USING DEEP GENERATIVE NEURAL NETWORKS TO ACCOUNT FOR MODEL ERRORS IN MARKOV CHAIN MONTE CARLO INVERSION

A PREPRINT

Shiran Levy
Institute of Earth Sciences
University of Lausanne
Lausanne, Switzerland

Jürg Hunziker
Institute of Earth Sciences
University of Lausanne
Lausanne, Switzerland

Eric Laloy
Belgian Nuclear Research Center
Mol, Belgium

James Irving
Institute of Earth Sciences
University of Lausanne
Lausanne, Switzerland

Niklas Linde
Institute of Earth Sciences
University of Lausanne
Lausanne, Switzerland

Original publication: Geophys. J. Int., 23 September 2021
DOI: https://doi.org/10.1093/gji/ggab391

ABSTRACT

Most geophysical inverse problems are nonlinear and rely upon numerical forward solvers involving discretization and simplified representations of the underlying physics. As a result, forward modeling errors are inevitable. In practice, such model errors tend to be either completely ignored, which leads to biased and over-confident inversion results, or only partly taken into account using restrictive Gaussian assumptions. Here, we rely on deep generative neural networks to learn problem-specific low-dimensional probabilistic representations of the discrepancy between high-fidelity and low-fidelity forward solvers. These representations are then used to probabilistically invert for the model error jointly with the target geophysical property field, using the computationally-cheap, low-fidelity forward solver. To this end, we combine a Markov-chain-Monte-Carlo (MCMC) inversion algorithm with a trained convolutional neural network of the spatial generative adversarial network (SGAN) type, whereby at each MCMC step, the simulated low-fidelity forward response is corrected using a proposed model-error realization. Considering the crosshole ground-penetrating radar traveltime tomography inverse problem, we train SGAN networks on traveltime discrepancy images between: (1) curved-ray (high fidelity) and straight-ray (low fidelity) forward solvers; and (2) finite-difference-time-domain (high fidelity) and straight-ray (low fidelity) forward solvers. We demonstrate that the SGAN is able to learn the spatial statistics of the model error and that suitable representations of both the subsurface model and model error can be recovered by MCMC. In comparison with inversion results obtained when model errors are either ignored or approximated by a Gaussian distribution, we find that our method has lower posterior parameter bias and better explains the observed traveltime data. Our method is most advantageous when high-fidelity forward solvers involve heavy computational costs and the Gaussian assumption of model errors is inappropriate. Unstable MCMC convergence due to nonlinearities introduced by our method remain a challenge to be addressed in future work.

Keywords Inverse theory · Neural networks · Ground penetrating radar · Probability distributions · Hydrogeophysics

1 Introduction

Bayesian inversion treats model parameters as random variables that are constrained by prior probability density functions and noise-contaminated data through a likelihood function (Tarantola, 2005; Gelman et al., 2013). The

Bayesian framework is flexible in that it allows accounting for uncertainties due to inaccurate or incomplete descriptions of the underlying physics of the problem, as well as for errors related to the measurement process. We refer to the former as model errors (Kaipio and Somersalo, 2007) because they describe inaccuracies in the forward modeling used to connect physical properties to observable data, while other authors have used the term "theoretical error" (Tarantola et al., 1982) in a similar context. Model errors are notoriously difficult to quantify, particularly when the forward problem at hand is nonlinear. Their magnitudes and correlation patterns can be highly complex and variable throughout the model parameter space, and deriving an appropriate statistical description of them is therefore challenging. At the same time, relying on accurate state-of-the-art forward solvers with minimal model errors is not always practical as they are generally computationally expensive, which becomes particularly problematic when the forward response has to be calculated many times. Surrogate models (also referred to as proxy models or low-fidelity models) implying an approximation or a simplified representation of the underlying physical process offer an attractive alternative provided that one can adequately account for the resulting model errors. Model errors are commonly an order of magnitude or so larger than measurement uncertainties (Tarantola et al., 1982; Kaipio and Somersalo, 2007; Hansen et al., 2014). Therefore, ignoring them might lead to severe bias, artifacts and over-confident results (Brynjarsdóttir and O-Hagan, 2014; Hansen et al., 2014).

Early pioneering work on model errors was conducted by Kennedy and O'Hagan (2001). They represent model errors as a Gaussian process (GP) that is conditioned at locations in the model parameter space where the model errors are known. The general applicability of this method for geoscientific inverse problems of high dimensional and multivariate nature remains unclear (Linde et al., 2017) even if some promising applications exist (Xu and Valocchi, 2015; Xu et al., 2017). Most approaches dealing with model errors involve building a statistical model of the discrepancy between a high-fidelity forward model and a cheaper, less-accurate counterpart. Some of these methods formulate the likelihood function such that prior knowledge about the mean and covariance of the model errors is incorporated (Kaipio and Somersalo, 2007; Cui et al., 2011; Hansen et al., 2014). Despite their proven value, the Gaussian assumptions made in these methods might be problematic when confronted with non-Gaussian priors, non-Gaussian observational noise and nonlinear problems. Traditionally, model errors are learned by evaluating modeling discrepancies using samples from the prior, yet, recent adaptive approaches in which the model error description is updated based on samples from the posterior region has shown important improvements (Cui et al., 2011; Calvetti et al., 2014). Other approaches for dealing with model errors involve estimating and removing them from the residual data term before calculating the likelihood function (Köpke et al., 2018, 2019). In such methods, the residuals are projected onto an orthogonal model-error basis, which is constructed either during the inversion using a dictionary-based K-nearest-neighbour approach, or before the inversion using principal component analysis (PCA) conducted on a suite of model-error realizations. The dynamic model error estimation methods of Cui et al. (2011), Calvetti et al. (2014) and Köpke et al. (2018) enjoy local statistics of model errors in regions of high posterior density; however, they do require occasional runs of a high-fidelity forward solver during the inversion. Another approach is presented by Rammay et al. (2019) who perform joint inversion of the model parameters and error-model in the context of reservoir history matching. They use PCA basis functions to parameterize the error-model and infer for the PCA coefficients during inversion.

Over the past decade, the use of machine learning (ML) in geophysical applications has become increasingly popular as a result of continuing growth in computational resources and numerous breakthroughs in ML research (Giannakis et al., 2019; Bergen et al., 2019; Dramsch, 2020; Yu and Ma, 2020). Deep learning models, an extension to machine learning models, can be trained to produce an amortized data-based alternative to expensive physics-based models (Tripathy and Bilionis, 2018; Tang et al., 2020; Jin et al., 2020). Nonetheless, these models are problem specific and their accuracy may vary depending on availability of training data and their ability to generalize. Furthermore, as surrogate models they still suffer from some degree of model errors when compared to the high-fidelity model which they aim to approximate. Here we give several examples of machine learning applications addressing model errors. The approach of Xiao (2019), in analogy to the GP approach of Kennedy and O'Hagan (2001), uses GP regression, an ML algorithm, to learn a set of error response functions associated with a low-fidelity flow model. The error response functions predict a set of parameters that through proper orthogonal decomposition are projected into the full error space and used to correct the low-fidelity model. Seillé and Visser (2020) utilize regression trees in order to learn a dimensionality discrepancy model (DDM) predicting the model errors associated with using 1D instead of 3D magnetotelluric modeling. The DDM is then used to define a likelihood function that is used within a reversible-jump Markov chain Monte Carlo (MCMC) procedure (Green, 1995). Sun et al. (2019) apply convolutional neural networks (CNN's) describing spatial and temporal discrepancies between land surface model (LSM) predictions and observations from the gravity recovery and climate experiment (GRACE). Their neural network combining three CNN architectures receives as an input the LSM output as well as additional predictors (precipitation and temperature) and in return outputs the mismatch between the LSM and GRACE observations. Their study shows an increased correlation between corrected LSM and observed data, thereby, highlighting the potential of deep-learning to improve geo-scientific models over different spatiotemporal scales. Machine and deep-learning algorithms have also been proven efficient for parameterizing geological models (Laloy et al., 2017, 2018; Mosser et al., 2020). Laloy et al. (2018) parameterized model realizations using a spatial

generative adversarial network (SGAN) and integrated the generating part of the network within an MCMC routine. In this type of neural network, a nonlinear transformation is learned using training images. The image space, representing the high-dimensional space on which forward simulations are performed, is connected to a lower dimensional space (latent space) through a series of nonlinear transformations in the form of convolution operations. The inversion is performed with respect to this lower-dimensional representation. Given the notable reduction in the number of inferred parameters, the spatial nature of the network and the fast generation of model realizations, the SGAN-parameterization was able to significantly improve the MCMC inversion performance compared with sequential geostatistical resampling (Mariethoz et al., 2010; Ruggeri et al., 2015).

In this study, we use SGANs to learn a low-dimensional parameterization of model errors associated with a low-fidelity forward solver. A notable characteristic of the SGAN is its localized nature, allowing for perturbations in a specific region of the image space following a perturbation in one of the latent parameters. Our approach takes advantage of spatial correlation within model-error realizations to transform the high-dimensional model-error space (same dimension as the data space) into a lower-dimensional latent space. We train two separate deep generative neural networks, one for the subsurface model parameters and the other for the model errors. Then, we perform MCMC inversion on the latent parameters to infer the joint posterior distribution of both. We consider numerical simulations in the context of crosshole ground-penetrating radar (GPR) traveltime tomography and test our method with synthetic data generated by either a (1) curved-ray (eikonal) or (2) finite difference time-domain full-waveform forward solver. The inversion on the other hand is performed using a low-fidelity straight-ray forward solver. The aim of our approach is to account for discrepancies in the modelling process when one replaces an expensive, high-fidelity solver with a cheap, less accurate solver to speed up the inversion process. By doing so, we hope to reduce the bias caused by using low-fidelity solvers while allowing for an efficient MCMC inversion. Note that the cheap low-fidelity solver could, in principle, also be a deep-learning based forward solver that was trained on the same database of high-fidelity forward solvers. We compare our approach against two alternative inversion approaches that also rely on the same low-fidelity forward solver, one where model errors are ignored and the other where they are approximated as Gaussian. For the case of the synthetic data being generated with the eikonal solver, we also compare with inversion results obtained without any model errors, that is, when using the eikonal solver as forward model in the MCMC inversions.

2 Methods

Our approach to account for model errors involves three main steps: (1) database preparation, (2) SGAN training, and (3) MCMC inversion. The database preparation involves setting up the database on which the neural networks for the subsurface model parameters and model error are trained. During training, information about the trained parameters of the generative network is given at regular intervals. The stage (generator iteration) at which training data are retained to generate realizations for subsequent inversions is chosen according to statistical measures as well as visual inspection. Finally, the deep generative neural networks are integrated into an MCMC inversion algorithm. Below we describe each of the three stages in detail in the context of the considered crosshole GPR traveltime tomography inverse problem.

2.1 Database preparation

2.1.1 Multi-Gaussian model database

The training image (TI) used as a basis to describe the spatial structure of our subsurface model-parameter prior is a 2500×2500 pixels, (250×250 m) anisotropic, multi-Gaussian, geostatistical realization with a variance of 1 and mean of zero. It was generated by Pirot et al. (2017) based on the geostatistical analysis of sediments at the Boise Hydrogeophysical Research Site conducted by Barrash and Clemo (2002). We split the TI into two parts: a segment of size 2250×2500 pixels, which is used for training the SGAN, and a segment of size 250×2500 pixels, from which we select the reference models used in our inversion examples. The training is performed on small patches $\mathbf{X}_{\mathbf{\Phi}}$ of pre-defined size which are randomly cropped from the segment of the TI intended for training. The porosity field $\mathbf{\Phi}$ is then computed from the multi-Gaussian realizations using the lognormal transformation

$$\mathbf{\Phi} = \exp(\mathbf{X}_{\mathbf{\Phi}} \times 0.22361 - 1.579) \tag{1}$$

in Pirot et al. (2017).

2.1.2 Crosshole GPR simulations and model-error database

In a crosshole GPR experiment, an electromagnetic impulse is emitted from a source antenna located in one borehole and registered in a receiver antenna positioned in an adjacent borehole. To create a model-error database of first-arrival

travel time residuals, we perform crosshole GPR numerical simulations based on the $\mathbf{\Phi}$-realizations described in subsection 2.1.1 using the low- and high-fidelity forward solvers, which we denote by g^{LF} and g^{HF}, respectively. The numerical simulations are performed on slowness $\mathbf{s}$ (1/velocity) fields, therefore, the porosity field of the subsurface model-parameter realizations must be converted to a slowness field. This can be done via the following relationships (Pride, 1994):

$$\boldsymbol{\kappa}_b = \mathbf{\Phi}^m \kappa_w + (1 - \mathbf{\Phi}^m)\kappa_s, \tag{2}$$

and

$$\mathbf{s} = \frac{\sqrt{\boldsymbol{\kappa}_b}}{c}, \tag{3}$$

where κ_w and κ_s are the water and rock dielectric constants, m is the cementation exponent, $\boldsymbol{\kappa}_b$ (b stands for "bulk") is the effective dielectric constant of the medium and c is the speed of light in vacuum. We ignore petrophysical prediction uncertainty related to scatter in the petrophysical relationship (Brunetti and Linde, 2018) and assume the petrophysical parameters to be known. Following Pirot et al. (2017), we set κ_w to be 81, κ_s to 6 and m to 1.48.

Assuming that the forward solver g^{HF} (HF stands for high-fidelity) describes perfectly the crosshole GPR experiment, we have:

$$\mathbf{d} = g^{HF}(\mathbf{s}) + \boldsymbol{\epsilon}, \tag{4}$$

where $\mathbf{d}$ represents the observed traveltime data corresponding to slowness parameters $\mathbf{s}$ with observational noise $\boldsymbol{\epsilon}$. The proxy solver g^{LF} gives rise to a model error $\eta(\mathbf{s})$:

$$\mathbf{d} = g^{LF}(\mathbf{s}) + \eta(\mathbf{s}) + \boldsymbol{\epsilon}, \tag{5}$$

describing the discrepancy between the two solvers for each source-receiver pair:

$$\eta(\mathbf{s}) = g^{HF}(\mathbf{s}) - g^{LF}(\mathbf{s}). \tag{6}$$

To test our method, we consider two different model errors for the crosshole GPR traveltime tomography problem. In both test cases, we use a straight-ray solver denoted by g^{SR} as our low-fidelity solver g^{LF}. In the first test case, we consider a finite difference approximation of the eikonal equation by Podvin and Lecomte (1991) as the high-fidelity model, such that $g^{HF} = g^{eikonal}$ and the model error is $\eta^{eikonal-SR}(\mathbf{s}) = g^{eikonal}(\mathbf{s}) - g^{SR}(\mathbf{s})$. In the second test case, the high-fidelity model is based on a finite difference time-domain scheme (FDTD) (Irving and Knight, 2006), such that $g^{HF} = g^{FDTD}$ and the model error is $\eta^{FDTD-SR}(\mathbf{s}) = g^{FDTD}(\mathbf{s}) - g^{SR}(\mathbf{s})$. We note that our method is almost fully amortized as the computationally expensive high-fidelity solver is only used prior to inversion to create the model-error database and, in a synthetic example such as ours, the data (observed data) that are to be inverted.

