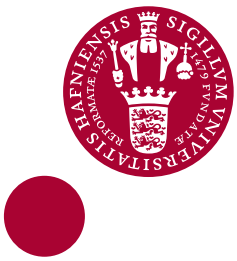

# Efficient Monte Carlo sampling of inverse problems using a neural network-based forward—applied to GPR crosshole traveltime inversion

Hansen, Thomas Mejer; Cordua, Knud Skou

*Published in:*
Geophysical Journal International

*DOI:*
10.1093/gji/ggx380

*Publication date:*
2017

*Document version*
Også kaldet Forlagets PDF

Geophysical Journal International

*Geophys. J. Int.* (2017) **211,** 1524–1533
Advance Access publication 2017 September 13
GJI Marine geosciences and applied geophysics

doi: 10.1093/gji/ggx380

# Efficient Monte Carlo sampling of inverse problems using a neural network-based forward—applied to GPR crosshole traveltime inversion

T.M. Hansen and K.S. Cordua

*Climate and Computational Geophysics, Niels Bohr Institute, University of Copenhagen,* 2100 *Copenhagen East, Denmark. E-mail: tmeha@nbi.ku.dk*

Accepted 2017 September 11. Received 2017 September 8; in original form 2017 March 13

## SUMMARY

Probabilistically formulated inverse problems can be solved using Monte Carlo-based sampling methods. In principle, both advanced prior information, based on for example, complex geostatistical models and non-linear forward models can be considered using such methods. However, Monte Carlo methods may be associated with huge computational costs that, in practice, limit their application. This is not least due to the computational requirements related to solving the forward problem, where the physical forward response of some earth model has to be evaluated. Here, it is suggested to replace a numerical complex evaluation of the forward problem, with a trained neural network that can be evaluated very fast. This will introduce a modeling error that is quantified probabilistically such that it can be accounted for during inversion. This allows a very fast and efficient Monte Carlo sampling of the solution to an inverse problem. We demonstrate the methodology for first arrival traveltime inversion of crosshole ground penetrating radar data. An accurate forward model, based on 2-D full-waveform modeling followed by automatic traveltime picking, is replaced by a fast neural network. This provides a sampling algorithm three orders of magnitude faster than using the accurate and computationally expensive forward model, and also considerably faster and more accurate (i.e. with better resolution), than commonly used approximate forward models. The methodology has the potential to dramatically change the complexity of non-linear and non-Gaussian inverse problems that have to be solved using Monte Carlo sampling techniques.

**Key words:** Ground penetrating radar; Inverse theory; Neural networks; Tomography.

## 1 INTRODUCTION

For geophysical inverse problems, the forward problem refers to the problem of computing some data $\mathbf{d}$, given a set of model parameters $\mathbf{m}$ (typically describing an Earth model) using a function $g$ (typically based on knowledge about a physical relation) :

$$\mathbf{d} = g(\mathbf{m}) \tag{1}$$

The corresponding inverse problem consists of finding information about the model parameters $\mathbf{m}$ given some observed data $\mathbf{d}_{\text{obs}}$:

$$\mathbf{m} = g^{-1}(\mathbf{d}_{\text{obs}}) \tag{2}$$

For most inverse problems, the *inverse* operator $g^{-1}$ either does not exist, or may be non-trivial to obtain. Further, as most inverse problems are underdetermined, the observed data are associated with uncertainty/noise, and $g$ is based on some approximation to the correct physical relation (leading to modeling errors ), more than one model will be able to explain the data within its uncertainty.

A probabilistic approach to inverse problems allows incorporating prior information (through the probability distribution $\rho(\mathbf{m})$) and information about measurement uncertainty and modeling errors (through the likelihood function $L(\mathbf{m})$). The solution to the inverse problem is then the *a posteriori* probability distribution (Tarantola & Valette 1982b):

$$\sigma(\mathbf{m}) = k\ \rho_M(\mathbf{m})\ L(\mathbf{m}), \tag{3}$$

where the likelihood function is given by

$$L(\mathbf{m}) = \int d\mathbf{d} \frac{\rho_D(g(\mathbf{m}))\ \theta(\mathbf{d}|\mathbf{m})}{\mu_D(\mathbf{d})}. \tag{4}$$

$\rho_D(g(\mathbf{m}))$ describes how well the forward response of a specific model $\mathbf{m}$ should match some observed data given measurement uncertainties. $\theta(\mathbf{d}|\mathbf{m})$ represents the modeling error that describes any imperfections in solving the forward problem using eq. (1).

The solution to probabilistically formulated inverse problems is in practice either an analytic description of $\sigma(\mathbf{m})$, such as in the case of linear Gaussian inverse problems (Tarantola & Valette 1982a). Or, in the general case, it can be in form of a sample from $\sigma(\mathbf{m})$

obtained using sampling methods, such as the extended Metropolis algorithm (Mosegaard & Tarantola 1995).

Using the extended Metropolis algorithm to sample the posterior distribution is attractive as it can be applied to, in principle, any inverse problem, with either a linear or non-linear forward operator $g$, as long as (1) the likelihood function can be evaluated and (2) the prior distribution can be sampled. That is, only the forward relation in eq. (1) needs to be evaluated, and the inverse operator in eq. (2) need not be known.

However, Monte Carlo-based sampling methods are computationally very costly, and for many geophysical inverse problems, sampling-based methods may not be useful in practice.

Evaluation of the forward model, eq. (1), is often the single most important contribution to the computational cost of running a sampling algorithm to sample the posterior distribution. Examples exist where a complex and realistic forward model is considered, but then the number of model parameters is either small and/or the number of considered data few. Most applications of using sampling methods to sample the posterior distribution of an inverse problem are based on quite simple forward models that are fast to evaluate (Mosegaard *et al.* 1997; Khan *et al.* 2000; Malinverno 2002; Minsley 2011; Hansen *et al.* 2012; Laloy *et al.* 2012; Cordua *et al.* 2012; Zunino *et al.* 2014). The choice of a simple (and fast) forward model is typically chosen as this is needed to practically make use of sampling methods, even though more accurate forward models most always exist.

The use of such approximate forward models introduce a modeling error that is quantified by $\theta(\mathbf{d}|\mathbf{m})$ in eq. (4). Often the modeling error is ignored in which case the likelihood function, eq. (4), simply becomes $L(\mathbf{m}) = \rho_D(g(\mathbf{m}))$ (Tarantola 2005). However, if the modeling error is ignored, the part of the data that should be treated as (modeling) errors will be treated as data. This can lead to mapping of modeling errors into the posterior distribution, which may lead to artefacts that appear as well-resolved features. Hansen *et al.* (2014) describe a methodology that allow sampling, that is, generating realizations of, the (usually unknown) probability density describing the modeling error, $\theta(\mathbf{d}|\mathbf{m})$, related to a specific choice of a priori model $\rho(\mathbf{m})$. Further they demonstrate how to infer a Gaussian model of the modeling error, $\theta(\mathbf{d}|\mathbf{m}) \sim \mathcal{N}(\mathbf{d}_t, \mathbf{C}_t)$. This is convenient as, if the noise on the data is Gaussian $\rho_D(g(\mathbf{m})) \sim \mathcal{N}(\mathbf{d}_0, \mathbf{C}_d)$ and the modeling error is Gaussian, then these Gaussian probability densities combine though addition of their mean and the covariances (Tarantola 2005). The full likelihood in eq. (4) is then also Gaussian distributed over $\mathbf{d}$ as $L(\mathbf{m}) = \rho_D(g(\mathbf{m})) \sim \mathcal{N}(\mathbf{d}_0 + \mathbf{d}_t, \mathbf{C}_d + \mathbf{C}_t)$. This is important as the uncertainty of data is very often described by a Gaussian probability density, and because this allows taking into account modeling errors without having to evaluate the integral in eq. (4), which will in most cases be computationally intractable or expensive.

