Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

Bayesian learning of gas transport in three-dimensional fracture networks

Yingqi Shi [a,b], Donald J. Berry [a,c], John Kath [a], Shams Lodhy [a,b], An Ly [a], Allon G. Percus [a,*], Jeffrey D. Hyman [d], Kelly Moran [e], Justin Strait [e], Matthew R. Sweeney [d], Hari S. Viswanathan [d], Philip H. Stauffer [d]

[a] Institute of Mathematical Sciences, Claremont Graduate University, Claremont, CA 91711, USA
[b] Drucker School of Management, Claremont Graduate University, Claremont, CA 91711, USA
[c] Center for Information Systems and Technology, Claremont Graduate University, Claremont, CA 91711, USA
[d] Energy and Natural Resources Security Group, Earth and Environmental Sciences Division, Los Alamos National Laboratory, Los Alamos, NM, 87545, USA
[e] Statistical Sciences Group, Computer, Computational, and Statistical Sciences Division, Los Alamos National Laboratory, Los Alamos, NM, 87545, USA

ARTICLE INFO

Keywords:
Machine learning
Gaussian process regression
Subsurface hydrology
Discrete fracture networks
Uncertainty quantification
Surrogate modeling

ABSTRACT

Modeling gas flow through fractures of subsurface rock is a particularly challenging problem because of the heterogeneous nature of the material. High-fidelity simulations using discrete fracture network (DFN) models are one methodology for predicting gas particle breakthrough times at the surface but are computationally demanding. We propose a Bayesian machine learning method that serves as an efficient surrogate model, or emulator, for these three-dimensional DFN simulations. Our model trains on a small quantity of simulation data with given statistical properties and, using a graph/path-based decomposition of the fracture network, rapidly predicts quantiles of the breakthrough time distribution on DFNs with those statistical properties. The approach, based on Gaussian Process Regression (GPR), outputs predictions that are within 20%–30% of high-fidelity DFN simulation results. Unlike previously proposed methods, it also provides uncertainty quantification, outputting confidence intervals that are essential given the uncertainty inherent in subsurface modeling. Our trained model runs within a fraction of a second, considerably faster than reduced-order models yielding comparable accuracy (Hyman et al., 2017; Karra et al., 2018) and multiple orders of magnitude faster than high-fidelity simulations.

1. Introduction

Modeling and predicting the passage of fluids through subsurface fracture networks is a fundamental and ongoing challenge in numerous civil, governmental, and industrial applications. These include the long-term storage of spent nuclear fuel, aquifer management and cleanup, enhanced geothermal energy systems, conventional/unconventional hydrocarbon extraction, geological sequestration of carbon, and the detection of chemical signatures from clandestine nuclear weapons tests (Selroos et al., 2002; Neuman, 2005; Follin et al., 2014; Sun and Carrigan, 2014; Jenkins et al., 2015; Middleton et al., 2015; Hyman et al., 2016b; Bourret et al., 2019; National Academies of Sciences, Engineering, and Medicine et al., 2020). There are a variety of computational methods for modeling gas transport through such systems, and specifically for obtaining the distributions of gas/solute travel times, i.e., the breakthrough curve. One distinguishing feature between the different modeling approaches is how, and with what level of fidelity, fractures and the surrounding host rock are represented (Hyman et al., 2022; Viswanathan et al., 2022).

There are continuum methods where the effects of fractures are accounted for using upscaled quantities, e.g., effective permeability and porosity, cf. Sweeney et al. (2020) for additional references and a more comprehensive description. However, continuum methods often poorly represent network connectivity, which is believed to be a key geostructural property that controls transport through fractured media (Hyman et al., 2020; Maillot et al., 2016). A result of upscaling the network structure into effective properties is that it becomes challenging to link transport observations with geostructural properties (Kang et al., 2020). In contrast, there are discrete fracture network (DFN) models where the individual fractures and the interconnected networks they form are explicitly represented. While the higher level of fidelity provides a more accurate representation of transport and the ability to link geostructural attributes with flow and transport observations (Hyman et al., 2019a,b), DFN simulations are more computationally expensive than continuum models. DFN models typically require unstructured computational mesh generation, have a large number of degrees of freedom, and use a computational physics simulator, all of which scale

[*] Corresponding author.
E-mail address: allon.percus@cgu.edu (A.G. Percus).

https://doi.org/10.1016/j.cageo.2024.105700
Received 3 June 2023; Received in revised form 7 August 2024; Accepted 8 August 2024
Available online 10 August 2024

with the spatio-temporal size of the simulations (Ushijima-Mwesigwa et al., 2021). Regardless of the adopted computational methodology, the large uncertainty inherent in subsurface modeling requires that ensembles of realizations are generated to obtain expected transport behavior along with confidence intervals of the breakthrough curves. This requirement exacerbates the computational demands of subsurface modeling and particularly DFN models.

A promising alternative to address these computational costs is the use of machine learning (ML) techniques that take geostructural information of the network, e.g., topological, geometric, and hydrological attributes, as inputs to provide estimates of flow and transport observations. Such surrogate models, or emulators, can be trained on a modest amount of high-fidelity simulation data to provide approximations of gas/solute breakthrough times. While generating the training data could still require many hours of simulation time for a given set of hydrological properties, these ML models, once trained, can generate predictions within a fraction of a second. Their predictions may not be as accurate as those of the direct simulations themselves, but owing to their speed, they allow for studying orders of magnitude more fracture networks (with a given set of geostructural properties) than would otherwise be possible using conventional computational physics models. This scalability, in turn, can provide a more comprehensive understanding of the underlying physical phenomena.

In our work, we aim to take advantage of ML-based acceleration in the context of nuclear treaty monitoring where one needs to predict gas migration quickly from underground nuclear explosions. In the DFN methodology, this requires running numerous models to obtain statistically stable estimates of a quantity of interest. Each DFN is generated randomly from an ensemble, with statistical properties representative of the geostructural attributes of the site under observation (Viswanathan et al., 2022). Given the computational cost of high-fidelity simulations, the use of emulators allows an efficient means of more fully exploring the space of network generation parameters consistent with the system's statistical properties. The emulator results therefore interpolate between simulation results, filling gaps in the parameter space that would be computationally unaffordable to fill using additional high-fidelity simulations.

There is an extensive body of literature on the use of emulators, also known as meta-models, for subsurface flow simulations. Thenon et al. (2016) study the use of multi-fidelity meta-models for reservoir engineering. Zhou et al. (2020) propose using deep learning for predicting THM (coupled thermo-hydro-mechanical) gas flow. Other researchers develop neural network emulators as surrogates for high-fidelity reactive transport models, for groundwater modeling (Thiros et al., 2022), and for carbonate reservoirs (Li et al., 2022). Focaccia et al. (2021) use Polynomial Chaos Expansion in order to approximate long-term forecasting in groundwater applications. These approaches focus on saturated flows. We are interested in diagnostic gas flow through subsurface fractured rock, where THMC (coupled thermo-hydro-mechanical–chemical processes) play an important role. These systems are heterogeneous, complex fracture networks, amenable to DFN modeling. Our contribution is the development of ML-based emulator models in the DFN setting.

While ML approaches have been used previously in this context, to identify primary flow subnetworks (Valera et al., 2018; Srinivasan et al., 2019, 2020), they have not been used to generate direct predictions of particle arrival times themselves. Furthermore, these techniques do not provide predictive uncertainties. Our work addresses both shortcomings. We use Gaussian Process Regression (GPR, also known as kriging) to obtain predictions for transport breakthrough times through fracture networks along with uncertainty quantification. Our method blends physics-based modeling with Bayesian inference to generate predictions with greater interpretability.

Specifically, we train our model on breakthrough curves (travel time distributions of particles transported along with the flow) generated using high-fidelity three-dimensional DFN simulations. These simulations are drawn from an ensemble of semi-generic DFNs based on commonly observed field site characteristics including hydrological properties. We characterize the networks using a graph representation to isolate features associated with simple source-to-target paths in the graph which are used as ML model inputs. The quantities of interest (QoI) predicted by the model, on other DFNs sharing the same hydrological properties, are quantiles of the breakthrough time distribution: notably the 0th (first arrival time), 20th, 50th, 70th, and 90th percentile, as well as the peak arrival time.

GPR, as a Bayesian method, outputs a full predictive distribution rather than simply a scalar prediction. Given a prior distribution of a Gaussian process, GPR uses the training data to update this prior and to generate a multivariate normal posterior distribution. GPR allows nonlinear fits with closed-form predictive uncertainties for the outputs. This helps address the crucial challenge in subsurface hydrology of quantifying the uncertainty in breakthrough curve predictions.