From the FDTD simulations, the first-arrival travel times are automatically chosen by identifying the first maximum of the signal and subtracting the time delay between the source wavelet's initiation and first peak. Due to an underlying infinite-frequency assumption, ray-based approaches (straight ray and eikonal solvers) provide the same arrival times in 2D and 2.5D media. This is not the case for FDTD simulations leading to important time shifts in the 2D FDTD first-break picks compared to the ray-based solvers. We correct this phase shift by applying a reversed geometrical correction to that found in Ernst et al. (2007), effectively performing a 2D to 2.5D correction of the FDTD data:

$$\hat{E}(\mathbf{x}_{trn}, \mathbf{x}_{rec}, \boldsymbol{\omega}) = \frac{E(\mathbf{x}_{trn}, \mathbf{x}_{rec}, \boldsymbol{\omega})}{\sqrt{\frac{2\pi T(\mathbf{x}_{trn}, \mathbf{x}_{rec})}{-i\boldsymbol{\omega}\bar{\kappa}\mu_0}}}, \tag{7}$$

where $E(\mathbf{x}_{trn}, \mathbf{x}_{rec}, \boldsymbol{\omega})$ and $\hat{E}(\mathbf{x}_{trn}, \mathbf{x}_{rec}, \boldsymbol{\omega})$ are the signal in the frequency domain before and after applying the correction from 2D to 2.5D, respectively, for source and receiver locations $\mathbf{x}_{trn}$ and $\mathbf{x}_{rec}$. Here, $T(\mathbf{x}_{trn}, \mathbf{x}_{rec})$ are the picked arrival times based on signal $E(\mathbf{x}_{trn}, \mathbf{x}_{rec})$ in the time domain, $\boldsymbol{\omega}$ refers to the angular frequency of the signal, $\bar{\kappa}$ is the mean dielectric constant of the medium, μ_0 is the magnetic permeability in vacuum and $i^2 = -1$. After correction, arrival times were repicked on the corrected signals.

2.2 Generative adversarial networks

In a fully connected neural network (see Goodfellow et al. (2016) for details), a single neuron with weight vector $\mathbf{w}$, bias term b, and input vector $\mathbf{x}$ can be represented as

$$h(\mathbf{x}; \mathbf{w}, b) = \varphi(\sum_{i=1}^{N_x} w_i x_i + b), \tag{8}$$

where φ is a nonlinear transformation referred to as the activation function. In a convolutional neural network applied to an image, a single pixel at location (u, v) in the output feature map $\mathbf{F}$ is a result of a convolution between a kernel $\mathbf{K}$ of size $N_H \times N_W$ and a sub-region of the same size in the input image $\mathbf{I}$:

$$\mathbf{F}_{u,v} = \varphi(\sum_{j=1}^{N_W} \sum_{i=1}^{N_H} \mathbf{K}_{i,j} \mathbf{I}_{u+i,v+j} + b). \tag{9}$$

The full feature map is the collection of pixels resulting from convolution operations over different locations in the input image. A convolutional layer produces multiple feature maps, each being a result of convolution between the input image and a different filter. All filters in a layer share the same dimensions, but contain different weights. A deep convolutional network is a network in which several convolutional layers are sequentially stacked. As the number of layers and neurons within layers increases, the ability of the network to express complex functions increases.

A generative adversarial network (GAN; Goodfellow et al., 2014) is a convolutional neural network (CNN), in which training is a zero-sum game between a generator G and a discriminator D. The GAN seeks to minimize a distance between the distribution P_r of the training data and the distribution P_g of the data created by the generator G. The generator input is usually a low-dimensional latent vector $\mathbf{z}$ drawn from a uniform distribution $\mathcal{U}(-1, 1)$ or a standard normal distribution $\mathcal{N}(0, 1)$, and the output is an image $\tilde{\mathbf{X}}$. Jetchev et al. (2016) extended the GAN into a spatial-GAN (SGAN), where the input $\mathbf{Z}$ becomes a 2D (later extended to 3D by Laloy et al., 2018) tensor of $n \times m$ ($\times q$) dimensions such that a perturbation in one tensor element corresponds to a change in a specific region of the output image $\tilde{\mathbf{X}}$. The input to the discriminator D is either an image $\tilde{\mathbf{X}}$ from the generator distribution P_g or an image $\mathbf{X}$ from the training distribution P_r (see Fig. 1). At each training iteration, a batch of generated images $\tilde{\mathbf{X}}$, and a batch of training images $\mathbf{X}$ are interchangeably fed into the discriminator and, according to the loss function in use, they are either classified as 0 (fake) or 1 (true), or are given a score. As opposed to other types of deep generative networks (e.g. variational autoencoders), training enforces only the distribution on $\tilde{\mathbf{X}}$ (P_g) to approximate the distribution on $\mathbf{X}$ (P_r) while the prior on $\mathbf{Z}$ is simply assigned such that, for example, all draws during training are drawn from a uniform distribution $\mathcal{U}(-1, 1)$. For an enhanced stability of training and better general performance, we use the Wasserstein loss function (Arjovsky et al., 2017), whereby the distance between distributions P_r and P_g is based on the Wasserstein-1 distance $W(P_r, P_g)$:

$$\min_G \max_{D \in \mathcal{D}} \mathbb{E}_{\mathbf{X} \sim P_r}[D(\mathbf{X})] - \mathbb{E}_{\mathbf{Z} \sim p_g}[D(G(\mathbf{Z}))]. \tag{10}$$

Given that the output of $D(\cdot)$ in equation (10) is a score rather than a classification to 0 and 1, it is referred to as a "critic". Once gradients of the loss function are calculated with respect to the network parameters, the error is back-propagated through the network, allowing updates of the weights and biases of each layer.

2.3 SGAN architecture and training

The network architecture of the generator and critic are asymmetric with respect to each other (see Appendix A.1 for details) and each of them contains five sequentially stacked convolutional layers. Spectral normalization is applied to the weights in each critic layer (Miyato et al., 2018). This normalizes the weight matrices $\mathbf{K}$ with respect to the spectral norm at each layer, thus forcing them to conform to the Lipschitz continuity condition. In the SGAN trained on model errors, we apply mean spectral normalization to critic layers (Subramanian and Chong, 2019) as it improved the quality of the generated model-error realizations. The generator feature maps are normalized with respect to features (elements) using instance normalization (Ulyanov et al. (2016)). The first four layers of the critic and the generator are followed by a rectified linear unit (ReLU): $f(x) = max(0, x)$ and a LeakyReLU: $f(x) = max(0.2x, x)$ activation function, respectively, and the last layer in the generator is followed by a tanh activation function. We set the learning

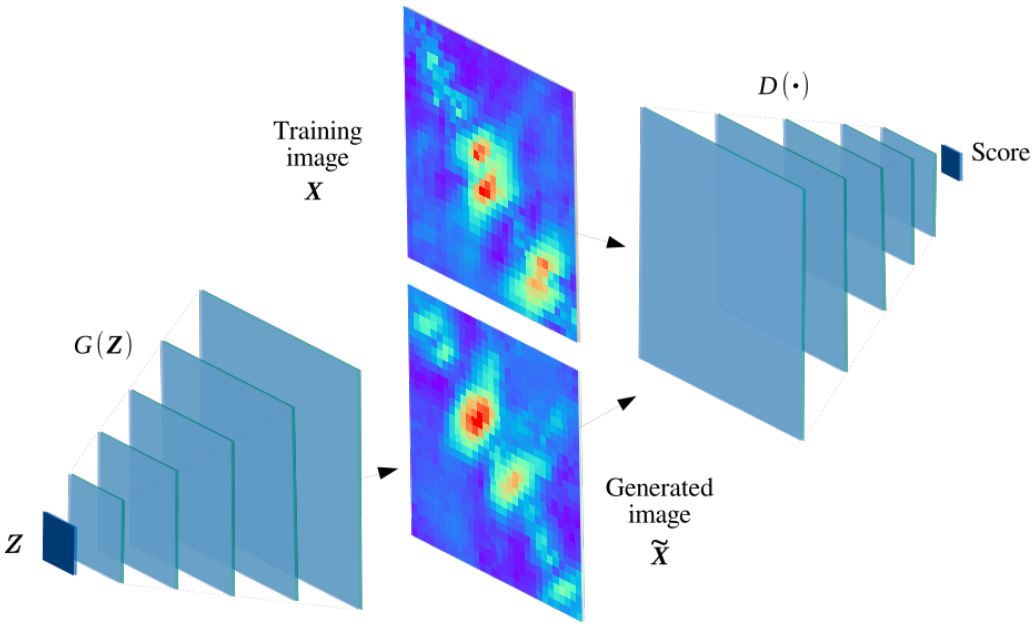

Figure 1: Illustration of our SGAN architecture with five layers when applied to represent model errors. During training, each parameter in the latent space $\mathbf{Z}$ is randomly drawn from a uniform distribution $\mathcal{U}(-1,1)$. Each $\mathbf{Z}$ is transformed into a single image $\tilde{X}$ through a nonlinear transformation $G(\cdot)$. At each iteration, a batch of images $\tilde{X}$ (generated) and a batch of images X (training) are interchangeably fed into the critic $D(\cdot)$, resulting in a score that is then used to update the network parameters through back-propagation.

rates of the generator and critic according to the two time-scale update rule (TTUR) by Heusel et al. (2017), with a ratio of $1:4$. The output size x of layers $l = 1, .., 5$ in the generator can be calculated via the following relationship:

$$x_l = s \cdot (x_{l-1} - 1) - 2 \cdot p + (k-1) + 1, \tag{11}$$

where s is the stride controlling movement of the filter along the image, p is the the number of padding columns/rows of zeros added to the layer's input, and k is the kernel size (see Dumoulin and Visin (2016) for more information). We use padding to control the output size and obtain an image with dimensions that are as close as possible to our model size (see Appendix A for more details).

All images fed into the critic must be normalized to a $[-1,1]$ range and have the same dimensions. Thus, TI's are either cropped (subsurface model parameter) or linearly interpolated (model error) into a size fitting that of the generative network's output. After training, the generated subsurface model parameter $\tilde{\mathbf{X}}_{\boldsymbol{\Phi}}$ or model-error $\tilde{\mathbf{X}}_{\boldsymbol{\eta}}$ realizations are either cropped or interpolated to the desired image size and re-scaled back to the original value range. In the case of the subsurface model-parameter realizations there is an additional step where porosity values are assigned according to equation 1.

2.4 Bayesian inference of latent parameters

We aim to estimate the low-dimensional (latent-space) representation of the subsurface model parameters and associated model error by incorporating the two trained generative networks within an MCMC inversion. Subsurface model-parameter and model-error prior realizations are generated using the SGAN and the forward responses during inversion are computed using the straight-ray solver $g^{LF} = g^{SR}$. The posterior probability density function (pdf) $p(\mathbf{Z}|\mathbf{d})$ is expressed through Bayes' theorem as:

$$p(\mathbf{Z}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{Z})p(\mathbf{Z})}{p(\mathbf{d})}, \tag{12}$$

where $p(\mathbf{d}|\mathbf{Z})$ is the likelihood function, $p(\mathbf{Z})$ is the prior pdf of latent parameters $\mathbf{Z}$, and $p(\mathbf{d})$ is the marginal likelihood (evidence). The latter is a constant that we ignore in this work and we thus focus on the unnormalized posterior

$p(\mathbf{Z}|\mathbf{d}) \propto p(\mathbf{d}|\mathbf{Z})p(\mathbf{Z})$. For numerical reasons we work with the log-likelihood which, assuming the measurement errors are independent, identical and normally distributed, is given by:

$$l(\mathbf{d}|\mathbf{Z}) = -\frac{N_d}{2}\log(2\pi) - N_d\log(\sigma) - \frac{1}{2}\sigma^{-2}\left[\mathbf{d} - \mathbf{d}_{sim}\right]^2, \tag{13}$$

where N_d is the number of data points, σ is the standard deviation of the measurement errors $\boldsymbol{\epsilon}$, and $\mathbf{d}_{sim}$ and $\mathbf{d}$ are the forward simulated and observed data, respectively. To sample from the posterior distribution, we rely on the differential evolution adaptive Metropolis (DREAM$_{(ZS)}$) algorithm, in which MCMC chains evolve in parallel and jumps are proposed based on candidate points from an archive of past states (ter Braak and Vrugt, 2008; Vrugt et al., 2009; Laloy and Vrugt, 2012). In this algorithm, the jump size is given by $\gamma = \frac{2.38}{\sqrt{2\delta d'}}\beta$, where β is a user defined scalar referred to here as the jump rate scaling factor, δ is the number of candidate points pairs used to generate the proposal, and $d^{'}$, the number of dimensions to be updated, varies during the inversion according to a crossover (CR) probability (Laloy and Vrugt, 2012). At each MCMC step and for each individual chain, a random sample is drawn from the proposal distribution $q(\mathbf{Z}', \mathbf{Z}^{t-1})$, which is symmetric with boundary handling to ensure that the samples are drawn proportionally to the uniform prior. As the prior is uniform, the sample is either accepted or rejected according to a transition acceptance rule $p_{acc}(\mathbf{Z}^{t-1} \rightarrow \mathbf{Z}') = e^{(l(\mathbf{d}|\mathbf{Z}') - l(\mathbf{d}|\mathbf{Z}^{t-1}))}$. If accepted, the chain moves to $\mathbf{Z}'$ such that $\mathbf{Z}^t = \mathbf{Z}'$. If rejected, the chain remains at the current sample and $\mathbf{Z}^t = \mathbf{Z}^{t-1}$. We run the inversion with eight parallel chains and, to improve the search, we allow for a 20% chance of snooker update (ter Braak and Vrugt, 2008) during the first $20,000$ steps (per chain) which we consider as the burn-in period. As opposed to parallel updating where sampling occurs along an axis that runs past states of a single chain, the snooker update involves an axis that runs along states of two different chains. The jump rate scaling factor β is varied adaptively during the burn-in period in order to reach a $20\% - 30\%$ MCMC acceptance rate. To prevent very high acceptance rates and slow mixing after the burn-in period, we set a minimum value to the β, beyond which it cannot decrease.

We jointly infer the posterior distribution of the two low-dimensional latent spaces: $\mathbf{Z}_{\boldsymbol{\Phi}}$ describing the subsurface model parameters and $\mathbf{Z}_{\boldsymbol{\eta}}$ describing the model error, both of which have uniform prior distributions $\mathcal{U}(-1, 1)$. The proposed latent parameter realizations are mapped into their respective high-dimensional image spaces $\boldsymbol{\Phi}$ and $\boldsymbol{\eta}_{app}$ (approximate model error), where a low-fidelity forward response is calculated on the porosity field $\boldsymbol{\Phi}$ converted to slowness $\mathbf{s}$ using equations (2) and (3). In addition to the subsurface model-parameter and model-error latent parameters, we infer an auxiliary parameter ν with a uniform prior distribution $\mathcal{U}(0, 1)$ that scales the model-error realization before it is added to the simulated data. This scalar was found to improve the inference and quality of the inferred model errors by providing additional means to control their magnitudes. When inferring model errors, $\mathbf{d}_{sim}$ in equation (13) is replaced with $g^{SR}(\mathbf{s}) + \nu\boldsymbol{\eta}_{app}$. The most salient features of our method, combining SGAN-ME (ME stands for model error) with MCMC inversion, is provided in Algorithm 1 and Figure 2.