In the following, the main idea is to use this methodology of quantifying the modeling error to allow using a neural network as a computationally efficient approximation to the forward problem, while at the same time accounting for the errors due to the use of this approximation.

### 1.1 Solving the forward problem using regression networks

Both the forward and inverse problems in eqs (1) and (2) describe a (possibly non-linear) mapping between, which is usually, one set of continuous parameters to another set of continuous parameters (Krasnopolsky & Schiller 2003). Supervised machine learning provides a number of methods that allow inferring this non-linear mapping operator from a set of available training data that consists of a set of data and model parameters [**D**, **M**]. We will refer to these methods in general as regression networks. Examples of these methods are support vector regression (Smola & Schölkopf 2004), Gaussian processes (Williams & Rasmussen 1995), regression trees (Breiman *et al.* 1984) and neural networks (Bishop 1995) to name a few.

Regression networks can be used to estimate the inverse operator $g^{-1}$, eq. (2), such that a model (a solution to the inverse problem) can be directly estimated from data, as demonstrated by Hoole (1993), Amari & Cichocki (1998), Gagunashvili (2010), Maiti *et al.* (2012), Singh *et al.* (2013) and Li *et al.* (2015). It is, though, not trivial to properly account for the uncertainty related to data and the model, and not trivial to fully describe the posterior distribution (hence nor the uncertainty related to the estimated solution model).

Regression networks can also be used to estimate the forward operator $g$, eq. (1) (Poulton 2001; Krasnopolsky & Schiller 2003). $g$ is typically parametrized using some model, and the parameters describing this model is inferred from an available training data set [**D**, **M**] representing **d** and **m** in eq. (1).

In the following, it is proposed to replace a computationally expensive forward problem, $g$ in eq. (1), with a regression network. Subsequently, the associated modeling error $\theta(\mathbf{d}|\mathbf{m})$ related to the use of this regression network is quantified using the approach of Hansen *et al.* (2014). The main idea is that this will allow a very fast evaluation of the forward problem and, hence, the likelihood $L(g(\mathbf{m}))$, which again may allow for a more efficient approach for sampling the posterior distribution $\sigma(\mathbf{m})$.

The developed methodology will be applicable for sampling of the posterior distribution related to inverse problems in general. As an example, the (forward and) inverse problem related to first arrival traveltime inversion of crosshole ground penetrating radar (GPR) data will be considered. An 'ideal' forward model will be based on 2-D full-waveform modeling followed by automatic first arrival picking. As an example, a multilayer perceptron neural network will be considered as an approximation of the forward problem, but note that in practice any regression-type network can be used. The modeling error related to the neural network-based forward will be quantified through a Gaussian probability distribution. Finally, the neural network-based forward model will be compared to using other widely used approximate forward models (based on physical approximations) in formulating and sampling of the posterior distribution of the inverse problem.

## 2 FIRST ARRIVAL CROSSHOLE TRAVELTIME INVERSION

Inversion of first traveltime data is a widely studied inverse problem, both with respect to elastic seismic and electromagnetic wave propagation (Bording *et al.* 1987; Scales 1987; Zelt & Smith 1992; Schuster & Quintus-Bosz 1993). The inverse problem consists of inferring information about the velocity field that has caused the observed decay in traveltime for specific phases in a wavefield. A traveltime-based inverse problem called 'crosshole first arrival time inversion', will be considered here, which is related to inversion of the first arrival time delay of a wavefield propagating from a (set of) source location(s) in one borehole, to a set of receiver locations in another borehole. The inverse problem consists of inferring information about the velocity distribution (described by **m**) between the boreholes.

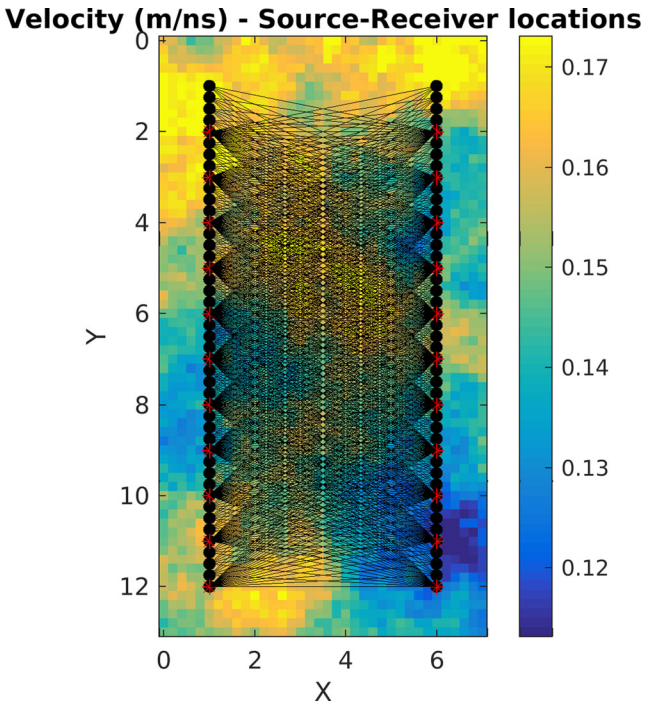


**Figure 1.** Reference velocity model (m ns$^{-1}$) and recording geometry. ND = 702 pairs of sources (red crosses) and receivers (black dots) are represented by a connecting black line.

While initially being developed in relation to seismic traveltime data (McMechan 1983; Hole 1992), this type of traveltime inversion has also been widely studied using near-surface GPR methods, where the traveltime delay of a propagating electromagnetic wave is recorded between boreholes (Maurer & Green 1997).

Inversion of crosshole GPR data is a widely considered inverse problem using both deterministic [e.g. Holliger *et al.* (2001); Linde *et al.* (2006); Ernst *et al.* (2007a); Looms *et al.* (2008b); Cordua *et al.* (2009); Meles *et al.* (2010); Dafflon *et al.* (2011); Klotzsche *et al.* (2013)] and probabilistic approaches [e.g. Gloaguen *et al.* (2007); Giroux *et al.* (2007); Nielsen *et al.* (2010); Hansen *et al.* (2013); Linde & Vrugt (2013)].

A synthetic case is considered, in which the reference model, $\mathbf{m}_{\text{ref}}$, of size 7 × 13 m (36 × 66 pixels of size 0.2 m × 0.2 m) is generated from a multivariate Gaussian probability distribution, with a mean velocity of 0.14 m ns$^{-1}$, a variance of 0.000215 m$^2$ ns$^{-2}$ and a spherical-type covariance model with an isotropic range of 6 m. This is chosen similar to the parameters inferred in Looms *et al.* (2010) using the same geometrical setup as in Hansen *et al.* (2013). A set of 702 noise-free traveltime data is computed using 2-D finite-difference modeling of electromagnetic wave propagation followed by automatic first arrival picking, using the method described by Hansen *et al.* (2014), based on Ernst *et al.* (2007a), Ernst *et al.* (2007b), Molyneux & Schmitt (1999) and Cordua *et al.* (2012). This forward model is referred to as $g_{\text{fd}}$ and will, from heron, be considered the reference for solving the forward problem. The calculated noise-free data are then contaminated with uncorrelated zero-mean Gaussian noise with a standard deviation of 0.1 m ns$^{-1}$, $\mathcal{N}(0, 0.1^2)$, mimicking measurement uncertainty, to obtain a reference data set $\mathbf{d}_{\text{ref}}$.

Fig. 1 shows the reference velocity model, $\mathbf{m}_{\text{ref}}$, and the locations of ND = 702 sets of source and receiver locations, as identified by connecting ray paths. Fig. 2 shows the corresponding simulated waveform data, as well as the automatically picked first arrival traveltime data.

