

# Deep Learning for Simultaneous Inference of Hydraulic and Transport Properties

Zitong Zhou<sup>1</sup>, Nicholas J Zabaraz<sup>2</sup>, and Daniel M Tartakovsky<sup>1</sup>

<sup>1</sup>Stanford University

<sup>2</sup>University of Notre Dame

November 24, 2022

## Abstract

Identifying the heterogeneous conductivity field and reconstructing the contaminant release history are key aspects of subsurface remediation. Achieving these two goals with limited and noisy hydraulic head and concentration measurements is challenging. The obstacles include solving an inverse problem for high-dimensional parameters, and the high-computational cost needed for the repeated forward modeling. We use a convolutional adversarial autoencoder (CAAE) for the parameterization of the heterogeneous non-Gaussian conductivity field with a low-dimensional latent representation. Additionally, we trained a three-dimensional dense convolutional encoder-decoder (DenseED) network to serve as the forward surrogate for the flow and transport processes. Combining the CAAE and DenseED forward surrogate models, the ensemble smoother with multiple data assimilation (ESMDA) algorithm is used to sample from the Bayesian posterior distribution of the unknown parameters, forming a CAAE-DenseED-ESMDA inversion framework. We applied this CAAE-DenseED-ESMDA inversion framework in a three-dimensional contaminant source and conductivity field identification problem. A comparison of the inversion results from CAAE-ESMDA with physical flow and transport simulator and CAAE-DenseED-ESMDA is provided, showing that accurate reconstruction results were achieved with a much higher computational efficiency.

# Deep Learning for Simultaneous Inference of Hydraulic and Transport Properties

Zitong Zhou<sup>1</sup>, Nicholas Zabararas<sup>2</sup>, and Daniel M. Tartakovsky<sup>1</sup>

<sup>1</sup>Department of Energy Resources Engineering, Stanford University, Stanford, CA 94305, USA

<sup>2</sup>Scientific Computing and Artificial Intelligence (SCAI) Laboratory, University of Notre Dame, South Bend, IN 46556, USA

## Key Points:

- We present a deep-learning strategy to reconstruct conductivity and contaminant release history in three-dimensional heterogeneous aquifers.
- Conductivity parameterization with convolutional adversarial autoencoder reduces the inverse problem's dimensionality.
- Convolutional encoder-decoder acts as a surrogate of forward models; ensemble smoother approximates parameters' posterior distribution.

---

Corresponding author: Daniel M. Tartakovsky, [tartakovsky@stanford.edu](mailto:tartakovsky@stanford.edu)

Corresponding author: Nicholas Zabararas, [nzabararas@gmail.com](mailto:nzabararas@gmail.com)

**Abstract**

Identifying the heterogeneous conductivity field and reconstructing the contaminant release history are key aspects of subsurface remediation. Achieving these two goals with limited and noisy hydraulic head and concentration measurements is challenging. The obstacles include solving an inverse problem for high-dimensional parameters, and the high-computational cost needed for the repeated forward modeling. We use a convolutional adversarial autoencoder (CAAE) for the parameterization of the heterogeneous non-Gaussian conductivity field with a low-dimensional latent representation. Additionally, we trained a three-dimensional dense convolutional encoder-decoder (DenseED) network to serve as the forward surrogate for the flow and transport processes. Combining the CAAE and DenseED forward surrogate models, the ensemble smoother with multiple data assimilation (ESMDA) algorithm is used to sample from the Bayesian posterior distribution of the unknown parameters, forming a CAAE-DenseED-ESMDA inversion framework. We applied this CAAE-DenseED-ESMDA inversion framework in a three-dimensional contaminant source and conductivity field identification problem. A comparison of the inversion results from CAAE-ESMDA with physical flow and transport simulator and CAAE-DenseED-ESMDA is provided, showing that accurate reconstruction results were achieved with a much higher computational efficiency.

**1 Introduction**

Design of regulatory and remedial actions for contaminated soils and aquifers rely on reconstruction of the contaminant release history. Given subsurface heterogeneity, this task is inseparable from the need to identify hydraulic and transport properties of the subsurface environment. Both tasks have to contend with sparse and noisy measurements collected many years or decades after the contamination event took place. Prior to recent breakthroughs in computer architecture and algorithmic development, this joint inversion of hydraulic and water-quality data for real-world problems was so demanding computationally as to defy a solution unless dramatic (and often unrealistic) simplifications of the problem were made. For example, past efforts to reconstruct a contaminant release history found it necessary to assume solute migration to be one- or two-dimensional and subsurface properties, such as hydraulic conductivity  $K(\mathbf{x})$ , to be known with certainty (Aral et al., 2001; Snodgrass & Kitanidis, 1997; Yeh et al., 2007, among many others). Yet, aquifers are seldom, if ever, homogeneous, with  $K(\mathbf{x})$  often varying by orders of magnitude within the same aquifer and exhibiting highly non-Gaussian, multimodal behavior (Tartakovsky & Winter, 2008; Winter et al., 2003; Yang et al., 2020). Likewise, while the assumption of two-dimensional groundwater flow is often valid, accounting for the three-dimensional nature of contaminant migration is essential to prediction accuracy.

Our effort in joint inversion of hydraulic conductivity and contaminant release history from error-prone measurements of hydraulic head and solute concentration revolves around two challenges. The first is to describe the unknown non-Gaussian heterogeneous conductivity field with an adequate prior distribution. The second is to estimate a large number of unknown parameters in the inverse problem.

To tackle the first challenge, a parameterization of the high-dimensional conductivity field with a low-dimensional latent variable is commonly used (Linde et al., 2015; H. Zhou et al., 2014). Parameterizations based on the principle component analysis (PCA) (Sarma et al., 2008; Vo & Durlofsky, 2014) perform well for Gaussian random fields, but require ad-hoc modifications for non-Gaussian fields (Liu et al., 2019). DNN-based parameterizations eliminate the need for the Gaussianity assumption (Canchumuni et al., 2019; Liu et al., 2019). Two popular methods of this class are the generative adversarial network (GAN) (I. J. Goodfellow et al., 2014) and

33 variational autoencoders (Kingma & Welling, 2013). Both produce a DNN that learns  
34 a two-way mapping between the conductivity field and a low-dimensional embedding.  
35 Random realizations from the latent variable distribution can be decoded to a con-  
36 ductivity field that is statistically similar to those drawn from the training data set.  
37 In addition to the large reduction in the number of the unknown parameters, such  
38 parameterizations make it feasible to tackle the latent variable distribution, which is  
39 typically a standard normal by construction. This simplicity, in turn, facilitates the  
40 solution of the inverse problem with ensemble methods discussed below.

41 The second challenge, high-dimensionality of the parameter space, manifests itself  
42 in significant computational burden of an inversion procedure. Parameter estimation,  
43 which lies at the heart of an inverse problem, is achieved by matching the noisy mea-  
44 surements with the prediction of a flow and solute transport model. Strategies for  
45 solving typical ill-posed inverse problems fall into two main categories, determinis-  
46 tic and probabilistic. Deterministic methods, such as least square regression (White,  
47 2015) and hybrid optimization with a genetic algorithm (Ayvaz, 2016; Leichombam  
48 & Bhattacharjya, 2018), seek a “best” estimate of the unknown parameters, without  
49 quantifying the uncertainty inherent in this type of problems. Probabilistic methods,  
50 such as Markov Chain Monte Carlo or MCMC (Gamerman & Lopes, 2006) and data  
51 assimilation via Kalman filters (Evensen, 1994, 2003; Xu & Gómez-Hernández, 2016,  
52 2018) and their variants (Emerick & Reynolds, 2013; Zhang et al., 2018), overcome  
53 this shortcoming of their deterministic counterparts. Yet, the high-cost of necessary  
54 repeated forward solves undermines their utility for large, complex inverse problems,  
55 unless dedicated high-performance computing facilities are available for the task.

56 Two complementary strategies can be deployed to alleviate this cost. The first  
57 aims to reduce the number of forward simulations needed for an inversion algorithm  
58 to converge. The second seeks to reduce the computational cost of each forward  
59 solve. Efforts in the former direction include the design of efficient MCMC vari-  
60 ants, such as delayed rejection adaptive Metropolis (DRAM) sampling (Haario et al.,  
61 2001, 2006) which slightly outperforms a random walk Metropolis-Hastings MCMC in  
62 terms of efficiency (Zhang et al., 2015; Z. Zhou & Tartakovsky, 2021; Xia & Zabaras,  
63 2021). Gradient-based MCMC methods, such as hybrid Monte Carlo (HMC) sam-  
64 pling (Barajas-Solano et al., 2019), converge faster than these and other MCMC  
65 variants. However, computation of the gradient of a Hamiltonian dynamical sys-  
66 tem is prohibitive for high-dimensional transport problems. Learning on statistical  
67 manifolds provides another possible solution (Boso & Tartakovsky, 2020b, 2020a).  
68 Ensemble-based inversion methods are generally faster since they allow nearly perfect  
69 parallelization, because of the independence of samples in the ensemble. Variants of  
70 Kalman filters, such as iterative Ensemble Kalman filter (IEnKF), have been used for  
71 estimation of three-dimensional heterogeneous permeability fields (Chaudhuri et al.,  
72 2018). A relatively new variant of Kalman filter, ensemble smoother with multiple data  
73 assimilation (ESMDA) (Emerick & Reynolds, 2013), has gained popularity in subsur-  
74 face flow history matching (Tang et al., 2021; Kim et al., 2019; Jiang & Durllofsky,  
75 2021).

76 In terms of inversion complexity, we subdivide recent groundwater-related stud-  
77 ies into three categories: estimation of hydraulic conductivity from measurements of  
78 hydraulic head and, optionally, of solute concentration (Mo et al., 2019b; Ju et al.,  
79 2018); estimation of contaminant release history from concentration measurements,  
80 for known flow and transport parameters (Z. Zhou & Tartakovsky, 2021; Zhang et al.,  
81 2015); and estimation of both contaminant release history and hydraulic conductiv-  
82 ity from hydraulic head and solute concentration data in two- (Mo et al., 2019a; Xu  
83 & Gómez-Hernández, 2018; Kang et al., 2021) and three-dimensional (Kang et al.,  
84 2020) aquifers. We briefly discuss the latter category to highlight the novelty of our  
85 approach.

86 A low-dimensional representation of the random log-normal conductivity ob-  
 87 tained via the Karhunen-Loève expansion (KLE) in (Mo et al., 2019a) loses its at-  
 88 tractiveness if the subsurface environment is highly heterogeneous, exhibiting short  
 89 correlation lengths and multimodal statistics; additionally, this study relies on a lin-  
 90 ear transport model. The deep learning-based strategies of ensemble inversion were  
 91 adopted in (Xu & Gómez-Hernández, 2018; Kang et al., 2021) to estimate both a  
 92 non-Gaussian conductivity field and the source of contamination, yet their reported  
 93 accuracy is relatively low. In the adjacent field of petroleum engineering, CNN post-  
 94 processing of PCA (CNN-PCA) parameterization and ESMDA were used to estimate  
 95 both a channelized permeability and the oil/water rate (Tang et al., 2021). However,  
 96 this application deals with an observable quantity (the oil/water production rate),  
 97 while ours has to contend with an unobservable one (the location and strength of a  
 98 contaminant release).

99 To address the shortcomings of the joint inversion strategies mentioned above, we  
 100 use a convolutional adversarial autoencoder (CAAE) to parameterize a non-Gaussian  
 101 conductivity field (Mo et al., 2019b), train a surrogate dense encoder-decoder DNN to  
 102 replace the PDE-based model of subsurface flow and transport, and apply the ESMDA  
 103 inversion framework to identify the spatiotemporally extended source of contamination  
 104 and the latent variables representing the conductivity field. We posit that combination  
 105 of these three components, which yields the method we refer to as CAAE–DenseED–  
 106 ESMDA, provides a fast and robust inversion solution.

107 In Section 2, we formulate the problem of joint reconstruction of hydraulic con-  
 108 ductivity field and contaminant release history from sparse and noisy measurements  
 109 of hydraulic head and solute concentration. Our inversion strategy, combining CAAE  
 110 parameterization of the conductivity field (Section 3.2), a convolutional DNN sur-  
 111rogate of the flow and transport model (Section 3.3), and the ESMDA inversion  
 112 method (Section 3.1), is described in Section 3. Results of our numerical experi-  
 113 ments are reported in Section 4; they demonstrate that our method is about 8 times  
 114 faster than CAAE-ESMDA with the PDE-based flow and transport model. Main  
 115 conclusions drawn from this study are summarized in Section 5. The details of the  
 116 neural network architectures are given in Appendix A. The datasets and the code  
 117 needed to reproduce the results can be downloaded after publication of this work from  
 118 <https://github.com/zabaras/3dinversion>.

## 119 2 Problem Formulation

120 The problem formulation consists of the description of a reactive transport model  
 121 (Section 2.1) and the specification of a data model (Section 2.2).

