

Water Resources Research®

RESEARCH ARTICLE

10.1029/2021WR031438

Deep Learning for Simultaneous Inference of Hydraulic and Transport Properties

Zitong Zhou¹, Nicholas Zabaraz² , and Daniel M. Tartakovsky¹ 

¹Department of Energy Science and Engineering, Stanford University, Stanford, CA, USA, ²Scientific Computing and Artificial Intelligence (SCAI) Laboratory, University of Notre Dame, South Bend, IN, USA

Key Points:

- Our deep-learning strategy reconstructs three-dimensional conductivity field and contaminant release history
- 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

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

D. M. Tartakovsky,
tartakovsky@stanford.edu

Citation:

Zhou, Z., Zabaraz, N., & Tartakovsky, D. M. (2022). Deep learning for simultaneous inference of hydraulic and transport properties. *Water Resources Research*, 58, e2021WR031438. <https://doi.org/10.1029/2021WR031438>

Received 20 OCT 2021

Accepted 14 SEP 2022

Abstract Identification of a heterogeneous conductivity field and reconstruction of a contaminant release history are key aspects of subsurface remediation. These two goals are achieved by combining model predictions with sparse and noisy hydraulic head and concentration measurements. Solution of this inverse problem is notoriously difficult due to, in part, high dimensionality of the parameter space and high computational cost of repeated forward solves. We use a convolutional adversarial autoencoder (CAAE) to parameterize a heterogeneous non-Gaussian conductivity field via a low-dimensional latent representation. A three-dimensional dense convolutional encoder-decoder (DenseED) network serves as a forward surrogate of the flow and transport model. The CAAE-DenseED surrogate is fed into the ensemble smoother with multiple data assimilation (ESMDA) algorithm to sample from the Bayesian posterior distribution of the unknown parameters, forming a CAAE-DenseED-ESMDA inversion framework. The resulting CAAE-DenseED-ESMDA inversion strategy is used to identify a three-dimensional contaminant source and conductivity field. A comparison of the inversion results from CAAE-ESMDA with physical flow and transport simulator and from CAAE-DenseED-ESMDA shows that the latter yields accurate reconstruction results at the fraction of the computational cost of the former.

Plain Language Summary Identification of a heterogeneous conductivity field and reconstruction of a contaminant release history are key aspects of subsurface remediation. These two goals are achieved by combining model predictions with sparse and noisy hydraulic head and concentration measurements. Solution of this inverse problem is notoriously difficult due to, in part, high dimensionality of the parameter space and high computational cost of repeated forward solves. We develop a deep-learning strategy to identify a three-dimensional contaminant source and conductivity field from sparse observations.

1. Introduction

Design of regulatory and remedial actions for contaminated soils and aquifers relies 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 computationally demanding 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 & Durlafsky, 2014) perform well for Gaussian random fields, but require ad-hoc modifications for non-Gaussian fields (Liu et al., 2019). Deep neural network (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) (Goodfellow et al., 2014) and variational autoencoders (Kingma & Welling, 2013). Both produce a DNN that learns a two-way mapping between the conductivity field and a low-dimensional embedding. Random realizations from the latent variable distribution can be decoded to a conductivity field that is statistically similar to those drawn from the training data set. A series of studies involving usage of variational autoencoders or GANs in Laloy et al. (2017, 2018, 2019) and Lopez-Alvis et al. (2021) have shown their superiority over PCA for inversion tasks in geophysics, specifically in geophysical formations exhibiting channel patterns. In addition to the large reduction in the number of unknown parameters, such parameterizations make it feasible to tackle the latent variable distribution, which is typically a standard normal by construction. This simplicity, in turn, facilitates the solution of an inverse problem with ensemble methods discussed below.

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

Two complementary strategies can be deployed to alleviate this cost. The first aims to reduce the number of forward simulations needed for an inversion algorithm to converge. The second seeks to reduce the computational cost of each forward solve. In pursuit of the first aim, the design and implementation of efficient MCMC variants remain an active research area. The most popular variant of non-gradient-based MCMC is, arguably, the delayed rejection adaptive Metropolis (DRAM) sampling (Haario et al., 2001, 2006), which slightly outperforms the random walk Metropolis-Hastings MCMC in terms of efficiency (Zhang et al., 2015; Z. Zhou & Tartakovsky, 2021; Xia & Zabarar, 2021). Gradient-based MCMC methods, such as hybrid Monte Carlo (HMC) sampling (Barajas-Solano et al., 2019), converge faster than these and other MCMC variants. However, computation of the gradient of a Hamiltonian dynamical system is prohibitive for high-dimensional transport problems. Learning on statistical manifolds (Boso & Tartakovsky, 2020a, 2020b, 2022) provides another possible solution. Regardless of their implementation, MCMC methods are computationally expensive and ill-suited for parallelization (Ghorbanidehno et al., 2020) because of the requirement for a Markov chain to reach its equilibrium state. Ensemble-based inversion methods are generally faster since they allow nearly perfect parallelization, because of the independence of samples in the ensemble. Variants of Kalman filters, such as iterative Ensemble Kalman filter, have been used for estimation of three-dimensional heterogeneous permeability fields (Chaudhuri et al., 2018). In principle, any implementation of Kalman filter can be used to solve inverse problems tackled in our study. A relatively new variant of Kalman filter, ensemble smoother with multiple data assimilation (ESMDA) (Emerick & Reynolds, 2013), has gained popularity in subsurface applications (Jiang & Durlafsky, 2021; Kim et al., 2019; Tang et al., 2021). Originally developed as a decision making tool in the fields of energy efficiency and petroleum engineering, Bayesian evidential learning has evolved rapidly and shown its potential in other fields (Hermans et al., 2018, 2019; Michel et al., 2020). Subsequent efforts adapted this method to the tasks of parameter estimation and optimal design (Park & Caers, 2020; Thibaut et al., 2021). This past experience suggests that ESMDA is an efficient tool for solving high-dimensional inverse problems with sparse and noisy observations.

In terms of inversion complexity, we subdivide recent groundwater-related studies into three categories: estimation of hydraulic conductivity from measurements of hydraulic head and, optionally, of solute concentration (Ju et al., 2018; Mo et al., 2019b); estimation of contaminant release history from concentration measurements, for known flow and transport parameters (Zhou & Tartakovsky, 2021; Zhang et al., 2015); and estimation of both

contaminant release history and hydraulic conductivity from hydraulic head and solute concentration data in two- (Kang et al., 2021; Mo et al., 2019a; Xu & Gómez-Hernández, 2018) and three-dimensional (Kang et al., 2020) aquifers. We briefly discuss the latter category to highlight the novelty of our approach.

A low-dimensional representation of the random log-normal conductivity obtained via the Karhunen-Loève expansion (Mo et al., 2019a) loses its attractiveness if the subsurface environment is highly heterogeneous, exhibiting short correlation lengths and multimodal statistics. Our work is among the few recent source identification studies dealing with nonlinear transport problems and nonstationary (statistically inhomogeneous) conductivity fields. The deep learning-based strategies of ensemble inversion were adopted by Xu and Gómez-Hernández (2018) and Kang et al. (2021) to estimate both a non-Gaussian conductivity field and contamination source, yet they deal only with two-dimensional problems, and their accuracy is relatively low. In the adjacent field of petroleum engineering, convolutional DNN (CNN) post-processing of PCA (CNN-PCA) parameterization and ESMDA were used to estimate both a channelized permeability and oil/water rate (Tang et al., 2021). However, this application deals with an observable quantity (oil/water production rate), while ours has to contend with an unobservable one (the location and strength of a contaminant release).

The shortcomings of the joint inversion strategies mentioned above fall into several groups: inaccurate representation of non-Gaussian or channelized fields; questionable applicability to high-dimensional inverse problems; and limited generalizability of inverse problem solutions to scenarios not seen in training. To alleviate these shortcomings, we use a convolutional adversarial autoencoder (CAAE) to parameterize a non-Gaussian conductivity field (Mo et al., 2019b), train a surrogate dense encoder-decoder DNN to replace the PDE-based model of subsurface flow and transport, and apply the ESMDA inversion framework to identify the spatiotemporally extended source of contamination and the latent variables representing the conductivity field. We posit that combination of these three components, which yields the method we refer to as CAAE-DenseED-ESMDA, provides a fast and robust inversion solution. Although advantages of each component were established in various disciplines, their synergy has remained unexplored. Our study demonstrates that CAAE-DenseED-ESMDA is a powerful tool for dynamic source identification and conductivity reconstruction in three spatial dimensions, when the number of unknown parameters is large ($\sim 20,000$). We are not aware of other machine learning strategies used to tackle inverse groundwater problems of similar complexity.

In Section 2, we formulate the problem of joint reconstruction of hydraulic conductivity field and contaminant release history from sparse and noisy measurements of hydraulic head and solute concentration. Our inversion strategy, combining CAAE parameterization of the conductivity field (Section 3.2), a CNN surrogate of the flow and transport model (Section 3.3), and the ESMDA inversion method (Section 3.1), is described in Section 3. Results of our numerical experiments are reported in Section 4; they demonstrate that our method is about 8 times faster than CAAE-ESMDA with the PDE-based flow and transport model. Main conclusions drawn from this study are summarized in Section 6. Details of the neural network architectures are given in Supporting Information S1.

2. Problem Formulation

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

2.1. Contaminant Transport Model

We consider transport of a reactive solute in a three-dimensional steady-state groundwater flow field. The latter is described by the partial differential equation (PDE)

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

where $K(\mathbf{x})$ is the hydraulic conductivity of the aquifer Ω , and $h(\mathbf{x})$ is the hydraulic head. This PDE is subject to appropriate boundary conditions on the simulation domain boundary $\partial\Omega$. After the flow equation is solved, the average pore velocity $\mathbf{u}(\mathbf{x}) = (u_1, u_2, u_3)^T$ is computed from Darcy's law,

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

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

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

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

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

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

where D_m is the coefficient of molecular diffusion for the contaminant in free water; α_L is the longitudinal dispersivity; and α_T and α_C are transverse dispersivities in the x_2 and x_3 directions, respectively.

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

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

where ρ_b is the bulk density and \tilde{c} is the concentration sorbed.

We assume the system to be in local chemical equilibrium, that is, sorption to be much faster than advection and dispersion. We also assume that sorption does not affect the porosity θ , 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:

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

Table 1
Physical Quantities in the Transport Model (6) and Their Units

Physical quantity	Units	Type
Solute concentration, c	ML^{-3}	Simulated
Porosity, θ	–	Known
Distance along the i th Cartesian coordinate axis, x_i	L	–
Hydrodynamic dispersion coefficient tensor, D_{ij}	L^2T^{-1}	Simulated
Pore water velocity, u_i	LT^{-1}	Simulated
Volumetric flow rate, q_s , sources (+) and sinks (–)	T^{-1}	Known
Concentration of source or sink flux, c_s	ML^{-3}	Estimated
Chemical reaction term, R_n	$\text{ML}^{-3}\text{T}^{-1}$	Simulated
Bulk density, ρ_b	ML^{-3}	Known
Sorbed concentration, \tilde{c}	ML^{-3}	Simulated
Freundlich constant, K_f	$(\text{L}^3\text{M}^{-1})^a$	Known
Freundlich exponent, a	–	Known

Note. The last column specifies whether the quantity is known from experiments, simulated for given conductivity field and source terms, or estimated from unspecified information.

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 Freundlich isotherm,

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

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

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, and so on, are assumed to be known. The contaminant release is temporally discretized into N_{re} intervals, with a constant release strength during each time interval. Identification of the source term $q_s c_s$ is tantamount to finding the location(s), \mathbf{S}_j , 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 ϵ_{mi}^c and ϵ_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 ϵ_{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; δ_{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 σ_c is the standard deviation of the measurement errors, and \mathbf{I} is the $(M \times M)$ identity matrix. The hydraulic head measurement errors ϵ_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 σ_h is the standard deviation of the measurement errors.