Our results are summarized as follows. On DFNs with several hundred fractures, our model trains within seconds and the trained model runs within a fraction of a second which is multiple orders of magnitude faster than the original simulation. We obtain predictions on breakthrough times that, depending on the quantile, are within 20%–30% of the values obtained from high-fidelity simulations. This is a high level of accuracy given the sparsity of experimental data on true subsurface fracture sites and the fact that DFNs themselves provide at best a statistical characterization of such data. Furthermore, our prediction quality is competitive with existing reduced-order models for breakthrough time prediction (Viswanathan et al., 2022), in particular an adaptation of a graph flow model (Karra et al., 2018) which obtains similar accuracy but takes approximately one minute (compared to a fraction of a second) to run due to the need to simulate particle flow. Finally, we obtain rigorous confidence bounds for our predictions. These results can impact numerous application areas, including gas seepage from underground nuclear explosions, natural gas extraction, and detection of methane leakage from wells.

2. Methods: Flow and transport in fracture networks

2.1. Three-dimensional discrete fracture networks

For the high-fidelity simulations used to generate our training data set, we adopt a Discrete Fracture Network (DFN) approach to model flow and transport through the fractured rock mass. In a DFN model, the fractures are represented as a network of intersecting planes whose sizes, shapes, orientations, and hydrological properties are sampled from distributions whose parameters are determined from a site characterization, cf. Viswanathan et al. (2022) for a comprehensive discussion of DFN modeling approaches. We use the dfnWorks software (Hyman et al., 2015a) which provides an end-to-end workflow from network generation to flow and transport simulation. We consider an ensemble of semi-generic DFNs generated in a cubic domain with sides of length $L = 100$ m. The networks are semi-generic in that they do not represent a particular field site, but the characteristics are loosely based on field observations (Bonnet et al., 2001). Within the DFN, there is one fracture family, with radii r drawn from a truncated power law distribution with a decay exponent of 1.5, and lower and upper cutoffs of 1 and 50 m. Fracture centers are uniformly distributed throughout the domain and their orientations follow a uniform distribution projected onto the unit sphere, which mimics a disordered media (Hyman and Jiménez-Martínez, 2017). These parameters ensure that there is no single fracture connecting inflow and outflow boundaries. Initially, 2000 fractures are placed in the domain. Isolated fractures and clusters that do not connect the inflow and outflow boundaries are removed because they do not contribute to flow. The resulting fracture networks contain around 300 fractures each. The hydraulic aperture of each fracture is positively correlated with the radius using a power-law relationship, $b = 5 \cdot 10^{-5} \sqrt{r}$. Such correlations are a common modeling assumption for a DFN model (de Dreuzy et al., 2002; Frampton and Cvetkovic, 2010; Hyman et al., 2016a; Joyce et al., 2014). This correlation leads to hydraulic variability as well as the geostructural variability of the network.

Next, we create a computational mesh representation of networks on which to simulate flow and transport. The mesh is a conforming Delaunay triangulation produced using the Feature Rejection algorithm for meshing (FRAM) combined with the near-maximal Poisson sampling method (nMAPS), which are described in Hyman et al. (2014) and Krotz et al. (2022). We simulate steady-state laminar flow on the dual mesh of the triangulation, the Voronoi control volumes, using the massively parallel flow and transport solver PFLOTRAN (Lichtner et al., 2020). The distribution of volumetric flow rates and pressure within the network is modeled using Darcy's Law

$$\mathbf{q} = -\frac{\kappa}{\mu}\nabla P \tag{1}$$

where $\mathbf{q}$ is the volumetric flux with explicit units of [m^3/(m^2 s)], κ is the permeability [m^2], μ is the fluid viscosity [Pa s], P is the fluid pressure [Pa], and ∇P is the pressure gradient [Pa/m]. Simulations are performed with the fluid temperature at 20 °C, which corresponds to a viscosity of $\mu = 8.9 \cdot 10^{-4}$ Pa s for the pressure values considered. Fracture permeability is determined using a local cubic law, $\kappa = b^2/12$. We drive flow through the domain by applying a pressure difference of 1 MPa across the x-axis using Dirichlet conditions on the inflow and outflow boundaries. Note that Eq. (1) is linear in the pressure gradient, and thus the value of 1 MPa is arbitrary with respect to the structure of the flow field within the network. Neumann, no-flow, boundary conditions are applied along lateral boundaries of the domain as well as along fracture boundaries. Gravity is not included in these simulations. In the DFN model, the matrix surrounding the fractures is impermeable, i.e., there is no interaction between flow within the fractures and the solid matrix. The numerical solution of Eq. (1) provides values for the pressure and volumetric flow rates throughout the domain that are used to reconstruct the Eulerian velocity field $\mathbf{u}(\mathbf{x})$ within the DFN using the method provided in Makedonska et al. (2015) and Painter et al. (2012).

Transport through the network is simulated using tracer particles that follow pathlines through the velocity field $\mathbf{u}(\mathbf{x})$. Particles are distributed along the inflow plane of the domain using flux-weighting so that the number of particles at a location is proportional to the incoming volumetric flow rate (Hyman et al., 2015b; Hyman and Dentz, 2021; Kang et al., 2017; Kreft and Zuber, 1978). The trajectory $\mathbf{x}(t;\mathbf{a})$ of a particle starting at a point $\mathbf{a}$ at time $t = 0$ is given by integrating the advection equation

$$\frac{d\mathbf{x}(t;\mathbf{a})}{dt} = \mathbf{v}_t(t;\mathbf{a}), \qquad \mathbf{x}(0;\mathbf{a}) = \mathbf{a}. \tag{2}$$

Here, the Lagrangian velocity $\mathbf{v}_t(t;\mathbf{a})$ is given in terms of the Eulerian velocity $\mathbf{u}(\mathbf{x})$ as $\mathbf{v}_t(t;\mathbf{a}) = \mathbf{u}[\mathbf{x}(t;\mathbf{a})]$. The dynamics occurring within intersections are a sub-grid scale process represented using a complete mixing model so that the probability of a particle exiting onto a fracture is proportional to the outgoing volumetric flow rate. The numerical method implemented is described in Sherman et al. (2019).

We record the arrival time of each particle to exit the domain having traveled a linear distance of x_L, which we denote as $\tau(x_L;\mathbf{a})$. These values are used to construct the relative solute breakthrough (probability density function) at a time t defined as

$$\psi(t, x_L) = \frac{1}{M}\int_{\Omega_a} d\mathbf{a}\ \delta[\tau(x_L,\mathbf{a}) - t], \tag{3}$$

where $\delta(t)$ is the Dirac delta function. We also compute the cumulative solute breakthrough (cumulative density function) as

$$\Psi(t, x_L) = \frac{1}{M}\int_{\Omega_a} d\mathbf{a}\ H[\tau(x_L,\mathbf{a}) - t], \tag{4}$$

where $H(t)$ is the Heaviside step function. In Eqs. (3) and (4), M is the total number of particles tracked through the domain, which is set to ten thousand in these simulations. A preliminary study showed that these primary observables were not dramatically affected by increasing the number of particles beyond this value. A single run using 32 cores takes around ≈1–2 h depending in the number of fractures in the network.

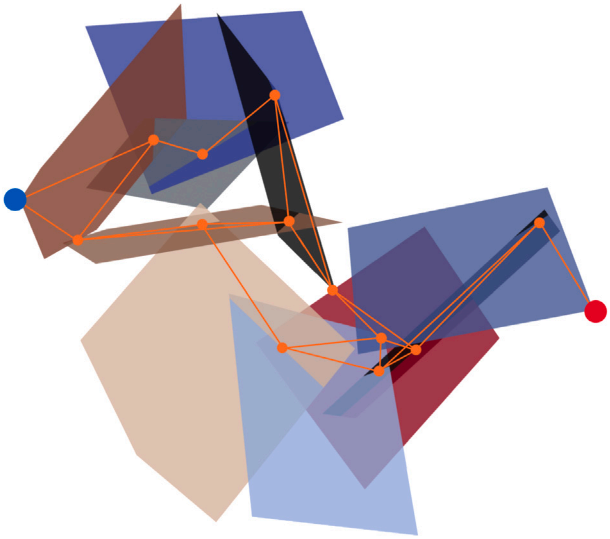

Fig. 1. DFN with planes representing fractures, overlaid with its associated intersection graph. The blue vertex on the left denotes the source (inlet) and the red vertex on the right denotes the target (outlet).

