An Improved Tandem Neural Network Architecture for Inverse Modeling of Multicomponent Reactive Transport in Porous Media

Junjun Chen1, Zhenzue Dai1, Zhijie Yang1, Yu Pan1, Xiaoying Zhang1, Jichun Wu2, and Mohamad Reza Soltanian4

1College of Construction Engineering, Jilin University, Changchun, China, 2School of Foreign Language Education, Jilin University, Changchun, China, 3School of Earth Sciences and Engineering, Key Laboratory of Surficial Geochemistry of Ministry of Education, Nanjing University, Nanjing, China, 4Departments of Geology and Environmental Engineering, University of Cincinnati, Cincinnati, OH, USA

Abstract Parameter estimation for reactive transport models (RTMs) is important in improving their predictive capacity for accurately simulating subsurface hydrogeochemical processes. This paper introduces a deep learning approach called the tandem neural network architecture (TNNA), which consists of a forward network and a reverse network to estimate input parameters for RTMs. The TNNA approach has a limitation in that the approximation error from the forward network often results in biased inversion results. To solve this problem, we proposed to enhance TNNA using an adaptive updating strategy (AUS), which locally reduces the approximation error of the forward network. The developed framework updates the forward network by iteratively using local sampling and transfer learning. The TNNA-AUS was verified by a cation exchange example. The results show that TNNA-AUS successfully reduces the inversion bias and improves the computational efficiency and inversion accuracy, compared with the global improvement strategy of adding training samples according to the prior distribution of model parameters. After verification, the TNNA-AUS was applied to a real-world and well-documented RTM problem of the Aquia aquifer, Maryland, USA. The inversion results demonstrate that the developed TNNA-AUS algorithm is an excellent tool for us to understand the complex subsurface hydrogeochemical processes and estimate the associated reaction parameters.

1. Introduction

Reactive transport models (RTMs) are vital tools for understanding solute transport and geochemical reaction processes in subsurface systems and terrestrial environments (Dai & Samper, 2004; Li et al., 2017; Meile & Scheibe, 2019; Sanchez-Vila et al., 2010; Seigneur et al., 2019; Sole-Mari et al., 2020; Steefel et al., 2005; Steefel et al., 2014; Steefel & Maher, 2009; Xu et al., 1999, 2011; Zhou & Zhan, 2018). These numerical models can reveal complex subsurface system processes at various spatial and temporal scales. RTMs have received significant attention recently and have been used in various areas, including nuclear waste disposal (Dai et al., 2009; De Windt & Spycher, 2019; Iwalewa & Farman, 2019; Viswanathana et al., 1998; Wolfsberg et al., 2017), geological carbon storage (Dai et al., 2014, 2020; Celia, 2017; Harrison et al., 2019), chemical weathering (Godderis et al., 2019; Heidari et al., 2017; Navarre-Sitchler et al., 2011), groundwater quality prediction (Ng et al., 2015; Prommer et al., 2019; Salmon et al., 2014), and hyporheic exchange (Dwivedi et al., 2018; Li et al., 2020; Pescimoro et al., 2019).

A significant challenge for developing effective RTMs is parameter estimation, since the existence of large uncertainties associated with reaction processes and the associated parameters in geochemical systems (Ampomah et al., 2017; Dai & Samper, 2006; Fakhreddine et al., 2016; McGrail, 2001; Oyanagi et al., 2020; Patil & McPherson, 2020; Ritschel & Totsche, 2016; Samper et al., 2008; Soltanian et al., 2015; Suescún-Díaz et al., 2021; Vasco et al., 2018). Measurements of RTM parameters (e.g., selectivity coefficients and initial concentrations of chemical components) are often complicated, time-consuming, and expensive (Dai & Samper, 2006; Xu et al., 1999). Thus, parameter estimation through inverse modeling has been proposed as a potential solution to enhance the application and accuracy of RTMs (Dai & Samper, 2004; Mo, Zabaras, et al., 2019; Sun, 2013; Zhang et al., 2020).
Over the past decades, numerous inverse modeling methods have been developed through data assimilation frameworks, where model parameters and system states are adjusted using available system observations and mathematical models (Q. He et al., 2020; Rajabi et al., 2018). They mainly include local optimization (Bjarkason et al., 2018; Dai et al., 2012; Munavalli & Mohan Kumar, 2005; Yang et al., 2014), global optimization (Bahraini et al., 2016; Chen et al., 2020; Gill et al., 2006), Bayesian Markov chain Monte Carlo (MCMC; Smith & Marshall, 2008; Vrugt, 2016; Vrugt et al., 2003; J. Zhang, Man, et al., 2018), and ensemble Kalman filter and its variants (Chen & Zhang, 2006; Emerick & Reynolds, 2013; Song et al., 2014, 2019; J. Zhang, Lin, et al., 2018). In these algorithms, a large number of forward model realizations are required. Hence, for CPU-intensive high-fidelity numerical models, the computational cost of inversion frameworks is prohibitive (Razavi et al., 2012; L. Yan & Zhou, 2019a; J. Zhang et al., 2020). Thus, there is a significant need for adopting surrogate models to approximate high-fidelity model outputs at lower computational costs (Asher et al., 2015; Razavi et al., 2012).

Several surrogate model methods have been proposed in prior studies. This includes and is not limited to polynomial chaos expansion (Laloy et al., 2013; Xiu & Karniadakis, 2002), Gaussian process (H. Wang & Li, 2018; J. Zhang et al., 2016), Kriging surrogate modeling (X. Yan et al., 2019; J. Zhou et al., 2018), support vector machine (Lal & Datta, 2018; Xingpo et al., 2021), conventional artificial neural network (i.e., single hidden layer neural network; Kourakos & Mantoglou, 2009; Shin et al., 2019), and radial basis function (Y. Liu, Wang, et al., 2019; Xing et al., 2019). Unfortunately, these conventional methods are faced with the problem of dimensionality (i.e., “curse of dimensionality”), since the computational cost for constructing surrogate models increases exponentially as the input dimensionality increases (Asher et al., 2015; Liao et al., 2017; Mo, Zabaras, et al., 2019). Convergence difficulties also exist if the relationship between model parameters and responses is strongly nonlinear (Mo, Zabaras, et al., 2019).

Deep neural network (DNN; i.e., deep learning) is a newly developed artificial intelligence technology (Goodfellow et al., 2016; LeCun et al., 2021). Previous studies have shown its good performance in dealing with high-dimensional and nonlinear problems (Shen, 2018; Sit et al., 2020). Related applications of DNN are available in various fields of hydrology and water resources, such as water storage measurements (Elneter et al., 2021; Irrgang et al., 2020; A. Y. Sun, Scanlon, et al., 2019), meteorological forecasting (Pudashine et al., 2020; F. Wang et al., 2021), flood predictions (Berkhahn et al., 2019; Kabir et al., 2020), and geological parameterizations (Laloy et al., 2018; Tang et al., 2021).

For surrogate model construction, DNNs are generally classified into data-driven and theory-guided approaches (Tang et al., 2020). In the data-driven approach, DNNs are trained purely based on the labeled data. For example, Mo, Zabaras, et al. (2019), Mo, Zhu, et al. (2019), and Mo et al. (2020) constructed surrogate models of multiphase flow and contaminant transport in heterogeneous media through the convolutional neural network (CNN)-based image regression procedure. Dagasan et al. (2020) used conditional generative adversarial networks to build a surrogate model for the study of inverse modeling of subsurface flow. A. Y. Sun (2018), A. Y. Sun, Scanlon, et al. (2019), A. Y. Sun, Zhong, et al. (2019), and L. Sun et al. (2019) used a conditional deep convolutional generative adversarial network to construct a surrogate model for modeling subsurface flow and carbon dioxide diffusion. While in the theory-guided approach, the governing equations are utilized as the constrain for DNN training instead of the labeled training data set. An advantage of this approach is that the labeled training data set does not need to be prepared through high-fidelity model realizations (Bandai & Ghezzehei, 2021; Q. He et al., 2020; Meng & Karniadakis, 2020; Raisi et al., 2019; L. Sun et al., 2019; N. Wang, et al., 2020). However, for models with high-order partial differential equations, DNNs may have convergence-related issues (R. Xu et al., 2020). Thus, current applications of theory-guided neural networks were mostly single-process models. More information could be found in prior works by Raisi et al. (2019), Tartakovsky et al. (2020), Q. He et al. (2020), A. Y. Sun, Scanlon, et al. (2019), A. Y. Sun, Zhong, et al. (2019), L. Sun et al. (2019), N. Wang et al. (2020), and N. Wang et al. (2021).

In recent inverse modeling studies cited above, the DNNs were mainly used to construct surrogate models and model parameters were estimated using conventional data assimilation methods such as MCMC or ensemble Kalman filter. In this study, another inversion strategy is introduced and the DNNs are used for parameter estimation in addition to surrogate modeling. Specifically, a forward network and a reverse network are constructed respectively for surrogate modeling and parameter estimation. Here, the inverse modeling framework is called the tandem neural network architecture (TNNA). This idea of TNNA is newly proposed and has been successfully developed for computed tomography reconstruction (Adler & Öktem, 2017), nanophotonic structure inverse
design (D. Liu et al., 2018; X. Xu et al., 2021; Yeung et al., 2021), and photonic topological state inverse design (Long et al., 2019).

The original TNNA method has a limitation in that its inverse modeling accuracy becomes highly dependent on the accuracy of forward network (i.e., surrogate model). Inevitably, a surrogate model has a large local approximate error near the true parameters, which will lead to biased inversion results (J. Zhang et al., 2020). One solution for this issue is to implement a two-stage data assimilation procedure. The parameter space is explored sufficiently with the surrogate model at stage 1 and then corrected by the high-fidelity model at stage 2 (Ba et al., 2018; Efendiev et al., 2005; Laloy et al., 2013; Zeng et al., 2012; J. Zhang et al., 2017). Nevertheless, extra high-fidelity numerical simulations are also required and the computational burden is still heavy. Another strategy is to use more training samples of prior distribution to construct a global accurate surrogate model. This approach requires more computational burden for a large amount of additional training samples in most cases. Moreover, the problem of dimensionality will be prominent with the increasing of model parameter number mentioned above. Generally, high probability regions of model parameters are usually a small area of prior parameter space (Gong & Duan, 2017; Gong et al., 2015; J. Zhang et al., 2020; J. Zhou et al., 2018). That is to say, the inverse modeling results of TNNA are determined by the approximate accuracy of forward network in this small area. Thus, the adaptive updating strategy (AUS) is introduced to force the surrogate model to be locally accurate in the region around true parameter values (Bacigalupo et al., 2021; C. Wang et al., 2014; G. Zhang et al., 2013). In each adaptive iteration, a small number of new training samples around the parameter estimation results are generated for updating the surrogate model and the local accuracy of the surrogate model is expected to be enhanced gradually (J. Zhang et al., 2020). AUS includes at least three strategies, which are retraining a surrogate model based on a new training data set (Gong & Duan, 2017; Gong et al., 2015; Xiao et al., 2020), second surrogate model constructing (L. Yan & Zhou, 2019a; J. Zhang et al., 2020), and multifidelity surrogate model constructing (Rumpfkeil & Beran, 2020; L. Yan & Zhou, 2019b; Zheng et al., 2019). Following the idea of the first AUS strategy, an improved TNNA method (i.e., TNNA-AUS) has been developed and tested in this paper. Since the reactive transports are coupled with multiple processes whose governing equations are complex, the forward network here was trained through the data-driven approach. To our knowledge, there is no study available on TNNA-AUS method in subsurface inverse modeling specifically for complex RTMs that involve multiple physical and chemical processes.

