Computers & Geosciences 209 (2026) 106123

Contents lists available at ScienceDirect

Computers and Geosciences

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

Research paper

Uncertainty quantification using Hamiltonian Monte Carlo for structural geological modelling with implicit neural representations (INR)

Kaifeng Gao [a,*], Michael Hillier [b], Florian Wellmann [a,c]

[a] Institute of Computational Geoscience, Geothermics and Reservoir Geophysics (CG3), RWTH Aachen University, Aachen, 52074, Germany
[b] Geological Survey of Canada, Natural Resources Canada, Ottawa, ON K1A 0E8, Canada
[c] Fraunhofer Research Institution for Energy Infrastructures and Geothermal Systems, Bochum, 44801, Germany

ARTICLE INFO

Dataset link: https://doi.org/10.5281/zenodo.15973414

Keywords:
Implicit geomodelling
Implicit neural representations
Bayesian neural network
Uncertainty quantification
Hamiltonian Monte Carlo

ABSTRACT

Three-dimensional geological modelling is an essential tool for understanding subsurface features, supporting advanced exploration of natural resources, their sustainable development, and the identification of optimal locations for carbon storage. Recently, efficient neural network approaches have been developed to handle large datasets and to integrate diverse observations and prior knowledge into geological models. Previous work has demonstrated that neural networks are powerful tools for geological modelling, but quantifying uncertainty in their predictions remains an open issue. In this work, we address the uncertainty arising from both network parameters and observational data. We explore the full space of possible geological model realizations using a Hamiltonian Monte Carlo sampler, and quantify the uncertainty of predicted geological interfaces within a Bayesian neural network framework. Our experimental results demonstrate that the Hamiltonian Monte Carlo sampler effectively explores the posterior distribution in function space and quantifies the uncertainty of predicted geological interfaces for both a noise-free borehole dataset from the North Sea and a noisy dataset interpreted from geophysical well logs in Saskatchewan, Canada. We also apply the method to a simple faulting scenario involving a normal fault in flat stratigraphy. Furthermore, in comparison with the commonly used Monte Carlo dropout approach, the Hamiltonian Monte Carlo sampler exhibits superior accuracy in assessing epistemic uncertainty in a noise-free dataset. However, computational efficiency remains a potential challenge in large dataset and network.

1. Introduction

Three-dimensional geological modelling is important in various fields, including mineral exploration, oil and gas field development, groundwater management, and engineering geology. Common approaches to 3-D geological modelling are generally divided into two main types: explicit modelling and implicit modelling (Wellmann and Caumon, 2018). Explicit approaches (Caumon, 2010; Mallet, 2002; Sprague and de Kemp, 2005) typically construct each interface independently, often with a high level of geometric clarity and detail. This makes them suitable when precise geometric definitions are required. However, explicit approaches have limitations in modelling efficiency and adaptability, due to the extent of manual intervention and data processing required (Hillier et al., 2023). Implicit approaches can be further categorized into meshless approaches (Calcagno et al., 2008; Hillier et al., 2014; Lajaunie et al., 1997; de la Varga et al., 2019; Martin and Boisvert, 2017) and mesh-based approaches (Frank et al., 2007; Caumon et al., 2013; Laurent, 2016). A significant feature of implicit approaches is that multiple conformal interfaces can be modelled by the same scalar field, allowing a more automated and flexible modelling process. These approaches are particularly effective in handling incomplete and scattered data. However, despite their advantages, the underlying mathematical models can limit scalability when incorporating large amounts of geological data and knowledge (Hillier et al., 2023), as implicit representations require an interpolation of a scalar field one dimension higher than the modelled object - so, in typical 3-D geological modelling cases, the interpolation of a 3-D scalar field to represent 2-D geological interfaces.

Recently, neural networks have been applied to overcome this limitation and shown to be effective interpolators for implicit geological structural modelling (Bi et al., 2022; Guo et al., 2021; Hillier et al., 2021, 2023). These neural network frameworks can integrate diverse geological data and incorporate domain knowledge through various strategies (Wu et al., 2023), especially by using appropriate feature encoding (Chen et al., 2024). In previous studies, predictions of geological

* Corresponding author.
E-mail address: kaifeng.gao@cg3.rwth-aachen.de (K. Gao).

https://doi.org/10.1016/j.cageo.2026.106123
Received 27 July 2025; Received in revised form 27 November 2025; Accepted 14 January 2026
Available online 15 January 2026

structures from multiple runs of the neural network are shown to be similar, but not identical; however, the confidence in these predictions is typically not considered. Goodfellow et al. (2016) reported that training deep neural networks entails minimizing a highly non-convex loss function, the variability in predictions across runs arises from the random initialization of network parameters. Generally, beginning from different initial weights and biases, the neural network converges to a solution that may vary, resulting in different predictive geological structures. A common practice to improve reproducibility is to set a fixed random seed or use pretrained models. However, such deterministic result represents only one realization in the solution space. Therefore, quantifying uncertainty in neural network-based geological modelling remains an open challenge.

Over the past two decades, Bayesian inference and Monte Carlo simulation have been widely used to evaluate structural uncertainty in geological modelling. These probabilistic approaches generate ensembles of subsurface realizations that take into account both observations and prior geological knowledge (Tarantola, 2005; Caers, 2011). For instance, stochastic sampling has been used to evaluate alternative geometric configurations of geological structures and quantify the resulting uncertainty (Wellmann et al., 2010; Wellmann and Regenauer-Lieb, 2012). More recent developments include stochastic inversion strategies, where gravity data is inverted via Monte Carlo sampling to assess the variability of subsurface structures (Athens and Caers, 2022; Liang et al., 2023), and the use of the abstract graph Markov Chain Monte Carlo (MCMC) framework to explore a variety of possible geological interface geometries (Huang et al., 2023). These approaches demonstrate the value of probabilistic sampling in capturing the range of possible subsurface structures. However, although traditional Bayesian methods and MCMC methods are theoretically rigorous, they still have some limitations when applied to neural networks with high-dimensional parameters, such as slow mixing (Neal, 2011; Izmailov et al., 2021).

To address the critical gap in characterizing the inherent uncertainty of predicted geological models generated by existing AI-based methods, Bayesian Neural Networks (BNNs) can be employed. These networks learn a distribution over neural network parameters and have shown promise in characterizing the space of plausible geological models. Two common approaches for performing Bayesian inference are MCMC and variational inference (e.g., Monte Carlo dropout (Gal and Ghahramani, 2015) and deep ensembles (Lakshminarayanan et al., 2016)). In geomodelling research, Monte Carlo dropout and related techniques have already been explored to quantify uncertainty in deep learning models. Jordão et al. (2023) investigated Monte Carlo dropout and deep variational inference to assess the uncertainty of deep convolutional neural network models trained to predict ore type boundaries. Maldonado-Cruz and Pyrcz (2021) proposed a method to improve dropout frequency tuning in Monte Carlo dropout for uncertainty quantification in subsurface flow prediction. Feng et al. (2021) used the Monte Carlo dropout approach to quantify uncertainty in fault detection using a convolutional neural network. Bao et al. (2023) applied the Monte Carlo dropout in multilayer perceptron to predict reservoir thickness and quantify uncertainty.

The aforementioned methods approximate the posterior distribution in different ways. Deep ensembles train multiple neural networks with different initializations and aggregate their predictions to estimate predictive variability. Monte Carlo dropout uses stochastic dropout masks during inference to mimic Bayesian sampling. Although these approaches provide practical uncertainty estimates, they only approximate the true posterior and may underestimate the uncertainty (Li and Gal, 2017). Hamiltonian Monte Carlo (HMC) is a promising alternative for Bayesian inference in neural networks due to its ability to draw samples from the posterior more accurately than approximate approaches (Izmailov et al., 2021). Moreover, unlike classical MCMC algorithms, which rely on random walk exploration and struggle in high-dimensional spaces, HMC utilizes gradient information from the posterior distribution to produce lower rejection rates and faster convergence, making it better-suited for high-dimensional spaces (Betancourt, 2017; Benker et al., 2021). Recently, advancements in accelerated computing hardware (e.g., GPUs) and modern computational frameworks such as PyTorch and TensorFlow have made full-batch HMC feasible for exploring network parameter distributions and quantifying uncertainty in neural network predictions (Cobb, 2023; Izmailov et al., 2021).

Among the existing deep learning frameworks for geological modelling, GeoINR (Hillier et al., 2023) has an efficient modelling implementation with a low-dimensional parametrization. This aspect makes it a good basic choice for applying HMC for uncertainty estimation. In this work, we present such a combination of an HMC sampler in the GeoINR geomodelling framework to learn distributions over neural network parameters instead of fixed values. These distributions enable the exploration of parameter sets to characterize the space of predicted geological interfaces. To demonstrate the performance of HMC in uncertainty quantification within the GeoINR framework, we use a noise-free borehole dataset from the North Sea and a noisy dataset interpreted from geophysical well logs from Saskatchewan. We evaluate the effectiveness of the HMC sampler in exploring the posterior distribution over both network parameters (parameter space) and predicted scalar values in observational data (function space), then quantify the uncertainty in the predicted geological interfaces. Moreover, we compare modelling results from an HMC sampler with the commonly used Monte Carlo dropout approach.

The remainder of this paper is organized as follows: Section 2 introduces the related works of neural network architecture in the GeoINR and the HMC uncertainty quantification approaches in deep learning. Section 3 outlines the model setting for the HMC sampler in neural networks and the evaluation metrics. Section 4 presents the uncertainty quantification performance in two case studies. Section 5 discusses the characteristics of the used method. Finally, Section 6 provides the conclusion.