2.2. Graph representation

At the center of the DFN methodology is the conceptual model that a set of fractures, which are discrete entities, intersect one another to form a network. This conceptual model may also be represented using the mathematical construct of a graph $G = (V, E)$, which is a tuple consisting of a vertex set V and an edge set E containing pairs of vertices connected by an edge. There are a variety of graph-based representations of a DFN. There is the canonical representation where vertices are fractures and edges are intersections (Andresen et al., 2013; Hope et al., 2015; Huseby et al., 1997; Hyman et al., 2017), which can be readily populated with particle information to create a flow topology graph (Aldrich et al., 2017) as well as a bipartite graph with one vertex set of fractures and another vertex set of intersections (Hyman et al., 2018). We adopt a representation where nodes correspond to intersections between fractures V, and two vertices are connected by an edge in E if the associated intersections are on the same fracture. We refer to this as the *intersection graph* representation. The precise mathematical formalization and relationship with other representations are presented in Hyman et al. (2018). Fig. 1 shows a small DFN, not one used in the simulations, along with the intersection graph representation as an exposition. Additional source and target vertices are added to represent an inlet plane and an outlet plane, thereby providing an orientation of the graph in topological space. Since a fracture can have multiple intersections, it may be represented not only by a single edge but by a clique (fully connected subgraph) whose edges represent all pairs of intersections. Additionally, flow and transport can be resolved on the graph representations using the same governing equations provided above (Dershowitz and Fidelibus, 1999), cf. the graph flow models of Karra et al. (2018) and Doolaeghe et al. (2020) for examples. For this reason, the intersection graph has been called a pipe or channel network representation of a fracture network (Dershowitz and Fidelibus, 1999). In turn, approximations of volumetric flux and travel times on edges within the graph can be readily obtained.

3. Gaussian process regression

Consider a collection of N DFN simulations. For the ith sample, let G_i be its intersection graph representation as defined above, y_i be its high-fidelity QoI (appropriate quantile of breakthrough time distribution), and x_i be features extracted from the graph G_i. Our objective is to learn a surrogate model $y_i = f(x_i)$ which allows for accurate and fast prediction of high-fidelity breakthrough times, directly as a function of the low-fidelity graph embedding, without resorting to expensive transport simulations. A commonly used class of surrogate models for computer simulators is Gaussian Process Regression (GPR), a flexible nonparametric regression model. GPR is well-suited for this application for various reasons: (i) it allows nonlinear fits of varying degrees of smoothness; (ii) minimal assumptions *a priori* are required about the nonlinear functional form; (iii) predictions are equipped with closed-form, rigorous confidence bounds, allowing for uncertainty quantification; (iv) using modest amounts of training data, it can be trained and used for prediction with reasonable computational speed. Other regression methods, such as Bayesian linear regression and support vector machines, tend to pre-specify the functional form, require stronger assumptions, or necessitate approximate methods for uncertainty quantification (Bishop, 2006).

A Gaussian process, denoted $f(x) \sim GP\left(\mu(x), \Sigma(x, x')\right)$ with mean function $\mu(x)$ and covariance function $\Sigma(x, x')$, is a distribution over functions such that any finite set of function evaluations at points $\boldsymbol{x} = (x_1, \dots, x_n)$ is multivariate normal (Rasmussen and Williams, 2006):

$$f(\boldsymbol{x}) \sim N\left(\mu(\boldsymbol{x}), \Sigma(\boldsymbol{x}, \boldsymbol{x})\right). \tag{5}$$

It is common to set $\mu(x) = 0$ *a priori*, thus focusing model specification on the covariance function $\Sigma(x, x')$. This governs the smoothness of $f(x)$, as it characterizes the correlation between two outputs as a function of the similarity between their corresponding inputs. In GPR, Σ is typically specified by a kernel $K(x, x')$ belonging to a chosen family. A common choice is a radial basis function (RBF) kernel,

$$K(x, x') = \exp\left(-\frac{1}{2\gamma^2}\|x - x'\|^2\right), \tag{6}$$

which is infinitely differentiable, thus producing smooth functions $f(x)$. Other families, like the Matérn class of kernels, can produce less smooth fits, rendering them potentially suitable for hydrogeological simulations. However, as we note in Section 5 below, we did not find any major differences between the results from GPR models fit using RBF and Matérn kernels.

For model inputs which are not deterministically related to outputs, it is common to add an additional white noise term with variance σ_ϵ^2. The full covariance matrix evaluated at points $\boldsymbol{x}$ is given by: $\Sigma(\boldsymbol{x}, \boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{x}) + \sigma_\epsilon^2 I_n$. Its hyperparameters can be estimated (Rasmussen and Williams, 2006) by numerically maximizing the log marginal likelihood of the observed data,

$$\max_{\gamma, \sigma_\epsilon} \log p(\boldsymbol{y}|\boldsymbol{x}, \gamma, \sigma_\epsilon) = \max_{\gamma, \sigma_\epsilon}\left(-\frac{1}{2}\left[\boldsymbol{y}^T\Sigma^{-1}(\boldsymbol{x}, \boldsymbol{x})\boldsymbol{y} + \log\left(\det \Sigma(\boldsymbol{x}, \boldsymbol{x})\right) + n\log 2\pi\right]\right). \tag{7}$$

A benefit of GPR is that its predictive distribution is normal. That is, for any arbitrary input x^*, its predicted surrogate output $f(x^*) \sim N\left(\mu(x^*), \sigma^2(x^*)\right)$ where:

$$\mu(x^*) = \Sigma_*(x^*, \boldsymbol{x})\Sigma^{-1}(\boldsymbol{x}, \boldsymbol{x})\boldsymbol{y}, \tag{8}$$

$$\sigma^2(x^*) = \Sigma_{**}(x^*, x^*) - \Sigma_*(x^*, \boldsymbol{x})\Sigma^{-1}(\boldsymbol{x}, \boldsymbol{x})\left[\Sigma_*(x^*, \boldsymbol{x})\right]^T, \tag{9}$$

for $\Sigma_*(x^*, \boldsymbol{x}) = \left[K(x_*, x_1), \dots, K(x_*, x_n)\right]$ and $\Sigma_{**}(x^*, x^*) = K(x_*, x_*)$. This enables fast prediction once the GPR is trained (making it a desirable candidate for surrogates of expensive computer models), while also facilitating estimates of predictive uncertainty.

4. Data

In this section, we describe our QoI data and the features that we construct as inputs to GPR. We train and test the model using a dataset $\mathcal{G}_1$ which contains 100 DFNs. For each one of the graphs $G \in \mathcal{G}_1$, a breakthrough curve has been generated using the 3D DFN model. In addition to $\mathcal{G}_1$, we have a second dataset $\mathcal{G}_2$ of 100 DFNs, generated using the same statistical generation properties. We use this second set only for feature tuning.

4.1. DFN - Breakthrough curves

Using dfnWorks, we generate a statistical ensemble of random DFNs, all with the same hydrological properties, for each of the two datasets and obtain cumulative particle breakthrough densities (Eq. (4)). We record quantiles that span much of the breakthrough time distribution, notably 0% (first arrival time), 20%, 50%, 70%, and 90%, as well as the peak arrival time. These form the target quantities of interest (QoI) for our ML model predictions.

GPR functions most effectively when applied to data with a Gaussian distribution. Fig. 2 demonstrates that the QoI is heavily skewed with a median breakthrough time (BTT) displaying a distribution that is closer to log-normal than to normal. We therefore take the log of this quantity, whose distribution is closer to normal (apart from an extended right-hand tail). We use the log of the BTT quantiles as the output of our model for training and prediction purposes.

4.2. Model input

Many methods have been developed to extract features from graphs for machine learning (Stamile et al., 2021). They can be grouped into a number of categories such as shallow embedding methods, regularization methods, and graph neural networks (Chami et al., 2021). As an alternative, we adopt the physics-informed approach presented by Srinivasan et al. (2020): in order to extract features that describe the flow through the graph, we consider features associated with physical information on topological paths within the graph that connect the source to the target node. This approach is well-suited to characterizing particle breakthrough times as it captures attributes that most influence transport across the network.

To form an input data point x_i to our surrogate GPR model $f(x)$ in a given DFN, we take a collection of source-to-target paths along with the set of features associated with each path. Given k possible paths and l features for each path, we sum the value of each feature over all k paths, resulting in a model input $x_i \in \mathbb{R}^l$. Following Srinivasan et al. (2020), we identify the k source-to-target simple paths that are shortest in terms of graph distance on the intersection graph. However, rather than taking a fixed value of k as in previous work, we leave it as a tunable quantity in our feature selection (see Section 4.3 below).

We fix $l = 5$ and select the five path-associated features that are described below and illustrated schematically in Fig. 3. While better features may well exist, these demonstrate how, empirically, a small number of interpretable physical quantities can describe network flow well enough to yield useful predictions. Our choice of features is motivated primarily by quantities used in the graph flow model presented by Karra et al. (2018), as mentioned in Section 2.2 above. They differ from the graph centrality-based features that were found to be most impactful in the identification of primary flow subnetworks (Valera et al., 2018; Srinivasan et al., 2019, 2020) but did not perform as well for the direct prediction of breakthrough times, as we have confirmed through a forward-selection process.

