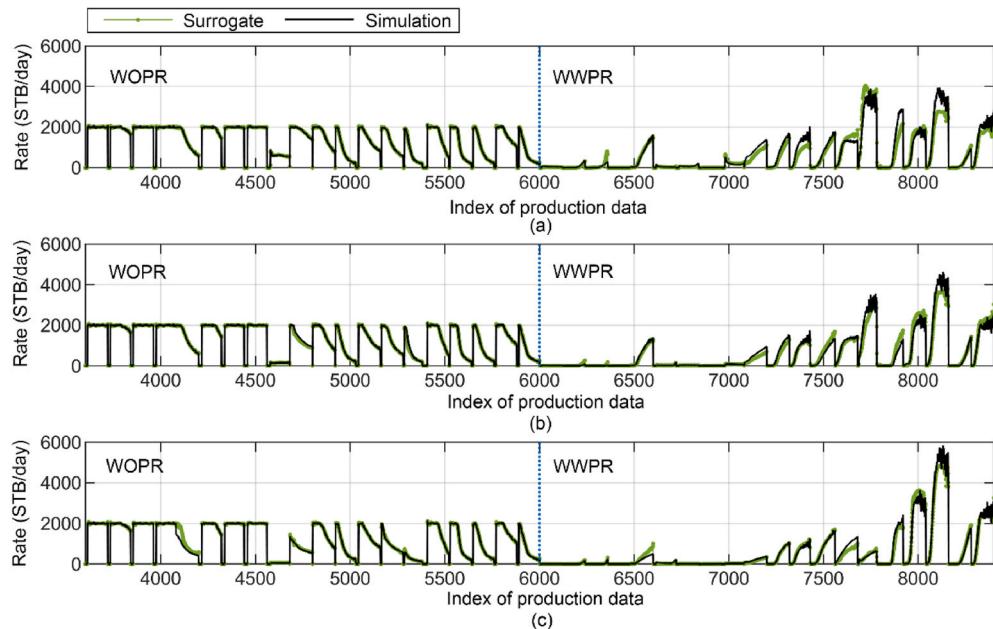


Fig. 23. Comparison of well rates from the numerical simulator (black lines) and surrogate model (green dots and lines) for the (a) P10, (b) P50 and (c) P90- R^2 test samples (3D Case). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

random PCA realizations and perform the production simulation of the history matching period as the data set for this case study. The dataset is divided into training set, validation set, and test set, including 2000, 250, and 250 samples respectively.

As same in Case 1, we first use 500 training samples for validation of the hyperparameters N_z and N_{unit} . The results of R^2 evaluation in training and test data are shown in Table 2. It can be seen that the first hyperparameters combination achieves the best test performance. Based on these results, then we compared the impact of different values of N_{train} . As shown in Fig. 20, the performance increases as the number of training samples increases. For this case, the validation R^2 obtained with different numbers of training samples is very close. This may be due to

less variation in production data due to intensive well control measures. To obtain a robust surrogate model, 1000 training samples are used in subsequent history matching.

To implement the history matching, we retrain the HRCN model. The training of the model needs about 17 min. The R^2 for each case in test samples is calculated using Eq. (8). Fig. 21 displays the test R^2 , sorted in increasing order, for all test cases. The P10, P50, and P90 of test R^2 for these cases are 0.9660, 0.9812, and 0.9897. Fig. 22 compares the surrogate predictions with numerical simulations for BHP of 30 wells. And Fig. 23 shows the comparison of surrogate predictions and numerical simulations for the oil and water production rate of all producers. As can be seen, the BHP and WOPR can be accurately predicted. But the WWPR