### 2.1 Using a neural network as an approximation to the forward problem

As discussed, many types of supervised machine-learning regression algorithms exist that can be used to obtain an approximation to the reference forward mapping $g_{\text{fd}}$. As an example, a simple two-layer feedforward neural network, that is, with one hidden layer, is considered to replace the accurate forward model $g_{\text{fd}}$. This neural network-based approximation to the forward model will be referred to as $g_{\text{nn}}$. Such a neural network that can estimate each of the ND = 702 data, $d_k$, corresponding to a specific model $\mathbf{m} = [m_1, m_2, \ldots, m_{\text{NM}}]$ can be formulated as

$$d_k = h_1\left(\sum_j^{\text{NH}} w_{jk}^2\, h_2\left(\sum_i^{\text{NM}} w_{ij}^1 m_i\right)\right), \tag{5}$$

where NM = 2376 is the number of model parameters and NH the number of hidden units (Bishop 1995). In this case, NH = 80 hidden units is considered. $h_1$ and $h_2$ refer to two activation functions, in this case both chosen to be of sigmoidal type. $w_{ij}^1$ and $w_{jk}^2$ refer to the weights of the units in the first and second layers. Given enough units in the hidden layer, any continuous mapping can be arbitrarily accurately represented (i.e. including $g_{\text{fd}}$), given a large enough training set (Bishop 1995). The main reason a two-layer feedforward network is considered, is that once trained, the evaluation of the network on a new model (i.e. solving the forward problem) is potentially very fast, and faster than, for example, using support vector regression.

The weights $w_{ij}^1$ and $w_{jk}^2$ must be chosen such that the neural network will be able to estimate the data $\mathbf{d}$ given a specific model $\mathbf{m}$ in an optimal way. This is obtained by 'training' the neural network based on a training data set consisting of NT sets of data and model parameters [$\mathbf{D}$, $\mathbf{M}$], such that $\mathbf{D}$ will be of size [ND, NT] and $\mathbf{M}$ of size [NM, NT]. Training a network in itself can be considered an inverse problem (Prato & Zanni 2008). Most often it is treated as an optimization problem, that is traditionally solved using the backpropagation algorithm.

The quality of the inferred forward operator is closely related to the size of the available training data set, NT. In some applications of neural networks obtaining a large training data set can be a challenge. However, in the context of probabilistically formulated inverse problems, the prior distribution $\rho(\mathbf{m})$ is, by construction, chosen prior to inversion. Further, it can be assumed that realizations from this prior distribution can be generated, as this is needed by the extended Metropolis algorithm (Mosegaard & Tarantola 1995). Therefore, in a probabilistic setting, it is trivial to generate an arbitrarily large sample of the prior as $\mathbf{M}$ consisting of NT realizations. Then, a corresponding set of data $\mathbf{D}$ can be simulated by evaluating the forward model (which may be computationally expensive) NT times.

Five training data sets based on NT = [1000, 5000, 10 000, 20 000, 40 000] sets of model and data parameters, [$\mathbf{D}$, $\mathbf{M}$], are considered. All the considered models are generated as unconditional realizations from the prior distribution $\rho(\mathbf{m})$, and the corresponding data computed using the exact, but expensive, forward model based on finite-difference modeling, $g_{\text{fd}}$, described previously.

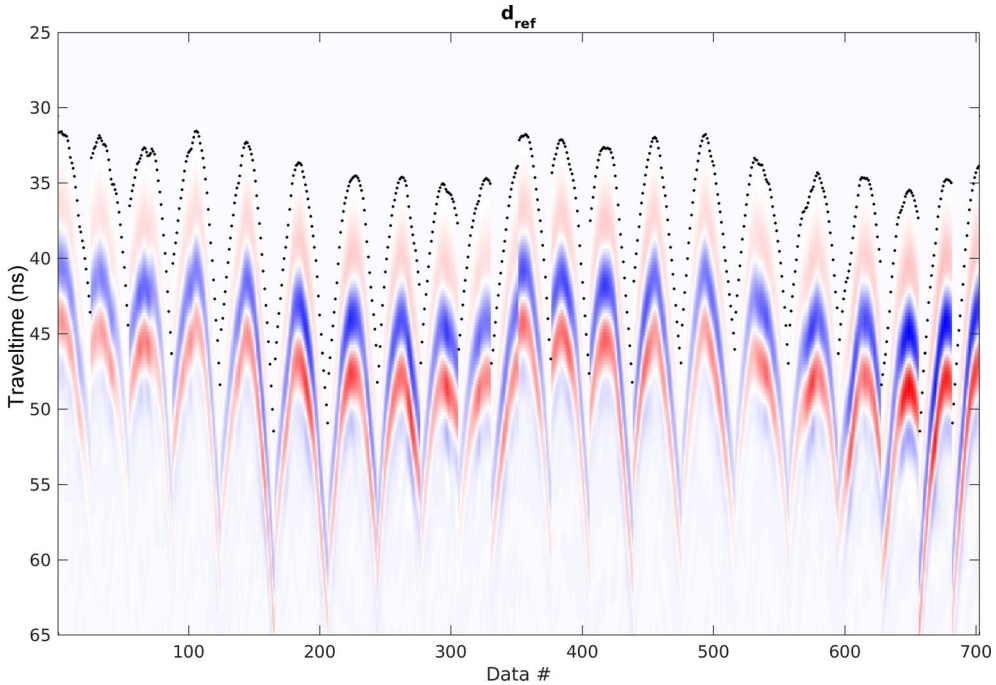


**Figure 2.** Simulated wavefield, observed for 702 sets of sources and receivers (as illustrated in Fig. 1), as well as automatic picks of first arrival traveltimes (black dots).

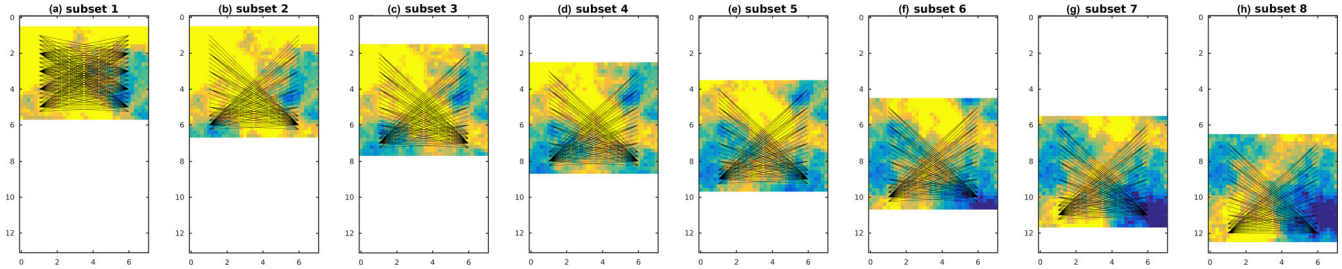


**Figure 3.** 702 sets of sources and receivers split into eight subsets. The 'rays' indicate used pairs of source and receiver. Background colour indicates the used model parameters (white colour refers to model parameters not used).

The network can be trained directly on the full [**D**, **M**] training set. However, acknowledging that the traveltime between a source and a receiver is mostly sensitive to an area between, and in the vicinity of the ray path between the source and receiver, the 702 data are split into eight subsets of data and corresponding model parameters. A unique neural network of the form in eq. (5) is then assigned to each subset of the data. Fig. 3 illustrates which model parameters and sets of sources and receivers that are considered for each subset.

The weights for each of the eight neural networks are found by minimizing the mean-squared error between known data **D** and data predicted by the neural network, using the Levenberg–Marquardt algorithm for backpropagation. The Matlab Neural Network Toolbox (Beale *et al.* 2016) is used for this.

This results in five forward models referred to as $g_{1000}$, $g_{5000}$, $g_{10000}$, $g_{20000}$ and $g_{40000}$, where the index defines the sample size used to train the neural network.