We compare SGAN-ME against cases where model errors are zero as the high-fidelity forward solver is used in MCMC inversions or model errors are either ignored or approximated to be Gaussian. In these latter cases, the inferred parameters are the latent parameters of the model alone, such that $\mathbf{Z} = \mathbf{Z}_{\boldsymbol{\Phi}}$ and we simply plug $\mathbf{d}_{sim} = g^{SR}(\mathbf{s})$ into equation (13).

To approximate the model errors as Gaussian, we follow Hansen et al. (2014) and learn their mean $\mathbf{d}_{ME}$ and a covariance matrix $\mathbf{C}_{ME}$, which are used to correct the residual term and inflate the likelihood function:

$$l(\mathbf{d}|\mathbf{Z}) = -\frac{N_d}{2}\log(2\pi) - \frac{1}{2}\log(|\mathbf{C}_D|) - \frac{1}{2}\left[\mathbf{d} - g^{SR}(\mathbf{s}) - \mathbf{d}_{ME}\right]^T \mathbf{C}_D^{-1}\left[\mathbf{d} - g^{SR}(\mathbf{s}) - \mathbf{d}_{ME}\right], \tag{14}$$

where $\mathbf{C}_D = \mathbf{C}_d + \mathbf{C}_{ME}$, with $\mathbf{C}_d$ being the traditional data covariance matrix and $\mathbf{C}_{ME}$ the learned model-error covariance matrix. The bias correction term $\mathbf{d}_{ME}$ is the model-error mean. We use 800 random model-error samples from the same database used for training the SGAN to learn $\mathbf{C}_{ME}$ and $\mathbf{d}_{ME}$, noting that Hansen et al. (2014) recommend to use at least 300 samples.

Algorithm 1: SGAN-ME inversion with differential evolution adaptive Metropolis DREAM$_{(ZS)}$

Set t = 1 and initialize the archive with realizations $\mathbf{Z}_{\boldsymbol{\Phi}}$, $\mathbf{Z}_{\boldsymbol{\eta}}$ and ν randomly drawn from $p(\mathbf{Z}_{\boldsymbol{\Phi}})$, $p(\mathbf{Z}_{\boldsymbol{\eta}})$ and $p(\nu)$ (respectively)
Initialize $\mathbf{Z}^t = [\mathbf{Z}^t_{\boldsymbol{\Phi}}, \mathbf{Z}^t_{\boldsymbol{\eta}}, \nu^t]$ for each MCMC chain
$\tilde{\mathbf{X}}^t_{\boldsymbol{\Phi}}, \tilde{\mathbf{X}}^t_{\boldsymbol{\eta}_{app}} \leftarrow \mathrm{G}_{\boldsymbol{\Phi}}(\mathbf{Z}^t_{\boldsymbol{\Phi}}), \mathrm{G}_{\boldsymbol{\eta}}(\mathbf{Z}^t_{\boldsymbol{\eta}})$
Perform post-processing (section 2.3): $\boldsymbol{\Phi}^t, \boldsymbol{\eta}^t_{app} \leftarrow \tilde{\mathbf{X}}^t_{\boldsymbol{\Phi}}, \tilde{\mathbf{X}}^t_{\boldsymbol{\eta}_{app}}$ and convert $\boldsymbol{\Phi}^t$ into slowness $\mathbf{s}^t$ (equations (2)-(3))
$\mathbf{d}_{sim} = g^{LF}(\mathbf{s}^t) + \nu^t \boldsymbol{\eta}^t_{app}$
Compute $l(\mathbf{d}|\mathbf{Z}^t)$ (equation (13))
while $t < N_{draw}$ **do**
  Propose a new sample $\mathbf{Z}'_{\boldsymbol{\Phi}}$, $\mathbf{Z}'_{\boldsymbol{\eta}}$ and ν' from proposal distribution $q(\mathbf{Z}', \mathbf{Z}^{t-1})$
  $\tilde{\mathbf{X}}^{'}_{\boldsymbol{\Phi}}, \tilde{\mathbf{X}}^{'}_{\boldsymbol{\eta}_{app}} \leftarrow \mathrm{G}_{\boldsymbol{\Phi}}(\mathbf{Z}'_{\boldsymbol{\Phi}}), \mathrm{G}_{\boldsymbol{\eta}}(\mathbf{Z}'_{\boldsymbol{\eta}})$
  Perform post-processing (section 2.3): $\boldsymbol{\Phi}^{'}, \boldsymbol{\eta}^{'}_{app} \leftarrow \tilde{\mathbf{X}}^{'}_{\boldsymbol{\Phi}}, \tilde{\mathbf{X}}^{'}_{\boldsymbol{\eta}_{app}}$ and convert $\boldsymbol{\Phi}^{'}$ into slowness $\mathbf{s}^{'}$ (equations (2)-(3))
  $\mathbf{d}_{sim} = g^{LF}(\mathbf{s}') + \nu' \boldsymbol{\eta}'_{app}$
  Compute $l(\mathbf{d}|\mathbf{Z}')$
  Compute probability of acceptance $\alpha \leftarrow e^{(l(\mathbf{d}|\mathbf{Z}') - l(\mathbf{d}|\mathbf{Z}^{t-1}))}$
  Draw U from a uniform distribution $\mathcal{U}(0,1)$
  if $U < \alpha$ **then**
    $\mathbf{Z}^t \leftarrow \mathbf{Z}'$
  else
    $\mathbf{Z}^t \leftarrow \mathbf{Z}^{t-1}$
  end
  $t = t + 1$
end
Function $\mathrm{G}_{\boldsymbol{\Phi}}(\mathbf{Z}_{\boldsymbol{\Phi}})$
  Performs a series of transposed convolution layers with pre-trained weights
  return $\tilde{\mathbf{X}}_{\boldsymbol{\Phi}}$
end
Function $\mathrm{G}_{\boldsymbol{\eta}}(\mathbf{Z}_{\boldsymbol{\eta}})$
  Performs a series of transposed convolution layers with pre-trained weights
  return $\tilde{\mathbf{X}}_{\boldsymbol{\eta}_{app}}$
end

3 Results

In our numerical experiments we consider two parallel vertically-oriented boreholes, one containing 30 sources and the other 30 receivers. The model domain on which the numerical experiment is performed is 4×6.1 m (40×61 pixels). Sources and receivers are distributed evenly between 0.2 and 6 m depth in intervals of 0.2 m and the two boreholes are located at 0 m and 4 m along the horizontal axis, respectively. In the straight-ray and eikonal forward solvers, the model domain is discretized evenly into 0.1 m square cells. The FDTD simulated responses are performed using a spatial discretization of 0.025 m and a time discretization of 0.15 ns. The FDTD simulation requires the dielectric constant of the medium $\boldsymbol{\kappa}_b$ and electrical conductivity fields as input. We assume a constant conductivity of 0.002 S/m across the model domain. The dielectric constant $\boldsymbol{\kappa}_b$ is obtained using equation (2). The model-error databases corresponding to $\boldsymbol{\eta}^{eikonal-SR}$ and $\boldsymbol{\eta}^{FDTD-SR}$ contain $10,000$ images, each of which requires a simulation using the low- and high-fidelity forward solvers. In the next subsections, we present results obtained from SGAN training and subsequent inversions.

3.1 Quality assessment of generative models

By training the SGAN on the subsurface model parameters and model error, we are able to reduce the two parameter spaces containing 2440 and 900 parameters (respectively) into two latent spaces, $\mathbf{Z}_{\boldsymbol{\Phi}}$ and $\mathbf{Z}_{\boldsymbol{\eta}}$, each of size $5 \times 5 \times 1$. In order to assess the quality of the generative models at a given training iteration, we calculate pixel-wise means and variances on a set of generated and training realizations. Based on this analysis, we found that the quality of the generated realizations could be further improved by scaling each realization by a spatially-varying correction factor intended to match the pixel-wise means of the TI's:

$$\tilde{\mathbf{X}} = G(\mathbf{Z}) \cdot (\mathbf{M}_x \oslash \mathbf{M}_{\tilde{x}}), \tag{15}$$

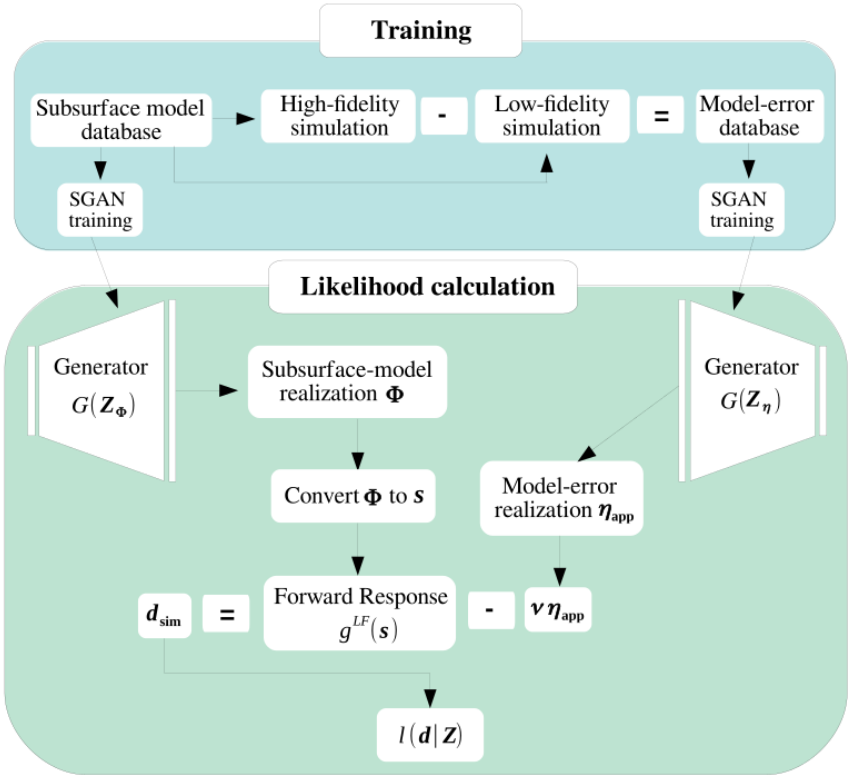

Figure 2: SGAN-ME workflow. Subsurface-model representation using SGAN is discussed in details in the work of Laloy et al. (2018), here we focus on model-error representation.

where $\mathbf{M}_x$ is the mean of $10,000$ TI's, $\mathbf{M}_{\tilde{x}}$ is the mean of $10,000$ SGAN realizations and $G(\mathbf{Z})$ is a single SGAN realization to be corrected. The correction matrix obtained by element-wise division $\mathbf{M}_x \oslash \mathbf{M}_{\tilde{x}}$ contains the mean of generated SGAN realizations, and, thus, it is specific to a given training iteration. For the subsurface model-parameter realizations, we also evaluate the spatial auto-correlation within each realization by calculating directional semivariograms using the *GSTools* package (Müller and Schüler, 2020).

Training the SGAN for $58,000$ iterations with a batch containing 64 images took about $8-9$ hours on one GPU GeForce GTX Titan X with 12 GB memory. Figure 3 provides a comparison between the statistics of the subsurface model-parameter training images and SGAN realizations. We show the pixel-wise mean and variance of the TI's (Figs 3a-b) and the SGAN realizations before (Figs 3c-d) and after (Figs 3e-f) applying the correction in equation 15. The SGAN mean image before correction shows horizontal band-like features. After mean correction, this effect decreases and the image becomes closer to homogeneous. The variance images, however, do not exhibit the same improvement following the correction and look overall similar in both cases (Figs 3d and f). The spatial statistics represented by the directional semivariograms in x- and y-directions are given in Figures 3g and h, respectively; the mean semivariograms are calculated over $5,000$ TI (blue) and corrected SGAN (red) model realizations. The two mean curves fall on top of each other, indicating a good agreement between the TI and corrected SGAN realizations. Furthermore, the semivariograms of single SGAN realizations (gray curves) are mostly concentrated within the ranges of the TI (dashed blue curves).

A similar mean correction and assessment to those described above for the subsurface model-parameter SGAN training are performed for the model error training. Example model-error TI's for the two types of model errors considered in this paper are shown in Figure 4. In most cases, $\boldsymbol{\eta}^{FDTD-SR}$ has a larger range of error values compared to $\boldsymbol{\eta}^{eikonal-SR}$ and displays similar features to $\boldsymbol{\eta}^{eikonal-SR}$ with additional off-diagonal patterns. Figure 5 provides a comparison between the pixel-wise mean and variance of the model-error TI's and those of the SGAN realizations before and after the mean correction. Although the mean image of the SGAN generated $\boldsymbol{\eta}_{app}^{eikonal-SR}$ realizations before correction (Fig. 5c) is close to that of the TI realizations (Fig. 5a), it underestimates the model-error mean on the diagonal. After correction

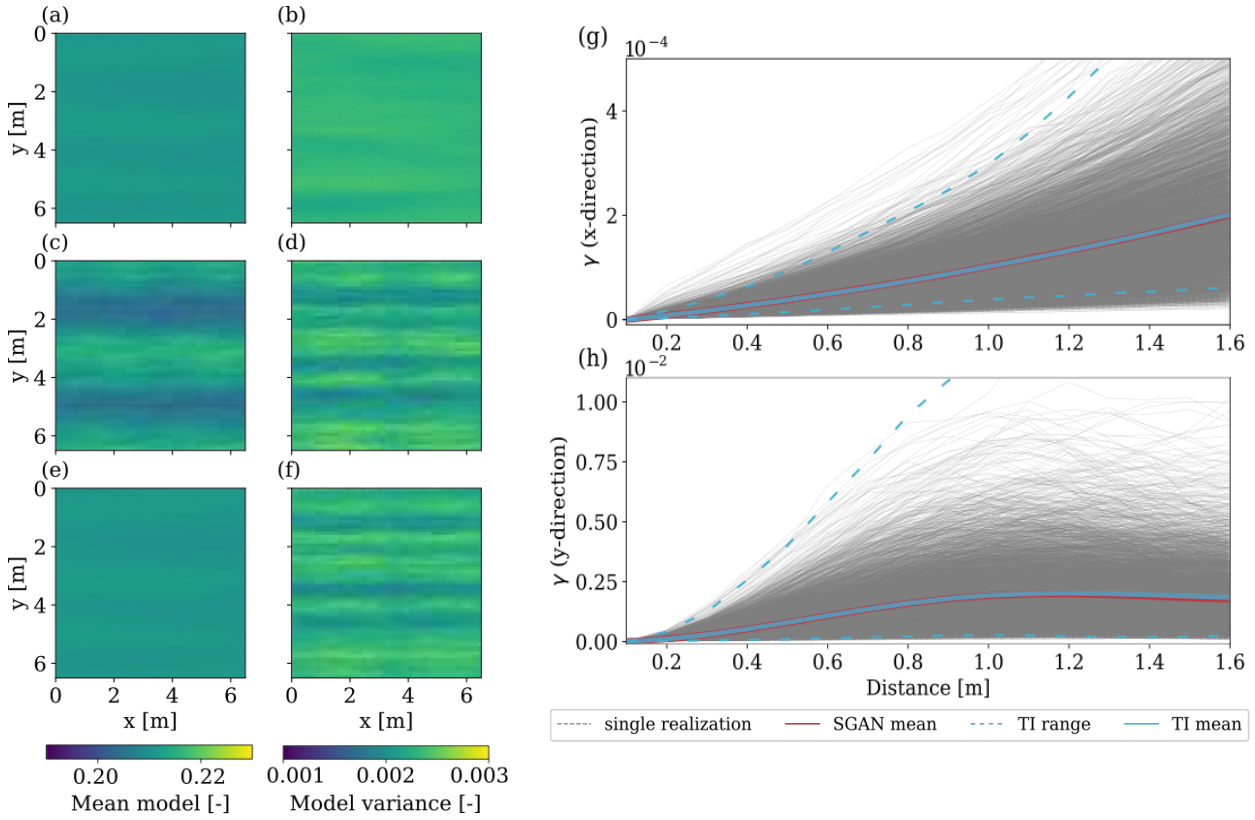

Figure 3: Statistics of subsurface model-parameter realizations after $58,000$ training iterations. Mean and variance images calculated on $5,000$: (a-b) TI realizations , (c-d) SGAN realizations before mean correction and (e-f) SGAN realizations after mean correction. The directional semivariograms in the (g) x- and (h) y-directions were calculated on $5,000$ TI realizations and SGAN realizations after mean correction. The gray lines are single semivariograms calculated on SGAN realizations after correction; their mean is marked as a solid red line and it is almost completely overlapped by the blue solid line, representing the mean of TI realizations. The blue dashed lines mark the TI realizations' range.

