



UNIL | Université de Lausanne

Unicentre

CH-1015 Lausanne

<http://serval.unil.ch>

Year : 2023

Efficient Bayesian inversion for geophysical applications: Leveraging deep learning and geostatistical methods

Levy Shiran

Levy Shiran, 2023, Efficient Bayesian inversion for geophysical applications: Leveraging deep learning and geostatistical methods

Originally published at : Thesis, University of Lausanne

Posted at the University of Lausanne Open Archive <http://serval.unil.ch>

Document URN : urn:nbn:ch:serval-BIB_08FBA69E09865

Droits d'auteur

L'Université de Lausanne attire expressément l'attention des utilisateurs sur le fait que tous les documents publiés dans l'Archive SERVAL sont protégés par le droit d'auteur, conformément à la loi fédérale sur le droit d'auteur et les droits voisins (LDA). A ce titre, il est indispensable d'obtenir le consentement préalable de l'auteur et/ou de l'éditeur avant toute utilisation d'une oeuvre ou d'une partie d'une oeuvre ne relevant pas d'une utilisation à des fins personnelles au sens de la LDA (art. 19, al. 1 lettre a). A défaut, tout contrevenant s'expose aux sanctions prévues par cette loi. Nous déclinons toute responsabilité en la matière.

Copyright

The University of Lausanne expressly draws the attention of users to the fact that all documents published in the SERVAL Archive are protected by copyright in accordance with federal law on copyright and similar rights (LDA). Accordingly it is indispensable to obtain prior consent from the author and/or publisher before any use of a work or part of a work for purposes other than personal use within the meaning of LDA (art. 19, para. 1 letter a). Failure to do so will expose offenders to the sanctions laid down by this law. We accept no liability in this respect.



Faculté des géosciences et de l'environnement
Institut des sciences de la Terre

Efficient Bayesian inversion for geophysical applications: Leveraging deep learning and geostatistical methods

Thèse de doctorat

Présentée à la
Faculté des géosciences et de l'environnement
Institut des sciences de la Terre
de l'Université de Lausanne

pour l'obtention du grade de

Docteur en sciences de la Terre

par

Shiran Levy

Diplôme (M.Sc.) en Géophysique Appliquée
IDEA League Joint Master Program
ETH Zürich - TU Delft - RWTH Aachen

Jury

Prof. Dr. Marie-Elodie Perga, Présidente du jury
Prof. Dr. Niklas Linde, Directeur de thèse
Prof. Dr. Grégoire Mariethoz, Expert interne
Prof. Dr. Thomas Mejer Hansen, Expert externe

Lausanne, 2023



UNIL | Université de Lausanne
Faculté des géosciences et de l'environnement
bâtiment Géopolis bureau 4631

IMPRIMATUR

Vu le rapport présenté par le jury d'examen, composé de

| | |
|------------------------------------|---------------------------------------|
| Présidente de la séance publique : | Mme la Professeure Marie-Elodie Perga |
| Présidente du colloque : | Mme la Professeure Marie-Elodie Perga |
| Directeur de thèse : | M. le Professeur Niklas Linde |
| Expert interne : | M. le Professeur Grégoire Mariéthoz |
| Expert externe : | M. le Professeur Thomas Mejer Hansen |

Le Doyen de la Faculté des géosciences et de l'environnement autorise l'impression de la thèse de

Madame Shiran LEVY

*Titulaire d'un
Master in Applied Geophysics
de l'ETH à Zürich*

intitulée

**EFFICIENT BAYESIAN INVERSION FOR GEOPHYSICAL APPLICATIONS:
LEVERAGING DEEP LEARNING AND GEOSTATISTICAL METHODS**

Lausanne, le 26 octobre 2023

Pour le Doyen de la Faculté des géosciences et de
l'environnement

Professeure Marie-Elodie Perga

Acknowledgements

Research requires us to confront our ignorance and acknowledge what we do not know. The quest for knowledge can be challenging, but it can also be rewarding, especially when pursued collaboratively, making it an enjoyable experience. Throughout the PhD journey, I came to realise the profound significance of having a supportive network of individuals around you. Thus, I would like to express my sincere gratitude to those who have contributed to this PhD work and have been part of my journey for the last four years.

Foremost, I would like to thank my supervisor, **Niklas Linde** for his constant support, patience, valuable guidance, and insightful feedback throughout the research process. I am truly grateful for having him as my PhD supervisor, his expertise have played a crucial role in shaping the direction of this thesis, constantly providing inspiration and encouragement.

I am deeply grateful to my collaborators for their valuable suggestions and positive feedback which greatly enhanced the quality of this work. I would like to give a special thanks to **Eric Laloy** for the interesting and pleasant discussions, his suggestion of using Pyro was an excellent advice and his work is a source of inspiration for me. I would like to thank **Jürg Hunziker** for his guidance at the beginning of the PhD providing me with a smooth transition into the PhD. Our conversations about scientific and non-scientific topics for the last four years have been truly enriching. I would also like to thank **Grégoire Mariéthoz** and **James Irving**, their thought-provoking suggestions and discussions, along with their positive and kind attitude, boost my confidence in my work.

Additionally, I am thankful to **Giovanni Meles** for his kind assistance and willingness to address any questions I had, to **Flavio Calvo** for helping me improve my code and patiently answering my many questions, it was always great to learn new things during the process and for **David Ginsbourger** and his group from the university of Bern, for the pleasant meetings and fruitful and friendly discussions.

My experience would not have been as enjoyable without the presence and support of my friends and colleagues from Géopolis. They are truly incredible individuals, and I want to express my gratitude to them for the conversations, lunches, and coffee breaks, which I looked forward to each day. A special thanks to **Lea Friedli** and **Macarena Amaya** for their friendship, dinners, meetings, fun conference days and support throughout the PhD, I was lucky to have shared the ups and downs of the PhD path together with you two.

I would like also to thank the dissertation committee members (**Niklas Linde**, **Grégoire Mariéthoz** and **Thomas Mejer Hansen**) for their interest in my work and constructive discussion and for the Swiss National Science Foundation (SNSF) for funding this work (grant number

184574).

Finally, this PhD would not have been possible without my husband and my family, whom I deeply love and who were my main source of support and comfort in good and challenging times.

Lausanne, August 2023

Shiran

Contents

| | |
|--|-------------|
| Acknowledgements | i |
| List of figures | v |
| List of tables | vii |
| Abstract | ix |
| Résumé | xi |
| Résumé pour un public général | xiii |
| 1 Introduction | 1 |
| 1.1 Machine learning in the geosciences | 2 |
| 1.2 The forward problem | 3 |
| 1.3 The deterministic inverse problem | 6 |
| 1.4 Bayesian inference | 8 |
| 1.4.1 Markov chain Monte Carlo | 9 |
| 1.4.2 Variational inference | 12 |
| 1.5 Conceptual models and geological realism | 13 |
| 1.5.1 Geostatistical models | 13 |
| 1.5.2 Generative models | 15 |
| 1.6 Objectives and outline | 17 |
| 2 Using deep generative neural networks to account for model errors in Markov chain Monte Carlo inversion | 21 |
| 2.1 Introduction | 22 |
| 2.2 Methods | 25 |
| 2.2.1 Database preparation | 25 |
| 2.2.2 Generative adversarial networks | 27 |
| 2.2.3 SGAN architecture and training | 28 |
| 2.2.4 Bayesian inference of latent parameters | 30 |
| 2.3 Results | 34 |
| 2.3.1 Quality assessment of generative models | 34 |
| 2.3.2 Inversion results | 38 |
| 2.4 Discussion | 47 |
| 2.5 Conclusions | 50 |
| 2.6 Appendix | 50 |
| 2.6.1 Details on SGAN Architecture and training | 50 |
| 2.6.2 Quality measure calculation | 53 |

| | | |
|----------|---|------------|
| 3 | Variational Bayesian inference with complex geostatistical priors using inverse autoregressive flows | 55 |
| 3.1 | Introduction | 56 |
| 3.2 | Methods | 59 |
| 3.2.1 | Inverse autoregressive flows | 60 |
| 3.2.2 | Variational Bayesian inference | 60 |
| 3.2.3 | Deep generative models | 62 |
| 3.2.4 | Crosshole traveltime tomography | 64 |
| 3.2.5 | Inversion in the latent space of a deep generative model with neural-transport | 65 |
| 3.2.6 | Performance assessment | 67 |
| 3.3 | Inversion results | 69 |
| 3.4 | Discussion | 74 |
| 3.5 | Conclusions | 78 |
| 3.6 | Appendix | 79 |
| 3.6.1 | IAF design | 79 |
| 3.6.2 | Hyperparameter calibration | 80 |
| 3.6.3 | Supplementary results | 81 |
| 4 | Conditioning of multiple-point statistics simulations to indirect geophysical data | 83 |
| 4.1 | Introduction | 84 |
| 4.2 | Methods | 86 |
| 4.2.1 | Bayesian formulation for conditional sequential simulation | 87 |
| 4.2.2 | QuickSampling algorithm | 91 |
| 4.2.3 | Forward response | 93 |
| 4.3 | Comparative approach and quality assessment criteria | 94 |
| 4.3.1 | Sequential Gibbs sampling | 94 |
| 4.3.2 | Performance metrics | 95 |
| 4.4 | Results | 96 |
| 4.4.1 | Linear physics | 97 |
| 4.4.2 | Non-linear physics | 102 |
| 4.5 | Discussion | 105 |
| 4.6 | Conclusions | 107 |
| 4.7 | Appendix | 107 |
| 4.7.1 | Analytical posterior PDF for a multi-Gaussian field | 107 |
| 4.7.2 | Choice of QS parameters | 108 |
| 4.7.3 | Sensitivity matrix: binary channelised subsurface model | 109 |
| 5 | Conclusions and outlook | 111 |
| 5.1 | Conclusions | 111 |
| 5.2 | Limitations and outlook | 113 |
| | Bibliography | 134 |
| | Curriculum Vitae | 135 |

List of Figures

| | | |
|------|---|----|
| 1.1 | Illustration of the objective function for linear and nonlinear least-square problems | 7 |
| 1.2 | Illustration of rejection sampling, MCMC and Variational inference | 10 |
| 1.3 | Illustration of multiple-point statistics and deep generative models | 14 |
| 2.1 | Illustration of our SGAN architecture with five layers when applied to represent model errors | 29 |
| 2.2 | SGAN-ME workflow | 33 |
| 2.3 | Statistics of subsurface model-parameter realisations mean, variance and directional semivariograms | 35 |
| 2.4 | Examples of model-error realisations | 36 |
| 2.5 | Pixel-wise mean and variance of training images and SGAN generated model error realisations | 37 |
| 2.6 | Closest SGAN realisations obtained from pixel-to-pixel inversion | 38 |
| 2.7 | Inversion results for reference Models 1 and 2 for Test Case 1 ($\boldsymbol{\eta}^{eikonal-SR}$) . . . | 41 |
| 2.8 | Inferred model errors for Test Case 1 representing the discrepancy between the eikonal and straight-ray solvers | 42 |
| 2.9 | Data fit plots for the compared inversion approaches and test cases | 44 |
| 2.10 | RMSE $_{\Phi}$ and SSIM distributions of posterior samples for the different inversion approaches and test cases | 45 |
| 2.11 | Inversion results for reference models 1 and 2 for Test Case 2 ($\boldsymbol{\eta}^{FDTD-SR}$) | 47 |
| 2.12 | Inferred model errors for Test Case 2 representing the discrepancy between the FDTD and straight-ray solvers | 48 |
| 2.13 | SGAN architecture | 51 |
| 2.14 | Effective receptive field of original and interpolated training image | 53 |
| 3.1 | Illustration of one training iteration of neural-transport combined with deep generative models | 67 |
| 3.2 | Reference models used for inversion | 69 |
| 3.3 | Inferred posterior distributions on the latent space of the SGAN for different reference models | 71 |
| 3.4 | Inferred posterior distributions on the latent space of the VAE for different reference models | 72 |
| 3.5 | Estimation of the variational PDF describing the marginal posterior of the 1 st , 10 th and 18 th latent parameters of the mv ₅ model at various training iterations. | 74 |
| 3.6 | Prior and approximate marginal posteriors on the latent space of the SGAN obtained with neural-transport and MCMC | 75 |
| 3.7 | Prior and approximate marginal posteriors on the latent space of the VAE obtained with neural-transport and MCMC | 76 |

| | | |
|------|---|-----|
| 3.8 | Schematic drawing of the IAF architecture. | 79 |
| 3.9 | Average $RMSE_d$ during inference as a function of the learning rate | 80 |
| 3.10 | Average $RMSE_d$ during inference as a function of the number of iterations and forward simulations | 81 |
| 3.11 | Mean and standard deviation of posterior models obtained from neural transport using ten particles and the VAE | 82 |
| 4.1 | Schematic illustration of one IDCS simulation step for a binary model. | 90 |
| 4.2 | Training images for the various tested models | 97 |
| 4.3 | IDCS results for a random multivariate Gaussian field and a linear forward solver | 98 |
| 4.4 | IDCS results for the isotropic field with connected high-conductivity structures and a linear forward solver | 100 |
| 4.5 | IDCS results for the binary channelised field and a linear forward solver | 101 |
| 4.6 | IDCS results given a non-linear forward solver | 102 |
| 4.7 | Data WRMSE curves during the IDCS run given a non-linear forward solver | 103 |
| 4.8 | IDCS results for the binary channelised subsurface model a non-linear forward solver | 104 |
| 4.9 | Data WRMSE curves during the IDCS run given a binary channelised subsurface model and a non-linear forward solver | 104 |
| 4.10 | True sensitivity associated with the binary channels subsurface model | 110 |

List of Tables

| | | |
|-----|---|-----|
| 2.1 | Inversion convergence summary | 39 |
| 2.2 | Inversion results for Test Case 1 | 42 |
| 2.3 | Inversion results in terms of model-error estimation for the two considered test cases | 43 |
| 2.4 | Inversion results for Test Case 2 | 48 |
| 2.5 | SGAN hyper-parameters | 52 |
| 3.1 | Summary of the results obtained from inversion with neural-transport for various reference models and using either the SGAN or the VAE as a DGM | 73 |
| 3.2 | Statistical summary of the posterior PDF in the latent space | 74 |
| 4.1 | True and estimated training image statistics | 96 |
| 4.2 | Summary of IDCS results for linear and non-linear physical responses | 99 |
| 4.3 | Average model SSIM and data WRMSE given different QS parameters values | 109 |
| 4.4 | Average model SSIM and data WRMSE given a weighting kernel | 109 |

Abstract

Inverse modelling is a core element in geophysics and is used in other geoscientific fields as well as in medical imaging, astrophysics and computer vision. In geophysics, the inverse problem aims at estimating the model parameters, describing a property field in the subsurface, given measured indirect data that are contaminated with some level of noise. Inverse modelling is generally split into deterministic and probabilistic methods. Deterministic methods try to minimise an objective function, describing the misfit between observed and simulated data and regularization constraints used to stabilise the solution, by following the gradient towards the global minimum. While these methods are efficient, they are sensitive to the choice of initial model and for complex, nonlinear problems, they might provide sub-optimal solutions and limited uncertainty quantification. Probabilistic methods rely on sampling the solution space and describe the solution as a random variable. They are more robust than deterministic methods and can handle inverse problems of varying complexity, but are generally computationally-intensive due to repeated calculations of the forward response. This thesis proposes three approaches that aim to improve the efficiency of probabilistic inversion methods by using (1) compact parameterization, (2) cheap surrogate models, (3) gradient-based optimisation and (4) parallel computation. These components reduce either the computational burden imposed by the forward solver, the overall number of forward response computations or distribute the computation in a way that maximise performance. The first approach leverages the first two factors. It considers a computationally cheap, but simplified surrogate model as the forward solver and corrects the simulated data with a modelling error generated by a generative adversarial network. The modelling error and the subsurface model are both encoded in the low-dimensional latent spaces of generative adversarial networks, reducing the number of parameters needed to be inferred. The second approach, leverages the compact parameterization of generative adversarial networks and variational autoencoders as well as the efficient optimisation offered by inverse autoregressive flows and variational Bayesian inference. The flows in the form of a neural network are used to parameterize the posterior distribution on the latent space of the deep generative models. These parameters are trained through variational Bayesian inference and, at the end of the training, the posterior can be effectively estimated and sampled from. The third approach leverages parallel computation of efficient multiple-point statistics simulations, which are conditioned on measured indirect data given a prior defined by a training image. These three approaches are tested using synthetic travel-time tomography data from crosshole ground-penetrating radar experiments. All three approaches demonstrate improvements in computation time compared to widely used MCMC methods while providing comparable uncertainty quantification. The thesis demonstrates the potential of incorporating techniques from both machine learning and geostatistics to perform geophysical inversions. This fusion of approaches opens new avenues for tackling complex inverse problems.

Key words: Bayesian inversion, machine learning, deep generative models, variational inference, multiple-point statistics, ground-penetrating radar, geophysics.

Résumé

La modélisation inverse est un élément essentiel de la géophysique et est utilisée dans d'autres domaines géoscientifiques ainsi que dans l'imagerie médicale, l'astrophysique et la vision par ordinateur. En géophysique, le problème inverse vise à estimer les paramètres du modèle, décrivant un champ de propriétés dans le sous-sol, à partir de données indirectes mesurées qui sont contaminées par un certain niveau de bruit. La modélisation inverse est généralement divisée en méthodes déterministes et probabilistes. Les méthodes déterministes tentent de minimiser une fonction objective, décrivant l'inadéquation entre les données observées et simulées et les contraintes de régularisation utilisées pour stabiliser la solution, en suivant le gradient vers le minimum global. Bien que ces méthodes soient efficaces, elles sont sensibles au choix du modèle initial et, pour les problèmes complexes et non linéaires, elles peuvent fournir des solutions sous-optimales et une quantification limitée de l'incertitude. Les méthodes probabilistes reposent sur l'échantillonnage de l'espace de solution et décrivent la solution comme une variable aléatoire. Elles sont plus robustes que les méthodes déterministes et peuvent traiter des problèmes inverses de complexité variable, mais sont généralement gourmandes en ressources informatiques en raison des calculs répétés de la réponse directe. Cette thèse propose trois approches visant à améliorer l'efficacité des méthodes d'inversion probabiliste en utilisant (1) une paramétrisation compacte, (2) des modèles de substitution bon marché, (3) une optimisation basée sur le gradient et (4) le calcul parallèle. Ces composants réduisent soit la charge de calcul imposée par le solveur numérique, soit le nombre total de calculs de la réponse physique, soit distribuent le calcul de manière à maximiser les performances. La première approche exploite les deux premiers facteurs. Elle considère un modèle de substitution simplifié mais peu coûteux en termes de calcul comme le solveur direct et corrige les données simulées avec une erreur de modélisation générée par un réseaux antagonistes génératifs. L'erreur de modélisation et le modèle de subsurface sont tous deux encodés dans les espaces latents de faible dimension des réseaux antagonistes génératifs, ce qui réduit le nombre de paramètres à déduire. La seconde approche tire parti de la paramétrisation compacte des réseaux adversaires génératifs et des autoencodeurs variationnels, ainsi que de l'optimisation efficace offerte par les flux autorégressifs inverses et l'inférence bayésienne variationnelle. Les flux sous la forme d'un réseau neuronal sont utilisés pour paramétrer la distribution postérieure sur l'espace latent des modèles génératifs profonds. Ces paramètres sont entraînés par inférence bayésienne variationnelle et, à la fin de l'entraînement, la distribution postérieure peut être efficacement estimée et échantillonnée. La troisième approche exploite le calcul parallèle de simulations statistiques efficaces à points multiples, qui sont conditionnées par des données indirectes mesurées, compte tenu d'un a priori défini par une image d'entraînement. Ces trois méthodes sont testées à l'aide de données synthétiques de tomographie de temps de parcours provenant d'expériences de radar à pénétration de sol. Les trois approches démontrent une amélioration du temps de calcul par rapport aux méthodes MCMC largement utilisées, tout en fournissant une quantification comparable de l'incertitude. La thèse démontre le poten-

tiel de l'incorporation de techniques d'apprentissage automatique et de géostatistique pour effectuer des inversions géophysiques. Cette fusion d'approches ouvre de nouvelles voies pour résoudre des problèmes inverses complexes.