This paper is organized as follows. In Section 2, the mathematical principles and algorithm design of TNNA-AUS for inverse modeling of RTMs are described. In Section 3, a synthetic model of the cation exchange column experiment is presented and used for method verification. Then, the application of TNNA-AUS for a real-world RTM of the Aquia aquifer is evaluated. Finally, Section 4 presents the discussion and conclusions of this study.

2. Methodology

2.1. TNNA Method for Inverse Modeling

A DNN is a specific type of nonlinear multiple regression model with a hierarchy of layers (i.e., vectors or matrices) connected by trainable parameters (i.e., weights and bias). The DNN training is to search possible sets of trainable parameters for satisfying predictions (Gu et al., 2018; K. He et al., 2016; Huang et al., 2017; LeCun et al., 2015). This study aims to search a reverse network ($F_{\text{Reverse}}$), which is able to map observation vector to model parameters $m \in \mathbb{R}^{N_{\text{m}}\times 1}$ as shown in Equation 1.

$$m = F_{\text{Reverse}}(\tilde{y}_{\text{obs}}, \theta_{\text{Reverse}})$$

(1)

where $\theta_{\text{Reverse}}$ is the trainable parameters of the $F_{\text{Reverse}}$. While for simplicity, the subsurface system can be represented in the following form:

$$\tilde{y}_{\text{obs}} = F_{HF}(m) + \epsilon$$

(2)

where $\epsilon \in \mathbb{R}^{N_{\text{obs}}\times 1}$ is the vector of observation errors, and $F_{HF}()$ is the operator of high-fidelity model. In Bayes’ theorem, the inversion problem is equivalent to an optimization problem of maximizing the posterior distribution probability function $p(m \mid \tilde{y}_{\text{obs}})$ (Dai & Samper, 2004; Vrugt, 2016; J. Zhang et al., 2020):
$m = \arg\max \ p(m \mid \tilde{y}_{obs})$

$$p(m \mid \tilde{y}_{obs}) = \frac{p(m)p(\tilde{y}_{obs} \mid m)}{\int p(\tilde{y}_{obs} \mid m) \, dm} = p(m)\mathcal{L}(m \mid \tilde{y}_{obs})$$ (3)

where $p(m)$ is the prior distribution of $m$, $\mathcal{L}(m \mid \tilde{y}_{obs}) = p(\tilde{y}_{obs} \mid m)$ is the likelihood function, and $p(\tilde{y}_{obs}) = \int p(\tilde{y}_{obs} \mid m) \, dm$ is the marginal likelihood (i.e., a constant value). The likelihood function is determined by the probability distribution of $\varepsilon$. If $\varepsilon$ obeys the Gaussian distribution as it is assumed in this paper, the likelihood function is expressed by

$$\mathcal{L}(m \mid \tilde{y}_{obs}) = \prod_{i=1}^{N_{obs}} \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left[-\frac{1}{2} \left(\tilde{y}_{obs}[i] - F_{HF}(m)\right)^2\right]$$ (4)

where $\sigma_i$ is the standard deviation of the $i$th observation, $\tilde{y}_{obs}[i]$ is the $i$th element in $\tilde{y}_{obs}$ vector, and $F_{HF}(m)$ is the $i$th element in the output vector of $F_{HF}(m)$. When the prior information of parameters is an interval form of uniform distribution, the inverse problem is transferred to the L2-norm-based optimization problem (Equation 5).

$$m = \arg\max \sum_{i=1}^{N_{obs}} \frac{1}{\sigma_i} \left[\tilde{y}_{obs}[i] - F_{HF}(m)\right]^2$$ (5)

In the TNNA inversion method, $m$ is proposed to be calculated by $F_{Reverse}$ according to Equation 1. As $m$ is also the input of $F_{HF}(.)$, $\theta_{Reverse}$ can be optimized through Equation 6 (i.e., the loss function for training $F_{Reverse}$).

$$\theta_{Reverse} = \arg\min \sum_{i=1}^{N_{obs}} \frac{1}{\sigma_i} \left[\tilde{y}_{obs}[i] - F_{HF}(F_{Reverse}(\tilde{y}_{obs}, \theta_{Reverse}))\right]^2$$ (6)

During $\theta_{Reverse}$ optimization, large amount of $F_{HF}(.)$ simulations is required. As described in Section 1, $F_{HF}(.)$ are usually computationally intensive. Hence, a forward network ($F_{Forward}$) for surrogate modeling is pretrained as Equation 7.

$$F_{Forward}(m, \theta_{Forward}) \approx F_{HF}(m) = \tilde{y}$$ (7)

where $\tilde{y} \in \mathbb{R}^{N_{obs}}$ is the output of $F_{HF}(m)$; $\theta_{Forward}$, as the trainable parameters of the $F_{Forward}$, is optimized through the regularized L1-norm as shown in Equation 8 (i.e., the loss function for training $F_{Forward}$).

$$\theta_{Forward} = \arg\min \frac{1}{N_{sample}} \sum_{i=1}^{N_{sample}} \left|F_{Forward}(m, \theta_{Forward}) - \tilde{y}\right|$$ (8)

where $N_{sample}$ is the number of samples in training data set; $m$ and $\tilde{y}$ are $i$th sample in training data set. The preparation for training data set is given in Section 2.3. Then, Equation 6 is transferred into Equation 9.

$$\theta_{Reverse} = \arg\min \sum_{i=1}^{N_{obs}} \frac{1}{\sigma_i} \left[\tilde{y}_{obs}[i] - F_{Forward}(F_{Reverse}(\tilde{y}_{obs}, \theta_{Reverse}), \theta_{Forward})\right]^2$$ (9)

where $F_{Forward}$ is the $i$th value in the output vector of $F_{Forward}$. Equation 9 shows the $F_{Reverse}$ and $F_{Forward}$ are connected in a TNNA as illustrated in Figure 1. After obtaining $F_{Forward}$, $\theta_{Forward}$ will be fixed and $\theta_{Reverse}$ will be optimized (D. Liu et al., 2018; Long et al., 2019).
The trainable parameters $\theta$ of a DNN are optimized following a cooperative strategy of error backpropagation and stochastic gradient descent (SGD) method (e.g., Adam, AdaGrad, and RMSProp; Goodfellow et al., 2016). The gradient is calculated through backpropagation. By taking a negative step along the gradient direction, the trainable parameter $\theta$ is updated at each iteration according to Equation 10.

$$\theta_{k+1} = \theta_k - \epsilon \nabla_{\theta_k} J(\theta_k)$$

where $\epsilon$ is the learning rate; $\theta_k$ and $\theta_{k+1}$ are the trainable parameters at $k$th and $k + 1$st iteration, respectively; $J$ represents the loss function for a target DNN (i.e., Equation 8 for $F_{\text{forward}}$ or Equation 9 for $F_{\text{Reverse}}$ in this paper). $J$ will be minimized after a certain number of iterations. Here, the Adam algorithm is chosen as the SGD optimization algorithm, which is able to adjust $\epsilon$ adaptively. The training data set is divided into multiple batches and $\theta$ is optimized through batch enumerations. A complete batch enumeration is called an epoch in deep learning. Generally, tens or hundreds of epochs are needed to ensure a reliable performance of a DNN (Goodfellow et al., 2016; LeCun et al., 2015; L. Yan & Zhou, 2019b).

2.2. TNNA With Adaptive Updating Strategy

As described in Section 1, the inversion result of TNNA is related to the approximate accuracy of $F_{\text{forward}}$ around the true value of $m$. Here, the AUS procedure to improve the local approximate accuracy of $F_{\text{forward}}$ (i.e., TNNA-AUS) is introduced.

First, the initial forward network, $F_{\text{forward}(0)}$, is trained using an initial training data set $D_{(0)} = \{M_{(0)}, Y_{(0)}\}$ with $N_{\text{sample(0)}}$ samples according to Equation 8. Then, obtain the initial reverse network, $F_{\text{reverse}(0)}$, and calculate the initial parameter estimation, $\hat{m}_0$. Generally, $\hat{m}_0$ may not be accurate enough as $F_{\text{forward}(0)}$ could have a large approximation error around the true value of the model parameter ($m_{true}$). Around $\hat{m}_0$, we can draw $N_{\text{sample(loc)}}$ new design points ($M_{(loc)}$), that are expected to be closer to the real parameter point than $M_{(0)}$. Then, a small local data set $D_{(loc)} = \{M_{(loc)}, Y_{(loc)}\}$ with $N_{\text{sample(loc)}}$ new samples is generated, where $Y_{(loc)}$ is the set of $M_{(loc)}$ corresponding high-fidelity model outputs. By adding $D_{(loc)}$ to $D_{(0)}$ and abandoning the $N_{\text{sample(loc)}}$ samples in $D_{(0)}$ with the largest Euclidean distance from $\hat{m}_0$, the $D_{(0)}$ is updated into a new training data set $D_{(1)} = \{M_{(1)}, Y_{(1)}\}$. The sample distribution of $D_{(1)}$ near $m_{true}$ is expected to be denser than that of $D_{(0)}$. Thus, $F_{\text{forward}(1)}$ updated from $D_{(1)}$ is considered to have a higher approximate accuracy near $m_{true}$ than $F_{\text{forward}(0)}$. As most of the training samples will not be changed in $D_{(1)}$, the transfer learning method is adopted for $F_{\text{forward}}$ training (Lu et al., 2015; Zhuang et al., 2020). It is a new machine learning technology aiming at solving new but similar problems by using previously gained knowledge. Comprehensive reviews on transfer learning are available in prior studies (Lu et al., 2015; Zhuang et al., 2020). Here, for update $F_{\text{forward}(0)}$ into $F_{\text{forward}(1)}$, only trainable parameters in the last layer of $F_{\text{forward}(0)}$ are trained. In this way, the updating process is kept at a higher level of efficiency. After we update $F_{\text{forward}(0)}$ into $F_{\text{forward}(1)}$, $F_{\text{Forward}(1)}$ and $\hat{m}_1$ are also obtained. The rest of the process for $F_{\text{forward}(0)}$, $F_{\text{reverse}(0)}$, and $\hat{m}_1$ is conducted following an iteration procedure. The termination condition for the iteration procedure is based on a threshold of the approximation error of $F_{\text{Forward}(i)}$ at $i$th iteration ($e_{\text{Forward}(i)}$) or the maximum iteration number $I_{\text{max}}$. Note that the $e_{\text{Forward}(i)}$ of each iteration may not follow the trend of an absolute decline. If the iteration procedure is stopped by $I_{\text{max}}$, the final inverse modeling results are the minimum $e_{\text{Forward}(i)}$-corresponded parameter estimation values. The computer program for TNNA-AUS is developed following the algorithm described in Appendix A.