(Fig. 5e), the bias in the mean is removed and the variance (Fig. 5f), which also suffers from underestimation on the diagonal, is slightly improved. Training with $\boldsymbol{\eta}^{FDTD-SR}$ realizations proved to be more challenging and required larger number of training iterations ($450,000$ iterations as opposed to $250,000$ for $\boldsymbol{\eta}^{eikonal-SR}$). The SGAN mean image before correction (Fig. 5i) is distorted compared to the TI mean image (Fig. 5g). We attribute this difference to the patchy nature of the $\boldsymbol{\eta}^{FDTD-SR}$ realizations and features that extend to elements further off-diagonal (Fig. 4). These distortions were reduced after applying the mean correction (Fig. 5k), although improvements in the variance (Fig. 5l) are not as visible. One can observe a broken pattern on the diagonal in the $\boldsymbol{\eta}^{FDTD-SR}$ TI's mean and variance images (Figs. 5g and h). This pattern can also be found in $\boldsymbol{\eta}^{eikonal-SR}$ TI's mean and variance images (Figs. 5a and b), albeit to a lesser extent. Since the subsurface model-parameter TIs on which model errors are calculated were randomly chosen, we attribute this pattern to be a result of the forward modeling process rather than a repetitive pattern in the subsurface model-parameter TIs.

Finally, we test the ability of the SGAN to capture the true model by performing a pixel-to-pixel MCMC inversion (i.e., the actual pixel values are considered as data in the inversion) on two reference models, cropped out of the testing segment of the subsurface model-parameter TI described in Section 2.1.1. We consider the maximum a posteriori estimate of pixel-to-pixel based inversion results as being the closest possible SGAN representation of the reference model ('closest SGAN realization'). Figure 6 shows the considered reference models and their corresponding closest SGAN realization illustrating the capability of the SGAN to generate model realizations that closely resemble their reference models.

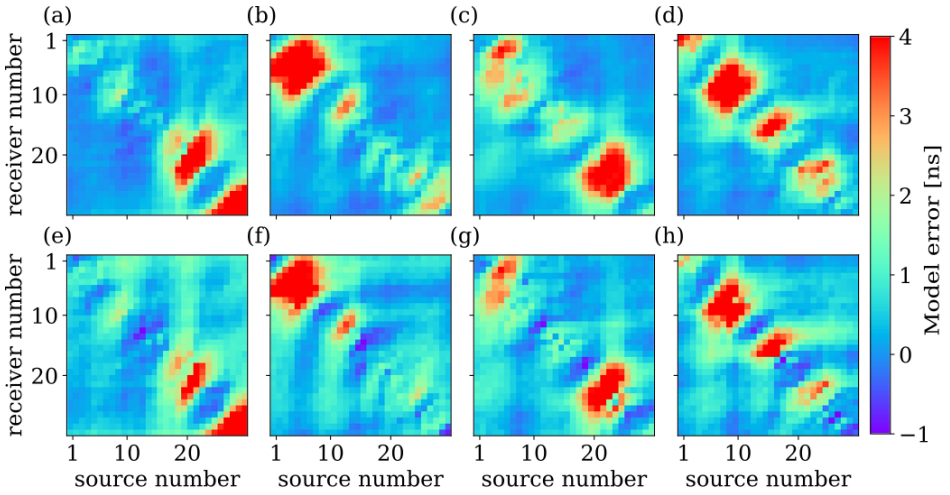

Figure 4: Examples of actual model-error realizations (a-d) $\boldsymbol{\eta}^{eikonal-SR}$ and (e-h) $\boldsymbol{\eta}^{FDTD-SR}$. Figures in the same column were calculated for the same subsurface model-parameter realization.

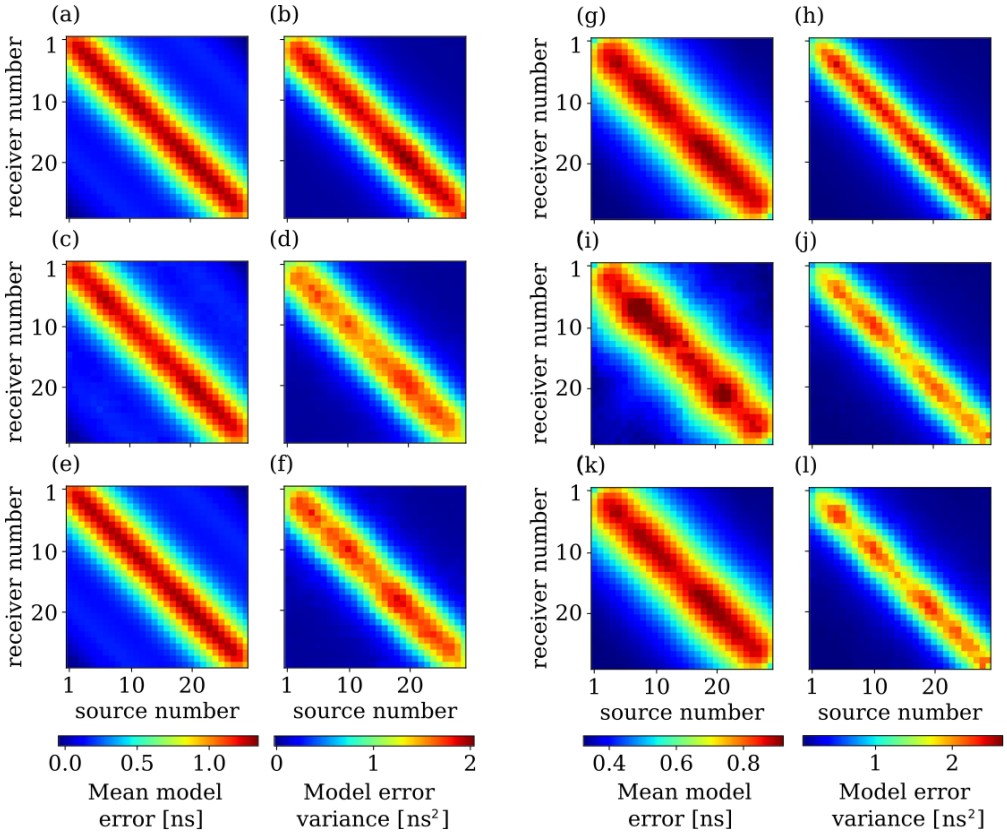

Figure 5: Model errors of (a-f) $\boldsymbol{\eta}^{eikonal-SR}$ and (g-l) $\boldsymbol{\eta}^{FDTD-SR}$. Pixel-wise mean and variance of $10,000$ (a-b and g-h) TI realizations, (c-d and i-j) SGAN realizations before mean correction and (e-f and k-l) SGAN realizations after mean correction (see equation (15)).

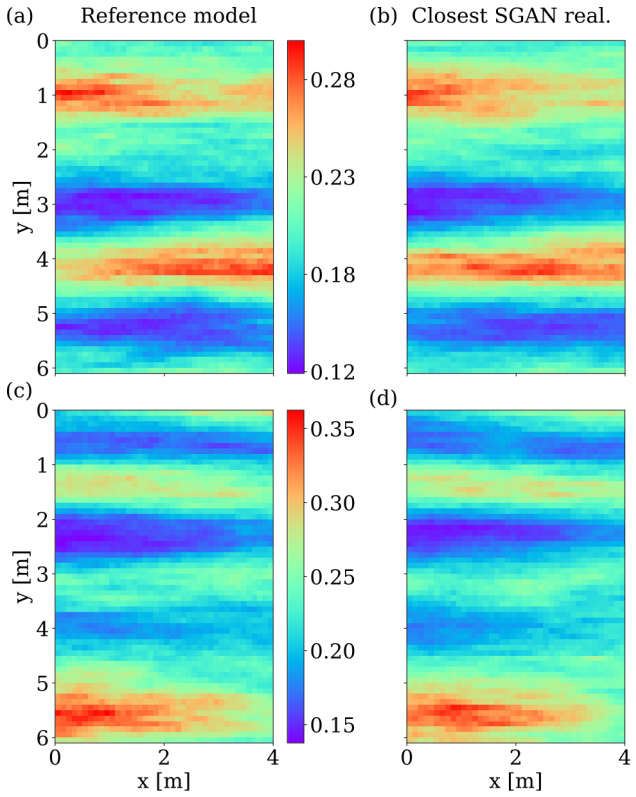

Figure 6: Reference models (a) 1 and (c) 2 and (b and d) corresponding closest SGAN realizations obtained from pixel-to-pixel inversion considering 25 latent parameters.

3.2 Inversion results

We perform inversion of data generated from the two multi-Gaussian reference models in Figures 6a and 6c that we refer to as 'Model 1' and 'Model 2', respectively. The synthetic data for each reference model are created using the high-fidelity forward solver, which is either $g^{eikonal}$ or g^{FDTD} depending on the type of model error considered (i.e., $\boldsymbol{\eta}^{eikonal-SR}$ or $\boldsymbol{\eta}^{FDTD-SR}$). The data are contaminated with random noise drawn from a normal distribution $\mathcal{N}(0, 0.5^2\ ns^2)$. We consider in our analysis only those traveltime data corresponding to source-receiver angles of less than $50°$ from the horizontal, as is commonly done with field data to avoid borehole and antenna effects (Irving and Knight, 2005). This leads to a total of 858 traveltimes to be considered in the inversion. Note that the number (25) of subsurface model parameters $\mathbf{Z}_{\boldsymbol{\Phi}}$ to be estimated is the same for all considered approaches. The SGAN-ME approach requires estimation of 26 additional parameters: 25 for the model error $\mathbf{Z}_{\boldsymbol{\eta}}$ along with auxiliary parameter ν. Note that the number of parameters in $\mathbf{Z}_{\boldsymbol{\Phi}}$ and $\mathbf{Z}_{\boldsymbol{\eta}}$ is chosen based on a trade-off between inversion performance and efficiency. It is chosen such that it remains low while enduring high-quality subsurface model estimation.

For each of the considered approaches, we show the maximum a posteriori estimate. Given the uniform prior on the parameters, this corresponds also to the maximum-likelihood solution. For comparison, we calculate the root mean-square-error (RMSE) and structural similarity (SSIM) index for each approach including that of the closest SGAN realization obtained by pixel-by-pixel inversion. We consider two different RMSE values: one on the subsurface model parameters denoted by $\text{RMSE}_{\boldsymbol{\Phi}}$ and the other on the data denoted $\text{RMSE}_{\mathbf{d}}$. The RMSE metric gives an indication as to the spread of residuals, with larger weight given to higher values, while the SSIM complements the latter by measuring the similarity of two images (here these are images of either the subsurface model parameters or model errors) in terms of their structure (see Appendix B). The above metrics are calculated for the maximum-likelihood realization in the case of pixel-to-pixel inversion whereas in data-based inversions, they represent an average value for the last 50% samples of the chains. In the case of the inferred model error, we also calculate what we refer to as "error recovery". This measure serves as an indication of how well the model error is approximated, by taking the average posterior mean-squared-error (MSE) between the approximated model error and the reference model error ($MSE(\boldsymbol{\eta}_{app}, \boldsymbol{\eta}_{ref})$) and dividing it by the MSE between the reference model and 0 ($MSE(\boldsymbol{\eta}_{ref}, 0)$).

Table 1: Inversion convergence for Test Case 1 ($\boldsymbol{\eta}^{eikonal-SR}$) and Test Case 2 ($\boldsymbol{\eta}^{FDTD-SR}$). The mean acceptance rate represents the average acceptance rate of the two tested reference models excluding the first $20,000$ steps.

Model error type	Inversion approach	Nr. of MCMC steps (per chain)		Mean acceptance rate (excl. burn-in) [%]
		Model 1	Model 2	
$\boldsymbol{\eta}^{eikonal-SR}$	straight-ray	$95,510$	$22,060$	28
	Covariance	$43,860$	$189,760$	36
	SGAN-ME	$108,960$	$382,620$	18
	eikonal	$34,710$	$61,160$	23
$\boldsymbol{\eta}^{FDTD-SR}$	straight-ray	$53,160$	$141,110$	18
	Covariance	$27,810$	$188,010$	35
	SGAN-ME	$363,760$	$437,810$	23

3.2.1 Convergence

We use the Gelman-Rubin diagnostic (Gelman and Rubin, 1992) and declare convergence when all inferred parameters satisfy $\hat{R} \leq 1.2$. The initial jump rate scaling factor was set to 5 for all inversion runs. The minimum jump rate scaling factor had to be adjusted in each inversion individually in order to achieve a reasonable acceptance rate (ideally $20-30\%$ and not more than 50%) and convergence. A value of 0.2 was often suitable to achieve convergence and reasonable acceptance rates with some SGAN-ME cases requiring slightly smaller values ($0.15 - 0.2$). In Table 1, we provide convergence information for each inversion approach. All inversions reached convergence, but the number of steps required differ between approaches. More steps are needed to reach convergence with the SGAN-ME approach. The mean acceptance rates in Table 1 are consistently higher for the covariance approach compared to the other approaches due to its inflated error term, which increases the chance for proposed samples to be accepted in the MCMC.