Note that the specific type of, and layout of, the neural network chosen here is just an example. Any method that can be used to estimate the forward model mapping between data and model parameters can in principle be used in the following framework.

## 2.2 Quantifying the modeling error

As the number of units in the neural network, and the size of the training data set, is finite, $g_{\mathrm{nn}}$ will provide an approximation of $g_{\mathrm{fd}}$. In other words, using $g_{\mathrm{nn}}$ as opposed to $g_{\mathrm{fd}}$ will lead to a (forward) modeling error. A sample of the forward modeling error can be obtained by comparing the difference in the forward response of $g_{\mathrm{nn}}$ and $g_{\mathrm{fd}}$ from a set of Nr realizations of the prior model (Hansen *et al.* 2014). Here, Nr = 6000 realizations from the prior has been generated as $\mathbf{M}^* = [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{\mathrm{Nr}}]$. For each model in $\mathbf{M}^*$, a reference data set has been computed using $g_{\mathrm{fd}}$ to obtain $\mathbf{D}_{\mathrm{fd}}$, and using $g_{\mathrm{nn}}$ to obtain $\mathbf{D}_{\mathrm{nn}}$. Then, a sample of the modeling error $\theta(\mathbf{d}|\mathbf{m})$ can be obtained as

$$\mathbf{D}_\theta = \mathbf{D}_{\mathrm{fd}} - \mathbf{D}_{\mathrm{nn}} \tag{6}$$

Finally, a Gaussian model of the modeling error, $\mathcal{N}(\mathbf{d}_t, \mathbf{C}_t)$, can be estimated from $\mathbf{D}_\theta$ using the methodology described in Hansen *et al.* (2014). This allows treating the Gaussian modeling error as Gaussian noise on the data (e.g. Mosegaard & Tarantola 2002).

For comparison, two widely used approximations for computing the first arrival are also considered. $g_{\mathrm{ray}}$ refers to the linear straight ray approximation, and $g_{\mathrm{eik}}$ to the non-linear bended ray

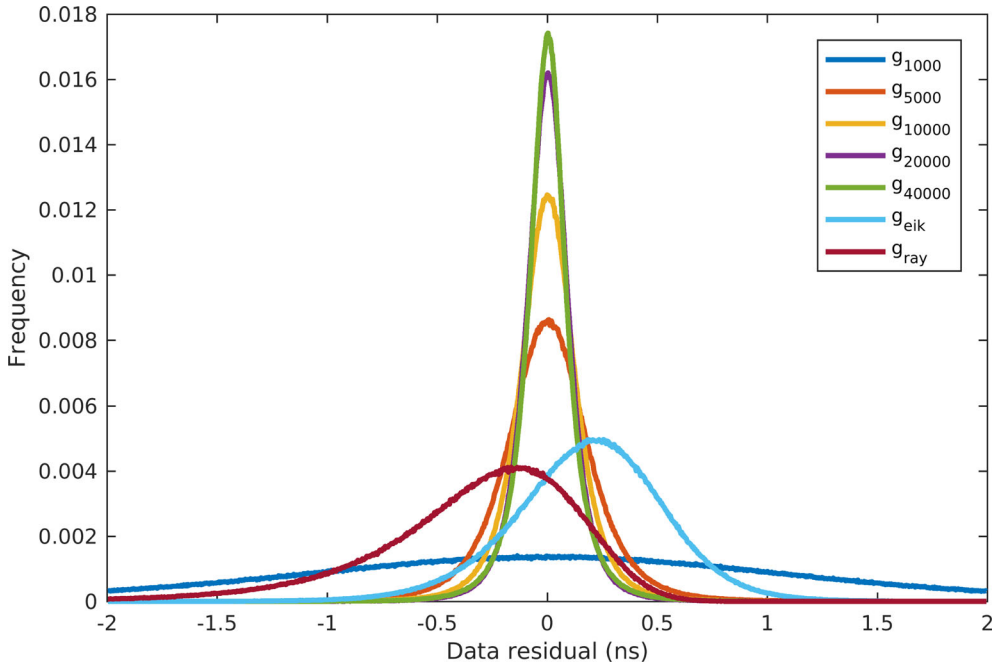


**Figure 4.** 1-D marginal distribution of the modeling error based on 6000 realizations of the modeling error ($\mathbf{D}^*$).

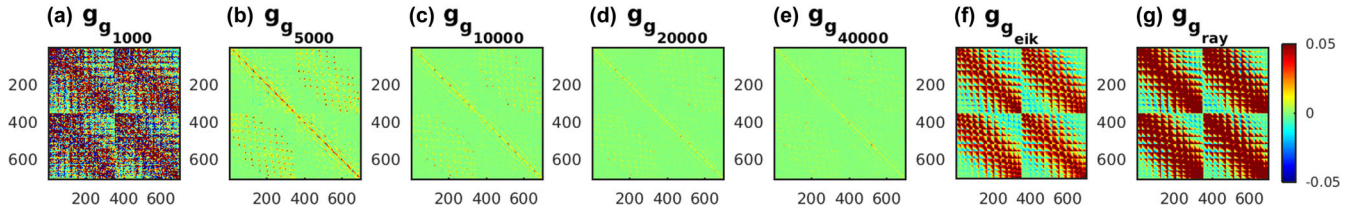


**Figure 5.** Estimated covariance matrix, $\mathbf{C}_t$, for the Gaussian modeling error related to seven different forward models.

**Table 1.** Row (1): mean modeling error, $\mathbf{d}_t$. Row (2): standard deviation of a 1-D uncorrelated Gaussian model for the modeling error, from $\mathbf{C}_t$. Row (3): computation time for solving the forward problem. Row (4): computational speed up of evaluating the forward model relative to using $g_{\text{fd}}$. Row (5): Average correlation coefficient between a set of realizations from the posterior distribution and the reference model. Row (6): mean difference (error) between a set of realizations from the posterior distribution and the reference model.

| Forward model | $g_{1000}$ | $g_{5000}$ | $g_{10000}$ | $g_{20000}$ | $g_{40000}$ | $g_{\text{eik}}$ | $g_{\text{ray}}$ | $g_{\text{fd}}$ |
|---|---|---|---|---|---|---|---|---|
| Mean $\mathbf{d}_t$ (ns) | −0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.18 | −0.34 | |
| Mean $\sigma_t$ (ns) | 1.1 | 0.23 | 0.17 | 0.13 | 0.13 | 0.35 | 0.47 | |
| Time (ms) | 1.9 | 1.9 | 1.7 | 2.0 | 2.1 | 32.1 | 5.7 | 2878.2 |
| Time (speedup) | 1538 | 1490 | 1657 | 1459 | 1386 | 90 | 506 | 1 |
| Correlation coefficient | 0.53 | 0.67 | 0.62 | 0.72 | 0.71 | 0.64 | 0.61 | |
| Mean error (m ns$^{-1}$) | 0.0098 | 0.0081 | 0.0088 | 0.0078 | 0.0078 | 0.0088 | 0.0089 | |

approximation based on the eikonal solution to the wave equation (Vidale 1990; Linde *et al.* 2006; Looms *et al.* 2008a, see more details on these types of forward models in Hansen *et al.* 2014). Fig. 4 shows the 1-D marginal probability distribution of the modeling error, obtained as the histogram of the sample of the modeling error, $\mathbf{D}_\theta$. Fig. 5 shows the covariance model of the Gaussian model describing the modeling error, $\mathbf{C}_t$, for each of the seven approximations (five neural network based and two ray based) to the forward models. The mean $\mathbf{d}_t$ and standard deviation of the inferred Gaussian modeling error is given in Table 1.

Figs 4 and 5 and Table 1 (first four rows) reveal the following:

(1) The use of any of the neural network-based forward model, $g_{\text{nn}}$, results in a modeling error bias, $\mathbf{d}_t$, very close to zero. This is not surprising as the neural network is optimized to minimize mean-squared error.