2.3. Training Data Set Preparation

In this study, the data set ($D = \{M, Y\}$) for training $F_{\text{Forward}}$ contains mappings from RTM parameters to chemical component concentrations. $M = \{m_1, ..., m_{N_{\text{Sample}}}\}$ are $N_{\text{Sample}}$ RTM parameter samples randomly drawn from the prior distribution and $Y = \{Y_1, ..., Y_{N_{\text{Sample}}}\}$ are their corresponded model responses. $m_i \in \mathbb{R}^{N \times 1}$ in $M$ is a vector of RTM parameter with $N_{m}$ parameters. $Y_i \in \mathbb{R}^{N \times N_{Y_i}}$ in $Y$ is a matrix of RTM outputs, which include the concentration values of $N$ components at $N$ different observation times or locations. While the training data set for $F_{\text{Reverse}}$ is a single matrix of $Y_{\text{obs}} \in \mathbb{R}^{N \times N_{Y_{\text{obs}}}}$.

Typically, RTM parameters and concentrations of different chemical components usually vary within different logarithmic ranges. It will make the network training process difficult to converge (Dai & Samper, 2004; Pal et al., 2019). Therefore, data normalization is performed on $M$, $Y$, and $Y_{\text{obs}}$ for adopting TNNA-AUS in RTM.
inversion. Let \( \tilde{y}_j^l \) and \( \tilde{y}_{\text{obs}}^j \) be the model response vector of the \( j \)th chemical component at \( N_j \) different times or observation locations in \( Y \) and \( Y_{\text{obs}} \). Then, the vectors of the normalized parameters \( (m_j^{\text{Norm}}) \), model responses \( (\tilde{y}_j^{\text{Norm}}) \), and observations \( (\tilde{y}_{\text{obs}}^j)^{\text{Norm}} \) of the \( j \)th chemical component are expressed using Equations 11–13, respectively.

\[
m_j^{\text{Norm}} = \frac{m_j - m_j^{\text{max}}}{m_j^{\text{max}} - m_j^{\text{min}}} \tag{11}
\]

\[
\tilde{y}_j^{\text{Norm}} = \frac{\tilde{y}_j - \tilde{y}_j^{\text{max}}}{\tilde{y}_j^{\text{max}} - \tilde{y}_j^{\text{min}}} \tag{12}
\]

\[
\tilde{y}_{\text{obs}}^j = \frac{\tilde{y}_{\text{obs}}^j - \tilde{y}_{\text{obs}}^{j\text{max}}}{\tilde{y}_{\text{obs}}^{j\text{max}} - \tilde{y}_{\text{obs}}^{j\text{min}}} \tag{13}
\]

where \( m_j^{\text{max}} \) and \( m_j^{\text{min}} \) are vectors consisting of the maximum and minimum values of each parameter respectively. Generally, \( m_j^{\text{max}} \) and \( m_j^{\text{min}} \) are set as the upper and lower bound of the prior interval of \( m \). \( \tilde{y}_j^{\text{max}} \) and \( \tilde{y}_j^{\text{min}} \) are the maximum and minimum values of the \( j \)th chemical component model response, respectively. After data normalization, values of model parameter and response vectors are between 0 and 1, and they are saved into a single vector. The normalized training data set for \( F_{\text{Forward}} \) is represented as \( D^{\text{Norm}} = \{M^{\text{Norm}}, Y^{\text{Norm}}\} \), where \( M^{\text{Norm}} = \{m_1^{\text{Norm}}, \ldots, m_n^{\text{Norm}}\} \) is the normalized model parameter vector group and \( Y^{\text{Norm}} = \{\tilde{y}_1^{\text{Norm}}, \ldots, \tilde{y}_m^{\text{Norm}}\} \) is the normalized model response vector group. Here, \( \tilde{y}_i^{\text{Norm}} \in \mathbb{R}^{N_{\text{obs}} x 1}, (i = 1, \ldots, N_{\text{sample}}) \) is the \( i \)th sample of training in the model response vector, and \( N_{\text{obs}} \) (equals to the value of \( N_c \times N_j \)) is total number of observations. Similarly, the normalized training data set for \( F_{\text{Reverse}} \) is a vector of \( \tilde{y}_{\text{obs}}^{\text{Norm}} \in \mathbb{R}^{N_{\text{obs}} x 1}, (i = 1, \ldots, N_{\text{sample}}) \). Using \( D^{\text{Norm}} \) and \( \tilde{y}_{\text{obs}}^{\text{Norm}} \), we can get a normalized parameter estimation \( (\tilde{m})^{\text{Norm}} \) and the final inversion result \( (\tilde{m}) \) is obtained according to Equation 14.

\[
\tilde{m} = (m^{\text{max}} - m^{\text{min}}) \times \tilde{m}^{\text{Norm}} + m^{\text{max}} \tag{14}
\]

### 2.4. Candidate DNN Architectures

The \( F_{\text{Forward}} \) and \( F_{\text{Reverse}} \) are constructed using the DNNs with a same architecture in this paper. The mathematic formula for a DNN with \( L_{\text{NN}} \) layers is expressed as Equation 15.

\[
y = f_{\text{DNN}}(z, \theta) = f_{L_{\text{NN}}}(...f_1(...f_1(z))) \tag{15}
\]

where \( f_{\text{DNN}}(.) \) represents the operator of a DNN and \( f_i(.) (i = 1, \ldots, L_{\text{NN}}) \) is the operator of \( i \)th layer. \( z \) and \( y \) are the input and expected output, respectively. The layers excluding input and output layer are called the hidden layers.

One of the most commonly used types of network layer for constructing DNNs is the fully connected layer, whose formula for the connection of two adjacent layers is expressed as

\[
z^{(l)} = f_j(z^{(l-1)}) = \sigma_j(w^{(l)} z^{(l-1)} + b^{(l)}) \tag{16}
\]

where \( w^{(l)} \in \mathbb{R}^{N_{l} x N_{l-1}} \) and \( b^{(l)} \in \mathbb{R}^{N_{l} x 1} \) are the weight matrix and bias vector of \( l \)th layer in DNN, respectively; \( z^{(l-1)} \in \mathbb{R}^{N_{l-1} x 1} \) and \( z^{(l)} \in \mathbb{R}^{N_{l} x 1} \) are the input and output vectors of the \( l \)th layer, respectively; \( \sigma_j(.) \) is a nonlinear active function of the \( l \)th layer, such as Sigmoid, ReLU, and Tanh functions (Goodfellow et al., 2016; Mo, Zabaras, et al., 2019; Mo, Zhu, et al., 2019; A. Y. Sun, Zhong, et al., 2019).

Another type of network layer is the convolutional layer, whose input and output are in matrix form. Each convolutional layer is composed of several convolution kernels, which are small matrices essentially, for computing features of the input layer. If \( x_{\alpha} \) is assumed as a pixel of the input layer matrix, the output feature \( h^q_{\alpha} \) at location \( (\alpha, \nu) \) can be obtained by employing a series of \( q = 1, \ldots, N_{\text{base}} \) convolutional kernel filters \( w^q \in \mathbb{R}^{k_x x k_y} \).

Where, \( k_x \times k_y \) represents the kernel size.
After a convolutional layer operation with \( N_{out} \) kernel filters, an output layer \( h \) with \( N_{out} \) feature maps is drawn:

\[
h_q(x_{u,v}) = f \left( \sum_{i=1}^{N_{in}} \sum_{j=1}^{N_{in}} w_{u+1,v+1} x_{u+i,v+j} + b \right)
\]

(17)

Figure 2. Fully connected deep neural network (FC-DNN)-based TNNA. The forward network and reverse network are constructed by multilayer FC-DNNs.

For this study, a fully connected deep neural network (FC-DNN) and CNNs can be designed. The difference between them is whether the convolution layers are used.

The FC-DNN is constructed using fully connected layers (Figure 2). Three hidden layers are designed in \( F_{\text{Forward}} \) and \( F_{\text{Reverse}} \) respectively. Each hidden layer consists of 256 neurons. The activation functions of the output layer is Sigmoid and the other hidden layers are Swishs (Ramachandran et al., 2017; Tripathy & Bilionis, 2018; L. Yan & Zhou, 2019b). The expressions of Sigmoid and Swish are as Equations 18 and 19, respectively.

\[
\sigma_{\text{Sigmoid}}(x) = \frac{1}{1 + e^{-x}}
\]

(18)

\[
\sigma_{\text{Swish}}(x) = x \cdot \sigma_{\text{Sigmoid}}(\beta \cdot x)
\]

(19)

where \( \beta \) is an adjustable parameter and is set as a constant of 1 in this paper. Batch normalization (BN) layers are set in each hidden layer, to avoid overfitting, gradient disappearance, and gradient explosion (Pan et al., 2019; Ramachandran et al., 2017; Tripathy & Bilionis, 2018; L. Yan & Zhou, 2019b).

The CNN architectures for \( F_{\text{Reverse}} \) and \( F_{\text{Forward}} \) are shown in Figure 3a and 3b, respectively. Aside from the input and output layers, \( F_{\text{Forward}} \) and \( F_{\text{Reverse}} \) respectively consist of one preprocess block, three convolutional blocks, and one fully connected block. In CNNs, the output size \( S_{out} \) of each convolutional layer is determined by the input size \( S_{in} \) and the hyperparameters (i.e., zero padding \( p \), kernel size \( k' \) and stride \( s \)) as shown in Equation 20.
Since $S_{\text{in}}$ is case dependent, the CNN will encounter a mismatch among layer connections if a fixed CNN is used for different cases. Hence, the preprocess block here is designed to transfer the input layer with any shape into a fixed shape. Then, the inversion program becomes universally applicable.

By applying different preprocessing layers, the input data are processed into one-dimensional (1D) or two-dimensional (2D) data with fixed shapes. Specifically, the 1D preprocess block is a fully connected layer, followed by a dimension expand operator, while the 2D preprocess block starts from an upsampling operator and followed by a convolutional layer. After preprocessing, the $F_{\text{Forward}}$ and $F_{\text{Reverse}}$ are constructed by 1D-CNN or 2D-CNN, respectively. The convolutional block consists of a convolutional layer and a maxpooling layer. Each of these two layers is followed by a BN layer. The maxpooling layers and BN layers in CNNs are able to avoid overfitting, gradient disappearance, and gradient explosion (Goodfellow et al., 2016; Gu et al., 2018; D. Zhou, 2020). The fully connected block that consists a flatten layer and a fully connected layer is designed to transfer the data to the output vector. The detailed information about the blocks for building 1D-CNN and 2D-CNN is displayed in Figure 3 and Table 1.