3.2.2 Test Case 1: eikonal - straight-ray model error

We first consider inversion results with model error $\boldsymbol{\eta}^{eikonal-SR}$ in terms of maximum-likelihood solutions of the straight-ray, covariance, SGAN-ME and eikonal-based inversion approaches in Figure 7 and $\text{RMSE}_{\boldsymbol{\Phi}}$, SSIM and $\text{RMSE}_{\mathbf{d}}$ in Table 2. Generally speaking and given values in Table 2, the SGAN-ME approach exhibits better overall performance compared to the straight-ray and covariance approaches, scoring lower RMSE and higher SSIM values. The SGAN-ME approach captures well the general structure of the various porosity zones in both test models. The spatial representation of model errors in Figure 8 together with values in Table 3, suggest that SGAN-ME is able to recover a large part of the model error (about 51% for Model 1 and 67% for Model 2). Table 3 also indicates that the closest SGAN realizations obtained by the pixel-based inversions consistently reached better scores than the closest of the $10,000$ model realizations used for training, thereby indicating that the SGAN generalizes well for the model error.

We now consider results for Model 1 specifically. The SGAN-ME and eikonal solutions exhibit similar structures between 0 and 5 m depth, resulting in similar SSIM values (0.79 and 0.78, respectively). The low porosity zone between 5 and 5.5 m depth is thinner in the SGAN-ME solution and the high-porosity zone between $4 - 4.5$ is overestimated. This can be explained by considering the SGAN-ME model-error posterior samples in Figures 8c-e. Although the features on the diagonal (and close to diagonal) are correctly located, they are underestimated for source-receiver pairs $(15, 15) - (20, 20)$ and overestimated for source-receiver pairs $(20, 20) - (25, 25)$ causing overestimation of porosity in the region corresponding to the latter source-receiver pairs. Furthermore, the model errors at the bottom right corner of all posterior samples in Figure 8c-e are overestimated and differ by up to ~ 2 ns from the truth, translating to a thicker high-porosity layer at the bottom of the subsurface model (5.5-6 m). The covariance solution overestimates the low-porosity zone at around 3 m depth. It scores the same $\text{RMSE}_{\boldsymbol{\Phi}}$ as the straight-ray solution (0.016) but receives higher SSIM (0.74 versus 0.72) and slightly lower $\text{RMSE}_{\mathbf{d}}$ (0.75 ns versus 0.77 ns) scores.

As for Model 2, the SGAN-ME maximum-likelihood realization is the only solution properly reconstructing the porosity structure between 0-1 m depths. Other approaches, including the eikonal solution do not have a clear layered structure around these depths. The eikonal solution tend to overestimate some high-porosity zones (4-4.5 m in Model 1 and around 1.5-1.7 m in Model 2) and exhibit rough texture in its solution to Model 2. The covariance solution underestimates the porosity at 4 m depth but still surpasses the straight-ray solution in both subsurface model-parameters scores ($\text{RMSE}_{\boldsymbol{\Phi}}$ and SSIM). As opposed to Model 1, here the straight-ray solution fits the data significantly better than the covariance solution ($\text{RMSE}_{\mathbf{d}}$ of 0.87 ns for straight-ray versus 1.17 ns for covariance). Furthermore, the straight-ray

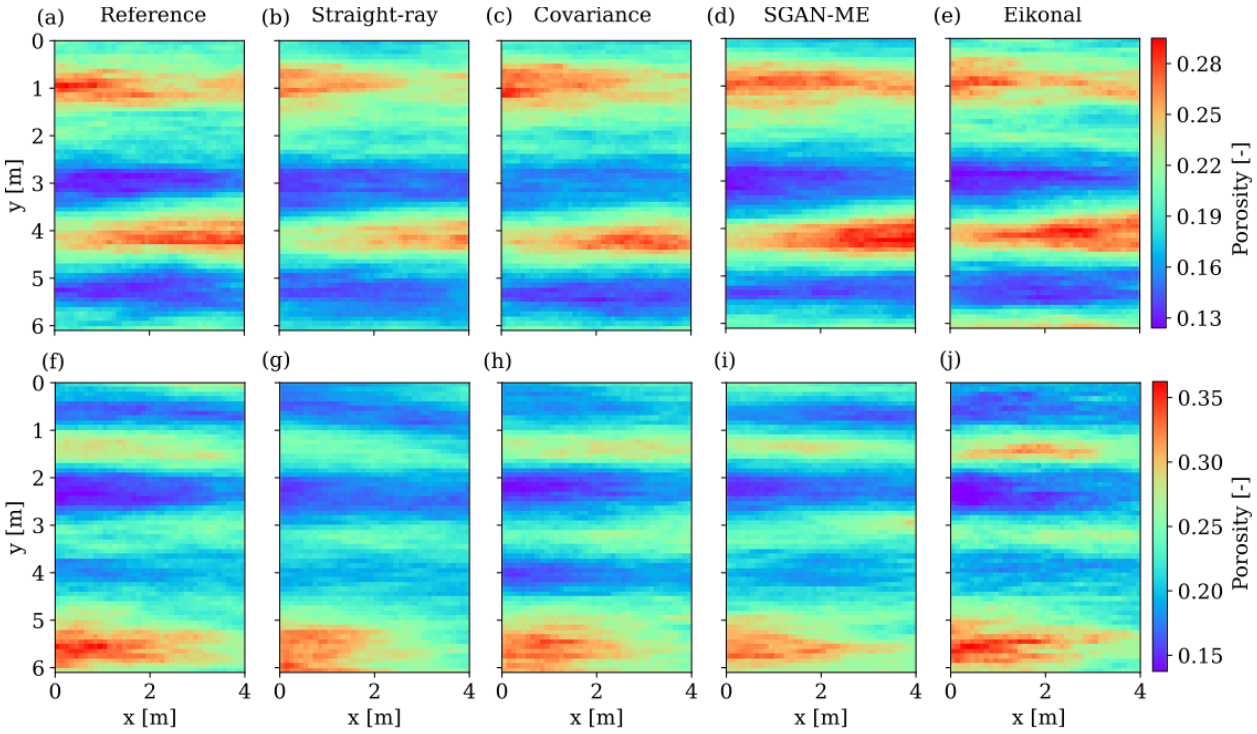

Figure 7: Inversion results for reference Models (a) 1 and (f) 2 for Test Case 1 ($\boldsymbol{\eta}^{eikonal-SR}$). (b)-(e) and (g)-(j) are the maximum-likelihood realizations obtained from inversion using the straight-ray, covariance, SGAN-ME and eikonal approaches. The first three approaches use the straight-ray solver for the forward response during inversion, while the observed data for all approaches were created using the eikonal solver.

solutions are smooth and do not contain major artifacts. They do however, generally underestimate high-porosity zones and receive the highest RMSE and lowest SSIM scores in most cases.

As can be seen in Table 2, the $\text{RMSE}_{\mathbf{d}}$ was also calculated for the closest SGAN realization using the high-fidelity forward solver, namely the eikonal solver. For better visualization, we show in Figure 9 the $\text{RMSE}_{\mathbf{d}}$ values of each approach and for each of its eight chains along $100,000$ sequential samples (per chain). The data fit plots corresponding to Model 1 and 2 and model error $\boldsymbol{\eta}^{eikonal-SR}$ (Figs 9a and b) indicate that our SGAN-ME approach fits the data as well as the eikonal solver, close to the noise level of 0.5 ns (indicated by the red dotted line) and significantly better than the straight-ray and covariance approaches. The $\text{RMSE}_{\mathbf{d}}$ of the closest SGAN realization (indicated by a dotted black line) is higher compared to that of the eikonal and SGAN-ME inversion approaches, but lower than that of the straight-ray and covariance approaches.

Finally, we represent posterior samples in the form of $\text{RMSE}_{\boldsymbol{\Phi}}$ and SSIM distributions (Figs. 10a,b,e,f). The $\text{RMSE}_{\boldsymbol{\Phi}}$ and SSIM values, calculated separately for each posterior sample, were plotted as a normalized density function to which a Gaussian kernel was fitted. It is observed that the SGAN-ME approach generally results in $\text{RMSE}_{\boldsymbol{\Phi}}$ and SSIM distributions that rank higher than the straight-ray and covariance approaches. For Model 2, the $\text{RMSE}_{\boldsymbol{\Phi}}$ and SSIM distributions associated with SGAN-ME almost completely overlap those corresponding to the model error-free eikonal approach. The SGAN-ME posterior distributions are characterized by intermediate widths as opposed to the covariance approach for which $\text{RMSE}_{\boldsymbol{\Phi}}$ and SSIM values vary widely and to the straight-ray approach for which the distribution is narrow and with the worst statistics.

3.2.3 Test Case 2: FDTD - straight-ray model error

We now consider the model error $\boldsymbol{\eta}^{FDTD-SR}$ for the same reference porosity models and create the synthetic data using the FDTD forward solver. Here we compare only between the straight-ray, covariance and SGAN-ME approaches due to the excessive computational time needed to perform MCMC inversion with the FDTD forward solver (Hunziker et al., 2019). Results for this test case can be found in Figure 11 and Table 4 which show the maximum-likelihood solution of the straight-ray, covariance and SGAN-ME approaches for the two reference models and their respective RMSE and SSIM scores.

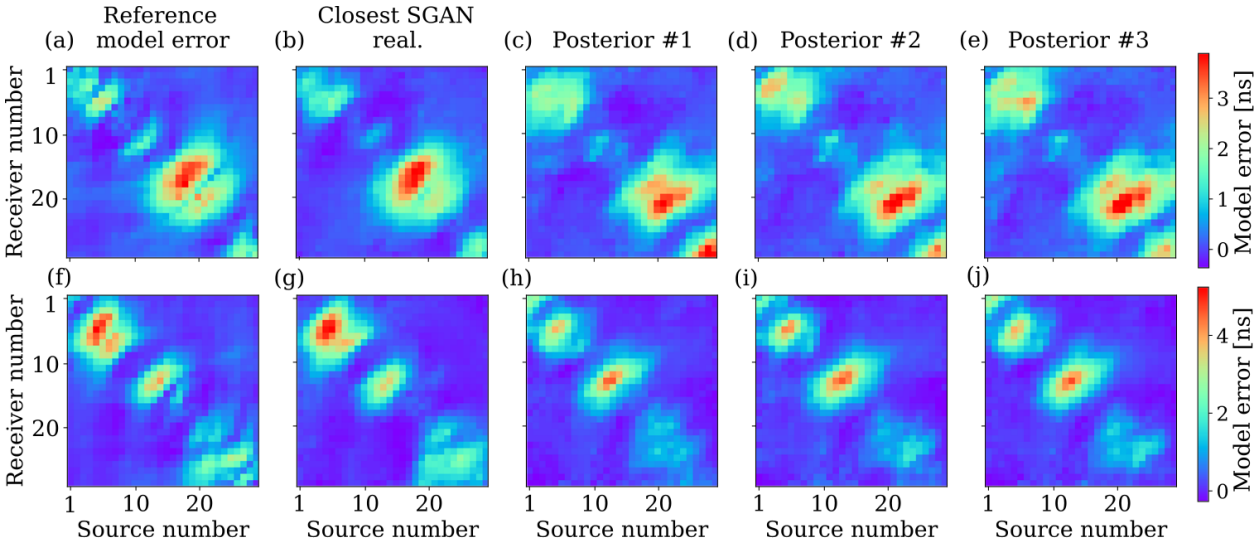

Figure 8: Model errors for Test Case 1 ($\boldsymbol{\eta}^{eikonal-SR}$) representing the discrepancy between the eikonal and straight-ray solvers. (a) and (f) are reference model errors calculated based on reference Models 1 and 2 in Figures 7a and 7f, respectively, (b) and (g) are the corresponding closest SGAN model-error realizations obtained from pixel-to-pixel inversion and (c)-(e) and (h)-(j) are posterior samples obtained from inversion with the SGAN-ME approach.

Table 2: Inversion results for Test Case 1 ($\boldsymbol{\eta}^{eikonal-SR}$) in terms of the subsurface model considering $g^{LF} = g^{SR}$ and $g^{HF} = g^{eikonal}$. The RMSE$_{\boldsymbol{\Phi}}$ and SSIM values are average values of the posterior samples. The RMSE$_{\boldsymbol{\Phi}}$ of each posterior sample was calculated on porosity values with respect to the corresponding reference model. The SSIM was calculated on normalized images in the range of $[0, 1]$. The SSIM can take values between -1 and 1, where 1 indicates identical images. The RMSE$_{\mathbf{d}}$ represents the data fit with respect to the observed data and is an average value over the last draws from the eight MCMC chains. For more details see Appendix B.

Model	Inv. approach	RMSE$_{\boldsymbol{\Phi}}$ [-]	SSIM [-]	RMSE$_{\mathbf{d}}$ [ns]
	True	0	1	0.5
1	straight-ray	0.016	0.72	0.77
	Covariance	0.016	0.74	0.75
	SGAN-ME	0.015	0.79	0.53
	eikonal	0.013	0.78	0.53
	Closest SGAN real.	0.010	0.83	0.62
2	straight-ray	0.021	0.72	0.87
	Covariance	0.018	0.75	1.17
	SGAN-ME	0.017	0.78	0.55
	eikonal	0.017	0.78	0.55
	Closest SGAN real.	0.013	0.83	0.63

Table 3: Inversion results in terms of model-error estimation for the two considered reference models (1 and 2) and Test Case 1 ($\boldsymbol{\eta}^{eikonal-SR}$) and 2 ($\boldsymbol{\eta}^{FDTD-SR}$) . The given RMSE and SSIM values are average values of the posterior samples of model errors. The RMSE of each posterior sample was calculated with respect to the corresponding reference model error. The SSIM was calculated on normalized images in the range of $[0, 1]$. The SSIM can take values between -1 and 1, where 1 indicate identical images. The error recovery represents the fraction of mean-squared-error (MSE) of posterior samples MSE($\boldsymbol{\eta}_{app}, \boldsymbol{\eta}_{ref}$) compared to the MSE($\boldsymbol{\eta}_{ref}, 0$) of the reference model with respect to 0 and can range between 0% to 100%. For more details see Appendix B.

Model error	Model	Inv. approach	RMSE [ns]	SSIM [-]	Error recovery [%]
		True	0	1	100
$\boldsymbol{\eta}^{eikonal-SR}$	1	SGAN-ME	0.67	0.56	51
		Closest SGAN real.	0.23	0.87	94
		Closest database real.	0.29	0.87	90
	2	SGAN-ME	0.66	0.64	67
		Closest SGAN real.	0.29	0.86	94
		Closest database real.	0.54	0.63	77
$\boldsymbol{\eta}^{FDTD-SR}$	1	SGAN-ME	0.49	0.68	74
		Closest SGAN real.	0.24	0.88	94
		Closest database real.	0.33	0.85	88
	2	SGAN-ME	0.63	0.72	71
		Closest SGAN real.	0.32	0.89	92
		Closest database real.	0.56	0.71	78

The maximum-likelihood solution together with the $\text{RMSE}_{\boldsymbol{\Phi}}$ and SSIM values in Table 4 suggest that the SGAN-ME results capture both the magnitude and structure of porosity for Model 1 and is the closest to values observed for the closest SGAN realization, having $\text{RMSE}_{\boldsymbol{\Phi}}$ of 0.0014 ns (versus 0.010 ns) and SSIM value of 0.75 (versus 0.83). The SGAN-ME approach is also able to recover large portions of the model error (74% error recovery) for this reference model (Table 3). The high-porosity zone between 0.5 and 1.5 m depth is wider in the all solutions compared to reference Model 1, although less visible in the SGAN-ME solution. Similarly, as was found previously for the case of $\boldsymbol{\eta}^{eikonal-SR}$, the covariance solution consistently overestimates the porosity around 3 m depth for Model 1 (Figs. 7c and 11c). All compared approaches underestimate the high porosity zone between ~ 3.8-4.5 m depth and overestimate the low-porosity zone between 5-5.5 m depth.