Mots clés : Inversion bayésienne, apprentissage automatique, modèles génératifs profonds, inférence variationnelle, statistiques à points multiples, radar à pénétration de sol, géophysique.

Résumé pour un public général

La modélisation inverse est un élément central de la géophysique et est utilisée pour trouver un modèle de subsurface ou un ensemble de modèles d'une propriété physique en accord avec les données mesurées et d'autres contraintes. La réponse physique reliant le champ de propriétés du sous-sol aux données mesurées par un instrument peut être approximée par un modèle mathématique ou numérique (calculable). Les modèles numériques nécessitent la discrétisation du modèle de subsurface, c'est à dire la paramétrisation du modèle, en divisant l'espace ou le temps en intervalles discrets. En simulant la réponse physique, également appelée réponse directe, pour un ensemble spécifique de paramètres du modèle (une réalisation souterraine) et en comparant les données obtenues avec les données observées lors d'une expérience, nous pouvons quantifier dans quelle mesure les paramètres du modèle proposé expliquent les données mesurées. En répétant ce processus avec différentes réalisations de la subsurface, nous pouvons explorer lesquelles d'entre elles sont susceptibles de bien décrire la distribution des propriétés physiques. Les méthodes d'inversion se divisent en méthodes déterministes et probabilistes. Nous considérons l'inversion probabiliste qui vise à trouver une distribution de solutions en accord avec les données et les connaissances antérieures. Les méthodes probabilistes sont très générales, mais elles sont souvent longues à calculer, en raison du calcul répété du modèle numérique. Dans cette thèse, nous présentons des approches de modélisation inverse qui améliorent l'efficacité des méthodes probabilistes traditionnelles et nous introduisons de nouvelles approches efficaces. La première approche décrit un moyen d'utiliser des solveurs à peu coûteux représentant une version plus simple d'une réponse physique complexe, tout en tenant compte des erreurs résultant de cette simplification. Pour ce faire, nous apprenons une paramétrisation de ces erreurs à l'aide de modèles génératifs profonds, une technique d'apprentissage profond qui fait partie du domaine de l'intelligence artificielle, et nous l'utilisons pour générer des erreurs afin de corriger les données simulées au cours du processus d'inversion. La seconde approche est basée sur la paramétrisation compacte de la subsurface par des modèles génératifs profonds et une autre technique d'apprentissage profond qui approxime efficacement le postérieur en utilisant l'optimisation basée sur le gradient et un modèle de flux transformant la distribution initiale en celle d'intérêt. La dernière approche limite les simulations géostatistiques séquentielles à la génération de réalisations de la subsurface qui correspondent étroitement aux données mesurées. Ces simulations construisent progressivement un modèle de réalisation de la subsurface, dans lequel les paramètres du modèle sont conditionnés à la fois par une image affichant le modèle géologique souhaité et par les données mesurées. Comparées aux méthodes d'inversion probabiliste traditionnelles, les trois approches ont démontré une réduction significative du temps de calcul tout en obtenant des résultats d'une exactitude similaire. Les approches développées dans la thèse créent de nouvelles opportunités et démontrent comment l'intégration des méthodes développées dans l'apprentissage automatique et les statistiques peut améliorer la performance des méthodes inverses géophysiques.

Chapter 1

Introduction

In many scientific fields and particularly in geophysics, it is not possible to directly observe the physical system or objects under investigation. Consequently, researchers often rely on indirect observations to gain insights about unknown system properties and state variables through inverse modelling (or simply inversion). While the forward problem aims to predict measurement outcomes based on a specific experimental design and an informed model, the inverse problem solves the opposite problem, namely, given observed data and a forward model it estimates the model parameters (*Tarantola, 2005*). Inversion plays a crucial role in various geoscientific fields including environmental and hydrogeological studies (*Hubbard and Rubin, 2000; Bagtzoglou and Atmadja, 2005; Milledge et al., 2012; Yongkai et al., 2022*), estimation of rock properties and reservoir characterization (*Wilt and Alumbaugh, 2003; Bosch et al., 2010*) and mineral (*Oldenburg and Pratt, 2007; Lelièvre et al., 2012*) and oil and gas exploration (*Virieux and Operto, 2009*). It serves as a valuable tool for decision making, monitoring and designing procedures for addressing local and global challenges affecting humans and ecological systems (*Lazaratos and Marion, 1997; Wegener and Amin, 2019; Maasackers et al., 2021; Scheidt et al., 2018; Gallet et al., 2022; Hu et al., 2023*).

The formulation and solution of an inverse problem can be complex and varies depending on the problem at hand (*Menke, 2018*). Most geophysical inversion problems are ill-posed as their solutions are either non-unique (finite data over continuous function) or unstable. To obtain geologically reasonable and stable solutions, geophysicists apply different types of regularisation constraints involving a priori geological or geophysical knowledge (*Zhdanov, 2015*). Generally speaking, inversion techniques are split into either deterministic or probabilistic. While deterministic inversion assumes that there is one true model to recover and uncertainty quantification focuses on errors in its estimation due to data errors or possibly forward modelling errors, probabilistic inversions describe the properties of interest in terms of random fields, implying that information about them is inherently uncertain. Deterministic inversions are generally quick and for linear problems yield a global solution. Conversely, nonlinear problems are often complex to address and might exhibit multiple local "best" solutions (local minima), making it challenging to reach the global solution. In such cases, linearization of the forward operator and an iterative updating scheme is needed. These type of schemes are sensitive to the level of nonlinearity of the objective function to be minimised and the choice of initial model, potentially leading to convergence to a local minimum (*Tarantola and Valette, 1982a; Tarantola, 2005*).

To circumvent the limitations of deterministic methods, global sampling techniques, such as Monte Carlo methods, are often employed to explore the model space and search for solutions in terms of distributions for nonlinear inversion problems. These methods explore a stationary probability distribution by repeated sampling and offer full uncertainty quantification and a higher chance of finding a global solution, yet they are computationally expensive. Rather than randomly sampling the model space as in pure Monte Carlo methods (*Kalos and Whitlock, 2009*), various algorithms use proposal distributions and acceptance rules to speed up the convergence to the stationary distribution. Such algorithms provide significant advantages, particularly in high-dimensional problems where a substantial portion of the solution space is characterised by low probabilities (curse of dimensionality; *Tarantola, 2005*). Nonetheless, these methods usually require hundreds of thousands of samples and may, for computationally-expensive forward solvers take days and in some cases months to execute (*Hunziker et al., 2019; Solonen et al., 2012*).

The limitations of deterministic inversion methods in terms of low robustness and limited possibilities for uncertainty quantification, as well as the computationally-intensive nature of standard probabilistic inversion methods, have motivated research to either enhance the efficiency of existing techniques or developing new efficient methodologies that provide reliable approximations and uncertainty quantification. Recent advancements in the fields of machine learning and geostatistics have created significant opportunities to improve geophysical inverse modelling, offering more accurate and efficient solutions to a range of increasingly-complex inverse problems.

1.1 Machine learning in the geosciences

Machine learning (ML) methods have recently garnered tremendous interest within the geoscientific community (*Bergen et al., 2019*) and geophysics in particular (*Yu and Ma, 2021*). Machine-learning techniques provide data-driven (and mostly black box) models that can handle and synthesise large and complex datasets quickly and efficiently, for tasks that would be overwhelming for humans. Machine-learning is split into supervised and unsupervised learning. In supervised learning, the data are organised in labels and the model is trained on these data labels. Examples for supervised learning methods are linear regression, logistic regression, decision trees, support vector machines (SVM), and neural networks (e.g. classification). Unsupervised learning on the other hand deals with unlabelled data and is trained based on relationships or patterns observed in the data. Unsupervised tasks such as clustering, dimensionality reduction and data generation are typically carried out using various clustering techniques and neural networks.

Machine-learning techniques are being applied and adapted to various tasks in the geosciences, ranging from pattern and object detection or segmentation, parameter estimation, prediction, surrogate modelling and dimensionality reduction (*Laloy et al., 2017; Karpatne et al., 2018; Mosavi et al., 2018; Chevotarese et al., 2018; Jin et al., 2020; Hu et al., 2020; Joshi et al., 2021; Guo et al., 2021*). Furthermore, these techniques have demonstrated their effectiveness in solving geophysical inverse problems both as a supplement to traditional methods and as a stand alone approach (*Reading et al., 2015; Zheng et al., 2019; Zhang and Alkhalifah,*

2020; *Puzyrev and Swidinsky, 2021*). In fact, geophysical inversion and machine learning share many common aspects as they both rely on statistical theory and tools of numerical analysis. Integrating machine learning approaches into the field of geophysics can bring benefits, and geophysical inverse modelling can particularly benefit from advancements in optimisation techniques, automatic-differentiation schemes and generative models developed in machine learning (*Sambridge et al., 2007; Kim and Nakata, 2018; Margossian, 2019; Laloy et al., 2019; Lopez-Alvis et al., 2021; Zhu et al., 2021; Sambridge et al., 2022; Valentine and Sambridge, 2023*).

Compared to classical physics-based models, ML models offer two significant advantages: (1) they excel at capturing highly-nonlinear and complex relationships between input and output variables, something that can be challenging for process-based physical models that often require simplifications and (2) once trained, they allow for significantly faster computation of a forward pass (response) compared to process-based models (*Russell, 2019; Zahura et al., 2020; Zhang et al., 2020*). While they offer numerous advantages, ML models also have certain disadvantages and limitations. They rely heavily on the quality, quantity, and representativeness of training data. They do not only require availability of data, but also data that represent the variability of the problem under consideration. Inadequate variability or small datasets might lead to over-fitting and limited generalisation capabilities. An additional major disadvantage of ML models, is their black box nature, which makes interpretation and understanding of their predictions challenging.

Nevertheless, there have been several recent advancements that help alleviate these limitations. Transfer learning enables the leveraging of knowledge from one task to enhance the performance of related tasks and facilitates the re-usability of ML models (*Yu and Ma, 2021*). Theory-guided designs of machine learning models (*Karpatne et al., 2017*) and physics-informed neural networks (PINNs; *Raissi et al., 2019*) integrate known domain knowledge and physical laws and constraints, allowing models to learn solutions that honour the underlying physical processes and exhibit generalisation capabilities even with limited data. Additionally, attribution and interpretation methods provide insights into the contributions and importance of input features in model predictions or outcomes, making ML models more interpretable (*Mamalakis et al., 2022; Toms et al., 2020*). These advancements together with the unique capabilities offered by ML models make ML highly suitable and appealing for many geoscience applications.

1.2 The forward problem

A general formulation of a forward problem is:

$$\mathbf{d} = g(\boldsymbol{\theta}) + \boldsymbol{\varepsilon}, \quad (1.1)$$

where $g(\cdot)$, the forward operator in geophysics is a mathematical representation of a geophysical experiment that projects the model parameters $\boldsymbol{\theta}$ in the model space into observable quantities that can be compared with the observed data \mathbf{d} . Here, $\boldsymbol{\varepsilon}$ represent errors associated with the forward response. These include two types of errors: epistemic and aleatoric, both

contribute to the overall uncertainty. The former is a result of factors such as measurement bias, simplifications and assumptions in models, or incomplete understanding of the underlying processes. These are often reducible through improved knowledge, or collection of more as well as other types of data. The latter type of errors are associated with the inherent uncertainty in the system being studied. These arise from factors that are inherently unpredictable, such as natural variability, measurement noise, or stochastic processes. Even though these error sources cannot be reduced by the accumulation of more data or additional information they can be quantified and accounted for statistically.

In this thesis, all the developed inversion approaches are tested for travel-time tomography experiments using crosshole ground-penetrating radar (GPR). This geophysical technique uses high-frequency electromagnetic waves to investigate structures and materials with applications mainly in the shallow subsurface. The basic principle of GPR involves transmitting a short pulse of electromagnetic energy into the ground using an antenna. This pulse travels as a wave through the subsurface and interacts with different materials and interfaces until a trace is recorded by the receiver antenna (Jol, 2008). The measured quantity in travel-time tomography is the first arrival of the signal to the receiver antenna. The governing equations of electromagnetic waves are the Maxwell's equations

$$\nabla \times \mathbf{E} = -i\omega\mu\mathbf{H} \quad (1.2)$$

$$\nabla \times \mathbf{H} = \sigma\mathbf{E} + i\omega\epsilon\mathbf{E}, \quad (1.3)$$

represented here in the frequency domain, where \mathbf{E} and \mathbf{H} are the electric and magnetic field vectors, ϵ , μ , and σ are the dielectric permittivity, magnetic permeability, and electrical conductivity parameters, respectively, ω is angular frequency and $i^2 = -1$. As can be understood from Eqs. 1.2-1.3, GPR responses are nonlinear by nature, and the electromagnetic waves interact with the medium they pass through and, therefore, their velocity and path can change depending on material properties (mainly electrical properties) causing wave reflection, refraction, scattering, attenuation and dispersion.

There are several ways to simulate the response of GPR electromagnetic waves numerically, depending on the purpose and desired accuracy. In this thesis, we rely on four types of travel-time tomography solvers for crosshole applications: straight-ray, shortest path, finite difference (Eikonal), and finite-difference time-domain, each based on different assumptions regarding the physics of the forward problem. The first three approaches are based on the ray-approximation, which describes the path travelled by electromagnetic waves as rays or beams. The ray approximation (known as geometrical optics) is based on the observation that for small wavelengths, a wave-field has the general characteristics of a plane wave, which can be treated as a collection of rays. This approximation is valid under certain conditions, particularly when the dimensions of objects and features in the subsurface are significantly larger than the wavelength (Born and Wolf, 2013).

A straight-ray solver is the simplest, cheapest and least accurate approach among the four solvers used here to model GPR responses. It makes two major assumptions: (1) the elec-

electromagnetic wave can be described by rays and (2) ray-paths are independent of material properties. The latter assumption implies linearity and straight paths between sources and receivers. The forward response is calculated on a discretized slowness (inverse of velocity) field, where the arrival time is an integration along the ray path of the ray length l times the slowness s (*Peterson, 2001*):

$$t = \sum_i l_i \cdot s_i. \quad (1.4)$$

Another solver, based on the shortest path, is used to determine the most efficient route between two points, considering a discretized slowness model and its available node points (*Dijkstra, 1959*). The shortest path implementation used herein is available in PyGIMLi (*Rücker et al., 2017*), an open-source library for multi-method modelling and inversion in geophysics, and utilises secondary nodes (found on the edges and sides of the cells) to enhance the accuracy of simulated travel times (*Giroux and Larouche, 2013*). Similar to the straight-ray approach, the travel time is calculated by integrating along the ray path:

$$t = \sum_i l_i(s) \cdot s_i, \quad (1.5)$$

however, it is important to note that the ray path is influenced by the slowness model, making it inherently nonlinear.

A solver based on a finite difference scheme, models the nonlinear response of the electromagnetic wave propagation by solving the Eikonal equation

$$\nabla t(\mathbf{x})^2 = s(\mathbf{x})^2, \quad (1.6)$$

a partial differential equation (PDE) that relates the wavefront travel time to the spatial coordinates in the model. In a finite difference approach, the Eikonal equation is discretized with a grid of nodes (stencils) approximating the continuous spatial domain. In this thesis we use the 2D finite difference algorithm introduced by *Podvin and Lecomte* (1991). This algorithm is based on the Huygens' principle and the plane wavefront approximation, which treats every point on a wavefront as a secondary source of a spherical wavelet that propagates outwards in all directions.

A widely used numerical method for simulating electromagnetic wave propagation is the finite-difference time-domain (FDTD) approach. As opposed to the Eikonal solver, which assumes infinite frequency, FDTD takes into consideration the frequency content of the emitted signal. It approximates the model domain both in space and in time and solves the Maxwell's equations to model electromagnetic wave propagation. As it better describes the finite wavelength of the wave it allows for the modelling of phenomena as diffraction, interference and attenuation, while handling complex geometries, materials, and boundary conditions. The algorithm used herein is a 2D MATLAB implementation by *Irving and Knight* (2006) that can perform surface reflection and borehole GPR modeling. The simulation is performed on a leapfrog, staggered-grid in which the electric and magnetic field components are computed and updated with respect to each other at non-overlapping locations and times.

As can be expected, the computation times grow with the complexity of the solver. The first three modelling approaches mentioned offer computational efficiency, making them beneficial when frequent calculations of the forward response are needed. However, it is important to acknowledge that using these simplified solvers introduces errors. These errors can be notable when the simplifying assumptions do not hold true (e.g., small features, sharp changes, etc.).

1.3 The deterministic inverse problem

For a linear over-determined inverse problem with Gaussian-distributed errors, a deterministic least-squares solution can be formulated as the minimisation of the squared residuals between the model predictions and the observed data (assuming identically-distributed and independent data errors) $E = (\mathbf{d} - \mathbf{G}\boldsymbol{\theta})^T (\mathbf{d} - \mathbf{G}\boldsymbol{\theta})$. Differentiating the objective function E and setting its derivative to zero gives the following solution (Menke, 2015):

$$\boldsymbol{\theta}_{est} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{d}. \quad (1.7)$$

This minimisation yields a single best estimate $\boldsymbol{\theta}_{est}$ that is optimal in terms of the data fit. The least-squares solution minimises the L_2 norm, also known as the Euclidean norm. This process is equivalent to maximising the likelihood of observing the data under the assumption of Gaussian errors (Menke, 2018). However, it is important to note that the solution to Eq. (1.7) exists only when the observations are sufficient to uniquely determine a solution. In practice, geophysical problems are often under-determined, resulting in an infinite number of solutions that exhibit similar data fits. To address this issue and stabilise the solution, additional constraints are incorporated into Eq. (1.7) through a process called regularisation. By using different types of regularisation, one can impose smoothness or other criteria on the solution, as well as similarity to some reference model (Tikhonov, 1963; Lelièvre et al., 2009; Menke, 2018). The resulting solution is a compromise between fitting the data and the regularisation. Determining the optimal balance between these two often requires an iterative process of trial and error.

For nonlinear problems, the inverse solution is obtained by iteratively linearizing the forward problem and relying on gradient information for minimisation. A widely used optimisation algorithm is gradient-descent which iteratively updates the model's parameters in the direction of the steepest descent of the objective function, scaled by a step length. Other algorithms for minimising the least-squares are the full Newton and Gauss-Newton methods, in which the gradient is scaled by the inverse of the Hessian matrix \mathbf{H} (second order derivative of E with respect to the model parameters) or by an approximation of it $\mathbf{H} \approx \mathbf{J}^T \mathbf{J}$ (where \mathbf{J} is the Jacobian, a first order derivative of the forward operator with respect to the model parameters). The use of the Hessian matrix or its approximation allows for considering second-order information, which can potentially provide faster convergence and better optimisation results compared to the gradient-descent method (Meju, 1994; Parker, 1994).