### 122 2.1 Contaminant Transport Model

123 We consider transport of a reactive solute in a three-dimensional steady-state  
 124 groundwater flow field. The latter is described by:

$$125 \quad \nabla \cdot (K \nabla h) = 0, \quad \mathbf{x} = (x_1, x_2, x_3)^\top \in \Omega \subset \mathbb{R}^3, \quad (1)$$

126 where  $K(\mathbf{x})$  is the hydraulic conductivity of the aquifer  $\Omega$ , and  $h(\mathbf{x})$  is the hydraulic  
 127 head. This PDE is subject to appropriate boundary conditions on the simulation  
 128 domain boundary  $\partial\Omega$ . After the flow equation is solved, the average pore velocity  
 129  $\mathbf{u}(\mathbf{x}) = (u_1, u_2, u_3)^\top$  is computed from Darcy’s law,

$$130 \quad \mathbf{u} = -\frac{K}{\theta} \nabla h, \quad (2)$$

131 where  $\theta(\mathbf{x})$  is the aquifer’s porosity.

132 Starting at some unknown time  $t_0$ , a contaminant with volumetric concentration  
 133  $c_s$  enters the aquifer through either point-wise or spatially distributed sources  $\Omega_s \subset \Omega$ .  
 134 The contaminant is released for an unknown duration  $T$  with unknown intensity  $q_s(\mathbf{x}, t)$   
 135 (volumetric flow rate per unit source volume), such that  $q_s(\mathbf{x}, t) \neq 0$  for  $t_0 \leq t \leq$   
 136  $t_0 + T$ . The contaminant is advected by the flow, while undergoing hydrodynamic  
 137 dispersion and sorption to the solid matrix with rate  $R_n$ . Without loss of generality,  
 138 the spatiotemporal evolution of the contaminant's volumetric concentration  $c(\mathbf{x}, t)$  is  
 139 described by an advection-dispersion-reaction equation:

$$140 \quad \frac{\partial \theta c}{\partial t} = \nabla \cdot (\theta \mathbf{D} \nabla c) - \nabla \cdot (\theta \mathbf{u} c) - R_n(c) + q_s c_s, \quad \mathbf{x} \in \Omega, \quad t > t_0, \quad (3)$$

141 where the dispersion coefficient  $\mathbf{D}$  is a semi-positive second-rank tensor. If the coordi-  
 142 nate system is aligned with the mean flow direction, such that  $\mathbf{u} = (u \equiv |\mathbf{u}|, 0, 0)^\top$ ,  
 143 then the components of this tensor are:

$$144 \quad D_{11} = \theta D_m + \alpha_L u, \quad D_{22} = \theta D_m + \alpha_T u, \quad D_{33} = \theta D_m + \alpha_C u, \quad D_{ij} = \theta D_m \text{ for } i \neq j, \quad (4)$$

145 where  $D_m$  is the coefficient of molecular diffusion for the contaminant in free water;  
 146  $\alpha_L$  is the longitudinal dispersivity; and  $\alpha_T$  and  $\alpha_C$  are transverse dispersivities in the  
 147  $x_2$  and  $x_3$  directions, respectively.

The chemical reactions considered represent sorption of the dissolved contami-  
 nant onto the solid surface of the porous media. Thus, the reaction terms  $R_n(c)$  take  
 the form:

$$R_n(c) = -\rho_b \frac{\partial \tilde{c}}{\partial t}, \quad (5)$$

148 where  $\rho_b$  is the bulk density and  $\tilde{c}$  is the concentration sorbed.

We assume the system to be in local chemical equilibrium, i.e., sorption to be  
 much faster than advection and dispersion. We also assume that sorption does not  
 affect the porosity  $\theta$ , which remains constant throughout the simulations. With these  
 assumptions, (3) reduces to:

$$R\theta \frac{\partial c}{\partial t} = \nabla \cdot (\theta \mathbf{D} \nabla c) - \nabla \cdot (\theta \mathbf{u} c) + q_s c_s, \quad (6)$$

wherein  $R(c)$  is the dimensionless retardation factor defined as:

$$R = 1 + \frac{\rho_b}{\theta} \frac{\partial \tilde{c}}{\partial c}. \quad (7)$$

A sorption isotherm defines the relationship between the sorbed concentration,  $\tilde{c}$ , and  
 the dissolved concentration,  $c$ . Among the popular isotherms—linear, Langmuir, and  
 Freundlich—we adopt the latter, for the sake of concreteness. According to the Fre-  
 undlich isotherm,

$$\tilde{c} = K_f c^a, \quad (8)$$

149 where  $K_f$  is the Freundlich constant,  $(L^3 M^{-1})^a$ ; and  $a$  is the Freundlich exponent.  
 150 The units of all relevant transport quantities are summarized in Table 1.

**Table 1.** *Quantities in the transport model (6) and their units.*

Term	Physical quantity	Units
$c$	dissolved concentration	$\text{ML}^{-3}$
$\theta$	porosity of the subsurface medium	-
$x_i$	the distance along the respective Cartesian coordinate axis	L
$D_{ij}$	hydrodynamic dispersion coefficient tensor	$\text{L}^2\text{T}^{-1}$
$u_i$	pore water velocity	$\text{LT}^{-1}$
$q_s$	volumetric flow rate per volume, sources (+) and sinks (-)	$\text{T}^{-1}$
$c_s$	concentration of source or sink flux	$\text{ML}^{-3}$
$R_n$	chemical reaction term	$\text{ML}^{-3}\text{T}^{-1}$
$\rho_b$	bulk density of the medium	$\text{ML}^{-3}$
$\tilde{c}$	concentration sorbed	$\text{ML}^{-3}$
$K_f$	Freundlich constant	$(\text{L}^3\text{M}^{-1})^a$
$a$	Freundlich exponent	-

## 2.2 Parameters of Interest

Our goal is to identify the conductivity field  $K(\mathbf{x})$  and the contaminant source  $q_s c_s(\mathbf{x}, t)$ , given the flow and transport models, (1)–(8), and measurements of contaminant concentration and hydraulic head. Other parameters in the transport model, such as porosity, reaction term coefficients, etc., are assumed to be known. Identification of the source term  $q_s c_s$  is tantamount to finding the location(s),  $\mathbf{S}_1$ , and strength,  $\mathbf{S}_s \in \mathbb{R}^{N_{\text{re}}}$ , of the contaminant source; with the elements  $S_{sj}$  ( $j = 1, \dots, N_{\text{re}}$ ) of the vector  $\mathbf{S}_s$  denoting the release strength at the  $j$ -th time interval.

Measurements of hydraulic head,  $\bar{h}_m = \bar{h}(\mathbf{x}_m)$ , and solute concentration,  $\bar{c}_{mi} = \bar{c}(\mathbf{x}_m, t_i)$ , are collected at locations  $\{\mathbf{x}_m\}_{m=1}^M$  at times  $\{t_i\}_{i=1}^I$ . In lieu of field observations, we generate these data by corrupting the solution of (1)–(8) obtained for the reference parameter values by random measurement errors  $\epsilon_{mi}^c$  and  $\epsilon_m^h$ , such that:

$$\bar{c}_{m,i} = c(\mathbf{x}_m, t_i) + \epsilon_{mi}^c, \quad \bar{h}_m = h(\mathbf{x}_m) + \epsilon_m^h; \quad m = 1, \dots, M, \quad i = 1, \dots, I, \quad (9)$$

where  $c(\mathbf{x}_m, t_i)$  and  $h(\mathbf{x}_m)$  are the model predictions. The zero-mean Gaussian random variables  $\epsilon_{mi}^c$  have covariance  $\mathbb{E}[\epsilon_{mi}^c \epsilon_{nj}^c] = \delta_{ij} R_{mn}^c$ , where  $\mathbb{E}[\cdot]$  denotes the ensemble mean;  $\delta_{ij}$  is the Kronecker delta function; and  $R_{mn}^c$  with  $m, n \in [1, M]$  are components of the  $M \times M$  spatial covariance matrix  $\mathbf{R}^c$  of measurements errors. To be specific, we set  $\mathbf{R}^c = \sigma_c^2 \mathbf{I}$ , where  $\sigma_c$  is the standard deviation of the measurement errors, and  $\mathbf{I}$  is the  $(M \times M)$  identity matrix. The hydraulic head measurement errors  $\epsilon_m^h$  are zero-mean Gaussian random variables with covariance  $\mathbb{E}[\epsilon_m^h \epsilon_n^h] = R_{mn}^h$  with  $m, n \in [1, M]$ . We set  $\mathbf{R}^h = \sigma_h^2 \mathbf{I}$ , where  $\sigma_h$  is the standard deviation of the measurement errors.

The error model in (9) assumes the flow and transport models (1)–(8) to be exact and the measurements errors to be unbiased and uncorrelated in time but not in space. The groundwater flow equation is solved with MODFLOW (Harbaugh, 2005), and the solute transport equation with MT3DMS (Zheng & Wang, 1999; Bedekar et al., 2016). We use Flopy (Bakker et al., 2016), a Python implementation of these two packages.

## 3 Methodology

Below we describe the three elements of our inversion framework: ensemble smoother with multiple data assimilation (ESMDA), convolutional adversarial autoencoder (CAAE) parameterization of the conductivity field, and a Dense encoder-decoder (DenseED) neural network surrogate of the forward model.

### 3.1 Ensemble Smoother with Multiple Data Assimilation (ESMDA)

Upon a spatiotemporal discretization, the uncertain (random) input parameters in (1)–(4) are rearranged into a vector  $\mathbf{m}$  of length  $N_m$ ; these inputs include the discretized source term ( $\mathbf{S}_1, \mathbf{S}_s$ ) and hydraulic conductivity  $K(\mathbf{x})$  in all discretized cells (in the applications in this study,  $K(\mathbf{x})$  is parameterized with a low dimensional variable, illustrated in detail in Section 3.2). Similarly, we arrange the random measurements  $\bar{c}_{m,i}$  and  $\bar{h}_m$  into a vector  $\mathbf{d}$  of length  $N_d = M(I + 1)$ , and the random measurement noise  $\epsilon_{mi}^c$  and  $\epsilon_m^h$  into a vector  $\boldsymbol{\varepsilon}$  of the same length. Then, the error model (9) takes the vector form,

$$\mathbf{d} = \mathbf{g}(\mathbf{m}) + \boldsymbol{\varepsilon}, \quad (10)$$

where  $\mathbf{g}(\cdot)$  is the vector, of length  $N_d$ , of the correspondingly arranged stochastic model predictions  $c(\mathbf{x}_m, t_i)$  and  $h(\mathbf{x}_m)$  predicated on the model inputs  $\mathbf{m}$ . Let  $\pi(\mathbf{m})$  denote a prior PDF of the inputs  $\mathbf{m}$ , which encapsulates the knowledge about the aquifer’s properties and contaminant source before any measurements are assimilated. Our goal is to improve this prior by assimilating the measurements  $\mathbf{d}$ , i.e., to compute the posterior PDF of the model parameters,  $\pi(\mathbf{m}|\mathbf{d})$ . This task is accomplished via Bayes’ rule,

$$\pi(\mathbf{m}|\mathbf{d}) = \frac{\pi(\mathbf{m})\pi(\mathbf{d}|\mathbf{m})}{\pi(\mathbf{d})}, \quad \pi(\mathbf{d}) = \int \pi(\mathbf{m})\pi(\mathbf{d}|\mathbf{m})d\mathbf{m}, \quad (11)$$

where  $\pi(\mathbf{d}|\mathbf{m})$  is the likelihood function; and  $\pi(\mathbf{d})$ , is the “evidence” that serves as a normalizing constant so that  $\pi(\mathbf{m})\pi(\mathbf{d}|\mathbf{m})$  integrates to 1.

To compute (11), we use ESMDA (Emerick & Reynolds, 2013), which is an ensemble updating method similar to ensemble smoother (ES) (Van Leeuwen & Evensen, 1996) or ensemble Kalman filter (EnKF) (Evensen, 1994, 2003). To place ESMDA in the proper perspective, we briefly describe ES. The method is initiated by drawing  $N_e$  samples  $\mathbf{M}^f = \{\mathbf{m}_1^f, \dots, \mathbf{m}_{N_e}^f\}$  from the prior PDF  $\pi(\mathbf{m})$ . These data are then updated as:

$$\mathbf{m}_j^a = \mathbf{m}_j^f + \mathbf{C}_{\text{MD}}^f (\mathbf{C}_{\text{DD}}^f + \mathbf{C}_{\text{D}})^{-1} [\mathbf{d}_{uc,j} - g(\mathbf{m}_j^f)], \quad j = 1, \dots, N_e, \quad (12)$$

forming  $\mathbf{M}^a = [\mathbf{m}_1^a, \dots, \mathbf{m}_{N_e}^a]$ , the updated ensemble conditioned on the measurements  $\mathbf{d}$ . Here,  $\mathbf{C}_{\text{D}} \in \mathbb{R}^{N_d \times N_d}$  is the covariance matrix of the measurement errors  $\boldsymbol{\varepsilon}$ ; we define an ensemble of perturbed measurements:  $\{\mathbf{d}_{uc,j}\}_{j=1}^{N_e}$ , which are obtained by sampling from the Gaussian distribution:  $\mathbf{d}_{uc,j} \sim \mathcal{N}(\mathbf{d}, \mathbf{C}_{\text{D}})$ ;  $\mathbf{C}_{\text{DD}}^f \in \mathbb{R}^{N_d \times N_d}$  is the auto-covariance matrix of the model predictions  $\mathbf{D}^f = \mathbf{D}^f = [g(\mathbf{m}_1^f), \dots, g(\mathbf{m}_{N_e}^f)]$ ; and  $\mathbf{C}_{\text{MD}}^f \in \mathbb{R}^{N_m \times N_d}$  is the cross-covariance matrix between  $\mathbf{M}^f$  and  $\mathbf{D}^f$ . During the update, all the data  $\mathbf{d}$  are used once, simultaneously. This global update may cause an unacceptably large mismatch between the model response and the measurements, which precipitated the development of an iterative ES with smaller-scale updates.