As for Model 2, the porosity structure between 0 and 1 m is better defined in the SGAN-ME solution compared to the other approaches. The porosity zone between 1.8 and 2.8 m depth is overestimated in the right hand side of the SGAN solution. This part of the subsurface model is covered by receivers 10-15. Indeed, the posterior samples displayed in Figure 12h-j show a larger diagonal feature between receivers 10-15 and sources 10-15 than in the reference model error for those source-receiver pairs. Nonetheless, the inferred SGAN-ME model error recovers 71% of the true model error (Table 3).

For both reference models, the $\text{RMSE}_{\boldsymbol{\Phi}}$ of the covariance and straight-ray approaches are increasing or remain the same when going from $\boldsymbol{\eta}^{eikonal-SR}$ to $\boldsymbol{\eta}^{FDTD-SR}$. Interestingly, the SGAN-ME inversion result corresponding to Model 1 improves from $\boldsymbol{\eta}^{eikonal-SR}$ to $\boldsymbol{\eta}^{FDTD-SR}$, with $\text{RMSE}_{\boldsymbol{\Phi}}$ decreasing from 0.015 to 0.014 while the SSIM value decreases from 0.79 to 0.75. This improvement in $\text{RMSE}_{\boldsymbol{\Phi}}$ score can be linked to better error recovery, which increases from 51% for $\boldsymbol{\eta}^{eikonal-SR}$ to 74% for $\boldsymbol{\eta}^{FDTD-SR}$. Notice that in both types of model errors the closest SGAN model-error realizations obtained by pixel-based inversion (Figs. 8b and 8g for $\boldsymbol{\eta}^{eikonal-SR}$ and Figs. 12b and 12g for $\boldsymbol{\eta}^{FDTD-SR}$) strongly resemble their reference model errors and their error recovery is between 92 to 94%, further exemplifying the ability of the SGAN to represent model errors. Tables 2 and 4 and Figure 9 show that the SGAN-ME approach is able to fit the data equally well in both test cases and approaches the noise contamination level. Finally, we observe that the posterior samples in the form of $\text{RMSE}_{\boldsymbol{\Phi}}$ and SSIM distribution (Figure 10c,d,g,h) show similar patterns as for Test Case 1, in the sense that the SGAN-ME approach generally results in $\text{RMSE}_{\boldsymbol{\Phi}}$ and SSIM distributions that rank higher than the straight-ray and covariance approaches. Again, the SGAN-ME distributions are characterized with intermediate widths as opposed to the covariance approach for which $\text{RMSE}_{\boldsymbol{\Phi}}$ and SSIM values vary widely and to the straight-ray approach for which the distribution is narrow and exhibits the worst statistics.

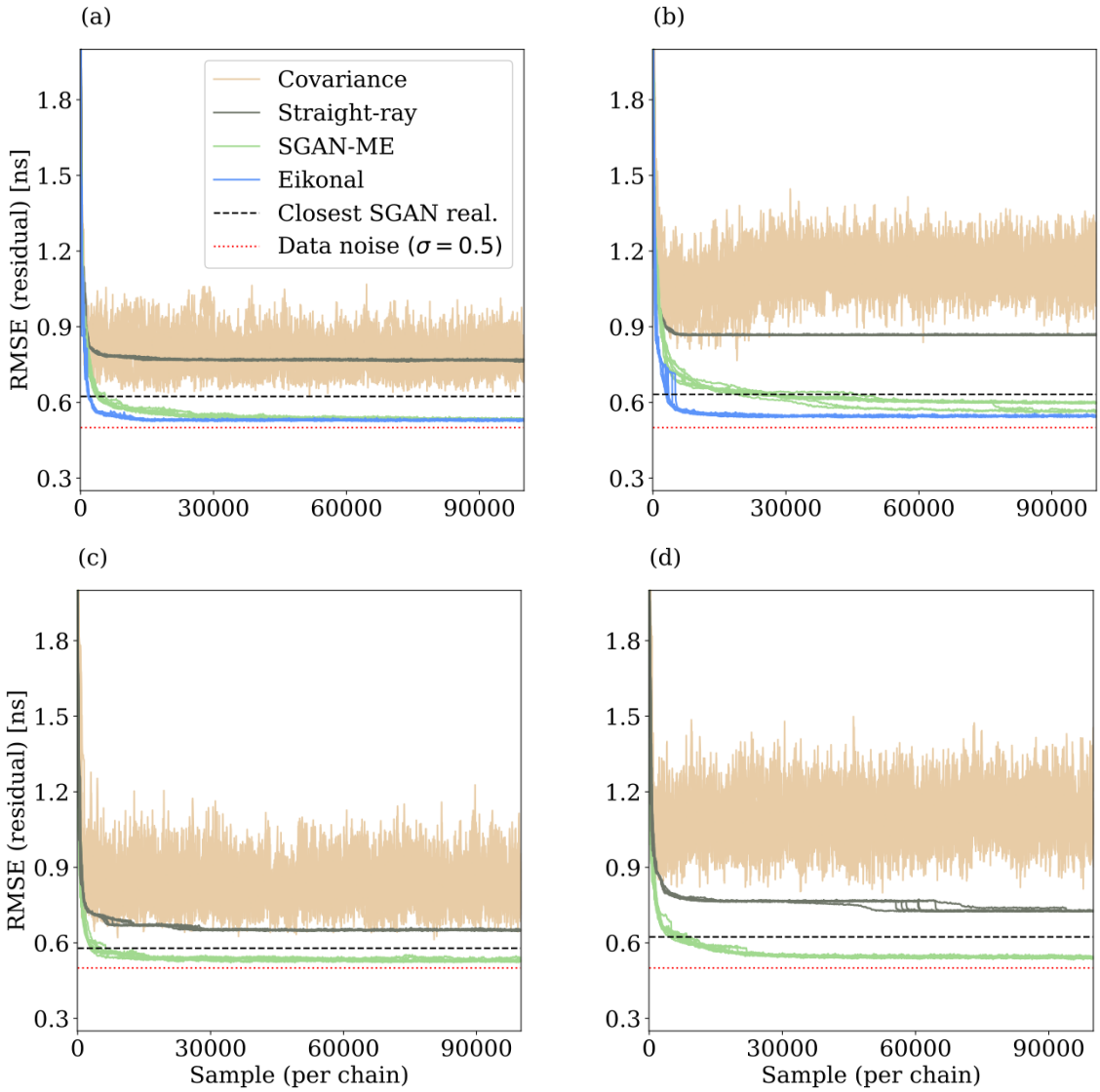

Figure 9: Data fit ($\text{RMSE}_{\mathbf{d}}$) for inversion considering: modelling error $\boldsymbol{\eta}^{eikonal-SR}$ for reference models (a) 1 and (b) 2 and modelling error $\boldsymbol{\eta}^{FDTD-SR}$ for reference models (c) 1 and (d) 2.

Table 4: Inversion results for Test Case 2 ($\boldsymbol{\eta}^{FDTD-SR}$) in terms of the subsurface model considering $g^{LF} = g^{SR}$ and $g^{HF} = g^{FDTD}$. The $\text{RMSE}_{\boldsymbol{\Phi}}$ and SSIM values are average values of the posterior samples. The $\text{RMSE}_{\boldsymbol{\Phi}}$ of each posterior sample was calculated on porosity values with respect to the corresponding reference model. The SSIM was calculated on normalized images in the range of $[0, 1]$. The SSIM can take values between -1 and 1, where 1 indicates identical images. The $\text{RMSE}_{\mathbf{d}}$ represents the data fit with respect to the observed data and is an average value over the last draws from the eight MCMC chains. For more details see appendix B.

Model	Inv. approach	$\text{RMSE}_{\boldsymbol{\Phi}}$ [-]	SSIM [-]	$\text{RMSE}_{\mathbf{d}}$ [ns]
	True	0	1	0.5
1	straight-ray	0.017	0.73	0.65
	Covariance	0.018	0.69	0.84
	SGAN-ME	0.014	0.75	0.53
	Closest SGAN real.	0.010	0.83	0.58
2	straight-ray	0.021	0.71	0.73
	Covariance	0.019	0.74	1.16
	SGAN-ME	0.018	0.76	0.54
	Closest SGAN real.	0.013	0.83	0.62

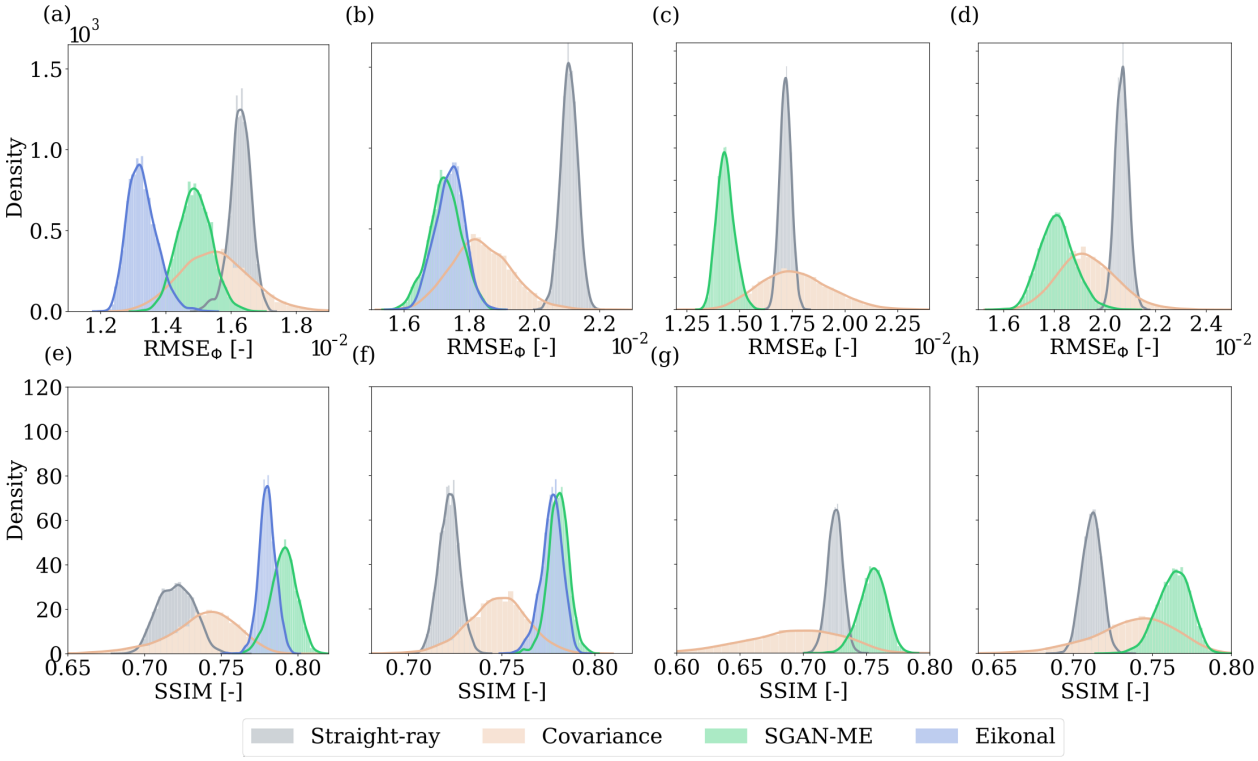

Figure 10: RMSE$_\Phi$ and SSIM distributions of posterior samples for inversion considering: Test Case 1 ($\boldsymbol{\eta}^{eikonal-SR}$) for reference models (a and e) 1 and (b and f) 2 and Test Case 2 ($\boldsymbol{\eta}^{FDTD-SR}$) for reference Models (c and g) 1 and (d and h) 2. The high-fidelity solution is only available in Test Case 1 (blue area in a, b, e and f).

4 Discussion

Our results demonstrate the suitability of our SGAN architecture and training procedure to represent model errors and the ability of SGAN-ME inversions to infer them for a given subsurface model realization (Figs. 8 and 12). Among the considered inversion methods employing a low-fidelity forward solver, the SGAN-ME inversion scored RMSE ($\boldsymbol{\Phi}$ and d) and SSIM values that are the closest to those obtained when the high-fidelity forward eikonal solver is used in the inversion (Table 2). This indicates that inferring the model error during inversion using the SGAN-ME offers an overall better performance compared to ignoring model errors or accounting for them by inflating the error term in the likelihood function following Hansen et al. (2014). Somewhat surprisingly, the straight-ray approach, where model errors are neglected, resulted in subsurface models with relatively minor artifacts (Figs. 7b, 7g, 11b and 11f). This is likely a consequence of the SGAN dimensionality reduction. The dimensionality of the subsurface model domain is reduced in our examples from 2440 parameters to 25 latent parameters, thus, limiting strong artifacts at the expense of the ability to achieve high likelihoods. We expect that more artifacts would appear when inverting the data in the original high-dimensional subsurface model space.

In all tested cases, the SGAN-ME is able to infer meaningful model-error representations (Figs. 8 and 12) ranging between 71 and 74% recovery of the true model error in the $\boldsymbol{\eta}^{FDTD-SR}$ Test case (Table 3). By jointly inferring the subsurface model parameters and the model error, SGAN-ME enables identification and localization of regions in the subsurface model that are prone to large model errors. Some of the inferred model errors are still misplaced (Figure 8c-e) or underestimated (Figure 8h-j). This could suggest that the inferred model error accommodates inadequacies between the subsurface-model realizations that can be generated by the SGAN and the reference subsurface model used to generate the data. Indeed, with 25 parameters it is of course impossible to fully represent all the geostatistical variability of our training image. Tables 2 and 4 reinforce this hypothesis, as they show that the closest SGAN realization obtained from a pixel-to-pixel inversion does not fit the data as well as the eikonal or our SGAN-ME approach, implying a certain bias in the SGAN-ME inversions. A possible solution to address this problem would be to perform a hierarchical inversion in which the standard deviation of the data error is one of the inferred parameters (Malinverno and Briggs, 2004). Initial results with such an hierarchical approach have been inconclusive to date and require further investigation.