The error model (9) assumes the flow and transport models (1)–(8) to be exact and the measurement 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 (Bedekar et al., 2016). The latter employs a standard finite-difference method with upstream or central-in-space weighting. 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: ESMDA, 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

Upon 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 ($\mathbf{S}_p, \mathbf{S}_s$) and hydraulic conductivity $K(\mathbf{x})$ in all discretized cells (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 ϵ_{mi}^c and ϵ_m^h into a vector $\boldsymbol{\epsilon}$ of the same length. Then, the error model (9) takes the vector form,

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

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 probability density function (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} , that is, 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 models are then linearly 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}_{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 α_i $i = 1, \dots, N_a$. The requirement $\sum_{i=1}^{N_a} \alpha_i = 1$ guarantees consistency with ES in the linear Gaussian case; it acts as a constraint for the ESMDA method in general. 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) $g(\mathbf{m}_j^i)$.
 2. Perturb the measurements with inflated measurement noise: $\mathbf{d}_{uc,j}^i \sim \mathcal{N}(\mathbf{d}, \alpha_i \mathbf{C}_{\text{D}})$.
 3. Compute the cross covariance matrix \mathbf{C}_{MD}^i and the auto-covariance matrix of the predicted data \mathbf{C}_{DD}^i .
 4. Update the ensemble as in 12, but with \mathbf{C}_{D} replaced by $\alpha_i \mathbf{C}_{\text{D}}$:

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

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

3.2. CAAE Parameterization of Conductivity Field

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

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

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

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

Table 2
Alternative Input-Output Frameworks for Construction of a Surrogate Model

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

Note. 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 $\mathbf{S}(\mathbf{x}, t)$.

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

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

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:

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

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

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

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

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

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

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

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

3.3. DenseED Neural Networks as Forward Model Surrogates

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

Several approaches to constructing an input-output surrogate are collated in Table 2. We choose an autoregressive model over a one-to-many model based on computer-memory considerations: for three-dimensional problems with I time steps, memory allocated for input and output can be prohibitively large; also, the autoregressive scheme reduces the number of DNN parameters needed for the regression task. The autoregressive structure enables us to predict the full images (image-to-image) at each time step. That strategy has a superior generalizability than its image-to-sensors counterparts, which predict concentration values only at sparse locations where measurements are collected (Zhou & Tartakovsky, 2021).

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 $\mathbf{S}(\mathbf{x}, t) \in \mathbb{R}^{W \times H \times D}$.

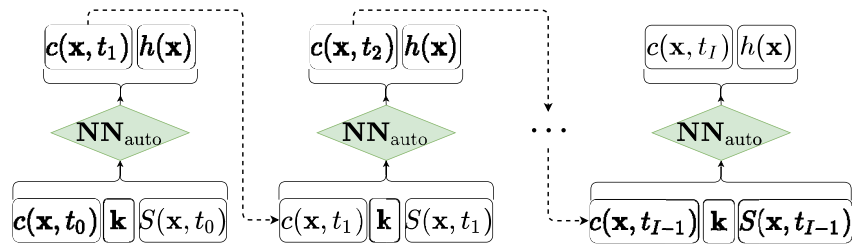


Figure 1. Autoregressive surrogate NN_{auto} of the PDE-based flow and transport model (1)–(8). The three input channels correspond to $c(\mathbf{x}, t)$, $S(\mathbf{x}, t)$, and $\ln K(\mathbf{x})$. The two output channels correspond to $c(\mathbf{x}, t + \Delta t)$ and $h(\mathbf{x})$. The concentration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

An autoregressive surrogate NN_{auto} replaces the PDE-based model:

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

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

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

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

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

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

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

One could improve the accuracy of our surrogate model by deploying an iterative optimization strategy, which would increase its computational cost. We chose not to do so because the convolutional encoder-decoder neural networks proved to be sufficiently accurate surrogates of contaminant transport models when used for inverse problems (Mo et al., 2019a; Z. Zhou & Tartakovsky, 2021). Mo et al. (2019b) provide a detailed analysis of the number of samples needed to train a three-dimensional surrogate of a problem whose spatial discretization ($6 \times 32 \times 64$) is similar to ours. In the spirit of transfer learning, we adopted their most efficient setting of the surrogate model. This is an empirical choice based on our past studies, and we encourage a thorough study on the performance of a surrogate model in other unexplored applications.

3.4. CAAE–DenseED–ESMDA Inversion Framework

We combine the CAAE parameterization of the conductivity field with the DenseED CNN surrogate of the forward model to obtain fast and accurate predictions of concentration $c(\mathbf{x}, t)$ and $h(\mathbf{x})$ for a given set of inputs. The coefficient of determination R^2 serves as a measure of accuracy of the forward surrogate. Then, we utilize ESMDA to identify the unknown parameters, including the conductivity field and the source terms ($\mathbf{S}_p, \mathbf{S}_s$). We use the box-plots of the ensembles for the assimilated terms to illustrate the quality and uncertainty of the inversion. The CAAE parameterization enables one to estimate the discretized log-conductivity field \mathbf{k} through the latent variable \mathbf{z} . Our CAAE–DenseED–ESMDA inversion framework is implemented in the following algorithm (Figure 2):

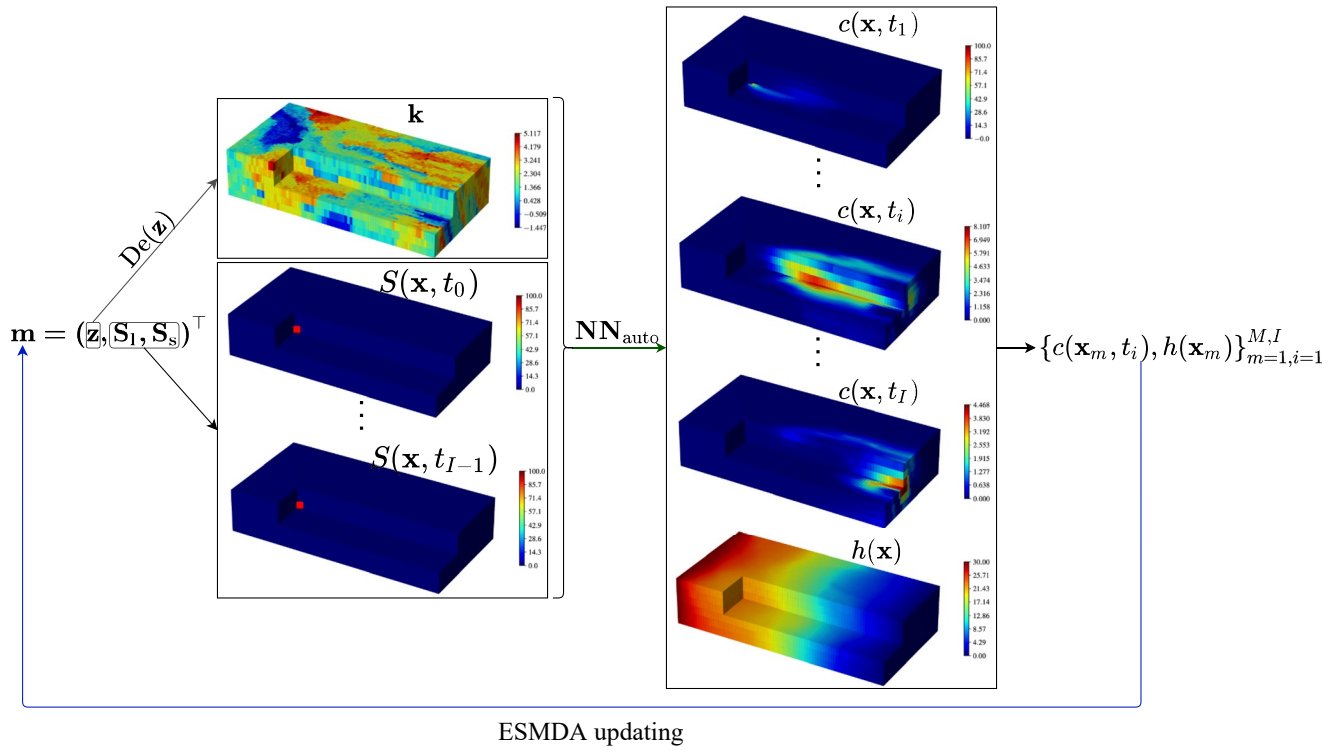


Figure 2. Schematic illustration of the CAAE-DenseED-ESMDA algorithm. $\text{De}(\cdot)$ denotes the decoder that obtains the log-conductivity field from the latent variable \mathbf{z} , NN_{auto} represents the autoregressive surrogate model which predicts the concentration and hydraulic head fields. 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} .

1. Train a CAAE; obtain the decoder $\text{De}(\cdot)$ that maps the low-dimensional latent variable \mathbf{z} back onto the log-conductivity field \mathbf{k} .
2. Train an autoregressive DenseED CNN NN_{auto} to predict $c(\mathbf{x}, t)$ and $h(\mathbf{x})$ for the input conductivity field and contaminant release history.
3. Generate the initial input ensemble \mathbf{M}^f of size N_e , whose elements \mathbf{m}_j^1 ($j = 1, \dots, N_e$) 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 variable for the log conductivity field; and \mathbf{S}_{1j}^1 and \mathbf{S}_{sj}^1 denote respectively the source location and strength in all release periods, drawn from an appropriate prior distribution.
4. Perform the ESMDA inversion with N_d data assimilation iterations and the inflation coefficients α_i ($i = 1, \dots, N_d$). For $i = 1, \dots, N_d$,
 - (a) Obtain the log-conductivity realizations $\mathbf{k}_j^i = \text{De}(\mathbf{z}_j^i)$ with $j = 1, \dots, N_e$;
 - (b) Form the release configuration $\{\mathbf{S}_{1j}^i, \mathbf{S}_{sj}^i\}$ into the input matrix \mathbf{S}_j^i , and predict $c(\mathbf{x}, t)$ and $h(\mathbf{x})$ at the measurement times and locations, $\text{NN}_{\text{auto}}(\mathbf{m}_j^i)$ for all j ;
 - (c) Update the ensemble \mathbf{m}_j^i via ESMDA with α_i to obtain \mathbf{m}_j^{i+1} .
5. The end result, $\mathbf{m}_j^{N_d+1}$, serves as the final ensemble from which PDFs of the log conductivity field and the contaminant release parameters are estimated.

4. Numerical Experiments

4.1. Experimental Setup

A confined heterogeneous aquifer is described as a rectangular cuboid Ω of size 2,500 m \times 1250 m \times 300 m; it is discretized with a mesh consisting of 81 \times 41 \times 6 cells. Groundwater flow is driven by constant heads $h_L = 30$ m and $h_R = 0$ m imposed along the left ($x_1 = 0$) and right ($x_1 = 2,500$ m) facets of the cuboid, respectively; the remaining boundaries are impermeable to flow. The hydraulic conductivity of this aquifer, $K(\mathbf{x})$, is unknown (except when generating the ground truth); equiprobable realizations of $Y(\mathbf{x}) = \ln K(\mathbf{x})$ are generated by extracting

