PROCEEDINGS, 51st Workshop on Geothermal Reservoir Engineering
Stanford University, Stanford, California, February 9-11, 2026
SGP-TR-230

Deep-Learning-Based Fracture Network Parameterization and History Matching for Enhanced Geothermal Reservoirs

Yuebin Fan[1], Tianjia Huang[2], Su Jiang[1]

[1] Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213

[2] Department of Petroleum Engineering, Texas A&M University, College Station, TX, 77843, USA

yuebinf@andrew.cmu.edu

Keywords: Enhanced geothermal systems; fracture network parameterization; generative model; data assimilation

ABSTRACT

Accurately predicting flow and thermal performance in fractured reservoirs is essential for the development and optimization of enhanced geothermal systems (EGS). Fracture geometry, connectivity, and heterogeneity strongly control reservoir behavior, but these features are high-dimensional and difficult to observe directly, leading to significant uncertainty in flow and thermal predictions. A key challenge is how to efficiently tune fracture networks using limited observations to reduce the prediction uncertainty. In this work, we propose a latent generative modeling framework to parameterize discrete fracture networks (DFNs) with low-dimensional latent variables while preserving fracture geometry and connectivity. The model is trained on ensembles of 2D DFNs and can generate new realizations that are visually and statistically consistent with the training data. We integrate this generative representation with embedded discrete fracture model (EDFM) simulations and apply an ensemble-based data assimilation method for history matching. By assimilating temperature and tracer observations, this framework updates the latent variables and tune fracture networks that better match observed data. We validate the method using a synthetic 2D EGS test case. The posterior results present significant uncertainty reduction in temperature predictions, along with fracture networks that more closely agree with the 'true' synthetic fracture networks. This study demonstrates the potential of deep generative modeling for efficient fracture parameterization and uncertainty reduction in geothermal reservoir characterization.

1. INTRODUCTION

Enhanced geothermal systems (EGS) extend geothermal development beyond naturally permeable hydrothermal resources by hydraulically stimulating hot, low-permeability rock to create or enhance fracture networks. These fractures govern reservoir-scale flow pathways and therefore control both fluid circulation and heat extraction between injection and production wells. Reliable prediction of thermal breakthrough and production performance depends on accurate modeling of coupled flow and heat transport in fractured reservoirs. However, the discrete fracture network is observed indirectly and is typically poorly constrained, leading to substantial uncertainty in forecasts and limiting reservoir management. History matching is commonly used to reduce this uncertainty by calibrating geomodels against monitoring data, but fracture characterization remains difficult because realistic networks are described by thousands of coupled geometric parameters (*e.g.*, fracture locations, lengths, orientations, intersections, and connectivity). Tuning these features fracture-by-fracture is inherently high-dimensional, and small geometric changes can alter connectivity and produce qualitatively different flow responses, which makes calibration challenging.

Our goal is to develop an efficient parameterization of discrete fracture networks (DFNs) that enables effective history matching for EGS. We propose a latent diffusion model (LDM) that learns a low-dimensional representation of the DFN geometry and generates geologically plausible network realizations from a small set of latent variables. We then integrate this parameterization into a history matching workflow by performing inversion in latent space rather than directly over thousands of geometric parameters. This framework enables the update of the entire fracture network while better preserving geometry and connectivity, and supports calibrated DFN realizations and reservoir forecasting with quantified uncertainty conditioned on the available observations.

The modeling challenge in EGS is to represent fractures and their complex connectivity at reservoir scale that is physically meaningful and compatible with inverse modeling. Dual-continuum models (Kazemi et al., 1976; Thomas et al., 1983) approximate fractures through upscaled properties, which can be effective for densely and uniformly connected fracture systems, but it is difficult to preserve explicit fracture geometry or network topology. Discrete fracture models (DFM) (Karimi-Fard & Durlofsky, 2016; Karimi-Fard et al., 2004) resolve fractures explicitly using conforming, unstructured meshes and can achieve high fidelity, but they rely on geometry-conforming grids and require re-meshing whenever fracture geometry is modified, which complicates the fracture tuning and calibration. The embedded discrete fracture model (EDFM) (Li & Lee, 2008; Moinfar et al., 2014) provides a practical way by embedding fractures into structured grids and treating them as source or sink terms, achieving accuracy comparable to DFMs while substantially reducing meshing complexity. This makes EDFM well suited for history matching in fractured geothermal reservoirs, where forward simulations must be repeated across many candidate fracture realizations to calibrate models and quantify uncertainty.

With EDFM enabling repeated forward simulation without re-meshing, fracture characterization remains the dominant bottleneck. The geometry and connectivity of DFN control the flow and thermal performance, but the network is only indirectly constrained at reservoir scale. History matching, the process of calibrating geological models so that simulation results match observations, is essential, but it is particularly challenging in fractured systems. The small geometry change can result in a large change in connectivity and reservoir performance. Some existing studies mitigate this difficulty by fixing geometry and only calibrating the fracture aperture or matrix

properties (Kim et al., 2021; Wu et al., 2024). However, direct geometry inversion has been demonstrated using transdimensional Bayesian frameworks that iteratively add, delete, and move fractures, first in 2D DFNs (Somogyvári et al., 2017) and later extended to 3D DFNs (Ringel et al., 2021). These approaches present the importance of geometry for capturing connectivity. Fracture-by-fracture updates remain difficult to scale to dense networks, and the geometry remains a fundamentally discrete object calibration. Transform-based parameterizations, such as Hough-transform methods (Lu & Zhang, 2015; Yao et al., 2018) offer an alternative by mapping fracture patterns into a continuous representation, which enables more efficient inversion. But these transformations are defined on individual fractures. Preserving network connectivity remains an open challenge.

Recent progress in deep generative modeling provides an efficient pathway for geological model parameterization. These models learn a low-dimensional latent representation, often designed to follow a Gaussian prior, so that sampling and updating in a continuous latent space can generate new realizations that reproduce the target geological statistics. Early work using variational autoencoders (VAEs) (Bao et al., 2022; Canchumuni et al., 2021) and generative adversarial networks (GANs) (Chen et al., 2023; Teng et al., 2025; Zhou et al., 2025) demonstrated that generative models can produce realistic non-Gaussian geological patterns and have begun to be explored for fracture network generation. More recently, diffusion models have been applied to geomodel generation and have shown a strong sample quality. Di Federico and Durlofsky (2025a) developed a latent diffusion model (LDM) to parameterize 2D channelized facies systems and integrate the learned parameterization into data assimilation, and later extended the approach to 3D parameterization for history matching (Di Federico & Durlofsky, 2025b). For fracture systems, Teng et al. (2024) applied an LDM to generate 2D DFNs, but validation against flow simulations and the application in inversion workflow remain limited. DFNs are substantially more complex because they involve dense fractures with frequent intersections, where the reservoir responses are governed by network connectivity. Preserving connectivity and associated topology during inversion remains a key challenge.

In this work, we develop an integrated workflow, referred to as LDM-EDFM-ESMDA, that couples LDM for efficient parameterization of DFNs, EDFM for accurate forward simulation of fractured reservoirs in many network realizations, and the ensemble smoother with multiple data assimilation (ESMDA) to update latent variables and reduce forecasting uncertainty. We perform history matching in the latent Gaussian space learned by the LDM, which aligns with the Gaussian assumption of ESMDA, while avoiding fracture by fracture tuning in a discrete geometric space. We parameterize DFNs on a structured grid that is directly compatible with EDFM. For each grid block with fracture, we represent geometry using two continuous variables (fracture surface area and distance to the grid center), which are then used to generate the EDFM mesh representation. Training samples are generated with the MATLAB Reservoir Simulation Toolbox (MRST) (Lie & Møyner, 2021), and forward simulations of coupled flow and heat transport are performed with TOUGH (Moridis, 2014; Moridis & Freeman, 2014). We evaluate the LDM in terms of fracture network statistics and reservoir response statistics. Finally, we apply ESMDA to assimilate temperature and tracer data under different monitoring schedules and assess the posterior DFNs and the uncertainty reduction in temperature predictions.