(2) The magnitude of modeling error related to using $g_{\text{nn}}$ decreases as the size of the training set used for learning the neural network increases. Using a training set larger than $N_t = 5000$ results in a smaller standard deviation of the prediction error than using both $g_{\text{ray}}$ and $g_{\text{eik}}$. When the training set is larger than $N_t = 20\,000$ no significant change in the magnitude of the modeling error is noted (in the present case the modeling error due to using $g_{40000}$ is slightly higher than using $g_{20000}$).

(3) The use of $g_{\text{nn}}$ leads to a less correlated covariance model as opposed to using $g_{\text{eik}}$ and $g_{\text{ray}}$, as seen in the (lack of) off-diagonal elements of the estimated covariance model (Fig. 5).

(4) Evaluating any of the neural network-based forward models is more than three orders of magnitude faster than evaluating the

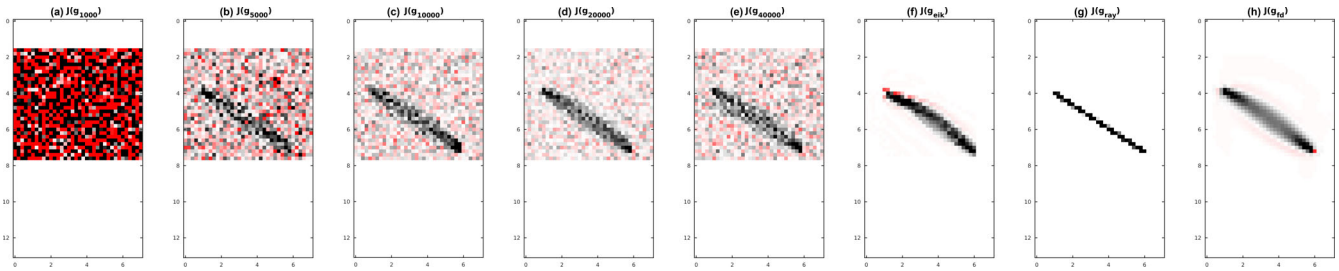


**Figure 6.** First-order Fréchet derivative with respect to the reference model for one specific source–receiver location. Black: positive, red: negative and white: zero sensitivity.

'exact' $g_{\text{fd}}$. This is the case even as the full-waveform modeling step of evaluating $g_{\text{fd}}$ is performed in parallel on four separate threads (using the implementation by Cordua *et al.* 2012). Further, it is about 15 times faster than using $g_{\text{eik}}$ (which is the most widely used methods for computing first arrival traveltimes) and about three times faster than using the simple linear $g_{\text{ray}}$ (see Table 1).

(5) The standard deviation of the modeling error becomes close to the standard deviation of the noise added to the reference data using $g_{20000}$ and $g_{40000}$ (0.13 ns versus 0.1ns).

Fig. 6 shows the sensitivity kernel, that is, the first-order Fréchet derivative, for the reference velocity model for one specific source–receiver location for all the considered approximate forward models as well as the reference forward model (Fig. 6h). Visible 'noise' is present for all neural network-based forward models (Figs 6a–e) decreasing as the size of the training data set increases. Therefore, it may be surprising that using, for example, $g_{20000}$ and $g_{40000}$ performs better than using $g_{\text{eik}}$. The 'noise' in the Fréchet derivatives when using the neural network-based forwards seems to be spatially uncorrelated and, therefore, the effect in the computed traveltime from the noise may tend to cancel out. Thus, while the forward model based on a neural network seems to predict traveltimes with relatively high accuracy, one should not use the forward model to, for example, estimate the first-order Frechet derivative and assign any physical meaning to these. One should use the network for the purpose it was trained, that is, in this case traveltime calculation.

These results are encouraging as they suggest a neural network-based forward model can be constructed that is both faster to evaluate, and results in less modeling error, than using any of the widely used approximations $g_{\text{eik}}$ and $g_{\text{ray}}$. However, the main point of interest is how much, and how efficient, information can be obtained from the posterior probability density using Monte Carlo sampling methods, based on the different forward models, and corresponding Gaussian models for describing the modeling error.

## 3 EFFICIENT MONTE SAMPLING OF THE POSTERIOR PROBABILITY DISTRIBUTION

The extended Metropolis algorithm is used to sample the posterior distribution, $\sigma(\mathbf{m})$, for the crosshole GPR inverse problem using the seven considered types of approximate forward models and the corresponding estimated Gaussian model for the modeling errors using the SIPPI Matlab toolbox (for details see e.g. Hansen *et al.* 2016). Each Metropolis sampler is run for $10^6$ iterations. The first $10^5$ iterations are discarded and considered as part of the burn-in phase of the extended Metropolis algorithm.

Fig. 7 shows eight independent realization of the posterior distribution $\sigma(\mathbf{m})$ using each of the seven considered forward model approximations. Fig. 8 shows the corresponding pointwise mean and standard deviation of all realizations from the posterior distribution.

Even when using the worst neural network-based forward approximation, namely the neural network based on a training set of size 1000 $g_{1000}$, provides results consistent with the reference model, in that no noise is apparently mapped into the posterior distribution as resolved features. In other words, the features that appear well resolved (that stand out in the mean model) are consistent with features in the reference model. This may be somewhat surprising considering the high degree of modeling error (Fig. 5a), and the very noisy Fréchet derivative (Fig. 6a).

As the accuracy of the used approximation increases, so does the resolution, which can be seen by the increase in details in the mean model (Figs 8a–h), along with a decrease in posterior variability (Fig. 7). This is also reflected in the pointwise standard deviation of the posterior distribution (Figs 8i–o), which decrease as the accuracy of the neural networks increase. Only the neural network based on training set of sample size 1000 provides a 1-D marginal posterior standard deviation lower than using $g_{\text{eik}}$ or $g_{\text{ray}}$. This suggests that more information about the 'exact' forward model $g_{\text{FD}}$ is provided by the neural network-based forward models (except $g_{1000}$) than by the ray-based approximations. This is of course only interesting if the apparent increase in resolution is due to actual data and not simply an effect of mapping, for example, modeling errors from the data space into the model space. As a synthetic case is studied, this can be analysed by comparing realizations from the posterior distribution to the reference model.

Table 1 (row 5) lists the average correlation coefficient between realizations of the posterior distribution and the reference model, revealing that $g_{20000}$ and $g_{40000}$ provide more information, than using either $g_{\text{eik}}$ or $g_{\text{ray}}$. A higher number indicate better performance. A similar property is reflected by the mean absolute error between the reference model and 100 realizations from the posterior distribution listed in Table 1 (row 6).

These results demonstrate that using a neural network-based forward as an approximation to the 'exact' forward, $f_{\text{fd}}$, allows inferring more information from a GPR crosshole traveltime inversion than using the more widely used forward models $g_{\text{eik}}$ and $g_{\text{ray}}$. In addition, it is important that this increase in resolution, compared to using $g_{\text{eik}}$ and $g_{\text{ray}}$, is consistent with features in the reference model. At the same time, the computational requirement to use any of the neural network-based forward models is about 15 times faster than using $g_{\text{eik}}$.