While ES performs a single large Gauss-Newton correction to the ensemble  $\mathbf{M}^f$ , ESMDA makes a smaller correction during each update and deploys the inflated covariance matrix  $\mathbf{C}_{\text{D}}$  to damp the changes in the ensemble at early iterations (Gao & Reynolds, 2004; Wu et al., 1999). (In the linear Gaussian case, ESMDA and ES yield identical results.) We use the following algorithm to implement ESMDA.

- Set the number of data assimilation iterations,  $N_a$ , and the corresponding inflation coefficients  $\alpha_i$   $i = 1, \dots, N_a$ . Requiring  $\sum_{i=1}^{N_a} \alpha_i = 1$  guarantees consistency with ES in the linear Gaussian case. Generate the initial ensemble  $\mathbf{m}_j^1$  ( $j = 1, \dots, N_e$ ) from the prior PDF  $\pi(\mathbf{m})$ .
- Repeat the following steps for  $i = 1, \dots, N_a$ :
  1. Run the forward simulation for each member  $\mathbf{m}_j^i$  ( $j = 1, \dots, N_e$ ) from the parameter ensemble  $\mathbf{M}^f$  to obtain the corresponding model predictions (and in the synthetic case, observations)  $g(\mathbf{m}_j^i)$ .

- 224 2. Perturb the measurements with inflated measurement noise:  $\mathbf{d}_{uc,j}^i \sim \mathcal{N}(\mathbf{d}, \alpha_i \mathbf{C}_D)$ .  
 225 3. Compute the cross covariance matrix  $\mathbf{C}_{MD}^i$  and the auto-covariance matrix  
 226 of the predicted data  $\mathbf{C}_{DD}^i$ .  
 227 4. Update the ensemble as in (12), but with  $\mathbf{C}_D$  replaced by  $\alpha_i \mathbf{C}_D$ :

$$228 \quad \mathbf{m}_j^{i+1} = \mathbf{m}_j^i + \mathbf{C}_{MD}^i (\mathbf{C}_{DD}^i + \alpha_i \mathbf{C}_D)^{-1} [\mathbf{d}_{uc,j}^i - g(\mathbf{m}_j^i)], \quad j = 1, \dots, N_e. \quad (13)$$

229 The inverse,  $\mathbf{C}_i^{-1}$ , of the matrix  $\mathbf{C}_i = \mathbf{C}_{DD}^i + \alpha_i \mathbf{C}_D$  is approximated by its pseudo-  
 230 inverse using a truncated singular value decomposition (TSVD).

### 231 3.2 CAAE Parameterization of Conductivity Field

232 Let the matrix  $\mathbf{k} \in \mathbb{R}^{W \times H \times D}$  denote the log-conductivity field  $\ln K(\mathbf{x})$  defined  
 233 on a three-dimensional numerical grid, which consists of  $W$ ,  $H$  and  $D$  elements in the  
 234 three spatial directions. We use CAAE to parameterize the high-dimensional  $\mathbf{k}$  with  
 235 a low-dimensional latent variable  $\mathbf{z}$ . The CAAE consists of two components, a GAN  
 236 and an autoencoder (AE).

237 The GAN (I. J. Goodfellow et al., 2014) is a DNN strategy for generating data  
 238 from complex distributions without having to actually acquire the full PDF. This  
 239 strategy comprises two networks: a generator  $\mathcal{G}(\cdot)$  that generates samples similar to  $\mathbf{k}$ ;  
 240 and a discriminator  $\mathcal{D}(\cdot)$  that is trained to distinguish between the generated samples  
 241 and the real data samples. By ‘‘playing an adversarial game’’, the discriminator  $\mathcal{D}(\cdot)$   
 242 improves its ability to catch flaws in the generated samples, and the generator  $\mathcal{G}(\cdot)$   
 243 improves its capacity to generate realistic samples that try to trick the discriminator.

244 The AE learns a low-dimensional representation  $\mathbf{z}$  of the data  $\mathbf{k}$ , and then gener-  
 245 ates a reconstruction  $\hat{\mathbf{k}}$  from  $\mathbf{z}$  that closely matches the original data  $\mathbf{k}$ . The encoded  
 246 latent variable  $\mathbf{z}$  is constructed to follow a PDF  $\pi(\mathbf{z})$  that is easy to sample from, e.g., a  
 247 standard normal PDF  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ . A variational autoencoder (VAE) (Kingma & Welling,  
 248 2013) forces the empirical PDF of  $\mathbf{z}$  computed from the samples of  $\mathbf{k}$ ,  $\pi(\mathbf{z}|\mathbf{k})$ , to be close  
 249 to the target PDF  $\pi(\mathbf{z})$  by adding the Kullback-Leibler divergence  $\text{KL}[\pi(\mathbf{z}|\mathbf{k})||\pi(\mathbf{z})]$   
 250 between the empirical and target PDFs to the total loss function:

$$251 \quad \mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{rec}}(\mathbf{k}, \hat{\mathbf{k}}) + \text{KL}[\pi(\mathbf{z}|\mathbf{k})||\pi(\mathbf{z})], \quad (14)$$

252 where  $\mathcal{L}_{\text{rec}}(\mathbf{k}, \hat{\mathbf{k}})$  is the discrepancy between the data  $\mathbf{k}$  and their reconstruction  $\hat{\mathbf{k}}$ .  
 253 Choices of this discrepancy function include  $L_1$  or  $L_2$  norm. We use the former to  
 254 define the average reconstruction error  $\mathcal{L}_{\text{rec}}$  over  $N$  training samples,

$$255 \quad \mathcal{L}_{\text{rec}} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{k}_i - \hat{\mathbf{k}}_i\|_1. \quad (15)$$

The CAAE differs from the VAE in the way it minimizes the discrepancy between  
 the empirical PDF  $\pi(\mathbf{z}|\mathbf{k})$  and the target PDF  $\pi(\mathbf{z})$  of the latent random variable  $\mathbf{z}$ .  
 Instead of minimizing the KL divergence  $\text{KL}[\pi(\mathbf{z}|\mathbf{k})||\pi(\mathbf{z})]$ , the adversarial autoencoder  
 (AAE) employs an adversarial training procedure to minimize this discrepancy. The  
 training of the encoder  $\mathcal{G}(\cdot)$ , decoder  $\text{De}(\cdot)$ , and the discriminator  $\mathcal{D}(\cdot)$  is divided  
 into the reconstruction phase and the regularization phase (Makhzani et al., 2016).  
 Parameters in the encoder and decoder are updated by minimizing the loss function:

$$256 \quad \mathcal{L}_{\text{ED}} = \mathcal{L}_{\text{rec}} + w\mathcal{L}_{\mathcal{G}}. \quad (16)$$

We use  $\mathcal{L}_{\mathcal{G}}$  to quantify the decoder’s ability to trick the discriminator,

$$257 \quad \mathcal{L}_{\mathcal{G}} = -\frac{1}{N} \sum_{i=1}^N \ln \{ \mathcal{D}[\mathcal{G}(\mathbf{k}_i)] \}. \quad (17)$$

258 The weight factor  $w$  in (16) is used to assign relative importance to these two sources  
 259 of error. In the simulations reported below, we set  $w = 0.01$ .

260 After the encoder and decoder are updated in the first training phase, the dis-  
 261 criminator  $\mathcal{D}(\cdot)$  is trained in the second phase to minimize the loss function:

$$262 \quad \mathcal{L}_{\mathcal{D}} = -\frac{1}{N} \sum_{i=1}^N \left\{ \ln [\mathcal{D}(\mathbf{z}_i)] + \ln [1 - \mathcal{D}[\mathcal{G}(\mathbf{k}_i)]] \right\}. \quad (18)$$

263 By iterating between these two training phases, one obtains the mappings from  $\mathbf{k}$  to  
 264  $\mathbf{z}$  and from  $\mathbf{z}$  to  $\hat{\mathbf{k}}$ , and the decoder reaches its goal of constructing realizations  $\hat{\mathbf{k}}_i$   
 265 similar to the data  $\mathbf{k}_i$ .

266 The architectures of each network in the CAAE in this study are adopted from (Mo  
 267 et al., 2019b), and illustrated with our modified schematics in Appendix A. We applied  
 268 slight modifications to fit the dimensions and specifics of the problem in this study.

### 269 3.3 DenseED Neural Networks as Forward Model Surrogates

270 ESMDA inversion requires a large number of forward solves of the PDE-based  
 271 model (1)–(8) for multiple realizations of the parameters  $\mathbf{m}$ . To alleviate the cost of  
 272 each forward run, we replace the PDE-based model with a CNN surrogate.

273 Several approaches to constructing an input-output surrogate are collated in Ta-  
 274 ble 2. We choose an autoregressive image-to-image (i-to-i) regression model, rather  
 275 than its image-to-sparse-observation counterpart, because of its superior generaliz-  
 276 ability (Z. Zhou & Tartakovsky, 2021). We choose an autoregressive i-to-i model  
 277 over a one-to-many i-to-i model based on computer-memory considerations: for three-  
 278 dimensional problems with  $I$  time steps, memory allocated for input and output can  
 279 be prohibitively large; also, the autoregressive scheme reduces the number of DNN  
 280 parameters needed for the regression task. The source location ( $\mathbf{S}_{1,t}$ ) and strength  
 281 ( $\mathbf{S}_{s,t}$ ) for the release period  $[t, t + \Delta t]$  are assembled into a three-dimensional matrix  
 282  $\mathbf{S}(\mathbf{x}, t) \in \mathbb{R}^{W \times H \times D}$ .

**Table 2.** *Alternative input-output frameworks for construction of a surrogate model. The data are collected at  $M$  locations  $\mathbf{x}_m$  ( $m = 1, \dots, M$ ) at  $I$  times  $t_i$  ( $i = 1, \dots, I$ ). The source location ( $\mathbf{S}_{1,t}$ ) and strength ( $\mathbf{S}_{s,t}$ ) for the release period  $[t, t + \Delta t]$  are assembled into a three-dimensional matrix  $\mathbf{S}(\mathbf{x}, t)$ .*

Model	Input	Output	Modeling frequency
PDE model	$\mathbf{m}$	$c(\mathbf{x}, t_i), h(\mathbf{x})$	1
Image-to-image	$\mathbf{m}$	$c(\mathbf{x}, t_i), h(\mathbf{x})$	1
Image-to-sensors	$\mathbf{m}$	$c(\mathbf{x}_m, t_i), h(\mathbf{x}_m)$	1
Autoregressive i-to-i	$c(\mathbf{x}, t), \ln K(\mathbf{x}), S(\mathbf{x}, t)$	$c(\mathbf{x}, t + \Delta t), h(\mathbf{x})$	$I$

An autoregressive surrogate  $\mathbf{NN}_{\text{auto}}$  replaces the PDF-based model:

$$283 \quad \mathbf{g} : \mathbf{m} \xrightarrow{\text{PDEs}} \{c(x_m, t_i), h(x_m)\}_{m,i=1}^{M,I} \quad (19)$$

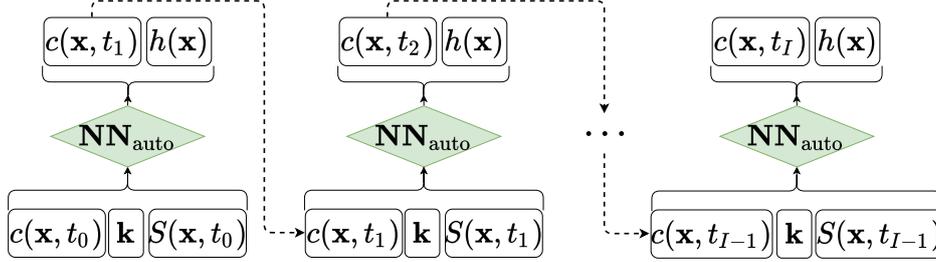
284 with a CNN that sequentially ( $I$  times) predicts the system state at the next time  
 step,

$$285 \quad \mathbf{NN}_{\text{auto}} : c(\mathbf{x}, t_i), K(\mathbf{x}), S(\mathbf{x}, t_i) \xrightarrow{\text{CNN}} \{c(\mathbf{x}, t_{i+1}), h(\mathbf{x})\}, \quad i = 0, \dots, I - 1. \quad (20)$$

287 If the three-dimensional simulation domain is discretized with a  $D \times H \times W$  grid,  
 288 then the autoregressive CNN surrogate  $\text{NN}_{\text{auto}}$  performs the following input-to-output  
 289 mapping:

$$290 \quad \text{NN}_{\text{auto}} : \mathbb{R}^{n_x \times W \times H \times D} \rightarrow \mathbb{R}^{n_y \times W \times H \times D}, \quad (21)$$

292 where  $n_x = 3$ , denotes the three channels representing the concentration  $c(\mathbf{x}, t_i)$  and  
 293 source terms  $S(\mathbf{x}, t_i)$  at time  $t_i$ , and the log-conductivity  $\ln K(\mathbf{x})$ ; and  $n_y = 2$  desig-  
 294 nates the two output channels representing the concentration  $c(\mathbf{x}, t_{i+1})$  at time  $t_{i+1}$   
 295 and the hydraulic head  $h(\mathbf{x})$ . A representative input-to-output example is shown in  
 296 Figure 1.