4.2.1. Path length

Path length is the number of distinct fractures involved in the path from the source to target node in the network. In the intersection graph representation, this is the number of edges along the path. For the two

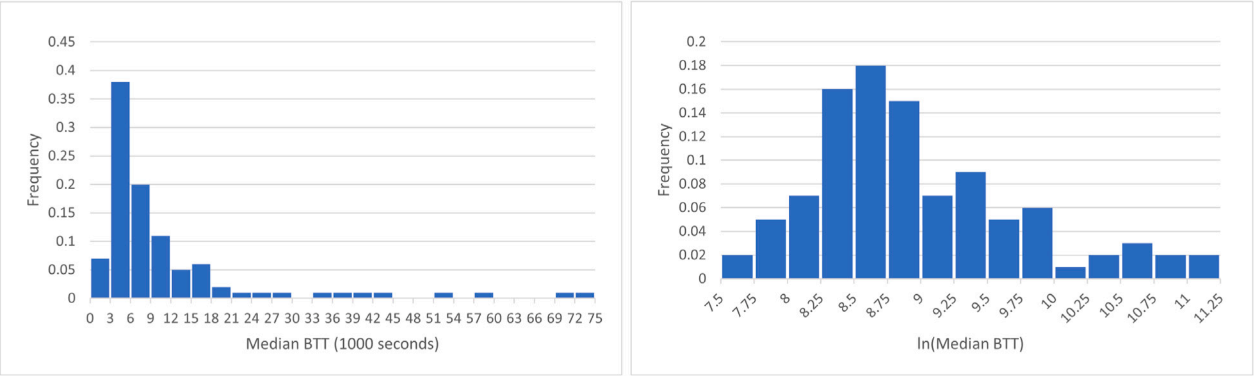

Fig. 2. Median BTT, over the 100 DFNs in the training and testing dataset $\mathcal{G}_1$. The raw times (left) have a distribution that is closer to log-normal than to normal. Taking the log of these values (right) transforms the data so that they are closer to normal.

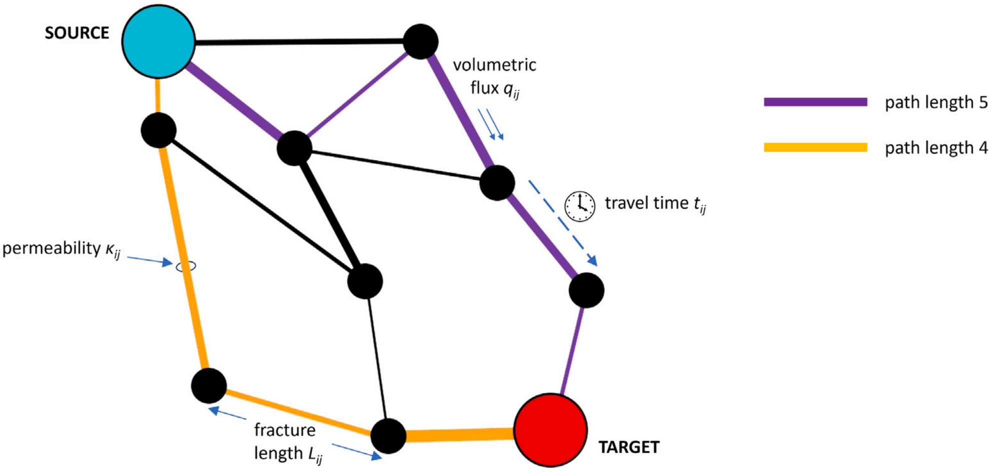

Fig. 3. Schematic representation of input features for GPR, defined on source-to-target paths in the intersection graph (two paths shown, in purple above and in yellow below) or on edges (fractures) along the path.

paths highlighted in Fig. 3, the path lengths are 5 (upper path in purple) and 4 (lower path in yellow).

4.2.2. *Inverse permeability:* r_{ij}

The permeability κ_{ij} of the fracture connecting intersections i and j is a measure of conductance, represented schematically in Fig. 3 by edge width. The inverse of conductance is a measure of resistance. In a resistive network, the resistance of two components in series is the sum of the resistances of the two components. By analogy with this, we consider the inverse permeability of a fracture,

$$r_{ij} = \frac{1}{\kappa_{ij}}, \tag{10}$$

to be a quantity whose sum over a sequential path of fractures has physical significance. We take the value of the inverse permeability feature on a path to be the sum of r_{ij} among all edges (fractures) on that path.

Note that the closely related arithmetic mean of the inverse permeabilities is equal to the reciprocal of the harmonic mean of permeabilities along the path. The latter quantity is the equivalent permeability for transporting, across the entire path, the same flux under the same pressure gradient.

4.2.3. *Fracture length:* L_{ij}

Fracture length is the Euclidean distance L_{ij} between the centroids of intersections i and j, representing the distance that a particle travels on the fracture $\{i, j\}$. This is illustrated in Fig. 3 by the length of an edge between two vertices. We take the value of the fracture length feature on a path to be the sum of L_{ij} among all edges (fractures) on that path, representing the shortest distance a particle can travel on it from source to target.

4.2.4. *Volumetric flux:* q_{ij}

From Darcy's Law (Eq. (1)), the volumetric flux on an edge is

$$q_{ij} = \frac{\kappa_{ij}}{\mu L_{ij}}(P_i - P_j), \tag{11}$$

where μ is the fluid viscosity and P_i, P_j are pressures at nodes i and j, calculated using the method described in Karra et al. (2018). This is a measure of particle flow along the edge, represented schematically in Fig. 3 by a double arrow. We take the value of the volumetric flux feature on a path to be the median of q_{ij} among all edges (fractures) on that path. The median is chosen to minimize sensitivity to outlier values along the path; through forward feature selection, we found that it outperformed the use of mean, sum, minimum, and maximum over the path.

4.2.5. *Travel time:* t_{ij}

Edge travel time is denoted as

$$t_{ij} = \frac{L_{ij}\phi_{ij}}{q_{ij}}, \tag{12}$$

where ϕ_{ij} is the porosity of the fracture $\{i, j\}$ taken to be 1 here indicating a completely open fracture. This is a measure of the time for a particle to travel along an edge, illustrated schematically in Fig. 3 by a

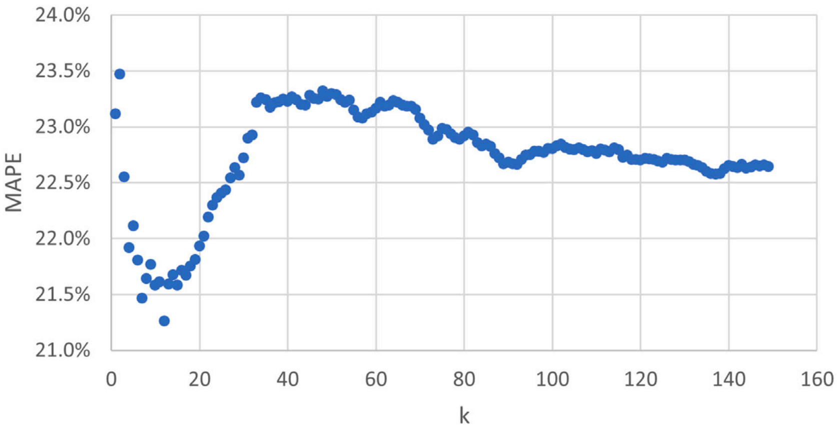

Fig. 4. Mean absolute percent error (MAPE) of predicted median BTT, over the 100 DFNs in the feature tuning dataset $\mathcal{G}_2$, when k shortest source-to-target paths are used in feature construction.

clock and dashed arrow. We take the travel time feature on a path to be the median of t_{ij} among all edges (fractures) on that path. While travel time along an edge is clearly not independent of fracture length and volumetric flux, the use of median distinguishes it qualitatively from the fracture length feature above which instead involves a sum. Here, as with the volumetric flux feature, the choice of median was further validated by forward feature selection.

4.3. Feature tuning

While our GPR model does not require any hyperparameter tuning, there is one tunable quantity in our feature construction: the number of shortest source-to-target paths, k. We find the value of k that optimizes model prediction quality by evaluating the absolute percent error of the predicted median BTT on the 100 DFNs in our second dataset, $G \in \mathcal{G}_2$, using 10-fold cross-validation. Implementation details for our GPR predictions are described in the Results section below. In order to avoid data leakage, we use $\mathcal{G}_2$ only for tuning k and not for any subsequent training or testing.