2. Materials and methods

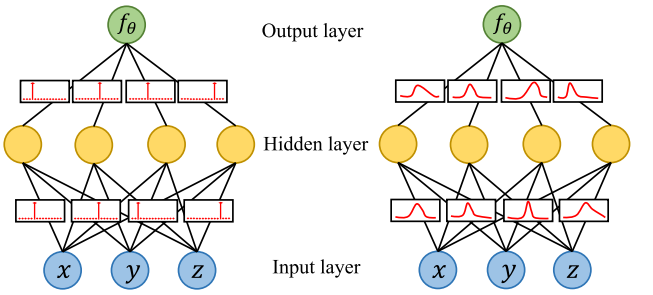
2.1. Geomodelling with implicit neural representations

Neural network approaches represent a shift from concept-driven modelling to data-driven modelling. Implicit Neural Representations (INR), also known as neural field interpolation, which draws on the neural radiance field (NeRF) (Mildenhall et al., 2020) technology in the field of 3-D reconstruction in computer vision, has been demonstrated as an effective geomodelling method and was first introduced under the term GeoINR by Hillier et al. (2023) and extended to fault representation by Gao and Wellmann (2025). GeoINR geomodelling framework utilizes neural networks to parametrize functions that directly map spatial coordinates to scalar or vector fields (see Fig. 1a), providing a more flexible framework for integrating prior knowledge into the model. This approach demonstrates significant promise in addressing efficiency and resolution dependency limitations in other neural network geomodelling techniques.

The function defining the INR network is given by

$$f_\theta(\mathbf{x}) = \phi_L\big(\mathbf{W}_L\,\phi_{L-1}\big(\cdots\,\phi_1(\mathbf{W}_1\mathbf{x}+\mathbf{b}_1)\big)+\mathbf{b}_{L-1}\big)+\mathbf{b}_L, \tag{1}$$

where $\mathbf{x} = (x, y, z)$ denotes a spatial coordinate, $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ is the observational dataset in which y_i is the scalar value associated with coordinate $\mathbf{x}_i$. The network parameters are $\theta = \{\mathbf{W}_i, \mathbf{b}_i\}_{i=1}^{L}$, and $\phi_i(\cdot)$ denotes the activation function at the i-th layer. Diverse data and model constraints can be incorporated into the modelling process by designing proper loss functions in GeoINR. The network parameters θ are optimized by minimizing the loss functions. Refer to Hillier et al. (2023), Gao and Wellmann (2025) for details of the loss function settings. The purpose of this work is to verify the effectiveness of the HMC sampler for uncertainty quantification within an INR-based geomodelling framework, using only the interface points as constraints.



(a) Standard neural network (b) Bayesian neural network

Fig. 1. Neural network architectures. (a) The GeoINR architecture uses fixed value for each trained network parameter; the input features are coordinates and the output is the corresponding predicted scalar value. (b) A Bayesian neural network with a distribution for each network parameter.

2.2. HMC-based BNN approach

2.2.1. Posterior distribution in BNN

BNNs are a class of neural networks that incorporate Bayesian inference to model uncertainty in predictions. In contrast to standard neural networks, which produce a single value for each network parameter, BNNs treat the network parameters as distributions (Fig. 1b). This is particularly useful for quantifying uncertainty in geomodelling using neural networks. Formally, the posterior distribution $p(\theta|D,f)$ of the network parameters θ given the observational data D and the neural network f is defined as (Lampinen and Vehtari, 2001):

$$p(\theta|D,f) = \frac{p(D|\theta,f)p(\theta|f)}{p(D|f)}, \tag{2}$$

where $p(D|\theta,f)$ is the likelihood of the parameters θ, $p(\theta|f)$ is the prior distribution of the parameters θ, and $p(D|f)$ is the marginal likelihood, which is defined as:

$$p(D|f) = \int_\theta p(D|\theta,f)p(\theta|f)d\theta. \tag{3}$$

However, the marginal likelihood is intractable due to the requirement to integrate over all network parameters, which can mean integration in a high-dimensional space. Consequently, conventional numerical integration methods such as quadrature become computationally prohibitive. A standard approach to overcome this issue is the use of sampling methods, such as MCMC. While being the gold standard for probabilistic inversion, it can be very slow to converge in practical applications. HMC methods mitigate these issues by using gradient information to efficiently explore the posterior space, making it a promising approach to better approximate the marginal likelihood.

2.2.2. HMC method

The HMC method is based on the principles of Hamiltonian mechanics, which describe the evolution of a system in terms of its position and momentum. Here, the position represents the parameters of interest (e.g., network parameters), while the momentum is an auxiliary variable introduced to improve sampling efficiency. The Hamiltonian is defined as:

$$H(\theta,p) = U(\theta) + K(p), \tag{4}$$

where $U(\theta)$ represents the potential energy, defined as the negative logarithm of the posterior probability of the parameters θ, formula as:

$$U(\theta) \propto \frac{1}{2\sigma^2}\sum_{i=1}^{N}\left(y_i - f_\theta(\mathbf{x}_i)\right)^2 + \frac{1}{2\lambda^2}\sum_j \theta_j^2, \tag{5}$$

where σ^2 is the noise of predicted scalar value in implicit geomodelling context, $f_\theta(\mathbf{x}_i)$ and y_i are predicted scalar value and the predefined scalar value respectively, λ^2 is the variance of the neural network parameter prior. For a detailed derivation of potential energy see Appendix A. The kinetic energy term, $K(p)$, is typically assumed to follow a Gaussian distribution and can be expressed as:

$$K(p) = \frac{1}{2}p^\top \boldsymbol{M}^{-1}p, \tag{6}$$

where M is the mass matrix - often set to the identity matrix.

In practice, we need to choose the initial parameters θ (i.e., the initial weights and bias of the neural network), as well as sample the initial momentum p from a Gaussian distribution. HMC proceeds by simulating the Hamiltonian dynamics using the following equations:

$$\frac{d\theta}{dt} = \frac{\partial H}{\partial p}, \frac{dp}{dt} = -\frac{\partial H}{\partial \theta}. \tag{7}$$

These equations are approximated using the leapfrog integration method (Brooks et al., 2011), which iteratively updates the network parameters and momentum as follows:

$$p_{t+\epsilon/2} = p_t - \frac{\epsilon}{2}\nabla_\theta U\left(\theta_t\right), \tag{8}$$

$$\theta_{t+\epsilon} = \theta_t + \epsilon\nabla_p K\left(p_{t+\epsilon/2}\right), \tag{9}$$

$$p_{t+\epsilon} = p_{t+\epsilon/2} - \frac{\epsilon}{2}\nabla_\theta U\left(\theta_{t+\epsilon}\right), \tag{10}$$

where ϵ is the step size of each updating time. In leapfrog integration, the gradient of the potential energy guides the update of the network parameters, driving them toward regions of higher posterior probability. Moreover, the step size ϵ and the leapfrog steps L can be adaptively determined using the NUTS algorithm in practice (Hoffman and Gelman, 2011). The probability of acceptance is calculated according to the Metropolis criterion (Metropolis et al., 1953):

$$\alpha = \min\left(1, \exp(H(\theta_0,p_0) - H(\theta^*,p^*))\right), \tag{11}$$

where $H(\theta_0,p_0)$ is the initial Hamiltonian system, and $H(\theta^*,p^*)$ is the new state of the Hamiltonian system after some updating steps. With probability α to accept the new state, otherwise, the current state is retained. The sampling procedure is repeated until sufficient samples have been collected.

Suppose we obtain a series of parameter samples $\{\theta^{(1)},\theta^{(2)},\ldots,\theta^{(T)}\}$, where T is the number of samples. A distribution over network parameters θ combined with a neural network architecture $f(\mathbf{x},\theta)$ induces a distribution of predicted results. The final predicted mean and variances are:

$$\mu[y^*|\mathbf{x}^*] \approx \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{x}^*,\theta^{(t)}), \tag{12}$$

$$\sigma^2[y^*|\mathbf{x}^*] = \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{x}^*,\theta^{(t)})f(\mathbf{x}^*,\theta^{(t)})^T - \mu[y^*|\mathbf{x}^*]\mu[y^*|\mathbf{x}^*]^T. \tag{13}$$

3. Combining HMC-BNN with geomodelling

The primary objective of this work is to integrate the HMC sampler into the GeoINR geomodelling framework and to quantify the uncertainty of predicted geological interfaces. Furthermore, the uncertainty information derived from these models should be assessed before being used for decision-making. The following sections describe how we address these two aspects.

3.1. Model definition

The neural network architecture employed in this study is the same as that of GeoINR (Hillier et al., 2023). The input features include the coordinates of observational or interpolated points, which are processed through several hidden layers to produce a scalar value for each point. To ensure stable network training, the input features are normalized to the range [−1, 1], and the predefined scalar values of the interfaces are assigned within the same range where younger interfaces

having higher scalar values. The Softplus activation function is used to ensure a smooth predictive scalar field.

The priors of neural network parameters play a critical role when employing HMC to explore their posterior distributions. GeoINR was implemented using the PyTorch programming framework, where the Kaiming normal distribution (He et al., 2015) is commonly used to initialize neural network parameters. It draws initial weights from a zero-mean Gaussian distribution with variance scaled by the number of neurons in each layer, thereby preserving the magnitude of the variance of the network parameters in the forward pass (Glorot and Bengio, 2010; He et al., 2015). By adopting this initialization as the prior for network parameters θ, we assume $\theta \sim \mathcal{N}\left(0, \sqrt{2/n_l}^2\right)$, where n_l refers to the input dimensionality of each neural network layer.