This paper proceeds as follows. Section 2 introduces the generation of fracture networks and the EGS simulation setup based on EDFM. Section 3 describes the proposed LDM parameterization for DFNs. Section 4 presents the ESMDA history matching process in latent space. Section 5 presents the validation of DFNs generated from LDM, and posterior results for the fracture networks calibration and uncertainty quantification for predictions. Section 6 summarizes this work and provides suggestions for future research.

2. GEOMODEL AND SIMULATION SETUP

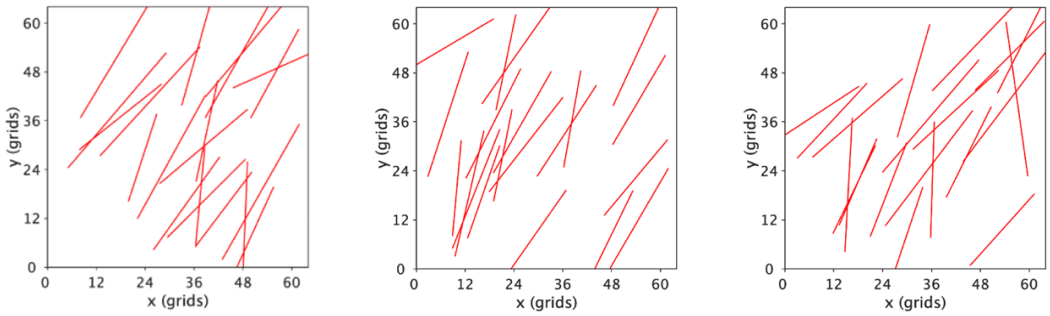
In this section, we describe the 2D discrete fracture network (DFN) generation process, followed by a high-level description of the embedded discrete fracture model (EDFM) used to simulate coupled flow and heat transport. Finally, we describe the simulation setup for this work.

2.1 DFN generation

We generate an ensemble of 2D DFN realizations in a domain of size 256 m × 256 m. The domain is discretized using a structured grid with 64 × 64 grids. Each DFN realization contains $N_f = 20$ fractures represented as line segments (Fig. 1). These fractures will be embedded in the structure grids. These fractures are parameterized by midpoint location (x_c, y_c), length l and orientations θ. To reduce boundary truncation effects and ensure consistent fracture statistics across realizations, fracture midpoints (x_c, y_c) are sampled uniformly within a centered squared window of length 153.6 m. To reduce the probability of unrealistically short or long segments relative to the domain scale, fracture lengths l are sampled following a truncated power-law distribution (de Dreuzy et al., 2002),

$$n(l) = \alpha l^{-a}, \tag{1}$$

where $n(l)$ denotes the number of fractures per unit length, α is the density coefficient, and a is the scaling exponent. We set $a = 2.34$, and truncate the fracture length within $[89.6, 153.6]$ m. Fracture orientations θ are sampled from a Gaussian distribution with mean 60° and standard deviation 15°. Using the location of the sampled (x_c, y_c), length l, and orientation θ, we determine the endpoint coordinates of each fracture, which are subsequently entered into MRST (Lie & Møyner, 2021) to generate DFN realizations. Figure 1 shows three randomly generated DFN realizations. Realization 1 is used as the 'true' model for the history matching tasks in Section 5. The variability in fracture geometry and network connectivity is clearly presented in these realizations. We generated 30,000 realizations as the training dataset for the proposed parameterization method.



(a) Real 1 (b) Real 2 (c) Real 3

Figure 1: Three random MRST-generated DFN realizations: (a) is used as the 'true' model for history matching in Section 5.

2.2. Embedded Discrete Fracture Model (EDFM)

We use the embedded discrete fracture model (EDFM) to represent fractures explicitly while avoiding re-meshing as fracture geometry varies. In EDFM, the fracture and matrix domains are discretized independently and coupled through specifically defined non-neighboring connections (NNCs). For the 2D setting, fractures are represented as lower-dimensional line segments, and the fracture apertures are used to define the corresponding control volumes. The matrix is discretized into rectangular control volumes on a structured grid, whereas fractures are embedded into the matrix and partitioned into segments according to matrix block boundaries. The fracture flow and matrix flow are coupled through transfer functions, resulting in two interacting mesh layers, *i.e.,* a matrix mesh and a fracture mesh.

In addition to standard matrix-matrix connections, three types of NNCs are defined: (i) connections between adjacent segments along the same fracture, (ii) connections between fracture segments and the matrix blocks they intersect, and (iii) connections between intersecting fracture segments. Fluxes across both neighboring connections and NNCs are computed using a two-point flux approximation. For single-phase aqueous flow, the volumetric flux between control volumes i and j follows Darcy's law,

$$F = \lambda T_{ij} \Delta P, \tag{2}$$

where λ denotes the phase mobility evaluated using upstream weighting, T_{ij} is the transmissibility between cells i and j, and ΔP is the pressure difference. For a pair of connected control volumes, the corresponding transmissibility is given by a general form:

$$T_{ij} = \frac{k_{ij} A_{ij}}{d_{ij}}, \tag{3}$$

where k_{ij} is an effective permeability for the connected features, A_{ij} is the characteristic flow area, and d_{ij} is the characteristic distance between the connected entities. The specific definitions of these parameters depend on the type of connection and the local fracture-matrix or fracture-fracture geometry. Detailed EDFM formulations and transmissibility construction are provided in references (Karimi-Fard et al., 2004; Li & Lee, 2008; Moinfar et al., 2014). In this study, the geometric parameters A_{ij} and d_{ij} for all NNCs are obtained from the Hierarchical Fracture Model (HFM) module implemented in MRST (Lie & Møyner, 2021). All control volumes and transmissibilities are assembled into a global discrete system, which is solved iteratively as part of the coupled flow-thermal simulation.

2.3 Numerical simulation setup

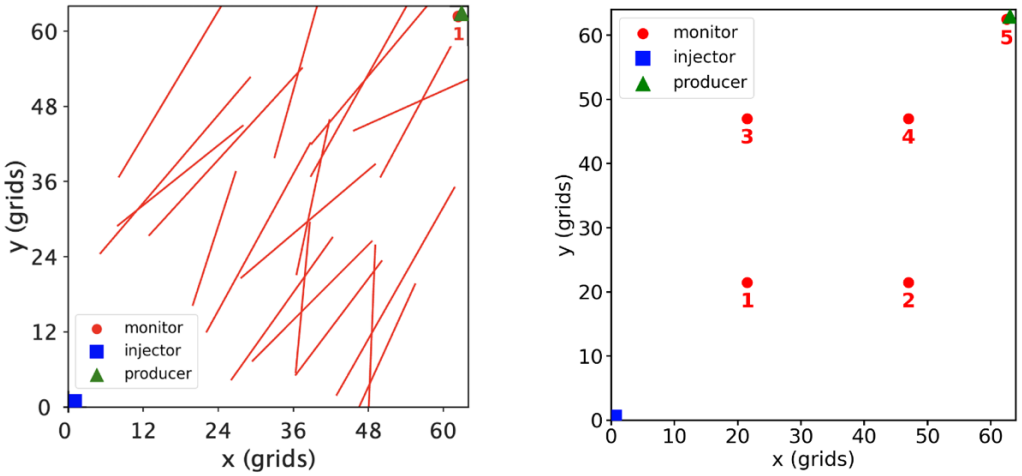
TOUGH+RealGasBrine (TRGB) (Moridis, 2014; Moridis & Freeman, 2014) is applied to simulate non-isothermal, single-phase aqueous flow and heat transport within the EDFM discretization. TRGB solves the coupled mass and energy conservation equations, written in the general form

$$\frac{\partial M^{\kappa}}{\partial t} = \nabla \cdot \mathbf{F}^{\kappa} + q^{\kappa}, \tag{4}$$

where $M^{\kappa}[kg\ m^{-3}$ or $J\ m^{-3}]$ denotes the accumulation term representing the mass of component κ (*e.g.*, water in this study) or the internal energy per control volume. The flux term $\mathbf{F}^{\kappa}$ represents the advective mass flux of component κ $[kg\ m^{-2}\ s^{-1}]$ or the heat flux $[W\ m^{-2}]$ across control-volume boundaries, and q^{κ} denotes source or sink contributions, and t is time $[s]$. The resulting nonlinear system is solved fully implicitly using Newton-Raphson iterations.