**Figure 1.** Autoregressive surrogate  $\text{NN}_{\text{auto}}$  of the PDE-based flow and transport model (1)–(8). The three input channels correspond to  $c(\mathbf{x}, t)$ ,  $S(\mathbf{x}, t)$ , and  $\ln K(\mathbf{x})$ . The two output channels correspond to  $c(\mathbf{x}, t + \Delta t)$  and  $h(\mathbf{x})$ . The concentration values at all time steps  $t_{i=1}^I$  are obtained through the iteration prediction with the autoregressive model.

297 We use a three-dimensional DenseED architecture to solve the image-to-image  
 298 regression task with a coarsen-refine process, with the convolutional operations. The  
 299 encoder extracts the high-level coarse features of the input maps, while the decoder  
 300 subsequently refines the coarse features to the full maps (Mo et al., 2019a, Fig. 2).  
 301 We use the  $L_1$ -norm loss function, the  $L_2$ -norm weight regularization, and stochastic  
 302 gradient descent (Bottou, 2010) in the CNN training process. A detailed description of  
 303 this surrogate model and its training procedure can be found in Appendix A, and (Mo  
 304 et al., 2019a). We have extended their procedure by adding the measurement locations  
 305 to the loss function. This allows us to penalize the prediction error at these specific  
 306 locations.

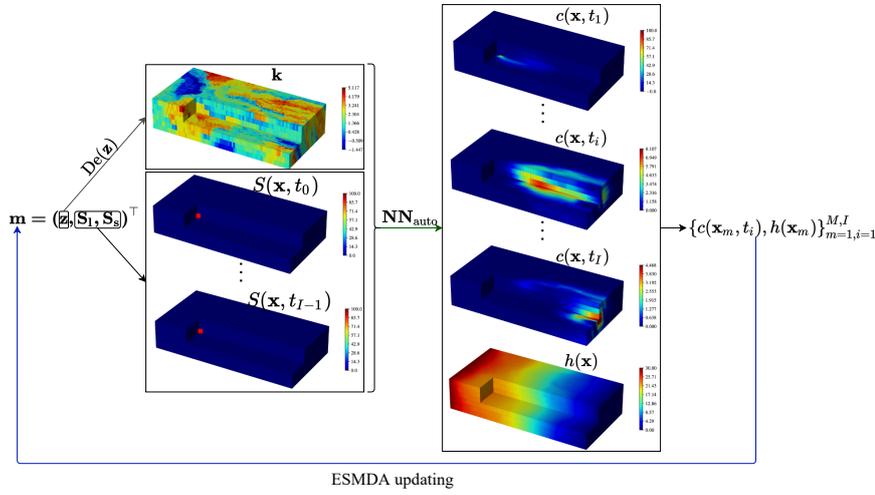
### 307 3.4 CAAE–DenseED–ESMDA Inversion Framework

308 We combine the CAAE parameterization of the conductivity field with the DenseED  
 309 CNN surrogate of the forward model to obtain fast and accurate predictions of concen-  
 310 tration  $c(\mathbf{x}, t)$  and  $h(\mathbf{x})$  for a given set of inputs. Then, we utilize ESMDA to identify  
 311 the unknown parameters, including the conductivity field and the source terms ( $\mathbf{S}_1, \mathbf{S}_s$ ).  
 312 The CAAE parameterization enables one to estimate the discretized log-conductivity  
 313 field  $\mathbf{k}$  through the latent variable  $\mathbf{z}$ . Our CAAE–DenseED–ESMDA inversion frame-  
 314 work is implemented in the following algorithm.

- 315 1. Train a CAAE; obtain the decoder  $\text{De}(\cdot)$  that maps the low-dimensional latent  
 316 variable  $\mathbf{z}$  back onto the log-conductivity field  $\mathbf{k}$ .
- 317 2. Train an autoregressive DenseED CNN  $\text{NN}_{\text{auto}}$  to predict  $c(\mathbf{x}, t)$  and  $h(\mathbf{x})$  for  
 318 a given conductivity field and contaminant release history.
- 319 3. Generate the initial input ensemble  $\mathbf{M}^f$  of size  $N_e$ , whose elements  $\mathbf{m}_j^1$  ( $j =$   
 320  $1, \dots, N_e$ ) are defined as  $\mathbf{m}_j^1 = (\mathbf{z}_j^1, \mathbf{S}_{1j}^1, \mathbf{S}_{sj}^1)^\top$ . Here,  $\mathbf{z}_j^1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_z)$  is the latent

- 321 variable for the log conductivity field; and  $\mathbf{S}_{1_j}^1$  and  $\mathbf{S}_{s_j}^1$  denote respectively the  
 322 source location and strength in all release periods, drawn from an appropriate  
 323 prior distribution.
- 324 4. Perform the ESM DA inversion with  $N_a$  data assimilation iterations and the  
 325 inflation coefficients  $\alpha_i$  ( $i = 1, \dots, N_a$ ). For  $i = 1, \dots, N_a$ ,
- 326 (a) Obtain the log-conductivity realizations  $\mathbf{k}_j^i = \text{De}(\mathbf{z}_j^i)$  with  $j = 1, \dots, N_e$ ;
- 327 (b) Form the release configuration  $\{\mathbf{S}_{1,j}^i, \mathbf{S}_{s,j}^i\}$  into the input matrix  $\mathbf{S}_j^i$ , and pre-  
 328 dict  $c(\mathbf{x}, t)$  and  $h(\mathbf{x})$  at the measurement times and locations,  $\text{NN}_{\text{auto}}(\mathbf{m}_j^i)$   
 329 for all  $j$ ;
- 330 (c) Update the ensemble  $\mathbf{m}_j^i$  via ESM DA with  $\alpha_i$  to obtain  $\mathbf{m}_j^{i+1}$ .
- 331 5. The end result,  $\mathbf{m}_j^{N_a+1}$ , serves as the final ensemble from which PDFs of the log  
 332 conductivity field and the contaminant release parameters are estimated.

This algorithm is illustrated in the schematic in Figure 2 as well.



**Figure 2.** Schematic illustrating the CAAE-DenseED-ESMDA algorithm.  $\text{De}(\cdot)$  denotes the decoder that obtains the log-conductivity field from the latent variable  $\mathbf{z}$ ,  $\text{NN}_{\text{auto}}$  represents the autoregressive surrogate model which predicts the concentration and hydraulic head fields. The observations at the measurement time and locations  $c(\mathbf{x}_m, t_i), h(\mathbf{x}_m)$  are then used to update the parameter  $\mathbf{m}$ .

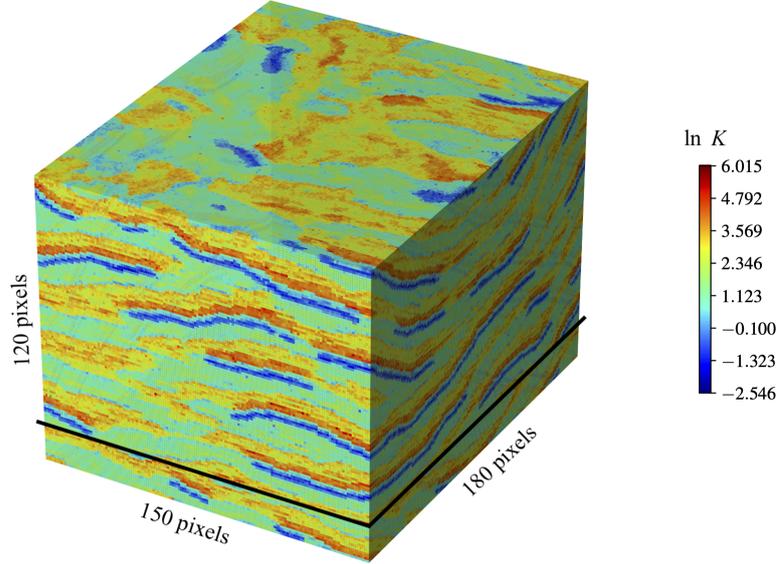
333

## 334 4 Numerical experiments

### 335 4.1 Experimental Setup

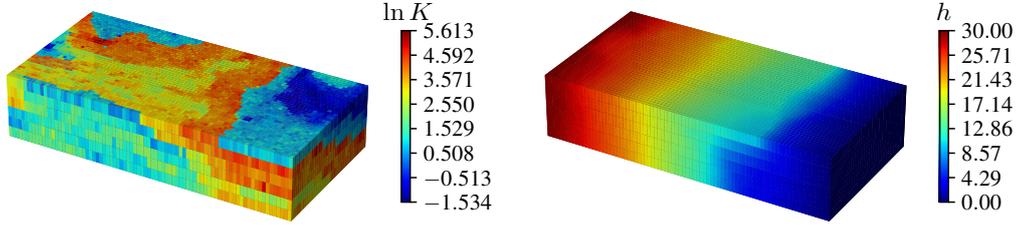
336 A confined heterogeneous aquifer is described as a rectangular cuboid  $\Omega$  of size  
 337  $2500 \text{ m} \times 1250 \text{ m} \times 300 \text{ m}$ ; it is discretized with a mesh consisting of  $81 \times 41 \times 6$  cells.  
 338 Groundwater flow is driven by constant heads  $h_L = 30 \text{ m}$  and  $h_R = 0 \text{ m}$  imposed  
 339 along the left ( $x_1 = 0$ ) and right ( $x_1 = 2500 \text{ m}$ ) facets of the cuboid, respectively;  
 340 the remaining boundaries are impermeable to flow. The hydraulic conductivity of this  
 341 aquifer,  $K(\mathbf{x})$ , is unknown (except when generating the ground truth); equiprobable  
 342 realizations of  $Y(\mathbf{x}) = \ln K(\mathbf{x})$  are generated by extracting  $81 \times 41 \times 6$  patches from  
 343 the  $150 \text{ px} \times 180 \text{ px} \times 120 \text{ px}$  training image (Mariethoz & Kelly, 2011) in Figure 3,  
 344 available at <https://github.com/GAIA-UNIL/trainingimages>. One such cropped

345 log-conductivity field  $Y(\mathbf{x})$  and the corresponding hydraulic head  $h(\mathbf{x})$ , obtained as  
 346 a solution of the groundwater flow equation (1), are shown in Figure 4. These fields  
 347 serve as the ground truth.



**Figure 3.** *Training image, consisting of  $150 \times 180 \times 120$  pixels. Equiprobable realizations of log-conductivity  $Y(\mathbf{x}) = \ln K(\mathbf{x})$  are generated by randomly selecting patches of size  $81 \times 41 \times 6$  pixels. The top  $150 \times 180 \times 105$  pixels serve as the training set. The bottom  $150 \times 180 \times 15$  pixels serve as the testing set. Conductivity  $K$  is in m/d.*

348 The porosity  $\theta$  and bulk density  $\rho$  of the soil; dispersivities  $\alpha_L$ ,  $\alpha_T$  and  $\alpha_C$ ;  
 349 and the parameters  $K_f$  and  $a$  of the Freundlich isotherm are constant. Values of  
 350 these transport parameters, which are representative of a sandy alluvial aquifer in  
 351 Southern California (Liggett et al., 2015), are presented in Table 3. The contami-  
 352 nant enters the aquifer via a point source, whose depth is known (the fourth layer  
 353 from the top of the domain) but the location in the horizontal plane ( $S_1^x$  and  $S_1^y$ ) is  
 354 uncertain. The contaminant release is known to occur during a 20-year period, but  
 355 its strength is uncertain. Following the standard practice in groundwater modeling,  
 356 we divide this time interval into  $N_{re} = 5$  sub-intervals (“stress periods” in the MOD-  
 357 FLOW/MT3D language) during each of which the release strength ( $S_s$ ) is constant.  
 358 In this configuration, the unknown contaminant release history is represented by the  
 359 vector  $\mathbf{S} = (\mathbf{S}_1, \mathbf{S}_s)$ , where  $\mathbf{S}_1 = (S_1^x, S_1^y)^\top$  and  $\mathbf{S}_s = (S_{s,1}, S_{s,2}, S_{s,3}, S_{s,4}, S_{s,5})^\top$ . The  
 360 values of  $\mathbf{S}$  used to generate the ground-truth concentrations are reported in Table 4.  
 361 Combined with the discretized version  $\mathbf{k}$  of the uncertain log-conductivity field  $Y(\mathbf{x})$ ,  
 362 this yields 19933 unknowns to be determined from the measurements of solute concen-  
 363 tration  $c(\mathbf{x}, t)$  and hydraulic head  $h(\mathbf{x})$ . Expert knowledge about possible location and  
 364 strength of the contaminant release is encapsulated in the uniform (“uninformative”)  
 365 prior distributions for  $\mathbf{S}_1$  and  $\mathbf{S}_s$ ), which are shown in Table 4.