The solution obtained by these methods depends on the complexity of the problem and the initial model (see Figure 1.1). If the initial model is far from the global minimum, and the ob-

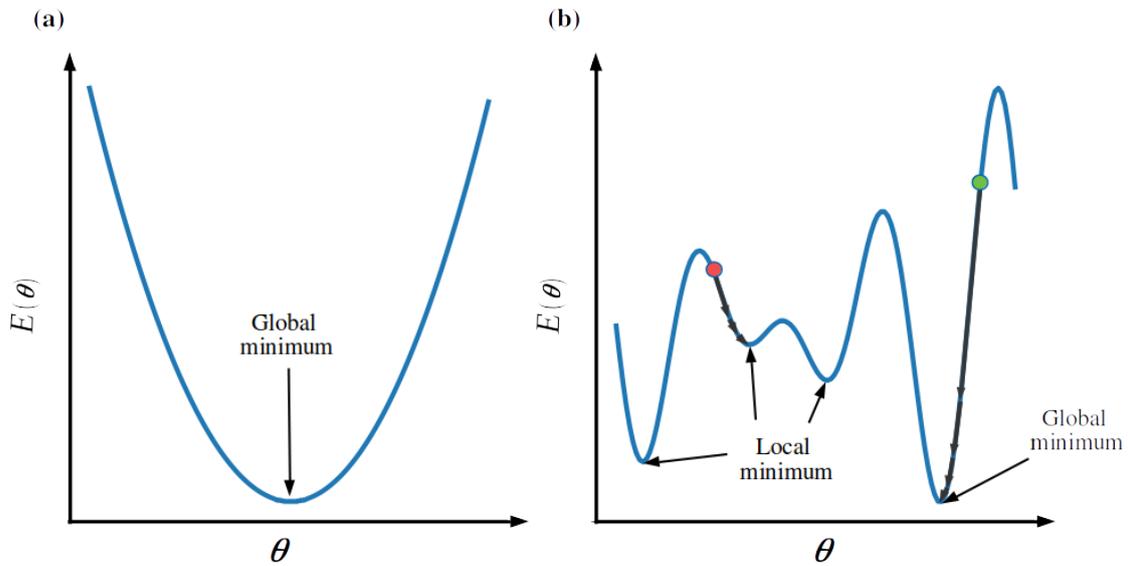


Figure 1.1: Illustration of the objective function for (a) linear and (b) nonlinear least-square problems. For a linear problem, the objective function is convex with a clear global minimum. For nonlinear problems, the objective function is non-convex, therefore, requiring linearization. A deterministic inversion in that case is sensitive to the choice of initial model as the objective function exhibits local minima. A close enough initial model (green circle) would allow the inversion algorithm to find a global minima and a far away model (red circle) would get stuck in a local minimum.

jective function exhibits high nonlinearity, there is a potential risk of the solution converging to a local minimum instead (Figure 1.1b). Another challenge arises when attempting to quantify uncertainty in the context of linearized operations. In such cases, the ability to accurately quantify uncertainty using linearized models is limited compared to the full complexity of the problem (*Alumbaugh and Newman, 2000*). One should also note that these approaches are limited to problems where data errors and regularisation measures can be assumed to have Gaussian distributions. Since these methods only work when obtaining a unique solution (typically the model that fits the data with the strongest regularisation constraints imposed), they offer limited abilities to assess the types of models of less regularised nature that are in agreement with the data.

Due to the limitations of deterministic inverse methods when confronting nonlinear problems and the growth of computer resources, probabilistic inversion methods are increasingly adopted (*Tarantola and Valette, 1982b*). These approaches offer a more robust and consistent framework for addressing the challenges associated with nonlinear inverse problems.

1.4 Bayesian inference

Bayesian inference is a statistical framework for updating our beliefs or knowledge about unknown parameters, based on new evidence or observational data. Considering a scenario where we have a set of N observations, denoted as $\mathbf{d} = (d_1, d_2, \dots, d_N)$, and we want to estimate a set of M parameters, denoted as $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_M)$. We can calculate the probability density function (PDF) of the parameters given the observations, known as the posterior, by combining a prior PDF and the conditional PDF of the data given the parameters using Bayes' rule:

$$p(\boldsymbol{\theta}|\mathbf{d}) = \frac{p(\mathbf{d}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{d})}. \quad (1.8)$$

The prior PDF $p(\boldsymbol{\theta})$ represents our knowledge or beliefs about the parameters $\boldsymbol{\theta}$ before we consider \mathbf{d} . One might have knowledge about the range of values that $\boldsymbol{\theta}$ can take on and perhaps also information about the distribution of different values within that range. The likelihood function $p(\mathbf{d}|\boldsymbol{\theta})$, is a conditional PDF that quantifies how well the model, defined by the parameter values $\boldsymbol{\theta}$, explains the observed data \mathbf{d} . The choice of likelihood function depends on the problem at hand and the specific characteristics of the data and their errors. Several common choices of likelihood functions include the Gaussian, binomial, Poisson, and exponential distributions, with the Gaussian likelihood being the most frequently used when dealing with continuous data. The term $p(\mathbf{d})$, referred to as the evidence, is a marginal likelihood over all possible values of $\boldsymbol{\theta}$

$$p(\mathbf{d}) = \int p(\mathbf{d}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}. \quad (1.9)$$

This normalising constant scales the posterior distribution to ensure that it is a valid distribution. In many cases, the evidence is intractable or very difficult to compute. As a result, when the model parameterization remains constant, implying that $p(\mathbf{d})$ is constant, and the goal is to approximate the posterior, the evidence is simply ignored. In this scenario, the posterior distribution is proportional to the product of the likelihood and the prior:

$$p(\boldsymbol{\theta}|\mathbf{d}) \propto p(\mathbf{d}|\boldsymbol{\theta})p(\boldsymbol{\theta}). \quad (1.10)$$

By disregarding the evidence, we can focus on approximating the relative posterior probabilities of different model realisations without explicitly computing the normalising constant. This makes Bayesian inference simpler using, for example, Monte Carlo techniques or variational inference, which provide approximate posterior distributions based on the proportional relationship stated above.

When relative posterior probabilities are insufficient or when comparing different conceptual models (Bayesian model selection; *Brunetti et al.*, 2017, 2019), it becomes necessary to consider the evidence, which may need to be estimated. When the evidence cannot be calculated analytically, various techniques can be employed, such as brute-force Monte Carlo sampling (BFMC; *Hammersley and Handscomb*, 1964), nested sampling (*Skilling*, 2006), annealed importance sampling (AIS; *Neal*, 2001), power posteriors (*Friel and Pettitt*, 2008) and sequential Monte Carlo (SMC) methods (*Friel and Wyse*, 2012). Evidence estimation

falls outside the scope of this thesis that primarily focuses on posterior approximation methods.

A general and straightforward approach to generate independent samples from the posterior distribution is rejection sampling (Figure 1.2a). In this algorithm, model proposals $\boldsymbol{\theta}_{prop}$ are drawn from a simple proposal distribution $g(\boldsymbol{\theta})$ and are accepted or rejected according to an acceptance probability. The acceptance probability is determined by drawing a sample from $u \sim \mathcal{U}(0, 1)$ and for a constant pre-determined bound c , the proposed model is accepted if $u > \frac{p(\boldsymbol{\theta}_{prop}|\mathbf{d})}{cg(\boldsymbol{\theta}_{prop})}$, otherwise it is rejected. The proposal distribution $g(\boldsymbol{\theta})$ should be chosen such that it covers the support of the target distribution. Although rejection sampling ensures that the accepted samples are distributed according to the target distribution, the efficiency of the algorithm depends heavily on the choice of the proposal distribution and the bounding constant (Ripley, 2009). Additionally, it often suffers from low acceptance rates making it inefficient and the time required to generate an adequate number of samples from the target distribution is unknown (Luengo et al., 2020).

A more efficient way to approximate the posterior is to use methods that prioritise exploitation, such as Markov chain Monte Carlo (MCMC; Robert et al., 1999) algorithms. These algorithms effectively explore the parameter space while enhancing efficiency by selectively sampling regions characterised by higher posterior probability.

1.4.1 Markov chain Monte Carlo

Markov chain Monte Carlo is a subset of Monte Carlo methods that construct a Markov chain with the desired target distribution $\pi(\boldsymbol{\theta})$ as its equilibrium distribution (Robert et al., 1999). In this thesis, the target distribution is the posterior from Eq. (1.8), thus, $\pi(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{d})$. As the name suggests, MCMC samplers rely on the dependence between subsequent states in the chain (Brooks et al., 2011). The model parameters $\boldsymbol{\theta}$ are treated as random variables, and the Markov chains are viewed as stochastic processes. A single or multiple MCMC chains are initialised with random values within the support of the prior distribution. They are then iteratively evolving by making model proposals in the form of small perturbations to the current state of the parameters. The new model proposal is subsequently accepted or rejected based on a probabilistic criterion. After an initial burn-in period (the period in which the chains are becoming independent of their initial state), the output of an MCMC algorithm is a sequence of correlated samples from the posterior distribution which can be used to calculate posterior distribution statistics as well as making predictions (Figure 1.2b). This methodology provides a way to draw samples from high-dimensional parameter spaces and complex posterior distributions without the need for explicit evidence calculations (Eq. 1.9).

In order to guarantee the convergence of the chains to their target distribution, they must maintain detailed balance and ergodicity. Detailed balance requires that the probability of transitioning between states with respect to the stationary distribution π is equal, such that $\pi(\boldsymbol{\theta}_i)p(\boldsymbol{\theta}_i \rightarrow \boldsymbol{\theta}_j) = \pi(\boldsymbol{\theta}_j)p(\boldsymbol{\theta}_j \rightarrow \boldsymbol{\theta}_i)$ (Mosegaard and Sambridge, 2002) and ergodicity ensures that the chain can explore the entire parameter space and reach any state with a non-zero probability. If an algorithm satisfies these two conditions, it will converge to the target distribution as long as their chains run for a sufficient duration, which in practice may

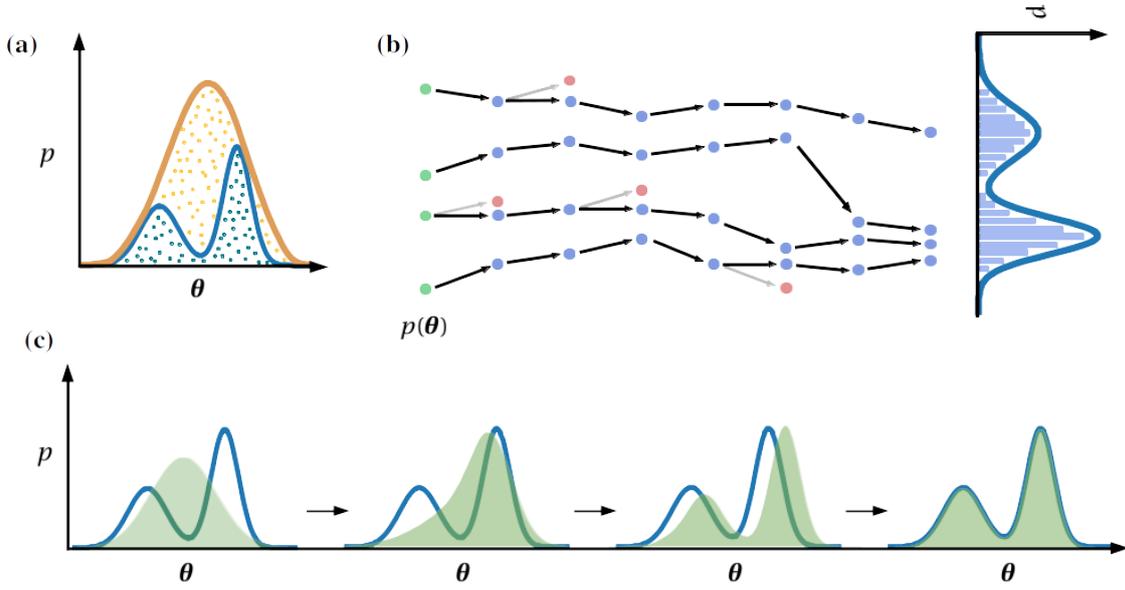


Figure 1.2: Illustration of (a) rejection sampling, (b) MCMC and (c) Variational inference. The blue curve in (a)-(c) is the posterior PDF $p(\boldsymbol{\theta}|\mathbf{d})$. The orange curve in (a) is the proposal distribution of the rejection sampling algorithm with the blue and orange dots being accepted and rejected samples. In (b), four MCMC chains are evolving simultaneously, where green, blue and red dots represent prior, accepted and rejected samples. The blue histogram under the blue curve in (b) is the posterior approximation using MCMC samples. The green distribution in (c) is a variational distribution that evolves to approximate the posterior PDF by learning its parameters.

be unacceptably long. A widely used and notable example of such a MCMC algorithm is the Metropolis-Hastings method.

In the Metropolis-Hastings algorithm, developed by *Metropolis and Ulam* (1949), *Metropolis et al.* (1953), and *Hastings* (1970), moves from the current state to the next state of the chain are proposed according to some proposal distribution $q(\boldsymbol{\theta}_{prop}|\boldsymbol{\theta}_{curr})$. Moves are accepted or rejected according to the acceptance probability

$$P_{accept} = \min\left(1, \frac{p(\boldsymbol{\theta}_{prop}|\mathbf{d})q(\boldsymbol{\theta}_{curr}|\boldsymbol{\theta}_{prop})}{p(\boldsymbol{\theta}_{curr}|\mathbf{d})q(\boldsymbol{\theta}_{prop}|\boldsymbol{\theta}_{curr})}\right). \quad (1.11)$$

In the special case where the the proposal distribution is symmetric such that $q(\boldsymbol{\theta}_{prop}|\boldsymbol{\theta}_{curr}) = q(\boldsymbol{\theta}_{curr}|\boldsymbol{\theta}_{prop})$, the acceptance probability reduces to $P_{accept} = \min\left(1, \frac{p(\boldsymbol{\theta}_{prop}|\mathbf{d})}{p(\boldsymbol{\theta}_{curr}|\mathbf{d})}\right)$ (Metropolis algorithm; *Metropolis and Ulam*, 1949). The performance and efficiency of the Metropolis-Hastings algorithm depend on factors such as the choice of the proposal distribution, the tuning of the proposal variance, and the convergence diagnostics used to assess the quality of the samples (*Cowles and Carlin*, 1996; *Laloy and Vrugt*, 2012).

A special case of the Metropolis-Hastings algorithm is the Gibbs sampler (*Geman and Geman, 1984*) where the acceptance probability of model proposals is always one (i.e., $P_{accept} = 1$; *Gelman, 1992; Hitchcock, 2003*). Instead of sampling from a joint multivariate distribution, the Gibbs sampler obtains samples from the marginal, conditional distributions. The Gibbs sampler works by sampling from the conditional distributions of each variable given the values of the other variables. A full cycle of the Gibbs sampler is obtained by iteratively updating all the variables $p(\theta_i | \theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_m)$. This however, requires that the conditional probabilities are readily available and can be effectively sampled from. An extension to Gibbs sampling is sequential Gibbs sampling (*Hansen et al., 2012, 2010*) in which sequential simulations are used to draw samples from a two- or multiple-point statistics (see Section 1.5.1) representation of the conditional prior distribution. Such an implementation of the Gibbs sampler can be used within the extended Metropolis algorithm (*Mosegaard and Tarantola, 1995*), where the acceptance probability becomes the ratio of likelihoods between proposed and current states.

Traditional MCMC algorithms, such as the Metropolis-Hastings and Gibbs sampling algorithms, use pre-defined proposal distributions throughout the entire simulation. However, these fixed proposals may not be optimal for efficiently exploring the posterior distribution, especially when the distribution has complex or multi-modal structures. Adaptive MCMC methods address this limitation by continuously updating the proposal distribution or tuning parameters based on the observed behaviour of the chain. The adaptation can be done in various ways, but the general idea is to use the information gathered from the chain to modify the proposal distribution to improve the exploration of the posterior distribution. Over the years, adaptive variants of the Metropolis algorithm have been developed, including adaptive Metropolis (AM; *Haario et al., 2001*), delayed rejection adaptive Metropolis (DRAM; *Haario et al., 2006*), differential evolution Markov chain (DE-MC; *Braak, 2006*), differential evolution adaptive Metropolis (DREAM; *Vrugt et al., 2008*), DREAM_(zS) and multiple-try DREAM_(zS) (MT-DREAM_(zS); *Laloy and Vrugt, 2012*). These adaptive methods aim to reduce the burn-in period, improve mixing between the chains and enhance convergence.

One popular adaptive MCMC algorithm is Hamiltonian Monte Carlo (HMC; *Duane et al., 1987; Neal, 2011*). It offers improved efficiency when sampling from complex probability distributions by combining ideas from Hamiltonian dynamics and Metropolis-Hastings. This algorithm consists of two main steps: a simulation period and a Metropolis-Hastings acceptance-rejection step. During the simulation period the algorithm simulates Hamiltonian dynamics using numerical integration methods, such as the leapfrog algorithm. This involves iteratively updating the momentum and position variables in Hamilton's equations over a fixed number of time steps, while preserving the total energy of the system. The final state resulting from the simulation is proposed as the new state and is accepted or rejected with some probability depending on the difference in energy between the current and proposed states. One key advantage of HMC is that it can propose moves to distant states as long as energy is conserved, which reduces the correlation between consecutive sampled states. This property enables more efficient exploration of the target distribution compared to traditional random-walk based MCMC algorithms. However, HMC requires the calculation of the gradient of the log-posterior with respect to the model parameters, which can be computationally expensive for complex models. Additionally, selecting appropriate

parameters for the simulation period can be non-trivial and impact the algorithm’s efficiency. Advancements in automatic differentiation strategies (*Griewank and Walther, 2008*) can help alleviate the former issue while extensions to HMC, such as the No U-Turn Sampler (NUTS) (*Hoffman and Gelman, 2014*), aim to address the latter.

1.4.2 Variational inference

Variational inference is a family of techniques used to approximate intractable target distributions (*Bishop and Nasrabadi, 2006*). It offers a computationally efficient alternative to more general Bayesian inference methods, such as MCMC. Instead of employing a random walk, variational inference, applied to Bayesian problems, approximates the posterior distribution by minimising the difference between a variational distribution and the true distribution. The variational distribution is typically a simpler distribution chosen from a parameterised family of distributions \mathcal{Q} . The approximation is formulated as an optimisation problem, where the objective is to find the set of parameters $q \in \mathcal{Q}$ that minimises the Kullback-Leibler Divergence (KLD; *Kullback and Leibler, 1951*), which is a measure of similarity between distributions (Figure 1.2c). As the marginal likelihood (evidence) is a constant, the KLD can be minimised by iteratively maximising the evidence lower bound (ELBO):

$$\log p(\mathbf{d}) = ELBO + D_{KL}(q_\phi(\boldsymbol{\theta}) || p(\boldsymbol{\theta}|\mathbf{d})), \quad (1.12)$$

where $ELBO = \mathbb{E}_q [\log p(\boldsymbol{\theta}, \mathbf{d})] - \mathbb{E}_q [\log q_\phi(\boldsymbol{\theta})]$. The expectation terms and their gradients can be approximated using a Monte Carlo estimator, which involves drawing samples from the variational distribution and the optimisation of the variational parameters is achieved through gradient-based techniques. Stochastic variational inference (SVI; *Hoffman et al., 2013*) is an efficient and scalable algorithm for performing variational inference. It uses natural gradients (gradients defined in the Riemannian space rather than in the Euclidean space) over a random mini-batch of samples to perform gradient ascent/descent, thereby, enhancing the optimisation process.

Variational inference has been widely adopted across numerous fields (*Blei et al., 2017* and references therein). In geophysics, it remains relatively understudied despite its potential for solving geophysical inverse problems (*Zhang et al., 2021b; Valentine and Sambridge, 2023*). Recent applications have demonstrated the effectiveness of variational inference in travel time tomography (*Zhang and Curtis, 2020a*), full-waveform inversion (*Zhang and Curtis, 2020b*), seismic image denoising (*Rizzuti et al., 2020*), well-log prediction (*Feng et al., 2021*), and hydrology (*Ramgraber et al., 2021*). Notably, many of these examples employ the Stein variational gradient descent (SVGD; *Liu and Wang, 2016*), a particle-based method that differs from traditional parametric variational inference algorithms. In addition, there is a growing interest in applying variational inference to train neural networks (NNs) (*Rizzuti et al., 2020; Lopez-Alvis et al., 2021; Zhao et al., 2022*).

1.5 Conceptual models and geological realism

Not all models obtained by inversion are meaningful or relevant for geoscientific purposes (*Lange et al.*, 2012). Models should not only be consistent with the observed data but also provide a realistic depiction of the physical property. For this reason it is essential to develop inversion approaches that can handle advanced and geologically-realistic priors.

A conceptual model is a simplified representation of a system or process that aims to capture the essential features or behaviours. It can be a geological description of spatial heterogeneity or the governing equations that describe the processes involved. In the context of this thesis, the term "conceptual models" specifically refers to the former and will be discussed in relation to the prior PDF.

The most basic representation of the prior is simply a uniform distribution with problem specific bounds. However, this representation is the least informative and one that maximises the entropy (*Gray*, 2011). A slightly more informative prior is a two-point geostatistical model defined by a mean and covariance assuming the subsurface structure to be a multivariate Gaussian. Nevertheless, these types of priors often fail to produce spatial models that reflect geological reality, and may even lead to incorrect predictions (*Gómez-Hernández and Wen*, 1998). This can become extremely important, for instance in flow and transport models, where two-points statistics poorly represents the connectivity of high-permeability values (*Zinn and Harvey*, 2003; *Jankovic et al.*, 2017 and reference therein). Over the past two decades, significant progress has been made in the fields of geostatistics and machine learning in capturing and representing higher-order statistics of geological models.