For the 2D case, our objective is to calibrate the fracture geometry only. We therefore set all other reservoir properties to be homogeneous across realizations. Each DFN realization generated in Section 2.1 is embedded into the 64 × 64 structured grids. The fracture aperture is 2 *mm*. Fracture porosity and permeability are set to $\phi_f = 1.0$ and $k_f = 1 \times 10^{-8}\ m^2$. The matrix is homogeneous with porosity $\phi_m = 0.10$ and permeability $k_m = 5 \times 10^{-15}\ m^2$. The initial pressure and temperature are set to $P_0 = 29\ MPa$ and $T_0 = 200\ °C$, respectively. We consider a single injector-producer configuration as shown in Fig. 2. Water is injected into the grid cell (1, 1) at a constant mass rate of 0.15 *kg/s* with injection temperature $T_{\text{inj}} = 50\ °C$ for 1800 days. The producer is located at grid (64, 64) and with a constant bottomhole pressure control of 29 *MPa*. To provide additional transport constraints, we perform a tracer test by co-injecting tracer at 100 *mg/kg*

(corresponding to the injection rate of 15 *mg/s*) for 12 hours and monitoring the tracer concentration at the monitors. The rest boundaries are closed boundaries, as the simulation domain is constructed to represent a symmetric region within a well network.



(a) One observation site (b) Five observation sites

Figure 2: Well and monitoring configurations used in simulation. A single injector and a single producer are considered, with monitoring data collected at (a) one observation site at the producer, and (b) five observation sites distributed in the domain. These observations are used for the history matching in Section 5.

3. LATENT DIFFUSION MODEL (LDM) BASED PARAMETERIZATION

In this section, we describe the latent diffusion model (LDM) used to parameterize DFN geometry, following Di Federico and Durlofsky (2025a). Our objective is to learn a continuous latent space for DFN representations and train a diffusion model in that latent space. This enables DFN realizations to be generated and updated through a small number of latent variables, rather than tuning fractures directly in a discrete geometric space. We introduce the DFN representation used in this work, which is designed to be compatible with EDFM forward simulations. We will then present the loss function and training details.

3.1. Latent diffusion model

The latent diffusion model combines two components: (i) variational autoencoder (VAE) that maps DFN representations into a low dimensional latent space and reconstructs them back to the model space, and (ii) a diffusion model trained in latent space to represent the DFN prior and enable sampling from Gaussian distribution. The architecture is presented in Fig. 3. Each DFN realization is represented on the structured grid by two continuous fields, $\mathbf{m} = \{\mathbf{m}_A, \mathbf{m}_D\} \in \mathbb{R}^{N_x \times N_y \times 2}$, where $\mathbf{m}_A \in \mathbb{R}^{N_x \times N_y}$ denotes the fracture surface area within each grid cell and $\mathbf{m}_D \in \mathbb{R}^{N_x \times N_y}$ denotes the distance from the fracture to the cell center. In this work, $N_x = N_y = 64$. This representation preserves subcell geometric information needed to reconstruct fractures for EDFM while keeping the parameterization continuous and grid-based.

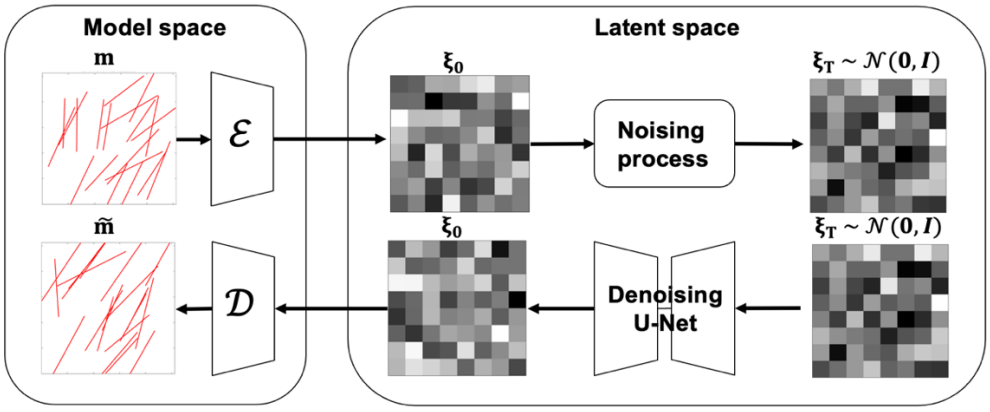


Figure 3: Framework of the latent diffusion model for DFN generation, modified based on Di Federico and Durlofsky (2025a).

The DFN representation $\mathbf{m}$ is first mapped to low dimensional latent variables $\boldsymbol{\xi} \in \mathbb{R}^{n_x \times n_y \times 2}$ using a VAE encoder, $\boldsymbol{\xi} = \mathcal{E}(\mathbf{m})$, where n_x and n_y denote latent resolution. The decoder maps latent variables back to the original model space, $\tilde{\mathbf{m}} = \mathcal{D}(\boldsymbol{\xi})$. The sampling and model updates can be performed in the low dimensional latent space. In our implementation, the encoder is a convolutional network with three downsampling layers and four residual blocks, where each residual block uses group normalization and two convolutional layers. This architecture yields an overall downsampling factor of 8 and a latent tensor of size $8 \times 8 \times 2$. The decoder mirrors the encoder with four residual blocks and three upsampling convolutional layers to reconstruct $\tilde{\mathbf{m}}$ from $\boldsymbol{\xi}$.

Using the VAE alone often produces overly smooth reconstructions and can miss the sharp, localized structure required to represent discrete fracture geometry. We therefore apply the diffusion model in the latent space, which provides a substantially more expressive generative prior for the latent variables $\boldsymbol{\xi}$ (Ho et al., 2020). The diffusion model consists of a forward noising process that gradually perturbs latent samples into Gaussian noise, and a learned reverse denoising process that maps Gaussian noise back to the data distribution. For any given step in the diffusion model, a neural network is trained to predict the injected noise, enabling iterative denoising during sampling. Let $\boldsymbol{\xi}_0$ denote the latent variables generated by the VAE encoder and $q(\boldsymbol{\xi}_0)$ denote its data distribution. The forward diffusion process is defined as a Markov chain of length T ,

$$q(\boldsymbol{\xi}_{1:T} \mid \boldsymbol{\xi}_0) = \prod_{t=1}^{T} q(\boldsymbol{\xi}_t \mid \boldsymbol{\xi}_{t-1}), \tag{5}$$

with Gaussian transitions

$$q(\boldsymbol{\xi}_t \mid \boldsymbol{\xi}_{t-1}) = \mathcal{N}\left(\boldsymbol{\xi}_t; \sqrt{\alpha_t}\,\boldsymbol{\xi}_{t-1}, (1-\alpha_t)\mathbf{I}\right), t = 1,2,\dots,T \tag{6}$$

where α_t (equivalently $\beta_t = 1 - \alpha_t$) specifies the noise schedule. As t increases, $\boldsymbol{\xi}_t$ approaches Gaussian noise and can be sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

The denoising process p_θ starting from $p(\boldsymbol{\xi}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is parameterized as

$$p_\theta(\boldsymbol{\xi}_{0:T}) = p(\boldsymbol{\xi}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{\xi}_{t-1} \mid \boldsymbol{\xi}_t), \tag{7}$$

where each transition is modeled as a Gaussian

$$p_\theta(\boldsymbol{\xi}_{t-1} \mid \boldsymbol{\xi}_t) = \mathcal{N}(\boldsymbol{\xi}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{\xi}_t, t), \boldsymbol{\Sigma}_t), \tag{8}$$

Following Ho et al. (2020), we set $\Sigma_t = \beta_t \mathbf{I}$ and parameterize the mean μ_θ using a noise prediction network $\boldsymbol{\epsilon}_\theta(\boldsymbol{\xi}_t, t)$,

$$\boldsymbol{\mu}_\theta(\boldsymbol{\xi}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{\xi}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{\epsilon}_\theta(\boldsymbol{\xi}_t, t)\right), \tag{9}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. In this work, $\boldsymbol{\epsilon}_\theta$ is implemented as a U-Net. After training, new latent variables are generated by iteratively applying the reverse updates from $\boldsymbol{\xi}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to $\boldsymbol{\xi}_0$. The corresponding denoising diffusion probabilistic models (DDPM) sampling step can be written as

$$\boldsymbol{\xi}_{t-1} = \boldsymbol{\mu}_\theta(\boldsymbol{\xi}_t, t) + \sigma_t \mathbf{z}, \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{10}$$

which introduces stochasticity through the random term $\mathbf{z}$.