Fig. 4 shows the mean absolute percent error (MAPE), over these 100 DFNs, of the predicted median BTT. We find that, as a function of k, the MAPE has relatively smooth behavior, with values reaching a minimum at smaller k values as more paths are considered. As k increases further, the inclusion of paths less relevant to particle flow appears to dilute the features, causing MAPE to increase and ultimately stabilize at large k. While the variations are not large (within 2%), we use the numerical minimum $k = 12$ as our optimized value.

Note that, while we optimize k on the basis of the median BTT alone, we use this same tuned value of $k = 12$ in the input features for predicting all of our QoI. It may seem that allowing additional source-to-target paths (larger k) could improve performance for larger quantiles of the breakthrough curve such as the 70th and 90th percentile, and that the reverse could hold for smaller quantiles such as the 0th and 20th percentiles. Empirically, however, we have not found that this gives consistently better predictive performance.

5. Results

We implement Gaussian Process Regression (GPR) using the Python *scikit-learn* function `GaussianProcessRegressor`. We choose an RBF plus white noise kernel, as described in Section 3, and set all other parameters to default values. In order to test whether our results are sensitive to the choice of kernel, we also implement a Matérn kernel with parameter $\nu = 3/2$ (also with white noise), which is far less smooth than RBF. As will be seen in Tables 1 and 3, this does not appear to have an important or systematic effect on model output. Where not explicitly noted, results given below use the RBF kernel.

Table 1
Mean Absolute Percent Error (MAPE) of model predictions for different BTT quantiles and peak BTT, over the 100 DFNs in the dataset $\mathcal{G}_1$ using 10-fold cross-validation. 0th percentile represents first arrival time of gas particles. For completeness, results are shown with GPR predictions using both the RBF kernel and the Matérn kernel, although neither one offers a clear improvement over the other.

BTT percentile	MAPE (RBF)	MAPE (Matérn)
0th	20.53%	19.34%
20th	22.53%	21.02%
50th (median)	21.80%	21.75%
70th	23.29%	24.65%
90th	30.24%	31.43%
Peak	26.93%	26.71%

5.1. Model predictions

We train and test the GPR model on the 100 DFNs in the first dataset $\mathcal{G}_1$ using 10-fold cross-validation. Recall that our QoI are quantiles of these breakthrough curves, including the 0th (first arrival time), 20th, 50th, 70th, and 90th percentile, as well as the peak arrival time. Fig. 5 shows the percent errors of the model predictions of the 50th percentile (median BTT) over this dataset. Inputs to GPR are standardized, resulting in a symmetric distribution (left) of percent errors in the logarithmic quantity that the model outputs. This distribution becomes asymmetric (right) when exponentiating the model output to recover one of our QoI, the raw median BTT. The model predicts this quantity with an error between −47% and 77%. Fig. 6 shows the corresponding *absolute* percent error in the predicted median BTT. For a majority of DFNs, the error is less than 20%, exceeding 50% in only 8 out of 100 cases.

Table 1 expands our comparisons to the other BTT quantiles and peak BTT, showing the mean absolute percent error (MAPE) of model predictions. Values range from approximately 20% (for 0th percentile or first arrival time) to 30% (for 90th percentile). For completeness, here we give results using both the RBF and Matérn kernels in GPR, although there are no major differences between them. These numerical results are broadly comparable to the most accurate BTT predictions obtained from reduced-order models (Viswanathan et al., 2022), namely using the graph flow model of Karra et al. (2018) mentioned in Section 2.2. By applying a bias-correcting postprocessing step based on simple linear regression to that method, using the same training data and 10-fold cross-validation approach as for GPR, we obtain an MAPE of 19.77% for median BTT, compared to 21.80% from our GPR model. However, once trained, GPR runs in a fraction of a second rather than the ≈1 min required for simulating transport in the graph flow model.

To put these run times in context, recall that any given DFN is simply a random realization of an ensemble of fracture networks with a given set of hydrological properties. Owing to statistical variations,

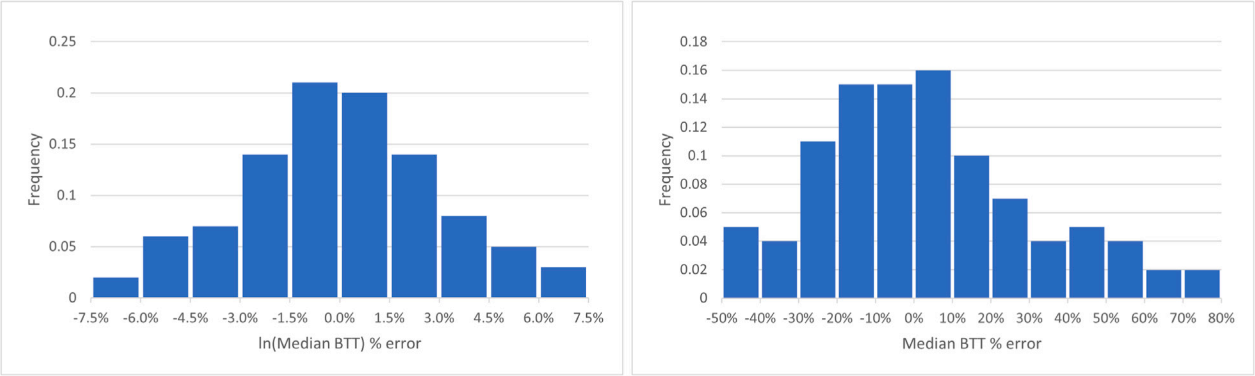

Fig. 5. Percent error of model predictions for median BTT, over the 100 DFNs in the dataset $\mathcal{G}_1$ using 10-fold cross-validation. Note that distribution is symmetric for predictions of the log of the median BTT (left) and asymmetric when transforming these back to raw median BTT (right).

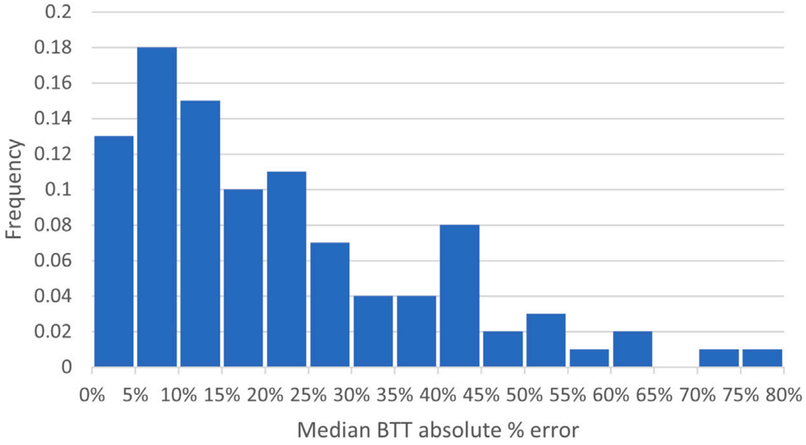

Fig. 6. Absolute percent error of model predictions for median BTT, over the 100 DFNs in the dataset $\mathcal{G}_1$ using 10-fold cross-validation.

Table 2
Approximate run times and MAPE for median BTT predictions on 10^4 DFNs using high-fidelity simulations (Hyman et al., 2015a), bias-corrected graph flow (Karra et al., 2018), and GPR. MAPE is with respect to high-fidelity simulations. For graph flow bias correction and for GPR, we account for the time of generating a training set of 100 DFN high-fidelity simulations.

Method	Total run time	MAPE
High-fidelity	5×10^7 s	–
Graph flow	5×10^5 s (training) + 6×10^5 s (model runs)	19.77%
GPR	5×10^5 s (training) + 4.6×10^2 s (model runs)	21.80%

exploring the space of network generation parameters consistent with these properties can require running a large number of DFN models. On such sample sizes, the difference in timings among the different methods can be considerable, even with the large amount of time required to simulate training data for ML models. Table 2 shows a comparison of the approximate CPU time needed to generate BTT predictions on 10^4 DFNs using high-fidelity simulations, the graph flow model, and the GPR model, along with their associated MAPE. The run times are based on 5×10^3 s (about 1.5 h) to run a high-fidelity simulation, 6×10^1 s (1 min) to run the graph flow model, and 4.6×10^{-2} s to run the GPR model, on a single processor with 32 cores. The run times further assume that both graph flow bias correction and GPR use a training set of 100 high-fidelity simulations. For such sizes, the time to train these two models is overwhelmed by the time to generate the training data, even though the training complexity of GPR scales as $O(N^3)$ for N training points (Rasmussen and Williams, 2006). As shown in Table 2, after accounting for the large data generation overhead, GPR still remains approximately two orders of magnitude faster than high-fidelity simulations and is twice as fast as graph flow.