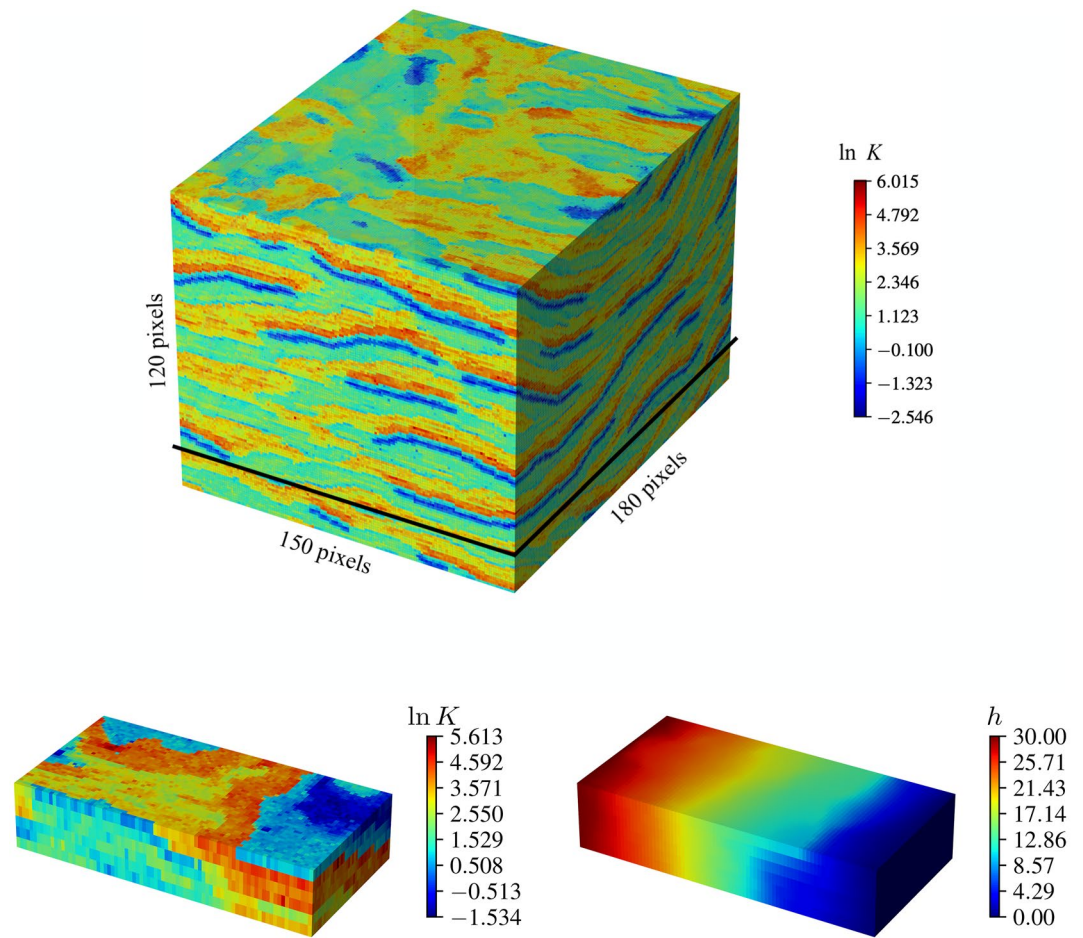


Figure 3. **Top:** Training image, consisting of $150 \times 180 \times 120$ pixels. Equiprobable realizations of log-conductivity $Y(\mathbf{x}) = \ln K(\mathbf{x})$ are generated by randomly selecting patches of size $81 \times 41 \times 6$ pixels. The top $150 \times 180 \times 105$ pixels serve as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

$81 \times 41 \times 6$ patches from the $150 \text{ px} \times 180 \text{ px} \times 120 \text{ px}$ training image (Mariethoz & Kelly, 2011) in Figure 3, available at <https://github.com/GAIA-UNIL/trainingimages>. One such cropped log-conductivity field $Y(\mathbf{x})$ and the corresponding hydraulic head $h(\mathbf{x})$, obtained as a solution of the groundwater flow Equation 1, are shown in Figure 3. These fields serve as the ground truth.

We assume to be known, and representative of a generic sandy alluvial aquifer in Southern California (Liggett et al., 2015), the following transport parameters: porosity $\theta = 0.3$, bulk density $\rho_b = 1,587 \text{ kg/m}^3$, molecular diffusion coefficient $D_m = 10^{-9} \text{ m}^2/\text{d}$, longitudinal dispersivity $\alpha_L = 35 \text{ m}$, transverse dispersivities $\alpha_T = \alpha_C = 0.3\alpha_L$, and parameters $a = 0.9$ and $K_f = 0.1 \text{ (m}^3/\text{g)}^a$ of the Freundlich isotherm. The contaminant enters the aquifer via a point source, whose depth is known (the fourth layer from the top of the domain) but the location in the horizontal plane (S_1^x and S_1^y) is uncertain. The contaminant release is known to occur during a 20-year period, but its strength is uncertain. Following the standard practice in groundwater modeling, we divide this time interval into $N_{re} = 5$ sub-intervals (“stress periods” in the MODFLOW/MT3D language) during each of which the release strength (S_s) is constant. In this configuration, the unknown contaminant release history is represented by the 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 values of \mathbf{S} used to generate the ground-truth concentrations are reported in Table 3. Combined with the discretized version \mathbf{k} of the uncertain log-conductivity field $Y(\mathbf{x})$, this yields 19,933 unknowns to be determined from the measurements of solute concentration $c(\mathbf{x}, t)$ and hydraulic head $h(\mathbf{x})$. Expert knowledge about possible location and strength of the contaminant release is encapsulated in the uniform (“uninformative”) prior distributions for \mathbf{S}_1 and \mathbf{S}_s , which are shown in Table 3.

Table 3

Parameters $S_1 = (S_1^x, S_1^y)^T$ and $S_s = (S_{s,1}, S_{s,2}, S_{s,3}, S_{s,4}, S_{s,5})^T$ Specifying, Respectively, the Location and Strength of a Contaminant Release

	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, 1,000]	[50, 1,000]	[50, 1,000]	[50, 1,000]	[50, 1,000]

Note. “Truth” refers to their (unknown) reference values used to generate ground truth and concentration measurements, and “Prior” to the intervals on which their uniform priors, $\mathcal{U}[\cdot, \cdot]$, are defined. The values of S_1 are in m, and of S_s in g/m^3 .

Measurements of hydraulic head and contaminant concentration used for data assimilation are collected at observation wells, whose completion allows one to collect water samples in each of the model’s six vertical layers. We consider the observation wells whose locations are depicted in Figure 4. During the simulated time horizon of 40 years, the contaminant concentration is sampled at $I = 10$ time intervals of 4 years each, and the hydraulic head is measured once since the flow is at steady-state. The data at all space-time locations are generated by adding zero-mean Gaussian measurement error with standard deviations $\sigma_c = \sigma_h = 0.5$, to the solution $\mathbf{g}(\mathbf{m})$ of the flow and transport model (1)–(8) with the input parameter values identified as “ground truth” above, corresponding to 2.5% of the maximum value of the concentration ($\sim 20 \text{ g}/\text{m}^3$), and $\sim 1.7\%$ of the maximum value of the hydraulic head ($\sim 30 \text{ m}$) on all sensor locations. We assimilate the data only once for each realization, such that all measurements at all I times are gathered in a single data set, which is then used to update the unknown parameters.

4.2. CAAE Training for Conductivity Parameterization

We train a CAAE DNN to parameterize the discretized log-conductivity field $\mathbf{k} \in \mathbb{R}^{81 \times 41 \times 6}$. The end goal is an encoder $\mathcal{G}(\mathbf{k})$ that maps an input field \mathbf{k} onto a low-dimensional latent variable $\mathbf{z} \in \mathbb{R}^{2 \times 2 \times 11 \times 21}$ with standard-Gaussian prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and a decoder $\text{De}(\cdot)$ that reconstructs \mathbf{k} from this latent variable. The training is done on $N = 23,000$ realizations of \mathbf{k} , obtained as randomly selected $(81 \times 41 \times 6)$ patches from the top 150

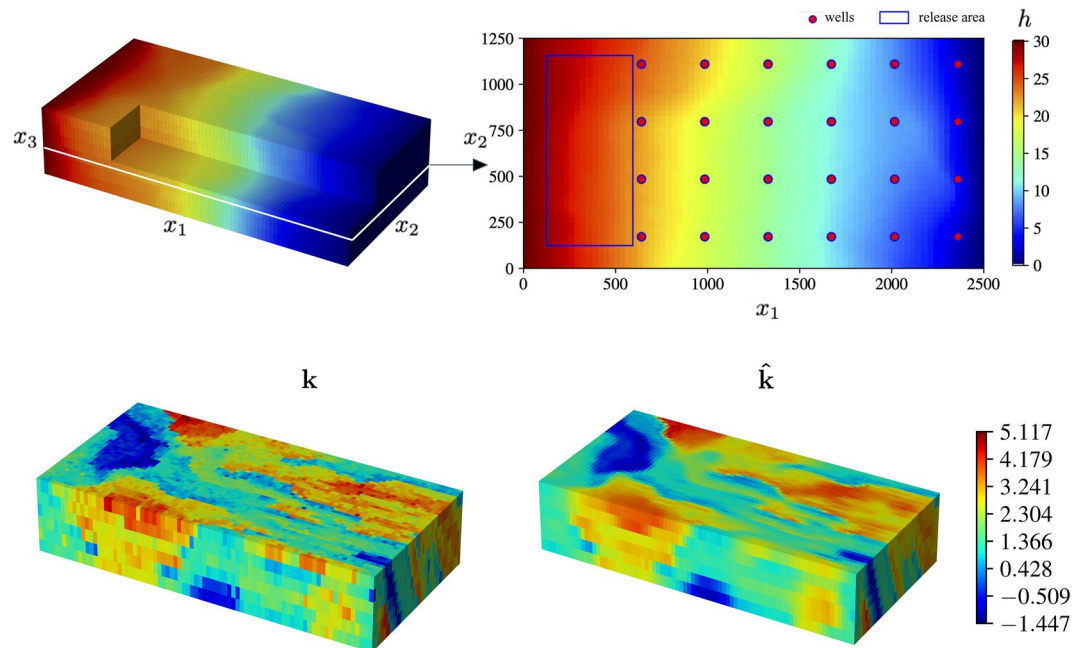


Figure 4. Top: Observational wells (red dots) in which measurements of hydraulic head h and solute concentration c are collected. The well locations are superposed on the ground-truth distribution of hydraulic head in the fourth layer of the MODFLOW model. The blue box represents a region of possible 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. Bottom: 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, $\hat{\mathbf{k}} = \text{De}(\mathbf{z})$.

px × 180 px × 105 px part of the large training image in Figure 3. Additional 2200 images cropped from the bottom 150 px × 180 px × 15 px part of the training image serve as the testing set. The latent variable \mathbf{z} has $2 \cdot 2 \cdot 11 \cdot 21 = 924$ elements. (These and other constants and matrix dimensions are collated in Table S1 in Supporting Information S1.) With 50-epochs training and the learning rate of $2 \cdot 10^{-4}$, the Adam optimizer is used to obtain the DNN parameters and, thus, build $\mathcal{G}(\mathbf{k})$ and $\text{De}(\mathbf{z})$. The details of the CAAE architecture are explained in the SI, with the dimensions of the internal layer outputs provided in Table S2 in Supporting Information S1.

A representative realization of $\ln K(\mathbf{x}) \rightarrow \mathbf{k}$ from the test set and its reconstruction via the decoder, $\hat{\mathbf{k}} = \text{De}(\mathbf{z})$, are shown in Figure 4. After the training is complete, the mean absolute error $\|\mathbf{k} - \hat{\mathbf{k}}\|_1$, averaged over all the elements of the numerical mesh and over the 2,200 members of the testing data set, is 0.2637. The reconstructed log-conductivity field $\hat{\mathbf{k}}$ captures the main structural features of its original counterpart \mathbf{k} . Some loss of information is unavoidable in reduced-order modeling but, overall, the performance of this autoencoder is adequate to achieve accurate inverse modeling results, as we show in Section 4.4. In fact, high accuracy of this autoencoder might have a negative impact on the inversion results (Lopez-Alvis et al., 2021), which also provides an analysis on the trade off between the inversion quality and the generative autoencoder accuracy.

In addition to the mean absolute error of CAAE, we also run the PDE-based forward model with the reconstructed $\hat{\mathbf{k}}$, and computed the coefficient of determination R^2 for the concentration and hydraulic head fields. This essential experiment indicates the impact of the error in the representation of the heterogeneity on the prediction of the concentration field. The steps for this test are as follows.

- Obtain N_{test} forward model inputs: $\mathbf{m}_i = (\mathbf{k}_i, \mathbf{S}_{li}, \mathbf{S}_{si})^T$, $i = 1, \dots, N_{\text{test}}$, where \mathbf{k}_i is the conductivity field sampled from the testing set in Figure 3.
- Obtain the concentration and hydraulic head predictions with the PDE-based forward model \mathbf{g} : $\mathbf{y}_i = \mathbf{g}(\mathbf{m}_i)$, $i = 1, \dots, N_{\text{test}}$.
- Obtain the reconstructed conductivity fields $\{\hat{\mathbf{k}}_i\}_{i=1}^{N_{\text{test}}}$ by applying the encoder and decoder obtained from the CAAE training: $\hat{\mathbf{k}}_i = \text{De}(\mathcal{G}(\mathbf{k}_i))$.
- Substitute the conductivity field with the reconstructed ones $\hat{\mathbf{k}}$, and run the PDE-based forward model again for the prediction: $\hat{\mathbf{y}}_i = \mathbf{g}(\hat{\mathbf{m}}_i)$, $\hat{\mathbf{m}}_i = (\hat{\mathbf{k}}_i, \mathbf{S}_{li}, \mathbf{S}_{si})^T$.
- Compute the coefficient of determination,

$$R^2 = 1 - \frac{\sum_i^{N_{\text{test}}} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|^2}{\sum_i^{N_{\text{test}}} \|\mathbf{y}_i - \bar{\mathbf{y}}\|^2}, \quad \bar{\mathbf{y}} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \mathbf{y}_i.$$

The evaluated R^2 on the 150 set of forward simulations is 0.9623.