**Figure 4.** Log-conductivity  $Y(\mathbf{x})$  (left) and the corresponding hydraulic head  $h(\mathbf{x})$  (right), which serve as ground truth and to generate measurements of  $h$  at observation wells. Conductivity  $K$  is in m/d and head  $h$  in m.

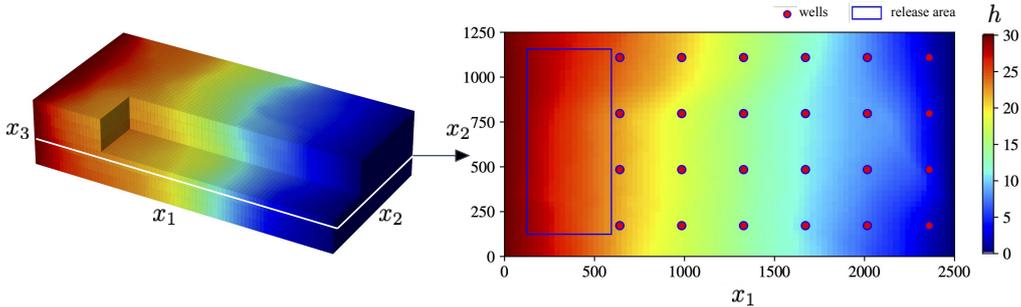
**Table 3.** Values of the transport parameters for a dissolved contaminant migrating in a generic sandy alluvial aquifer in Southern California (Liggett et al., 2015).

Property	Value	Units
$\phi$	0.3	–
$K_f$	0.1	(m <sup>3</sup> /g) <sup>a</sup>
$a$	0.9	–
$\rho$	1587	kg/m <sup>3</sup>
$\alpha_L$	35	m
$\alpha_T/\alpha_L$	0.3	–
$\alpha_C/\alpha_L$	0.3	–
$D_m$	10 <sup>-9</sup>	m <sup>2</sup> /d

**Table 4.** Parameters  $\mathbf{S}_l = (S_l^x, S_l^y)^\top$  and  $\mathbf{S}_s = (S_{s,1}, S_{s,2}, S_{s,3}, S_{s,4}, S_{s,5})^\top$  used to represent, respectively, the location and strength of a contaminant release. Reported as “Truth” are their (unknown) reference values used to generate ground truth and concentration measurements, and “Prior” the intervals on which their uniform priors,  $U[\cdot, \cdot]$ , are defined. The values of  $\mathbf{S}_l$  are in m, and of  $\mathbf{S}_s$  in g/m<sup>3</sup>.

	$S_1^x$	$S_1^y$	$S_{s,1}$	$S_{s,2}$	$S_{s,3}$	$S_{s,4}$	$S_{s,5}$
Truth	291	625	224	174	869	201	741
Prior	[125, 625]	[125, 1125]	[50, 1000]	[50, 1000]	[50, 1000]	[50, 1000]	[50, 1000]

366 These measurements are collected at observations wells, whose completion allows  
 367 one to collect water samples in each of the model’s six vertical layers. We consider  
 368 the observation wells whose locations are depicted in Figure 5. During the simulated  
 369 time horizon of 40 years, the contaminant concentration is sampled at  $I = 10$  time  
 370 intervals of four years each, and the hydraulic head is measured once since the flow  
 371 is at steady-state. The data at all space-time locations are generated by adding zero-  
 372 mean Gaussian measurement error with standard deviations  $\sigma_c = \sigma_h = 0.5$ , to the  
 373 solution  $\mathbf{g}(\mathbf{m})$  of the flow and transport model (1)–(8) with the input parameter values  
 374 identified as “ground truth” above, corresponding to 2.5% of the maximum value of  
 375 the concentration ( $\sim 20 \text{ g/m}^3$ ), and  $\sim 1.7\%$  of the maximum value of the hydraulic  
 376 head ( $\sim 30 \text{ m}$ ) on all sensor locations.

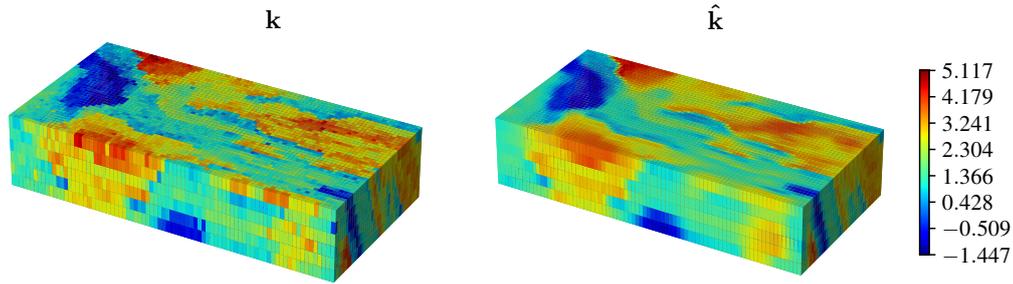


**Figure 5.** *Observational wells (red dots) in which measurements of hydraulic head  $h$  and solute concentration  $c$  are collected. The well locations are superposed on the ground-truth distribution of hydraulic head in the fourth layer of the MODFLOW model. The blue box represents a region of possible contaminant release from a point source that is known to be located in the fourth model-layer,  $x_1, x_2, x_3$  in m.*

## 377 4.2 CAAE Training for Conductivity Parameterization

378 We train a CAAE DNN to parameterize the discretized log-conductivity field  
 379  $\mathbf{k} \in \mathbb{R}^{81 \times 41 \times 6}$ . The end goal is an encoder  $\mathcal{G}(\mathbf{k})$  that maps an input field  $\mathbf{k}$  onto a  
 380 low-dimensional latent variable  $\mathbf{z} \in \mathbb{R}^{2 \times 2 \times 11 \times 21}$  with standard-Gaussian prior  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ ,  
 381 and a decoder  $\text{De}(\cdot)$  that reconstructs  $\mathbf{k}$  from this latent variable. The training is done  
 382 on 23000 realizations of  $\mathbf{k}$ , obtained as randomly selected  $(81 \times 41 \times 6)$  patches from the  
 383 top  $150 \text{ px} \times 180 \text{ px} \times 105 \text{ px}$  part of the large training image in Figure 3. Additional  
 384 2200 images cropped from the bottom  $150 \text{ px} \times 180 \text{ px} \times 15 \text{ px}$  part of the training  
 385 image serve as the testing set. The latent variable  $\mathbf{z}$  has  $2 \cdot 2 \cdot 11 \cdot 21 = 924$  elements.  
 386 With 50-epochs training and the learning rate of  $2 \cdot 10^{-4}$ , the Adam optimizer is used  
 387 to obtain the DNN parameters and, thus, build  $\mathcal{G}(\mathbf{k})$  and  $\text{De}(\mathbf{z})$ . The details of the  
 388 CAAE architecture are explained in Appendix A, and the dimensions of the internal  
 389 layers outputs are listed in Table 6.

390 A representative realization of  $\ln K(\mathbf{x}) \mapsto \mathbf{k}$  from the test set and its reconstruction  
 391 via the decoder,  $\hat{\mathbf{k}} = \text{De}(\mathbf{z})$ , are shown in Figure 6. After the training is complete,  
 392 the mean absolute error  $\|\mathbf{k} - \hat{\mathbf{k}}\|_1$ , averaged over all the elements of the numerical mesh  
 393 and over the 2200 members of the testing data set, is 0.2637. The reconstructed log-  
 394 conductivity field  $\hat{\mathbf{k}}$  captures the main structural features of its original counterpart  
 395  $\mathbf{k}$ . Some loss of information is unavoidable in reduced-order modeling but, overall,  
 396 the performance of this autoencoder is adequate to achieve accurate inverse modeling  
 397 results, as we show in Section 4.4 below.



**Figure 6.** A representative realization of  $\ln K(\mathbf{x}) \mapsto \mathbf{k}$  from the test set (left) and its reconstruction (right) via the CAAE encoder,  $\mathbf{z} = \mathcal{G}(\mathbf{k})$ , and decoder,  $\hat{\mathbf{k}} = \text{De}(\mathbf{z})$ .

398

### 4.3 DenseED Surrogate Model

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

As mentioned in Section 3, although only model predictions at the well locations are necessary for the inversion, a DNN that predicts  $c(\mathbf{x}, t_i)$  and  $h(\mathbf{x})$  at all points  $\mathbf{x}$  of the simulation domain has better generalization properties. We train our CNN on  $N = 800$  Monte Carlo realizations of the PDE-based model (1)–(8) with corresponding realizations of the input parameters  $\mathbf{m}$  (the discretized log-conductivity  $\mathbf{k}$  and contaminant release history  $\mathbf{S}$ ). Another set of  $N_{\text{test}} = 150$  realizations are retained for testing. These 950 realizations form  $950 \times 10$  autoregressive input-output pairs. The CNN contains three dense blocks with  $N_l = 3, 6,$  and  $3$  internal layers, has the growth rate of  $R_g = 48$  and  $N_{\text{in}} = 48$  initial features; it was trained for 200 epochs with the learning rate of  $5 \cdot 10^{-3}$ . We use the  $L_1$ -norm loss function and the  $L_2$ -norm weight regularization, apply stochastic gradient descent (Bottou, 2010) in the parameter estimation process, and add 5 times the  $L_1$ -norm loss at the source pixel and its surrounding pixels, 5 times the  $L_1$ -norm loss at the well locations to the total loss to penalize the prediction error at the source locations and the observation wells. The CNN’s output is the hydraulic head  $h(\mathbf{x})$  and the solute concentration  $c(\mathbf{x}, t_i)$  at the next time step  $t_i$ . The details of the architecture are explained in Appendix A and the dimensions of the internal layers outputs are listed in Table 7.

416

417

418

419

420

421

422

423

424

425

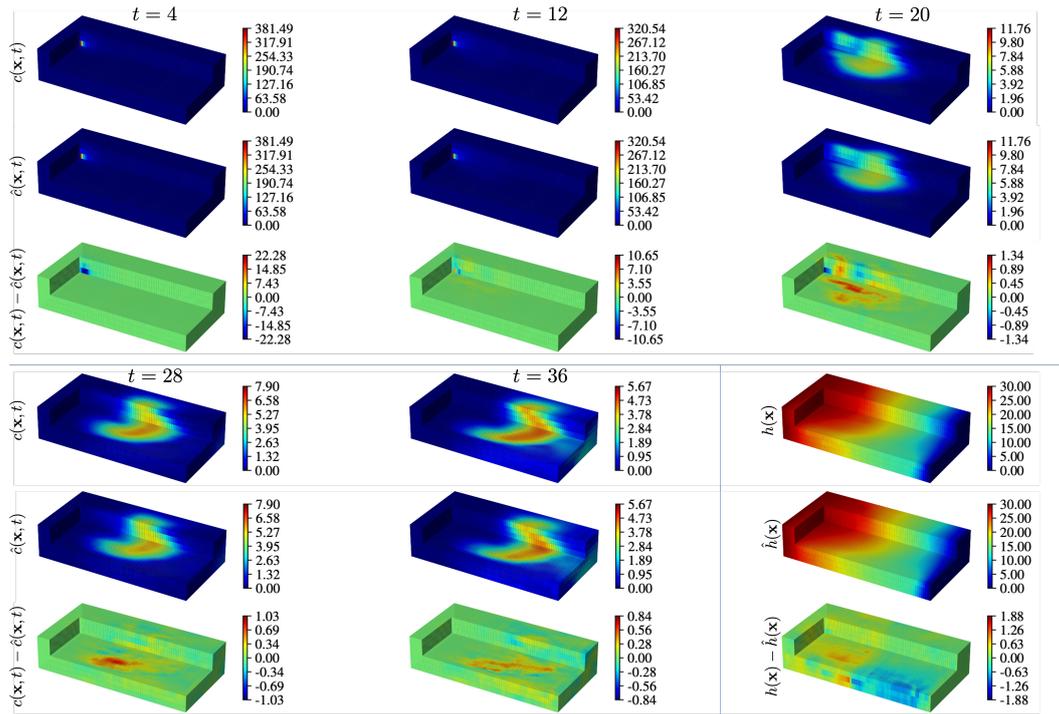
426

427

428

429

Figure 7 exhibits temporal snapshots of the solute concentrations alternatively predicted with the PDE-based model solved with MODFLOW and MT3DMS simulators,  $c(\mathbf{x}, t_i)$ , and the CNN surrogate,  $\hat{c}(\mathbf{x}, t_i)$ , for a given realization of the log-conductivity field and the contaminant release configuration (both drawn from the test set). Also presented are the hydraulic head maps predicted by the autoregressive model,  $\hat{h}(\mathbf{x})$ , and the PDE-based model solved with MODFLOW simulator,  $h(\mathbf{x})$ . The accuracy of our CNN surrogate is quantified by the total root mean square error,  $(\|c(\mathbf{x}, t) - \hat{c}(\mathbf{x}, t)\|_2 + \|h(\mathbf{x}) - \hat{h}(\mathbf{x})\|_2)/2$ . It falls to 0.853 at the end of the training process. It is worthwhile emphasizing here that the  $N_{\text{NMC}} = 800$  Monte Carlo realizations used to train the CNN surrogate are but a small fraction of the forward runs required by ESMDA inversion framework. One could achieve more accurate predictions for three-dimensional problems by either deploying a more complex DNN architecture (Wen et al., 2021; Mo et al., 2019b) or using much larger  $N_{\text{NMC}}$  or both. However, similar to the CAAE training, we focus on the development of efficient methodologies for three-



**Figure 7.** Predictions of the solute concentration obtained with the PDE-based model,  $c(\mathbf{x}, t)$ , and its DenseED CNN surrogate,  $\hat{c}(\mathbf{x}, t)$ , times  $t = 4, 12, 20, 28, 36$ ,  $t$  in year. Also shown are the corresponding predictions of the hydraulic head,  $h(\mathbf{x})$  and  $\hat{h}(\mathbf{x})$ ; and the difference between these two types  $c(\mathbf{x}, t) - \hat{c}(\mathbf{x}, t)$ ,  $h(\mathbf{x}) - \hat{h}(\mathbf{x})$  of prediction.

dimensional inverse modeling that accommodate the trade-off between the accuracy and computational feasibility.