3.2. Evaluation metrics

3.2.1. Checking convergence of HMC sampler

The HMC sampler is used to generate a sequence of parameters from the posterior distribution of the network parameters. Since the prediction $f(\mathbf{x}, \theta)$ depends on these parameters, placing a distribution over θ induces a corresponding distribution over the predicted geological interfaces. Typically, deep learning involves a very high-dimensional space of parameters, making exploration of the exact posterior of each parameter (parameter space) computationally expensive. However, in using BNN, our goal is not to perform inference on the network parameters, rather to make accurate and well-calibrated predictions (Margossian and Gelman, 2024). Izmailov et al. (2021) reported that the HMC sampler efficiently explores a posterior distribution in function space that outperforms the posterior distribution obtained in parameter space. Similarly, Benker et al. (2021) stated that the acceptance of proposed network parameters depends on the resulting model fit to the training data. If the predicted result is accepted, the new parameter state becomes part of the Markov chain. We are primarily concerned with the posterior distribution in function space. Therefore, the following analysis of the posterior focuses on function space rather than the posterior and convergence of parameter space.

To evaluate the convergence of the HMC sampler, we use two metrics: drawn trajectory visualizations and $\hat{R}$ diagnostics (Gelman and Rubin, 1992). For the drawn trajectory visualizations, a good sample chain typically exhibits high-frequency fluctuations, indicating low autocorrelation between successive samples, i.e., the degree of correlation of the same variable between successive samples is lower, which is beneficial for uncertainty quantification. The $\hat{R}$ diagnostic compares the variance within each independent Markov chain to the variance between those chains (each sampling the same parameter) to assess whether they have all converged to the same target distribution. Approximate convergence is diagnosed when $\hat{R}$ is close to 1. The recommended proximity of each $\hat{R}$ to 1 depends on each dataset, but a general goal is to achieve $\hat{R} < 1.1$ (Brooks and Gelman, 1998). We use the open-source package ArviZ (Kumar et al., 2019) to compute $\hat{R}$ diagnostics.

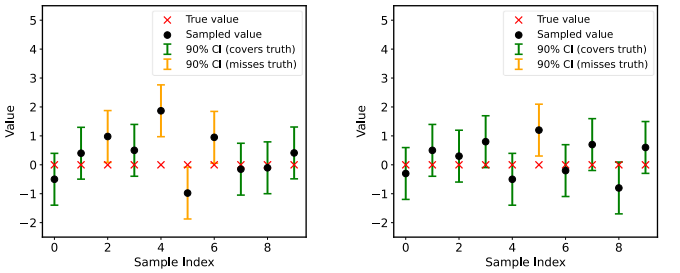
3.2.2. Checking distributions of uncertainty

To analyse the uncertainty in the location of geological interfaces (function space), we employ accuracy plots (Pyrcz and Deutsch, 2014; Maldonado-Cruz and Pyrcz, 2021) to evaluate uncertainties across multiple predicted geological interfaces for a given dataset. The accuracy and precision of an uncertainty distribution are based on the proportion of true values falling within different symmetric probability intervals of the sampled values.

We construct an indicator function $\xi(\mathbf{u}_i; \omega)$ and consider a range of credible intervals $\omega = \frac{j}{100}, j = 1, \ldots, 99$. For each interval, if the true value lies within the corresponding credible interval of the sampled values $f(\mathbf{u}_i)$, we set $\xi(\mathbf{u}_i; \omega) = 1$; otherwise $\xi(\mathbf{u}_i; \omega) = 0$. Then the proportion of true values within a given interval is calculated as the average of this indicator:

$$\overline{\xi(\omega)} = \frac{1}{n} \sum_{i=1}^{N} \xi(\mathbf{u}_i; \omega), \tag{14}$$

the uncertainty model is considered accurate when the average of the indicator function $\overline{\xi(\omega)} \geq \omega$. It is both accurate and precise when $\overline{\xi(\omega)} = \omega$. Conversely, it is inaccurate and imprecise when $\overline{\xi(\omega)} < \omega$. As an illustration of this concept, we consider a single interface point and display ten representative samples from the posterior distribution of its scalar value in Fig. 2. Each black dot corresponds to one such sample, while the coloured error bars represent the associated credible intervals. The red cross indicates the true value used for calibration. If, at a given credible level (e.g., 0.9 in Fig. 2), the true values fall within the 90% credible intervals approximately 90% of the time (i.e., 9 out of 10 samples), the uncertainty model is considered well-calibrated, as shown in Fig. 2b.



(a) Poorly calibrated at 90% credible level (b) Well calibrated at 90% credible level

Fig. 2. Example of uncertainty calibration at the 90% credible level. Each black dot represents one sample drawn from the posterior distribution of the scalar value at a single interface point, and the coloured bars show its corresponding 90% credible interval. The red crosses denote the true scalar value of this interface point. (a) Under-calibrated model: only 60% of true values lie within the 90% credible intervals. (b) Well-calibrated model: 90% of true values fall within the 90% credible intervals.

Following Pyrcz and Deutsch (2014), Maldonado-Cruz and Pyrcz (2021), we quantify overall accuracy and precision of the uncertainty as:

$$a(\omega) = \begin{cases} 1, & \text{if } \overline{\xi(\omega)} \geq \omega, \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

$$A = \int_0^1 a(\omega)\, d\omega, \tag{16}$$

$$P = 1 - 2 \int_0^1 a(\omega)\left(\overline{\xi(\omega)} - \omega\right) d\omega, \tag{17}$$

where $A = 1$ indicates maximum accuracy over all intervals, and $P = 1$ indicates maximum precision (the calibration curve coincides with the ideal diagonal).

The pseudo code for the comprehensive HMC-BNN method with uncertainty quantification is presented in Algorithm 1.

4. Results

4.1. Uncertainty in noise-free dataset

We first consider a noise-free borehole dataset from the North Sea to test the implementation of the HMC-BNN approach. The dataset was obtained from DINOloket[1], an open-source geological database developed by the TNO Geological Survey of the Netherlands. Specifically, we

[1] https://www.dinoloket.nl/en/subsurface-models/map (last access: 16 July 2025)

Algorithm 1 Comprehensive HMC-BNN with Uncertainty Quantification

Input: Dataset $D = \{(\mathbf{x}_i, y_i)\}$, Network architecture f, HMC hyper-parameters (step size ϵ (optional for NUTS), leapfrog steps L (optional for NUTS), number of samples T, warm-up steps, number of chains)
Output: Posterior samples $\{\theta^{(1)}, \theta^{(2)}, ..., \theta^{(T)}\}$, Predictions $\mu[y^*|\mathbf{x}^*], \sigma^2[y^*|\mathbf{x}^*]$, Convergence metric $\hat{R}$, Uncertainty metrics A, P
// Phase 1: Model Definition & Initialization (Section 3.1)
Normalize input features $\mathbf{x}$ to range $[-1, 1]$
Define activation function as Softplus
Initialize network parameters $\theta^{(0)}$
// Phase 2: HMC Sampling (Section 2.2)
Initialize $k = 0$
while $k < T$ **do**
    Sample momentum $p^k \sim \mathcal{N}(0, I)$
    Set $(\theta_0, p_0) = (\theta^k, p^k)$
    Perform leapfrog integration (using L steps if HMC; optional for NUTS)
    Update momentum and parameters:
        $p_i = p_{i-1} - \frac{\epsilon}{2}\nabla_\theta U(\theta_{i-1})$
        $\theta_i = \theta_{i-1} + \epsilon \nabla_p K(p_i)$
        $p_i = p_i - \frac{\epsilon}{2}\nabla_\theta U(\theta_i)$
    Compute acceptance probability: $\alpha = \min\left(1, \exp(H(\theta_0, p_0) - H(\theta_L, p_L))\right)$
    if accepted with probability α **then**
        $\theta^{k+1} = \theta_L$
    else
        $\theta^{k+1} = \theta^k$
    end if
    Save θ^{k+1} as posterior sample
    $k = k + 1$
end while
Discard warm-up samples; Collect posterior samples $\{\theta^{(1)}, \theta^{(2)}, ..., \theta^{(T)}\}$
Conduct forward pass for all collected θ sets
// Phase 3: Convergence Diagnostics (Section 3.2.1)
Visualize: Generate trace plots for function space and parameter space
Compute $\hat{R}$: Calculate Gelman-Rubin statistic for predictions $f(\mathbf{x}, \theta)$
// Phase 4: Prediction & Uncertainty Analysis (Section 3.2.2)
for test input $\mathbf{x}^*$ **do**
    Compute mean $\mu[y^*|\mathbf{x}^*]$ Eq. (12)
    Compute variance $\sigma^2[y^*|\mathbf{x}^*]$ Eq. (13)
end for
Calibration Check:
Compute total accuracy score A Eq. (16)
Compute total precision score P Eq. (17)

used the DGMdeep v5.0 model to select 13 boreholes (Fig. 3a) from a region of the North Sea. A total of 63 interface points corresponding to five geological units were used as modelling constraints. These units include the Upper North Sea Group (NU), North Sea Supergroup (N), Chalk Group (CK), Rijnland Group (KN), and basement.

In this case study, the input to the BNN consists of coordinates, while the output is a scalar value. The network architecture includes two hidden layers, each containing 128 neurons. We assume the borehole dataset is noise-free, and thus focus on the uncertainty from network parameters (i.e., epistemic uncertainty). This uncertainty relates to the interpolation uncertainty, also referred to as "Type 2" uncertainty in the context of geological modelling (see Mann et al., 1993; Wellmann et al., 2010). In practice, the potential energy of HMC is directly related to the predicted scalar value variance σ^2 (Eq. (5)). Consequently, selecting a value of σ^2 that is excessively small can lead to a lower acceptance rate (Eq. (11)). In this case study, we set the initial value of σ^2 to 10^{-6} and allow it to adapt to the dataset. Probabilistic inference is performed using the NUTS sampler implemented in the open-source library NumPyro (Phan et al., 2019), which extends HMC by automatically selecting both the step size and the number of leapfrog steps. Since the HMC sampler starts randomly and stabilizes after a certain number of steps, the warm-up phase is set to 400 steps. The number of samples should be determined based on model complexity, data size, and target posterior quantity. When the sample size is small, increasing the number of samples can significantly reduce the Monte Carlo standard error; however, when the samples reach stability, the marginal benefits of further increasing samples decrease. At this point, a more effective strategy is to tune the hyper-parameters of NUTS sampler to improve sampling efficiency (Margossian and Gelman, 2024). To obtain a stable and reliable measure of uncertainty while keeping the computational cost within reasonable limits, we sample 1500 sets of neural network parameters which are then used to generate corresponding scalar field predictions for modelling geological interfaces. From these scalar fields, iso-surfaces representing the target geological interfaces are extracted. Table 1 summaries the hyper-parameters of the model.