4.3. DenseED Surrogate Model

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 950×10 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 Supporting Information S1, and the dimensions of the internal layers outputs are listed there in Table S2 in Supporting Information S1.

Figure 5 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

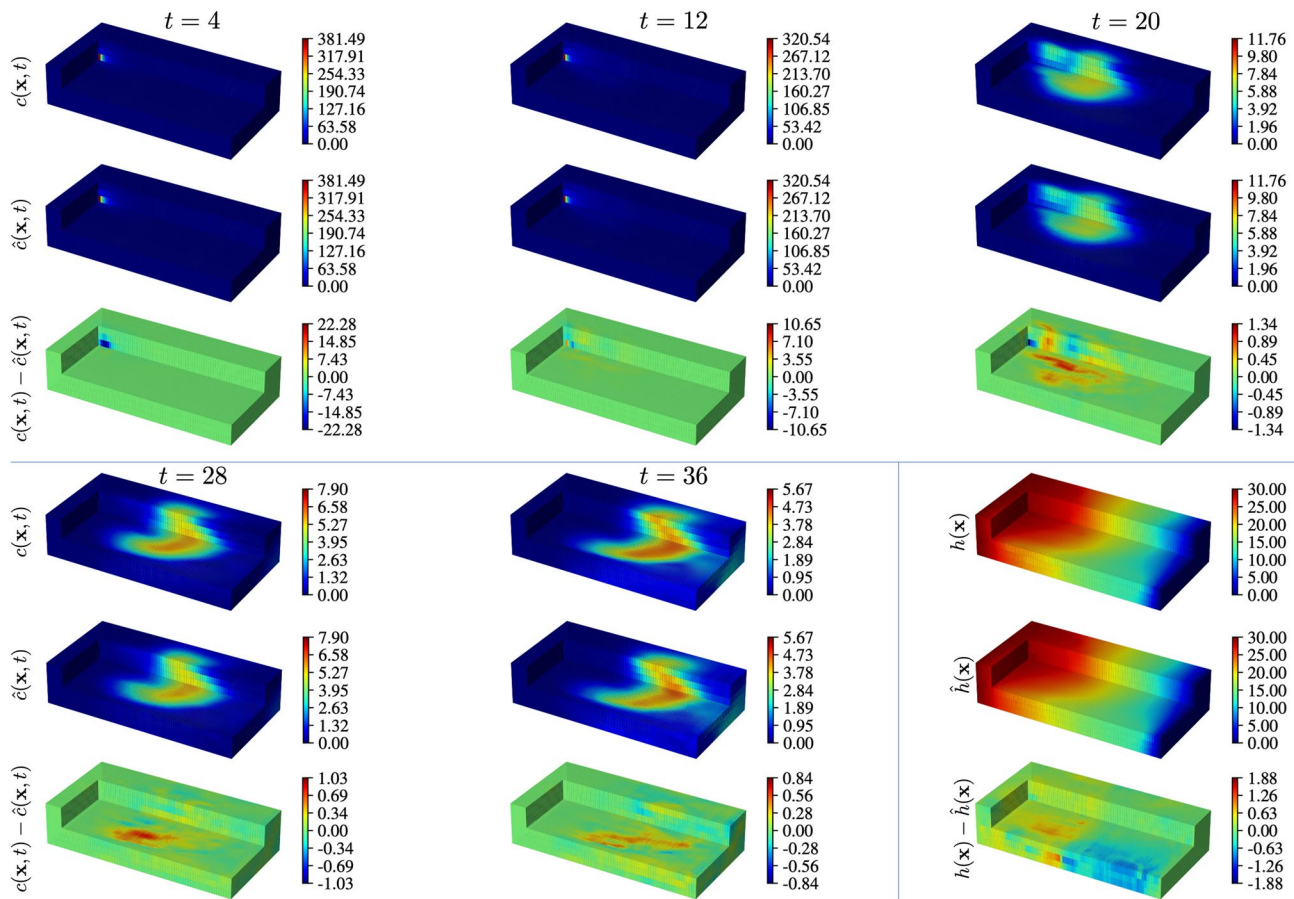


Figure 5. Predictions of the solute concentration obtained with the PDE-based model, $c(\mathbf{x}, t)$, and its DenseED CNN surrogate, $\hat{c}(\mathbf{x}, t)$, times $t = 4, 12, 20, 28, 36, t$ in year. Also shown are the corresponding predictions of the hydraulic head, $h(\mathbf{x})$ and $\hat{h}(\mathbf{x})$; 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.

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. The coefficient of determination is $R^2 = 0.79$. 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 ESM DA inversion framework. One could achieve more accurate predictions for three-dimensional problems by either deploying a more complex DNN architecture (Mo et al., 2019b; Wen et al., 2021) or using much larger N_{NMC} or both. However, similar to the CAAE training, we focus on the development of efficient methodologies for three-dimensional inverse modeling that accommodate the trade-off between the accuracy and computational feasibility.

4.4. ESM DA Inversion

We demonstrate the use of the CAAE parameterization and the DenseED CNN surrogate of the PDE-based forward model to accelerate the ESM DA inversion. The combination of these three techniques constitutes our CAAE-DenseED-ESM DA 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 ESM DA with 10 iterations. To ascertain the impact of the DenseED CNN surrogate on the inversion accuracy, we also run CAAE-ESM DA with the PDE-based forward model implemented in MODFLOW and MT3DMS. The ensemble size for ESM DA for both CAAE-ESM DA and CAAE-DenseED-ESM DA are set to $N_e = 960$.

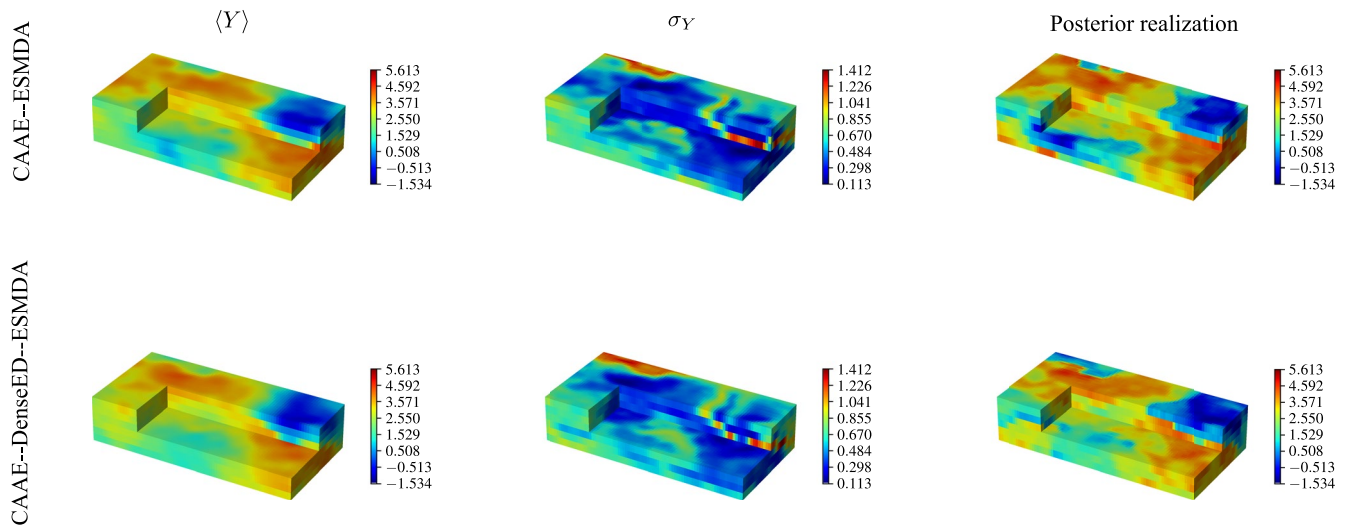


Figure 6. Posterior mean ($\langle Y \rangle$, left column) and standard deviation (σ_Y , middle column) of the log-conductivity field $Y(\mathbf{x})$ obtained upon assimilation of concentration and head measurements from the dense observation network. These statistics are obtained via our inversion 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 from the resulting posterior ensemble (right column).

The measurements are taken at 24 wells that are completed in all six 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 $N_d = 144 \cdot (10 + 1) = 1584$ measurements in 40 years of the modeling time. These measurements are generated with the hydraulic conductivity field \mathbf{k} shown in Figure 4.

Figure 6 exhibits posterior statistics (mean $\langle Y \rangle$ and standard deviation σ_Y) of the log-conductivity $Y(\mathbf{x})$, obtained after the assimilation of all 1,584 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 3), but the realizations from the posterior ensemble exhibit more realistic features (right column in Figure 6). Regardless of the forward model used (the only difference in these two experiments), our data assimilation framework yields consistent predictions of σ_Y (middle column in Figure 6). 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 σ_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 not significantly affected by 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 histograms of the standard deviation σ_Y on all pixels for CAAE-ESMDA and CAAE-DenseED-ESMDA are shown in Figure S1 in Supporting Information S1; σ_Y for the latter is slightly higher than σ_Y for the former.

The same inversion experiments yield estimates of the contaminant release history \mathbf{S} , which are shown in Figure 7 for CAAE-ESMDA and CAAE-DenseED-ESMDA. Regardless of the forward model used, our inversion algorithm accurately estimates the release strength during stress periods 1 and 2 ($S_{s,1}$, $S_{s,2}$); for period 4 ($S_{s,4}$), the discrepancy between the two experiments indicates the impact of the DenseED surrogate model error, which can be reduced by improving the DenseED training process; the estimates are close to their reference values and have tight 95% confidence intervals. Although the estimates of the source strength during stress periods 3 and 5 ($S_{s,3}$ and $S_{s,5}$) are not as close to the reference values, they still fall within the 95% confidence intervals. A potential cause for this discrepancy is the magnitude of the reference value: the error for the assimilated source terms is larger when the reference value is large, not only in this experiment but also in the two additional experiments reported in the SI. The divergence of $S_{s,3}$ and $S_{s,5}$ occurred in both experiments, without the surrogate forward

DenseED model, implying that the surrogate model error is not the source of the uncertainty in these two parameters. Additionally, since the R^2 of the prediction with CAAE is as high as 0.96, we rule out CAAE as the source of this uncertainty and claim that this uncertainty is mostly due to the ill-posedness of the problem caused by the sparse and noisy measurements. The two assimilation strategies yield very similar estimates of the contaminant release location, $\mathbf{S}_l = (S_l^x, S_l^y)^T$; the estimates of both quantities have tight confidence intervals, but the estimated value of S_l^x for CAAE-DenseED-ESMDA inversion lies slightly farther from the reference value than that of CAAE-ESMDA.

The computational costs of CAAE-ESMDA with the PDE-based forward model and its counterpart with the DenseED CNN surrogate are shown in Table 4. 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 $N_{\text{sum}} = N_e \times (10 + 1) = 10,560$ forward model runs. Overall, CAAE-DenseED-ESMDA is one order of magnitude faster than CAAE-ESMDA with the PDE-based forward model. The representation errors of the neural networks are similar, as reported in Table S3 in Supporting Information S1 in terms of the coefficient of determination, R^2 .

5. Discussion

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 multi-core parallelization for the PDE-based forward model.