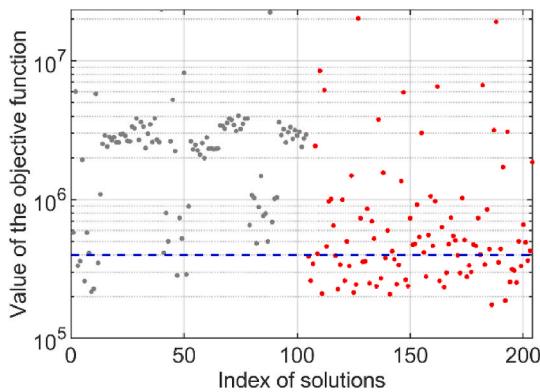


Fig. 24. The objective function values of the prior parameter fields and the updated parameter fields obtained by surrogate-based MEDA, and the selected fitting filter value.

of some wells changes dramatically in the later stage of production, and the forecast cannot capture the local changes. Generally, the HRCN model can obtain reliable predictions.

Afterward, we use the surrogate-based MEDA for history matching. The algorithm parameters are set the same as in Case 1. **Fig. 24** compares the objective functions of prior models (grey points) and updated models (red points). It is evident that the objective function values of the

updated models are significantly smaller than that of prior models. For this case, some updated models could not fit the production data well. According to the fitting effect, an appropriate objective function value is selected to carry out the history-matching filtering (the dashed blue horizontal line). Finally, 42 history-matched models are obtained.

The history-matching results of WBHP and WWPR of four selected production wells are shown in **Fig. 25**. The production data of prior models (grey lines) have relatively large uncertainty, especially for the water production rates. In contrast, the uncertainty of the production data of posterior models is significantly reduced. Particularly, as shown in **Fig. 25a-d**, the BHP has achieved a good fitting effect for each producer. **Fig. 26** displays the simulation results of field oil production rate (FOPR) and field water production rate (FWPR). As shown in **Fig. 26a**, the FOPR of the posterior models is consistent with that of the reference model when the simulator is controlled to match well oil rates. In addition, the FWPR of posterior models covers the results of the reference model better than the FWPR of prior models.

Fig. 27 shows the reference, posterior mean, and standard deviation of permeability in the X-direction of four layers. The black dots indicate wells. It can be seen that the permeability distribution of the mean field (**Fig. 27b**) is closed to that of the reference model (**Fig. 27a**). Additionally, it can be seen that the standard deviation (**Fig. 27c**) in the areas without wells is larger which indicates these regions have greater uncertainty.

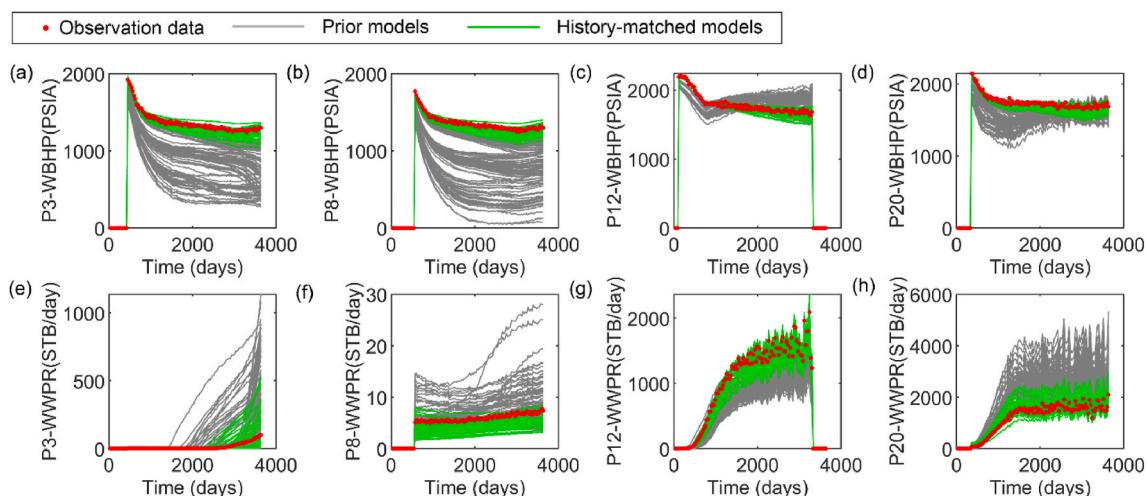


Fig. 25. Results of the history matching and the prediction results of posterior parameter fields obtained by the surrogate-based MEDA.