After performing HMC sampling with the parameter settings described above, the acceptance rate in this case study is 87%. Fig. 3b shows the predicted mean of the rock units, and Fig. 3c shows the mean of the geological interfaces. In geomodelling, information entropy is an effective method to visualize and quantify the uncertainty in geological models. We computed the entropy of the sampled geological interfaces following Wellmann and Regenauer-Lieb (2012), as shown in Fig. 3d. The accuracy plot of the resulting predictive uncertainty for this case study is shown in Fig. 4. The uncertainty accuracy and precision (by Eqs. (16)–(17)) are 0.7 and 0.99, respectively. This means that for roughly 70% of the credible intervals, the proportion of true values falling inside the predicted credible intervals met or exceeded

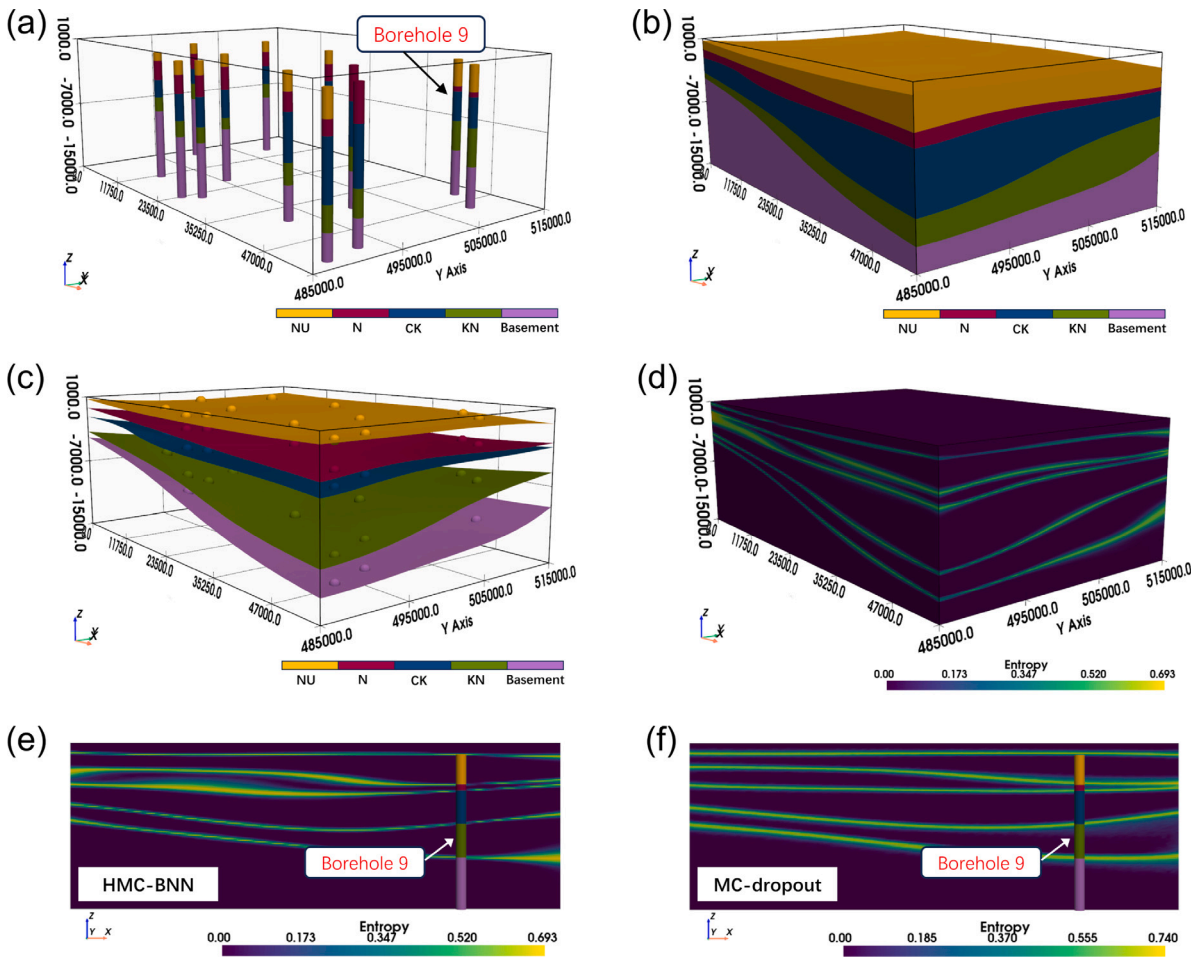


Fig. 3. Results of the North Sea borehole dataset. (a) Dataset visualization, with 13 boreholes and 5 geological units, including Upper North Sea Group (NU), North Sea Supergroup (N), Chalk Group (CK), Rijnland Group (KN), and basement from top to bottom. (b) Predicted mean of rock units. (c) Predicted mean of geological interfaces. (d) Entropy computed from 1500 sampled scalar fields. (e) Slice of the entropy result of the presented HMC-BNN method at the location of the borehole 9 along the y direction. (f) Slice of the entropy result of the Monte Carlo dropout method at the location of the borehole 9 along the y direction.

Table 1
Summary of hyper-parameter settings in the case studies.

Hyper-parameters (HMC-BNN)		North Sea dataset	Saskatchewan dataset
Neural network	Input features	3	3
	Number of hidden layers	2	2
	Dimension of hidden layers	128	128
	Output dimension	1	1
	Activation function	Softplus ($\beta = 1$)	Softplus ($\beta = 1$)
	Data normalization	$[-1, 1]$	$[-1, 1]$
HMC	Prior of network parameters	Kaiming normal	Kaiming normal
	Leapfrog steps	determined by NUTS	determined by NUTS
	Step size	determined by NUTS	determined by NUTS
	Prior of noise	$\mathcal{N}(0, 10^{-6})$	$\mathcal{N}(0, 10^{-2})$
	Warm-up steps	400	200
	Sampling size	1500	1500
	Number of chains	3	3

the expected coverage. At the same time, the calibration curve (blue line in Fig. 4) was nearly coincident with the ideal diagonal when the credible intervals were accurate, reflecting high precision in the uncertainty estimates. In this noise-free scenario, we only accounted for epistemic uncertainty, which led to a slight underestimation of overall uncertainty and therefore slightly narrower credible intervals. This effect is most evident in the higher probability interval, where the calibration curve falls below the perfect calibration line; as a result, the overall accuracy is 0.7. Note that in Fig. 4, the accuracy and precision metrics are computed by comparing the distributions of the predicted scalar-field values at the borehole interface locations with the predefined scalar values associated with those interfaces, as described in 3.2.2.

To compare HMC with Monte Carlo dropout, we set the neural network hyper-parameters the same for both methods and generate 1500 realizations with each. The result, shown in Fig. 3e-f, indicates that Monte Carlo dropout fails to distinguish uncertainty between observed and interpolated points when the data are noise-free, and the uncertainty along the predicted interface is nearly uniform. Another drawback of Monte Carlo dropout is its lower predictive accuracy

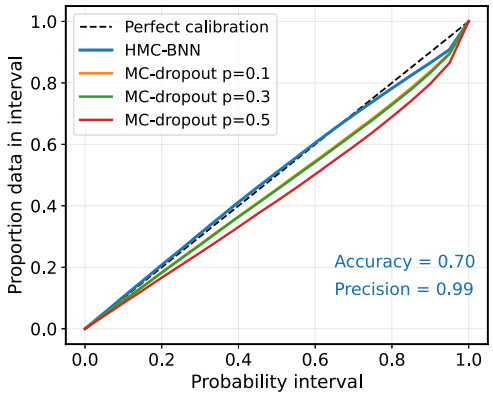


Fig. 4. Accuracy plot of predictive uncertainty for the North Sea model using HMC-BNN and Monte Carlo dropout (dropout rates 0.1, 0.3, and 0.5). The figure shows the proportion of data captured within predicted probability intervals against the ideal 1:1 calibration line, along with overall accuracy and precision metrics for the presented HMC-BNN method.

Table 2
Performance comparison of HMC and Monte Carlo dropout methods.

Method	MSELoss	Realizations/s
HMC	1.83×10^{-4}	3.15
MC-dropout (p=0.1)	1.10×10^{-3}	46.03
MC-dropout (p=0.3)	2.07×10^{-3}	46.16
MC-dropout (p=0.5)	4.09×10^{-3}	46.11

compared to HMC. In particular, the mean squared error loss (MSELoss) at constrained points is larger when using Monte Carlo dropout than when employing HMC, as shown in Table 2. This is due to random dropout of neurons during training and prediction, which effectively removes information and reduces accuracy. The calibration plot in Fig. 4 also indicates that HMC yields more reliable uncertainty quantification. However, because Monte Carlo dropout requires training only a single neural network and then performing multiple forward passes through that network to generate realizations, it is significantly more computationally efficient than HMC sampling.

We also analyse convergence in both parameter space and function space. Fig. 5 shows trace plots of three randomly chosen parameters in parameter space and three interface points (from borehole 9, see Fig. 3a) in function space. The weight traces in Fig. 5a show high autocorrelation because 1500 samples are insufficient to fully explore the high-dimensional parameter space, as many network parameter settings give the same network function due to symmetries (Neal, 1996). However, in using BNN, our goal is to make accurate and well-calibrated predictions. Therefore, we focus on summarizing the posterior distribution of function space, rather than the weight space. In this case study, the function space appears efficient and stable sampling, as shown in Fig. 5b.