In this study, the approximation accuracy of FC-DNN, 1D-CNN, and 2D-CNN will be compared in Section 3.1.2 and the best of them will be selected for $F_{\text{Forward}}$ and $F_{\text{Reverse}}$.

### 3. Verification and Applications

The proposed TNNA-AUS inverse modeling framework is first verified by a synthetic example, involving a cation exchange column experiment. After evaluating TNNA-AUS performance through the column experiment example, we will show its application in the well-documented RTM of Aquia aquifer, which includes several hydrogeochemical processes. The RTMs are solved in TOUGHREACT, a verified and well-documented numerical...
3.1. Verification Example

3.1.1. Problem Description

This synthetic example is a 0.08-m-long column filled with a porous material. The initial dissolved chemical components are 1 mmol/L NaNO$_3$ and 0.2 mmol/L KNO$_3$. The column is flushed by 0.6 mmol/L CaCl$_2$ at a constant rate of 0.1 m/hr. The porosity and solid density of porous media are 0.3 and 2,650 kg/m$^3$, respectively. The column was divided into 20 grid cells for TOUGHREACT simulations. The dispersivity was assumed $\alpha = \Delta x/2 = 0.002$ m where $\Delta x$ is the grid cell size. The cation exchange capacity (CEC) is 0.01779 meq/100 g solid. There are two cation exchange reactions occurring: $\text{Na}^+ + \text{K-X} = \text{K}^+ + \text{Na-X}$ and $\text{Na}^+ + 0.5\text{Ca}^2+\text{X}_2 = 0.5\text{Ca}^2+ + \text{Na-X}$, where X is the exchange site. Selectivity coefficients of $\text{K}^+$ and $\text{Ca}^2+$ with respect to $\text{Na}^+$ are $K_{\text{Na/K}} = 0.1995$ and $K_{\text{Na/Ca}} = 0.3981$. The initial and boundary concentrations of each chemical component are listed in Table 2.

Based on the above information, the concentrations of Ca$^{2+}$, Na$^+$, K$^+$, and Cl$^-$ at the discharge end for 25 time steps were generated by the numerical model and treated as noise-free observations. In order to evaluate the performance of the proposed TNNA-AUS inversion method, we tested four observation scenarios corrupted by Gaussian noises of different levels as $\epsilon \sim N(1, 0^2)$, $\epsilon \sim N(1, 0.01^2)$, $\epsilon \sim N(1, 0.05^2)$, and $\epsilon \sim N(1, 0.1^2)$.
Seven parameters need to be estimated in this case model. They include selectivity coefficients of $K_{Na/K}$ and $K_{Na/Ca}$, initial concentrations of Na$^+$ ($C_{0Na^+}$) and K$^+$ ($C_{0K^+}$), boundary concentrations of Ca$^{2+}$ ($C_{BCa^{2+}}$) and Cl$^-$ ($C_{BCl^-}$), and CEC. The prior model parameters intervals are listed in Table 3. It is noted that parameter sensitivity plays a vital role in evaluating inversion methods. In the worst (unlikely) case, if all parameters are not sensitive to the objective functions of data assimilation methods (Equation 9), no inverse modeling method will perform satisfactorily (Siade et al., 2017; H. Wang & Li, 2018; J. Zhang et al., 2015, 2020). Sensitivity analysis is conducted here to help us understand the identifiability of the RTM parameters. The sensitivity analysis for these parameters is based on the observation scenario of $\epsilon \sim N(1, 0^2)$ and results are shown in Figure 4.

### 3.1.2. DNN Design

The DNN with the best performance among FC-DNN, 1D-CNN, and 2D-CNN was selected for constructing $F_{Forward}$ and $F_{Reverse}$. A training data set with 1,000 samples and a testing data set with 100 samples were prepared. The prediction accuracy of the three proposed DNN architectures was evaluated by the root mean square error (RMSE) and correlation coefficient ($R^2$), which are defined as

$$RMSE = \sqrt{\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \parallel \tilde{y}_{i(test)}^{Norm} - \tilde{y}_{i}^{DNN} \parallel^2}$$

and

$$R^2 = 1 - \frac{\sum_{i=1}^{N_{test}} \parallel \tilde{y}_{i(test)}^{Norm} - \bar{y}_{test} \parallel^2}{\sum_{i=1}^{N_{test}} \parallel \tilde{y}_{i(test)}^{Norm} - \bar{y}_{test} \parallel^2}$$

respectively. Here, $\tilde{y}_{i}$ and $\tilde{y}_{i(test)}^{Norm}$ are respectively the DNN prediction of $i$th normalized model parameter vector and its corresponding normalized model response vector in test data set. $\bar{y} = 1/N_{test} \sum_{i=1}^{N_{test}} \tilde{y}_{i(test)}^{Norm}$ is the mean vector for all $\tilde{y}_{i(test)}^{Norm}$ ($i = 1, ..., N_{test}$). A lower RMSE and a $R^2$ value approaching 1.0 suggest better DNN prediction ability.

As shown in Figure 5, the FC-DNN has the largest RMSE and the smallest $R^2$. It means that its prediction accuracy is the lowest. While the 2D-CNN has the smallest RMSE and the largest $R^2$ confirming its robust performance. The prediction accuracy of 1D-CNN is close to 2D-CNN and their performances are nearly equal. However, trainable parameters in 1D-CNN (i.e., 16,969,844) is about 7 times more than 2D-CNN (i.e., 2,527,796). This means that more computer memory is required if one uses 1D-CNN. Hence, $F_{Forward}$ and $F_{Reverse}$ were constructed by the 2D-CNN in this study.

### 3.1.3. Verification Results

Three initial forward networks are constructed during verification using three training data sets with $N_{sample}$ = 200, 500, and 1,000. They are called as $F_{Forward(0)-200}$, $F_{Forward(0)-500}$, and $F_{Forward(0)-1000}$, respectively. The RMSE and $R^2$ values of these three forward networks confirm that the prediction accuracy of DNN increases with training samples (Figure 6). Hence, $F_{Forward(0)-200}$ and $F_{Forward(0)-500}$ can be treated as initial forward networks with different global approximate accuracies. Then, the inversion programs of TNNA and TNNA-AUS based on $F_{Forward(0)-200}$ (i.e., TNNA-200 and TNNA-AUS-200), $F_{Forward(0)-500}$ (i.e., TNNA-500 and TNNA-AUS-500), and $F_{Forward(0)-1000}$ (i.e., TNNA-1000 and TNNA-AUS-1000) are performed using observations of four designed scenarios. The $N_{sample(loc)}$ and $I_{max}$ were set as 10 and 20, respectively.

### Table 2

| Chemical components | Initial concentrations | Boundary concentrations |
|---------------------|------------------------|-------------------------|
| $H^+$               | $1 \times 10^{-7}$     | $1 \times 10^{-7}$      |
| $Ca^{2+}$           | $1 \times 10^{-10}$    | $6 \times 10^{-4}$      |
| $Na^+$              | $1 \times 10^{-3}$     | $1 \times 10^{-10}$     |
| $K^+$               | $2 \times 10^{-4}$     | $1 \times 10^{-10}$     |
| $Cl^-$              | $1 \times 10^{-10}$    | $1.2 \times 10^{-3}$    |
| $NO_3^-$            | $1.2 \times 10^{-3}$   | $1 \times 10^{-10}$     |

### Table 3

| Parameter | Prior interval                  |
|-----------|---------------------------------|
| $K_{Na/K}$ | [0.1, 0.3]                      |
| $K_{Na/Ca}$| [0.25, 0.42]                    |
| $C_{Na^+]$ | [8 $\times$ 10$^{-1}$, 8 $\times$ 10$^{-3}$] |
| $C_{Ca^{2+}}$ | [1 $\times$ 10$^{-1}$, 1 $\times$ 10$^{-3}$] |
| $C_{Na/Ca}$ | [3 $\times$ 10$^{-1}$, 8 $\times$ 10$^{-4}$] |
| $C_{BCa^{2+}}$ | [5 $\times$ 10$^{-1}$, 5 $\times$ 10$^{-3}$] |
| CEC        | [5 $\times$ 10$^{-3}$, 5 $\times$ 10$^{-2}$] |
The inverse modeling results were evaluated according to the curve fitting and parameter estimation errors. Figures 7–10 display the inverse modeling results of four designed observation scenarios, respectively. In each of Figures 7–10, (a)–(f) are the curve fitting of computed (lines) and measured (points) concentrations. The terminal values of Equation 9 ($MSE_{\text{fit}}$) are labeled in (a)–(f) as quantitative values for measuring curve fitting results; (g)–(m) are the comparisons of estimated and true values of model parameters, and (n)–(p) are the relative errors of estimated model parameter values.

According to the results of observation scenario $\epsilon \sim N(1, 0^2)$ in Figure 7, there are obvious bias between computed lines of TNNA-200 and observations, with a $MSE_{\text{fit}}$ value of $7.12 \times 10^{-3}$. Among the candidate parameters, only the estimated values of $K_{\text{Na/Ca}}$ and $C_{\text{CEC}}$ are close to their true values, whose relative errors are less than 5%. The estimated values of some other parameters (i.e., $K_{\text{Na/K}}, K_{\text{Na}}, C_{\text{CEC}}$) are far from their true values. Their relative errors can reach up to 20% or above. By using TNNA-500, the computed lines match the observations better than that of TNNA-200 and the value of $MSE_{\text{fit}}$ decreases to $4.1 \times 10^{-4}$. More parameters with relative errors are less than 5% (i.e., $C_{\text{Na}^+}, C_{\text{BCa}^{2+}}, C_{\text{Cl}^{-}}$). Besides the relative errors of parameter estimated values are all less than 10% except $K_{\text{Na/K}}$ (close to 15%). When the TNNA-1000 is chosen as the inverse modeling strategy, the value of $MSE_{\text{fit}}$ will further decrease to $5.14 \times 10^{-5}$. The estimated values of most parameters are close to their true values.

Figure 4. Sensitivity analysis results for seven parameters of observation scenario: $\epsilon \sim N(1, 0^2)$. In each panel, the horizontal axis is parameter variation and the vertical axis represents loss function value expressed in the form of $a \times 10^n$, where $a$ is in the range of 0–10. Different colors are set for parameter sensitivity lines according to the value of $n$ (i.e., the order of magnitude).
Only the relative errors of $K_{NaK}$ and $K_{NaCa}$ are larger than 5%. These results illustrate that a low accurate $F_{Forward(0)}$ will result in bias TNNA inversion values. By adding training samples globally for $F_{Forward(0)}$ constructing, the inversion performance of TNNA can be improved.

In addition to enhancing the global accuracy of $F_{Forward(0)}$ by adding training samples in the whole prior region, the AUS can also reduce inversion bias by adding new samples in the local region iteratively. According to Figure 7, the computed lines of TNNA-AUS-200 match better with observations than that of TNNA-200. With an updating

---

**Figure 5.** Root mean square error (RMSE) and $R^2$ values of three deep neural network (DNN) architectures (i.e., 2D-CNN, 1D-CNN, and FC-DNN) evaluated on the test data set of 100 samples. Networks were trained by using a training data set of 1,000 samples.