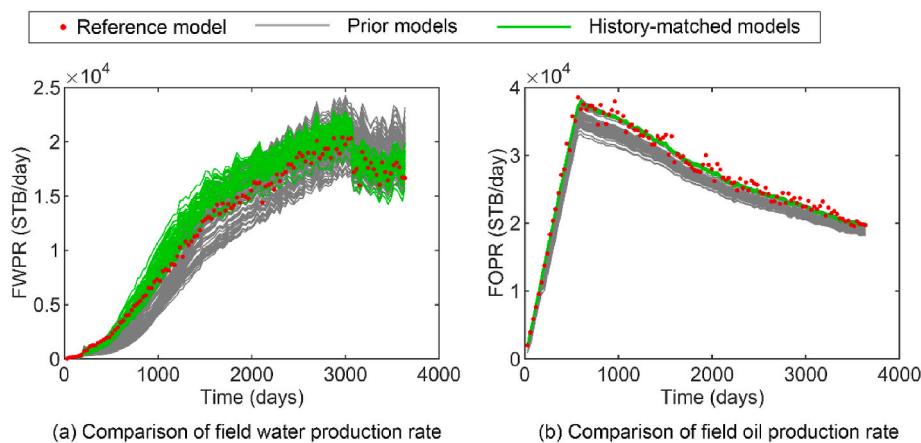


Fig. 26. Simulation of field (a) oil production rate and (b) water production rate.

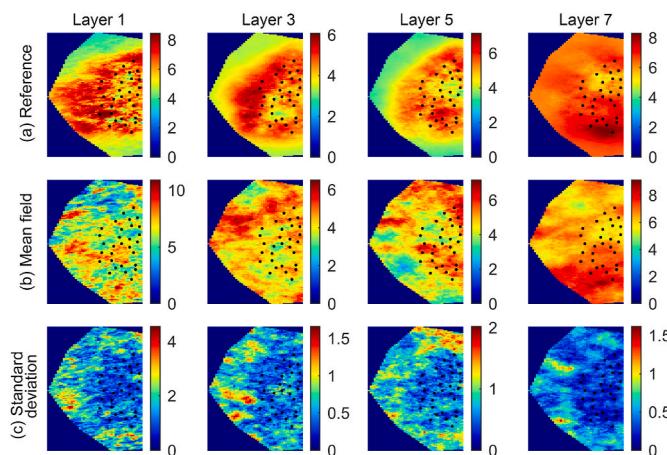


Fig. 27. The inversion results of permeability field (Brugge Case).

6. Conclusion

In this paper, we propose the HRCN model for surrogate modeling of numerical simulation used in AHM. The HRCN model is end-to-end trainable for predicting the well production data of high-dimensional parameter fields. In HRCN, the CNN with residual blocks can learn the high-level spatial feature representations of the input parameter fields. The extracted spatial features and well control parameters are integrated as the input for the RNN with BiLSTM aims to model the complex temporal dynamics and predict the production data. Based on the proposed HRCN model, we developed a novel surrogate-based AHM workflow, in which a MEDA algorithm is integrated. MEDA algorithm can maintain the diversity of the population while keeping convergence, which is conducive to finding more possible solutions. Numerical experiments on the 2D Gaussian model and the 3D Brugge benchmark case demonstrated the proposed surrogate model can accurately predict the corresponding production data from the parameter fields and this surrogate model provides an effective replacement of the time-consuming numerical simulations in AHM. Based on this surrogate, the strong exploration ability of MEDA is released and we can search the posterior space sufficiently to obtain a more comprehensive estimation of the uncertain parameter fields. Although the proposed HRCN surrogate model has shown promising prediction performance on the two case studies, the applicability to the non-Gaussian complex geological parameter fields needs further study.