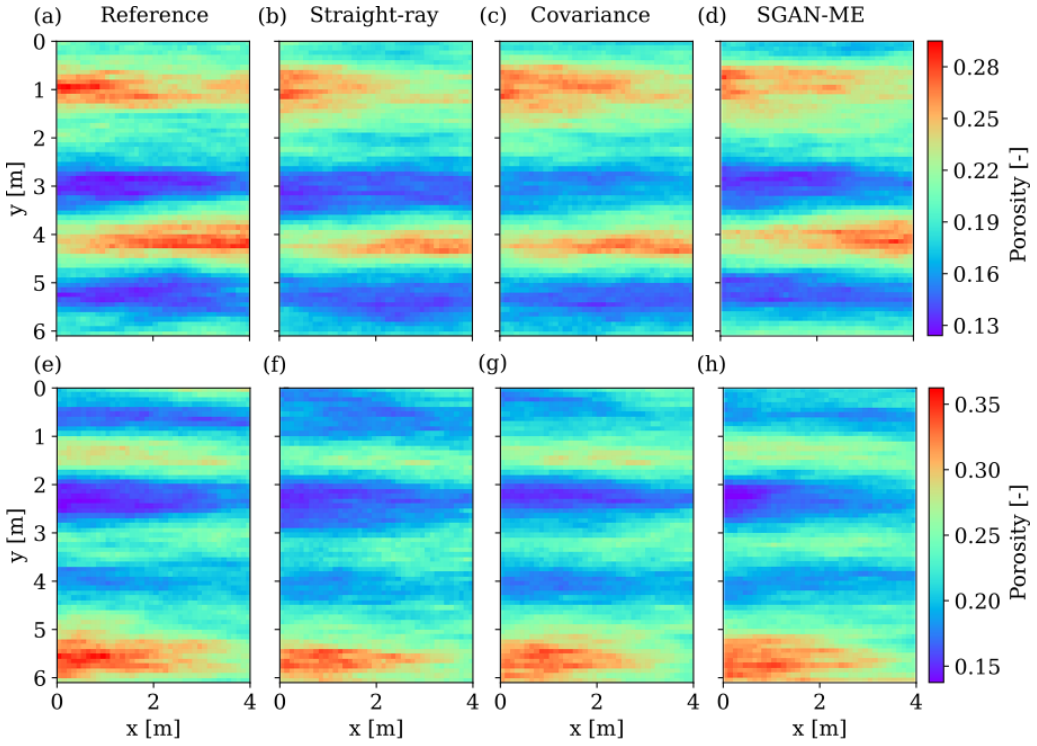

Figure 11: Inversion results for reference models (a) 1 and (e) 2 for Test Case 2 ($\boldsymbol{\eta}^{FDTD-SR}$). (b)-(d) and (f)-(h) are the maximum-likelihood realizations obtained from inversion using the straight-ray, covariance and SGAN-ME approaches. All three approaches use the straight-ray solver for the forward response during inversion, while the observed data were created using the FDTD solver.

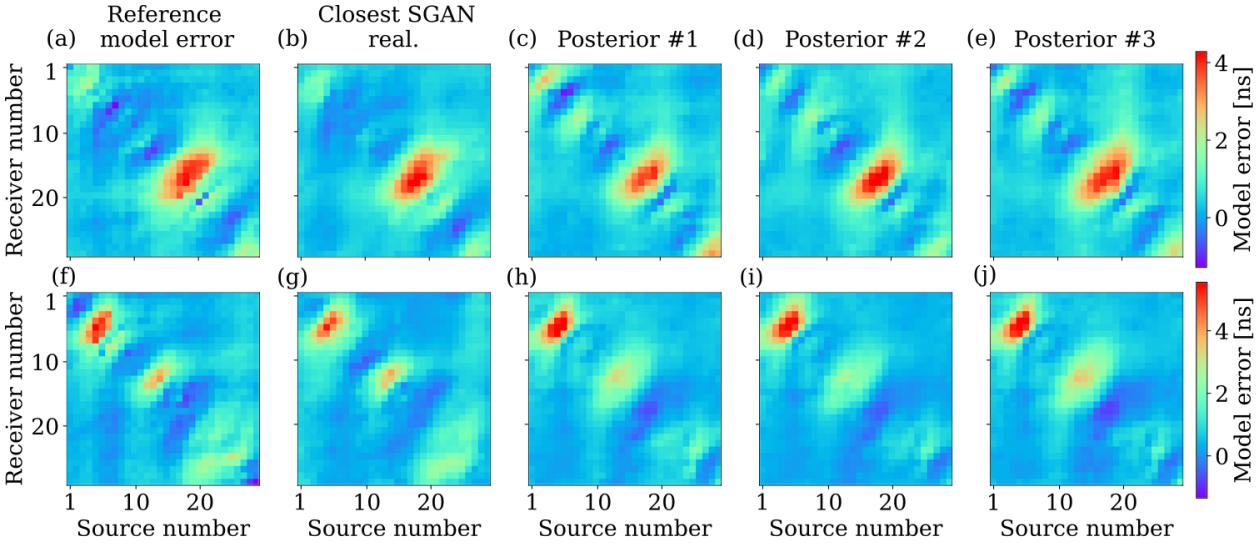

Figure 12: Model errors for Test Case 2 ($\boldsymbol{\eta}^{FDTD-SR}$) representing the discrepancy between the FDTD and straight-ray solvers. (a) and (f) are reference model errors calculated based on reference models 1 and 2 in Figures 11a and 11e, respectively, (b) and (g) are the corresponding closest SGAN model error realizations obtained from pixel-to-pixel inversion and (c)-(e) and (h)-(j) are three posterior samples obtained from inversion with the SGAN-ME approach.

The $\text{RMSE}_{\mathbf{d}}$ values corresponding to SGAN-ME are very similar to those obtained when using the high-fidelity eikonal solver (Table 2). For both types of errors, $\boldsymbol{\eta}^{eikonal-SR}$ and $\boldsymbol{\eta}^{FDTD-SR}$, SGAN-ME is found to fit the data significantly better than the straight-ray and covariance approaches with values close to the noise level of $\sigma = 0.5$ ns (Tables 2 and 4). We have seen that the impact of the type of model-error size on data fit is small in the SGAN-ME approach, indicating its robustness in fitting the data by inferring the model error. The covariance approach is characterized by a large variability of $\text{RMSE}_{\mathbf{d}}$ values throughout the inversion due to the inflation of the likelihood function, and hence a wide range of realizations are accepted. This variability in model realizations is also observed in Figure 10, where the covariance-based $\text{RMSE}_{\boldsymbol{\Phi}}$ and SSIM distributions exhibit the largest variance. The straight-ray approach spans a smaller range of posterior realizations, but those present poor $\text{RMSE}_{\boldsymbol{\Phi}}$ and SSIM scores. In that regard, the SGAN-ME presents a combination of small uncertainty (intermediate posterior widths) and the best $\text{RMSE}_{\boldsymbol{\Phi}}$ and SSIM scores.

In agreement with other approaches treating model errors as the discrepancy between a low- and a high-fidelity solver, we stress that our method is unable to quantify any model errors arising from simplifications in the high-fidelity solver or an inappropriate prior model (training data) of the subsurface properties. As a deep learning method, our approach depends on the availability of training data (i.e. subsurface-model representation and two fidelity-varying forward solvers). Note also that the networks are model and model-error specific, meaning that new training is required if considering a different set-up. Furthermore, our SGAN-ME approach combines multiple nonlinear transformations leading to MCMC convergence issues. Here, we relied on the $DREAM_{(ZS)}$ algorithm and found that convergence was highly sensitive to the chosen jump-rate scaling factor. In the future, it would be beneficial to assess if convergence could be improved by using other MCMC samplers such as gradient-, Hamiltonian-dynamics- (Duane et al., 1987; Neal, 2011) or diffusion- (Roberts et al., 1996; Roberts and Rosenthal, 1998) based samplers.

5 Conclusions

We present a methodology accounting for model errors in Bayesian inversion using deep generative neural networks. In contrast to most existing methods, our approach makes no restrictive Gaussian assumptions about the statistical distribution of the model errors arising from using a fast low-fidelity solver instead of a slow high-fidelity solver. We use SGANs to learn two separate generative models: one for the subsurface model parameters of interest and the other for the model errors. The underlying low-dimensional latent parameterizations are then used to jointly infer the subsurface model parameters and model error via MCMC using the fast low-fidelity forward solver, thereby, allowing for significant speed-up. By doing so, we are able to improve the posterior estimates of subsurface model parameters and model errors. Our SGAN-ME method is shown to perform better than in cases where model errors are ignored or accounted for using a Gaussian error model. In fact, the quality of the posterior solutions is close to results obtained when using a high-fidelity forward solver in the MCMC. By providing posterior distributions of the model errors, it is possible to visualize where model errors occur and to identify regions where inversion results might be less reliable. This information could be used to locally replace low-fidelity simulations with high-fidelity simulations. Our focus has been on model errors due to simplified physics, but our approach and the extension discussed above could also be useful when considering coarse meshes for the forward computations. In addition, our approach could be extended to other fields of geophysics, for example, full-waveform inversion. Even if our SGAN-ME method works well in the considered test examples, we highlight the need to address MCMC instabilities due to the underlying nonlinearity of the SGAN transformation. Since the performance of our approach depends on the quality of the SGAN realizations, there is a need to further advance network architectures and training procedures for both subsurface model parameters and model errors. Further improvements could also be made by training the subsurface model and model error jointly with shared latent parameters or by combining our SGAN-ME approach with deep-learning based surrogate modeling.

6 Acknowledgements

This research was supported by the Swiss National Science Foundation (project number: 184574). We thank associated editor Juan-Carlos Afonso, as well as reviewers Brent Wheelock and Jianwei Ma for their constructive comments. The SGAN and MCMC scripts as well as test examples can be found in the following GitHub repository: `https://github.com/ShiLevy/SGAN_ME`.

Citation

This is a pre-copyedited, author-produced PDF of an article accepted for publication in *Geophysical Journal International* following peer review. The version of record

Shiran Levy, Jürg Hunziker, Eric Laloy, James Irving, Niklas Linde, Using deep generative neural networks to account for model errors in Markov chain Monte Carlo inversion, *Geophysical Journal International*, 2021;, ggab391, https://doi.org/10.1093/gji/ggab391

is available online at: `https://academic.oup.com/gji/advance-article/doi/10.1093/gji/ggab391/6374556`

References

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein Generative Adversarial Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 214–223. PMLR.

Barrash, W. and Clemo, T. (2002). Hierarchical geostatistics and multifacies systems: Boise Hydrogeophysical Research Site, Boise, Idaho. *Water Resources Research*, 38(10):1–18.

Bergen, K., Johnson, P., Maarten, V., and Beroza, G. (2019). Machine learning for data-driven discovery in solid Earth geoscience. *Science*, 363(6433).

Brunetti, C. and Linde, N. (2018). Impact of petrophysical uncertainty on Bayesian hydrogeophysical inversion and model selection. *Advances in Water Resources*, 111:346–359.

Brynjarsdóttir, J. and O-Hagan, A. (2014). Learning about physical parameters: The importance of model discrepancy. *Inverse Problems*, 30(11):114007.

Calvetti, D., Ernst, O., and Somersalo, E. (2014). Dynamic updating of numerical model discrepancy using sequential sampling. *Inverse Problems*, 30(11):114019.

Cui, T., Fox, C., and O'Sullivan, M. J. (2011). Bayesian calibration of a large-scale geothermal reservoir model by a new adaptive delayed acceptance Metropolis Hastings algorithm. *Water Resources Research*, 47(10):[W10521].

Dramsch, J. S. (2020). 70 years of machine learning in geoscience in review. *Advances in Geophysics*, 61:1–55.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222.

Dumoulin, V. and Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.

Ernst, J. R., Green, A. G., Maurer, H., and Holliger, K. (2007). Application of a new 2D time-domain full-waveform inversion scheme to crosshole radar data. *Geophysics*, 72(5):J53–J64.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472.

Giannakis, I., Giannopoulos, A., and Warren, C. (2019). A machine learning-based fast-forward solver for ground penetrating radar with application to full-waveform inversion. *IEEE Transactions on Geoscience and Remote Sensing*, 57(7):4417–4426.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732.

Hansen, T. M., Cordua, K. S., Jacobsen, B. H., and Mosegaard, K. (2014). Accounting for imperfect forward modeling in geophysical inverse problems — Exemplified for crosshole tomography. *Geophysics*, 79(3):H1–H21.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6629–6640.

Hunziker, J., Laloy, E., and Linde, N. (2019). Bayesian full-waveform tomography with application to crosshole ground penetrating radar data. *Geophysical Journal International*, 218(2):913–931.

Irving, J. and Knight, R. (2006). Numerical modeling of ground-penetrating radar in 2-D using MATLAB. *Computers & Geosciences*, 32(9):1247–1258.

Irving, J. D. and Knight, R. J. (2005). Effect of antennas on velocity estimates obtained from crosshole GPR data. *Geophysics*, 70(5):K39–K42.

Jetchev, N., Bergmann, U., and Vollgraf, R. (2016). Texture synthesis with spatial generative adversarial networks. *arXiv preprint arXiv:1611.08207*.

Jin, Z. L., Liu, Y., and Durlofsky, L. J. (2020). Deep-learning-based surrogate model for reservoir simulation with time-varying well controls. *Journal of Petroleum Science and Engineering*, 192:107273.

Kaipio, J. and Somersalo, E. (2007). Statistical inverse problems: Discretization, model reduction and inverse crimes. *Journal of Computational and Applied Mathematics*, 198(2):493–504.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464.

Köpke, C., Irving, J., and Elsheikh, A. H. (2018). Accounting for model error in Bayesian solutions to hydrogeophysical inverse problems using a local basis approach. *Advances in Water Resources*, 116:195–207.

Köpke, C., Irving, J., and Roubinet, D. (2019). Stochastic inversion for soil hydraulic parameters in the presence of model error: An example involving ground-penetrating radar monitoring of infiltration. *Journal of Hydrology*, 569:829–843.

Laloy, E., Hérault, R., Jacques, D., and Linde, N. (2018). Training-image based geostatistical inversion using a spatial generative adversarial neural network. *Water Resources Research*, 54(1):381–406.

Laloy, E., Hérault, R., Lee, J., Jacques, D., and Linde, N. (2017). Inversion using a new low-dimensional representation of complex binary geological media based on a deep neural network. *Advances in Water Resources*, 110:387–405.

Laloy, E. and Vrugt, J. A. (2012). High-dimensional posterior exploration of hydrologic models using multiple-try DREAM$_{(ZS)}$ and high-performance computing. *Water Resources Research*, 48(1).

Le, H. and Borji, A. (2017). What are the receptive, effective receptive, and projective fields of neurons in convolutional neural networks? *arXiv preprint arXiv:1705.07049*.

Linde, N., Ginsbourger, D., Irving, J., Nobile, F., and Doucet, A. (2017). On uncertainty quantification in hydrogeology and hydrogeophysics. *Advances in Water Resources*, 110:166–181.

Malinverno, A. and Briggs, V. A. (2004). Expanded uncertainty quantification in inverse problems: Hierarchical Bayes and empirical Bayes. *GEOPHYSICS*, 69(4):1005–1016.

Mariethoz, G., Renard, P., and Caers, J. (2010). Bayesian inverse problem and optimization with iterative spatial resampling. *Water Resources Research*, 46(11).

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*.

Mosser, L., Dubrule, O., and Blunt, M. J. (2020). Stochastic seismic waveform inversion using generative adversarial networks as a geological prior. *Mathematical Geosciences*, 52(1):53–79.

Müller, S. and Schüler, L. (2020). GeoStat-Framework/GSTools. Zenodo. https://doi.org/10.5281/zenodo.1313628.

Neal, R. M. (2011). MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):113–162.