The $\hat{R}$ diagnostic results for both parameter space and function space are presented in Fig. 6. The $\hat{R}$ values of the predicted scalar values at the interface points show that approximately 97% of the test points achieved $\hat{R} < 1.1$, indicating convergence of the HMC sampler in function space. However, the $\hat{R}$ values for the network parameters remain high, indicating that the HMC sampler has not yet converged in parameter space. Achieving accurate posterior distributions for every network parameter would therefore require increasing the number of samples and carefully tuning the HMC (NUTS) hyper-parameters to ensure full convergence. As the goal in this paper is to quantify uncertainty in the predictive scalar values, we primarily focus on function space.

4.2. Uncertainty in noisy dataset

In the previous case, we focused on epistemic uncertainty from network parameters. However, real field data (e.g., from well logs or seismic) are often noisy and this is an additional aspect of uncertainty that has to be considered in the geological model interpolation ("Type 1" uncertainty in Wellmann et al., 2010, following Mann et al., 1993). It is important to quantify uncertainty from both network parameters and data noise. Appendix B shows an example of estimating noise level with HMC. In this section, we use a noisy dataset from the Western Canadian Sedimentary Basin (WCSB) in Saskatchewan, Canada (Hillier et al., 2023). We select three conformable horizons and randomly sample 4500 interface points as constraints (3000 for training and 1500 for validation). These interface points are unevenly distributed and contain noise due to interpretation limits (see Fig. 7a).

The neural network architecture consists of two hidden layers, each layer containing 128 neurons, where the input are geological interface coordinates and the output is the scalar value for the implicit field. We assume homoscedastic noise of dataset, meaning each point has the same noise variance. Using a Gaussian prior $\mathcal{N}(0, l^2)$ with a standard deviation $l = 10^{-1}$ for the inherent scalar value variance within the given dataset. In the sampling process, the HMC sampler will sample this variance adaptively to the noisy data. We use the open-source software NumPyro to conduct probabilistic inference. The warm-up size is set to 200, the step size and the leapfrog steps of the HMC sampler were automatically determined by the build-in NUTS (Hoffman and Gelman, 2011) algorithm.

We sampled 1500 realizations. The predictive mean of the geological interfaces is shown in Fig. 7b, and the entropy of these interfaces in Fig. 7c; Fig. 7d provides a slice through the entropy volume. After sampling, the mean estimated noise variance is 0.218. Using this variance, we extracted the 95% credible interval for the predicted interface locations (Fig. 7e). These results demonstrate that the HMC-BNN approach has robust predictive performance on noisy datasets. The mean $\hat{R}$ diagnostic in function space is 1.0, indicating the HMC sampler has achieved convergence in function space. Using the 1500 validation points, we evaluate the uncertainty distribution; the predictive uncertainty has accuracy 0.83 (Eq. (16)) and precision 0.98 (Eq. (17)), as shown in Fig. 8, indicating good calibration of uncertainty.

5. Discussion

The results have demonstrated the HMC-BNN approach based on the aforementioned hypothesis reliably captures uncertainty arising both from neural network parameters and from data noise in applying the INR method to geomodelling. In this section, we compare HMC-BNN with a traditional MCMC algorithm, evaluate its performance in fault modelling, and discuss its limitations and potential future improvements.

5.1. Comparative study with traditional MCMC algorithm

We further compared the HMC sampler with a standard random-walk Metropolis–Hastings (RMH) sampler under identical conditions on the TNO borehole model. We ran a standard RMH sampler using a Gaussian random-walk proposal distribution, with its scale tuned to achieve the widely cited optimal acceptance rate of approximately 0.234 (Gelman et al., 1997; Roberts and Rosenthal, 2001). Autocorrelation is a standard diagnostic for assessing Markov chain mixing and effective exploration of the posterior (Cowles and Carlin, 1996). The lag-k autocorrelation, defined as the correlation between samples k iterations apart, is shown in Fig. 9, averaged over all interface points in function space for both samplers. As shown, HMC produces samples with substantially faster decay of autocorrelation, indicating far more effective exploration of the posterior. In contrast, RMH samples remain highly correlated even at large lags, despite operating at the

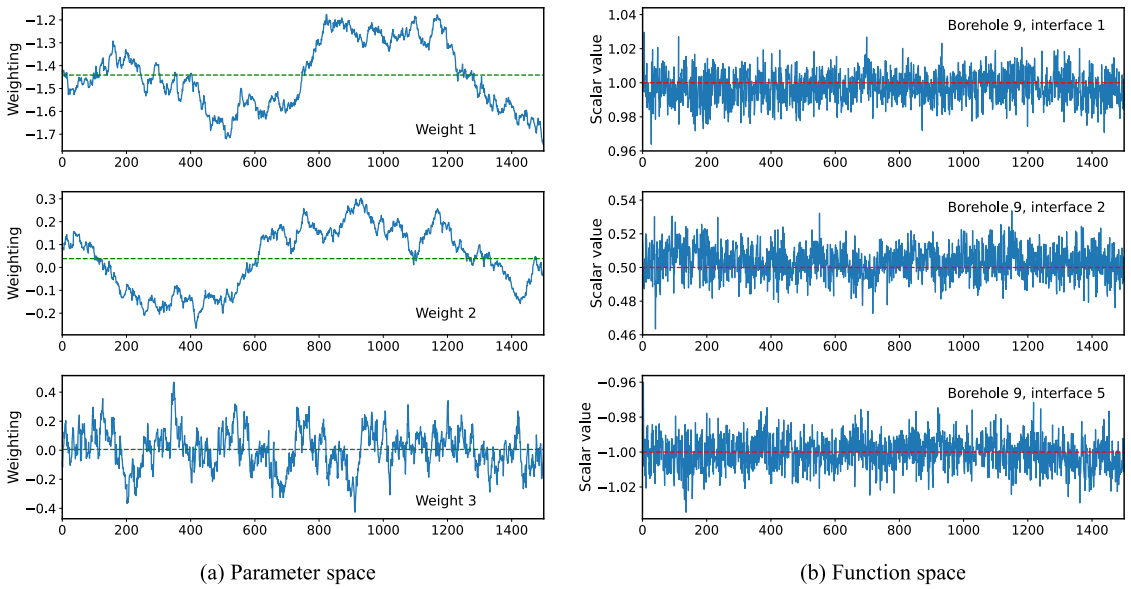


Fig. 5. The draw trajectory of weightings and predictive values of North Sea model. (a) Weighting draw trajectory of 3 randomly chosen parameters, the green dashed line represents the mean value of samples. (b) Predictive scalar value of 3 chosen interface points from borehole 9, the red dashed line represents the predefined scalar value.

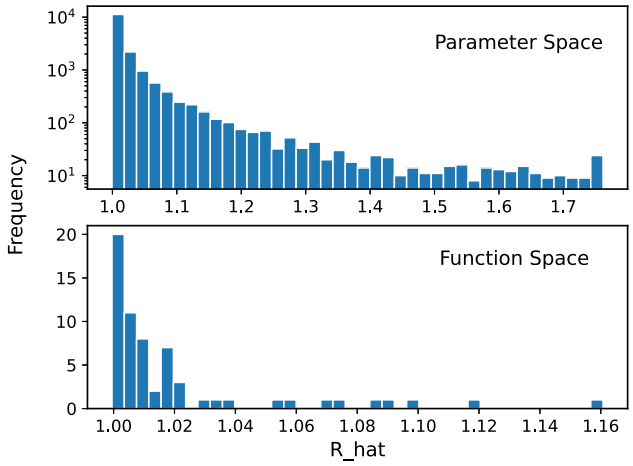


Fig. 6. The $\hat{R}$ diagnostic results of the North Sea model. In function space, approximately 97% of the testing points achieve $\hat{R} < 1.1$, which indicating the HMC sampler has converged in function space.

theoretically optimal acceptance rate. This behaviour is consistent with the well-known result that, in d-dimensional targets, RMH must use a proposal scale $\sigma = \mathcal{O}(d^{-1/2})$ to maintain a reasonable acceptance probability (Gelman et al., 1997), which forces the chain to take very small steps and explore only a small-radius neighbourhood of its current state in high-dimensional weight spaces.

These empirical findings align with the literature on HMC's exploration ability in high dimensions (Neal, 2011; Hoffman and Gelman, 2011). While RMH can be tuned to reduce autocorrelation by decreasing the proposal scale, doing so inevitably lowers the acceptance rate and further slows exploration (Liang et al., 2023). Moreover, because RMH lacks gradient guidance, its warm-up phase requires considerable manual tuning to locate a suitable region of parameter space, whereas HMC can traverse the posterior much more efficiently (Hoffman and Gelman, 2011).

5.2. Performance in modelling faults

To demonstrate the ability of the HMC-BNN approach in the faulted scenario, a synthetic model with two interfaces and one infinite fault is used. Note that in this faulted scenario we consider only uncertainty from network parameters (i.e., the interface constraint points are assumed noise-free and the fault location is fixed). The HMC-BNN hyper-parameters were set the same as in the case study in Section 4.1, except that the number of input features is 4, which has an additional fault feature beyond spatial coordinates. We sampled 1500 realizations and computed the mean predicted interfaces (Fig. 10a) and entropy (Fig. 10b). The uncertainty accuracy and precision of the interface positions are 1.0 and 0.78 respectively, indicating good calibration of uncertainty. Similar to the results of the first case study, the uncertainty produced by the Monte Carlo dropout method is less accurate and fails to distinguish between observed and interpolated points, resulting in an almost uniform uncertainty along the predicted interface (Figs. 10c and 10d).