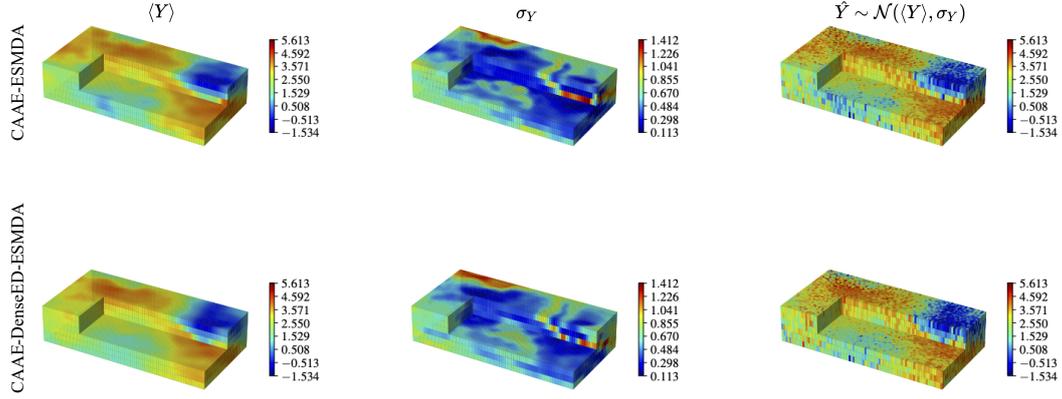
#### 4.4 ESMDA Inversion

We demonstrate the use of the CAAE parameterization and the DenseED CNN surrogate of the PDE-based forward model to accelerate the ESMDA inversion. The combination of these three techniques constitutes our CAAE-DenseED-ESMDA framework to approximate the joint posterior PDF of the uncertain model parameters  $\mathbf{m}$  consistent with both model predictions and field observations. In the simulations reported below, we select  $N_a = 10$  inflation factors in (13) and set their values to  $\alpha_i = 10$  for  $i = 1, \dots, N_a$ , and perform ESMDA with 10 iterations. To ascertain the impact of the DenseED CNN surrogate on the inversion accuracy, we also run CAAE-ESMDA with the PDE-based forward model implemented in MODFLOW and MT3DMS. The ensemble size for ESMDA for both CAAE-ESMDA and CAAE-DenseED-ESMDA are set to  $N_e = 960$ .

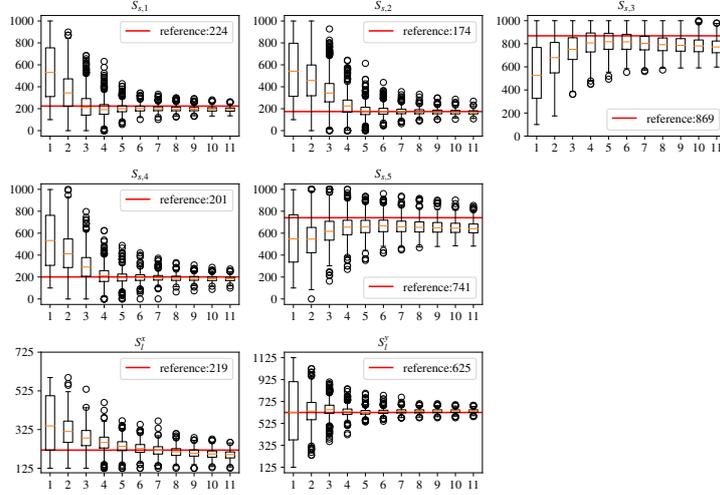
The measurements are taken at 24 wells that are completed in all 6 layers of the model, yielding  $24 \cdot 6 = 144$  measurements of the solute concentration and hydraulic head at each observation time, the hydraulic head is only measured once, resulting in  $144 \cdot (10 + 1) = 1584$  measurements in 40 years of the modeling time.

Figure 8 exhibits posterior statistics (mean  $\langle Y \rangle$  and standard deviation  $\sigma_Y$ ) of the log-conductivity  $Y(\mathbf{x})$ , obtained after the assimilation of all 1584 measurements via either CAAE-ESMDA or CAAE-DenseED-ESMDA. In both scenarios, the posterior ensemble mean  $\langle Y \rangle$ , reconstructed from the latent variable  $\mathbf{z}$ , correctly identifies the low-conductivity region in the right top region of the three-dimensional domain and the high-conductivity regions elsewhere. As expected, the mean log-conductivity fields,  $\langle Y \rangle$ , are smoother than the reference field  $Y$  (Figure 4), but the realizations drawn from the posterior Gaussian PDF  $\mathcal{N}(\langle Y \rangle, \sigma_Y)$  exhibit more realistic features (right column in Figure 8). Regardless of the forward model used (the only difference in these two experiments), our data assimilation framework yields consistent predictions of  $\sigma_Y$  (middle column in Figure 8). It is small throughout most of the domain, indicating the reduced uncertainty in the estimation of hydraulic conductivity  $K(\mathbf{x})$  due to assimilation of the concentration and head measurements. The maximum values of  $\sigma_Y$  and, hence, the largest predictive uncertainty in the  $K(\mathbf{x})$  estimation, are along the interface between the high- and low-conductivity regions. This finding suggests that the model predictions of hydraulic head and solute concentration are least sensitive to the changes in hydraulic conductivity in that domain; it reaffirms the conclusion of the sensitivity analysis of the relative importance of uncertainties in the spatial arrangement of hydrofacies and their hydraulic conductivities (Winter et al., 2006).

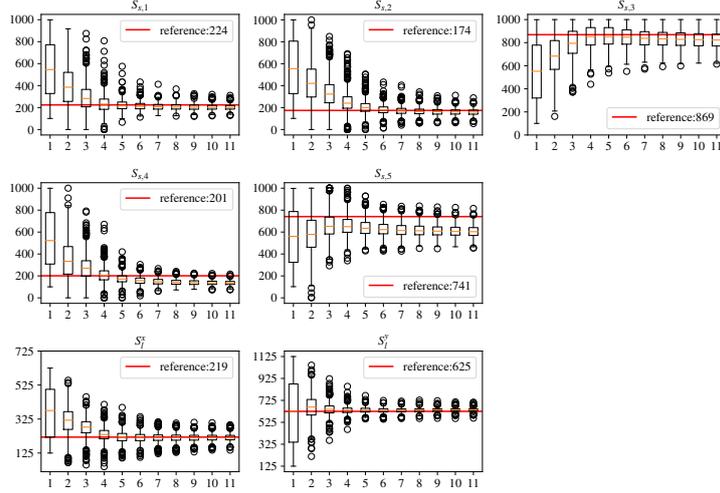
The same inversion experiments yield estimates of the contaminant release history  $\mathbf{S}$ , which are shown in Figures 9 and Figure 10 for CAAE-ESMDA and CAAE-DenseED-ESMDA, respectively. Regardless of the forward model used, our inversion algorithm accurately estimates the release strength during stress periods 1, 2, and 3 ( $S_{s,1}$ ,  $S_{s,2}$ , and  $S_{s,3}$ ); the estimates are close to their reference values and have tight 95% confidence intervals. At the same time, the estimates of the source strength during stress periods 4 and 5 ( $S_{s,4}$  and  $S_{s,5}$ ) fail to converge to their reference values and exhibit large error bars. The two assimilation strategies yield very similar estimates of the contaminant release location,  $\mathbf{S}_1 = (S_1^x, S_1^y)^\top$ ; the estimates of both quantities have tight confidence intervals, but the estimated value of  $S_1^x$  for CAAE-DenseED-ESMDA inversion lies slightly farther from the reference value than that of CAAE-ESMDA.



**Figure 8.** Posterior mean ( $\langle Y \rangle$ , left column) and standard deviation ( $\sigma_Y$ , middle column) of the log-conductivity field  $Y(\mathbf{x})$  obtained upon assimilation of concentration and head measurements from the dense observation network. These statistics are obtained via our inversion algorithm CAAE-ESMDA that relies on either the PDE-based forward model (top row) or its DenseED CNN surrogate (CAAE-DenseED-ESMDA, bottom row). Also shown are representative realizations of  $Y(\mathbf{x})$  drawn from the resulting posterior PDF  $\mathcal{N}(\langle Y \rangle, \sigma_Y)$  (right column).



**Figure 9.** Boxplots of the ensembles for the contaminant release terms,  $\mathbf{S} = (\mathbf{S}_l, \mathbf{S}_s)$  with  $\mathbf{S}_l = (S_l^x, S_l^y)^\top$  and  $\mathbf{S}_s = (S_{s,1}, \dots, S_{s,5})^\top$ , and their confidence intervals, obtained via the CAAE-ESMDA inversion with the PDE-based forward model. These quantities are plotted as function of the ESMDA iterations and contrasted with their reference values (horizontal lines). The source location  $\mathbf{S}_l$  is in  $m$ ; and the contaminant release strength in each of the five stress periods,  $\mathbf{S}_s$ , is in  $g/m^3$ .



**Figure 10.** *Boxplots of the ensembles for the contaminant release terms,  $\mathbf{S} = (\mathbf{S}_l, \mathbf{S}_s)$  with  $\mathbf{S}_l = (S_l^x, S_l^y)^\top$  and  $\mathbf{S}_s = (S_{s,1}, \dots, S_{s,5})^\top$ , and their confidence intervals, obtained via the CAAE-ESMDA inversion with the DenseED CNN surrogate. These quantities are plotted as function of the ESMDA iterations and contrasted with their reference values (horizontal lines). The source location  $\mathbf{S}_l$  is in  $m$ ; and the contaminant release strength in each of the five stress periods,  $\mathbf{S}_s$ , is in  $g/m^3$ .*

479

#### 4.5 Comparison of Computational Costs

480

481

482

483

484

485

486

487

The computational costs of CAAE-ESMDA with the PDE-based forward model and its counterpart with the DenseED CNN surrogate are shown in Table 5. CAAE-ESMDA with the PDE-based model ran on CPU, while the DenseED CNN surrogate was trained and simulated on GPUs provided by GoogleColab. In both cases, ESMDA consists of  $N_e = 960$  samples in each ensemble and 10 iterations are performed, resulting in  $N_{\text{sum}} = N_e \times (10 + 1) = 10560$  forward model runs. Overall, CAAE-DenseED-ESMDA is two-orders of magnitude faster than CAAE-ESMDA with the PDE-based forward model.

**Table 5.** *Total run time of the CAAE-ESMDA,  $T_{\text{run}}$ , includes the costs of the PDE-based forward model and its CNN surrogate. The average run-time per sample,  $T_{\text{ave}}$ , is defined as  $T_{\text{ave}} = (T_{\text{run}} + T_{\text{dataset}} + T_{\text{train}})/N_{\text{sum}}$ , where  $T_{\text{dataset}}$  is the time for obtaining the training and testing data sets, and  $T_{\text{train}}$  is the CNN training time. CAAE parameterization is used in both cases, the training time is 18678.23, the running time of CAAE is negligible in both data assimilation strategies. All times are in seconds.*

	$T_{\text{run}}$	$T_{\text{dataset}}$	$T_{\text{train}}$	$T_{\text{ave}}$
CAAE-ESMDA	360885.0	0.0	0.0	36.8
CAAE-DenseED-ESMDA	1893.9	34922.0	9439.2	4.4

## 5 Conclusions and Discussion

We proposed an CAAE-DenseED-ESMDA algorithm to infer the statistics of both aquifer properties (e.g., hydraulic conductivity) and contaminant release history from sparse and noisy observations of hydraulic head and solute concentration. The algorithm relies on CAEE to obtain a low-dimensional representation of the high-dimensional discretized conductivity field (and, if necessary, other spatially distributed input parameters); deploys a DenseED CNN surrogate of the PDE-based transport model to accelerate the forward runs; and adopts ESMDA to solve the inverse problem. The algorithm’s computational efficiency is such that it enables one to handle three-dimensional problems.