In standard DDPM sampling, each reverse step includes an additional random perturbation, so the mapping from the initial latent noise to the generated sample is inherently stochastic. This stochasticity is not ideal for history matching, where latent updates are supposed to produce consistent and interpretable changes in the decoded DFN and the corresponding simulation responses. We therefore use denoising diffusion implicit models (DDIM) (Song et al., 2020), which reparametrize the reverse dynamics using a deterministic, non-Markovian update:

$$\boldsymbol{\xi}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{\boldsymbol{\xi}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_\theta(\boldsymbol{\xi}_t, t)}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}}\,\boldsymbol{\epsilon}_\theta(\boldsymbol{\xi}_t, t). \tag{11}$$

DDIM typically achieves comparable sample quality with substantially fewer reverse steps (often tens rather than thousands), and its deterministic mapping provides a stable latent-to-DFN transformation that is better aligned with ensemble-based updates in history matching.

3.2 Loss function and training process

We train the LDM in two stages. We first train a VAE to learn the latent representation whose distribution is approximately Gaussian. We then freeze the VAE and train the diffusion model on the latent variables to learn the prior for DFNs in latent space. The VAE is trained by minimizing a reconstruction error together with a Kullback-Leibler(K-L) divergence loss (Kingma & Welling, 2013):

$$\mathcal{L}_{\text{VAE}} = ||\mathbf{m} - \tilde{\mathbf{m}}||_2^2 + \lambda_{\text{KL}} D_{\text{KL}}\big(q(\boldsymbol{\xi} \mid \mathbf{m}) \,||\, \mathcal{N}(\mathbf{0}, \mathbf{I})\big), \tag{12}$$

where KL divergence quantifies the difference between the latent distribution learned by the encoder and a standard Gaussian distribution, λ_{KL} controls the trade-off between reconstruction fidelity and Gaussian regularization in latent space.

After VAE training, each sample is encoded into latent variables $\boldsymbol{\xi}_0$. We train a diffusion model on $\boldsymbol{\xi}_0$ using the DDPM noise prediction objective (Ho et al., 2020). We train DDPM by minimizing

$$\mathcal{L}(\boldsymbol{\theta}) = E_{\xi_t,\epsilon}[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{\xi}_t, t)||^2] = E_{t,\xi_0,\epsilon}\left[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\,\boldsymbol{\xi}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, t\right)||^2\right]. \quad (13)$$

where $\boldsymbol{\epsilon}_\theta(\cdot, t)$ is implemented as a time-conditional U-Net. In this work, we use DDIM (Song et al., 2020) for sampling in inversion process after training, leveraging its deterministic and fast reverse updates for integration with the history matching workflow.

The training dataset contains 30,000 DFN realizations generated with MRST (Lie et al., 2012). Each realization is represented by a two-channel field $\mathbf{m} = \{\mathbf{m}_A, \mathbf{m}_D\} \in \mathbb{R}^{64\times64\times2}$. We randomly split the dataset into 80% for training and 20% for testing. Each channel is normalized to [0,1]. Cells with values below 10^{-8} are treated as background and mapped to -1 to increase contrast between fracture and non-fracture regions, which improves reconstruction and simplifies thresholding during postprocessing.

We first train the VAE by minimizing loss function in Eq. (12). After convergence, we freeze encoder $\mathcal{E}$ and decoder $\mathcal{D}$ and train the diffusion model by minimizing loss function in Eq. (13). Both stages use the Adam optimizer (Kingma, 2014) with learning rate 1×10^{-4}, weight decay, and batch size 10. We employ a linear noise schedule with $\beta_1 = 1.5 \times 10^{-3}$ and $\beta_T = 1.95 \times 10^{-2}$ over T=1000 diffusion steps. During inference we use DDIM with 100 steps. We set $\lambda_{KL} = 10^{-6}$ following Rombach et al. (2022). Both the VAE and diffusion model are trained for 1000 epochs. Training is conducted on a single NVIDIA H100 GPU, with a total time of approximately 25 hours.

4. HISTORY MATCHING WITH LDM-EDFM-ESMDA

In this work, history matching is conducted using an integrated LDM-EDFM-ESMDA workflow (Fig. 4). The LDM provides a low-dimensional, continuous parameterization of DFN geometry. EDFM enables forward simulation on a fixed structured grid without re-meshing across DFN realizations. The ensemble smoother with multiple data assimilation (ESMDA) (Emerick & Reynolds, 2013) updates the latent variables and further calibrate geological models conditioned on observations.

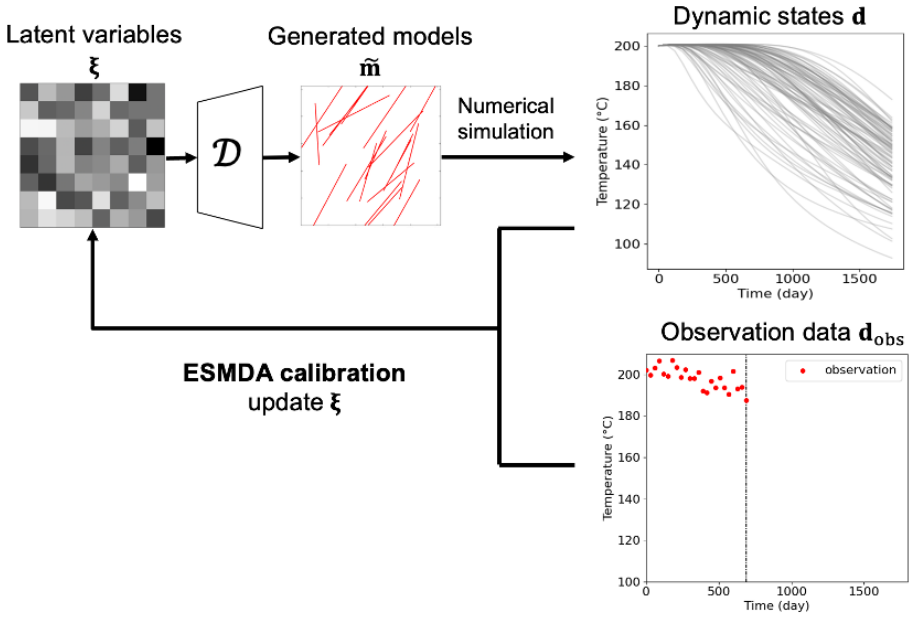


Figure 4: Integrated LDM-EDFM-ESMDA workflow for DFN history matching. Latent variables are updated by ESMDA, decoded to DFN representations by the LDM, embedded into the structured grid via EDFM, and evaluated through coupled flow-thermal simulations to compute predicted observations.

ESMDA has been widely used with reduced-order and learned parameterizations to calibrate geomodel parameters (Di Federico & Durlofsky, 2025a; Tang et al., 2021) and predict dynamic states (Jiang & Durlofsky, 2021). In our setting, the updated variables are the LDM latent variables $\boldsymbol{\xi}_T$ used for sampling (we update the flattened DDIM initial latent variables, denoted here by $\boldsymbol{\xi}$ for simplicity). The prior ensemble $\{\boldsymbol{\xi}_j^0\}_{j=1}^{N_r}$ is sampled from a standard normal distribution, consistent with the approximately Gaussian latent structure enforced during LDM training and aligned with the Gaussian assumptions underlying ESMDA.