Our numerical experiments utilize CAAE for parameterization, which has two main benefits. First, it relieves the computational burden of the ESMDA inversion. In our experiments, CAAE reduced the number of the total parameters to be inferred from 19,933 to 931. Since the computational cost of ESMDA is linear in the state size (Evensen et al., 2019), the ESMDA inversion is accelerated by ≈ 20 times. Second, CAAE is capable of capturing the channelized characteristics of a conductivity field as this prior information is integrated into the CAAE at the training stage. The ESMDA inversion without CAAE might yield a conductivity field that either has a lower resolution or loses the sharp edges of the channels (Kang et al., 2021). Hence, even if the computation savings of CAAE (the first benefit) do not outweigh its training cost, the presence of channels calls for its use (the second benefit).

The integration of the three distinct components into a single inversion framework has its limitations. **CAAE:** The loss of fine features of a conductivity field is unavoidable in reduced-order modeling (see, e.g., Figure 4). It is not trivial to perform ESMDA without CAAE because ESMDA assumes Gaussianity of an unknown parameter, which is invalid for the original conductivity field. This undermines the veracity of the inversion procedure with or without CAAE. In related studies, the training times for CAAE neural networks are 1.7 and 2.7 hr for two-dimensional fields with 32×64 and 40×80 discretizations (Kang et al., 2021; Mo et al., 2019b); 13.1 hr for three-dimensional fields with $6 \times 32 \times 64$ discretization (Mo et al., 2019b); and only 5.2 hr for our experiments. The CAAE training time might negate the computational gain from the parameterization in the inversion process. **ESMDA:** Bench-marking against other more accurate inversion methods such as MCMC requires exhaustive simulations given the large number of unknown parameters. Since ESMDA requires an unknown parameter to be Gaussian, its application to a typical subsurface problem is impossible without a parameterization. A particular choice of the hyper-parameters in ESMDA (number of samples in each ensemble, number of iterations, etc.) might affect the inversion accuracy; we leave a detailed analysis of this issue for a follow-up study. **DenseED:** The inversion accuracy of CAAE-DenseED-ESMDA is related to the quality of both a CAAE parameterization and a DenseED surrogate model. There is a trade-off between the computational time and accuracy unless neural network surrogates can be improved such that they would not require many hours to train or require significantly fewer simulations than the ESMDA procedure (Song & Tartakovsky, 2021). As mentioned above, the performance of the DenseED surrogate model depends on a particular choice of hyper-parameters such as number of training samples, network architecture, training epochs, and so on; an analysis of the relative importance of these parameters could reveal the upper bound of the quality of DenseED, and provide more insight into the robustness of the framework. Another limitation of our method stems from the fixed input-output structure of the DenseED

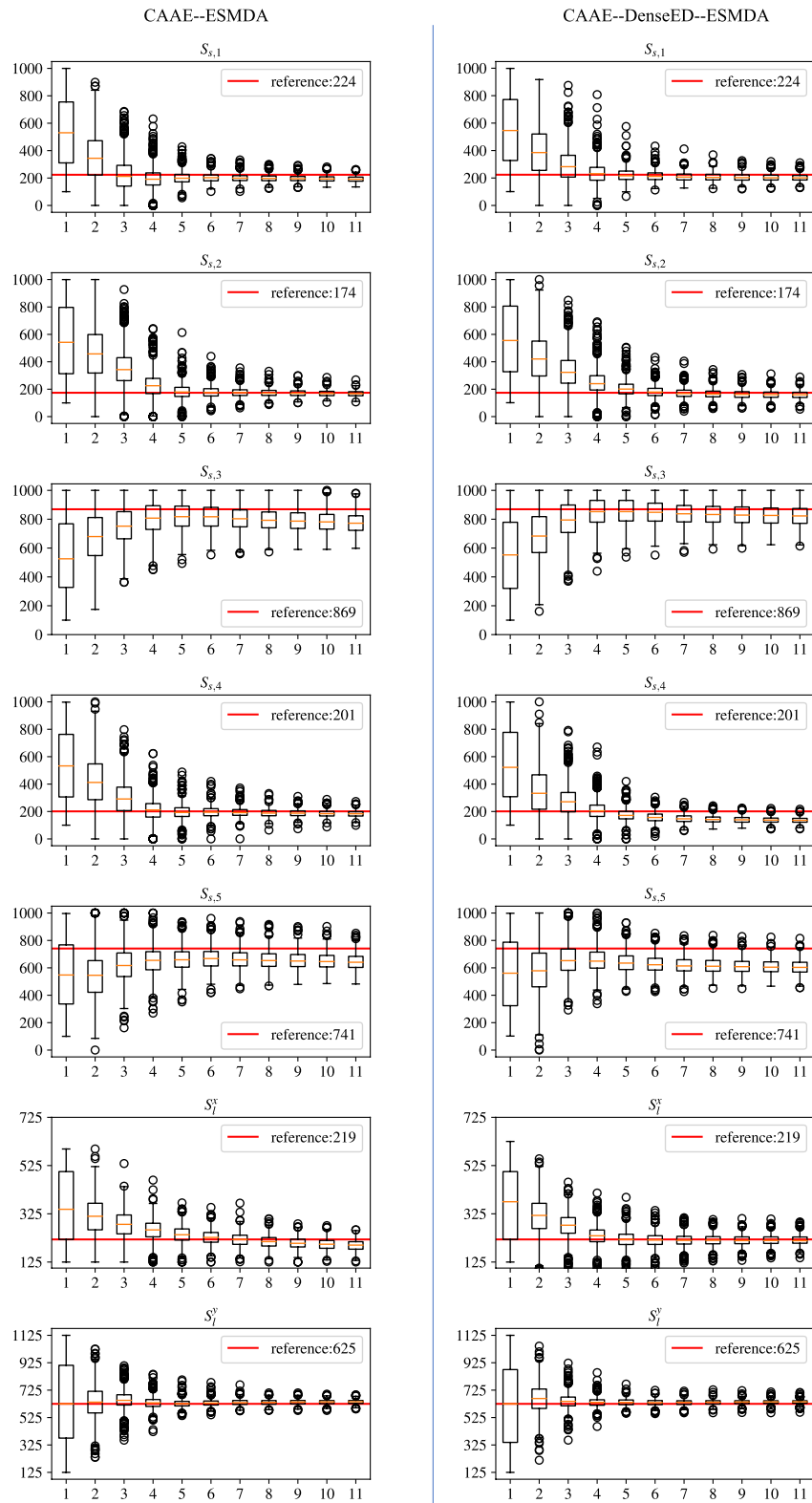


Figure 7. Boxplots of the ensembles for the contaminant release terms, $S = (S_1, S_s)$ with $S_1 = (S_1^x, S_1^y)^T$ and $S_s = (S_{s,1}, \dots, S_{s,5})^T$, and their confidence intervals. These are computed with either the CAAE-ESMDA inversion of the PDE-based forward model (left column) or the AAE-DenseED-ESMDA inversion with the DenseED CNN surrogate. The results are plotted as function of the ESMDA iterations and contrasted with their reference values (horizontal lines). Source location S_1 is in m; contaminant release strength in each of the five stress periods, S_s , is in g/m^3 .

Table 4

Total Run Time of the CAAE-ESMDA, T_{run} , Includes the Costs of the PDE-Based Forward Model and Its CNN Surrogate

	N_{sum}	T_{run}	$T_{dataset}$	$T_{train}^{DenseED}$	T_{ave}	T_{train}^{CAAE}
CAAE-ESMDA	10,560	388,200.0	0.0	0.0	36.8	18,678.2
CAAE-DenseED-ESMDA	10,560	1893.9	34,922.0	9439.2	4.4	18,678.2

Note. The average run-time per sample, T_{ave} , is defined as $T_{ave} = (T_{run} + T_{dataset} + T_{train}^{DenseED}) / N_{sum}$, where $T_{dataset}$ is the time for obtaining the training and testing data sets, and T_{train}^{CAAE} is the CNN training time. Since the CAAE parameterization is used in both cases, the training time T_{train}^{CAAE} is the same. The running time of CAAE is negligible in both data assimilation strategies. All times are in seconds.

surrogate forward model. Once the DenseED is trained, it does not generalize to produce predictions at arbitrary time, limiting the contaminant release history to predefined times.

Our method can be extended to handle other unknown parameters, such as porosity or reaction rate constants, with no significant adjustment. For example, a spatially variable porosity field can be treated similarly to the permeability field in our experiment, that is, CAAE parameterization can be used to re-parameterize the porosity field, and the corresponding latent variable can be inferred by ESMDA. A parameter that is not spatially-dependent can be added as an extra channel to the input of the CNN surrogate forward model. The corresponding ESMDA inversion part would be similar to that for other parameters. While our framework is sufficiently general to be used on other applications, doing so would require hyper-parameter tuning and might result in a different level of uncertainty of the inversion results.

6. Conclusion

We proposed a CAAE-DenseED-ESMDA algorithm to infer both heterogeneous 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 CAAE 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 flow and transport models to accelerate the forward runs; and adopts ESMDA to solve the inverse problem.

The deployment of CAAE-DenseED-ESMDA allows one to investigate questions, such as the rate of decay of the data assimilation error with the ensemble size or the optimal design of a network of observational wells, that cannot be answered with CAAE-ESMDA with the PDE-based forward model, whose computational cost might be prohibitive. Our 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 history 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 fall within its 95% confidence interval.
3. The DenseED CNN surrogate used in our CAAE-ESMDA inversion framework provides an order-of-magnitude speed up, while giving visually similar 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.

The SI contains two additional sets of CAAE-DenseED-ESMDA experiments with different conductivity fields and release histories. In both experiments, the ensembles for contaminant release history terms converge to their

true values with at most 11.75% error; higher uncertainty and larger errors are associated with the terms that have high absolute value. The reconstruction errors of the conductivity fields are comparable to those reported in the main text.

Data Availability Statement

There are no data sharing issues since all of the numerical information is provided in the figures produced by solving the equations in the paper. We reused the channelized conductivity field data from the open sourced dataset in <https://github.com/GAIA-UNIL/trainingimages>, and simulated the contaminant transport processes with MODFLOW and MT3DMS. These data and the source code are available at <https://doi.org/10.5281/zenodo.6443086>.

Acknowledgments

ZZ and DT were supported in part by National Science Foundation grant EAR-2100927 and by Air Force Office of Scientific Research under award number FA9550-21-1-0381. NZ acknowledges support from ARPA-E, award # DE-AR0001204.