To demonstrate the salient features of our inversion methodology, we conduct a series of numerical experiments. They deal with flow and transport in a three-dimensional heterogeneous aquifer with uncertain hydraulic conductivity field; our goal is to estimate the latter, and the contaminant release history, from the measurements of hydraulic head and contaminant concentration collected in a few observation wells. These numerical experiments lead to the following conclusions.

1. The CAAE-DenseED-ESMDA inversion framework is capable of both identifying the contaminant release source and reconstructing a three-dimensional hydraulic conductivity field from sparse (in space and time) and noisy measurements of solute concentration and hydraulic head.
2. The CAAE-ESMDA inversion, with or without the DenseED CNN surrogate of the PDE-based forward model, yields estimates of the contaminant release strength that differ from the reference values by up to 30.42%. That can be attributed to the imperfect reconstruction of hydraulic conductivity field or relative insensitivity of the observed solute concentrations to the contaminant release strengths in each stress period (the inverse problem’s ill-posedness).
3. Deployment of the DenseED CNN surrogate within our CAAE-ESMDA inversion framework provides an order of magnitude speed up, while giving identical estimates of the hydraulic conductivity field; it also increases the predictive uncertainty (posterior standard deviation) relative to that obtained via the CAAE-ESMDA inversion with the PDE-based model.
4. The computational efficiency of CAAE-DenseED-ESMDA, relative to that of CAAE-ESMDA with the high-fidelity PDE model, is mostly due to the use of GPUs for CNN-related computations, while the PDE solver for the flow and transport model (e.g., MODFLOW and MT3DMS) utilizes CPUs.
5. Deployment of CAAE-DenseED-ESMDA allows one to investigate questions, such as measuring the data assimilation accuracy versus the ensemble size or designing a network of observation wells, that cannot be answered with CAAE-ESMDA with the PDE-based forward model, whose computational cost might be prohibitive.

Although the flow and transport simulators, MODFLOW and MT3DMS, can be parallelized to run on multiple CPU cores, that is a much more arduous task than carrying out NN-related computations on GPUs available in Google-Colab or other cloud computing environments. The latter takes very little implementation effort and can be done on a personal computer. The advantage of our method largely depends on the feasibility of accessing GPU computing resources versus deploying multicores parallelization with the physics-based forward model.

## Acknowledgements

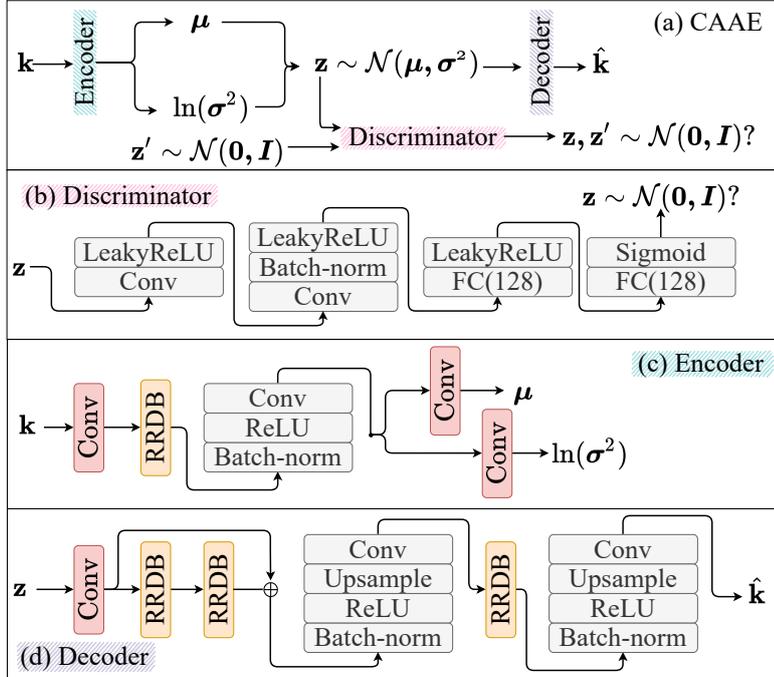
NZ acknowledges support from ARPA-E (award # DE-AR0001204).

537 **Appendix A**

538 The following sections discuss the details of the DNNs used in this study: CAAE  
 539 for the parameterization of the hydraulic conductivity field, and the DenseED surrogate  
 540 model predicting the groundwater flow and the contaminant transport.

541 **CAAE**

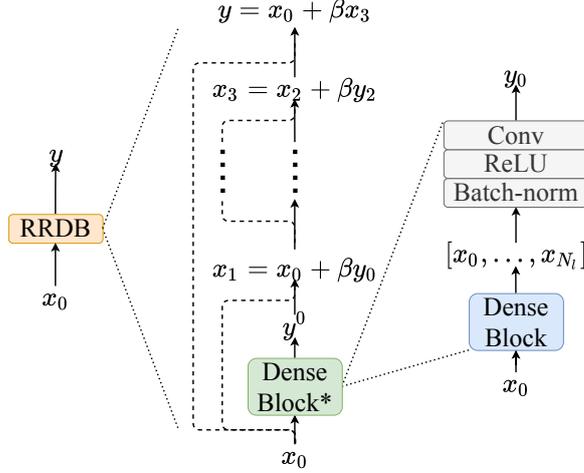
542 As briefly introduced in Section 3.2, the CAAE consists of three networks: an  
 543 encoder ( $\mathcal{G}$ ), a decoder ( $\mathcal{D}$ ), and a discriminator ( $\mathcal{D}$ ). The workflow of these three  
 544 networks and the architecture of each network are shown in Figure 11. The residual-  
 545 in-residual dense block (RRDB) used in the encoder and the decoder is illustrated  
 546 in Figure 12, in which the dense block is reused in the DenseED surrogate model as  
 547 well, and is shown in Figure 14. The dimensions of the internal layers of the encoder  
 548 and the decoder are shown in Table 6. Batch normalization (BN) (Ioffe & Szegedy,  
 549 2015), three dimensional convolutional operations (Conv) (I. Goodfellow et al., 2016),  
 550 Sigmoid, ReLU and LeakyReLU nonlinear activation functions (He et al., 2015) are  
 551 used in these neural networks; “FC(128)” denotes a fully-connected layer with the  
 552 output vector length being 128, “Upsample” layer doubles the size of the input feature  
 553 maps with the nearest upsampling method.



**Figure 11.** (a) CAAE, (b) Discriminator, (c) Encoder, and (d) Decoder. The CAAE (a) consists of an encoder (c) and a decoder (d), the discriminator (b) is trained as well to enforce the distribution of the low-dimensional latent variable  $\mathbf{z}$ . “RRDB” blocks are depicted in Figure 12, with the slope parameter in “LeakyReLU” being 0.2 in this study.  $\oplus$  denotes element-wise summation.

554 **DenseED**

555 The architecture of the DenseED in this study is shown in Figure 13. The  
 556 DenseED neural network structure consists of three main sub-structures: dense blocks,

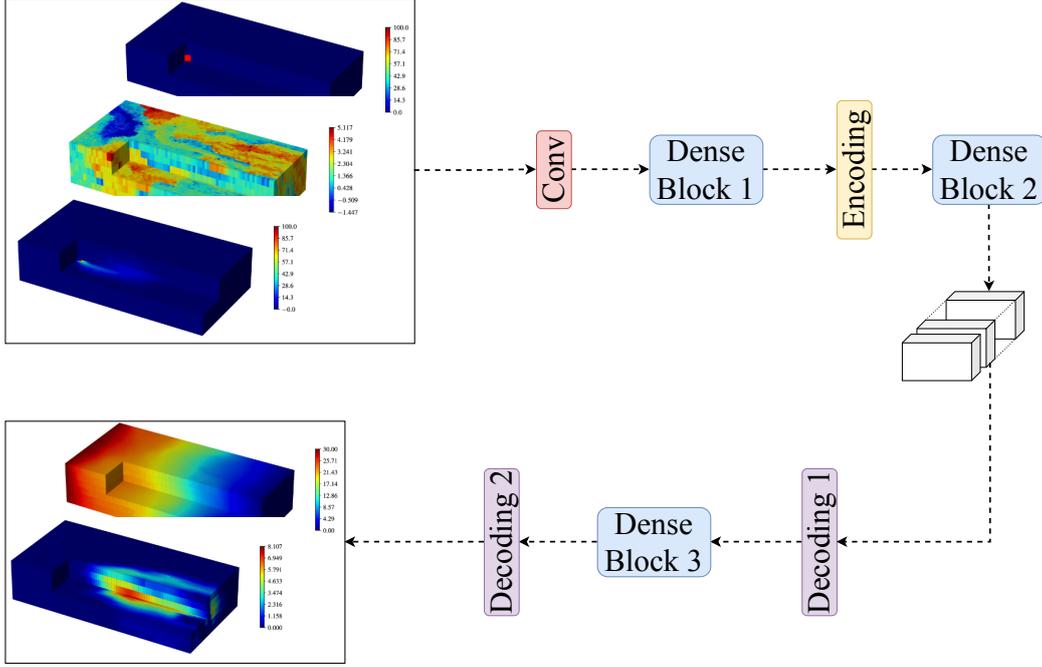


**Figure 12.** *RRDB block structure. Each RRDB block used in this study consists of  $N_{DB} = 3$  dense blocks, with the internal layer number of each dense block being  $N_l = 5$ . The dense block is illustrated in (a) in Figure 14.*

**Table 6.** *Dimension of the internal layer outputs of the encoder and decoder in CAAE. The encoder outputs  $\boldsymbol{\mu}$  and  $\ln \boldsymbol{\sigma}^2$ .  $N_{DB}$  denotes the number of dense blocks in a residual-in-residual block, shown in Figure 12.*

Encoder		
Layers	Number of features $C_f$	Resolution $W \times H \times D$
Input: $\mathbf{k}$	1	$81 \times 41 \times 6$
Conv	48	$41 \times 21 \times 3$
RRDB, $N_{DB} = 3$	48	$41 \times 21 \times 3$
BN-ReLU-Conv	48	$41 \times 21 \times 3$
Conv: $\boldsymbol{\mu}$	2	$21 \times 11 \times 2$
Conv: $\ln \boldsymbol{\sigma}^2$	2	$21 \times 11 \times 2$
Decoder		
Layers	Number of features $C_f$	Resolution $W \times H \times D$
Input: $\mathbf{z}$	2	$21 \times 11 \times 2$
Conv	48	$21 \times 11 \times 2$
RRDB, $N_{DB} = 3$	48	$21 \times 11 \times 2$
RRDB, $N_{DB} = 3$	48	$21 \times 11 \times 2$
BN-ReLU-UP-Conv	48	$41 \times 21 \times 4$
RRDB, $N_{DB} = 3$	48	$41 \times 21 \times 4$
BN-ReLU-UP-Conv: $\hat{\mathbf{k}}$	1	$81 \times 41 \times 6$

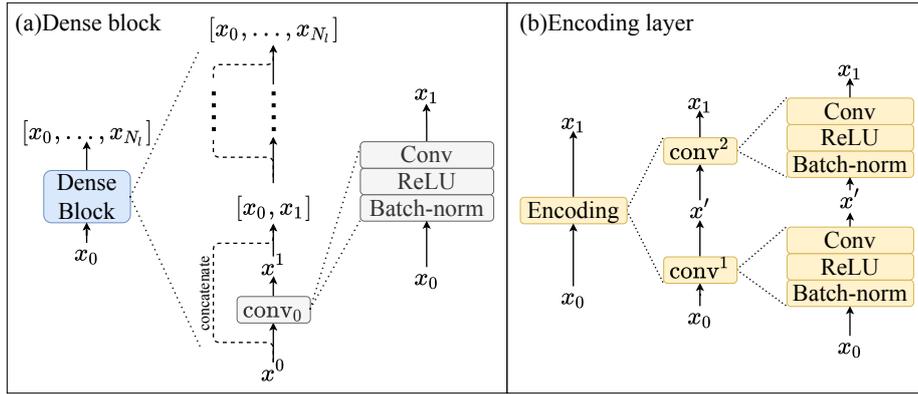
557 encoding layers, and decoding layers. The structure of a dense block is illustrated in  
 558 (a) in Figure 14. An encoding layer is shown in (b) in Figure 14, with which both  
 559 the feature number and the size of the features are halved:  $x^0 \in \mathbb{R}^{C \times W \times H \times D}$ ,  $x^1 \in$   
 560  $\mathbb{R}^{\frac{1}{2}C \times W \times H \times D}$ ,  $x^1 \in \mathbb{R}^{\frac{1}{2}C \times \frac{1}{2}W \times \frac{1}{2}H \times \frac{1}{2}D}$ . This figure can represent a decoding layer  
 561 as well, with the feature number halved, the size doubled:  $x^0 \in \mathbb{R}^{C \times W \times H \times D}$ ,  $x^1 \in$   
 562  $\mathbb{R}^{\frac{1}{2}C \times W \times H \times D}$ ,  $x^1 \in \mathbb{R}^{\frac{1}{2}C \times 2W \times 2H \times 2D}$ . In addition to these three main elements, the  
 563 size of the features are first halved with the very first Conv layer. The last decoding  
 564 layer maps the feature number to that of the output. The dimensions of the outputs  
 565 from each block are shown in Table 7.