1.5.1 Geostatistical models

Geostatistical techniques use statistical models of random functions and variables to account for the uncertainty associated with simulating and estimating spatial variability. Traditional geostatistical methods like kriging and various simulation techniques such as sequential Gaussian simulations used in the geosciences (*Matheron*, 1963; *Azevedo and Soares*, 2017; *Sagar et al.*, 2018) are primarily based on variograms and represent two-point statistics. While these methods offer a straightforward and mathematically understandable approach to incorporating prior knowledge (*Hansen et al.*, 2006; *Linde et al.*, 2015a), they may not adequately address the complexity of the data (*Mariethoz*, 2018; *Tahmasebi*, 2018). The exploration of multiple-point statistics (MPS) began with early contributions from *Deutsch* (1992) and *Guardiano and Srivastava* (1993). In particular, *Guardiano and Srivastava* (1993) introduced the ENESIM algorithm enabling the reproduction of multiple-point statistics learned from a training image (TI). In the context of this thesis, a training image is a geological or geophysical representation that acts as a template or a reference that contains the desired spatial patterns and geological structures (i.e. channels, layers, or facies).

The idea of MPS is to consider higher-order interactions among multiple data points within a defined neighbourhood (*Hu and Chuginova*, 2008). MPS algorithms generate simulations by sequentially sampling multiple-point patterns from a TI that is consistent with a prior

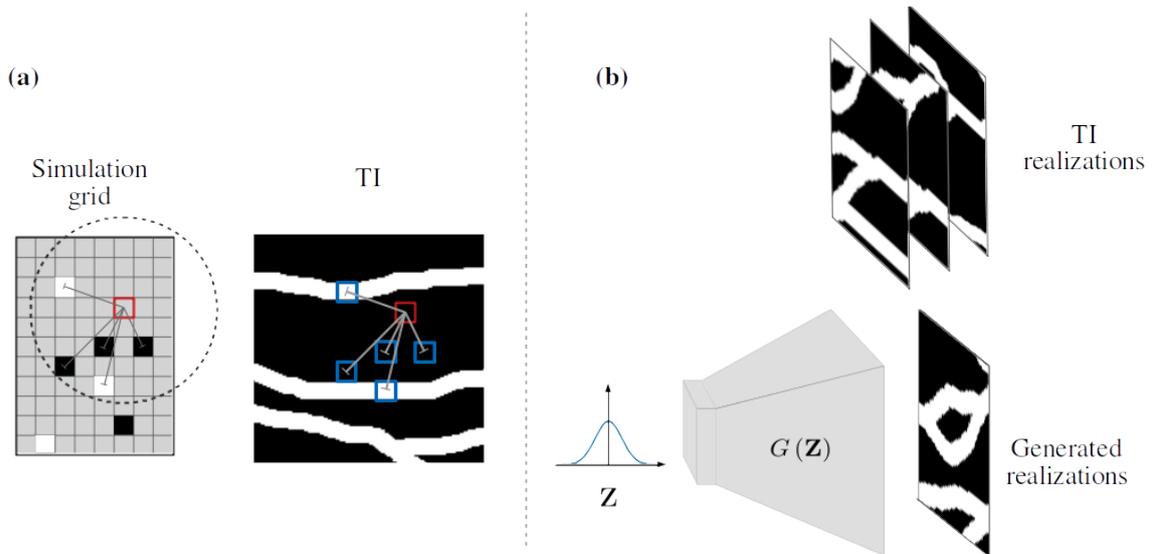


Figure 1.3: Illustration of (a) multiple-point statistics (MPS) and (b) deep generative model (DGM) simulations. In MPS, the simulation grid is populated sequentially based on patterns borrowed from a TI. For each simulated location x (red square), the TI is scanned to find a pattern (data event) that is similar to the one in the neighbourhood around x (dashed circle). In DGMs the generating transformation G (generator in generative adversarial networks and decoder in variational autoencoders) maps random variables \mathbf{Z} in the low-dimensional latent space into a high-dimensional image space to generate random realisations. Typically, the latent variables in DGMs are chosen to follow a multivariate Gaussian or uniform distribution and by sampling different values for \mathbf{Z} , DGMs can produce a wide range of realisations, including ones that are not present in the TI.

geological conceptualisation. To simulate a value at location x , the TI is scanned to identify occurrences of observed values at specific locations or points in the neighbourhood of x (data event; see Figure 1.3a). The values at location x associated with each identified occurrence in the TI are used to construct a conditional distribution $p(Z(x)|Z(x_1, \dots, Z(x_n)))$ from which $Z(x)$ is drawn to fill the value at location x in the simulation grid. This procedure is repeated for all the points in the simulation grid. The resulting simulations reproduce the spatial patterns and complex structures observed in the training image.

The ENESIM algorithm of *Guardiano and Srivastava* (1993) requires scanning the TI for each simulated location in the grid to build a conditional distribution. However, this is computationally intensive and impractical. To address this limitation, more advanced algorithms have been developed that use search trees. These algorithms scan the TI only once and organise patterns at different distances from a central location as nodes in a tree-like diagram (*Strebelle and Remy*, 2005). This allows for efficient retrieval of conditional histograms corresponding to different levels in the tree. However, these algorithms are typically designed for binary and categorical variables, and cannot handle continuous variables. The direct sampling (DS) introduced by *Mariethoz et al.* (2010a), allows direct sampling from the TI avoiding the storage of conditional distributions including for continuous variables. It defines a distance metric for different types of variables and a threshold in case the exact pattern is not found. This

random search and direct sampling from the TI is a prominent advantage of DS, however, it leads to unpredictable computational time and does not scale well to new evolving hardware. QuickSampling (QS; *Gravey and Mariethoz, 2020*), an efficient pixel-based MPS algorithm, leverages decomposition of standard distance metrics and fast Fourier transforms to compute a mismatch map between the data event in the simulation grid and the TI. Candidate values are sorted in ascending order of mismatch and one candidate is drawn according to a user pre-defined rank probability. This algorithm offers simulation quality that is comparable to DS, while taking advantage of available resources and providing a predictable computational time for a given TI size (*Gravey and Mariethoz, 2020*).

The finite nature of MPS priors imposes a significant limitation on uncertainty quantification because the number of patterns available is limited and extreme events cannot be generated beyond what is present in the training image. However, despite this constraint, MPS algorithms serve as valuable tools for representing conceptual models that exhibit complex pattern behaviours. In fact, there is a growing interest in integrating MPS algorithms into various inference frameworks (*Alcolea and Renard, 2010; Cordua et al., 2012; Lochbühler et al., 2014; Zahner et al., 2016; Brunetti et al., 2019*).

1.5.2 Generative models

A generative model is a mathematical or computational model that enables the generation of new samples representing a distribution of an underlying process. In the context of deep generative models (DGMs), these models are data-based deep neural networks that learn the statistical properties, patterns, and structures of a given training dataset or image. By capturing the essence of the data, DGMs generate new samples with similar characteristics as the original dataset. Various types of DGMs exist (*Bond-Taylor et al., 2022*); two well-known ones that have gained significant popularity in the geosciences are generative adversarial networks (GANs; *Goodfellow et al., 2014*) and variational autoencoders (VAEs; *Kingma and Welling, 2014*).

GANs are composed of two main components: a generator and a discriminator. The generator aims to transform random noise samples from a distribution of latent variables to data samples that follows some real data distribution, while the discriminator tries to distinguish between the real data (coming from a training dataset) and the generated data (coming from the generator). The generator and discriminator are trained together in a competitive (adversarial) process, whereby the generator improves its ability to generate data samples by learning from the feedback provided by the discriminator.

Autoencoders are also composed of two main components: an encoder and a decoder. Unlike GANs, autoencoders do not use adversarial training. Instead, the encoder maps input data to a low-dimensional representation in a latent space, where it learns a distribution over the latent variables. The decoder, in turn, reconstructs the data from the latent space. In traditional autoencoders, the autoencoder transformations is trained by minimising a misfit function between the input and output data. This leads to an unknown, unstructured latent space. An improved version of autoencoders that forces a structure on the latent space is the variational autoencoder. The VAE uses variational Bayesian inference to train the two

networks. With the same principle as in section 1.4.2, the VAE is trained by maximising the ELBO which is composed of a reconstruction error and a regularisation term encouraging the encoded distribution to be close to some prior (typically standard normal distribution).

Compared to MPS, DGMs have several advantages. First, DGMs use a continuous prior probability distribution to generate realisations (Figure 1.3), therefore, are able to produce patterns that are not necessarily present in the training image/dataset. Second, they provide model parameterization, therefore, allow a direct continuous perturbation of model parameters that is essential in some inversion algorithms (e.g. gradient-based; *Laloy et al., 2019; Lopez-Alvis et al., 2021*). In addition to their data generation capabilities, GANs and VAEs offer a significant advantage: the ability to reduce the model dimension by using a low-dimensional latent space. As opposed to other dimensionality-reduction techniques (e.g. principal component analysis, discrete cosine transform and singular value decomposition), DGMs can handle strongly nonlinear relationships and provide random samples that are in agreement with the dataset on which they were trained (*Laloy et al., 2017*). Depending on the spatial correlation in the training samples, dimensionality reduction can be substantial, resulting sometimes in a difference of more than four orders of magnitude between the sizes of the latent and high-dimensional spaces. Due to their ability to rapidly generate realisations (orders of magnitude faster than MPS) and perform inference in a lower-dimensional latent space, DGMs present a substantial reduction in computational demand, especially for sampling methods such as MCMC (*Laloy et al., 2017, 2018*). Despite the advantages of DGMs, they require large training datasets and time for training, involving tasks such as building the network architecture through trial and error and the actual training process. The dimensionality of the latent space, for example, is one of the hyperparameters and can vary depending on the problem being addressed, the spatial correlation in the training samples and the DGM used. Moreover, compared to MPS, conditioning realisations to hard data is not as straightforward in DGMs.

Both GANs and VAEs have proven to be versatile tools in a range of applications due to their representation and dimensionality reduction capabilities. They have been successfully applied in signal processing tasks (*Si et al., 2020; Siahkoohi et al., 2019*), data conditioning scenarios (*Zhang et al., 2021a; Dupont et al., 2018*), as well as for enhanced representation in ensemble and sampling methods (*Scheiter et al., 2022*). Furthermore, these models have been used in prior and posterior parameterizations, enabling efficient inversion and estimation of model parameters (*Laloy et al., 2017, 2018; Canchumuni et al., 2019; Mosser et al., 2020; Zheng et al., 2020; McAliley and Li, 2021*).

A third type of generative model considered in this thesis is flow based models, referred to as normalising flows (*Papamakarios et al., 2021*). These models are based on invertible, smooth transformations, mapping samples from a simple distribution to a target distribution. The transformation is learned such that the target distribution becomes an approximation to an otherwise intractable distribution. A key advantage of normalising flows arises from the use of invertible transformations. This property enables density estimation and efficient sampling from the learned distribution. However, this property is also a limiting factor as it requires the input dimension to be equal to the output dimension. As a consequence, these models might scale poorly to high-dimensional problems and, therefore, may be considered

less attractive as generators compared to GANs and VAEs while still being very attractive for inversion-purposes.

1.6 Objectives and outline

Sampling-based methods such as MCMC used to approximate distribution densities often suffer from excessive computing times. This happens as they typically require a large number of samples to explore the posterior and accurately approximate it. Generating each sample involves a model proposal step that can be expensive (e.g., MPS) and the evaluation of the forward response that can be demanding when considering elaborate forward models. This thesis aims to improve existing methods, such as MCMC, or introducing new efficient approaches to perform inverse modelling for geophysical applications.

Improved efficiency can be achieved by: (1) low-dimensional parameterizations of the prior, (2) reducing the computational requirements associated with the forward response, (3) minimising the total number of forward response computations needed and (4) parallelization of computation. In terms of prior representations, both MPS and DGMs offer distinct advantages and disadvantages. MPS allows for easy sampling of various priors and straightforward conditioning to hard data while modifying a prior would necessitate retraining of the DGM, and conditioning to hard data is challenging. On the other hand, DGMs provide a compact parameterization of the prior and fast model proposals that are compatible with the majority of available inverse modelling methods as prior probabilities can be calculated and gradients with respect to model parameters are easy to obtain. The use of either MPS or DGMs is subjected to the objectives and requirement of the inverse problem. In this thesis, both MPS and DGMs are employed as tools for representing and sampling from the prior distribution, thereby, preserving geological realism.

The second point mentioned above can be addressed by surrogate modelling. Nevertheless, using computationally-efficient yet simplified forward solvers, such as those described in section 1.2, come with a price: modelling errors. These errors have to be accounted for in order to avoid bias and over-confident estimations (*Brynjarsdóttir and O-Hagan, 2014*). We explore the idea of encoding the model errors arising from surrogate modelling in the latent space of a DGM, and simultaneously inferring both the subsurface model (also encoded in the latent space of a DGM) and the modelling errors (Chapter 2). The improvement has two aspects. First, by encoding both the subsurface model and the model error in a compact, low-dimensional space, there is a substantial reduction in the number of parameters to be inferred. Second, the use of an efficient solver that is significantly faster than its high-fidelity counterpart allows for important gains in computing time.

To achieve a reduction in the number of forward responses, one can leverage gradient information. While gradient-based inversion methods are known for their efficiency due to their explorative nature, they are sensitive to the initial model and can encounter local minima when employed for nonlinear problems (Section 1.3). This issue becomes even more pronounced when dealing with highly nonlinear transformations, such as those used in DGMs (*Laloy et al., 2019*). We explore how to exploit the advantages of methods that combine

stochastic aspects with gradient-based optimisation to gain an efficient yet robust approach for inferring model parameters encoded in the latent space of a DGM (as in Chapter 3). By incorporating gradient information, the model parameter space can be efficiently explored. By estimating gradients using random samples, the limitations of purely deterministic methods can be mitigated, as the gradients are defined over a larger support.

A different approach that avoids elaborate sampling to explore the posterior and is fully stochastic, is to condition sequential simulations to indirect geophysical data. The objective here is to develop a methodology that enables such conditioning. In this approach, the number of forward evaluations are finite and known (the number of model cells) while the simulation can be performed using efficient MPS schemes (Chapter 4). An additional advantage of this approach is that individual simulation runs are independent of each other, allowing them to be executed in parallel. This leads us to the aspect of parallelism. The parallelization of MCMC samplers is limited by the number of chains employed (provided that the individual forward solvers are not parallelized as, for example, in *Hunziker et al.* (2019)), as the sampling process occurs sequentially within each chain. The objective of this thesis is to introduce methods that can be effectively scaled to high-dimensional problems, enabling the distribution of work across available computational resources (as discussed in Chapters 3 and 4).

This thesis touches on each of the aforementioned points and include work that was published in peer-reviewed journals (Chapters 2 and 3) and work that is soon to be submitted (Chapter 4). It is outlined as follows:

Chapter 2 introduces an approach to account for modelling errors that arise when using surrogate models to reduce the computational cost of MCMC sampling. The model error represents the discrepancy between two fidelity-varying forwards solvers. The model error as well as the model are encoded into two separate low-dimensional, latent spaces of a GAN. During each MCMC step, the simple low-fidelity solver is used while the observed (test) data are obtained using a high-fidelity solver. The model error realisation, generated by the GAN is subtracted from the forward response to correct the simulated data before being compared to the observed data. The results of the MCMC inversions are estimates of the posterior PDFs of the model and the model errors. We compare this approach with inversion approaches in which model errors are either not considered or considered by inflating the error term in the likelihood function using an error covariance model.

Chapter 3 introduces an approach to speed up the inversion process by using inverse autoregressive flows (a type of normalising flows) and variational Bayes to approximate the posterior in the latent space of generative models. In this approach a transformation, representing a mapping between a standard normal distribution and a target distribution with which the posterior distribution is approximated, is trained using variational Bayes. The posterior is approximated for the model parameters in the latent spaces of either a GAN or a VAE. We compare this approach with MCMC sampling in terms of performance and computational time.

Chapter 4 presents a new inversion approach that is based on conditioning MPS simulations to indirect geophysical (linear) data. During the simulation, the MPS algorithm consid-

ered (QS) provides candidate values sampled according to a TI for which likelihoods are approximated. The likelihood approximation is based on kriging, where the unknown model parameters (not yet simulated) are conditional on known as well as the candidate samples. A value is assigned to a simulated location in the simulation grid by drawing one candidate proportionally to the approximated likelihood. One complete conditional realisation is considered a draw from the posterior. Simulation runs are independent and can be executed in parallel and used to estimate posterior statistics at much shorter times than when using MCMC.

Chapter 5 provides a summary of the thesis as well as general conclusions, remarks on limitations and an outlook.

Contribution statement: I had the main responsibility for the development of all research projects (Chapters 2, 3 and 4) in this thesis, including conceptualisation, methodology, software implementation, formal analysis and writing the original draft of the manuscripts.

Code availability: The SGAN-ME, Neural-transport and IDCS codes are available at the following GitHub repository: <https://github.com/ShiLevy>.

Chapter 2

Using deep generative neural networks to account for model errors in Markov chain Monte Carlo inversion

Shiran Levy, Jürg Hunziker, Eric Laloy, James Irving, Niklas Linde

Published¹ in *Geophysical Journal International* and herein slightly adapted to fit the theme of this thesis

¹Levy, S., Hunziker, J., Laloy, E., Irving, J., and Linde, N. (2022). Using deep generative neural networks to account for model errors in Markov chain Monte Carlo inversion. *Geophysical Journal International*, **228**(2), 1098-1118.

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 traveltimes tomography inverse problem, we train SGAN networks on traveltimes 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 traveltimes 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.

2.1 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 and Valette, 1982b) 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 and Valette, 1982b; 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 *Rammy 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 Billionis, 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., 2010b; 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.

2.2 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 travelttime tomography inverse problem.

2.2.1 Database preparation

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}_Φ of pre-defined size which are randomly cropped from the segment of the TI intended for training. The porosity field Φ is then computed from the multi-Gaussian realizations using the lognormal transformation

$$\Phi = \exp(\mathbf{X}_\Phi \times 0.22361 - 1.579) \quad (2.1)$$

in *Pirot et al.* (2017).

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 Φ -realizations described in subsection 2.2.1 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*):

$$\boldsymbol{\kappa}_b = \Phi^m \boldsymbol{\kappa}_w + (1 - \Phi^m) \boldsymbol{\kappa}_s, \quad (2.2)$$

and

$$\mathbf{s} = \frac{\sqrt{\boldsymbol{\kappa}_b}}{c}, \quad (2.3)$$

where $\boldsymbol{\kappa}_w$ and $\boldsymbol{\kappa}_s$ 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 $\boldsymbol{\kappa}_w$ to be 81, $\boldsymbol{\kappa}_s$ 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:

$$\mathbf{d} = g^{HF}(\mathbf{s}) + \boldsymbol{\epsilon}, \quad (2.4)$$

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

$$\mathbf{d} = g^{LF}(\mathbf{s}) + \boldsymbol{\eta}(\mathbf{s}) + \boldsymbol{\epsilon}, \quad (2.5)$$

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

$$\boldsymbol{\eta}(\mathbf{s}) = g^{HF}(\mathbf{s}) - g^{LF}(\mathbf{s}). \quad (2.6)$$

To test our method, we consider two different model errors for the crosshole GPR traveltimes 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:

$$\hat{E}(\mathbf{x}_{trn}, \mathbf{x}_{rec}, \boldsymbol{\omega}) = \frac{E(\mathbf{x}_{trn}, \mathbf{x}_{rec}, \boldsymbol{\omega})}{\sqrt{\frac{2\pi T(\mathbf{x}_{trn}, \mathbf{x}_{rec})}{-i\boldsymbol{\omega}\bar{\kappa}\mu_0}}}, \quad (2.7)$$

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.

2.2.2 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

$$h(\mathbf{x}; \mathbf{w}, b) = \varphi\left(\sum_{i=1}^{N_x} w_i x_i + b\right), \quad (2.8)$$

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} :

$$\mathbf{F}_{u,v} = \varphi\left(\sum_{j=1}^{N_W} \sum_{i=1}^{N_H} \mathbf{K}_{i,j} \mathbf{I}_{u+i,v+j} + b\right). \quad (2.9)$$

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. 2.1). 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 $W(P_r, P_g)$:

$$\min_G \max_{D \in \mathcal{D}} \mathbb{E}_{\mathbf{X} \sim P_r} [D(\mathbf{X})] - \mathbb{E}_{\mathbf{Z} \sim p_g} [D(G(\mathbf{Z}))]. \quad (2.10)$$

Given that the output of $D(\cdot)$ in equation (2.10) 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.