Let $\mathbf{d} \in \mathbb{R}^{N_d}$ denote the data variables from numerical simulation in the historical period and $\mathbf{d}_{\text{obs}} \in \mathbb{R}^{N_d}$ denote the observations. ESMDA updates each ensemble member j over N_a assimilation steps as Emerick and Reynolds (2013)

$$\boldsymbol{\xi}_j^{k+1} = \boldsymbol{\xi}_j^k + C_{\xi d}^k \left(C_{dd}^k + \alpha_k C_d\right)^{-1} \left(\mathbf{d}_{\text{obs}} + \sqrt{\alpha_k}\mathbf{e}_j^k - \mathbf{d}_j^k\right), \quad j = 1, \ldots, N_r, \quad k = 1, \ldots, N_a, \tag{14}$$

where $C_{\xi d}^k$ is the cross-covariance between $\boldsymbol{\xi}$ and $\mathbf{d}$, C_{dd}^k is the auto-covariance of $\mathbf{d}$, and C_d is the observation error covariance. The perturbations $\mathbf{e}_j^k \sim \mathcal{N}(\mathbf{0}, C_d)$ account for measurement errors, and the inflation coefficients $\{\alpha_k\}$ satisfy $\sum_{k=1}^{N_a} \alpha_k^{-1} = 1$.

At each assimilation step k (Fig. 4), each updated latent vector $\boldsymbol{\xi}_j^k$ is mapped to a DFN realization $\widetilde{\mathbf{m}}_j^k$ using DDIM sampling in latent space followed by decoding. The resulting DFN is then embedded into the structured grid using EDFM and evaluated with coupled flow-thermal simulations (TRGB) to generate the data $\mathbf{d}_j^k$ in historical period. After the final assimilation step, posterior forecasts are obtained by sampling from the posterior ensembleof latent variables $\boldsymbol{\xi}_{\text{post}}$ and running the same LDM-EDFM forward workflow, generating calibrated fracture networks and corresponding posterior predictions.

5. LDM VALIDATION AND HISTORY MATCHING RESULTS

In this section, we first validate DFNs sampled from the LDM prior by comparing their fracture statistics and EDFM-based thermal and tracer responses against the MRST-generated training ensemble. We then present history matching results from the proposed LDM-EDFM-ESMDA workflow and present how monitoring design influences fracture-network inference and the resulting uncertainty in temperature forecasts.

5.1 Validation of DFN realizations generated from LDM

The generative quality of the LDM is evaluated first by comparing DFNs sampled from the learned prior with the MRST-generated training ensemble. Figure 5 shows three random DFN realizations sampled from the LDM. Since the LDM operates on the two-channel DFN representation introduced in Section 3, we apply a simple thresholding and postprocessing step to obtain a binary fracture-matrix field for visualization. The sampled DFNs reproduce the dominant orientation, length scale, and spatial distributions observed in the training realizations (Fig. 1), suggesting that the LDM captures the main geometric features of the prescribed DFN prior.

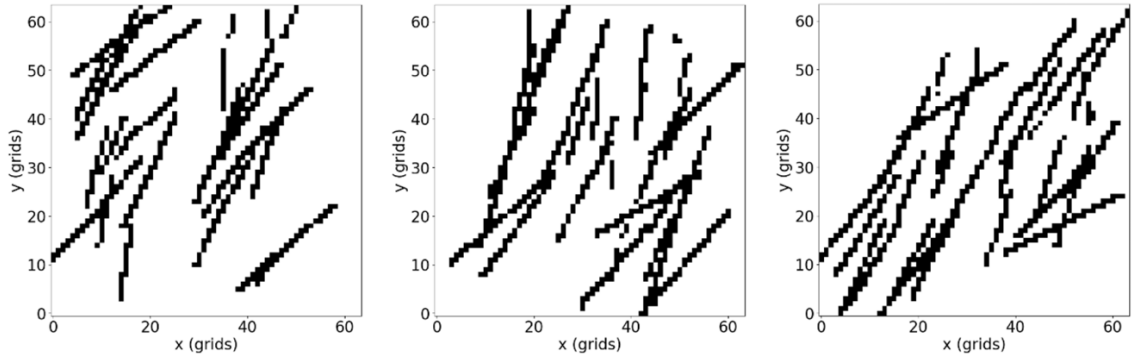


Figure 5: Three random DFN realizations sampled using LDM after thresholding and postprocessing.

To quantify statistical consistency, we extract fracture features from the gridded DFNs using the probabilistic Hough transform (PHT) (Galamhos et al., 1999). We then compare distributions of the key fracture characteristics used to define the DFN generator: fracture length, orientation, fracture count, and midpoint location. Figure 6 summarizes the comparison using 200 training realizations and 200 LDM-generated realizations. In Figs. 6(a)-(c), colored histograms show the distributions of the generated ensemble and the solid red curves show the training distributions. Figure 6(d) compares midpoint locations (generated in black, training in red). Overall, the generated ensemble closely matches the training distributions and reproduces the primary statistical characteristics of the DFN prior. We observe a slight underestimation in the detected fracture number for the generated ensemble. This discrepancy may result from postprocessing and line-detection limitations. Blurring fractures in the generated realizations, combined with fixed thresholding, can introduce small discontinuities along long fractures (*e.g.*, Fig. 5(a)). These breakpoints reduce line continuity and can cause the PHT to segment or miss portions of fractures, biasing the extracted fracture number statistics.

To verify that the LDM preserves flow-relevant connectivity, we compare dynamic responses from EDFM-based forward simulations. We randomly select 2000 training DFNs and 400 LDM-generated DFNs and simulate coupled flow-thermal and tracer transport for each realization. Figure 7 compares the ensemble percentiles of temperature and tracer mass fraction at the producer monitor (observation site 1 in Fig. 2). Solid curves show the median (P_{50}) and dashed curves show the P_{10} and P_{90} of the ensemble. The dynamic state distributions for the generated ensemble closely matches those of the training ensemble for both temperature and tracer. It demonstrates that the LDM generates DFNs that are not only statistically consistent but also dynamically consistent under the governing physics. This demonstrates that the latent parameterization retains the connectivity patterns that dominate advective heat and tracer transport.