References

- 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. [https://doi.org/10.1061/\(asce\)1084-0699\(2001\)6:3\(225\)](https://doi.org/10.1061/(asce)1084-0699(2001)6:3(225))
- Ayvaz, M. T. (2016). A hybrid simulation–optimization approach for solving the areal groundwater pollution source identification problems. *Journal of Hydrology*, 538, 161–176. <https://doi.org/10.1016/j.jhydrol.2016.04.008>
- Bakker, M., Post, V., Langevin, C. D., Hughes, J. D., White, J. T., Starn, J. J., & Fioren, M. N. (2016). Scripting MODFLOW model development using Python and FloPy. *Groundwater*, 54(5), 733–739. <https://doi.org/10.1111/gwat.12413>
- Barajas-Solano, D. A., Alexander, F. J., Anghel, M., & Tartakovsky, D. M. (2019). Efficient gHMC reconstruction of contaminant release history. *Frontier in Environment Science*, 7, 149. <https://doi.org/10.3389/fenvs.2019.00149>
- 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.)*. US Geological Survey.
- Boso, F., & Tartakovsky, D. M. (2020a). Data-informed method of distributions for hyperbolic conservation laws. *SIAM Journal on Scientific Computing*, 42(1), A559–A583. <https://doi.org/10.1137/19m1260773>
- Boso, F., & Tartakovsky, D. M. (2020b). Learning on dynamic statistical manifolds. *Proceedings of the Royal Society A*, 476(2239), 20200213. <https://doi.org/10.1098/rspa.2020.0213>
- Boso, F., & Tartakovsky, D. M. (2022). Information geometry of physics-informed statistical manifolds and its use in data assimilation. *Journal of Computational Physics*, 467, 111438. <https://doi.org/10.1016/j.jcp.2022.111438>
- Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of compstat'2010* (pp. 177–186). Springer. https://doi.org/10.1007/978-3-7908-2604-3_16
- 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. <https://doi.org/10.1016/j.petrol.2019.02.037>
- 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. *Advances in Water Resources*, 122, 340–354. <https://doi.org/10.1016/j.advwatres.2018.10.023>
- Emerick, A. A., & Reynolds, A. C. (2013). Ensemble smoother with multiple data assimilation. *Computers & Geosciences*, 55, 3–15. <https://doi.org/10.1016/j.cageo.2012.03.011>
- Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99(C5), 10143–10162. <https://doi.org/10.1029/94jc00572>
- Evensen, G. (2003). The ensemble kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4), 343–367. <https://doi.org/10.1007/s10236-003-0036-9>
- Evensen, G., Raanes, P. N., Stordal, A. S., & Hove, J. (2019). Efficient implementation of an iterative ensemble smoother for data assimilation and reservoir history matching. *Frontiers in Applied Mathematics and Statistics*, 47. <https://doi.org/10.3389/fams.2019.00047>
- Gamerman, D., & Lopes, H. F. (2006). *Markov chain Monte Carlo: Stochastic simulation for Bayesian inference*. Chapman and Hall/CRC.
- Gao, G., & Reynolds, A. C. (2004). An improved implementation of the LBFSG algorithm for automatic history matching. In *SPE annual technical conference and exhibition*. <https://doi.org/10.2118/90058-MS>
- Ghorbanidehno, H., Kokkinaki, A., Lee, J., & Darve, E. (2020). Recent developments in fast and scalable inverse modeling and data assimilation methods in hydrology. *Journal of Hydrology*, 591, 125266. <https://doi.org/10.1016/j.jhydrol.2020.125266>
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial networks. arXiv preprint arXiv:1406.2661.
- Haario, H., Laine, M., Mira, A., & Saksman, E. (2006). Dram: Efficient adaptive MCMC. *Statistics and Computing*, 16(4), 339–354. <https://doi.org/10.1007/s11222-006-9438-0>
- Haario, H., Saksman, E., & Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, 7(2), 223–242. <https://doi.org/10.2307/3318737>
- Harbaugh, A. W. (2005). *MODFLOW-2005, the US Geological Survey modular ground-water model: The ground-water flow process*. US Department of the Interior, US Geological Survey.
- Hermans, T., Lesparre, N., De Schepper, G., & Robert, T. (2019). Bayesian evidential learning: A field validation using push-pull tests. *Hydrogeology Journal*, 27(5), 1661–1672. <https://doi.org/10.1007/s10040-019-01962-9>
- Hermans, T., Nguyen, F., Klepikova, M., Dassargues, A., & Caers, J. (2018). Uncertainty quantification of medium-term heat storage from short-term geophysical experiments using Bayesian evidential learning. *Water Resources Research*, 54(4), 2931–2948. <https://doi.org/10.1002/2017wr022135>
- Jiang, S., & Durlafsky, L. J. (2021). Data-space inversion using a recurrent autoencoder for time-series parameterization. *Computational Geosciences*, 25(1), 411–432. <https://doi.org/10.1007/s10596-020-10014-1>
- Ju, L., Zhang, J., Meng, L., Wu, L., & Zeng, L. (2018). An adaptive Gaussian process-based iterative ensemble smoother for data assimilation. *Advances in Water Resources*, 115, 125–135. <https://doi.org/10.1016/j.advwatres.2018.03.010>
- Kang, X., Kokkinaki, A., Kitanidis, P. K., Shi, X., Lee, J., Mo, S., & Wu, J. (2021). Hydrogeophysical characterization of nonstationary dnapl source zones by integrating a convolutional variational autoencoder and ensemble smoother. *Water Resources Research*, 57(2), e2020WR028538. <https://doi.org/10.1029/2020wr028538>

- Kang, X., Kokkinaki, A., Kitanidis, P. K., Shi, X., Revil, A., Lee, J., et al. (2020). Improved characterization of dnapl source zones via sequential hydrogeophysical inversion of hydraulic-head, self-potential and partitioning tracer data. *Water Resources Research*, 56(8), e2020WR027627. <https://doi.org/10.1029/2020wr027627>
- Kim, S., Min, B., Kwon, S., & Chu, M.-G. (2019). *History matching of a channelized reservoir using a serial denoising autoencoder integrated with ES-MDA*. Geofluids.
- Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114.
- Laloy, E., Héroult, R., Jacques, D., & Linde, N. (2018). Training-image based geostatistical inversion using a spatial generative adversarial neural network. *Water Resources Research*, 54(1), 381–406. <https://doi.org/10.1002/2017wr022148>
- Laloy, E., Héroult, R., Lee, J., Jacques, D., & Linde, N. (2017). Inversion using a new low-dimensional representation of complex binary geological media based on a deep neural network. *Advances in Water Resources*, 110, 387–405. <https://doi.org/10.1016/j.advwatres.2017.09.029>
- Laloy, E., Linde, N., Ruffino, C., Héroult, R., Gasso, G., & Jacques, D. (2019). Gradient-based deterministic inversion of geophysical data with generative adversarial networks: Is it feasible? *Computers & Geosciences*, 133, 104333. <https://doi.org/10.1016/j.cageo.2019.104333>
- Leichombam, S., & Bhattacharjya, R. K. (2018). New hybrid optimization methodology to identify pollution sources considering the source locations and source flux as unknown. *Journal of Hazardous, Toxic, and Radioactive Waste*, 23(1), 04018037.
- Liggett, J. E., Partington, D., Frei, S., Werner, A. D., Simmons, C. T., & Fleckenstein, J. H. (2015). An exploration of coupled surface–subsurface solute transport in a fully integrated catchment model. *Journal of Hydrology*, 529, 969–979. <https://doi.org/10.1016/j.jhydrol.2015.09.006>
- Linde, N., Renard, P., Mukerji, T., & Caers, J. (2015). Geological realism in hydrogeological and geophysical inverse modeling: A review. *Advances in Water Resources*, 86, 86–101. <https://doi.org/10.1016/j.advwatres.2015.09.019>
- Liu, Y., Sun, W., & Durlflosky, L. J. (2019). A deep-learning-based geological parameterization for history matching complex models. *Mathematical Geosciences*, 51(6), 725–766. <https://doi.org/10.1007/s11004-019-09794-9>
- Lopez-Alvis, J., Laloy, E., Nguyen, F., & Hermans, T. (2021). Deep generative models in inversion: The impact of the generator's nonlinearity and development of a new approach based on a variational autoencoder. *Computers & Geosciences*, 152, 104762. <https://doi.org/10.1016/j.cageo.2021.104762>
- 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). <https://doi.org/10.1029/2011wr010412>
- Michel, H., Nguyen, F., Kremer, T., Elen, A., & Hermans, T. (2020). 1D geological imaging of the subsurface from geophysical data with Bayesian evidential learning. *Computers & Geosciences*, 138, 104456. <https://doi.org/10.1016/j.cageo.2020.104456>
- Mo, S., Zabarar, 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. <https://doi.org/10.1029/2018wr024638>
- Mo, S., Zabarar, 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.
- Park, J., & Caers, J. (2020). Direct forecasting of global and spatial model parameters from dynamic data. *Computers & Geosciences*, 143, 104567. <https://doi.org/10.1016/j.cageo.2020.104567>
- Sarma, P., Durlflosky, L. J., & Aziz, K. (2008). Kernel principal component analysis for efficient, differentiable parameterization of multipoint geostatistics. *Mathematical Geosciences*, 40(1), 3–32. <https://doi.org/10.1007/s11004-007-9131-7>
- Snodgrass, M. F., & Kitanidis, P. K. (1997). A geostatistical approach to contaminant source identification. *Water Resources Research*, 33(4), 537–546. <https://doi.org/10.1029/96wr03753>
- Song, D. H., & Tartakovsky, D. M. (2021). Transfer learning on multi-fidelity data. *Journal of Machine Learning for Modeling and Computing*, 3(1), 31–47. <https://doi.org/10.1615/JMachLearnModelComput.2021038925>
- Tang, M., Liu, Y., & Durlflosky, 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. <https://doi.org/10.1016/j.cma.2020.113636>
- Tartakovsky, D. M., & Winter, C. L. (2008). Uncertain future of hydrogeology. *Journal of Hydrologic Engineering*, 13(1), 37–39. [https://doi.org/10.1061/\(asce\)1084-0699\(2008\)13:1\(37\)](https://doi.org/10.1061/(asce)1084-0699(2008)13:1(37))
- Thibaut, R., Laloy, E., & Hermans, T. (2021). A new framework for experimental design using Bayesian evidential learning: The case of wellhead protection area. *Journal of Hydrology*, 603, 126903. <https://doi.org/10.1016/j.jhydrol.2021.126903>
- 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. [https://doi.org/10.1175/1520-0493\(1996\)124<2898:daaimi>2.0.co;2](https://doi.org/10.1175/1520-0493(1996)124<2898:daaimi>2.0.co;2)
- Vo, H. X., & Durlflosky, 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. <https://doi.org/10.1007/s11004-014-9541-2>
- 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. <https://doi.org/10.1016/j.ijggc.2020.103223>
- White, R. E., & Wu, Y. H. (2015). Nonlinear least squares algorithm for identification of hazards. *Cogent Mathematics*, 2(1), 1118219. <https://doi.org/10.1080/23311835.2015.1118219>
- Winter, C. L., Guadagnini, A., Nychka, D., & Tartakovsky, D. M. (2006). Multivariate sensitivity analysis of saturated flow through simulated highly heterogeneous groundwater aquifers. *Journal of Computational Physics*, 217(1), 166–175. <https://doi.org/10.1016/j.jcp.2006.01.047>
- 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. <https://doi.org/10.1023/a:1022277418570>
- Wu, Z., Reynolds, A. C., & Oliver, D. S. (1999). Conditioning geostatistical models to two-phase production data. *SPE Journal*, 4(02), 142–155. <https://doi.org/10.2118/56855-pa>
- Xia, Y., & Zabarar, 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 Resource Research*, 52(8), 6587–6595. <https://doi.org/10.1002/2016wr019111>
- Xu, T., & Gómez-Hernández, J. J. (2018). Simultaneous identification of a contaminant source and hydraulic conductivity via the restart normal-score ensemble Kalman filter. *Advances in Water Resources*, 112, 106–123. <https://doi.org/10.1016/j.advwatres.2017.12.011>
- 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. <https://doi.org/10.1029/2020WR027643>
- Yeh, H.-D., Chang, T.-H., & Lin, Y.-C. (2007). Groundwater contaminant source identification by a hybrid heuristic approach. *Water Resources Research*, 43(9). <https://doi.org/10.1029/2005wr004731>

- Zhang, J., Lin, G., Li, W., Wu, L., & Zeng, L. (2018). An iterative local updating ensemble smoother for estimation and uncertainty assessment of hydrologic model parameters with multimodal distributions. *Water Resources Research*, *54*(3), 1716–1733. <https://doi.org/10.1002/2017wr020906>
- Zhang, J., Zeng, L., Chen, C., Chen, D., & Wu, L. (2015). Efficient Bayesian experimental design for contaminant source identification. *Water Resources Research*, *51*(1), 576–598. <https://doi.org/10.1002/2014wr015740>
- Zhou, H., Gómez-Hernández, J. J., & Li, L. (2014). Inverse methods in hydrogeology: Evolution and recent trends. *Advances in Water Resources*, *63*, 22–37. <https://doi.org/10.1016/j.advwatres.2013.10.014>
- Zhou, Z., & Tartakovsky, D. M. (2021). Markov chain Monte Carlo with neural network surrogates: Application to contaminant source identification. *Stochastic Environmental Research and Risk Assessment*, *35*(3), 639–651. <https://doi.org/10.1007/s00477-020-01888-9>

References From The Supporting Information

- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026–1034). <https://doi.org/10.1109/iccv.2015.123>
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448–456).

Supporting Information for “Deep Learning for Simultaneous Inference of Hydraulic and Transport Properties”

Zitong Zhou¹, Nicholas Zabaras², and Daniel M. Tartakovsky¹

¹Department of Energy Science and Engineering, Stanford University, Stanford, CA 94305, USA

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

Contents of this file

1. Text S1 and S2
2. Figures S1 to S10
3. Tables S1 to S4

S1. Description of DNN

In this section we provide details of the DNNs used in this study: CAAE for parameterization of a hydraulic conductivity field, and the DenseED surrogate model predicting groundwater flow and contaminant transport.