2.2.3 SGAN architecture and training

The network architecture of the generator and critic are asymmetric with respect to each other (see Appendix 2.6.1 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

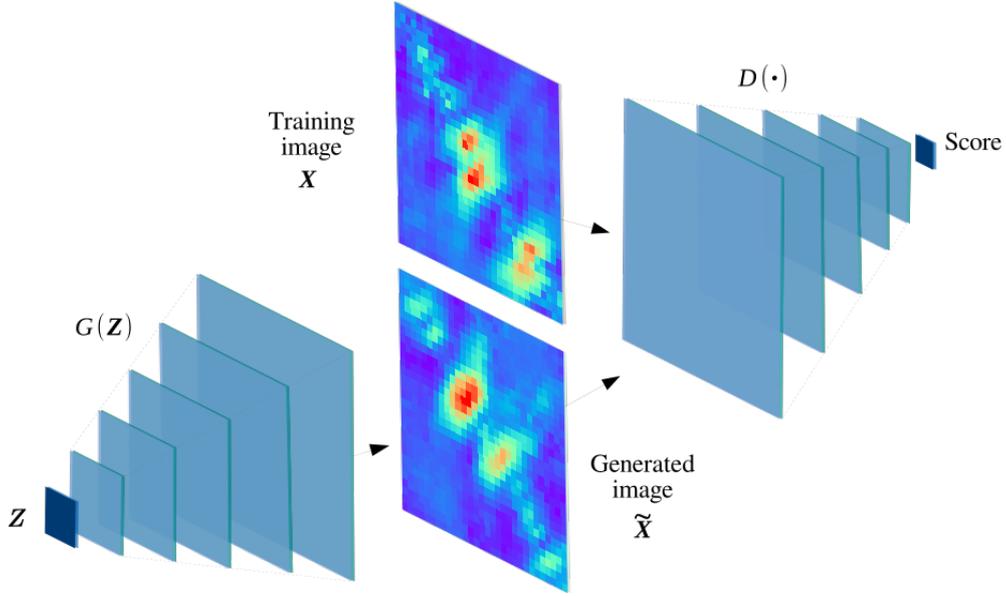


Figure 2.1: Illustration of our SGAN architecture with five layers when applied to represent model errors. During training, each parameter in the latent space \mathbf{Z} is randomly drawn from a uniform distribution $\mathcal{U}(-1, 1)$. Each \mathbf{Z} is transformed into a single image \tilde{X} through 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.

(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 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 $l = 1, \dots, 5$ in the generator can be calculated via the following relationship:

$$x_l = s \cdot (x_{l-1} - 1) - 2 \cdot p + (k - 1) + 1, \quad (2.11)$$

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 B for more details).

All images fed into the critic must be normalized to a $[-1, 1]$ range and have the same dimensions. Thus, TIs 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 $\tilde{\mathbf{X}}_{\Phi}$ or model-error $\tilde{\mathbf{X}}_{\eta}$ 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 2.1.

2.2.4 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:

$$p(\mathbf{Z}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{Z})p(\mathbf{Z})}{p(\mathbf{d})}, \quad (2.12)$$

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:

$$l(\mathbf{d}|\mathbf{Z}) = -\frac{N_d}{2} \log(2\pi) - N_d \log(\sigma) - \frac{1}{2} \sigma^{-2} [\mathbf{d} - \mathbf{d}_{sim}]^2, \quad (2.13)$$

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}_Φ describing the subsurface model parameters and \mathbf{Z}_η 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 Φ and η_{app} (approximate model error), where a low-fidelity forward response is calculated on the porosity field Φ converted to slowness \mathbf{s} using equations (2.2) and (2.3). 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 (2.13) is replaced with $g^{SR}(\mathbf{s}) + \nu\eta_{app}$. 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 2.2.

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 $\mathbf{Z} = \mathbf{Z}_\Phi$ and we simply plug $\mathbf{d}_{sim} = g^{SR}(\mathbf{s})$ into equation (2.13).

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:

$$l(\mathbf{d}|\mathbf{Z}) = -\frac{N_d}{2} \log(2\pi) - \frac{1}{2} \log(|\mathbf{C}_D|) - \frac{1}{2} [\mathbf{d} - g^{SR}(\mathbf{s}) - \mathbf{d}_{ME}]^T \mathbf{C}_D^{-1} [\mathbf{d} - g^{SR}(\mathbf{s}) - \mathbf{d}_{ME}], \quad (2.14)$$

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)

```

1 Set  $t = 1$  and initialize the archive with realizations  $\mathbf{Z}_\Phi, \mathbf{Z}_\eta$  and  $v$  randomly drawn from  $p(\mathbf{Z}_\Phi)$ ,
    $p(\mathbf{Z}_\eta)$  and  $p(v)$  (respectively)
2 Initialize  $\mathbf{Z}^t = [\mathbf{Z}_\Phi^t, \mathbf{Z}_\eta^t, v^t]$  for each MCMC chain
3  $\tilde{\mathbf{X}}_\Phi^t, \tilde{\mathbf{X}}_{\eta_{app}}^t \leftarrow G_\Phi(\mathbf{Z}_\Phi^t), G_\eta(\mathbf{Z}_\eta^t)$ 
4 Perform post-processing (section 2.2.3):  $\Phi^t, \eta_{app}^t \leftarrow \tilde{\mathbf{X}}_\Phi^t, \tilde{\mathbf{X}}_{\eta_{app}}^t$  and convert  $\Phi^t$  into slowness  $\mathbf{s}^t$ 
   (equations (2.2)-(2.3))
5  $\mathbf{d}_{sim} = g^{LF}(\mathbf{s}^t) + v^t \eta_{app}^t$ 
6 Compute  $l(\mathbf{d}|\mathbf{Z}^t)$  (equation (2.13))
7 while  $t < N_{draw}$  do
8   | Propose a new sample  $\mathbf{Z}'_\Phi, \mathbf{Z}'_\eta$  and  $v'$  from proposal distribution  $q(\mathbf{Z}', \mathbf{Z}^{t-1})$ 
9   |  $\tilde{\mathbf{X}}'_\Phi, \tilde{\mathbf{X}}'_{\eta_{app}} \leftarrow G_\Phi(\mathbf{Z}'_\Phi), G_\eta(\mathbf{Z}'_\eta)$ 
10  | Perform post-processing (section 2.2.3):  $\Phi', \eta'_{app} \leftarrow \tilde{\mathbf{X}}'_\Phi, \tilde{\mathbf{X}}'_{\eta_{app}}$  and convert  $\Phi'$  into
   | slowness  $\mathbf{s}'$  (equations (2.2)-(2.3))
11  |  $\mathbf{d}_{sim} = g^{LF}(\mathbf{s}') + v' \eta'_{app}$ 
12  | Compute  $l(\mathbf{d}|\mathbf{Z}')$ 
13  | Compute probability of acceptance  $\alpha \leftarrow e^{(l(\mathbf{d}|\mathbf{Z}') - l(\mathbf{d}|\mathbf{Z}^{t-1}))}$ 
14  | Draw  $U$  from a uniform distribution  $\mathcal{U}(0, 1)$ 
15  | if  $U < \alpha$  then
16  |   |  $\mathbf{Z}^t \leftarrow \mathbf{Z}'$ 
17  | else
18  |   |  $\mathbf{Z}^t \leftarrow \mathbf{Z}^{t-1}$ 
19  | end
20  |  $t = t + 1$ 
21 end
22 Function  $G_\Phi(\mathbf{Z}_\Phi)$ 
23   | Performs a series of transposed convolution layers with pre-trained weights
24   | return  $\tilde{\mathbf{X}}_\Phi$ 
25 end
26 Function  $G_\eta(\mathbf{Z}_\eta)$ 
27   | Performs a series of transposed convolution layers with pre-trained weights
28   | return  $\tilde{\mathbf{X}}_{\eta_{app}}$ 
29 end

```

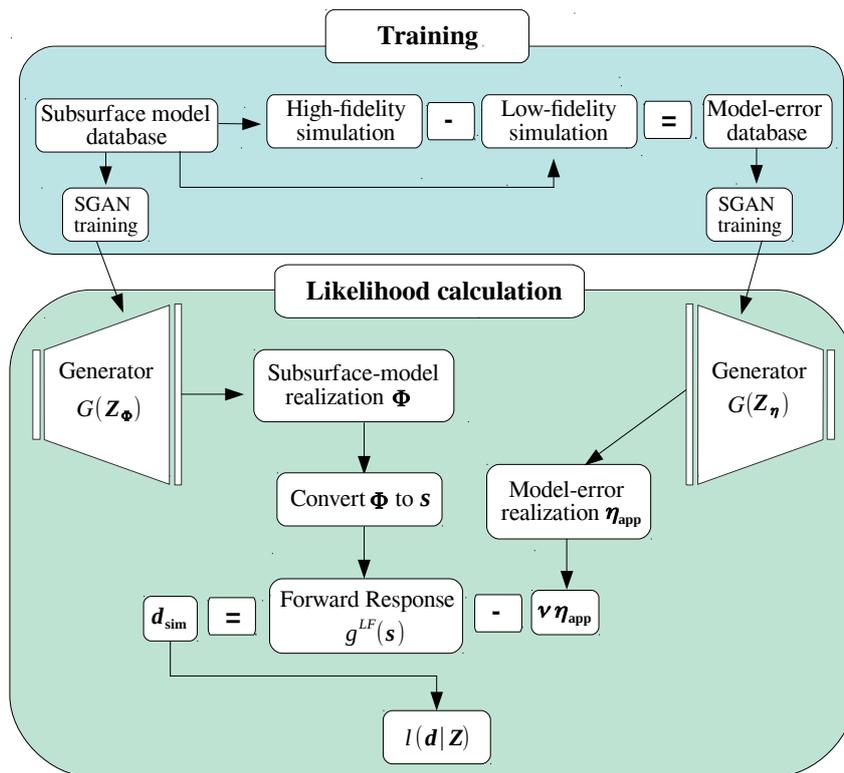


Figure 2.2: SGAN-ME workflow. Subsurface-model representation using SGAN is discussed in details in the work of *Laloy et al. (2018)*, here we focus on model-error representation.

2.3 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 κ_b and electrical conductivity fields as input. We assume a constant conductivity of 0.002 S/m across the model domain. The dielectric constant κ_b is obtained using equation (2.2). The model-error databases corresponding to $\eta^{eikonal-SR}$ and $\eta^{FDTD-SR}$ 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.

2.3.1 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}_Φ and \mathbf{Z}_η , 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:

$$\tilde{\mathbf{X}} = G(\mathbf{Z}) \cdot (\mathbf{M}_x \oslash \mathbf{M}_{\tilde{x}}), \quad (2.15)$$

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 2.3 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 2.3a-b) and the SGAN realizations before (Figs 2.3c-d) and after (Figs 2.3e-f) applying the correction in equation 2.15. 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 2.3d and f). The spatial statistics represented by the di-

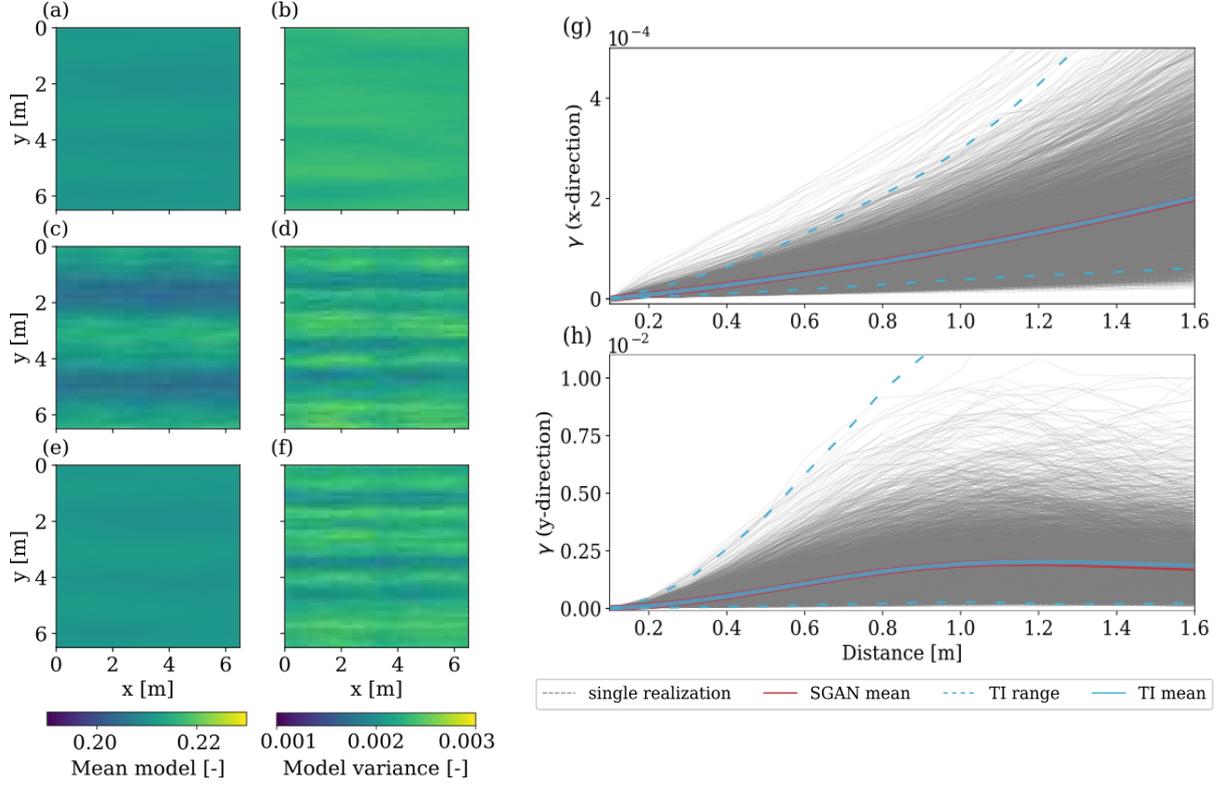


Figure 2.3: Statistics of subsurface model-parameter realizations after 58,000 training iterations. Mean and variance images calculated on 5,000: (a-b) TI realizations, (c-d) SGAN realizations before mean correction and (e-f) SGAN realizations after mean correction. The directional semivariograms in 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.

rectional semivariograms in x - and y -directions are given in Figures 2.3g 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 2.4. In most cases, $\eta^{FDTD-SR}$ has a larger range of error values compared to $\eta^{eikonal-SR}$ and displays similar features to $\eta^{eikonal-SR}$ with additional off-diagonal patterns. Figure 2.5 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 $\eta_{app}^{eikonal-SR}$ realizations before correction (Fig. 2.5c) is close to that of the TI realizations (Fig. 2.5a), it underestimates the model-error mean on the diagonal. After

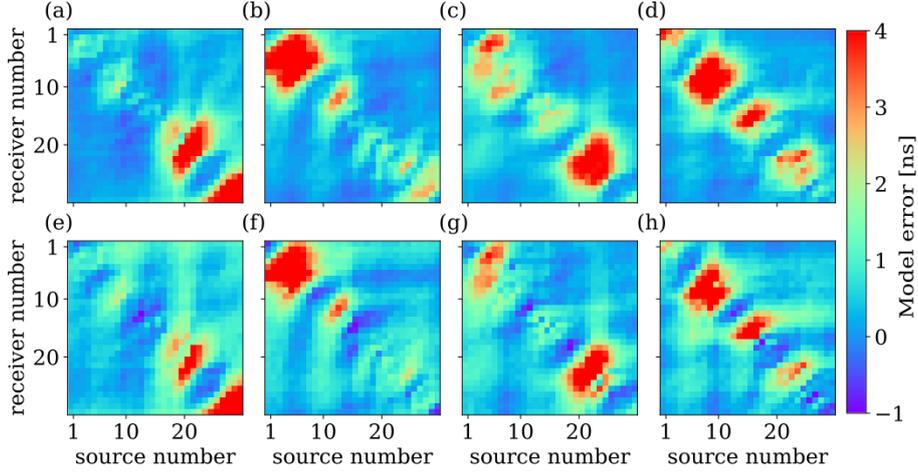


Figure 2.4: Examples of actual model-error realizations (a-d) $\eta^{eikonal-SR}$ and (e-h) $\eta^{FDTD-SR}$. Figures in the same column were calculated for the same subsurface model-parameter realization.

correction (Fig. 2.5e), the bias in the mean is removed and the variance (Fig. 2.5f), which also suffers from underestimation on the diagonal, is slightly improved. Training with $\eta^{FDTD-SR}$ realizations proved to be more challenging and required larger number of training iterations (450,000 iterations as opposed to 250,000 for $\eta^{eikonal-SR}$). The SGAN mean image before correction (Fig. 2.5i) is distorted compared to the TI mean image (Fig. 2.5g). We attribute this difference to the patchy nature of the $\eta^{FDTD-SR}$ realizations and features that extend to elements further off-diagonal (Fig. 2.4). These distortions were reduced after applying the mean correction (Fig. 2.5k), although improvements in the variance (Fig. 2.5l) are not as visible. One can observe a broken pattern on the diagonal in the $\eta^{FDTD-SR}$ TI's mean and variance images (Figs. 2.5g and h). This pattern can also be found in $\eta^{eikonal-SR}$ TI's mean and variance images (Figs. 2.5a 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 2.2.1. 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 2.6 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.

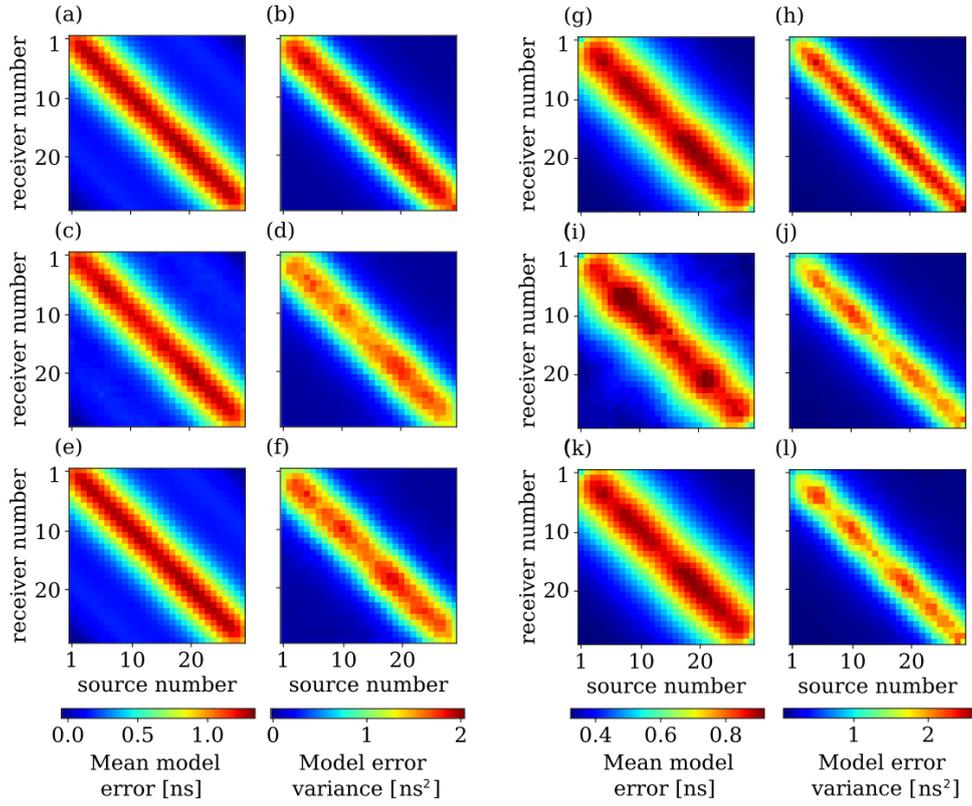


Figure 2.5: Model errors of (a-f) $\eta^{eikonal-SR}$ and (g-l) $\eta^{FDTD-SR}$. Pixel-wise mean and variance of 10,000 (a-b and g-h) TI realizations, (c-d and i-j) SGAN realizations before mean correction and (e-f and k-l) SGAN realizations after mean correction (see equation (2.15)).