---

**Figure 6.** RMSE and $R^2$ values of three forward networks (i.e., $F_{Forward(0)-200}$, $F_{Forward(0)-500}$, and $F_{Forward(0)-1000}$) evaluated on the test data set of 100 samples.
Figure 7. Inverse modeling results of TNNA and TNNA-adaptive updating strategy (AUS) strategies on the observation scenario of $\varepsilon \sim N(1, 0^2)$, including the matching results between computed lines and observations (a–f), comparison of estimated parameter values with their true values (g–m), and relative errors of each parameter (n–p). F-200, F-500, and F-1000 represent $F_{Forward(0)-200}$, $F_{Forward(0)-500}$, and $F_{Forward(0)-1000}$, respectively.
Figure 8. Inverse modeling results of TNNA and TNNA-AUS strategies on the observation scenario of $\varepsilon \sim N(1, 0.01^2)$, including the matching results between computed lines and observations (a–f), comparison of estimated parameter values with their true values (g–m), and relative errors of each parameter (n–p). F-200, F-500, and F-1000 represent $F_{\text{Forward}(0)-200}$, $F_{\text{Forward}(0)-500}$, and $F_{\text{Forward}(0)-1000}$, respectively.
Figure 9. Inverse modeling results of TNNA and TNNA-AUS strategies on the observation scenario of $\varepsilon \sim N(1, 0.05^2)$, including the matching results between computed lines and observations (a–f), comparison of estimated parameter values with their true values (g–m), and relative errors of each parameter (n–p). F-200, F-500, and F-1000 represent $F_{\text{Forward(0)-200}}$, $F_{\text{Forward(0)-500}}$, and $F_{\text{Forward(0)-1000}}$ respectively.
Figure 10. Inverse modeling results of TNNA and TNNA-AUS strategies on the observation scenario of $\epsilon \sim N(1, 0.1^2)$, including the matching results between computed lines and observations (a–f), comparison of estimated parameter values with their true values (g–m), and relative errors of each parameter (n–p). F-200, F-500, and F-1000 represent $F_{\text{Forward}(0)-200}$, $F_{\text{Forward}(0)-500}$, and $F_{\text{Forward}(0)-1000}$ respectively.
of 12 iterations for forward network in TNNA-AUS-200, the MSE_{fit} decrease from $7.12 \times 10^{-3}$ (TNNA-200) to $2.03 \times 10^{-5}$ (TNNA-AUS-200). It is even smaller than TNNA-1000, which is $5.14 \times 10^{-5}$. The estimated values of candidate inverse modeling parameters are close to their true values, with relative errors less than 5%. This result is obviously better than that of TNNA-1000. In terms of their algorithm efficiency, the TNNA-AUS-200 reaches the termination condition after 12 iterations. Hence, the total number of calling high-fidelity numerical simulations in TNNA-AUS-200 is $200 + 12 \times 10 = 320$, which is less than a third of 1,000 in TNNA-1000. It also indirectly shows that the dimensionality problem is attenuated by TNNA-AUS compared to TNNA. Similarly, the performances of TNNA-AUS-500 and TNNA-AUS-1000 are also improved comparing to TNNA-500 and TNNA-1000.

Significantly, the estimated value of $K_{\text{Na/Ca}}$ is still far from its true value with a relative error of about 15%, according to the result of TNNA-AUS-1000. One possible reason is that the sensitivity of $K_{\text{Na/Ca}}$ to the loss function is at a relatively low level. According to Figure 4, when the variation of $K_{\text{Na/Ca}}$ is 15%, the corresponding variation of the loss function is smaller than $1 \times 10^{-6}$, whereas the value of MSE_{fit} for TNNA-AUS-1000 is $9.67 \times 10^{-6}$, which is more significant than $1 \times 10^{-6}$. Hence, $K_{\text{Na/Ca}}$ is determined as a low identifiability RTM parameter. On the other hand, the approximation error of forward network still exists, although its local accuracy is enhanced. Apart from $K_{\text{Na/Ca}}$, some other parameters estimated by TNNA-AUS may also be slightly larger (less than 1%) than that of TNNA. For example, the estimation of $K_{\text{Na/Ca}}$ by TNNA-AUS-200, $C_{\text{BCl}}$ by TNNA-AUS-500, and $C_{\text{BCl}}$ by TNNA-AUS-1000. However, their relative errors are less than 5%, which is very small. The influence of these abnormal increases in relative error after adopting AUS can be neglected.

For the results of observation scenarios $\varepsilon \sim N(1, 0.01^2)$, $\varepsilon \sim N(1, 0.05^2)$, and $\varepsilon \sim N(1, 0.1^2)$ in Figures 8–10, the performance of each of the designed inverse modeling strategies is not significantly affected when the observations are corrupted by noises. And the variation trend of inverse modeling accuracy in Figures 8–10 is consistent with Figure 7. The inversion results are also less affected by the approximate error of $F_{\text{Forward}(0)}$ in TNNA-AUS. Note that there are very marginal improvements between TNNA-1000 and TNNA-AUS-1000. Especially, for the observation scenarios of $\varepsilon \sim N(1, 0.05^2)$, the MSE_{fit} of TNNA-AUS-1000 ($1.61 \times 10^{-4}$) is slightly larger than that of TNNA-1000 ($1.53 \times 10^{-4}$). One probable reason is that the $F_{\text{Forward}(0)-1000}$ can be considered a global accurate surrogate model with higher forward approximate accuracy around the real parameter point. Hence, none of the total iteration number of these scenarios is beyond two (see (f) of Figures 7–10). On the other hand, as mentioned in Section 2.2, the $\varepsilon_{\text{Forward},2}$ may not follow an absolute decline trend at each iteration and a value of $\varepsilon_{\text{Forward},2} = 1.61 \times 10^{-4}$ satisfies the terminal criteria of the inversion algorithm.

Overall, both enhancing $F_{\text{Forward}(0)}$ accuracy using more training samples and modifying TNNA into TNNA-AUS are beneficial to reduce the inversion biases. Between them, the TNNA-AUS has more advantages. Especially, when the $F_{\text{Forward}(0)}$ is trained using a small size of data set, the inversion result of TNNA-AUS can be significantly improved than TNNA with a small extra computational cost.

### 3.2. Application Example

#### 3.2.1. Aquia Aquifer

This section illustrates the application of TNNA-AUS in the Aquia aquifer. The Aquia aquifer is located in the eastern coastal plain of Maryland, United States (Figure 11). This formation is a Paleocene layer of unconsolidated marine deposit, which consists of quartz sands (50%–75%), glauconite (20%–40%), and carbonate shell debris (about 1%–5%, but in some place, it can reach up to 20%). The deposit environment of the Aquia aquifer is considered a regressive marine environment. Figure 11a shows an outline of the Aquia aquifer, which suggests a confined aquifer in the upstream part and a gradual loss of water in the downstream part of the aquifer. According to available site data, an approximate flow path exists crossing from northwest to southeast. The schematic cross section along the flow direction is also displayed in Figure 11b, which shows the spatial coverage of strata in the aquifer system (Appelo, 1994; Chapelle & Knobel, 1983; Dai et al., 2006).

The Upper Cretaceous Seven formations are considered as a confined bed of this subsurface system. It is overlaid by the Aquia formation. The overlying Aquia formation is the Marlboro Clay and the lower Nanjemoy formation. Marlboro Clay is a kind of plastic clay, which mainly consists of kaolinite and mixed clay. The Nanjemoy formation is composed of glauconitic silty clay. In addition, the Nanjemoy formation is overlaid by Piney Point...
formation conformably, which mainly consists of medium and coarse glauconitic quartz sand. It is considered that there might be an upward leakage from the Aquia aquifer to the Piney Point formation via Pleistocene channels.

3.2.2. Problem Description and Model Setup

The geochemical data collected from observation wells show that the groundwater is of calcium and magnesium bicarbonate type near the outcrop of the Aquia aquifer with a pH of 6.5–7.5. Along the flow direction, the water type gradually evolves into sodium bicarbonate type, and the pH value increases to 7.5–8.5 (Aeschbach-Hertig et al., 2002; Chapelle & Knobel, 1983; Dai & Samper, 2006). As a result, there are significant geochemical reactions occurring along the Aquia aquifer from the upstream to downstream. Related studies have described these dominant geochemical processes (Dai & Samper, 2006; Dai et al., 2006; T. Xu et al., 2008).

Here, TOUGHREACT was used to build a flow and RTMs for the Aquia aquifer. The model has a length of 96.56 km. The recharge zone is located at 0 km, and the leakage is at 61.16 km from the recharge area. Although processes identification is important in inverse modeling tasks, our concern here is to test the parameter estimation
### Table 4
Geochemical Reactions Considered for the Aquifer Aquifer

| Geochemical reactions                                                                 | log(K) at 25°C |
|--------------------------------------------------------------------------------------|----------------|
| Aqueous complexation                                                                  |                |
| \( OH^- = H_2O - H^+ \)                                                               | 13.995         |
| \( CO_3^{2-} = HCO_3^- + H^+ \)                                                       | 10.329         |
| \( CO_3^{2-} (aq) = HCO_3^- + H^+ - H_2O \)                                           | -6.345         |
| \( CaHCO_3 = Ca^{2+} + HCO_3^- \)                                                    | -1.043         |
| \( MgHCO_3 = Mg^{2+} + HCO_3^- \)                                                    | -1.033         |
| NaHCO_3 (aq) = Na^+ + HCO_3^-                                                        | -0.17          |
| NaSO_4 = Na^+ + SO_4^{2-}                                                            | -0.811         |
| KSO_4 = K^+ + SO_4^{2-}                                                              | -0.875         |
| NaCO_3 = Na^+ + HCO_3^- - H^+                                                        | 9.816          |
| Methane oxidation                                                                    |                |
| \( CH_4 (aq) + O_2 (aq) = H_2O + H^+ + HCO_3^- \)                                    | 144.141        |
| Proton surface complexification                                                       |                |
| \( XOH_2 = XOH + H^+ \)                                                               | -7.4           |
| \( XO = XOH - H^+ \)                                                                 | 9.24           |
| Calcite dissolution/precipitation                                                     |                |
| Calcite = Ca^{2+} + HCO_3^- - H^+                                                    | 1.854          |
| Cation exchange                                                                       |                |
| \( Na^+ + 0.5Ca - X_2 = 0.5Ca^{2+} + Na - X \)                                        |                |
| \( Na^+ + 0.5Mg - X_2 = 0.5Mg^{2+} + Na - X \)                                        |                |
| \( Na^+ + 0.5K - X = K^+ + Na - X \)                                                 |                |

Note. K represents the equilibrium constant of the chemical reactions.