CAAE

As briefly introduced in Section 3.2, the CAAE consists of three networks: an encoder (\mathcal{G}), a decoder (\mathcal{D}_e), and a discriminator (\mathcal{D}). The workflow of these three networks and the architecture of each network are shown in Figure S2. The residual-in-residual dense block (RRDB) used in the encoder and the decoder is illustrated in Figure S3, in which

the dense block is reused in the DenseED surrogate model as well, and is shown in Figure S4. The dimensions of the internal layers of the encoder and the decoder are shown in Table S2. Batch normalization (BN) (Ioffe & Szegedy, 2015), three-dimensional convolutional operations (Conv) (Goodfellow et al., 2016), Sigmoid, ReLU and LeakyReLU non-linear activation functions (He et al., 2015) are used in these neural networks; “FC(128)” denotes a fully-connected layer with the output vector length being 128, “Upsample” layer doubles the size of the input feature maps with the nearest upsampling method.

Figure S5 provides a visual comparison of the PDE-based simulation with the reconstructed conductivity field. It demonstrates the effect of using CAAE on the predicted fields. This figure and Figure 5 use the same set of release history and conductivity field input.

DenseED

The architecture of the DenseED in this study is shown in Figure S6. The DenseED neural network structure consists of three main sub-structures: dense blocks, encoding layers, and decoding layers. The structure of a dense block is illustrated in Figure S4a. An encoding layer is shown in Figure S4b, with which both the feature number and the size of the features are halved: $x^0 \in \mathbb{R}^{C \times W \times H \times D}$, $x' \in \mathbb{R}^{\frac{1}{2}C \times W \times H \times D}$, $x^1 \in \mathbb{R}^{\frac{1}{2}C \times \frac{1}{2}W \times \frac{1}{2}H \times \frac{1}{2}D}$. This figure can represent a decoding layer as well, with the feature number halved, the size doubled: $x^0 \in \mathbb{R}^{C \times W \times H \times D}$, $x' \in \mathbb{R}^{\frac{1}{2}C \times W \times H \times D}$, $x^1 \in \mathbb{R}^{\frac{1}{2}C \times 2W \times 2H \times 2D}$. In addition to these three main elements, the size of the features are first halved with the very first Conv layer. The last decoding layer maps the feature number to that of the output. The dimensions of the outputs from each block are shown in Table S4.

S2. Two additional sets of CAAE-DenseED-ESMDA experiments

This section contains two sets of CAAE-DenseED-ESMDA inversion experiments with release histories and conductivity fields that are different from their counterparts used in Section 4. The results of the second set of experiments are shown in Figures S7 and S8. The results of the third set of experiments are shown in Figures S9 and S10. These two sets of results are obtained with our CAAE-DenseED-ESMDA framework. The reference (true) value of the conductivity field and the release history are plotted in those figures as well. Both experiments demonstrate the high accuracy of contamination strength identification, with the maximum discrepancy of the release strength being 11.75% in the second experiment, and 9.05% in the third experiment. The assimilation for the second experiment of the release location achieves similar performance with the experiment in Section 4, and that of the third experiment was the best among all experiments. The reconstructed conductivity fields in Figures S8 and S10 capture a part of the true fields, yet with high uncertainty. This might be attributed to the effect of CAAE reconstruction error, forward surrogate model DenseED error, and the uncertainty arising from the sparse and noisy measurements.

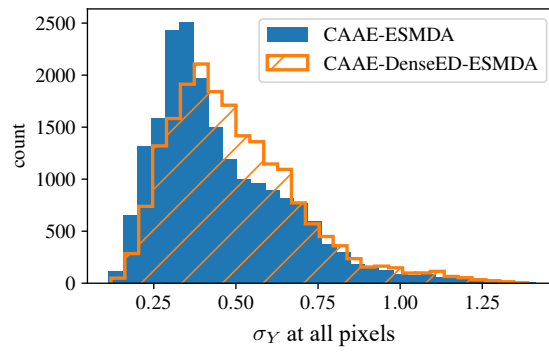


Figure S1. Histograms of the standard deviation σ_Y on all pixels for CAAE-ESMDA and CAAE-DenseED-ESMDA.

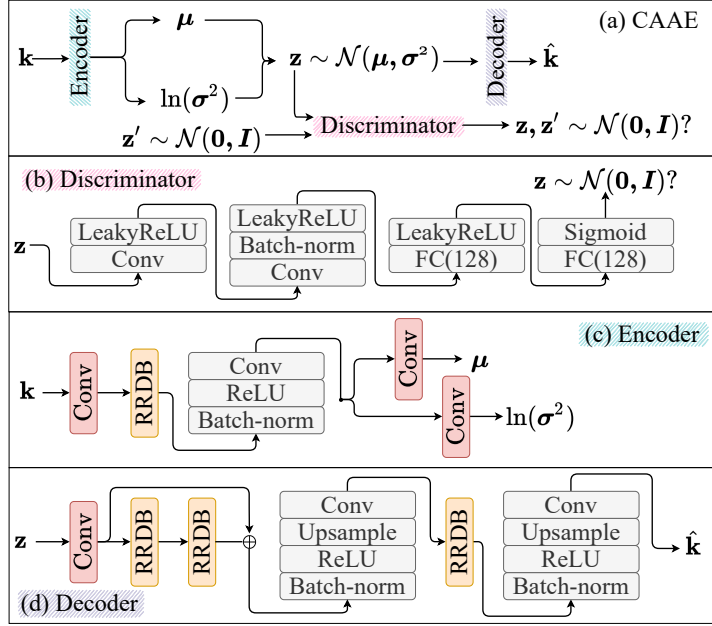


Figure S2. (a) CAAE, (b) Discriminator, (c) Encoder, and (d) Decoder. The CAAE (a) consists of an encoder (c) and a decoder (d), the discriminator (b) is trained as well to enforce the distribution of the low-dimensional latent variable \mathbf{z} . “RRDB” blocks are depicted in Figure S3, with the slope parameter in “LeakyReLU” being 0.2 in this study. \oplus denotes element-wise summation.

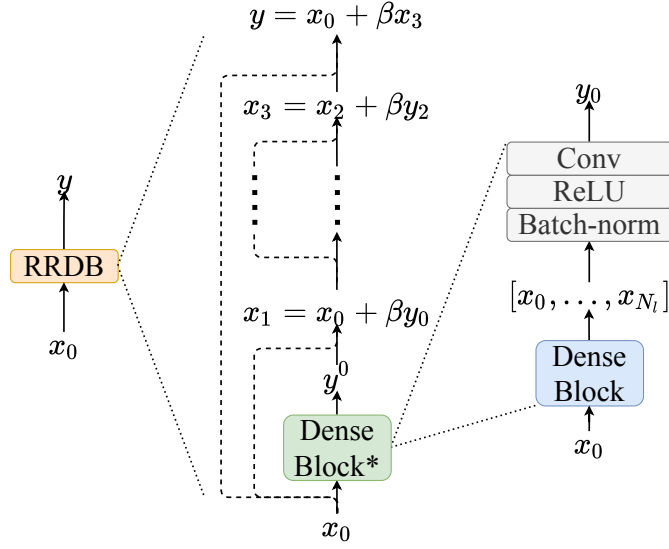


Figure S3. RRDB block structure. Each RRDB block used in this study consists of $N_{DB} = 3$ dense blocks, with the internal layer number of each dense block being $N_l = 5$. The dense block is illustrated in (a) in Figure S4.

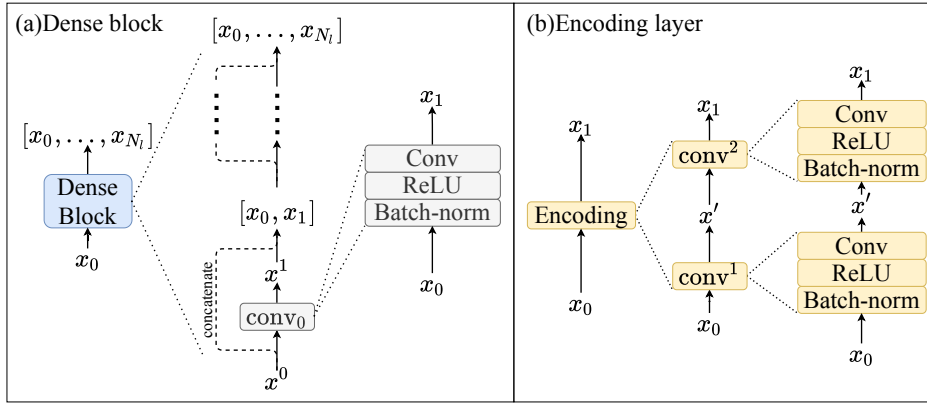


Figure S4. (a) A dense block with N_l internal layers. (b) Encoding layer structure in Figure S6. The feature number of x^0 is halved first, then the size of the features are halved: $x^0 \in \mathbb{R}^{C \times W \times H \times D}$, $x' \in \mathbb{R}^{\frac{1}{2}C \times W \times H \times D}$, $x^1 \in \mathbb{R}^{\frac{1}{2}C \times \frac{1}{2}W \times \frac{1}{2}H \times \frac{1}{2}D}$.

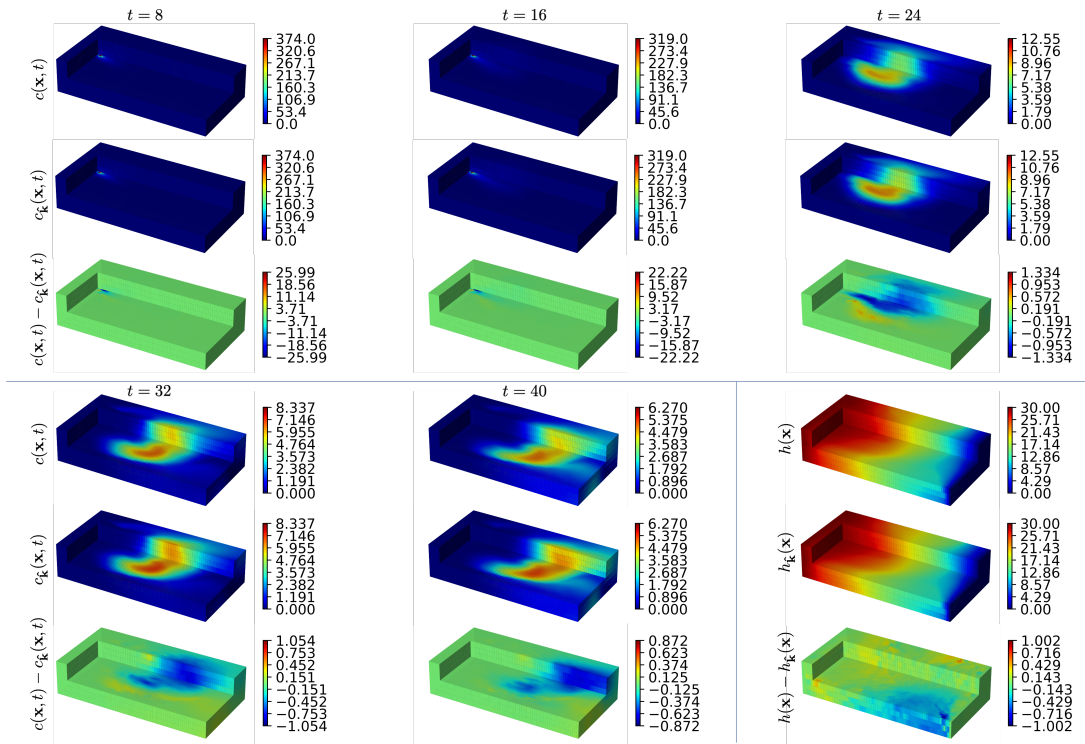


Figure S5. Prediction of the solute concentration obtained with the original training set of conductivity field, $c(\mathbf{x}, t)$, and with the CAAE reconstructed conductivity field, $c_{\hat{\mathbf{k}}}(\mathbf{x}, t)$. Also shown are the corresponding predictions of the hydraulic head, $h(\mathbf{x})$ and $h_{\hat{\mathbf{k}}}(\mathbf{x})$; and the difference between these two types.