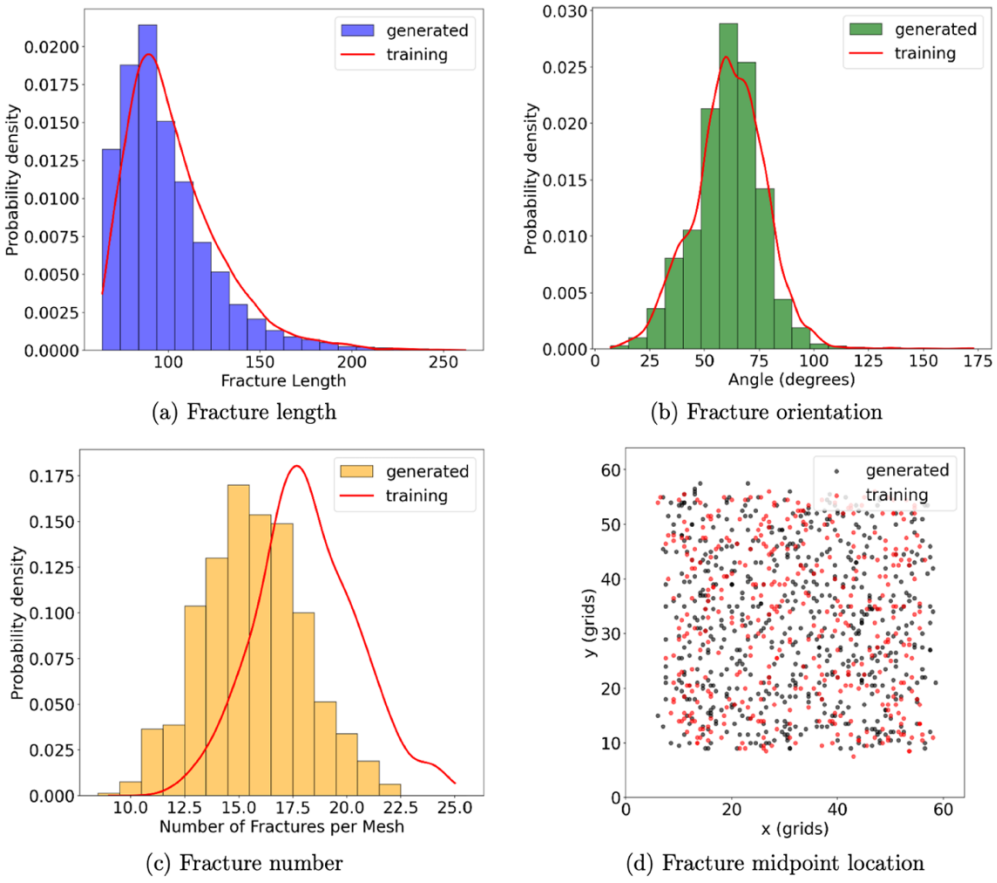


Figure 6: Statistical comparison between training DFNs and LDM-generated DFNs based on PHT extraction (Galamhos et al., 1999): (a) fracture length distribution; (b) fracture orientation distribution; (c) fracture-count distribution; (d) fracture midpoint locations. Results are computed from 200 training and 200 generated realizations.

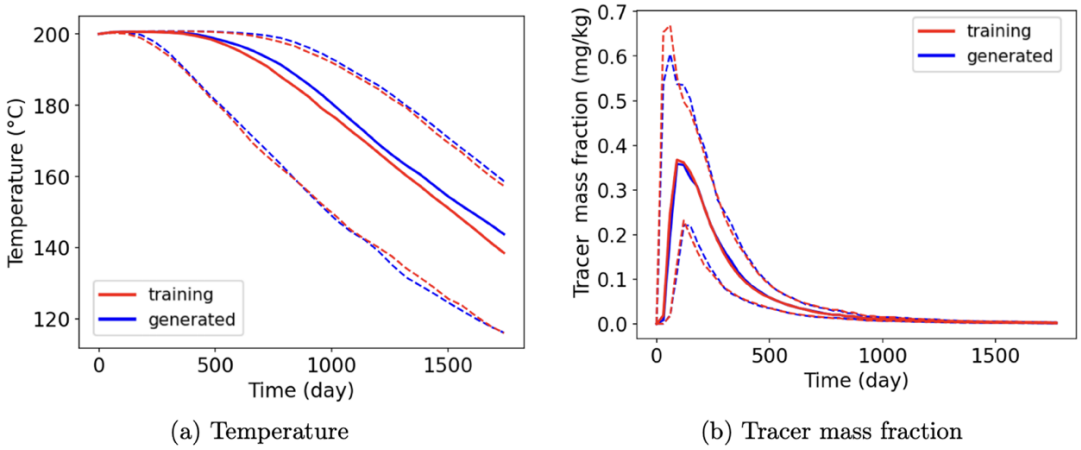


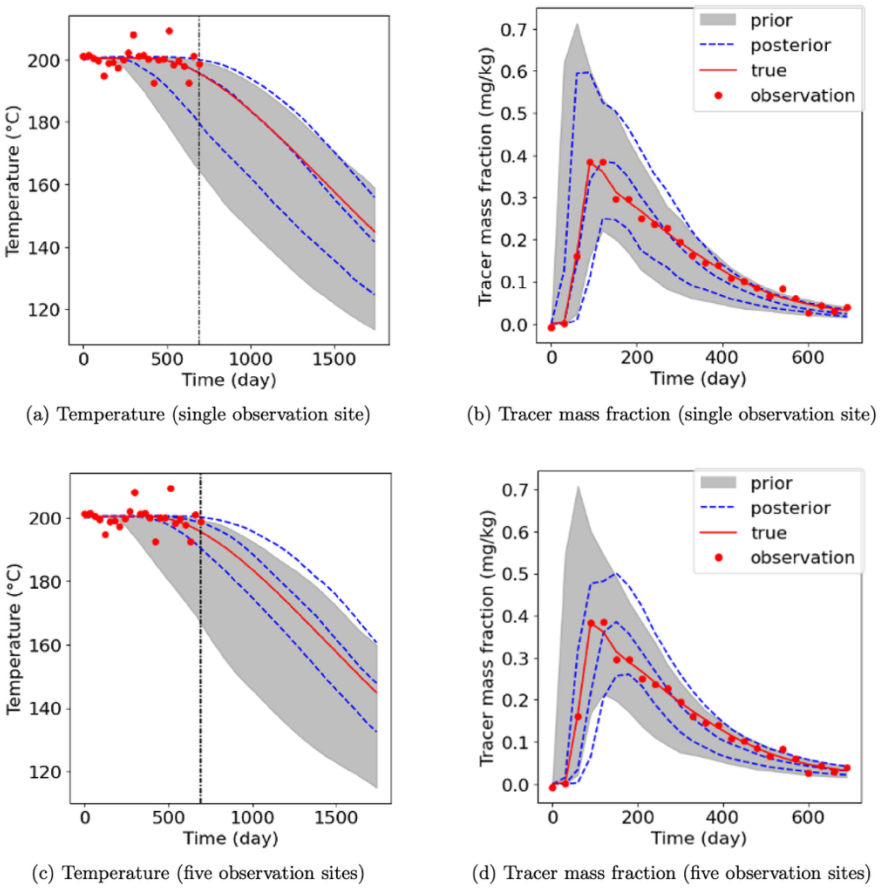
Figure 7: Comparisons of (a) temperature and (b) tracer mass fraction simulation results of training (red curves) and generated (blue curves) DFN realizations at the monitor at the producer. The lower, middle, and upper curves denote P_{10}, P_{50}, and P_{90} results.

5.2 History matching results using LDM-EDFM-ESMDA

We next evaluate the proposed LDM-EDFM-ESMDA workflow for DFN history matching. The inversion is conducted in the LDM latent space using ESMDA. For each realization and assimilation step, latent variables are decoded to a DFN realization, embedded into the structured grid via EDFM, and used in TRGB simulations to generate coupled flow-thermal and tracer responses. This setup updates DFN geometry through continuous latent variables, while preserving geological plausibility by the LDM prior and retaining flow-accurate forward modeling through EDFM.

History matching uses time series of temperature and tracer mass fraction over Day 0-720, sampled every 30 days (N_t=24). We consider two monitoring configurations (Fig. 2). The first is a single observation site at the producer, and the second is five observation sites distributed in the domain (four interior monitors and one at the producer). Synthetic observations are generated from a selected 'true' DFN realization by adding Gaussian noise with standard deviation of 2 °C for temperature and 0.01 *mg/kg* for tracer concentration. ESMDA is conducted with an ensemble size of N_r=200 and N_a=4 assimilation steps using inflation coefficients α = [4,4,4,4].

Figure 8 summarizes prior and posterior uncertainty in temperature and tracer time series at the producer. The gray shaded region denotes the prior P_{10}-P_{90} range, and the blue curves denote posterior P_{10}, P_{50}, and P_{90} percentiles. The 'true' response is shown in red, and red dots denote the observations. The vertical dash-dotted line marks the end of the historical period (Day 720). For both monitoring schemes, the posterior ensemble shows substantially reduced uncertainty and improved agreement with the 'true' responses in the historical period. The five-site configuration produces a further reduced posterior uncertainty ranges and improved prediction distributions. It demonstrates that richer spatial monitoring provides stronger constraints on DFN connectivity and therefore improves the identifiability of fracture geometry and the reliability of posterior forecasts.