### Table 5
Prior Intervals for the Permeability (m²), Cation Exchange Coefficients, Initial and Boundary pHs, Initial and Boundary Concentrations of Some Dissolved Chemical Components (mol/L), and CEC (meq/100 g)

| Model parameter | Prior interval         | Model parameter | Prior interval         |
|-----------------|------------------------|-----------------|------------------------|
| Permeability    | \([1 \times 10^{-12}, 9.9 \times 10^{-13}]\) | \(C_{0K}^+\)   | \([1 \times 10^{-3}, 4 \times 10^{-4}]\) |
| \(K_{NaK}\)    | \([7 \times 10^{-3}, 5 \times 10^{-4}]\)   | \(C_{0HCO_3^-}\) | \([8 \times 10^{-3}, 4 \times 10^{-4}]\) |
| \(K_{NaCa}\)   | \([1 \times 10^{-1}, 9 \times 10^{-4}]\)  | \(\text{pH}_B\) | \([6.5, 9]\)           |
| \(K_{NaMg}\)   | \([1 \times 10^{-1}, 9 \times 10^{-4}]\)  | \(C_{0Ca^{2+}}\) | \([8 \times 10^{-3}, 4 \times 10^{-4}]\) |
| \(K_{NaH}\)    | \([1 \times 10^{-6}, 8 \times 10^{-6}]\)  | \(C_{0Mg^{2+}}\) | \([8 \times 10^{-7}, 4 \times 10^{-4}]\) |
| \(\text{pH}_B\) | \([6.5, 9]\)           | \(C_{0Na^+}\)   | \([7 \times 10^{-3}, 4 \times 10^{-4}]\) |
| \(C_{0Ca^{2+}}\) | \([1 \times 10^{-3}, 9 \times 10^{-3}]\)  | \(C_{0Ba^+}\)   | \([2 \times 10^{-5}, 9 \times 10^{-5}]\) |
| \(C_{0Mg^{2+}}\) | \([7 \times 10^{-3}, 4 \times 10^{-2}]\)  | \(C_{0HCO_3^-}\) | \([1 \times 10^{-3}, 7 \times 10^{-4}]\) |
| \(C_{0Na^+}\)  | \([5 \times 10^{-2}, 1 \times 10^{-1}]\)  | CEC             | \([1, 6]\)            |

It has been verified that the initial sample size has little effect on the inversion result for TNNA-AUS method. However, when the initial sample size is too small, it may be difficult to give appropriate inversion results within the specified number of iterations. Hence, the parameters of this case are estimated using the TNNA-AUS-500 with 20 iterations. The model response for sorption is 1 × 10^3 m²/min, and the sorption site density is 5 × 10^3 mol/m²/min. The Gouy–Chapman model (Chapman, 1913; Gouy, 1910) of the double layer was used as the adsorption. The total simulation time was set as 100,000 years.

### 3.2.3. Estimation of the Aquifer Aquifer Parameters

There are 18 parameters to be estimated in this case, including permeability, cation exchange coefficients \((K_{NaK}, K_{NaCa}, K_{NaMg}, \text{ and } K_{NaH})\), initial value of pH \((\text{pH}_B)\), boundary value of pH \((\text{pH}_B)\), initial concentrations of some chemical components \((C_{0Ca^{2+}}, C_{0Mg^{2+}}, C_{0Na^+}, C_{0K^+}, \text{ and } C_{0HCO_3^-})\), boundary concentrations of some chemical components \((C_{BCa^{2+}}, C_{BMg^{2+}}, C_{BNa^+}, C_{BK^+})\), and \(C_{BHCO_3^-}\), and CEC. The prior parameter intervals were determined by using information available in previous studies (Dai & Samper, 2006; see Table 5). Available observations on \(Na^+, K^+, Ca^{2+}, Mg^{2+}, HCO_3^-\), and pH collected from observation wells were used here as model responses.
the corresponding observations (e.g., Ca\textsuperscript{2+}). The possible reason is that an inevitable gap between the numerical model and the real-world system still exists. For example, the Aquia aquifer model was conceptualized with homogeneous parameters here.

4. Conclusions and Discussion

This paper developed a deep-learning-based inverse modeling algorithm, TNNA-AUS, for estimating RTM parameters. By adding AUS to TNNA, we were able to enhance the accuracy of the forward network. The DNN architecture for constructing the forward network and reverse network is a 2D-CNN based on the prediction accuracy compared with 1D-CNN and FC-DNN. The proposed inverse modeling method of TNNA-AUS was verified by a synthetic example of cation exchange reactions. It was shown that if the sample size for training $F_{\text{Forward}(0)}$ is relatively small, the inverse modeling results of TNNA are biased. With an increasing number of training samples in the prior region, the bias of inverse results is gradually reduced. However, this strategy is computationally inefficient. In contrast, the TNNA-AUS can improve the accuracy and efficiency of inverse modeling and shows excellent performance in all designed scenarios. After verification, the TNNA-AUS was applied in RTM for the well-documented Aquia aquifer. The final inversion results indicate that the developed TNNA-AUS framework could be applied for parameter estimation related to complex reactive transport problems.

TNNA-AUS is similar to most existing inverse modeling methods in convergence conditions, which is the maximum likelihood estimation developed from Bayes’ theorem. Hence, the model calibration error is a major judgment for assessing inverse modeling methods. The model calibration errors of the synthetic example are very small according to the curve fitting results. The

| Table 6 Parameter Estimated Values of the Aquia Aquifer Model, Including the Permeability (m\textsuperscript{2}), Cation Exchange Coefficients, Initial and Boundary pHs, Initial and Boundary Concentrations of Some Dissolved Chemical Components (mol/L), and CEC (meq/100 g) |
|---------------------------------|------------------|---------------------------------|------------------|
| Model parameter                | Estimated value  | Model parameter                | Estimated value  |
| Permeability                    | $3.086 \times 10^{-12}$ | $C_{0K^+}$                     | $2.163 \times 10^{-3}$ |
| $K_{NaK}$                      | $2.173 \times 10^{-1}$ | $C_{OHCO}$                     | $2.159 \times 10^{-2}$ |
| $K_{NaCa}$                     | $5.927 \times 10^{-1}$ | $pH_B$                         | 6.923             |
| $K_{NaMg}$                     | $5.842 \times 10^{-1}$ | $C_{CaCO}$                     | $1.284 \times 10^{-3}$ |
| $K_{NaH}$                      | $3.407 \times 10^{-6}$ | $C_{BHCO}$                     | $3.193 \times 10^{-6}$ |
| $pH_0$                         | 6.716             | $C_{H^+}$                      | $9.638 \times 10^{-5}$ |
| $C_{CaCO^2+}$                  | $6.118 \times 10^{-3}$ | $C_{Ca^2+}$                    | $7.995 \times 10^{-5}$ |
| $C_{0Mg^2+}$                   | $1.284 \times 10^{-2}$ | $C_{Mg^2+}$                    | $2.746 \times 10^{-3}$ |
| $C_{0Na^+}$                    | $8.85 \times 10^{-2}$ | CEC                            | 2.7867            |

Figure 12. Measured (symbols; Dai & Samper, 2006) and computed concentrations (lines) of Na\textsuperscript{+}, K\textsuperscript{+}, Ca\textsuperscript{2+}, Mg\textsuperscript{2+}, HCO\textsubscript{3}\textsuperscript{−}, and pH of the Aquia aquifer. Note the trend of computed lines is consistent with the observations.
numerical model of Aquia aquifer is also calibrated to have the same accuracy level as our previous study (Dai & Samper, 2006). In conclusion, TNNA-AUS can estimate parameter values with high accurate convergence and outstanding performance when the unknown parameter number is relatively small. However, the global accuracy of $F_{\text{Forward}}(0)$ trained by limited training samples decreases with the input dimension. When the parameter dimension is large, enormous AUS iterations are required, or more pair-wise training data are used to construct the $F_{\text{Forward}}(0)$ with high global accuracy. Hence, in this case, the TNNA-AUS also suffers from the problem of dimensionality. To extend TNNA-AUS for high-dimensional parameter inversions, necessary modifications are required. Specifically, construct $F_{\text{Forward}}$ and $F_{\text{Reverse}}$ using deeper DNNs with specifically designed architectures, such as ResNet (K. He et al., 2016) or DenseNet (Huang et al., 2017) to capture complex regressions among high-dimensional data (Kuang et al., 2021; Mo, Zhu, et al., 2019). It is also practical to train $F_{\text{Forward}}(0)$ for RTM surrogate modeling using theory-guided approaches, whose loss function is derived from physical equation constraints, and pair-wise training data sets are not required. Hence, it is possible to construct a $F_{\text{Forward}}(0)$ with higher accuracy than the data-driven strategy with less computational burden. When the high-dimensional parameters are space dependent, we can construct an operator mapping from low-dimensional latent space to high-dimensional space-dependent parameters employing variational autoencoder (Kingma & Welling, 2013), and their variants (Cheng et al., 2020; Mo et al., 2020). These methods can be used for dimensionality reduction, and related applications can refer from (Bao et al., 2020; Chan & Elsheikh, 2019; Laloy et al., 2018; Lopez-Alvis et al., 2021; Mo et al., 2020). For example, the parameter with 129 × 129 dimension is represented by 5 × 5 dimension latent space in the study of Bao et al. (2020). After that, the high-dimensional space-dependent parameters can be estimated indirectly by estimating the low-dimensional latent space through TNNA-AUS algorithm. For CPU-intensive RTM problems, other AUS methods such as secondary surrogate and multifidelity surrogate models may also be employed in future studies.

For ill-posed problems, more comprehensive information is required for regularization constraints to attenuate the ill-posedness (Kravaris & Seinfeld, 1985; N. Sun, 2013). On the other hand, the TNNA-AUS inverse framework developed in this paper was based on deterministic deep learning and had difficulty dealing with parameters with multimodal distributions. Hence, it is also practical to use probabilistic deep learning methods such as Bayesian neural networks to construct $F_{\text{Reverse}}$ for multimodal parameters. In that case, TNNA-AUS can search the whole posterior distribution and quantify parameter uncertainty (Brusaferri et al., 2019; Y. Liu, Qin, et al., 2019). For the Aquia aquifer, our subsequent studies may incorporate heterogeneous structures into the aquifer model to improve our understanding of the subsurface dynamic processes and the relevant parameters.