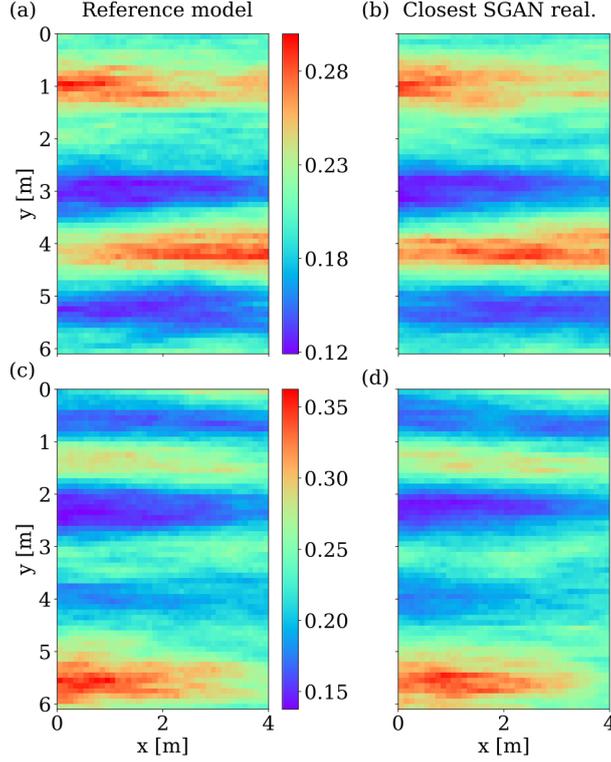


Figure 2.6: Reference models (a) 1 and (c) 2 and (b and d) corresponding closest SGAN realizations obtained from pixel-to-pixel inversion considering 25 latent parameters.

2.3.2 Inversion results

We perform inversion of data generated from the two multi-Gaussian reference models in Figures 2.6a and 2.6c 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 traveltimes 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}_Φ 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}_η along with auxiliary parameter ν . Note that the number of parameters in \mathbf{Z}_Φ and \mathbf{Z}_η is chosen based on a trade-off between inversion performance and efficiency. It is chosen such that it remains low while ensuring 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 $RMSE_\Phi$ and the other on the data denoted $RMSE_d$.

Table 2.1: 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 |

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 2.6.2). 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 ($MSE(\boldsymbol{\eta}_{ref}, 0)$).

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 2.1, 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 2.1 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.

Test Case 1: eikonal - straight-ray model error

We first consider inversion results with model error $\eta^{eikonal-SR}$ in terms of maximum-likelihood solutions of the straight-ray, covariance, SGAN-ME and eikonal-based inversion approaches in Figure 2.7 and $RMSE_{\Phi}$, SSIM and $RMSE_d$ in Table 2.2. Generally speaking and given values in Table 2.2, 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 2.8 together with values in Table 2.3, 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 2.3 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 2.8c-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 2.8c-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 $RMSE_{\Phi}$ as the straight-ray solution (0.016) but receives higher SSIM (0.74 versus 0.72) and slightly lower $RMSE_d$ (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 ($RMSE_{\Phi}$ and SSIM). As opposed to Model 1, here the straight-ray solution fits the data significantly better than the covariance solution ($RMSE_d$ of 0.87 ns for straight-ray versus 1.17 ns for covariance). Furthermore, the straight-ray 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 2.2, the $RMSE_d$ 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 2.9 the $RMSE_d$ 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 $\eta^{eikonal-SR}$ (Figs 2.9a and b) indicate that our SGAN-ME approach fits the

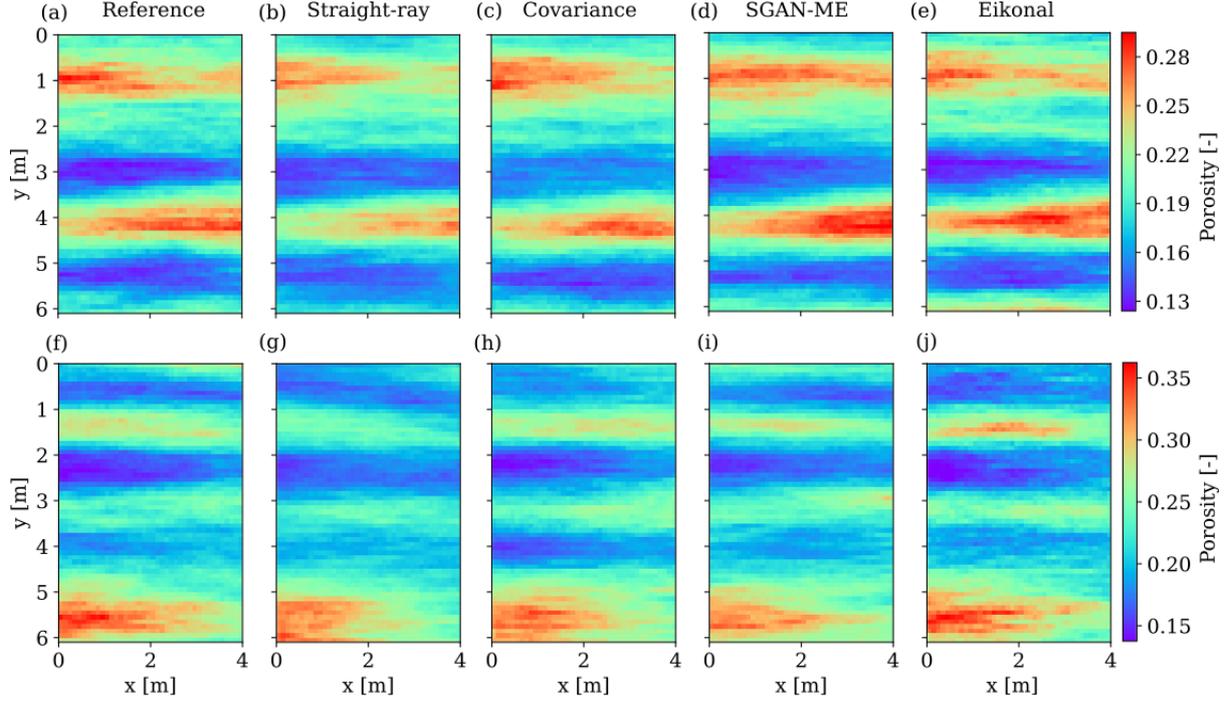


Figure 2.7: Inversion results for reference Models (a) 1 and (f) 2 for Test Case 1 ($\eta^{eikonal-SR}$). (b)-(e) and (g)-(j) are the maximum-likelihood realizations obtained from inversion using the straight-ray, covariance, SGAN-ME and eikonal approaches. The first three approaches use the straight-ray solver for the forward response during inversion, while the observed data for all approaches were created using the eikonal solver.

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 $RMSE_d$ 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 $RMSE_{\Phi}$ and SSIM distributions (Figs. 2.10a,b,e,f). The $RMSE_{\Phi}$ 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 $RMSE_{\Phi}$ and SSIM distributions that rank higher than the straight-ray and covariance approaches. For Model 2, the $RMSE_{\Phi}$ 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 $RMSE_{\Phi}$ and SSIM values vary widely and to the straight-ray approach for which the distribution is narrow and with the worst statistics.

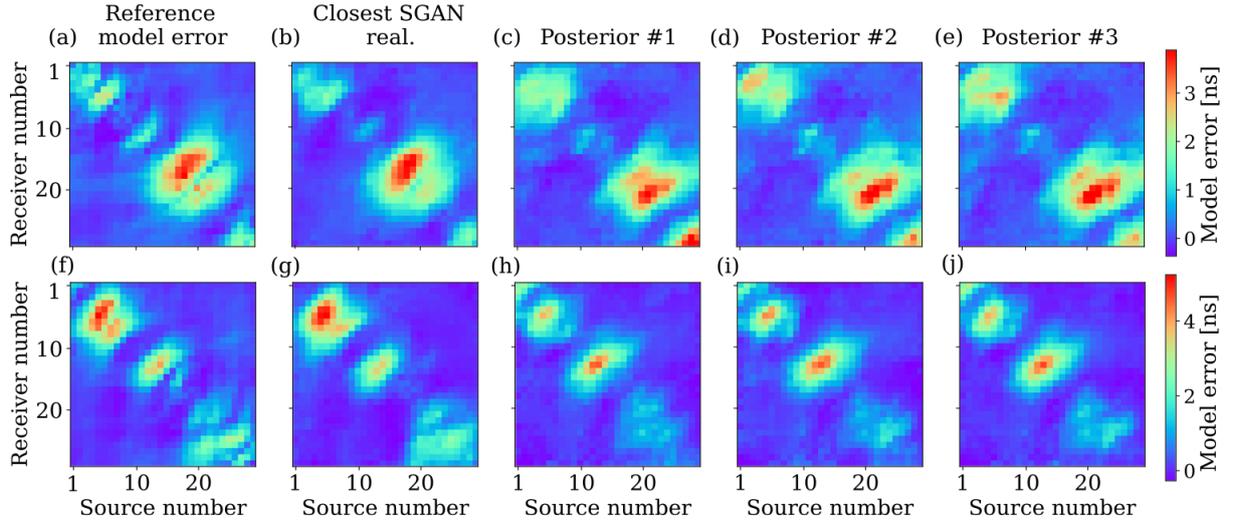


Figure 2.8: Model errors for Test Case 1 ($\eta^{eikonal-SR}$) representing the discrepancy between the eikonal and straight-ray solvers. (a) and (f) are reference model errors calculated based on reference Models 1 and 2 in Figures 2.7a and 2.7f, respectively, (b) and (g) are the corresponding closest 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 2.2: Inversion results for Test Case 1 ($\eta^{eikonal-SR}$) in terms of the subsurface model considering $g^{LF} = g^{SR}$ and $g^{HF} = g^{eikonal}$. The $RMSE_{\Phi}$ and SSIM values are average values of the posterior samples. The $RMSE_{\Phi}$ 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 2.6.2.

| Model | Inv. approach | $RMSE_{\Phi}$ [-] | 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 2.3: 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 2.6.2).

| 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 |

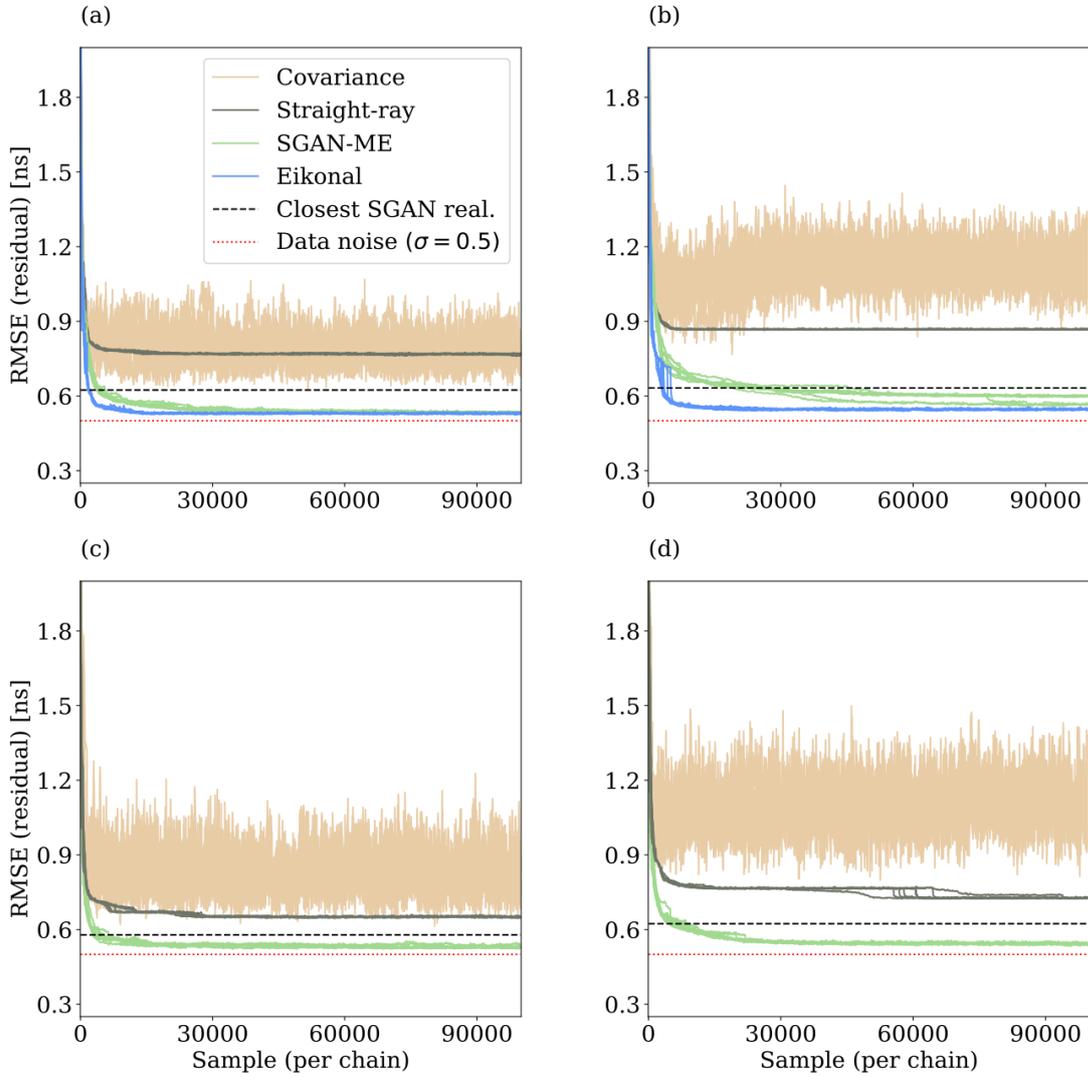


Figure 2.9: Data fit ($\text{RMSE}_{\mathbf{d}}$) for inversion considering: modelling error $\boldsymbol{\eta}^{\text{eikonal-SR}}$ for reference models (a) 1 and (b) 2 and modelling error $\boldsymbol{\eta}^{\text{FDTD-SR}}$ for reference models (c) 1 and (d) 2.

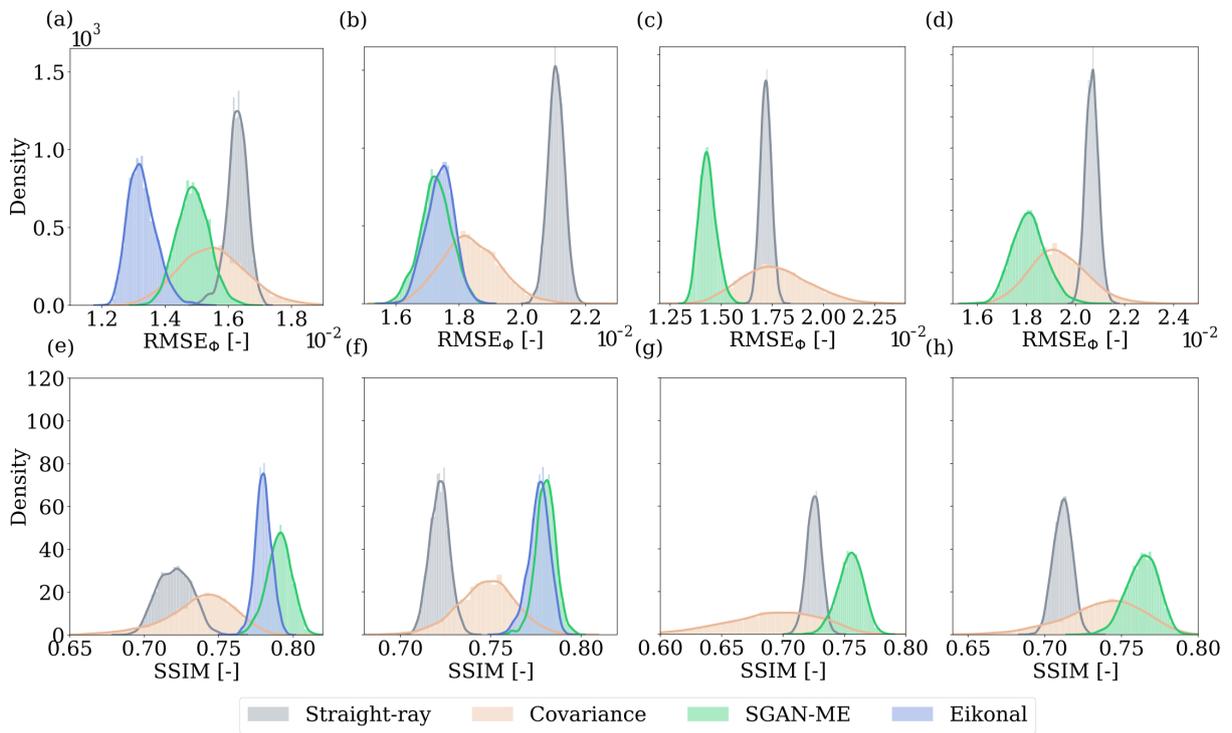


Figure 2.10: $RMSE_{\Phi}$ and SSIM distributions of posterior samples for inversion considering: Test Case 1 ($\eta^{eikonal-SR}$) for reference models (a and e) 1 and (b and f) 2 and Test Case 2 (η^{FTD-SR}) for reference Models (c and g) 1 and (d and h) 2. The high-fidelity solution is only available in Test Case 1 (blue area in a, b, e and f).

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 2.11 and Table 2.4 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.

The maximum-likelihood solution together with the $RMSE_{\Phi}$ and SSIM values in Table 2.4 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 $RMSE_{\Phi}$ 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 2.3). 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. 2.7c and 2.11c). 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 2.12h-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 2.3).

For both reference models, the $RMSE_{\Phi}$ 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 $RMSE_{\Phi}$ decreasing from 0.015 to 0.014 while the SSIM value decreases from 0.79 to 0.75. This improvement in $RMSE_{\Phi}$ 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. 2.8b and 2.8g for $\boldsymbol{\eta}^{eikonal-SR}$ and Figs. 2.12b and 2.12g 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 2.2 and 2.4 and Figure 2.9 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 $RMSE_{\Phi}$ and SSIM distribution (Figure 2.10c,d,g,h) show similar patterns as for Test Case 1, in the sense that the SGAN-ME approach generally results in $RMSE_{\Phi}$ 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

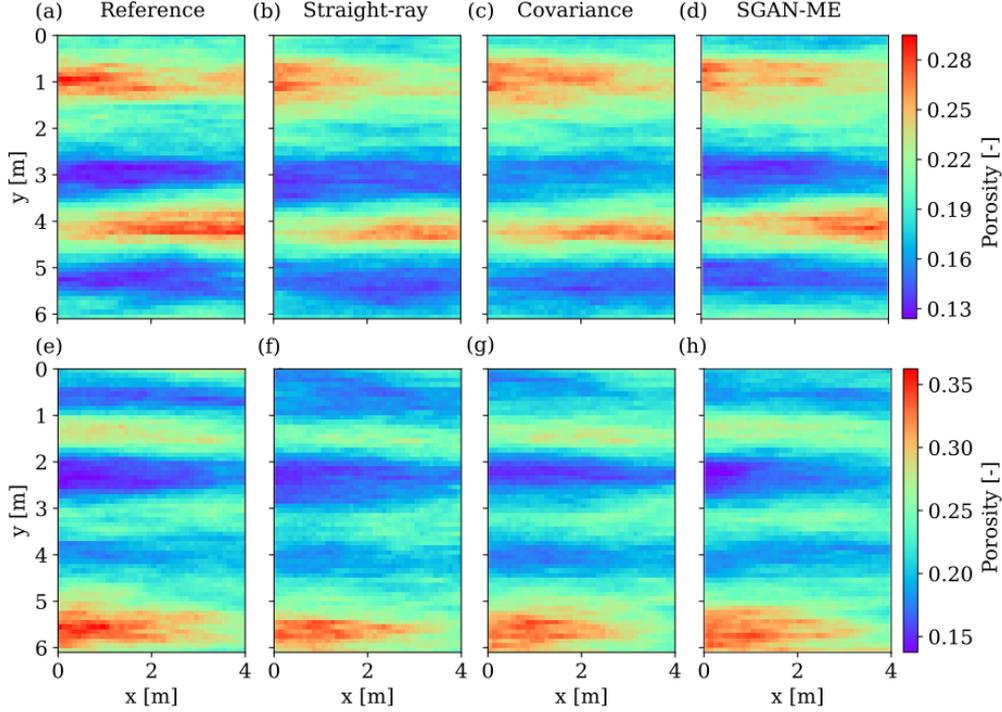


Figure 2.11: Inversion results for reference models (a) 1 and (e) 2 for Test Case 2 ($\eta^{FDTD-SR}$). (b)-(d) and (f)-(h) are the maximum-likelihood realizations obtained from inversion using the straight-ray, covariance and SGAN-ME approaches. All three approaches use the straight-ray solver for the forward response during inversion, while the observed data were created using the FDTD solver.

for which $RMSE_{\Phi}$ and SSIM values vary widely and to the straight-ray approach for which the distribution is narrow and exhibits the worst statistics.