## 4 DISCUSSION

The results presented above demonstrate the potential of using regression-type networks, here a neural network, as a replacement for computational demanding forward models in

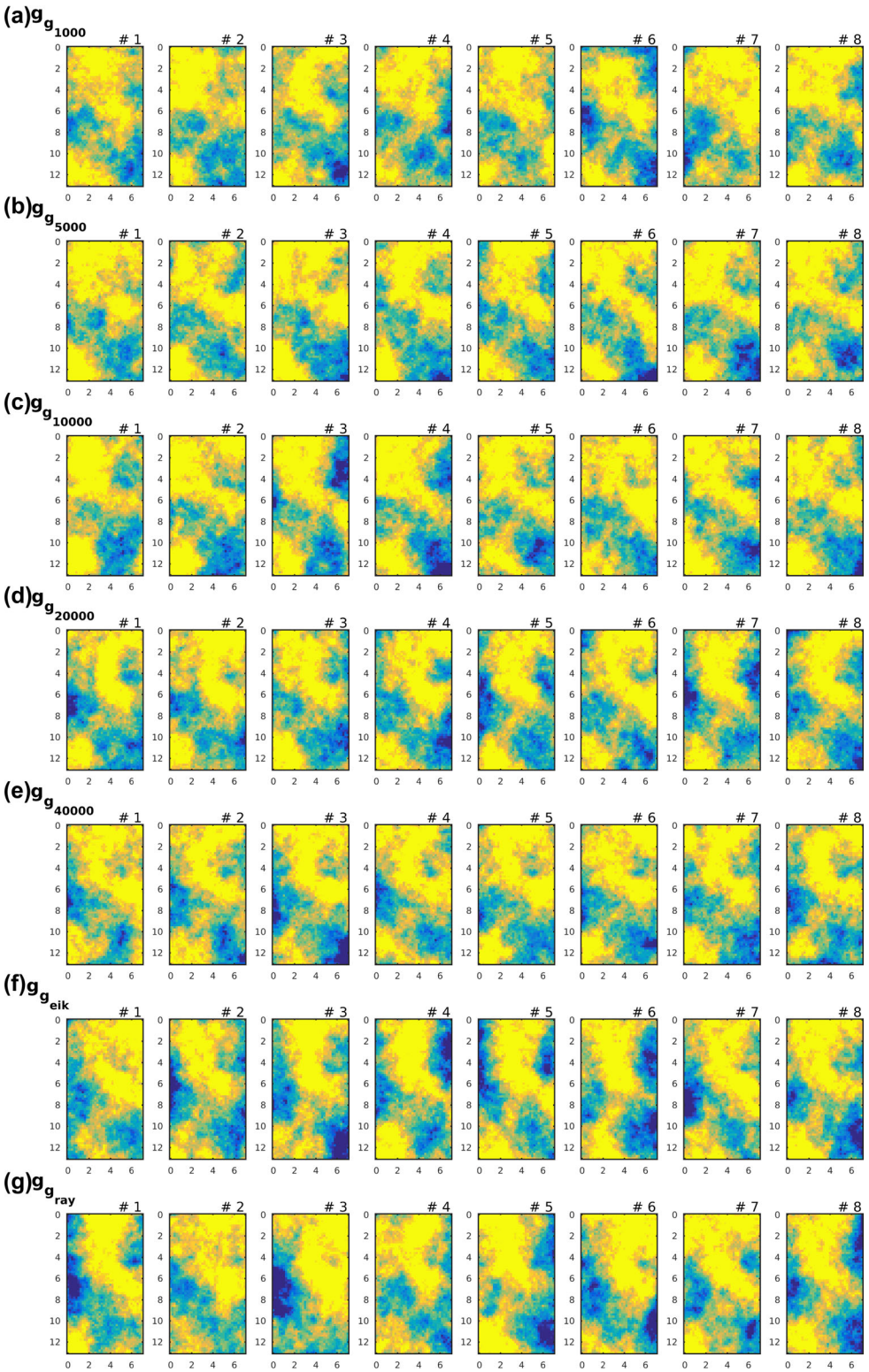


**Figure 7.** (a)–(g) Eight independent realizations from the posterior probability density $\sigma(\mathbf{m})$ for the seven considered forward models.

physics/geophysics. The proposed methodology is general, and can potentially be applied to a wide range of forwards for inverse problems, in which the posterior distribution can be sampled as follows

(i) Estimate an approximation $g_{\text{app}}$ to the full forward $g_{\text{ref}}$ using a regression network.

(1) Generate NT realizations of the prior model $\rho(\mathbf{m})$ as $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{\text{NT}}]$.

(2) Compute the forward response of the NT realizations of the prior model, as $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_{\text{NT}}]$ using a reference forward model, $g_{\text{ref}}$.

(3) Estimate an approximation to $g_{\text{ref}}$ as $g_{\text{app}}$ from the training data set $[\mathbf{D}, \mathbf{M}]$ using a regression network (for example, a neural network).

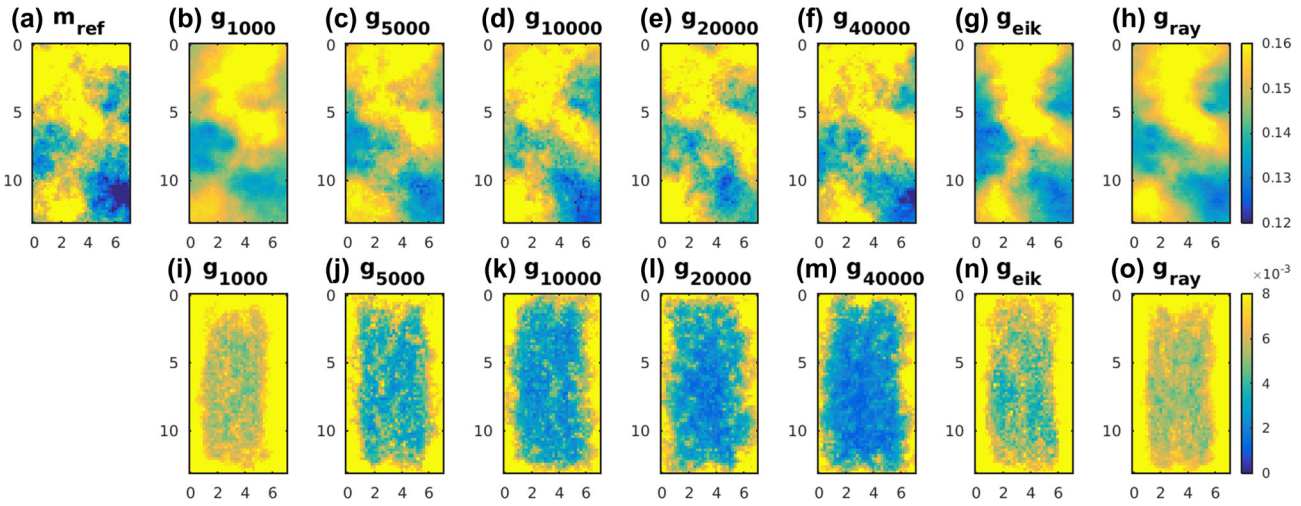


**Figure 8.** (a) Reference model, (b)–(h) mean of posterior realizations, (i)–(o) standard deviation of posterior realizations.

(ii) Estimate a Gaussian model of the modeling error $\theta(\mathbf{d}|\mathbf{m})$.

(1) Generate a large sample of the prior $\mathbf{M}^*$ (independent of $\mathbf{M}$ above).

(2) Compute the forward response for all models in $\mathbf{M}^*$, using both the correct, $g_{\text{ref}}$, and approximate, $g_{\text{app}}$, forward models, as $\mathbf{D}_{g_{\text{ref}}}$ and $\mathbf{D}_{g_{\text{app}}}$. A sample of the modeling error is then given by $\mathbf{D}_\theta = \mathbf{D}_{g_{\text{ref}}} - \mathbf{D}_{g_{\text{app}}}$.

(3) From $\mathbf{D}_\theta$ estimate a mean and covariance model describing the modeling error as a Gaussian probability density through $\theta(\mathbf{d}|\mathbf{m}) \sim \mathcal{N}(\mathbf{d}_t, \mathbf{C}_t)$

(iii) Accounting for the modeling error as a Gaussian uncertainty on data.

(iv) Use the extended Metropolis algorithm to sample from the posterior distribution.