## Appendix A

**Algorithm** for TNNA-AUS

1. Draw $N_{\text{sample}}(0)$ random samples from the prior region of parameter, that is, $M_{0,i}^p$, calculate $Y_{0,i}$. Obtain the initial training data set $D_{0} = \{M_{0,i}^p, Y_{0,i}\}$.
2. Train the $F_{\text{Forward}}(0)$ according to $D_{0}$.
3. Train the $F_{\text{Reverse}}(0)$ according to $F_{\text{Forward}}(0)$ and $\tilde{y}_{\text{obs}}$. Calculate the initial estimate of the parameter vector $\tilde{m}_0$ through $F_{\text{Reverse}}(0)$\(\tilde{y}_{\text{obs}} \cdot \theta_{\text{Reverse}}\).
4. for $i = 1, \ldots, f_{\text{max}}$:
   1. Draw $N_{\text{sample}(i)}(0)$ new samples around $\tilde{m}_{i-1}$, that is, $M_{i-1,i}^p$, calculate $Y_{i-1,i}$. Obtain the local data set $D_{i-1} = \{M_{i-1,i}, Y_{i-1,i}\}$, after data normalization. Obtain $D_{i}$ by adding $D_{i-1}$ into $D_{i-1}$ and abandoning the $N_{\text{sample}(i)}$ samples with the largest Euclidean distance from $m_{i-1}$.
   2. Obtain the $F_{\text{Forward}}(i)$ by updating the parameters in the last layer of $F_{\text{Forward}}(i-1)$.
   3. Train the $F_{\text{Reverse}}(i)$ according to $F_{\text{Forward}}(i)$ and normalized observation vector $\tilde{y}_{\text{obs}}$, calculated $\tilde{m}_i$ by feeding $\tilde{y}_{\text{obs}}$ into $F_{\text{Reverse}}(i)$.
   4. Calculate the approximation error $e_{\text{Forward}}$ between surrogate model and high-fidelity numerical model outputs at $\tilde{m}_i$.
      - if $e_{\text{Forward}}(i) \leq e_{\text{stop}}$: output $\tilde{m}_i$ and **break out** the iteration.
      - else: **continue** iteration.
Data Availability Statement

The data sets for DNN design in Section 3.1.2 and the observations of synthetic example are both available on the following site: http://doi.org/10.5281/zenodo.4972060. The observations of Aquia aquifer are available from Dai and Samper (2006).

References

Adler, J., & Öktem, O. (2017). Solving ill-posed inverse problems using iterative deep neural networks. Inverse Problems, 33(12). https://doi.org/10.1088/1361-6420/aa9f58

Aeschbach-Hertig, W., Stute, M., Clark, J. F., Reuter, R. F., & Schlosser, P. (2002). A paleotemperature record derived from dissolved noble gases in groundwater of the Aquia aquifer (Maryland, USA). Geochimica et Cosmochimica Acta, 66(5), 797–817. https://doi.org/10.1016/S0016-7037(01)00804-3

Ampomah, W., Balch, R. S., Grigg, R. B., McPherson, B., Will, R. A., Lee, S.-Y., et al. (2017). Co-optimization of CO2-ESR and storage processes in mature oil reservoirs. Greenhouse Gases: Science and Technology, 7(1), 128–142. https://doi.org/10.1002/gts.1618

Appelo, C. A. J. (1994). Cation and proton exchange, pH variations, and carbonate reactions in a freshening aquifer. Water Resources Research, 30(10), 2793–2805. https://doi.org/10.1029/94WR01048

Arsenault, R., & Brissette, F. P. (2014). Continuous streamflow prediction in ungauged basins: The effects of equilibrium and parameter selection on uncertainty in regularization approaches. Water Resources Research, 50, 6135–6153. https://doi.org/10.1002/2013WR014898

Asher, M. J., Croke, B. F. W., Jakeman, A. J., & Peeters, L. J. M. (2015). A review of surrogate models and their application to groundwater modeling. Water Resources Research, 51, 5957–5973. https://doi.org/10.1002/2015WR016967

Ba, Y., Jiang, L., & Ou, N. (2018). A two-stage ensemble Kalman filter based on multiscale model reduction for inverse problems in time fractional diffusion-wave equations. Journal of Computational Physics, 374, 300–330. https://doi.org/10.1016/j.jcp.2018.06.077

Bacigalupo, A., Gnecco, G., Lepidi, M., & Gambarotta, L. (2021). Computational design of innovative mechanical metafilters via adaptive surrogate-based optimization. Computer Methods in Applied Mechanics and Engineering, 375, 113623. https://doi.org/10.1016/j.cma.2020.113623

Bahrami, S., Ardejani, F. D., & Baafi, E. (2016). Application of artificial neural network coupled with genetic algorithm and simulated annealing to solve groundwater inflow problem to an advancing open pit mine. Journal of Hydrology, 536, 471–484. https://doi.org/10.1016/j.jhydrol.2016.03.002

Bandai, T., & Guozejie, T. A. (2021). Physics-informed neural networks with monotonicity constraints for Richardson–Richards equation: Estimation of constitutive relationships and soil water flux density from volumetric water content measurements. Water Resources Research, 57, e2020WR027642. https://doi.org/10.1029/2020WR027642

Bao, J., Li, L., & Redoloza, F. (2020). Coupling ensemble smoother and deep learning with generative adversarial networks to deal with non-Gaussianity in flow and transport data assimilation. Journal of Hydrology, 590, 125443. https://doi.org/10.1016/j.jhydrol.2020.125443

Berkhahn, S., Puchs, L., & Neuweiler, I. (2019). An ensemble neural network model for real-time prediction of urban floods. Journal of Hydrology, 575, 743–754. https://doi.org/10.1016/j.jhydrol.2019.05.066

Bjarkason, E. K., Maclaren, O. J., & Neuweiler, I. (2019). Randomized truncated SVD Levenberg–Marquardt approach to geothermal natural state and history matching. Water Resources Research, 55, 2376–2404. https://doi.org/10.1029/2017WR021870

Brusafieri, A., Matteucci, M., Portolani, P., & Vitali, A. (2019). Bayesian deep learning based method for probabilistic forecast of day-ahead electricity prices. Applied Energy, 250, 1158–1175. https://doi.org/10.1016/j.apenergy.2019.05.068

Celia, M. A. (2017). Geological storage of captured carbon dioxide as a large-scale carbon mitigation option. Water Resources Research, 53, 3527–3533. https://doi.org/10.1002/2017WR020841

Chen, Y., Yu, H., Ma, H., Li, X., Hu, R., & Yang, Z. (2020). Inverse modeling of saturated–unsaturated flow in site-scale fractured rocks using the continuum approach: A case study at Baihetan dam site, Southwest China. Journal of Hydrology, 584, 124693. https://doi.org/10.1016/j.jhydrol.2020.124693

Chen, Y., & Zhang, D. (2006). Data assimilation for transient flow in geologic formations via ensemble Kalman filter. Advances in Water Resources, 29(8), 1107–1122. https://doi.org/10.1016/j.advwatres.2005.09.007

Cheng, M., Fang, F., Pain, C. C., & Navon, I. M. (2020). An advanced hybrid deep adversarial autoencoder for parameterized nonlinear fluid flow models. Computer Methods in Applied Mechanics and Engineering, 372, 113375. https://doi.org/10.1016/j.cma.2020.113375

Dagasan, Y., Jha, A., & Renard, P. (2020). Using generative adversarial networks as a fast forward operator for hydrogeological inverse problems. Groundwater, 58(6), 938–950. https://doi.org/10.1111/gwat.13005

Dai, Z., Middleton, R., Viswanathan, H., Fessenden-Rahn, J., Bauman, J., Pawar, R., et al. (2014). An integrated framework for optimizing CO2 sequestration and enhanced oil recovery. Environmental Science and Technology Letters, 1(1), 49–54. https://doi.org/10.1021/ez4001033

Dai, Z., & Samper, J. (2004). Inverse problem of multicomponent reactive chemical transport in porous media: Formulation and applications. Water Resources Research, 40, W07407. https://doi.org/10.1029/2003WR002348

Dai, Z., & Samper, J. (2006). Inverse modeling of water flow and multicomponent reactive transport in coastal aquifer systems. Journal of Hydrology(327), 447–461. https://doi.org/10.1016/j.jhydrol.2005.11.052

Dai, Z., Samper, J., & Ritzi, R. (2006). Identifying geochemical processes by inverse modeling of multicomponent reactive transport in the Aquia aquifer. Geosphere, 2(4), 210–219. https://doi.org/10.1130/ges00211.1

Dai, Z., Wolsberg, A., Liu, Z., & Deng, H. (2009). Scale dependence of sorption coefficients for contaminant transport in saturated fractured rock. Geophysical Research Letters, 36, L01403. https://doi.org/10.1029/2008GL036516

Dai, Z., Wolsberg, A., Reimus, P., Deng, H., Kwicklis, E., Ding, M., et al. (2012). Identification of sorption processes and parameters for radiouclide transport in fractured rock. Journal of Hydrology, 414–415, 516–526. https://doi.org/10.1016/j.jhydrol.2011.10.035

Dai, Z., Xu, L., Xiao, T., McPherson, B., Zhang, X., Zheng, L., et al. (2020). Reactive chemical transport simulations of geologic carbon sequestration: Methods and applications. Earth-Science Reviews, 208, 103265. https://doi.org/10.1016/j.earscirev.2020.103265
Daneshmand, H., Alaghmeh, S., Camporese, M., Talei, A., & Daly, E. (2019). Water and salt balance modelling of intermittent catchments using a physically-based integrated model. *Journal of Hydrology, 568*, 1017–1030. https://doi.org/10.1016/j.jhydrol.2018.11.035

De Windt, L., & Spycker, N. F. (2019). Reactive transport modeling: A key performance assessment tool for the geologic disposal of nuclear waste. *Elements, 15*(2), 99–102. https://doi.org/10.2138/elements.15.2.99

Dwivedi, D., Steefel, C. I., Arora, B., Newcomer, M., Moutlon, J. D., Daflon, B., et al. (2018). Geochemical exports to river from the intramontane hyporheic zone under transient hydrologic conditions: East River Mountainous Watershed, Colorado. *Water Resources Research, 54*, 8456–8477. https://doi.org/10.1029/2018WR023377

Efendiev, Y., Datta-Gupta, A., Ginting, V., Ma, X., & Mallick, B. (2005). An efficient two-stage Markov chain Monte Carlo method for dynamic water data integration. *Water Resources Research, 41*, W12423. https://doi.org/10.1029/2004WR003764

Elnner, A., Bressan, P. O., Akiyama, T., Gonçalves, W. N., & Marcato Junior, J. (2021). Using deep learning for automatic water stage measurements. *Water Resources Research, 57*, e2020WR027608. https://doi.org/10.1029/2020WR027608

Emerick, A. A., & Reynolds, A. C. (2013). Ensemble smoother with multiple data assimilation. *Computers & Geosciences, 55*(S), 3–15. https://doi.org/10.1016/j.cageo.2012.03.011

Fakhreddine, S., Lee, J., Kitandis, P. K., Fendorf, S., & Rolle, M. (2016). Imaging geochemical heterogeneities using inverse reactive transport modeling: An example relevant for characterizing arsenic mobilization and distribution. *Advances in Water Resources, 88*, 186–197. https://doi.org/10.1016/j.adwres.2015.12.005

Gill, M. K., Kaevel, Y. H., Khalil, A., McKee, M., & Bastidas, L. (2006). Multiobjective particle swarm optimization for parameter estimation in hydrology. *Water Resources Research, 42*, W07417. https://doi.org/10.1029/2005WR004528

Goddériès, Y., Schott, J., & Branley, S. L. (2019). Reactive transport models of weathering. *Elements, 15*(2), 103–106. https://doi.org/10.2138/elements.15.2.103

Gong, W., & Duun, Q. (2017). An adaptive surrogate modeling-based sampling strategy for parameter optimization and distribution estimation (ASMO-PODE). *Environmental Modelling & Software, 95*, 61–75. https://doi.org/10.1016/j.envsoft.2017.05.005

Gong, W., Duun, Q., Li, J., Wang, C., Di, Z., Dai, Y., et al. (2015). Multi-objective parameter optimization of common land model using adaptive surrogate modeling. *Hydrology and Earth System Sciences, 19*(5), 2409–2425. https://doi.org/10.5194/hess-19-2409-2015

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning*. The MIT Press.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., et al. (2018). Recent advances in convolutional neural networks. *Pattern Recognition, 77*, 354–377. https://doi.org/10.1016/j.patcog.2017.10.013

Harrison, A. L., Tutolo, B. M., & DePaolo, D. J. (2019). The role of reactive transport modeling in geologic carbon storage. *Elements, 15*(2), 93–98. https://doi.org/10.2138/elements.15.2.93

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770–778)*.