5.3. Limitations and future work

In the Saskatchewan case study, we assumed a constant noise variance for all interface points primarily because the data consisted of interpreted well-log horizons where an overall uncertainty level could be reasonably assumed. Each borehole pick was considered to have roughly the same order of magnitude of interpretational error, and we did not have spatially varying uncertainty information for different points. Thus, a homoscedastic noise model was a practical and appropriate simplification for this dataset. However, our HMC-BNN framework can be extended to heteroscedastic noise. If different data points had different noise levels, one could model this by allowing a separate noise variance parameter for subsets of data, or by making the neural network output both a mean and a variance for each prediction and using negative log-likelihood as a loss function (Gal and Ghahramani, 2015; Kirkwood et al., 2022), effectively learning a noise model that could vary with different inputs. This was beyond our current scope, but heteroscedastic noise will be considered in future work.

In our experiments, we used a simple neural network architecture to predict layer-cake geological models, which are relatively straightforward structures. Table 3 shows the sampling times on North Sea dataset

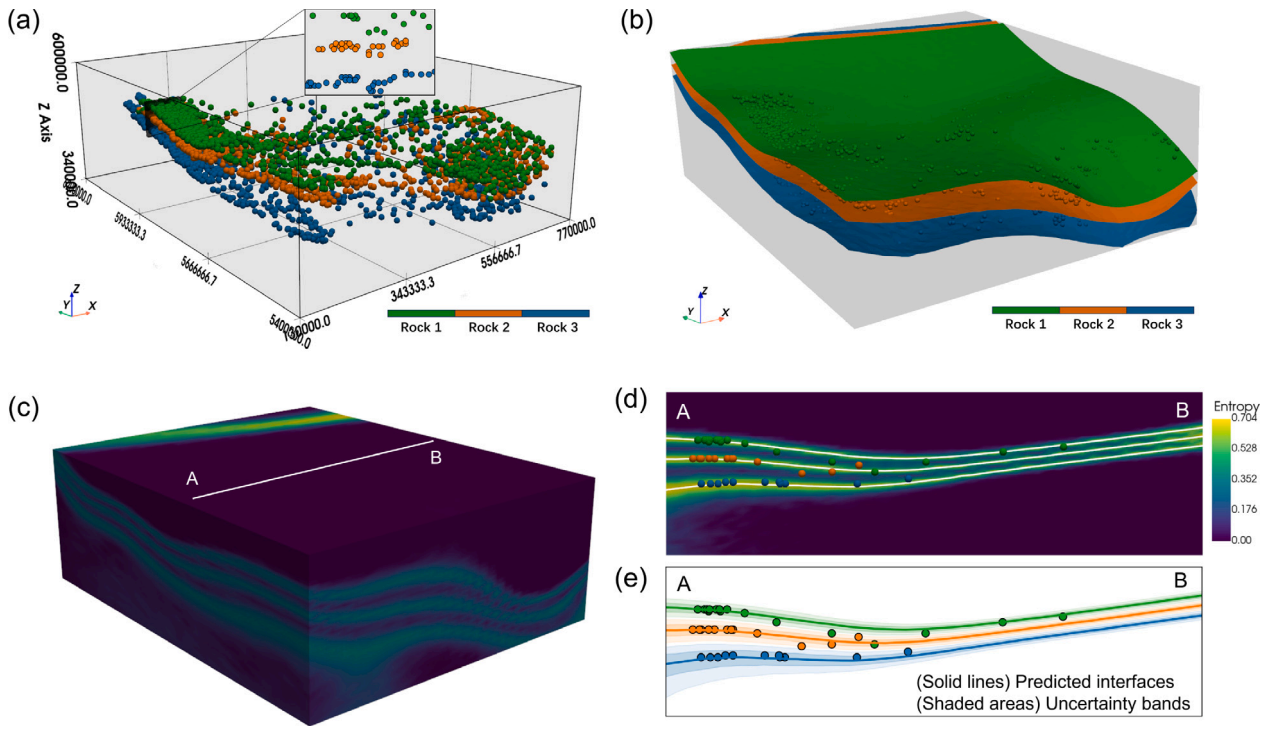


Fig. 7. Results of noisy dataset case study. (a) Scatter interface points for training and validation. (b) Predicted interfaces using 1500 sampled realizations. (c) Entropy of the sampled geological interfaces. (d) A slice of entropy result. The solid white lines represent the predicted interfaces and points are the corresponding interface observations on the section. (e) Uncertainty bands. The inner shaded area represents one standard deviation ($\pm 1\sigma$), the outer shaded area is two standard deviations ($\pm 2\sigma$).

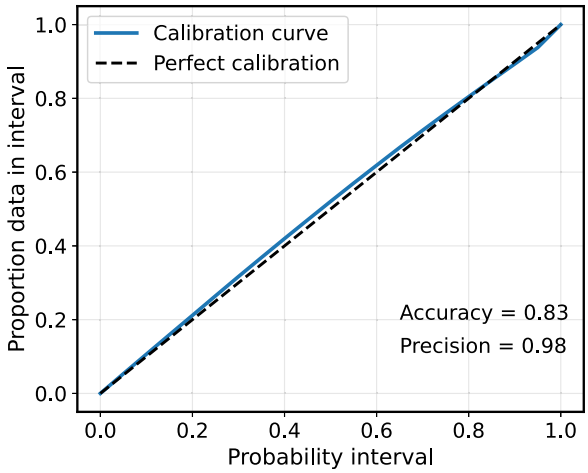


Fig. 8. Accuracy plot of predictive uncertainty for the noisy dataset case study.

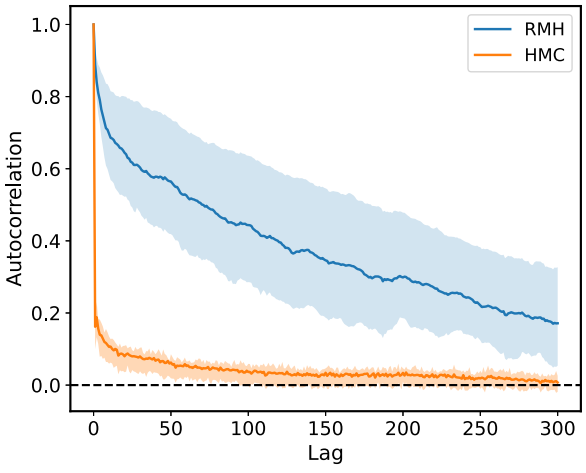


Fig. 9. Comparison of lag k autocorrelation for RMH and HMC. HMC shows fast decay and efficient mixing, whereas RMH exhibits persistently high autocorrelation. Shaded regions denote the 50% intervals across dimensions.

(used in Section 4.1) and Saskatchewan dataset (used in Section 4.2). All experiments were conducted on a Windows 11 desktop PC (Intel Core i7-12700K CPU and NVIDIA RTX 3070 GPU). Note that for both cases, we used NumPyro for inference, which (at the time of writing) does not support GPU acceleration on Windows. As geological models become more complex, larger networks are required, which greatly increases the dimensionality and computational cost. One potential solution is to use split HMC (Cobb and Jalaian, 2021), which partitions the parameter space to improve scalability. This approach could help mitigate the challenges of applying HMC to complex models and reduce computational cost.

Table 3
Sampling efficiency of the HMC sampler on the North Sea and Saskatchewan datasets.

Dataset	Training points	Per-realization time cost
North Sea dataset	63	0.317 s
Saskatchewan dataset	3000	6.636 s

6. Conclusion

In this work, we extend the implicit neural representation geomodelling framework (GeoINR) by incorporating HMC sampler to quantify,

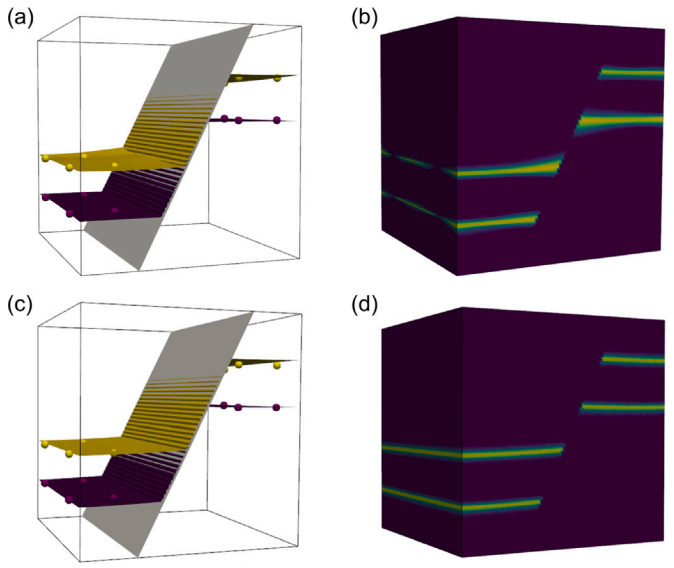


Fig. 10. Comparison of uncertainty quantification in the fault scenario. (a) Predicted geological interface obtained with the HMC-BNN method, together with interface constraint points. (b) Entropy of the sampled interfaces produced by HMC-BNN. (c) Predicted interface using the Monte Carlo dropout method. (d) Entropy of the sampled geological interfaces generated by Monte Carlo dropout.

for the first time, the prediction uncertainty of geological models generated by these neural networks. We focused on uncertainty arising from network parameters and observational data, exploring the full space of possible realizations of predicted geological interfaces. Two case studies demonstrated that the HMC-BNN method can generate high-quality posterior samples and well-calibrated uncertainty estimates. Moreover, the HMC-BNN method outperforms the commonly used Monte Carlo dropout method in the accuracy of uncertainty quantification under noise-free conditions. While computational cost increases with model complexity, our results highlight the practical feasibility of HMC-BNNs for geological modelling. Future work may investigate more scalable HMC variants and improved hardware acceleration to further enhance efficiency.