This approach will potentially allow using sampling-based approaches to some inverse problems that has until now not been feasible due to computational demands.

The main benefits of using the neural network-based forward model is that it provides an alternative forward that is both faster and more accurate than traditional ray-based forward models. Another key advantage of using the neural network-based forward models is that they are specifically designed for whatever strategy is used to pick the first arrival. The strategy for picking the first arrival time need not to be chosen to match a specific choice of forward model, but can be chosen from what method for picking arrival time is most suited for the data that provides the most robust estimate. It does not matter if the traveltime is chosen to be the first break, the arrival of the maximum amplitude, or any other attribute, which will directly affect the validity of any of the ray-based approaches (Jacobsen *et al.* 2010). If only the same strategy is used to pick the arrival time, then the neural network will learn a forward model specifically for that picking strategy.

The main challenges using the methodology are: (1) *To generate a large training data set.* The full forward model needs to be evaluated for as many times as the size of the training data set. This may be time-consuming, but need only to be done once, and typically the Monte Carlo sampling algorithm will need many more iterations than the size of the training data set. (2) *Choosing a network design.* In the present case, a very simple neural networks design is chosen, and applied successfully. However, other regression network types and designs could be chosen for better performance, better suited for other types of forward models, and other types of prior models. (3) *Training the neural network.* It takes time to train the neural network. However, recently highly efficient methods for training neural networks on graphical processing units have been developed, which allow handling larger networks more efficiently.

In addition, the neural network obtained using the procedure above is applicable only to the specific recording geometry given in Fig. 1. If the recording geometry changes a new training data set needs to be computed, and a new neural network needs to be trained.

In the example considered here, the training data set was generated from a specific prior distribution. Thus, if the prior distribution changes, a training data set needs to be created and a new neural network needs to be trained. It would, however, be possible to generate a training data set, where models are generated from a much broader prior model, that is, a prior model that consist of models with more spatial variability than considered here. Such a training data set could be used as the base to learning a neural that can be used for many different types of prior models (i.e. Earth model variabilities) without the need for a specific training data set. This may be a subject for future research.

## 5 CONCLUSIONS

A general framework has been demonstrated that allow replacing a computationally complex forward model with a computationally efficient (fast to evaluate) forward model using a neural network. At the same time, the forward modeling error associated with the approximate neural network-based forward has been quantified. This has the potential to allow using more realistic forward models, on more realistic sized models, using otherwise computationally expensive Monte Carlo-based sampling methods to solve non-linear inverse problems.

The approach has been demonstrated on a GPR crosshole traveltime inverse problem. A realistic, but computationally heavy, forward model based on full-waveform modeling has been replaced by a neural network, which turns out to be more than three orders of magnitude faster to evaluate than the full 'exact' solution to compute the first arrival traveltime. It has been demonstrated that if the sample size of the training data set is large enough (NT > 5000), the forward model based on a neural network can be evaluated faster, with less error (in terms of the magnitude of the modeling error),

compared to using other widely used approximations to compute first arrival times (i.e. the eikonal solution and the linear straight ray assumption). Moreover, using the neural network based forward model leads to inversion results faster and with better resolution. As the modeling errors are accounted for in a probabilistic model, the use of approximate forward models does not lead to any (observed) bias in the final inversion results.

In the present case, the proposed methodology has allowed an otherwise hard sampling-based inverse problem to be replaced by a much simpler and easier sampling problem. The presented methodology is general, and has potential to change the scale and complexity of non-linear inverse problems that have to be solved through Monte Carlo sampling, using, for example, the extended Metropolis algorithm.

The results demonstrate that a neural network can be used to describe a (relatively) complex forward model ($g_{\mathrm{fd}}$), and that it can be both relatively accurate, and fast to evaluate. For inverse modeling, it provides a better resolution of the posterior probability density than two other commonly used approximate forward models, obtained with significantly less computational usage.

## ACKNOWLEDGEMENTS

We thank everyone at the Climate and Computational Geophysics group at the Niels Bohr Institute for a daily encouraging and appealing research environment. Thank you to Jacques Ernst and ETH Zurich, for making the finite-difference modeling code available, such that a parallel Matlab interface could be developed (https://sites.google.com/site/kscordua/downloads/tomographic-full-waveform-inversion). Matlab code to reproduce the results are available at https://github.com/cultpenguin/sippi/tree/master/examples/papers/hansen_and_cordua_2017_nn/.

## REFERENCES

Amari, S. & Cichocki, A., 1998. Adaptive blind signal processing-neural network approaches, *Proc. IEEE,* **86**(10), 2026–2048.

Beale, M.H., Hagan, M. & Demuth, H., 2016. *Matlab Neural Network Toolbox,* The Mathworks Inc.

Bishop, C.M., 1995. *Neural Networks for Pattern Recognition,* Oxford Univ. Press.

Bording, R.P., Gersztenkorn, A., Lines, L.R., Scales, J.A. & Treitel, S., 1987. Applications of seismic travel-time tomography, *Geophys. J. Int.,* **90**(2), 285–303.

Breiman, L., Friedman, J., Stone, C.J. & Olshen, R.A., 1984. *Classification and Regression Trees,* CRC Press.

Cordua, K.S., Nielsen, L., Looms, M.C., Hansen, T.M. & Binley, A., 2009. Quantifying the influence of static-like errors in least-squares-based inversion and sequential simulation of cross-borehole ground penetrating radar data, *J. appl. Geophys.,* **68**(1), 71–84.

Cordua, K.S., Hansen, T.M. & Mosegaard, K., 2012. Monte Carlo full waveform inversion of crosshole GPR data using multiple-point geostatistical a priori information, *Geophysics,* **77,** H19–H31.

Dafflon, B., Irving, J. & Barrash, W., 2011. Inversion of multiple intersecting high-resolution crosshole gpr profiles for hydrological characterization at the boise hydrogeophysical research site, *J. appl. Geophys.,* **73**(4), 305–314.

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

Ernst, J.R., Maurer, H., Green, A.G. & Holliger, K., 2007b. Full-waveform inversion of crosshole radar data based on 2-D finite-difference time-domain solutions of Maxwell's equations, *IEEE Trans. Geosci. Remote Sens.,* **45**(9), 2807–2828.

Gagunashvili, N., 2010. Machine learning approach to inverse problem and unfolding procedure, arXiv:1004.2006.

Giroux, B., Gloaguen, E. & Chouteau, M., 2007. bh_tomo-a Matlab borehole georadar 2d tomography package, *Comput. Geosci.,* **33**(1), 126–137.

Gloaguen, E., Giroux, B., Marcotte, D. & Dimitrakopoulos, R., 2007. Pseudo-full-waveform inversion of borehole GPR data using stochastic tomography, *Geophysics,* **72**(5), J43–J51.

Hansen, T.M., Cordua, K.C. & Mosegaard, K., 2012. Inverse problems with non-trivial priors - efficient solution through sequential Gibbs sampling, *Computational Geosciences,* **16**(3), 593–611.

Hansen, T., Cordua, K., Looms, M. & Mosegaard, K., 2013. SIPPI: a Matlab toolbox for sampling the solution to inverse problems with complex prior information. Part 2, Application to cross hole GPR tomography, *Computers & Geosciences,* **52,** 481–492.

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

Hansen, T.M., Cordua, K.S., Zunino, A. & Mosegaard, K., 2016. Probabilistic integration of geo-information, in *Integrated Imaging of the Earth: Theory and Applications,* Vol. 218, pp. 93–116, Wiley.

Hole, J., 1992. Nonlinear high-resolution three-dimensional seismic travel time tomography, *J. geophys. Res.,* **97**(B5), 6553–6562.

Holliger, K., Musil, M. & Maurer, H., 2001. Ray-based amplitude tomography for crosshole georadar data: a numerical assessment, *J. appl. Geophys.,* **47**(3), 285–298.