Finally, we note that expanding the size of our training set to more than 100 DFNs does not noticeably improve prediction quality. For instance, increasing the training sample to 200 DFNs results in only a very modest decrease in the MAPE for median BTT, from 21.80% to 21.64%. Conversely, when the size of the training set is decreased, performance does degrade. Switching from a 10-fold cross-validation framework to 4-fold cross-validation, which decreases the number of training DFNs used in a fold from 90 to 75, increases the MAPE for median BTT from 21.80% to 22.59%. These observations suggest that prediction quality saturates at approximately the training sample size that we use for the results above, and that our specific choice of features may be a more important limitation than the quantity of training data available.

5.2. Uncertainty quantification

A further advantage of the GPR model is that it outputs not only a value but a predictive distribution which provides rigorous confidence bounds on predictions. In order to illustrate the effectiveness of the

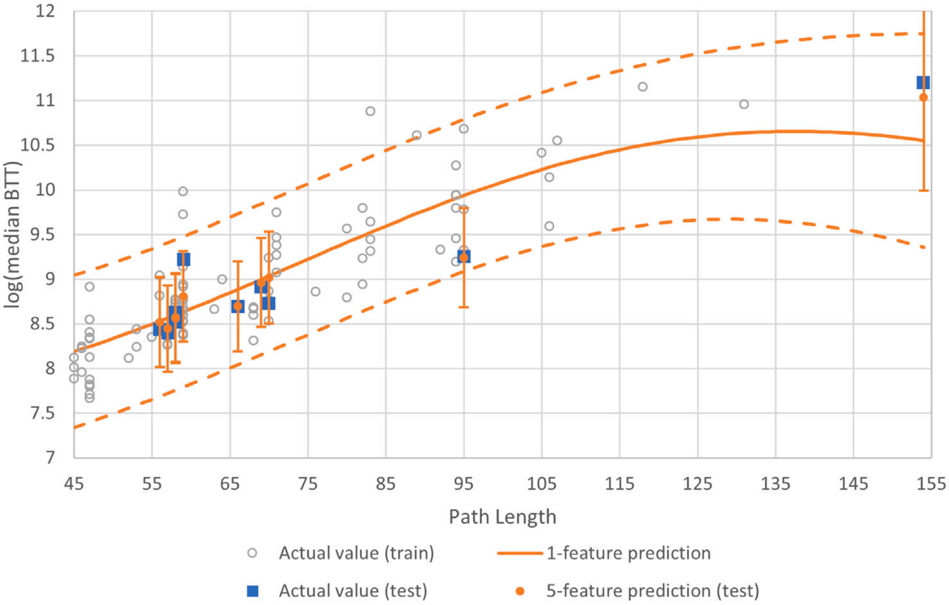

Fig. 7. Results from simplified GPR model trained only on 1 feature (path length), along with 95% confidence bounds, training data points, test data points, and 5-feature predictions (with full GPR model) on test data points. Gray circles show 90 DFNs from dataset $\mathcal{G}_1$ used to train GPR model. Orange solid line shows 1-feature predictions varying smoothly with model input (path length). Orange dashed lines show associated 95% confidence bounds. Blue squares show 10 DFNs from dataset $\mathcal{G}_1$ used to test GPR model. Orange points show 5-feature predictions on test DFNs, along with bars showing 95% confidence bounds for those predictions.

method on our problem, we first show the results on a simplified version of it. Fig. 7 displays predictions for the log of the median BTT when the model is trained using data described by only one feature, the path length. This allows us to represent, on a 2D plot, a smooth curve showing the model output as that one feature is varied, along with the predictive interval delimited by 95% confidence bounds (dashed lines). In this example, the training data consist of 90 fixed DFNs from dataset $\mathcal{G}_1$, denoted by gray circles, and the test data consist of the remaining 10 DFNs, denoted by blue squares. Finally, orange points represent predictions given on the test data by the full 5-feature model, with error bar whiskers representing 95% confidence bounds for those predictions.

A few properties are notable in the results in Fig. 7. Qualitatively, the predictive interval widens as expected at large path lengths, where training data is sparser and the model is less confident in its predictions. Quantitatively, the model correctly learns confidence intervals from the training data, with approximately 95% of those data points (gray) contained within the 95% confidence bounds for the 1-feature prediction. This effect is validated on test data, again with approximately 95% of those data points (blue) contained within the 95% confidence bounds for the 5-feature prediction. Not surprisingly, these confidence bounds are considerably tighter than those for the 1-feature prediction, demonstrating the decrease in uncertainty from including the four additional features.

Table 3 summarizes uncertainty quantification results for all of our QoI. At a given quantile, and for each DFN, we consider the log BTT predicted by the model along with its associated 95% confidence interval. We express the width of this confidence interval as a percentage of the log BTT and average that quantity over all 100 DFNs in the dataset $\mathcal{G}_1$. This relative confidence interval width varies from 12.30% of the predicted log BTT (for 50th percentile, i.e., median) to 15.43% of the predicted log BTT (for peak breakthrough). Note, though, that relative confidence intervals widths are considerably larger (and asymmetric) when expressed as a percentage of the raw BTT, as one might expect given the MAPE values of 20% to 30%.

Table 3 also shows the confidence interval coverage: the fraction of actual values that are within the 95% confidence bounds of the prediction. We show these values for results from both the RBF and Matérn kernel. They are all close to 0.95, consistent with the model correctly learning confidence intervals over all quantiles.

Table 3
% Width of CI represents relative width of predicted confidence interval, expressed as percentage of predicted log BTT, and averaged over the 100 DFNs in the dataset $\mathcal{G}_1$ using 10-fold cross-validation. CI coverage is the fraction of actual values that are within the 95% confidence bounds of the prediction. Results are shown for different BTT quantiles and peak BTT. For completeness, we show CI coverage values not only from the RBF kernel but also from the Matérn kernel, although there are no major differences between those values.

BTT percentile	% Width of CI	CI coverage (RBF)	CI coverage (Matérn)
0th	12.76%	0.92	0.91
20th	12.93%	0.97	0.95
50th (median)	12.30%	0.93	0.93
70th	13.11%	0.95	0.91
90th	14.84%	0.93	0.95
Peak	15.43%	0.96	0.95

6. Discussion and conclusions

Predicting gas breakthrough times in fracture networks is a crucial scientific challenge for which high-fidelity DFN simulation methods are computationally expensive. In this paper, we have introduced a Bayesian machine learning approach that uses Gaussian Process Regression (GPR) as an emulator for these simulations, providing rapid predictions of quantiles of the breakthrough time distribution along with confidence intervals that are consistent with the simulation data. Our model is trained on a modest amount of high-fidelity simulation data, using a combination of topological and geological attributes of the DFNs as features. The ensemble we consider is composed of synthetic semi-generic DFNs. They are semi-generic in that they do not represent a particular field site, but the characteristics are loosely based on field observations. Likewise, the Dirichlet pressure boundary conditions are common in the field. Thus, the results here are rather general in the sense that the methodology is not limited to a specific site (Viswanathan et al., 2022).

Once trained on simulations representing a given set of statistical (hydrological/structural) properties, our method generates results within a fraction of a second for different DFNs with those properties. Even considering the significant time needed to generate training data, this method can be an attractive choice for modeling subsurface systems where uncertainties in hydrological properties require a large-scale ensemble approach. Our results show that the model's predictions are within 20%–30% of those from high-fidelity simulations. This is competitive in terms of accuracy with existing reduced-order models for breakthrough time prediction, which run many times slower than our model, and is of sufficient quality to impact application areas ranging from extraction of hydrocarbons to detection of underground nuclear explosions (Viswanathan et al., 2022). Moreover, our Bayesian approach provides rigorous uncertainty quantification, offering analytically tractable confidence bounds. These are invaluable given the modeling uncertainties in subsurface hydrology and are essential to interpreting model predictions.

It is important to note certain limitations of our approach. First of all, we would expect a trained model to perform well only on DFNs generated with the same overall statistical properties as those it is trained on. If those properties change, the model would need to be retrained. However, if the changes are not too large, it may be possible to implement a separate "discrepancy" model that corrects BTT predictions according to DFN generation parameters, analogous to the bias correction that we learn for the graph flow model. This discrepancy model could be an independent GPR that is stacked on top of our BTT prediction model, essentially correcting the original predictions to reflect the change in DFN properties.