CRediT authorship contribution statement

Kaifeng Gao: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Funding acquisition, Conceptualization. **Michael Hillier:** Writing – review & editing, Methodology, Investigation, Conceptualization. **Florian Wellmann:** Writing – review & editing, Supervision, Resources, Conceptualization.

Code availability section

The source codes and data used in this paper are available for download from: https://doi.org/10.5281/zenodo.15973414.

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.

Acknowledgements

The first author acknowledges financial support from the China Scholarship Council (CSC) under Grant No. 202106400002. The authors gratefully acknowledge and appreciate the suggestions from Eric de Kemp from the Geological Survey of Canada and David Nathan from RWTH Aachen University. Additionally, we are thankful for all the discussions with colleagues from RWTH Aachen University. We would also like to thank two anonymous reviewer for their thorough review, which greatly improved the manuscript.

Appendix A. Derivation of the potential energy

The posterior distributions of network parameters is

$$P(\theta \mid D) \propto P(D \mid \theta)P(\theta), \tag{18}$$

where θ is the network parameters, $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ is the dataset. In the HMC, the potential energy $U(\theta)$ is defined as the negative logarithm of the posterior:

$$U(\theta) = -\log P(D \mid \theta) - \log P(\theta). \tag{19}$$

In context, the HMC sampler is used for regression task, assume the outputs $f_\theta(\mathbf{x}_i)$ are conditionally independent given $\mathbf{x}_i$ and network parameters θ, and each y_i follows a Gaussian distribution:

$$y_i \sim \mathcal{N}(f_\theta(\mathbf{x}_i), \sigma^2), \tag{20}$$

where σ is the predicted data noise. Thus, the likelihood for each data point is

$$P(y_i \mid \mathbf{x}_i, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(y_i - f_\theta(\mathbf{x}_i)\right)^2}{2\sigma^2}\right), \tag{21}$$

then the joint probability distribution of all constraint points is

$$P(D \mid \theta) = \prod_{i=1}^{N} P(y_i \mid \mathbf{x}_i, \theta), \tag{22}$$

taking the negative log-likelihood yields:

$$-\log P(D \mid \theta) = \frac{1}{2\sigma^2} \sum_{i=1}^{N} \left(y_i - f_\theta(\mathbf{x}_i)\right)^2 + \text{constant}. \tag{23}$$

For the prior term, assume a Gaussian prior for the network parameters:

$$\theta_j \sim \mathcal{N}(0, \lambda^2), \tag{24}$$

and the joint probability distribution of all network parameters is

$$P(\theta) = \prod_j \frac{1}{\sqrt{2\pi\lambda^2}} \exp\left(-\frac{\theta_j^2}{2\lambda^2}\right), \tag{25}$$

then the negative log-prior becomes

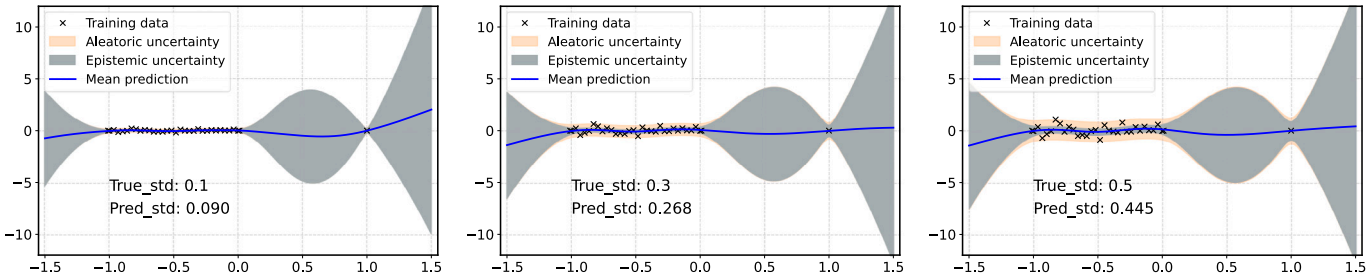
$$-\log P(\theta) = \frac{1}{2\lambda^2} \sum_j \theta_j^2 + \text{constant}. \tag{26}$$

In practice, the constant is ignored since it does not affect the HMC dynamics. Final form of the potential energy is

$$U(\theta) = \frac{1}{2\sigma^2} \sum_{i=1}^{N} \left(y_i - f_\theta(\mathbf{x}_i)\right)^2 + \frac{1}{2\lambda^2} \sum_j \theta_j^2. \tag{27}$$

Appendix B. Data noise estimation with HMC-BNN

In Eq. (5), the noise of predicted scalar value σ^2 is integrated into the potential energy term and can be treated as an additional learnable parameter. Therefore, it is feasible to sample the noise parameter concurrently with the network parameters using an HMC sampler. To demonstrate this, we conducted a simple curve-fitting experiment: we generated synthetic observation points with a target value $y = 0$ and



(a) Assigned data noise ($\sigma = 0.1$) (b) Assigned data noise ($\sigma = 0.3$) (c) Assigned data noise ($\sigma = 0.5$)

Fig. 11. Uncertainty quantification with the HMC sampler in different noise levels. The blue line is the predictive mean, the shaded area represents the 95% confidence intervals.

added Gaussian noise $\mathcal{N}(0, \sigma^2)$ to them. An HMC-BNN with two hidden layers, each consisting of 128 neurons, is employed, where the input is x and the output is predicted y. We adopt a Gaussian prior $\mathcal{N}(0, 1)$ for the learnable standard deviation of data noise and network parameters. The sample size is 1500, and three noise standard deviations of 0.1, 0.2, and 0.5 are tested.

The predicted uncertainties and estimated noise standard deviations obtained are shown in Fig. 11. The results demonstrate that the HMC sampler effectively estimates the noise levels. The 95% confidence intervals adequately encompass all observational data points. In regions with dense data distribution, epistemic uncertainty is substantially reduced, and aleatoric uncertainty predominates. In contrast, epistemic uncertainty dominates in regions lacking observational data.

Data availability

The data used in this paper can be found at https://doi.org/10.5281/zenodo.15973414.

References

Athens, N., Caers, J., 2022. Stochastic inversion of gravity data accounting for structural uncertainty. Math. Geosci. 54 (2), 413–436. http://dx.doi.org/10.1007/s11004-021-09978-2.

Bao, L.-L., Zhang, J.-S., Zhang, C.-X., Guo, R., Wei, X.-L., Jiang, Z.-L., 2023. A reliable Bayesian neural network for the prediction of reservoir thickness with quantified uncertainty. Comput. Geosci. 178, 105409. http://dx.doi.org/10.1016/j.cageo.2023.105409.

Benker, M., Furtner, L., Semm, T., Zaeh, M.F., 2021. Utilizing uncertainty information in remaining useful life estimation via Bayesian neural networks and Hamiltonian Monte Carlo. J. Manuf. Syst. 61, 799–807. http://dx.doi.org/10.1016/j.jmsy.2020.11.005.

Betancourt, M., 2017. A conceptual introduction to Hamiltonian Monte Carlo. http://dx.doi.org/10.48550/ARXIV.1701.02434.

Bi, Z., Wu, X., Li, Z., Chang, D., Yong, X., 2022. DeepISMNet: three-dimensional implicit structural modeling with convolutional neural network. Geosci. Model. Dev. 15 (17), 6841–6861. http://dx.doi.org/10.5194/gmd-15-6841-2022.

Brooks, S.P., Gelman, A., 1998. General methods for monitoring convergence of iterative simulations. J. Comput. Graph. Statist. 7 (4), 434–455. http://dx.doi.org/10.1080/10618600.1998.10474787.

Brooks, S., Gelman, A., Jones, G., Meng, X.-L., 2011. Handbook of Markov Chain Monte Carlo. CRC Press, New York, http://dx.doi.org/10.1201/b10905.

Caers, J., 2011. Modeling Uncertainty in the Earth Sciences. John Wiley & Sons.

Calcagno, P., Chilès, J., Courrioux, G., Guillen, A., 2008. Geological modelling from field data and geological knowledge. Phys. Earth Planet. Inter. 171 (1–4), 147–157. http://dx.doi.org/10.1016/j.pepi.2008.06.013.

Caumon, G., 2010. Towards stochastic time-varying geological modeling. Math. Geosci. 42 (5), 555–569. http://dx.doi.org/10.1007/s11004-010-9280-y.

Caumon, G., Gray, G., Antoine, C., Titeux, M.-O., 2013. Three-dimensional implicit stratigraphic model building from remote sensing data on tetrahedral meshes: Theory and application to a regional model of La Popa Basin, NE Mexico. IEEE Trans. Geosci. Remote Sens. 51 (3), 1613–1621. http://dx.doi.org/10.1109/TGRS.2012.2207727.

Chen, M., Qian, Z., Boers, N., Creutzig, F., Camps-Valls, G., Hubacek, K., Claramunt, C., Wilson, J.P., Nativi, S., Jakeman, A.J., Müller, R.D., Batty, M., Zhou, C., Chen, F., Wang, Q., Zhang, F., Barton, C.M., Strobl, J., Meadows, M., Ratti, C., Hess, P., Xu, Q., Zhang, Z., Gu, Q., Zhu, A.-X., Lin, H., Yuan, L., Lü, G., 2024. Collaboration between artificial intelligence and Earth science communities for mutual benefit. Nat. Geosci. 17 (10), 949–952. http://dx.doi.org/10.1038/s41561-024-01550-x.

Cobb, A.D., 2023. Hamiltorch: A PyTorch-based library for Hamiltonian Monte Carlo. In: Proceedings of Cyber-Physical Systems and Internet of Things Week 2023. ACM, San Antonio TX USA, pp. 114–115. http://dx.doi.org/10.1145/3576914.3587528.