He, Q., Barajas-Solano, D., Tartakovsky, G., & Tartakovsky, A. M. (2020). Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport. *Advances in Water Resources, 141*, 103610. https://doi.org/10.1016/j.advwater.2020.103610

Heidari, P., Li, L., Jin, L., Williams, J. Z., & Branley, S. L. (2017). A reactive transport model for Marcellus shale weathering. *Geochimica et Cosmochimica Acta, 217*, 421–440. https://doi.org/10.1016/j.gca.2017.08.011

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4700–4708)*.

Irrgang, C., Saynisch-Wagner, J., Dill, R., Boergens, E., & Thomas, M. (2020). Self-validating deep learning for recovering terrestrial water storage from gravity and altimetry measurements. *Geophysical Research Letters, 47*, e2020GL089258. https://doi.org/10.1029/2020GL089258

Iwalewa, T. M., & Farnan, I. (2019). Coupling radioactive waste glass dissolution measurements in generic groundwaters with reactive transport modeling of repository scenarios. *Water Resources Research, 55*, 8010–8027. https://doi.org/10.1029/2019WR024218

Kabir, S., Patidar, S., Xia, X., Liang, Q., Neal, J., & Pender, G. (2020). A deep convolutional neural network model for rapid prediction of fluvial flood inundation. *Journal of Hydrology, 590*, 125481. https://doi.org/10.1016/j.jhydrol.2020.125481

Kingma, D. P., & Welling, M. (2013). *Auto-encoding variational Bayes*. arXiv preprint arXiv:1312.1172. 2013.

Kourakos, G., & Mantoglou, A. (2009). Pumping optimization of coastal aquifers based on evolutionary algorithms and surrogate modular neural network models. *Advances in Water Resources, 32*(4), 507–521. https://doi.org/10.1016/j.advwater.2009.01.001

Kravaris, C., & Seinfeld, J. H. (1985). Identification of parameters in distributed parameter systems by regularization. *SIAM Journal on Control and Optimization, 23*(2), 217–241. https://doi.org/10.1137/0323017

Kuang, W., Yuan, C., & Zhang, J. (2021). Real-time determination of earthquake focal mechanism via deep learning. *Nature Communications, 12*(1), 1432. https://doi.org/10.1038/s41467-021-21670-x

Lal, A., & Datta, B. (2018). Development and implementation of support vector machine regression surrogate models for predicting groundwater pumping-induced saltwater intrusion into coastal aquifers. *Water Resources Management, 32*(7), 2405–2419. https://doi.org/10.1007/s11269-018-1936-2

Laloy, E., Rogiers, B., Vrugt, J. A., Mallants, D., & Jacques, D. (2013). Efficient posterior exploration of a high-dimensional groundwater model from two-stage Markov chain Monte Carlo simulation and polynomial chaos expansion. *Water Resources Research, 49*, 2664–2682. https://doi.org/10.1002/wrcr.20226

Lee-Cun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539

Li, B., Liu, X., Kaufman, M. H., Turetcaia, A., Chen, X., & Cardenas, M. B. (2020). Flexible and modular simultaneous modeling of flow and reactive transport in rivers and hyporheic zones. *Water Resources Research, 56*, e2019WR026528. https://doi.org/10.1029/2019WR026528

Li, M., Mahler, K., Navarre-Stitchler, A., Druhan, J., Meile, C., Lawrence, C., et al. (2017). Expanding the role of reactive transport models in critical zone processes. *Earth-Science Reviews, 165*, 280–301. https://doi.org/10.1016/j.earscirev.2016.09.001

Liao, Q., Zhang, D., & Tchelepi, H. (2017). A two-stage adaptive stochastic collocation method on nested sparse grids for multiphase flow in randomly heterogeneous porous media. *Journal of Computational Physics, 350*, 828–845. https://doi.org/10.1016/j.jcp.2016.10.061

Liu, D., Tan, Y., Khoram, E., & Yu, Z. (2018). Training deep neural networks for the inverse design of nanophotonic structures. *ACS Photonics, 5*(4), 1365–1369. https://doi.org/10.1021/acsphotonics.7b01377
Smith, T. J., & Marshall, L. A. (2008). Bayesian methods in hydrologic modeling: A study of recent advancements in Markov chain Monte Carlo techniques. *Water Resources Research*, 44, W00B05. https://doi.org/10.1029/2007WR006705

Sole-Mari, G., Fernández-Garcia, D., Sanchez-Vila, X., & Bolster, D. (2020). Lagrangian modeling of mixing-limited reactive transport in porous media: Multivariate interaction by exchange with the mean. *Water Resources Research*, 56, e2019WR026993. https://doi.org/10.1029/2019WR026993

Soliman, M. R., Ritz, R. W., Huang, C. C., & Dai, Z. (2015). Relating reactive solute transport to hierarchical and multiscale sedimentary architecture in a Lagrangian-based transport model: 2. Particle displacement variance. *Water Resources Research*, 51, 1601–1618. https://doi.org/10.1002/2014WR016354

Song, X., Chen, X., Ye, M., Dai, Z., Hammond, G., & Zachara, J. M. (2019). Delineicing facies spatial distribution by integrating ensemble data assimilation and indicator geostatistics with level-set transformation. *Water Resources Research*, 33, 2652–2671. https://doi.org/10.1029/2019WR023262

Song, X., Shi, L., Ye, M., Yang, J., & Navon, I. M. (2014). Numerical comparison of iterative ensemble Kalman filters for unsaturated flow inverse modeling. * Vadose Zone Journal*, 13(2), 1–12. https://doi.org/10.2136/vzj2013.05.0083

Steefel, C., Depaolo, D., & Lichtner, P. (2005). Reactive transport modeling: An essential tool and a new research approach for the Earth sciences. *Earth and Planetary Science Letters*, 240(3–4), 539–558. https://doi.org/10.1016/j.epsl.2005.09.017

Steefel, C. I., Appelo, C. A. J., Arora, B., Jacques, D., Kalbacher, T., Kolditz, O., et al. (2014). Reactive transport codes for subsurface environmental simulation. *Computational Geosciences*, 19(3), 445–478. https://doi.org/10.1007/s10596-014-9443-x

Steefel, C. I., & Maher, K. (2009). Fluid-rock interaction: A reactive transport approach. *Reviews in Mineralogy and Geochemistry*, 70(1), 485–532. https://doi.org/10.2138/rmg.2009.70.11

Suárez-Díaz, D., Espinosa-Paredes, G., & Escobar, F. H. (2021). Inverse problems in groundwater modeling. Springer Science & Business Media.

Tartakovsky, A. M., Ortiz Marrero, C., Perdikaris, P., Tartakovsky, G. D., & Barajas-Solano, D. (2020). Physics-informed deep neural networks for inverse problems in groundwater modeling. *Inverse problems in groundwater modeling* (2), 619–644. Springer Science & Business Media.

Sun, A. Y., Zhong, Z., Jeong, H., & Yang, Q. (2019). Building complex event processing capability for intelligent environmental monitoring. *Environmental Modelling & Software*, 116, 1–6. https://doi.org/10.1016/j.envsoft.2019.02.015

Sun, L., Gao, H., Pan, S., & Wang, J.-X. (2019). Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering*, 361, 112732. https://doi.org/10.1016/j.cma.2019.112732

Sun, N. (2013). *Computational Physics*, 413, 109456. https://doi.org/10.1016/j.jcp.2020.10.0456

Tang, M., Liu, Y., & Durlowski, L. J. (2020). A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems. *Journal of Computational Physics*, 413, 109456. https://doi.org/10.1016/j.jcp.2020.10.0456

Vrugt, J. A. (2016). Markov chain Monte Carlo simulation using the DREAM software package: Theory, concepts, and MATLAB implementation. *Reviews in Mineralogy and Geochemistry*, 75, 273–316. https://doi.org/10.2138/rmg.2015.75.02

Xiu, D. B., & Karniadakis, G. E. (2002). The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2), 619–644. https://doi.org/10.1137/S1064827501387826
Xu, R., Zhang, D., Rong, M., & Wang, N. (2020). Weak form theory-guided neural network (TgNN-wf) for deep learning of subsurface single and two-phase flow. arXiv preprint arXiv:2009.04543.

Xu, T., Samper, J., Ayora, C., Manzano, M., & Custodio, E. (1999). Modeling of non-isothermal multi-component reactive transport in field scale porous media flow systems. Journal of Hydrology, 214(1–4), 144–164. https://doi.org/10.1016/S0022-1694(98)00283-2

Xu, T., Sonnenthal, E., Spycher, N., & Pruess, K. (2006). TOUGHREACT—A simulation program for non-isothermal multiphase reactive geochemical transport in variably saturated geologic media: Applications to geothermal injectivity and CO2 geological sequestration. Computers & Geosciences, 32(2), 145–165. https://doi.org/10.1016/j.cageo.2005.06.014

Xu, T., Sonnenthal, E., Spycher, N., & Pruess, K. (2008). TOUGHREACT user’s guide: A simulation program for non-isothermal multiphase reactive geochemical transport in variably saturated geologic media, V1.2 J. Lawrence Berkeley National Lab (LBNL).

Xu, T., Spycher, N., Sonnenthal, E., Zhang, G., Zheng, L., & Pruess, K. (2011). TOUGHREACT Version 2.0: A simulator for subsurface reactive transport under non-isothermal multiphase flow conditions. Computers & Geosciences, 37(6), 763–774. https://doi.org/10.1016/j.cageo.2010.10.007

Xu, X., Sun, C., Li, Y., Zhao, J., Han, J., & Huang, W. (2021). An improved tandem neural network for the inverse design of nanophotonics devices. Optics Communications, 481, 126513. https://doi.org/10.1016/j.optcom.2020.126513

Yan, L., & Zhou, T. (2019a). Adaptive multi-fidelity polynomial chaos approach to Bayesian inference in inverse problems. Journal of Computational Physics, 381, 110–128. https://doi.org/10.1016/j.jcp.2018.12.025

Yan, L., & Zhou, T. (2019b). An adaptive surrogate modeling based on deep neural networks for large-scale Bayesian inverse problems. arXiv preprint arXiv:1909.01914.