An additional limitation of our approach is that the raw accuracy of our GPR model remains slightly inferior to that of the corrected graph flow model. It is possible that some improvements to our input features could result in improvements in prediction accuracy. For both the volumetric flux and travel time, instead of using the median value along a path, we can set a different quantile as a tunable quantity in our feature construction. Currently, our only such tunable quantity is k, the number of shortest source-to-target paths used in feature construction. In principle, there is no difficulty in optimizing multiple quantities in feature selection, using a grid search, over the same separate dataset $\mathcal{G}_2$ that we now use to tune k alone. One might also consider the reverse approach of eliminating the volumetric flux and travel time features altogether, sacrificing accuracy for speed, as the calculation of pressures (required for volumetric flux and travel time) dominates both the training and the prediction time in our current model. For networks of several hundred fractures, eliminating that bottleneck could speed up the process by several orders of magnitude. The results in Fig. 7, obtained from running GPR with only the path length feature, suggest that such an approach may be viable. A further possible improvement concerns our choice of kernel function for GPR. While using a Matérn kernel with a smoothness parameter of $\nu = 3/2$ (a standard choice) has not led to noticeably different results, one can also learn an optimal value of the parameter from the data as well as adopt an isotropic kernel that attributes different sensitivities to different features. These could lead to improved predictions.

Finally, work is currently underway on generalizing our GPR method to a multi-fidelity setting, analogous to the co-kriging meta-modeling approach that has been proposed by Thenon et al. (2016) for the case of reservoir engineering. We have seen that graph-based algorithms can be used to supply low-fidelity BTT predictions. While such methods are slower than GPR, they are orders of magnitude faster than high-fidelity DFN simulations and could therefore serve as a source of less expensive (and consequently more abundant) training data. Preliminary results suggest that, with a multi-fidelity approach based on either an autoregressive training scheme or a linear Gaussian network, one can obtain results of comparable quality to those in this paper while using far less high-fidelity training data. This research is ongoing.

CRediT authorship contribution statement

Yingqi Shi: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Donald J. Berry:** Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **John Kath:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Shams Lodhy:** Writing – original draft, Investigation, Data curation. **An Ly:** Writing – review & editing, Writing – original draft, Visualization, Validation, Investigation, Data curation. **Allon G. Percus:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Jeffrey D. Hyman:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Kelly Moran:** Writing – review & editing, Resources, Formal analysis, Conceptualization. **Justin Strait:** Writing – review & editing, Supervision, Software, Resources, Methodology, Formal analysis, Conceptualization. **Matthew R. Sweeney:** Resources, Investigation, Data curation, Conceptualization. **Hari S. Viswanathan:** Writing – review & editing, Supervision, Project administration, Methodology, Conceptualization. **Philip H. Stauffer:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

The authors thank the anonymous reviewers, whose comments and suggestions have improved the quality of the manuscript. We acknowledge support from Los Alamos National Laboratory LDRD Award 20220019DR. Los Alamos National Laboratory is operated by Triad National Security, LLC, for the National Nuclear Security Administration of U.S. Department of Energy (Contract No. 89233218CNA000001). This work was conducted within Claremont Graduate University's Engineering and Computational Mathematics Clinic program.

Computer code availability

Name: dfnWorks
Contact: Jeffrey D. Hyman, jhyman@lanl.gov
No hardware requirement
Program language: Python 3
Software requirement: Docker (optional)

The software is open source and can be obtained at https://github.com/lanl/dfnWorks. A docker container is also available at https://hub.docker.com/r/ees16/dfnworks.

References

Aldrich, G., Hyman, J.D., Karra, S., Gable, C.W., Makedonska, N., Viswanathan, H., Woodring, J., Hamann, B., 2017. Analysis and visualization of discrete fracture networks using a flow topology graph. IEEE Trans. Vis. Comput. Graphics 23, 1896–1909.

Andresen, C.A., Hansen, A., Le Goc, R., Davy, P., Hope, S.M., 2013. Topology of fracture networks. Front. Phys. 1 (August), 1–5. http://dx.doi.org/10.3389/fphy.2013.00007.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer-Verlag, Berlin, Heidelberg.

Bonnet, E., Bour, O., Odling, N.E., Davy, P., Main, I., Cowie, P., Berkowitz, B., 2001. Scaling of fracture systems in geological media. Rev. Geophys. 39 (3), 347–383.

Bourret, S., Kwicklis, E., Miller, T., Stauffer, P., 2019. Evaluating the importance of barometric pumping for subsurface gas transport near an underground nuclear test site. Vadose Zone J. 18 (1), 180134. http://dx.doi.org/10.2136/vzj2018.07.0134.

Chami, I., Abu-El-Haija, S., Perozzi, B., Re, C., Murphy, K., 2021. Machine learning on graphs: A model and comprehensive taxonomy. arXiv:2005.03675.

de Dreuzy, J., Davy, P., Bour, O., 2002. Hydraulic properties of two-dimensional random fracture networks following power law distributions of length and aperture. Water Resour. Res. 38, 1276. http://dx.doi.org/10.1029/2001WR001009.

Dershowitz, W., Fidelibus, C., 1999. Derivation of equivalent pipe network analogues for three-dimensional discrete fracture networks by the boundary element method. Water Resour. Res. 35 (9), 2685–2691.

Doolaeghe, D., Davy, P., Hyman, J.D., Darcel, C., 2020. Graph-based flow modeling approach adapted to multiscale discrete-fracture-network models. Phys. Rev. E 102 (5), 053312.

Focaccia, S., Panini, G., Pedrazzoli, P., Ciriello, V., 2021. A meta-modeling approach for hydrological forecasting under uncertainty: Application to groundwater nitrate response to climate change. J. Hydrol. 603, 127173. http://dx.doi.org/10.1016/j.jhydrol.2021.127173.

Follin, S., Hartley, L., Rhén, I., Jackson, P., Joyce, S., Roberts, D., Swift, B., 2014. A methodology to constrain the parameters of a hydrogeological discrete fracture network model for sparsely fractured crystalline rock, exemplified by data from the proposed high-level nuclear waste repository site at Forsmark, Sweden. Hydrogeol. J. 22 (2), 313–331.

Frampton, A., Cvetkovic, V., 2010. Inference of field-scale fracture transmissivities in crystalline rock using flow log measurements. Water Resour. Res. 46 (W11502), URL: https://doi.org/10.1029/2009WR008367.

Hope, S.M., Davy, P., Maillot, J., Le Goc, R., Hansen, A., 2015. Topological impact of constrained fracture growth. Front. Phys. 3, 75.

Huseby, O., Thovert, J.-F., Adler, P.M., 1997. Geometry and topology of fracture systems. J. Phys. A: Math. Gen. 30 (5), 1415. http://dx.doi.org/10.1088/0305-4470/30/5/012.

Hyman, J.D., Aldrich, G., Viswanathan, H., Makedonska, N., Karra, S., 2016a. Fracture size and transmissivity correlations: Implications for transport simulations in sparse three-dimensional discrete fracture networks following a truncated power law distribution of fracture size. Water Resour. Res. 52, 6472–6489, URL: https://doi.org/10.1002/2016WR018806.

Hyman, J.D., Dentz, M., 2021. Transport upscaling under flow heterogeneity and matrix-diffusion in three-dimensional discrete fracture networks. Adv. Water Resour. 155, 103994.

Hyman, J.D., Dentz, M., Hagberg, A., Kang, P., 2019a. Emergence of stable laws for first passage times in three-dimensional random fracture networks. Phys. Rev. Lett. 123 (24), 248501.

Hyman, J.D., Dentz, M., Hagberg, A., Kang, P., 2019b. Linking structural and transport properties in three-dimensional fracture networks. J. Geophys. Res. Solid. Earth.

Hyman, J.D., Gable, C.W., Painter, S.L., Makedonska, N., 2014. Conforming delaunay triangulation of stochastically generated three dimensional discrete fracture networks: A feature rejection algorithm for meshing strategy. SIAM J. Sci. Comput. 36 (4), A1871–A1894.

Hyman, J.D., Hagberg, A., Osthus, D., Srinivasan, S., Viswanathan, H., Srinivasan, G., 2018. Identifying backbones in three-dimensional discrete fracture networks: A bipartite graph-based approach. Multisc. Model. Simul. 16, 1948–1968.

Hyman, J.D., Hagberg, A., Srinivasan, G., Mohd-Yusof, J., Viswanathan, H., 2017. Predictions of first passage times in sparse discrete fracture networks using graph-based reductions. Phys. Rev. E 96, 013304. http://dx.doi.org/10.1103/PhysRevE.96.013304.

Hyman, J.D., Jiménez-Martínez, J., 2017. Dispersion and mixing in three-dimensional discrete fracture networks: Nonlinear interplay between structural and hydraulic heterogeneity. Water Resour. Res. 54 (5), 3243–3258.