Cobb, A.D., Jalaian, B., 2021. Scaling Hamiltonian Monte Carlo inference for Bayesian neural networks with symmetric splitting. In: Uncertainty in Artificial Intelligence. PMLR, pp. 675–685, URL https://proceedings.mlr.press/v161/cobb21a.html.

Cowles, M.K., Carlin, B.P., 1996. Markov chain Monte Carlo convergence diagnostics: a comparative review. J. Amer. Statist. Assoc. 91 (434), 883–904.

de la Varga, M., Schaaf, A., Wellmann, F., 2019. GemPy 1.0: open-source stochastic geological modeling and inversion. Geosci. Model. Dev. 12 (1), 1–32. http://dx.doi.org/10.5194/gmd-12-1-2019.

Feng, R., Grana, D., Balling, N., 2021. Uncertainty quantification in fault detection using convolutional neural networks. Geophysics 86 (3), M41–M48. http://dx.doi.org/10.1190/geo2020-0424.1.

Frank, T., Tertois, A.-L., Mallet, J.-L., 2007. 3D-reconstruction of complex geological interfaces from irregularly distributed and noisy point data. Comput. Geosci. 33 (7), 932–943. http://dx.doi.org/10.1016/j.cageo.2006.11.014.

Gal, Y., Ghahramani, Z., 2015. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. http://dx.doi.org/10.48550/ARXIV.1506.02142.

Gao, K., Wellmann, F., 2025. Fault representation in structural modelling with implicit neural representations. Comput. Geosci. 199, 105911. http://dx.doi.org/10.1016/j.cageo.2025.105911.

Gelman, A., Gilks, W.R., Roberts, G.O., 1997. Weak convergence and optimal scaling of random walk Metropolis algorithms. Ann. Appl. Probab. 7 (1), 110–120.

Gelman, A., Rubin, D.B., 1992. Inference from iterative simulation using multiple sequences. Statist. Sci. 7 (4), http://dx.doi.org/10.1214/ss/1177011136.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, M. (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. In: Proceedings of Machine Learning Research, vol. 9, PMLR, Chia Laguna Resort, Sardinia, Italy, pp. 249–256.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning. In: Adaptive computation and machine learning, The MIT Press, Cambridge, Mass.

Guo, J., Li, Y., Jessell, M.W., Giraud, J., Li, C., Wu, L., Li, F., Liu, S., 2021. 3D geological structure inversion from Noddy-generated magnetic data using deep learning methods. Comput. Geosci. 149, 104701. http://dx.doi.org/10.1016/j.cageo.2021.104701.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. http://dx.doi.org/10.48550/ARXIV.1502.01852.

Hillier, M.J., Schetselaar, E.M., de Kemp, E.A., Perron, G., 2014. Three-dimensional modelling of geological surfaces using generalized interpolation with radial basis functions. Math. Geosci. 46 (8), 931–953. http://dx.doi.org/10.1007/s11004-014-9540-3.

Hillier, M., Wellmann, F., Brodaric, B., de Kemp, E.A., Schetselaar, E., 2021. Three-dimensional structural geological modeling using graph neural networks. Math. Geosci. 53 (8), 1725–1749. http://dx.doi.org/10.1007/s11004-021-09945-x.

Hillier, M., Wellmann, F., de Kemp, E.A., Brodaric, B., Schetselaar, E., Bédard, K., 2023. GeoINR 1.0: an implicit neural network approach to three-dimensional geological modelling. Geosci. Model. Dev. 16 (23), 6987–7012. http://dx.doi.org/10.5194/gmd-16-6987-2023.

Hoffman, M.D., Gelman, A., 2011. The No-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. http://dx.doi.org/10.48550/ARXIV.1111.4246.

Huang, J., Deng, H., Chen, J., Li, N., Wang, J., Liu, Z., Mao, X., 2023. Assessing geometrical uncertainties in geological interface models using Markov chain Monte Carlo sampling via abstract graph. Tectonophysics 864, 230032.

Izmailov, P., Vikram, S., Hoffman, M.D., Wilson, A.G., 2021. What are Bayesian neural network posteriors really like?. http://dx.doi.org/10.48550/ARXIV.2104.14421.

Jordão, H., Sousa, A.J., Soares, A., 2023. Using Bayesian neural networks for uncertainty assessment of ore type boundaries in complex geological models. Nat. Resour. Res. 32 (6), 2495–2514. http://dx.doi.org/10.1007/s11053-023-10265-6.

Kirkwood, C., Economou, T., Pugeault, N., Odbert, H., 2022. Bayesian deep learning for spatial interpolation in the presence of auxiliary information. Math. Geosci. 54 (3), 507–531. http://dx.doi.org/10.1007/s11004-021-09988-0.

Kumar, R., Carroll, C., Hartikainen, A., Martin, O., 2019. ArviZ a unified library for exploratory analysis of Bayesian models in python. J. Open Source Softw. 4 (33), 1143. http://dx.doi.org/10.21105/joss.01143.

Lajaunie, C., Courrioux, G., Manuel, L., 1997. Foliation fields and 3D cartography in geology: Principles of a method based on potential interpolation. Math. Geol. 29 (4), 571–584. http://dx.doi.org/10.1007/BF02775087.

Lakshminarayanan, B., Pritzel, A., Blundell, C., 2016. Simple and scalable predictive uncertainty estimation using deep ensembles. http://dx.doi.org/10.48550/ARXIV.1612.01474.

Lampinen, J., Vehtari, A., 2001. Bayesian approach for neural networks—review and case studies. Neural Netw. 14 (3), 257–274. http://dx.doi.org/10.1016/S0893-6080(00)00098-8.

Laurent, G., 2016. Iterative thickness regularization of stratigraphic layers in discrete implicit modeling. Math. Geosci. 48, 811–833. http://dx.doi.org/10.1007/s11004-016-9637-y.

Li, Y., Gal, Y., 2017. Dropout inference in bayesian neural networks with alpha-divergences. In: International Conference on Machine Learning. PMLR, pp. 2052–2061.

Liang, Z., Wellmann, F., Ghattas, O., 2023. Uncertainty quantification of geologic model parameters in 3D gravity inversion by Hessian-informed Markov chain Monte Carlo. Geophysics 88 (1), G1–G18.

Maldonado-Cruz, E., Pyrcz, M.J., 2021. Tuning machine learning dropout for subsurface uncertainty model accuracy. J. Pet. Sci. Eng. 205, 108975. http://dx.doi.org/10.1016/j.petrol.2021.108975.

Mallet, J.-L., 2002. Geomodeling. In: Applied Geostatistics, Oxford University Press, Cary, Incorporated.

Mann, C.J., Davis, J., Herzfeld, U., 1993. Uncertainty in geology. Comput. Geol.—25 Years Prog. 20, 241–254.

Margossian, C.C., Gelman, A., 2024. For how many iterations should we run Markov chain Monte Carlo?. http://dx.doi.org/10.48550/arXiv.2311.02726.

Martin, R., Boisvert, J.B., 2017. Iterative refinement of implicit boundary models for improved geological feature reproduction. Comput. Geosci. 109, 1–15. http://dx.doi.org/10.1016/j.cageo.2017.07.003.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of state calculations by fast computing machines. J. Chem. Phys. 21 (6), 1087–1092. http://dx.doi.org/10.1063/1.1699114.

Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R., 2020. NeRF: Representing scenes as neural radiance fields for view synthesis. http://dx.doi.org/10.48550/ARXIV.2003.08934.

Neal, R.M., 1996. Bayesian learning for neural networks. In: Lecture Notes in Statistics, vol. 118, Springer New York, New York, NY, http://dx.doi.org/10.1007/978-1-4612-0745-0.

Neal, R.M., 2011. MCMC using Hamiltonian dynamics. http://dx.doi.org/10.1201/b10905.

Phan, D., Pradhan, N., Jankowiak, M., 2019. Composable effects for flexible and accelerated probabilistic programming in NumPyro. http://dx.doi.org/10.48550/arXiv.1912.11554.

Pyrcz, M., Deutsch, C.V., 2014. Geostatistical Reservoir Modeling, second ed. Oxford University Press, New York, Oxford.

Roberts, G.O., Rosenthal, J.S., 2001. Optimal scaling for various Metropolis-Hastings algorithms. Statist. Sci. 16 (4), 351–367.

Sprague, K.B., de Kemp, E.A., 2005. Interpretive Tools for 3-D structural geological modelling Part II: Surface design from sparse spatial data. GeoInformatica 9 (1), 5–32. http://dx.doi.org/10.1007/s10707-004-5620-8.

Tarantola, A., 2005. Inverse Problem Theory and Methods for Model Parameter Estimation. SIAM.

Wellmann, F., Caumon, G., 2018. 3-d Structural geological models: Concepts, methods, and uncertainties. In: Advances in Geophysics, vol. 59, pp. 1–121. http://dx.doi.org/10.1016/bs.agph.2018.09.001.

Wellmann, J.F., Horowitz, F.G., Schill, E., Regenauer-Lieb, K., 2010. Towards incorporating uncertainty of structural data in 3D geological inversion. Tectonophysics 490 (3–4), 141–151. http://dx.doi.org/10.1016/j.tecto.2010.04.022.

Wellmann, J.F., Regenauer-Lieb, K., 2012. Uncertainties have a meaning: Information entropy as a quality measure for 3-D geological models. Tectonophysics 526–529, 207–216. http://dx.doi.org/10.1016/j.tecto.2011.05.001.

Wu, X., Ma, J., Si, X., Bi, Z., Yang, J., Gao, H., Xie, D., Guo, Z., Zhang, J., 2023. Sensing prior constraints in deep neural networks for solving exploration geophysical problems. Proc. Natl. Acad. Sci. 120 (23), e2219573120. http://dx.doi.org/10.1073/pnas.2219573120.