2.4 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. 2.8 and 2.12). Among the considered inversion methods employing a low-fidelity forward solver, the SGAN-ME inversion scored $RMSE(\Phi)$ 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 2.2). 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. 2.7b, 2.7g, 2.11b and 2.11f). 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

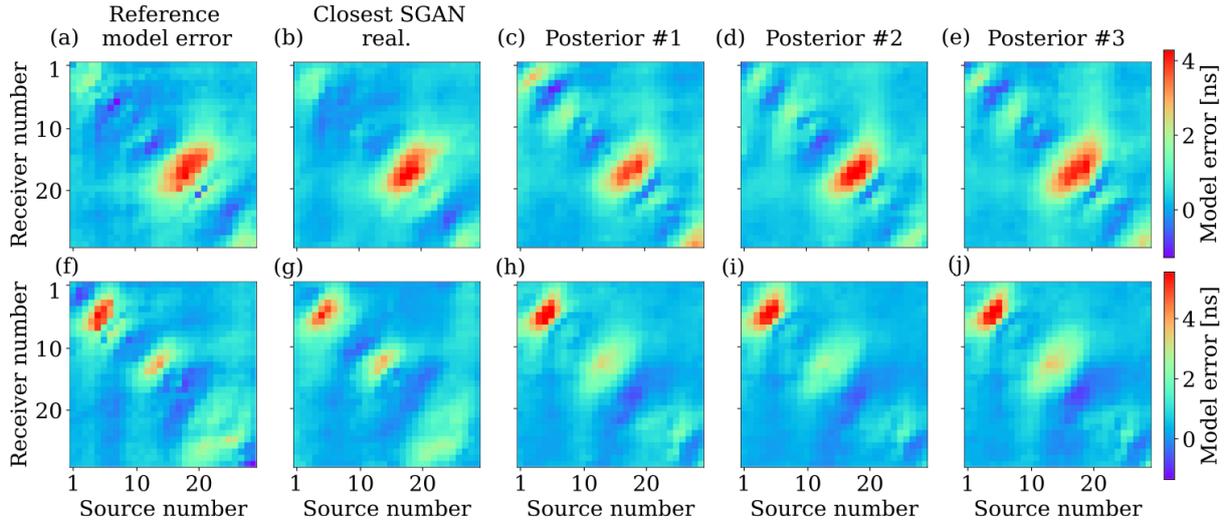


Figure 2.12: Model errors for Test Case 2 ($\eta^{FDTD-SR}$) representing the discrepancy between the FDTD and straight-ray solvers. (a) and (f) are reference model errors calculated based on reference models 1 and 2 in Figures 2.11a and 2.11e, respectively, (b) and (g) are the corresponding closest 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.

Table 2.4: Inversion results for Test Case 2 ($\eta^{FDTD-SR}$) in terms of the subsurface model considering $g^{LF} = g^{SR}$ and $g^{HF} = g^{FDTD}$. The $RMSE_{\Phi}$ and SSIM values are average values of the posterior samples. The $RMSE_{\Phi}$ 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 2.6.2.

| Model | Inv. approach | $RMSE_{\Phi}$ [-] | SSIM [-] | $RMSE_{\mathbf{d}}$ [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 |

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. 2.8 and 2.12) ranging between 71 and 74% recovery of the true model error in the $\boldsymbol{\eta}^{FDTD-SR}$ Test case (Table 2.3). 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 2.8c-e) or underestimated (Figure 2.8h-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 2.2 and 2.4 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.

The $RMSE_d$ values corresponding to SGAN-ME are very similar to those obtained when using the high-fidelity eikonal solver (Table 2.2). 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 2.2 and 2.4). 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 $RMSE_d$ 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 2.10, where the covariance-based $RMSE_\Phi$ and SSIM distributions exhibit the largest variance. The straight-ray approach spans a smaller range of posterior realizations, but those present poor $RMSE_\Phi$ and SSIM scores. In that regard, the SGAN-ME presents a combination of small uncertainty (intermediate posterior widths) and the best $RMSE_\Phi$ 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.

2.5 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.

2.6 Appendix

2.6.1 Details on SGAN Architecture and training

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

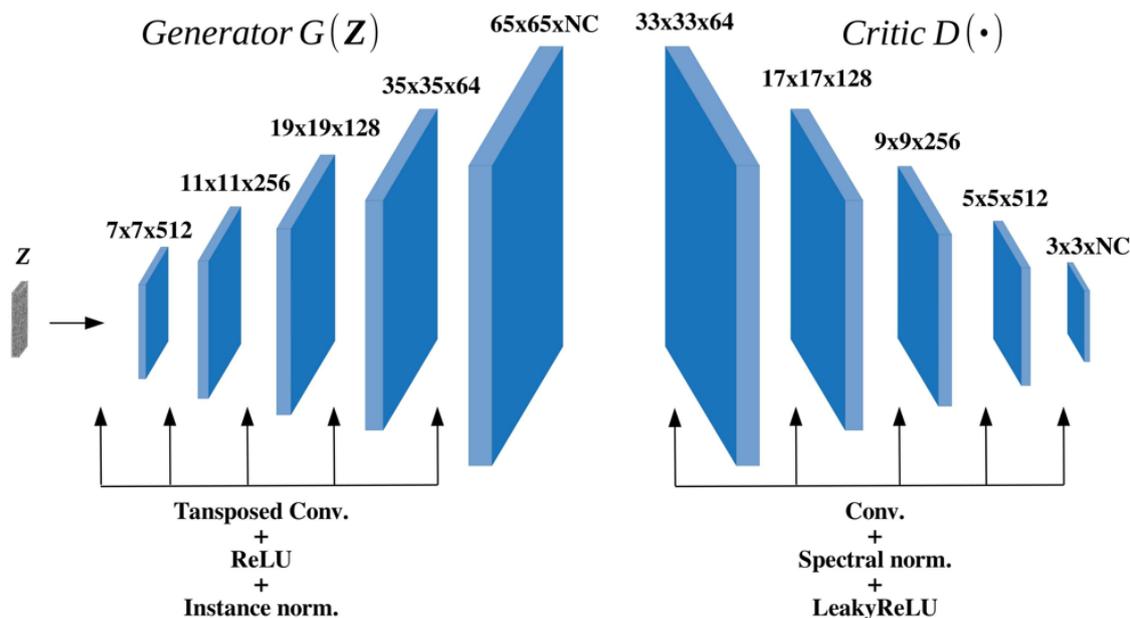


Figure 2.13: SGAN architecture showing the activation and normalization types and output size after each convolution (/transposed convolution) with NC being the number of image channels (e.g. three channels in RGB images). When training over model errors the critic layers include mean-spectral-normalization as opposed to spectral normalization alone for subsurface-model training.

Network architecture

Figure 2.13 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 2.5 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.

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. (2.11)). 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.

Table 2.5: 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 |

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 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*):

$$R_l = R_{l-1} + (k_l - 1) \prod_{i=1}^{l-1} s_i, \quad (2.16)$$

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 2.14). As illustrated in Figure 2.14, 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 2.2.3), in our work we limited the generated image size using padding when we increased the number of latent parameters.

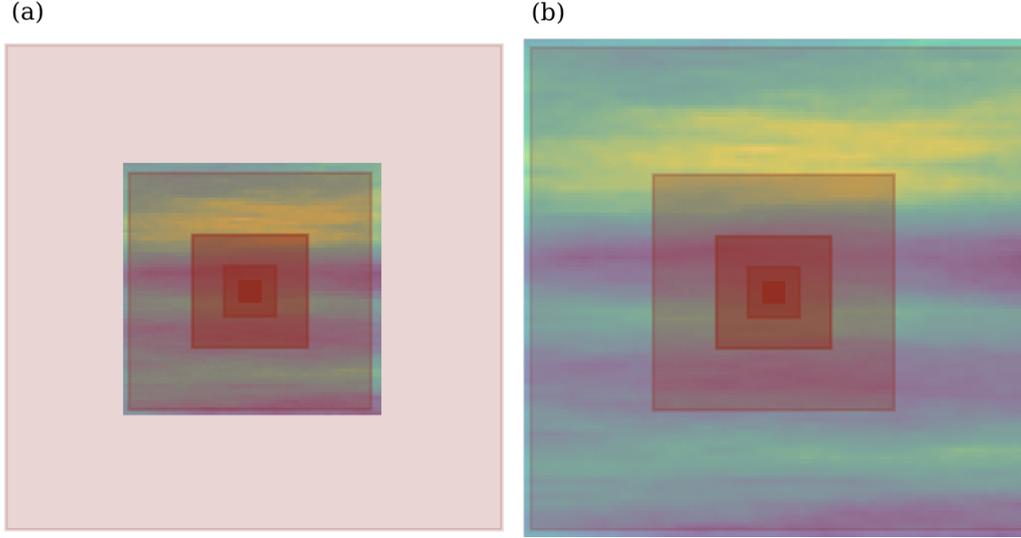


Figure 2.14: (a) Multi-Gaussian TI of dimensions 65x65 pixels and (b) the same TI interpolated into 129x129 pixels, both overlaid by the ERF's of neurons computed for 5 sequential convolutional layers. The ERF is computed given $k = 5$ and $s = 2$ for all layers.

2.6.2 Quality measure calculation

Here we expand the information concerning the quantitative measures appearing in Tables 2.2, 2.3 and 2.4. We use RMSE as a metric for model and data fit. The RMSE of the model ($RMSE_{\Phi}$) is calculated on porosity values of individual posterior realizations (only the last 50% of each chain is considered) with respect to the reference model:

$$RMSE_{\Phi} = \sqrt{\frac{\sum_{n=1}^{N_{\Phi}} (\Phi_{ref} - \Phi_n)^2}{N_{\Phi}}}, \quad (2.17)$$

where N_{Φ} is the number of subsurface model parameters. The final reported $RMSE_{\Phi}$ 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

$$RMSE_d = \sqrt{\frac{\sum_{n=1}^{N_d} (\mathbf{d} - \mathbf{d}_n^{sim})^2}{N_d}}, \quad (2.18)$$

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,

$$SSIM(\mathbf{u}, \mathbf{v}) = \frac{(2\mu_{\mathbf{u}}\mu_{\mathbf{v}} + C_1)(2\sigma_{\mathbf{uv}} + C_2)}{(2\mu_{\mathbf{u}}^2 + \mu_{\mathbf{v}}^2 + C_1)(2\sigma_{\mathbf{u}}^2 + \sigma_{\mathbf{v}}^2 + C_2)}, \quad (2.19)$$

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 ($MSE(\boldsymbol{\eta}, \boldsymbol{\eta}_{ref})$):

$$MSE(\boldsymbol{\eta}_{ref}, 0) = \frac{\sum_{n=1}^{N_{\boldsymbol{\eta}}} (0 - \boldsymbol{\eta}_{ref,n})^2}{N_{\boldsymbol{\eta}}}, \quad (2.20)$$

$$MSE(\boldsymbol{\eta}_{app}, \boldsymbol{\eta}_{ref}) = \frac{\sum_{n=1}^{N_{\boldsymbol{\eta}}} (\boldsymbol{\eta}_{ref} - \boldsymbol{\eta}_{app,n})^2}{N_{\boldsymbol{\eta}}}, \quad (2.21)$$

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:

$$ER = \frac{\overline{MSE(\boldsymbol{\eta}_{app}, \boldsymbol{\eta}_{ref})}}{MSE(\boldsymbol{\eta}_{ref}, 0)} * 100\%. \quad (2.22)$$

Chapter 3

Variational Bayesian inference with complex geostatistical priors using inverse autoregressive flows

Shiran Levy, Eric Laloy, Niklas Linde

Published¹ in *Computers & Geosciences* and herein slightly adapted to fit the theme of this thesis.

¹Levy, S., Laloy, E. and Linde, N.(2022). Variational Bayesian inference with complex geostatistical priors using inverse autoregressive flows. *Computers & Geosciences*, **171**, 105263.

Abstract

We combine inverse autoregressive flows (IAF) and variational Bayesian inference (variational Bayes) in the context of geophysical inversion parameterized with deep generative models encoding complex priors. Variational Bayes approximates the unnormalized posterior distribution parametrically within a given family of distributions by solving an optimization problem. Although prone to bias if the chosen family of distributions is too limited, it provides a computationally-efficient approach that scales well to high-dimensional inverse problems. To enhance the expressiveness of the variational distribution, we explore its combination with IAFs that allow samples from a simple base distribution to be pushed forward through a series of invertible transformations onto an approximate posterior. The IAF is learned by maximizing the lower bound of the evidence (marginal likelihood), which is equivalent to minimizing the Kullback-Leibler divergence between the approximation and the target posterior distribution. In our examples, we use either a deep generative adversarial network (GAN) or a variational autoencoder (VAE) to parameterize complex geostatistical priors. Although previous attempts to perform Gauss-Newton inversion in combination with GANs of the same architecture were proven unsuccessful, the trained IAF provides a good reconstruction of channelized subsurface models for both GAN- and VAE-based inversions using synthetic crosshole ground-penetrating-radar data. For the considered examples, the computational cost of our approach is seven times lower than for Markov chain Monte Carlo (MCMC) inversion. Furthermore, the VAE-based approximations in the latent space is in good agreement. The VAE-based inversion requires only one sample to estimate gradients with respect to the IAF parameters at each iteration, while the GAN-based inversions need more samples and the corresponding posterior approximation is less accurate.

3.1 Introduction

Probabilistic inverse modeling is often based on Bayes' theorem:

$$p(\mathbf{m}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{m})p(\mathbf{m})}{p(\mathbf{d})}, \quad (3.1)$$

where \mathbf{m} are unobserved model parameters, \mathbf{d} are the measured data, $p(\mathbf{m}|\mathbf{d})$ is the posterior probability density function (PDF) of interest, $p(\mathbf{m})$ is the prior PDF, $p(\mathbf{d}|\mathbf{m})$ is the likelihood and $p(\mathbf{d}) = \int p(\mathbf{d}|\mathbf{m})p(\mathbf{m})d\mathbf{m}$ is the marginal likelihood that is often referred to as the evidence. The latter is very challenging to estimate, especially for problems of large dimensionality, due to the requirement of integrating the likelihood over the prior of all possible model parameters \mathbf{m} . Markov chain Monte Carlo (MCMC) methods circumvent this problem of evidence estimation by making model proposals using formalized rules and comparing posterior probability ratios, thereby, enabling unbiased sampling from $p(\mathbf{m}|\mathbf{d})$ provided that the MCMC chain(s) are long enough (*Robert et al.*, 1999). In practice, MCMC methods can incur prohibitive computational costs for many problems encountered in the geosciences.

Variational inference (VI; *Blei et al.*, 2017) or its Bayesian variant termed variational Bayes (VB; *Kingma and Welling*, 2014) provides an attractive alternative to MCMC methods as it replaces a sampling problem with an optimization problem. It proceeds by approximating the posterior PDF of interest using a surrogate distribution referred to as the variational density, which is adjusted such that the evidence lower bound (ELBO, see section 3.2.2) is maximized. The variational density belongs to a family of distributions from which it inherits its parameterization. The approximation resulting from VI is limited by the chosen parametric family of distributions. For instance, a classical choice is to use a Gaussian distribution with unknown hyperparameters, which often offers a poor approximation.

Various variational techniques involving intermediate invertible transformations have been developed to allow for more expressive variational densities. Automatic differential variational inference (ADVI; *Kucukelbir et al.*, 2017), for instance, attempts to accommodate different probabilistic models by transforming the original latent space of the model into an unconstrained real-valued space, serving as a "common space". VI is then performed on the common space and differentiation is performed with respect to the original latent space. This approach offers an automatic, comfortable and efficient way to perform VI for a variety of models. Nevertheless, there are several limitations to ADVI: (1) the approximation might suffer from bias due to implicit Gaussian approximations, (2) the approximation is sensitive to the choice of the invertible transformation connecting the variational density in the real-valued space and the original space and (3) it might not be suitable when the posterior is multi-modal (*Kucukelbir et al.*, 2017; *Zhang and Curtis*, 2020a; *Zhao et al.*, 2022). Another approach that has seen multiple applications in geophysics (*Zhang and Curtis*, 2020b; *Ramgraber et al.*, 2021; *Zhang and Curtis*, 2021) is Stein variational gradient descent (SVGD; *Liu and Wang*, 2016). It uses an ensemble of particles, initialized from a base analytical distribution, that are iteratively updated to approximate the posterior using a smooth transformation describing an incremental perturbation. In each step, particles are updated via perturbations in the direction of the steepest descent, where the magnitude and direction of perturbations are determined based on the Stein operator (more specifically Stein's identity and kernelized Stein discrepancy), to minimize the Kullback-Leibler divergence (D_{KL} , *Kullback and Leibler*, 1951) between the current distribution of the particles and a target distribution. An advantage of SVGD is that it does not require explicit parameterization. However, SVGD underestimates the variance as the dimensionality of the problem increases and, therefore, performs poorly on high-dimensional problems (*Ba et al.*, 2019). *Ba et al.* (2019) argue that accurate estimates using SVGD could be obtained by either increasing the number of particles, but this comes at a high computational cost and might not always be practical for high-dimensional problems, or by introducing re-sampling to avoid deterministic bias.

In this study, we consider the increasingly popular family of transformations referred to as normalizing flows (*Rezende and Mohamed*, 2015; *Papamakarios et al.*, 2021; *Kobyzev et al.*, 2021). Normalizing flows transform an initial density of random variables into a target density of richer form through a series of invertible, differentiable and volume conserving maps. Their combination with VI enables a more flexible and scalable approach allowing for approximate posterior distributions of high complexity (*Rezende and Mohamed*, 2015; *Kingma et al.*, 2016). Some example applications are flow-based generative models (*Dinh et al.*, 2016, 2014; *Kingma and Dhariwal*, 2018), inference, reparameterization and representation learning

(Papamakarios *et al.*, 2021 and references therein). In a geophysical context, Zhao *et al.* (2022) assessed normalizing flows expressed by neural networks on two tomographic problems and found that it can significantly reduce the number of forward evaluations needed to reach a solution compared to SVGD and MCMC, while at the same time being less biased than ADVI. However, the authors indicate a possible drawback when training the neural network for high-dimensional problems, for example, in 3D problems. This motivates our work which seeks to combine such approaches with dimensionality reduction.

One of the most popular techniques to reduce dimensionality is principal component analysis (PCA; Wold *et al.*, 1987), although a plethora of other methods exist (e.g. Kernel-PCA, linear discriminant analysis and deep neural networks; Dejtrakulwong *et al.*, 2012; Konaté *et al.*, 2015; Hinton and Salakhutdinov, 2006). For example, Urozayev *et al.* (2021) used VB to infer the low-dimensional latent variables describing the coefficients of a discrete cosine transform (DCT) in a seismic imaging problem. By reducing the dimensionality and using VB, they could reduce the computational complexity and ensure that geologically-meaningful solutions were obtained. Laloy *et al.* (2017) and Laloy *et al.* (2018) showed that deep generative neural networks, such as variational autoencoders (VAEs) or generative adversarial networks (GANs), are well-suited for dimensionality reduction when working on inverse problems with complex prior models. Such methods allow for fast sampling from the prior and the reduction in dimensionality makes MCMC inversions more efficient compared to alternative approaches relying on a training image (TI) such as sequential geostatistical resampling (Mariethoz *et al.*, 2010b; Hansen *et al.*, 2012; Tahmasebi, 2018).