Author contributions

Xiaopeng Ma: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. Kai Zhang: Methodology, Funding acquisition, Supervision, Project administration. Jinding Zhang: Validation, Resources. Yanzhong Wang: Software, Validation. Liming Zhang: Data curation, Resources. Piyang Liu: Visualization, Validation. Yongfei Yang: Software, Validation. Jian Wang: Data curation, Resources.

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.

Acknowledgments

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.
- Bhark, Eric, Dehghani, Kaveh, 2014. Assisted history matching benchmarking: design of experiments-based techniques. In: Proc., SPE Annual Technical Conference and Exhibition.
- Chang, Haibin, Liao, Qinzhao, Zhang, Dongxiao, 2017. Surrogate model based iterative ensemble smoother for subsurface flow data assimilation. *Adv. Water Resour.* 100, 96–108.
- 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.
- Christlein, V., Spranger, L., Seuret, M., et al., 2019. Deep generalized max pooling. In: Proc., 2019 International Conference on Document Analysis and Recognition (ICDAR), 20–25 Sept. 2019, pp. 1090–1096.
- Dachanuwattana, Silpakorn, Jin, Jianli, Zuloaga-Molero, Pavel, et al., 2018. Application of proxy-based MCMC and EDFM to history match a Vaca Muerta shale oil well. *Fuel* 220, 490–502.
- Elsheikh, Ahmed H., Hoteit, Ibrahim, Wheeler, Mary F., 2014. Efficient Bayesian inference of subsurface flow models using nested sampling and sparse polynomial chaos surrogates. *Comput. Methods Appl. Mech. Eng.* 269, 515–537.
- Emerick, Alexandre A., Reynolds, Albert C., 2013. Ensemble smoother with multiple data assimilation. *Comput. Geosci.* 55, 3–15.
- Goodfellow, Ian J., Pouget-Abadie, Jean, Mirza, Mehdi, et al., 2014. Generative adversarial nets. In: Paper Presented at the Proceedings of the 27th International Conference on Neural Information Processing Systems, ume 2 (Montreal, Canada).
- 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.
- Graves, Alex, Schmidhuber, Jürgen, 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Network.* 18 (5), 602–610.
- He, K., Zhang, X., Ren, S., et al., 2016. Deep residual learning for image recognition. In: Proc., 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 27–30 June 2016, pp. 770–778.
- Himmelblau, D.M., 1972. Applied Nonlinear Programming. McGraw-Hill, New York.
- Hochreiter, Sepp, Schmidhuber, Jürgen, 1997. Long short-term memory. *Neural Comput.* 9 (8), 1735–1780.
- Kinghorn, Philip, Zhang, Li, Shao, Ling, 2018. A region-based image caption generator with refined descriptions. *Neurocomputing* 272, 416–424.
- 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.
- Li, Heng, Sarma, Pallav, Zhang, Dongxiao, 2011. A comparative study of the probabilistic-collocation and experimental-design methods for petroleum-reservoir uncertainty quantification. *SPE J.* 16 (2), 429–439.
- Li, Weixuan, Lin, Guang, 2015. An adaptive importance sampling algorithm for Bayesian inversion with multimodal distributions. *J. Comput. Phys.* 294, 173–190.
- Liao, Q., Zhang, D., 2015. Data assimilation for strongly nonlinear problems by transformed ensemble Kalman filter. *SPE J.* 20 (1), 202–221.
- Liao, Qinzhao, Zeng, Lingzao, Chang, Haibin, et al., 2019. Efficient history matching using the markov-chain Monte Carlo method by means of the transformed adaptive stochastic collocation method. *SPE J.* 24 (4), 1468–1489.
- Liu, N., Oliver, D.S., 2005. Ensemble Kalman filter for automatic history matching of geological facies. *J. Petrol. Sci. Eng.* 47 (3–4), 147–161.
- Ma, Xiaopeng, Zhang, Kai, Yao, Chuanjin, et al., 2020. Multiscale-network structure inversion of fractured media based on a hierarchical-parameterization and data-driven evolutionary-optimization method. *SPE J.* 25 (5), 2729–2748.
- Ma, Xiaopeng, Zhang, Kai, Zhang, Liming, et al., 2021. Data-driven niching differential evolution with adaptive parameters control for history matching and uncertainty quantification. *SPE J.* 26 (2), 993–1010.
- Oliver, Dean S., Chen, Yan, 2011. Recent progress on reservoir history matching: a review. *Comput. Geosci.* 15 (1), 185–221.
- Oliver, Dean S., Reynolds, A.C., et al., 2008. Inverse Theory for Petroleum Reservoir Characterization and History Matching. Cambridge University Press, Cambridge.
- Peaceman, D.W., 1978. Interpretation of well-block pressures in numerical reservoir simulation(includes associated paper 6988). *Soc. Petrol. Eng. J.* 18 (3).
- Peters, E., Chen, Yunxia, Leeuwenburgh, O., et al., 2013. Extended Brugge benchmark case for history matching and water flooding optimization. *Comput. Geosci.* 50, 16–24.