(a) Temperature (single observation site)

(b) Tracer mass fraction (single observation site)

(c) Temperature (five observation sites)

(d) Tracer mass fraction (five observation sites)

Figure 8: History matching results for temperature and tracer mass fraction at the monitor at producer. Gray shaded areas show the prior $P_{10}-P_{90}$ range, blue dashed lines denote the posterior P_{10}, P_{50}, P_{90} curves, and red points and red curves represent observations and 'true' data. The vertical dash-dotted line shows the end of the history matching period: (a) temperature and (b) tracer for the single-site scheme; (c) temperature and (d) tracer for the five-site scheme.

Figure 9 compares the 'true' DFN with the prior and posterior ensembles using grid-level fracture-occurrence probability maps. For each realization, we construct a binary fracture indicator (fracture cells =1, matrix cells =0), and then compute the ensemble mean at each grid cell. The resulting field can be interpreted as the probability of fracture occurrence conditioned on the assimilated data. The prior mean in Fig. 9(b) reflects large geometric uncertainty. After data assimilation, the posterior probability fields in Figs. 9(c)-(d) become noticeably more organized, with coherent, high-probability features that align with the dominant connectivity patterns of the 'true' DFN. It demonstrates that the inversion framework successfully tuning the fracture networks toward geometries that generate flow and thermal responses matching observed temperature and tracer data. The five-site monitoring configuration consistently produces a more constrained posterior than the single-site case, since spatially distributed measurements provide more information about fracture geometry.

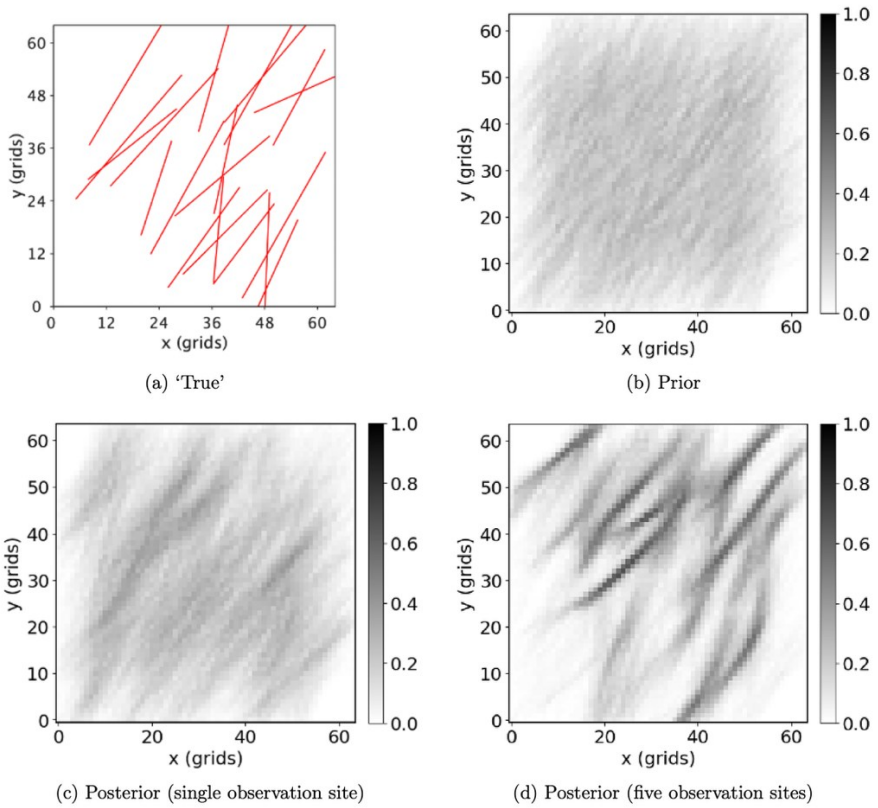


Figure 9: Prior and posterior fracture-occurrence probability maps from ESMDA: (a) 'True' DFN; (b) Grid-level mean of prior binary DFNs ensemble; (c) Grid-level mean of posterior binary DFNs ensemble using a single observation site; (d) Grid-level mean of posterior binary DFNs ensemble using five observation sites.

Figure 10 compares the spatial distributions of temperature (Day 720) for the 'true' DFN, the prior ensemble mean, and the posterior ensemble mean (one-monitor and five-monitor cases). Relative to the prior, the posterior mean fields exhibit transport patterns that more closely match the 'true' responses: the cooled region and tracer plume are more aligned with dominant preferential pathways, indicating that assimilation improves the inferred fracture-controlled advection structure. Remaining discrepancies are expected given the non-uniqueness of DFN inference (distinct networks can generate similar data) and the fact that ensemble means smooth high-contrast features. The posterior ensemble consistently recovers the key connectivity signatures that govern heat and tracer transport, leading to improved spatial predictions and reduced uncertainty in dynamic responses.

6. CONCLUDING REMARKS

In this work, we developed an integrated workflow that couples a latent diffusion model (LDM) for DFN parameterization, an embedded discrete fracture model (EDFM) for forward simulation on a structured grid, and the ensemble smoother with multiple data assimilation (ESMDA) for history matching. The central idea is to perform inversion in the continuous, low-dimensional latent space learned by the LDM, avoid fracture-by-fracture tuning in a discrete geometric space, while using EDFM to preserve fracture connectivity and enable repeated coupled flow-thermal simulations without re-meshing.

We demonstrated the framework on a synthetic 2D fractured EGS example defined on a 256 m × 256 m domain with 20 embedded fractures. Fracture lengths follow a truncated power-law distribution and orientations are sampled from a prescribed distribution. Coupled flow-thermal simulations and tracer transport are performed under using TRGB within an EDFM discretization with one injector and one producer. In this setting, the LDM provides an approximately Gaussian latent representation of DFN geometry, EDFM maps decoded DFNs into a simulation-ready fracture embedding, and ESMDA updates the latent ensemble by assimilating temperature and tracer data.

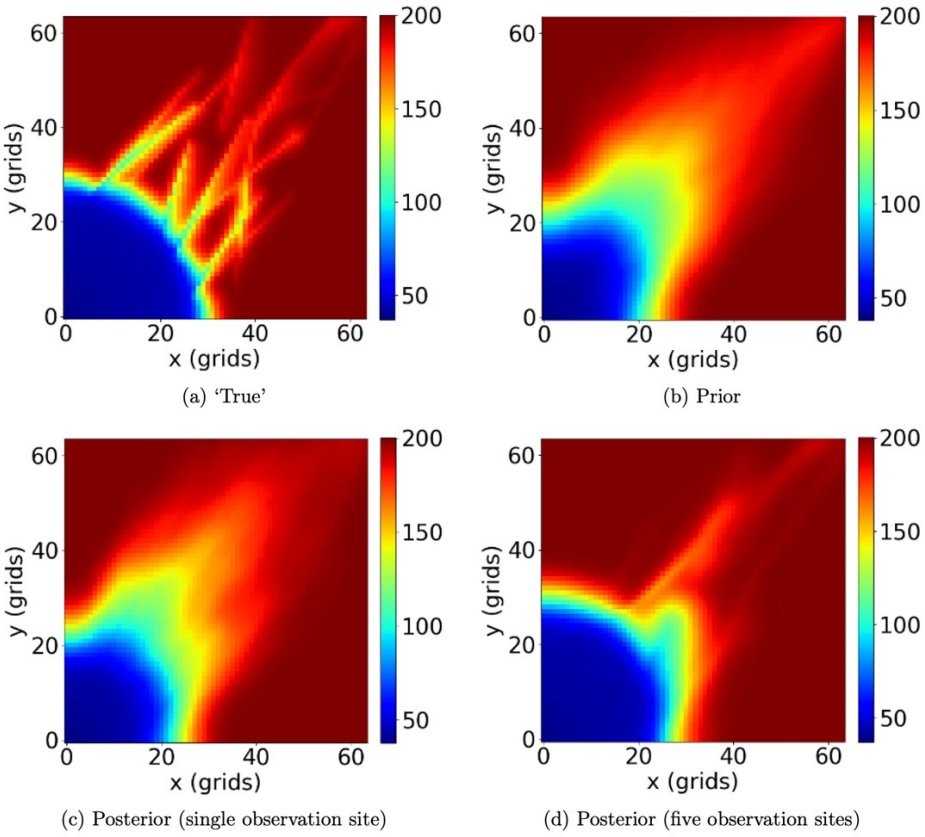


Figure 10: Prior and posterior temperature field at Day 720 from ESMDA: (a) 'True' temperature field; (b) Grid-level mean of prior temperature field ensemble; (c) Grid-level mean of posterior temperature field ensemble using a single observation site; (d) Grid-level mean of posterior temperature field ensemble using five observation sites.