Pirot, G., Linde, N., Mariethoz, G., and Bradford, J. H. (2017). Probabilistic inversion with graph cuts: Application to the Boise Hydrogeophysical Research Site. *Water Resources Research*, 53(2):1231–1250.

Podvin, P. and Lecomte, I. (1991). Finite difference computation of traveltimes in very contrasted velocity models: a massively parallel approach and its associated tools. *Geophysical Journal International*, 105(1):271–284.

Pride, S. (1994). Governing equations for the coupled electromagnetics and acoustics of porous media. *Physical Review B*, 50(21):15678–15696.

Rammay, M. H., Elsheikh, A. H., and Chen, Y. (2019). Quantification of prediction uncertainty using imperfect subsurface models with model error estimation. *Journal of Hydrology*, 576:764–783.

Roberts, G. O. and Rosenthal, J. S. (1998). Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268.

Roberts, G. O., Tweedie, R. L., et al. (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363.

Ruggeri, P., Irving, J., and Holliger, K. (2015). Systematic evaluation of sequential geostatistical resampling within MCMC for posterior sampling of near-surface geophysical inverse problems. *Geophysical Journal International*, 202(2):961–975.

Seillé, H. and Visser, G. (2020). Bayesian inversion of magnetotelluric data considering dimensionality discrepancies. *Geophysical Journal International*, 223(3):1565–1583.

Subramanian, A. K. and Chong, N. Y. (2019). Mean spectral normalization of deep neural networks for embedded automation. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pages 249–256.

Sun, A. Y., Scanlon, B. R., Zhang, Z., Walling, D., Bhanja, S. N., Mukherjee, A., and Zhong, Z. (2019). Combining physically based modeling and deep learning for fusing GRACE satellite data: Can we learn from mismatch? *Water Resources Research*, 55(2):1179–1195.

Tang, M., Liu, Y., and Durlofsky, L. J. (2020). A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems. *Journal of Computational Physics*, 413:109456.

Tarantola, A. (2005). *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics.

Tarantola, A., Valette, B., et al. (1982). Inverse Problems= Quest for Information. *Journal of Geophysics*, 50(1):159–170.

ter Braak, C. J. and Vrugt, J. A. (2008). Differential Evolution Markov Chain with snooker updater and fewer chains. *Statistics and Computing*, 18(4):435–446.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31.

Tripathy, R. K. and Bilionis, I. (2018). Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *Journal of Computational Physics*, 375:565–588.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2016). Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.

Vrugt, J. A., Ter Braak, C. J., Gupta, H. V., and Robinson, B. A. (2009). Equifinality of formal (DREAM) and informal (GLUE) Bayesian approaches in hydrologic modeling? *Stochastic Environmental Research and Risk Assessment*, 23(7):1011–1026.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

Xiao, D. (2019). Error estimation of the parametric non-intrusive reduced order model using machine learning. *Computer Methods in Applied Mechanics and Engineering*, 355:513–534.

Xu, T. and Valocchi, A. J. (2015). A Bayesian approach to improved calibration and prediction of groundwater models with structural error. *Water Resources Research*, 51(11):9290–9311.

Xu, T., Valocchi, A. J., Ye, M., and Liang, F. (2017). Quantifying model structural error: Efficient Bayesian calibration of a regional groundwater flow model using surrogates and a data-driven error model. *Water Resources Research*, 53(5):4084–4105.

Yu, S. and Ma, J. (2020). Data-driven geophysics: from dictionary learning to deep learning. *arXiv preprint arXiv:2007.06183*.

A Details on SGAN Architecture and training

Below we discuss the SGAN architecture and provide practical information about its training.

A.1 Network architecture

Figure 13 details the architecture of the SGAN used in this study. The learning rate of the generator (ratio of $1:4$ in learning rate between generator and critic) in subsurface-model training is $5e-05$ while it is $1e-06$ in model-error training. We found that using such a low learning rate was essential to avoid artifacts from appearing in the generated images. We used a batch size of 64 even if a batch size of 32 provides similar results. The hyper-parameters of each layer are detailed in Table 5 and include the kernel, stride and padding sizes. We use the RMSProp (Tieleman and Hinton, 2012) optimizer in both generator and critic to update the parameters of the network.

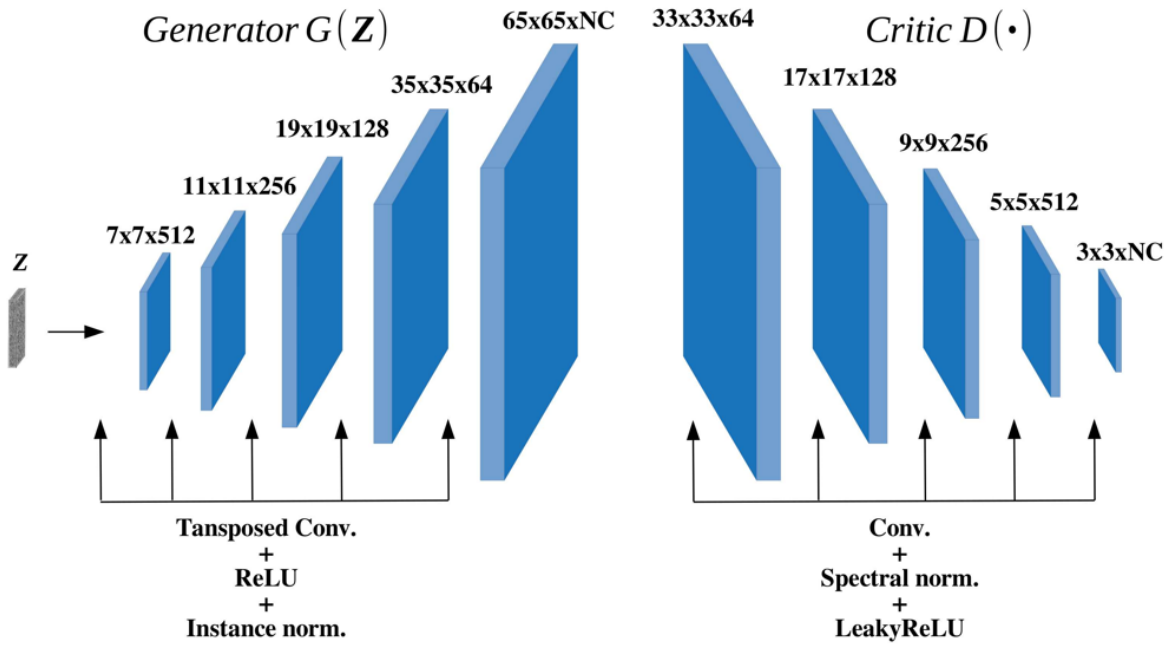

Figure 13: SGAN architecture showing the activation and normalization types and output size after each convolution (/transposed convolution) with NC being the number of image channels (e.g. three channels in RGB images). When training over model errors the critic layers include mean-spectral-normalization as opposed to spectral normalization alone for subsurface-model training.

A.2 Effective receptive field and feature size

A distinct difference between SGANs and GANs is the way information in the latent space is being translated into the image space. GANs usually involve a latent space vector where each latent parameter affects the resulting images globally, while in SGANs the latent parameters are ordered within a 2D/3D tensor and contain local information which overlaps in the image space. One of the limitations arising from using spatially-dependent information within a convolutional network is that a change in the dimensions of the latent space affects the output image size (see eq. (11)). This means that the network output size is determined by the dimensions of the latent space. All input to the critic in SGANs must have the same dimensions, therefore, the dimensions of the TIs should match those of the generated images.

We can easily match image sizes by performing an interpolation on the TI to match the generated image dimensions (or vice versa). Note though that there is an indirect effect of image interpolation on the learning process that is related to the effective receptive field (ERF). The ERF is the area in the input (or output in the case of a generator) influencing a

Table 5: SGAN hyper-parameters.

	layer	kernel	stride	padding
Generator	1	5	2	3
	2	5	2	3
	3	5	2	3
	4	5	2	3
	5	5	2	4
Critic	1	5	2	2
	2	5	2	2
	3	5	2	2
	4	5	2	2
	5	1	2	0

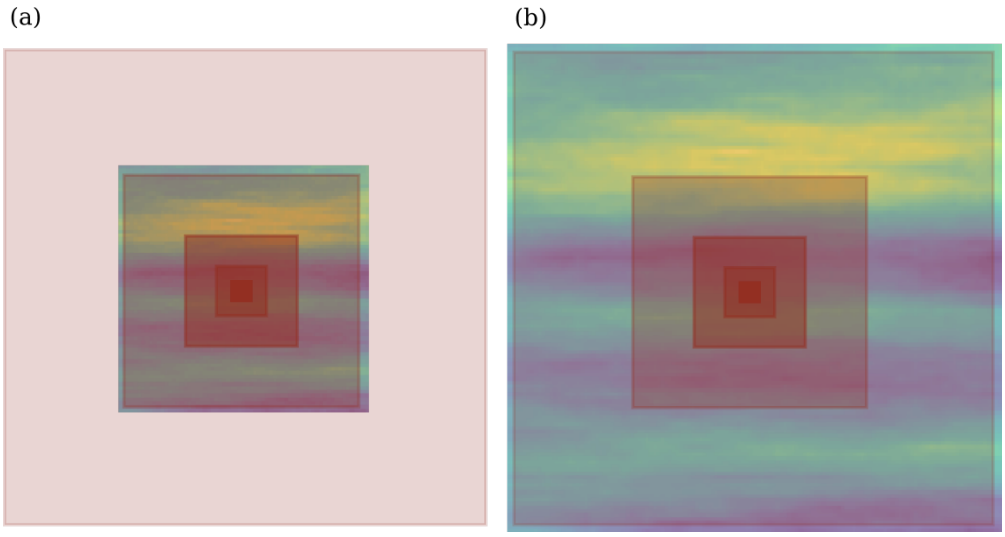

Figure 14: (a) Multi-Gaussian TI of dimensions $65x65$ pixels and (b) the same TI interpolated into $129x129$ pixels, both overlaid by the ERF's of neurons computed for 5 sequential convolutional layers. The ERF is computed given $k = 5$ and $s = 2$ for all layers.

neuron in a given convolutional layer. The ERF is a function of the kernel and stride sizes and can be computed for the l^{th} layer in the following way (Le and Borji, 2017):

$$R_l = R_{l-1} + (k_l - 1) \prod_{i=1}^{l-1} s_i, \tag{16}$$

where R_l and R_{l-1} are the ERF's of a neuron in the current and previous layers, k_l is the kernel size in the current layer, s_i is the stride in layer i and $R_0 = 1$. Although the ERF size does not depend on the size of the image or latent space, an interpolation to the TI affect the network for given kernel and stride sizes. The reason is that for an interpolated TI, features within the image are larger/smaller and therefore, the portion of the features seen by a neuron is changed (see Figure 14). As illustrated in Figure 14, where the ERFs of 5 layers are plotted on top of a TI before and after interpolation for a given network architecture, the resolution in which neurons in each layer 'see' features of difference scales changes with interpolation. This means that some scales cannot be properly resolved which can lead to a mode collapse or a failure of the network to learn the underlying data distribution.

Hence, it is important to test how well the output/input image is covered by the ERF's of neurons in different layers. Since the SGAN was proven to be substantially more sensitive to changes in k or s than in p (padding; see section 2.3), in our work we limited the generated image size using padding when we increased the number of latent parameters.

B Quality measure calculation

Here we expand the information concerning the quantitative measures appearing in Tables 2, 3 and 4. We use RMSE as a metric for model and data fit. The RMSE of the model ($RMSE_{\mathbf{\Phi}}$) is calculated on porosity values of individual posterior realizations (only the last 50% of each chain is considered) with respect to the reference model:

$$RMSE_{\mathbf{\Phi}} = \sqrt{\frac{\sum_{n=1}^{N_{\mathbf{\Phi}}}(\mathbf{\Phi}_{ref} - \mathbf{\Phi}_n)^2}{N_{\mathbf{\Phi}}}}, \tag{17}$$

where $N_{\mathbf{\Phi}}$ is the number of subsurface model parameters. The final reported $RMSE_{\mathbf{\Phi}}$ is the average value of posterior samples. The data RMSE ($RMSE_d$) is the average RMSE value in the last draw of the MCMC chains

$$RMSE_{\mathbf{d}} = \sqrt{\frac{\sum_{n=1}^{N_d}(\mathbf{d} - \mathbf{d}_n^{sim})^2}{N_d}}, \tag{18}$$

where N_d is the number of data points.

The structural similarity (SSIM; Wang et al., 2004) index of two images U and V is a common quantitative measure in image processing. It is calculated using sliding windows $\mathbf{u}$ and $\mathbf{v}$ of dimension $M \times M$ (we use a 7×7 window) subsampling the $[0, 1]$ normalized images,

$$SSIM(\mathbf{u}, \mathbf{v}) = \frac{(2\mu_{\mathbf{u}}\mu_{\mathbf{v}} + C_1)(2\sigma_{\mathbf{uv}} + C_2)}{(2\mu_{\mathbf{u}}^2 + \mu_{\mathbf{v}}^2 + C_1)(2\sigma_{\mathbf{u}}^2 + \sigma_{\mathbf{v}}^2 + C_2)}, \tag{19}$$

where $\mu_{\mathbf{u}}$ and $\mu_{\mathbf{v}}$ are the mean values over $\mathbf{u}$ and $\mathbf{v}$, $\sigma_{\mathbf{u}}^2$ and $\sigma_{\mathbf{v}}^2$ are the respective variances of $\mathbf{u}$ and $\mathbf{v}$ and $\sigma_{\mathbf{uv}}$ is the covariance between $\mathbf{u}$ and $\mathbf{v}$. We follow Wang et al. (2004) and set $C_1 = 0.01$ $C_2 = 0.03$.

The error recovery value is calculated based on MSE values of the reference model error with respect to 0 ($MSE(\boldsymbol{\eta}_{ref}, 0)$) and the MSE of the inferred model error with respect to the reference model ($MSE(\boldsymbol{\eta}, \boldsymbol{\eta}_{ref})$):

$$MSE(\boldsymbol{\eta}_{ref}, 0) = \frac{\sum_{n=1}^{N_{\boldsymbol{\eta}}}(0 - \boldsymbol{\eta}_{ref,n})^2}{N_{\boldsymbol{\eta}}}, \tag{20}$$

$$MSE(\boldsymbol{\eta}_{app}, \boldsymbol{\eta}_{ref}) = \frac{\sum_{n=1}^{N_{\boldsymbol{\eta}}}(\boldsymbol{\eta}_{ref} - \boldsymbol{\eta}_{app,n})^2}{N_{\boldsymbol{\eta}}}, \tag{21}$$

where $N_{\boldsymbol{\eta}}$ is the number of model error parameters. The error recovery is the fraction of the average $MSE(\boldsymbol{\eta}_{app}, \boldsymbol{\eta}_{ref})$ within posterior samples and $MSE(\boldsymbol{\eta}_{ref}, 0)$ given in percentage:

$$ER = \frac{\overline{MSE}(\boldsymbol{\eta}_{app}, \boldsymbol{\eta}_{ref})}{MSE(\boldsymbol{\eta}_{ref}, 0)} * 100\%. \tag{22}$$