**Figure 13.** Dense encoder decoder (DenseED) architecture with three dense blocks. The cubes as the output of “Dense Block 2” represent the encoded coarse high-level features.

**Table 7.** Dimension of the internal layer outputs of the DenseED network.  $N_l$  is the number of the internal layers in a dense block shown in Figure 14.

Layer	Number of features $C_f$	Resolution $W \times H \times D$
Input: $(c(\mathbf{x}, t), \mathbf{k}, S(\mathbf{x}, t))$	3	$81 \times 41 \times 6$
Conv	48	$41 \times 21 \times 3$
Dense Block 1, $N_l = 3$	192	$41 \times 21 \times 3$
Encoding	96	$21 \times 11 \times 2$
Dense Block 2, $N_l = 6$	384	$21 \times 11 \times 2$
Decoding 1	192	$41 \times 21 \times 3$
Dense Block 3, $N_l = 3$	336	$41 \times 21 \times 3$
Decoding 2: $(c(\mathbf{x}, t + \Delta t), h(\mathbf{x}))$	2	$81 \times 41 \times 6$



**Figure 14.** (a) A dense block with  $N_i$  internal layers. (b) Encoding layer structure in Figure 13. The feature number of  $x^0$  is halved first, then the size of the features are halved:  $x^0 \in \mathbb{R}^{C \times W \times H \times D}$ ,  $x' \in \mathbb{R}^{\frac{1}{2}C \times W \times H \times D}$ ,  $x^1 \in \mathbb{R}^{\frac{1}{2}C \times \frac{1}{2}W \times \frac{1}{2}H \times \frac{1}{2}D}$ .

566

## References

567

Aral, M. M., Guan, J., & Maslia, M. L. (2001). Identification of contaminant source location and release history in aquifers. *Journal of hydrologic engineering*, 6(3), 225–234.

568

569

Ayvaz, M. T. (2016). A hybrid simulation–optimization approach for solving the areal groundwater pollution source identification problems. *J. Hydrol.*, 538, 161–176.

570

571

Bakker, M., Post, V., Langevin, C. D., Hughes, J. D., White, J., Starn, J., & Fienen, M. N. (2016). Scripting modflow model development using python and flopy. *Groundwater*, 54(5), 733–739.

572

573

Barajas-Solano, D. A., Alexander, F. J., Anghel, M., & Tartakovsky, D. M. (2019). Efficient gHMC reconstruction of contaminant release history. *Front. Environ. Sci.*, 7, 149. doi: 10.3389/fenvs.2019.00149

574

575

Bedekar, V., Morway, E. D., Langevin, C. D., & Tonkin, M. J. (2016). *MT3D-USGS version 1: A US Geological Survey release of MT3DMS updated with new and expanded transport capabilities for use with MODFLOW* (Tech. Rep.). Reston, VA: US Geological Survey.

576

577

Boso, F., & Tartakovsky, D. M. (2020a). Data-informed method of distributions for hyperbolic conservation laws. *SIAM Journal on Scientific Computing*, 42(1), A559–A583.

578

579

Boso, F., & Tartakovsky, D. M. (2020b). Learning on dynamic statistical manifolds. *Proceedings of the Royal Society A*, 476(2239), 20200213.

580

581

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of compstat'2010* (pp. 177–186). Springer.

582

583

Canchumuni, S. W., Emerick, A. A., & Pacheco, M. A. C. (2019). History matching geological facies models based on ensemble smoother and deep generative models. *Journal of Petroleum Science and Engineering*, 177, 941–958.

584

585

Chaudhuri, A., Hendricks-Franssen, H.-J., & Sekhar, M. (2018). Iterative filter based estimation of fully 3D heterogeneous fields of permeability and Mualem-van Genuchten parameters. *Adv. Water Resour.*, 122, 340–354.

586

587

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

588

589

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geo-*

590

591

- 601 *physical Research: Oceans*, 99(C5), 10143–10162.
- 602 Evensen, G. (2003). The ensemble kalman filter: Theoretical formulation and practical  
603 implementation. *Ocean dynamics*, 53(4), 343–367.
- 604 Gamerman, D., & Lopes, H. F. (2006). *Markov chain monte carlo: stochastic simulation for bayesian inference*. Chapman and Hall/CRC.
- 605
- 606 Gao, G., & Reynolds, A. C. (2004). An improved implementation of the LBFGS  
607 algorithm for automatic history matching. In *Spe annual technical conference and exhibition*. doi: 10.2118/90058-MS  
608
- 609 Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
- 610 Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair,  
611 S., ... Bengio, Y. (2014). Generative adversarial networks. *arXiv preprint arXiv:1406.2661*.  
612
- 613 Haario, H., Laine, M., Mira, A., & Saksman, E. (2006). DRAM: efficient adaptive  
614 MCMC. *Stat. Comput.*, 16(4), 339-354.
- 615 Haario, H., Saksman, E., & Tamminen, J. (2001). An adaptive Metropolis algo-  
616 rithm. *Bernoulli*, 7(2), 223-242.
- 617 Harbaugh, A. W. (2005). *MODFLOW-2005, the US Geological Survey modular*  
618 *ground-water model: the ground-water flow process*. Reston, VA: US Depart-  
619 ment of the Interior, US Geological Survey.
- 620 He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpass-  
621 ing human-level performance on imagenet classification. In *Proceedings of the*  
622 *ieee international conference on computer vision* (pp. 1026–1034).
- 623 Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network  
624 training by reducing internal covariate shift. In *International conference on*  
625 *machine learning* (pp. 448–456).
- 626 Jiang, S., & Durlflosky, L. J. (2021). Data-space inversion using a recurrent au-  
627 toencoder for time-series parameterization. *Computational Geosciences*, 25(1),  
628 411–432.
- 629 Ju, L., Zhang, J., Meng, L., Wu, L., & Zeng, L. (2018). An adaptive gaussian  
630 process-based iterative ensemble smoother for data assimilation. *Advances in*  
631 *water resources*, 115, 125–135.
- 632 Kang, X., Kokkinaki, A., Kitanidis, P. K., Shi, X., Lee, J., Mo, S., & Wu, J. (2021).  
633 Hydrogeophysical characterization of nonstationary dnapl source zones by inte-  
634 grating a convolutional variational autoencoder and ensemble smoother. *Water*  
635 *Resources Research*, 57(2), e2020WR028538.
- 636 Kang, X., Kokkinaki, A., Kitanidis, P. K., Shi, X., Revil, A., Lee, J., ... Wu, J.  
637 (2020). Improved characterization of dnapl source zones via sequential hydro-  
638 geophysical inversion of hydraulic-head, self-potential and partitioning tracer  
639 data. *Water Resources Research*, 56(8), e2020WR027627.
- 640 Kim, S., Min, B., Kwon, S., & Chu, M.-g. (2019). History matching of a channelized  
641 reservoir using a serial denoising autoencoder integrated with es-mds. *Geoflu-*  
642 *ids, 2019*.
- 643 Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv*  
644 *preprint arXiv:1312.6114*.
- 645 Leichombam, S., & Bhattacharjya, R. K. (2018). New hybrid optimization method-  
646 ology to identify pollution sources considering the source locations and source  
647 flux as unknown. *J. Hazar. Tox. Radioact. Waste*, 23(1), 04018037.
- 648 Liggett, J. E., Partington, D., Frei, S., Werner, A. D., Simmons, C. T., & Fleck-  
649 enstein, J. H. (2015). An exploration of coupled surface–subsurface solute  
650 transport in a fully integrated catchment model. *Journal of Hydrology*, 529,  
651 969–979.
- 652 Linde, N., Renard, P., Mukerji, T., & Caers, J. (2015). Geological realism in hydro-  
653 geological and geophysical inverse modeling: A review. *Advances in Water Re-*  
654 *sources*, 86, 86–101.
- 655 Liu, Y., Sun, W., & Durlflosky, L. J. (2019). A deep-learning-based geological param-

- eterization for history matching complex models. *Mathematical Geosciences*, 51(6), 725–766.
- Makhzani, A., Shlens, J., Jaitly, N., & Goodfellow, I. (2016). Adversarial autoencoders. In *International Conference on Learning Representations (ICLR)*.
- Mariethoz, G., & Kelly, B. F. (2011). Modeling complex geological structures with elementary training images and transform-invariant distances. *Water Resources Research*, 47(7).
- Mo, S., Zabararas, N., Shi, X., & Wu, J. (2019a). Deep autoregressive neural networks for high-dimensional inverse problems in groundwater contaminant source identification. *Water Resources Research*, 55(5), 3856–3881.
- Mo, S., Zabararas, N., Shi, X., & Wu, J. (2019b). Integration of adversarial autoencoders with residual dense convolutional networks for inversion of solute transport in non-gaussian conductivity fields. *arXiv preprint arXiv:1906.11828*.
- Sarma, P., Durlofsky, L. J., & Aziz, K. (2008). Kernel principal component analysis for efficient, differentiable parameterization of multipoint geostatistics. *Mathematical Geosciences*, 40(1), 3–32.
- Snodgrass, M. F., & Kitanidis, P. K. (1997). A geostatistical approach to contaminant source identification. *Water Resources Research*, 33(4), 537–546.
- Tang, M., Liu, Y., & Durlofsky, L. J. (2021). Deep-learning-based surrogate flow modeling and geological parameterization for data assimilation in 3d subsurface flow. *Computer Methods in Applied Mechanics and Engineering*, 376, 113636.
- Tartakovsky, D. M., & Winter, C. L. (2008). Uncertain future of hydrogeology. *Journal of Hydrologic Engineering*, 13(1), 37–39.
- Van Leeuwen, P. J., & Evensen, G. (1996). Data assimilation and inverse methods in terms of a probabilistic formulation. *Monthly Weather Review*, 124(12), 2898–2913.
- Vo, H. X., & Durlofsky, L. J. (2014). A new differentiable parameterization based on principal component analysis for the low-dimensional representation of complex geological models. *Mathematical Geosciences*, 46(7), 775–813.
- Wen, G., Tang, M., & Benson, S. M. (2021). Towards a predictor for co2 plume migration using deep neural networks. *International Journal of Greenhouse Gas Control*, 105, 103223.
- White, R. E. (2015). Nonlinear least squares algorithm for identification of hazards. *Cogent Math.*, 2(1), 1118219.
- Winter, C. L., Guadagnini, A., Nychka, D., & Tartakovsky, D. M. (2006). Multivariate sensitivity analysis of saturated flow through simulated highly heterogeneous groundwater aquifers. *J. Comput. Phys.*, 217(1), 166–175.
- Winter, C. L., Tartakovsky, D. M., & Guadagnini, A. (2003). Moment equations for flow in highly heterogeneous porous media. *Surveys in Geophysics*, 24(1), 81–106.
- Wu, Z., Reynolds, A. C., & Oliver, D. S. (1999). Conditioning geostatistical models to two-phase production data. *SPE journal*, 4(02), 142–155.
- Xia, Y., & Zabararas, N. (2021). Bayesian multiscale deep generative model for the solution of high-dimensional inverse problems. *arXiv preprint arXiv:2102.03169*.
- Xu, T., & Gómez-Hernández, J. J. (2016). Joint identification of contaminant source location, initial release time, and initial solute concentration in an aquifer via ensemble Kalman filtering. *Water Resour. Res.*, 52(8), 6587–6595.
- Xu, T., & Gómez-Hernández, J. J. (2018). Simultaneous identification of a contaminant source and hydraulic conductivity via the restart normal-score ensemble Kalman filter. *Adv. Water Resour.*, 112, 106–123.
- Yang, H.-J., Boso, F., Tchelepi, H. A., & Tartakovsky, D. M. (2020). Method of distributions for quantification of geologic uncertainty in flow simulations. *Water Resources Research*, 56(7), e2020WR027643. doi: 10.1029/2020WR027643
- Yeh, H.-D., Chang, T.-H., & Lin, Y.-C. (2007). Groundwater contaminant source

- 711 identification by a hybrid heuristic approach. *Water Resources Research*,  
712 43(9).
- 713 Zhang, J., Lin, G., Li, W., Wu, L., & Zeng, L. (2018). An iterative local updating  
714 ensemble smoother for estimation and uncertainty assessment of hydrologic  
715 model parameters with multimodal distributions. *Water Resources Research*,  
716 54(3), 1716–1733.
- 717 Zhang, J., Zeng, L., Chen, C., Chen, D., & Wu, L. (2015). Efficient bayesian experi-  
718 mental design for contaminant source identification. *Water Resources Research*,  
719 51(1), 576–598.
- 720 Zheng, C., & Wang, P. P. (1999). *MT3DMS: a modular three-dimensional multi-*  
721 *species transport model for simulation of advection, dispersion, and chemical*  
722 *reactions of contaminants in groundwater systems; documentation and user's*  
723 *guide*. Environmental Laboratory (US).
- 724 Zhou, H., Gómez-Hernández, J. J., & Li, L. (2014). Inverse methods in hydrogeol-  
725 ogy: Evolution and recent trends. *Advances in Water Resources*, 63, 22–37.
- 726 Zhou, Z., & Tartakovsky, D. M. (2021). Markov chain monte carlo with neural net-  
727 work surrogates: Application to contaminant source identification. *Stochastic*  
728 *Environmental Research and Risk Assessment*, 35(3), 639–651.