The results show that the LDM generates DFN realizations that are consistent with the training ensemble in both geometric statistics and flow behaviors. The generated networks reproduce key fracture characteristics (*e.g.*, length and orientation distributions) and statistics of temperature and tracer responses. When combined with ESMDA, latent-space data assimilation substantially reduces uncertainty in both DFN geometry and reservoir predictions. Posterior ensembles exhibit improved agreement with the reference DFN, and the corresponding dynamic predictions show closer match to the observations and significant uncertainty reduction. Overall, the proposed LDM-EDFM-ESMDA workflow provides an efficient framework for DFN calibration and uncertainty quantification in fractured geothermal systems.

Several directions are worth pursuing. First, monitoring design should be optimized to maximize information gain for DFN inference, including the placement of monitoring locations and the selection of sampling windows. Second, this study focuses on geometry only. Extending the inversion to jointly infer fracture properties (*e.g.*, aperture) and matrix properties (*e.g.*, permeability) would better reflect the geological models. Third, extending the workflow to 3D DFNs with more realistic operational scenarios and data types is an important step toward practical deployment in field-scale EGS applications.

REFERENCES

Bao, J., Li, L., & Davis, A. (2022). Variational autoencoder or generative adversarial networks? A comparison of two deep learning methods for flow and transport data assimilation. *Mathematical Geosciences*, *54*(6), 1017-1042.

Canchumuni, S. W., Castro, J. D., Potratz, J., Emerick, A. A., & Pacheco, M. A. C. (2021). Recent developments combining ensemble smoother and deep generative networks for facies history matching. *Computational Geosciences*, *25*(1), 433-466.

Chen, G., Luo, X., Jiao, J. J., & Jiang, C. (2023). Fracture network characterization with deep generative model based stochastic inversion. *Energy*, *273*, 127302.

de Dreuzy, J.-R., Davy, P., & Bour, O. (2002). Hydraulic properties of two-dimensional random fracture networks following power law distributions of length and aperture. *Water Resources Research*, *38*(12), 12-11-12-19.

Di Federico, G., & Durlofsky, L. J. (2025a). Latent diffusion models for parameterization of facies-based geomodels and their use in data assimilation. *Computers & Geosciences*, *194*, 105755.

Di Federico, G., & Durlofsky, L. J. (2025b). Three-dimensional latent diffusion models for parameterizing and history matching facies systems under hierarchical uncertainty. *Mathematical Geosciences*, 1-33.

Emerick, A. A., & Reynolds, A. C. (2013). Ensemble smoother with multiple data assimilation. *Computers & Geosciences*, *55*, 3-15.

Galamhos, C., Matas, J., & Kittler, J. (1999). Progressive probabilistic Hough transform for line detection. Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149),

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, *33*, 6840-6851.

Jiang, S., & Durlofsky, L. J. (2021). Data-space inversion using a recurrent autoencoder for time-series parameterization. *Computational Geosciences*, *25*(1), 411-432.

Karimi-Fard, M., & Durlofsky, L. J. (2016). A general gridding, discretization, and coarsening methodology for modeling flow in porous formations with discrete geological features. *Advances in water resources*, *96*, 354-372.

Karimi-Fard, M., Durlofsky, L. J., & Aziz, K. (2004). An efficient discrete-fracture model applicable for general-purpose reservoir simulators. *SPE Journal*, *9*(02), 227-236.

Kazemi, H., Merrill Jr, L., Porterfield, K., & Zeman, P. (1976). Numerical simulation of water-oil flow in naturally fractured reservoirs. *SPE Journal*, *16*(06), 317-326.

Kim, H., Onishi, T., Chen, H., & Datta-Gupta, A. (2021). Parameterization of embedded discrete fracture models (EDFM) for efficient history matching of fractured reservoirs. *Journal of Petroleum Science and Engineering*, *204*, 108681.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*.

Li, L., & Lee, S. H. (2008). Efficient field-scale simulation of black oil in a naturally fractured reservoir through discrete fracture networks and homogenized media. *SPE Reservoir Evaluation & Engineering*, *11*(04), 750-758.

Lie, K.-A., & Møyner, O. (2021). *Advanced modelling with the MATLAB reservoir simulation toolbox*. Cambridge University Press.

Lie, K. A., Krogstad, S., Ligaarden, I. S., Natvig, J. R., Nilsen, H. M., & Skaflestad, B. (2012). Open-source MATLAB implementation of consistent discretisations on complex grids. *Computational Geosciences*, *16*(2), 297-322.

Lu, L., & Zhang, D. (2015). Assisted history matching for fractured reservoirs by use of Hough-transform-based parameterization. *SPE Journal*, *20*(05), 942-961.

Moinfar, A., Varavei, A., Sepehrnoori, K., & Johns, R. T. (2014). Development of an efficient embedded discrete fracture model for 3D compositional reservoir simulation in fractured reservoirs. *SPE Journal*, *19*(02), 289-303.

Moridis, G. (2014). User's manual of the TOUGH+ CORE code v1. 5: A general-purpose simulator of non-isothermal flow and transport through porous and fractured media.

Moridis, G., & Freeman, C. (2014). User's manual for the RealGasBrine v1. 0 option of TOUGH+ v1. 5: A code for the simulation of system behavior in gas-bearing geologic media.

Ringel, L. M., Jalali, M., & Bayer, P. (2021). Stochastic inversion of three-dimensional discrete fracture network structure with hydraulic tomography. *Water Resources Research*, *57*(12), e2021WR030401.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,

Somogyvári, M., Jalali, M., Jimenez Parras, S., & Bayer, P. (2017). Synthetic fracture network characterization with transdimensional inversion. *Water Resources Research*, *53*(6), 5104-5123.

Song, J., Meng, C., & Ermon, S. (2020). Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Tang, M., Liu, Y., & Durlofsky, L. J. (2021). Deep-learning-based surrogate flow modeling and geological parameterization for data assimilation in 3D subsurface flow. *Computer Methods in Applied Mechanics and Engineering*, *376*, 113636.

Teng, Z., Wu, H., Zhang, J., & Ju, X. (2024). A generative model of discrete fracture networks based on latent diffusion model. ARMA/DGS/SEG International Geomechanics Symposium,

Teng, Z., Wu, H., Zhang, J., Ju, X., & Qi, S. (2025). Generating high-fidelity discrete fracture networks from low-dimensional latent spaces using generative adversarial network. *International Journal of Rock Mechanics and Mining Sciences*, *196*, 106301.

Thomas, L. K., Dixon, T. N., & Pierson, R. G. (1983). Fractured reservoir simulation. *SPE Journal*, *23*(01), 42-54.

Wu, H., Jin, Z., Jiang, S., Tang, H., Morris, J. P., Zhang, J., & Zhang, B. (2024). Selecting appropriate model complexity: An example of tracer inversion for thermal prediction in enhanced geothermal systems. *Water Resources Research*, *60*(7), e2023WR036146.

Yao, M., Chang, H., Li, X., & Zhang, D. (2018). Tuning fractures with dynamic data. *Water Resources Research*, *54*(2), 680-707.

Zhou, C., Lyu, B., & Wang, Y. (2025). FracGen: Natural fracture networks reconstruction and upscaling using generative adversarial networks. *International Journal of Rock Mechanics and Mining Sciences*, *191*, 106116.