- Raskutti, G., Wainwright, M.J., Yu, B., 2011. Early stopping for non-parametric regression: an optimal data-dependent stopping rule. In: Proc., 2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 28-30 Sept. 2011, pp. 1318–1325.
- 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.
- Saad, George, Ghanem, Roger, 2009. Characterization of reservoir simulation models using a polynomial chaos-based ensemble Kalman filter. *Water Resour. Res.* 45 (4).
- Sagheer, Alaa, Kotib, Mostafa, 2019. Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing* 323, 203–213.
- Stuart, A.M., 2010. Inverse problems: a Bayesian perspective. *Acta Numer.* 19, 451–559.
- Tan, Ying Hua, Chan, Chee Seng, 2019. Phrase-based image caption generator with hierarchical LSTM network. *Neurocomputing* 333, 86–100.
- 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.
- Xue, L., Gu, S., Mi, L., Zhao, L., Liu, Y., Liao, Q., 2022. An automated data-driven pressure transient analysis of water-drive gas reservoir through the coupled machine learning and ensemble Kalman filter method. *J. Petrol. Sci. Eng.* 208, 109492.
- Yang, Q., Chen, W., Li, Y., et al., 2017. Multimodal estimation of distribution algorithms. *IEEE Trans. Cybern.* 47 (3), 636–650.
- Zeng, Lingzao, Zhang, Dongxiao, 2010. A stochastic collocation based Kalman filter for data assimilation. *Comput. Geosci.* 14 (4), 721–744.
- Zhang, Jiangjiang, Jun, Man, Guang, Lin, et al., 2018. Inverse modeling of hydrologic systems with adaptive multifidelity Markov chain Monte Carlo simulations. *Water Resour. Res.* 54 (7), 4867–4886. <https://doi.org/10.1029/2018WR022658>.
- Zhang, Kai, Wang, Xiaoya, Ma, Xiaopeng, et al., 2022. The prediction of reservoir production based proxy model considering spatial data and vector data. *J. Petrol. Sci. Eng.* 208, 109694.
- Zhao, Zijing, Kumar, Ajay, 2019. A deep learning based unified framework to detect, segment and recognize irises using spatially corresponding features. *Pattern Recogn.* 93, 546–557.
- Zhong, Zhi, Sun, Alexander Y., Ren, Bo, et al., 2021. A deep-learning-based approach for reservoir production forecast under uncertainty. *SPE J.* 1–27.
- 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.
- Zhu, Yinhao, Zabaras, Nicholas, 2018. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *J. Comput. Phys.* 366, 415–447.