Hyman, J.D., Jimenez-Martinez, J., Gable, C.W., Stauffer, P.H., Pawar, R.J., 2020. Characterizing the impact of fractured caprock heterogeneity on supercritical CO_2 injection. Transp. Porous Media 131 (3), 935–955.

Hyman, J.D., Jiménez-Martínez, J., Viswanathan, H., Carey, J., Porter, M., Rougier, E., Karra, S., Kang, Q., Frash, L., Chen, L., Lei, D., Makedonska, N., 2016b. Understanding hydraulic fracturing: a multi-scale problem. Phil. Trans. R. Soc. A 374 (2078), 20150426.

Hyman, J.D., Karra, S., Makedonska, N., Gable, C.W., Painter, S.L., Viswanathan, H.S., 2015a. dfnWorks: A discrete fracture network framework for modeling subsurface flow and transport. Comput. Geosci. 84, 10–19.

Hyman, J.D., Painter, S.L., Viswanathan, H.S., Makedonska, N., Karra, S., 2015b. Influence of injection mode on transport properties in kilometer-scale three-dimensional discrete fracture networks. Water Resour. Res. 51 (9), 7289–7308.

Hyman, J.D., Sweeney, M.R., Gable, C.W., Svyatsky, D., Lipnikov, K., Moulton, J.D., 2022. Flow and transport in three-dimensional discrete fracture matrix models using mimetic finite difference on a conforming multi-dimensional mesh. J. Comput. Phys. 466, 111396.

Jenkins, C., Chadwick, A., Hovorka, S.D., 2015. The state of the art in monitoring and verification—ten years on. Int. J. Greenh. Gas Control 40, 312–349.

Joyce, S., Hartley, L., Applegate, D., Hoek, J., Jackson, P., 2014. Multi-scale groundwater flow modeling during temperate climate conditions for the safety assessment of the proposed high-level nuclear waste repository site at Forsmark, Sweden. Hydrogeol. J. 22, 1233–1249.

Kang, P.K., Dentz, M., Le Borgne, T., Lee, S., Juanes, R., 2017. Anomalous transport in disordered fracture networks: spatial Markov model for dispersion with variable injection modes. Adv. Water Resources 106, 80–94.

Kang, P.K., Hyman, J.D., Han, W.S., Dentz, M., 2020. Anomalous transport in three-dimensional discrete fracture networks: Interplay between aperture heterogeneity and injection modes. Water Resour. Res. 56 (11), e2020WR027378. http://dx.doi.org/10.1029/2020WR027378.

Karra, S., O'Malley, D., Hyman, J.D., Viswanathan, H.S., 2018. Modeling flow and transport in fracture networks using graphs. Phys. Rev. 97 (033304).

Kreft, A., Zuber, A., 1978. On the physical meaning of the dispersion equation and its solutions for different initial and boundary conditions. Chem. Eng. Sci. 33 (11), 1471–1480. http://dx.doi.org/10.1016/0009-2509(78)85196-3.

Krotz, J., Sweeney, M.R., Gable, C.W., Hyman, J.D., Restrepo, J.M., 2022. Variable resolution Poisson-disk sampling for meshing discrete fracture networks. J. Comput. Appl. Math. 407, 114094. http://dx.doi.org/10.1016/j.cam.2022.114094.

Li, Y., Lu, P., Zhang, G., 2022. An artificial-neural-network-based surrogate modeling workflow for reactive transport modeling. Pet. Res. 7 (1), 13–20. http://dx.doi.org/10.1016/j.ptlrs.2021.06.002.

Lichtner, P.C., Hammond, G.E., Lu, C., Karra, S., Bisht, G., Andre, B., Mills, R.T., Kumar, J., Frederick, J.M., 2020. PFLOTRAN User Manual. Technical Report, http://documentation.pflotran.org.

Maillot, J., Davy, P., Le Goc, R., Darcel, C., De Dreuzy, J.-R., 2016. Connectivity, permeability, and channeling in randomly distributed and kinematically defined discrete fracture network models. Water Resour. Res. 52 (11), 8526–8545.

Makedonska, N., Painter, S.L., Bui, Q.M., Gable, C.W., Karra, S., 2015. Particle tracking approach for transport in three-dimensional discrete fracture networks. Computat. Geosci. 19 (5), 1123–1137.

Middleton, R., Carey, J., Currier, R., Hyman, J., Kang, Q., Karra, S., Jiménez-Martínez, J., Porter, M., Viswanathan, H., 2015. Shale gas and non-aqueous fracturing fluids: Opportunities and challenges for supercritical CO_2. Appl. Energy 147, 500–509.

National Academies of Sciences, Engineering, and Medicine, et al., 2020. Characterization, Modeling, Monitoring, and Remediation of Fractured Rock. National Academies Press.

Neuman, S., 2005. Trends, prospects and challenges in quantifying flow and transport through fractured rocks. Hydrogeol. J. 13 (1), 124–147.

Painter, S.L., Gable, C.W., Kelkar, S., 2012. Pathline tracing on fully unstructured control-volume grids. Comput. Geosci. 16 (4), 1125–1134.

Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian Processes for Machine Learning. MIT Press.

Selroos, J.-O., Walker, D.D., Ström, A., Gylling, B., Follin, S., 2002. Comparison of alternative modelling approaches for groundwater flow in fractured rock. J. Hydrol. 257 (1–4), 174–188.

Sherman, T., Hyman, J.D., Bolster, D., Makedonska, N., Srinivasan, G., 2019. Characterizing the impact of particle behavior at fracture intersections in three-dimensional discrete fracture networks. Phys. Rev. E 99 (1), 013110.

Srinivasan, S., Cawi, E., Hyman, J., Osthus, D., Hagberg, A., Viswanathan, H., Srinivasan, G., 2020. Physics-informed machine learning for backbone identification in discrete fracture networks. Comput. Geosci. 24, 1429–1444.

Srinivasan, S., Karra, S., Hyman, J., Viswanathan, H., Srinivasan, G., 2019. Model reduction for fractured porous media: a machine learning approach for identifying main flow pathways. Comput. Geosci. 23, 617–629.

Stamile, C., Marzullo, A., Deusebio, E., 2021. Graph Machine Learning. Packt Publishing.

Sun, Y., Carrigan, C.R., 2014. Modeling noble gas transport and detection for the comprehensive nuclear-test-ban treaty. Pure Appl. Geophys. 171 (3), 735–750. http://dx.doi.org/10.1007/s00024-012-0514-4.

Sweeney, M.R., Gable, C.W., Karra, S., Stauffer, P.H., Pawar, R.J., Hyman, J.D., 2020. Upscaled discrete fracture matrix model (UDFM): an octree-refined continuum representation of fractured porous media. Comput. Geosci. 24 (1), 293–310. http://dx.doi.org/10.1007/s10596-019-09921-9.

Thenon, A., Gervais, V., Le Ravalec, M., 2016. Multi-fidelity meta-modeling for reservoir engineering — application to history matching. Comput. Geosci. 20, 1231–1250.

Thiros, N.E., Gardner, W.P., Maneta, M.P., Brinkerhoff, D.J., 2022. Quantifying subsurface parameter and transport uncertainty using surrogate modelling and environmental tracers. Hydrol. Process. 36 (11), e14743. http://dx.doi.org/10.1002/hyp.14743.

Ushijima-Mwesigwa, H., Hyman, J.D., Hagberg, A., Safro, I., Karra, S., Gable, C.W., Sweeney, M.R., Srinivasan, G., 2021. Multilevel graph partitioning for three-dimensional discrete fracture network flow simulations. Math. Geosci. 1–26.

Valera, M., Guo, Z., Kelly, P., Matz, S., Cantu, V.A., Percus, A.G., Hyman, J.D., Srinivasan, G., Viswanathan, H.S., 2018. Machine learning techniques for graph-based representations of three-dimensional discrete fracture networks. Comput. Geosci. 22, 695–710.

Viswanathan, H.S., Ajo-Franklin, J., Birkholzer, J.T., Carey, J.W., Guglielmi, Y., Hyman, J.D., Karra, S., Pyrak-Nolte, L.J., Rajaram, H., Srinivasan, G., Tartakovsky, D.M., 2022. From fluid flow to coupled processes in fractured rock: Recent advances and new frontiers. Rev. Geophys. 60 (1), e2021RG000744. http://dx.doi.org/10.1029/2021RG000744.

Zhou, M., Shadabfar, M., Huang, H., Leung, Y.F., Uchida, S., 2020. Meta-modelling of coupled thermo-hydro-mechanical behaviour of hydrate reservoir. Comput. Geotech. 128, 103848. http://dx.doi.org/10.1016/j.compgeo.2020.103848.