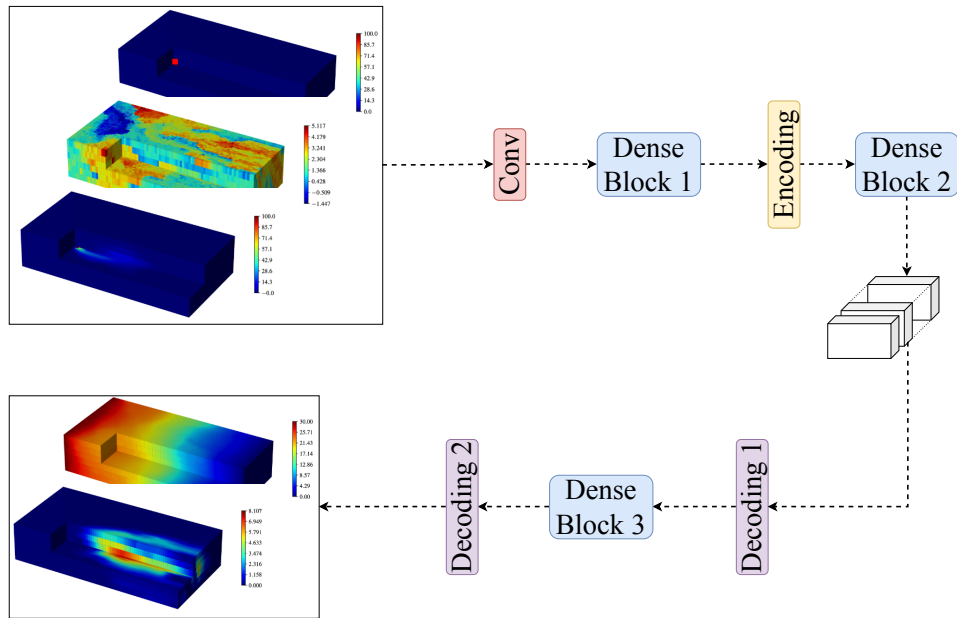


Figure S6. Dense encoder decoder (DenseED) architecture with three dense blocks. The cubes as the output of “Dense Block 2” represent the encoded coarse high-level features.

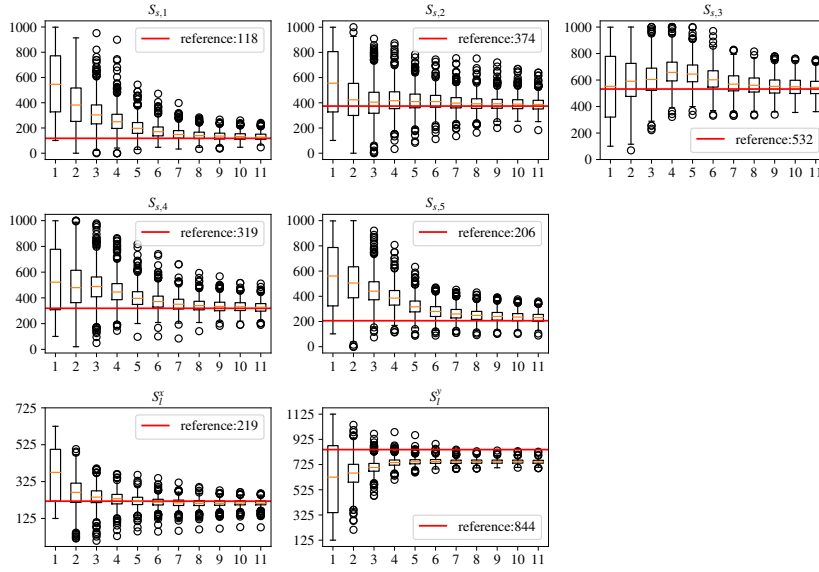


Figure S7. Boxplots of the ensembles for the contaminant release terms, $\mathbf{S} = (\mathbf{S}_1, \mathbf{S}_s)$ with $\mathbf{S}_1 = (S_1^x, S_1^y)^\top$ and $\mathbf{S}_s = (S_{s,1}, \dots, S_{s,5})^\top$, and their confidence intervals, obtained via the CAAE-DenseED-ESMDA inversion with the PDE-based 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}_1 is in m; and the contaminant release strength in each of the five stress periods, \mathbf{S}_s , is in g/m^3 .

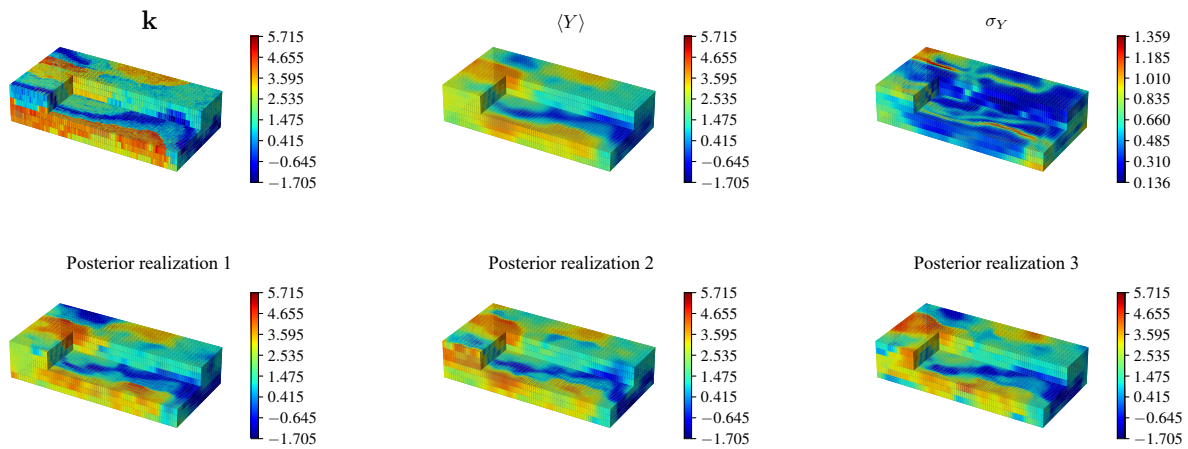


Figure S8. True conductivity field \mathbf{k} , posterior mean ($\langle Y \rangle$, left column) and standard deviation (σ_Y , middle column) of the log-conductivity field $Y(\mathbf{x})$ obtained upon assimilation of concentration and head measurements from the dense observation network. These statistics are obtained via our inversion algorithm CAAE-DenseED-ESMDA. Also shown are three representative realizations from the resulting posterior ensemble (bottom row).

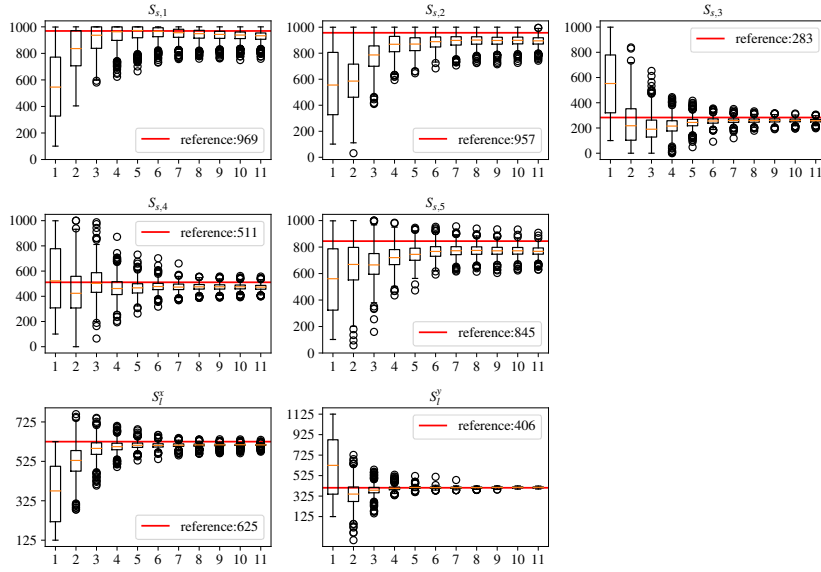


Figure S9. Boxplots of the ensembles for the contaminant release terms, $\mathbf{S} = (\mathbf{S}_1, \mathbf{S}_s)$ with $\mathbf{S}_1 = (S_1^x, S_1^y)^\top$ and $\mathbf{S}_s = (S_{s,1}, \dots, S_{s,5})^\top$, and their confidence intervals, obtained via the CAAE-DenseED-ESMDA inversion with the PDE-based 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}_1 is in m; and the contaminant release strength in each of the five stress periods, \mathbf{S}_s , is in g/m^3 .

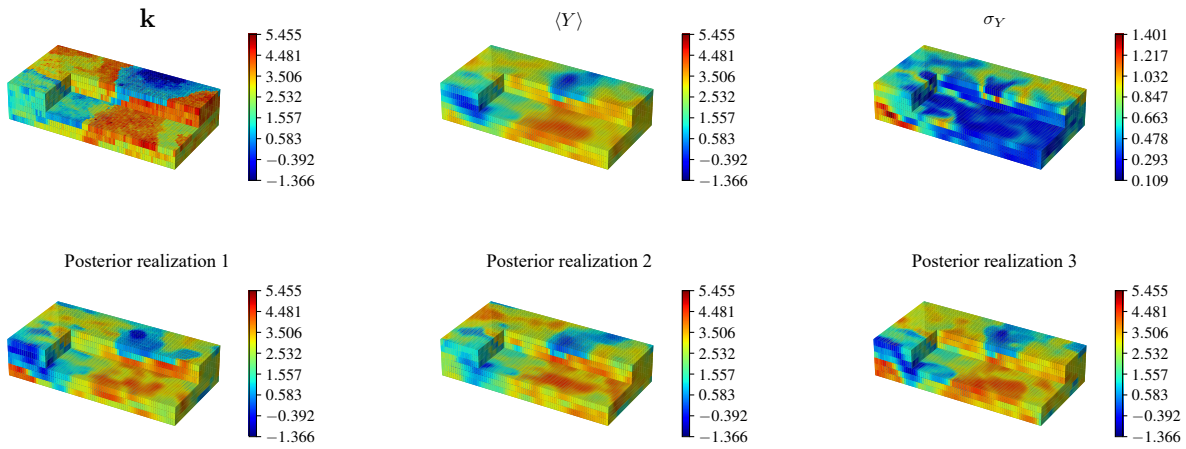


Figure S10. True conductivity field \mathbf{k} , posterior mean ($\langle Y \rangle$, left column) and standard deviation (σ_Y , middle column) of the log-conductivity field $Y(\mathbf{x})$ obtained upon assimilation of concentration and head measurements from the dense observation network. These statistics are obtained via our inversion algorithm CAAE-DenseED-ESMDA. Also shown are three representative realizations from the resulting posterior ensemble (bottom row).

Table S1. Constants and matrix dimensions for the first set of experiments.

Term	Representation	Value
N_{re}	number of release period	5
M	number of measurement location	144
I	number of measurement time for concentration	10
$\text{dim}(\mathbf{k})$	dimension of conductivity field	$81 \times 41 \times 6$
N_d	number of total measurements	$M(I + 1) = 1584$
$\text{dim}(\mathbf{z})$	dimension of latent \mathbf{z}	$2 \times 2 \times 11 \times 21$
N_m	number of unknown parameters	$N_{\text{re}} + 2 + 2 \cdot 2 \cdot 11 \cdot 21 = 931$
N_e	number of samples in an ESMDA ensemble	960
N_a	number of ESMDA iteration	10

Table S2. Dimensions 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 S3.

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	$81 \times 41 \times 6$

Table S3. The R^2 coefficient of determination for the CAAE and DenseED. CAAE is trained with 23000 conductivity realizations; DenseED is trained with 800 realizations of different conductivity field and release histories. $R^2(\mathbf{k}, \hat{\mathbf{k}})$ is computed by comparing the concentration and hydraulic head field (averaged over 150 realizations) with the reconstructed conductivity fields and the original conductivity fields.

	N_{train}	$N_{\text{eval}}^{\text{simu}}$	R^2
CAAE	23000	150	0.96
DenseED	800	150	0.79

Table S4. 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 S4.

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	$81 \times 41 \times 6$