Hoole, S.R.H., 1993. Artificial neural networks in the solution of inverse electromagnetic field problems, *IEEE Trans. Magn.,* **29**(2), 1931–1934.

Jacobsen, B.H., Hansen, T.M. & Cordua, K.S., 2010. Understanding why first-arrival travel times do not obey banana-doughnut sensitivity, *Geophys. Res. Abstr*, Vol. 12, EGU2010-6480-1, EGU General Assembly, http://meetingorganizer.copernicus.org/EGU2010/EGU2010-6480-1.pdf.

Khan, A., Mosegaard, K. & Rasmussen, K.L., 2000. A new seismic velocity model for the Moon from a Monte Carlo inversion of the Apollo lunar seismic data, *Geophys. Res. Lett.,* **27**(11), 1591–1594.

Klotzsche, A., van der Kruk, J., Linde, N., Doetsch, J. & Vereecken, H., 2013. 3-D characterization of high-permeability zones in a gravel aquifer using 2-D crosshole GPR full-waveform inversion and waveguide detection, *Geophys. J. Int.,* **195**(2), 932–944.

Krasnopolsky, V.M. & Schiller, H., 2003. Some neural network applications in environmental sciences. Part I: Forward and inverse problems in geophysical remote measurements, *Neural Netw.,* **16**(3), 321–334.

Laloy, E., Linde, N. & Vrugt, J.A., 2012. Mass conservative three-dimensional water tracer distribution from Markov chain Monte Carlo inversion of time-lapse ground-penetrating radar data, *Water Resour. Res.,* **48,** W07510, doi:10.1029/2011WR011238.

Li, G., You, J. & Liu, X., 2015. Support vector machine (SVM) based prestack avo inversion and its applications, *J. appl. Geophys.,* **120,** 60–68.

Linde, N. & Vrugt, J.A., 2013. Distributed soil moisture from crosshole ground-penetrating radar travel times using stochastic inversion, *Vadose Zone J.,* **12**(1), doi:10.2136/vzj2012.0101.

Linde, N., Binley, A., Tryggvason, A., Pedersen, L.B. & Revil, A., 2006. Improved hydrogeophysical characterization using joint inversion of crosshole electrical resistance and ground-penetrating radar traveltime data, *Water Resour. Res.,* **42,** W12404, doi:10.1029/2006WR005131.

Looms, M., Hansen, T., Cordua, K., Nielsen, L., Jensen, K. & Binley, A., 2010. Geostatistical inference using crosshole ground-penetrating radar, *Geophysics,* **75**(6), J29–J41.

Looms, M.C., Binley, A., Jensen, K.H., Nielsen, L. & Hansen, T.M., 2008a. Identifying unsaturated hydraulic parameters using an integrated data fusion approach on cross-borehole geophysical data, *Vadose Zone J.,* **7**(1), 238–248.

Looms, M.C., Jensen, K.H., Binley, A. & Nielsen, L., 2008b. Monitoring unsaturated flow and transport using cross-borehole geophysical methods, *Vadose Zone J.,* **7**(1), 227–237.

Maiti, S., Erram, V.C., Gupta, G. & Tiwari, R.K., 2012. ANN based inversion of DC resistivity data for groundwater exploration in hard rock terrain of western maharashtra (India), *J. Hydrol.,* **464,** 294–308.

Malinverno, A., 2002. Parsimonious bayesian markov chain monte carlo inversion in a nonlinear geophysical problem, *Geophys. J. Int.,* **151**(3), 675–688.

Maurer, H. & Green, A.G., 1997. Potential coordinate mislocations in crosshole tomography: results from the grimsel test site, Switzerland, *Geophysics,* **62**(6), 1696–1709.

McMechan, G.A., 1983. Seismic tomography in boreholes, *Geophys. J. Int.,* **74**(2), 601–612.

Meles, G.A., Van der Kruk, J., Greenhalgh, S.A., Ernst, J.R., Maurer, H. & Green, A.G., 2010. A new vector waveform inversion algorithm for simultaneous updating of conductivity and permittivity parameters from combination crosshole/borehole-to-surface GPR data, *IEEE Trans. Geosci. Remote Sens.,* **48**(9), 3391–3407.

Minsley, B.J., 2011. A trans-dimensional bayesian markov chain monte carlo algorithm for model assessment using frequency-domain electromagnetic data, *Geophys. J. Int.,* **187**(1), 252–272.

Molyneux, J. & Schmitt, D., 1999. First break timing: arrival onset times by direct correlation, *Geophysics,* **64**(5), 1492–1501.

Mosegaard, K. & Tarantola, A., 1995. Monte carlo sampling of solutions to inverse problems, *J. geophys. Res,* **100**(B7), 12 431–12 447.

Mosegaard, K. & Tarantola, A., 2002. Probabilistic approach to inverse problems, in *International Handbook of Earthquake and Engineering Seismology,* Vol. 81A, chap. 16, pp. 237–265, eds Lee, W., Kanamori, H., Jennings, P. & Kisslinger, C., Academic Press.

Mosegaard, K., Singh, S., Snyder, D. & Wagner, H., 1997. Monte carlo analysis of seismic reflections from Moho and the W reflector, *J. geophys. Res.,* **102**(B2), 2969–2981.

Nielsen, L., Looms, M.C., Hansen, T.M., Cordua, K.S. & Stemmerik, L., 2010. Estimation of chalk heterogeneity from stochastic modelling conditioned by crosshole GPR travel times and log data, in *Advances in Near-surface Seismology and Ground-penetrating Radar: SEG Geophysical Development Series,* Vol. 15, pp. 379–398, Society of Exploration Geophysicists, American Geophysical Union, Environmental and Engineering Geophysical Society.

Poulton, M.M., 2001. *Computational Neural Networks for Geophysical Data Processing,* Vol. 30, Elsevier.

Prato, M. & Zanni, L., 2008. Inverse problems in machine learning: an application to brain activity interpretation, *J. Phys.: Conf. Ser.*, **135**(1), 012085.

Scales, J.A., 1987. Tomographic inversion via the conjugate gradient method, *Geophysics,* **52**(2), 179–185.

Schuster, G.T. & Quintus-Bosz, A., 1993. Wavepath eikonal traveltime inversion: theory, *Geophysics,* **58**(9), 1314–1323.

Singh, U.K., Tiwari, R. & Singh, S., 2013. Neural network modeling and prediction of resistivity structures using VES Schlumberger data over a geothermal area, *Comput. Geosci.,* **52,** 246–257.

Smola, A.J. & Schölkopf, B., 2004. A tutorial on support vector regression, *Stat. Comput.,* **14**(3), 199–222.

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

Tarantola, A. & Valette, B., 1982a. Generalized nonlinear inverse problems solved using the least squares criterion, *Rev. Geophys. Space Phys,* **20**(2), 219–232.

Tarantola, A. & Valette, B., 1982b. Inverse problems = quest for information, *J. geophys. Res.,* **50**(3), 150–170.

Vidale, J.E., 1990. Finite-difference calculation of traveltimes in three dimensions, *Geophysics,* **55**(5), 521–526.

Williams, C.K. & Rasmussen, C.E., 1995. Gaussian processes for regression, in *Advances in Neural Information Processing Systems 8,* pp. 514–520, eds Touretzky, D.S., Mozer, M.C. & Hasselmo, M.E., https://papers.nips.cc/book/advances-in-neural-information-processing-systems-8-1995.

Zelt, C. & Smith, R., 1992. Seismic traveltime inversion for 2-D crustal velocity structure, *Geophys. J. Int.,* **108**(1), 16–34.

Zunino, A., Mosegaard, K., Lange, K., Melnikova, Y. & Mejer Hansen, T., 2014. Monte Carlo reservoir analysis combining seismic reflection data and informed priors, *Geophysics,* **80**(1), R31–R41.