Generally speaking, there are two ways in which generative neural networks can be used in inverse modelling. In the first approach, a pre-trained generative network is combined with an inference framework (e.g. MCMC). In the second approach, the generative network serves as the inference network that is trained to generate realizations that honor the data (Dupont *et al.*, 2018; Mosser *et al.*, 2018; Song *et al.*, 2021b,c; Laloy *et al.*, 2021). The first approach can be further split into two sub-approaches: 1) The distribution conditional on the data is explored in the latent space of the generative network by sampling, minimization or optimization methods (Laloy *et al.*, 2017, 2019; Mosser *et al.*, 2020; Levy *et al.*, 2022) and 2) a mapping is learned between an initial simple distribution and a distribution on the latent space of the generative network which is conditioned on data and from which we can sample conditional realizations (Chan and Elsheikh, 2019). Here we study this latter sub-approach for inversion and build on previous works on normalizing flows and VI (Rezende and Mohamed, 2015; Kingma *et al.*, 2016; Hoffman *et al.*, 2019). We train inverse autoregressive flows (IAF; Kingma *et al.*, 2016), a type of normalizing flows, using stochastic variational inference (SVI; Hoffman *et al.*, 2013) to invert synthetic, noise contaminated (indirect) geophysical data in presence of a complex geostatistical prior. We refer to this approach as neural-transport (Hoffman *et al.*, 2019). Our model parameters are parameterized within the latent space of a deep generative model (DGM): either a GAN or a VAE. Training of the IAF proceeds by randomly drawing samples from a standard normal distribution and pushing them through the IAF transform into a space in which VI is performed. The parameters of the IAF are updated at each training iteration using stochastic gradient-based optimization with the objective to maximize the ELBO.

For the same type of subsurface models as considered herein, *Laloy et al. (2019)* attempted to infer the latent parameters of a GAN using two different deterministic gradient-based inversion approaches. They found that even when a linear forward solver was used, both approaches performed poorly given the high non-linearity of the GAN. Their conclusion was later reinforced by *Lopez-Alvis et al. (2021)* who suggested to replace the GAN with a VAE, for which they obtained better inversion results. This is because the VAE generator was found to be less nonlinear and to better preserve topology compared to the GAN generator (see *Lopez-Alvis et al., 2021*, for details). As neural-transport is a stochastic approach that relies on gradient-based optimization, we expect it to perform better than deterministic gradient-based approaches and, thereby, at least partly avoid pitfalls due to the non-linearity and complex manifold topology of the GAN. Additionally, neural-transport may offer a potentially-significant speedup compared to MCMC given that (1) it allows parallelization of the problem making it well suited to high-dimensional problems and (2) it solves an optimization problem using gradient-based information. The objective of this study is to assess the performance of the neural-transport approach with respect to using either a GAN or a VAE and compare its performance against MCMC results.

The remainder of the paper is structured as follows. Section 2 briefly describes the theory behind each component of the methodology namely, the used DGMs, IAF, VI and the combined neural-transport routine. In section 3, a sensitivity analysis with respect to training and algorithmic parameters is presented. Section 4 presents inversion results obtained from neural-transport and a comparison of neural-transport against MCMC. Section 5 discusses the results, advantages and limitations of neural-transport and outlines possible future developments. Section 6 concludes the study.

3.2 Methods

We build upon the work of *Hoffman et al. (2019)* who coined the term neural-transport (NT) to describe the trained IAF transformation. Compared to previous work with NT, here we consider an intermediate latent space of a DGM: either a SGAN or a VAE. The IAF (section 3.2.1) serves as an inference network in which samples from a standard normal distribution are mapped into a target distribution. The IAF is trained through variational Bayesian inference (section 3.2.2), in which the parameters of the transformation are iteratively updated through gradient-based optimization. The inferred model parameters are those within the latent space of the DGM (section 3.2.3) while the physical forward response (section 3.2.4) is computed on high-dimensional model realizations following the DGM transformation. Finally, the resulting approximate posterior distribution, that is conditioned on indirect (noise-contaminated) geophysical data, can be sampled and estimated. We describe the implementation of this approach, combining NT and DGMs, in section 3.2.5. To assess the quality of the IAF approximation we use several metrics (section 3.2.6) such as the root-mean-squared error and structural similarity index and compare with results obtained by MCMC inversion.

3.2.1 Inverse autoregressive flows

IAF is a class of normalizing flows, in which a random variable $\mathbf{z}^{(0)}$ drawn from a known probability density function (base distribution) $\mathbf{z}^{(0)} \sim q(\mathbf{z}^{(0)})$ is mapped into a random variable \mathbf{m} from the target distribution $\mathbf{m} \sim q(\mathbf{m})$. Given a transformation $\mathbf{m} = f(\mathbf{z}^{(0)})$ where $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is an invertible, continuous and differentiable mapping between two random variables, one can sample from the target distribution by applying the transformation and evaluating the target distribution using the change of variables theorem

$$q(\mathbf{m}) = q(\mathbf{z}^{(0)}) \left| \det \frac{df(\mathbf{z}^{(0)})}{d\mathbf{z}^{(0)}} \right|^{-1}, \quad (3.2)$$

where $\frac{df(\mathbf{z}^{(0)})}{d\mathbf{z}^{(0)}}$ is the Jacobian matrix \mathbf{J} and $\det \frac{df(\mathbf{z}^{(0)})}{d\mathbf{z}^{(0)}}$ its determinant, representing the change in volume as a result of the transformation f . If the mapping consists of several transformations, the logarithmic form of $q(\mathbf{m})$ can be evaluated by:

$$\log q(\mathbf{m}) = \log q(\mathbf{z}^{(0)}) - \sum_{k=1}^K \log \left| \det \frac{df_k(\mathbf{z}^{(k-1)})}{d\mathbf{z}^{(k-1)}} \right|, \quad (3.3)$$

where $k = 1, \dots, K$ is the number of sequential transformations and $\mathbf{m} = \mathbf{z}^{(K)}$. In IAF, the transformation f_k applied on the random variable $z_i^{(k-1)}$ ($i = 1, 2, \dots, n$) is conditional on previous instances and can be formulated as:

$$z_i^{(k)} \sim q(z_i^{(k)} | \mathbf{z}_{1:i-1}^{(k-1)}) = f_k(z_i^{(k-1)}) = z_i^{(k-1)} \odot \sigma_{\phi, i}(\mathbf{z}_{1:i-1}^{(k-1)}) + \mu_{\phi, i}(\mathbf{z}_{1:i-1}^{(k-1)}), \quad (3.4)$$

where ϕ are the trainable parameters of the IAF and σ and μ are the scale and shift functions, respectively, conditional on previous instances. For this type of transformation $|\det \mathbf{J}_k| = \prod_i^n \sigma_i$, making the determinant of the Jacobian easy to compute and the target distribution easier to evaluate. Since $z_i^{(k)}$ only depends on known variables $\mathbf{z}_{1:i}^{(k-1)}$, the mapping can be computed in parallel.

3.2.2 Variational Bayesian inference

Variational Bayes is an approach to approximate an intractable posterior distribution by optimization. The approximation of $p(\mathbf{m}|\mathbf{d})$ is made with a surrogate distribution $q^*(\mathbf{m})$ defined within a family \mathcal{Q} , for which:

$$q^*(\mathbf{m}) = \arg \min_{q(\mathbf{m}) \in \mathcal{Q}} D_{KL}(q(\mathbf{m}) || p(\mathbf{m}|\mathbf{d})). \quad (3.5)$$

The notation D_{KL} in eq. (3.5) indicates the Kullback-Leibler divergence (KL; *Kullback and Leibler, 1951*), a statistical measure of the distance between two distributions defined as $D_{KL}(f(x)||g(x)) = \int f(x) \log(\frac{f(x)}{g(x)}) dx$ for a given random variable x . We define ϕ as the parameterization of the variational density $q(\mathbf{m})$ and it depends on our choice of the distribution family \mathcal{Q} . Since the posterior distribution $p(\mathbf{m}|\mathbf{d})$ is intractable in most cases and the evidence $p(\mathbf{d})$ is a constant, a common approach is to instead maximize the evidence lower bound (ELBO; see *Blei et al., 2017*)

$$\log p(\mathbf{d}) = \underbrace{\mathbb{E}_{\mathbf{m} \sim q} \log p(\mathbf{m}, \mathbf{d}) - \mathbb{E}_{\mathbf{m} \sim q} \log q_\phi(\mathbf{m})}_{ELBO} + D_{KL}(q_\phi(\mathbf{m})||p(\mathbf{m}|\mathbf{d})). \quad (3.6)$$

The name "evidence lower bound" comes from the fact that $\log \mathbb{E}_{q(x)} p(x) \geq \mathbb{E}_{q(x)} \log p(x)$ and that $D_{KL}(q(\mathbf{m})||p(\mathbf{m}|\mathbf{d})) \geq 0$, resulting in the following inequality (*Jordan et al., 1999*):

$$\log p(\mathbf{d}) \geq \mathbb{E}_{\mathbf{m} \sim q} \log p(\mathbf{m}, \mathbf{d}) - \mathbb{E}_{\mathbf{m} \sim q} \log q(\mathbf{m}) = ELBO. \quad (3.7)$$

As we maximize the ELBO in eq. (3.6), it approaches $\log p(\mathbf{d})$. We define a corresponding loss function $\mathcal{L}(\phi) = ELBO$ which depends on the parameterization ϕ of the variational density

$$\begin{aligned} \mathcal{L}(\phi) &= \int q_\phi(\mathbf{m}) \log p(\mathbf{m}, \mathbf{d}) d\mathbf{m} - \int q_\phi(\mathbf{m}) \log q_\phi(\mathbf{m}) d\mathbf{m} \\ &= \int q_\phi(\mathbf{m}) \log \frac{p(\mathbf{m}, \mathbf{d})}{q_\phi(\mathbf{m})} d\mathbf{m} = \mathbb{E}_{\mathbf{m} \sim q} \left[\log \frac{p(\mathbf{m}, \mathbf{d})}{q_\phi(\mathbf{m})} \right]. \end{aligned} \quad (3.8)$$

Then, ϕ is optimized to maximize $\mathcal{L}(\phi)$ (and as a consequence it also minimizes $D_{KL}(q_\phi(\mathbf{m})||p(\mathbf{m}|\mathbf{d}))$) via gradient-based optimization in which gradients of $\mathcal{L}(\phi)$ are computed with respect to ϕ using samples from $q_\phi(\mathbf{m})$

$$\nabla_\phi \mathcal{L}(\phi) = \mathbb{E}_{\mathbf{m} \sim q} \left[\nabla_\phi \log \frac{p(\mathbf{m}, \mathbf{d})}{q_\phi(\mathbf{m})} \right]. \quad (3.9)$$

An unbiased Monte Carlo estimation of the ELBO (and its derivatives) can be computed by evaluating the logarithmic ratios in eqs. (3.8) and (3.9) at N_s samples from $q_\phi(\mathbf{m})$.

3.2.3 Deep generative models

DGMs are artificial neural networks that are trained to generate data according to an underlying distribution of a dataset of interest. A network is composed of input, hidden and output layers, where the input is our input features, the output is our generated data and hidden layers are intermediate layers connecting the input to the output. The hidden layers are composed of small units (nodes) referred to as neurons. Mathematically, a hidden layer can be formulated as:

$$h(\mathbf{X}) = \varphi(\mathbf{X}^T \mathbf{W} + \mathbf{b}), \quad (3.10)$$

where \mathbf{X} is the input vector to the layer, \mathbf{W} contains the weights connecting input features to individual neurons, \mathbf{b} is a vector of biases and φ is an activation function (sigmoid, tanh, ReLU etc.) introducing non-linearity. In convolutional neural networks (CNNs), such as those used herein, weights in each layer are organized in a series of matrices (kernels) that are convolved with the input to form a series of feature maps. Convolutional networks reduces the number of weights required, especially for large inputs and they are advantageous in tasks where the input exhibits local interactions between features (see *Goodfellow et al., 2016* for more information). In this study, we consider two types of DGMs: spatial generative adversarial network (SGAN) and a variational autoencoder (VAE). These DGMs are introduced to reduce the dimensionality of the inverse problem by learning an encoding of a complex prior, thereby, aiming at reducing the computational cost and improving inversion performance. Both DGMs are trained using the binary channelized image of size 2500×2500 pixels introduced by *Zahner et al. (2016)* and later used by *Laloy et al. (2018)* (Fig. 2a) and *Lopez-Alvis et al. (2021)*.

Spatial generative adversarial networks

The SGAN (*Jetchev et al., 2016; Laloy et al., 2018*) is a type of generative adversarial network (GAN; *Goodfellow et al., 2014*), that is, a CNN consisting of a discriminator D and a generator G . Adversarial training consists of optimization with the generator and discriminator competing against each other. The input to the generator in a SGAN is a noise tensor \mathbf{Z} of 2D or 3D shape, which is typically drawn from a standard normal or uniform distribution. For convenience, in this paper we represent \mathbf{Z} in its vector form \mathbf{z} , however, in practice the input to the generator of the SGAN is a tensor of rank that is higher than one. For a 2D model domain, the output is an image $\tilde{\mathbf{X}}$ of size $m \times n \times q$, where q represent the number of RGB channels. The size of $\tilde{\mathbf{X}}$ is determined by the depth of the network as well as the number of spatial parameters (m and n). The significance of having an input tensor in a SGAN as opposed to a 1D vector in a standard GAN, is the way perturbations in the latent space are translated into changes in the image space. As opposed to a global update, perturbing one of the SGAN's latent parameters leads to a localized change in $\tilde{\mathbf{X}}$. The input to the discriminator is either the generated image $\tilde{\mathbf{X}}$ or an image \mathbf{X} from a training set, containing the patterns we would like to learn. The output of the discriminator is a score of either 0 or 1, representing the belief that the input is either generated by the generator or is a part of the training set (i.e., training image), respectively. The network is trained using the following minimization-maximization loss function:

$$\min_{G(\cdot)} \max_{D(\cdot)} \mathbb{E}_{\mathbf{X} \sim P_r} [\log D(\mathbf{X})] + \mathbb{E}_{\mathbf{z} \sim p_g} [\log(1 - D(G(\mathbf{z})))] \quad (3.11)$$

The discriminator D will try to maximize the function in eq. (3.11) by correctly labeling \mathbf{X} as 1 and $G(\mathbf{z})$ as 0, while the generator G will try to minimize it through fooling the discriminator. For numerical stability, an l_2 -norm regularization $\alpha_{GAN} \|\Omega\|_2^2$ is applied to both the generator and discriminator, where Ω contains the network weights and regularization increases as we increase α_{GAN} , the weighting factor. This type of regularization encourages the individual weights to be small, thus, preventing large weights on a few layer units (neurons). The loss from eq. (3.11) is then used to update the parameters of the discriminator and generator, where the weights of the discriminator are updated first and the weights of the generator are updated in a second stage. The update to the parameters of the network is performed by back-propagating the error computed in the forward pass (going from input to output) through the respective network. The update to each network parameter is proportional to a specified learning rate and the gradients of the error with respect to that parameter (see *Laloy et al. (2018)* for more details).

We adopt the SGAN architecture of *Laloy et al. (2019)* who used a generator with seven convolutional layers, instance normalization and ReLU activation function except for the last layer which is only followed by a tanh activation function. We train the network with a square latent space \mathbf{z} of 12×12 out of which we use only 5×3 (15) latent variables to generate images of size 65×129 . The training images are normalized into a range of $[-1, 1]$ before they are fed into the discriminator. Consequently, when using the trained generator, images are also re-scaled into the $[0, 1]$ range. We train the SGAN with the ADAM optimizer (*Kingma and Ba, 2014*) using a batch of 32 images at each training iteration and the following hyperparameters: $\beta_{GAN,1} = 0.5$, $\beta_{GAN,2} = 0.999$, learning rate of $5e^{-4}$ and $\alpha_{GAN} = 1e^{-7}$.

Variational autoencoders

Variational autoencoders are a type of generative models proposed by *Kingma and Welling (2014)* for various deep learning tasks (e.g. recognition, denoising, representation and visualization) involving intractable posteriors. VAEs include two neural networks: a probabilistic encoder described by $q_\vartheta(\mathbf{z}|\mathbf{X})$ and a probabilistic decoder described by $p_\theta(\mathbf{X}|\mathbf{z})$, where ϑ and θ are the parameters of the encoder and decoder, respectively. The former transforms an input $\mathbf{X}^{(i)}$ from the training set $\{\mathbf{X}_{i=1}^N\}$ into a probabilistic n -dimensional representation \mathbf{z} and the latter samples \mathbf{z} and transform it into $\tilde{\mathbf{X}}^{(i)}$, that is, a reconstruction of $\mathbf{X}^{(i)}$. The training objective is to maximize the ELBO (*Kingma and Welling, 2014*):

$$\mathcal{L}(\vartheta, \theta) = \mathbb{E}_{q_\vartheta(\mathbf{z}|\mathbf{X})} [\log(p_\theta(\mathbf{X}|\mathbf{z}))] - D_{KL}(q_\vartheta(\mathbf{z}|\mathbf{X}) \| p(\mathbf{z})). \quad (3.12)$$

The first term represents the reconstruction error of the decoder when transforming samples from \mathbf{z} into $\tilde{\mathbf{X}}$ while the second term encourages the variational density $q_\vartheta(\mathbf{z}|\mathbf{X})$ to be close to $p(\mathbf{z}) \equiv \mathcal{N}(\mathbf{0}_n, \mathbf{I}_n)$. The model becomes non-differentiable if we sample \mathbf{z} directly from a distribution parameterized by the output of the encoder as we would need to compute the gradients with respect to a random sample. This is problematic for gradient-based

optimization during which gradients are back-propagated through the network. In order to solve this problem, \mathbf{z} is re-parameterized using a random auxiliary noise $\boldsymbol{\varepsilon}$ such that (Kingma and Welling, 2014):

$$\tilde{\mathbf{z}} = \boldsymbol{\mu}_\theta(\mathbf{X}) + \boldsymbol{\sigma}_\theta(\mathbf{X}) \odot \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim p(\boldsymbol{\varepsilon}) \quad (3.13)$$

where \odot denotes an element-wise product and $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\sigma}_\theta$ are mean and standard deviation vectors provided by the encoder. After reparameterization, \mathbf{z} becomes deterministic and gradients can be back-propagated through it. Here we use the VAE proposed by Lopez-Alvis *et al.* (2021) which has the same layer architecture as the SGAN and was trained on the same training images. Although VAE can be fully probabilistic, Lopez-Alvis *et al.* (2021) considered only the mean of the decoder, therefore, making it a deterministic generator $G_\theta(\mathbf{z})$. After training, θ is constant and to generate $\tilde{\mathbf{X}}$ samples, we simply draw samples from $\tilde{\mathbf{z}} \approx \mathcal{N}(\mathbf{0}_n, \mathbf{I}_n)$ and push it through the generator $G_\theta(\mathbf{z})$. Lopez-Alvis *et al.* (2021) discuss the importance of two hyper-parameters that needs to be specified when training the VAE: β_{VAE} which is a weighting factor multiplying the second term in eq. (3.12) and α_{VAE} which controls the distribution from which the auxiliary noise is drawn from: $p(\boldsymbol{\varepsilon}) = \mathcal{N}(\mathbf{0}_n, \alpha_{VAE} \cdot \mathbf{I}_n)$. They illustrate how well-chosen α_{VAE} - and β_{VAE} -values leads to a well-behaved generator and better inversion performance relative to other choices. The hyper-parameters with which the VAE is trained are as follows: $\beta_{VAE} = 1000$, $\alpha_{VAE} = 0.1$ and a learning rate of $1e^{-3}$ (for more details, see Lopez-Alvis *et al.*, 2021). The VAE decoder contains two fully connected layers followed by four convolutional layers that are all followed by instance normalization and ReLU activation function except for the last layer which is only followed by a sigmoid activation function. The latent space \mathbf{z} of the VAE is composed of a vector of 20 parameters corresponding to output images of 65×129 pixels.

3.2.4 Crosshole traveltime tomography

In our test examples, we consider a crosshole ground penetrating radar (GPR) setup in which source and receiver antennas are distributed within two vertically-oriented boreholes. The forward response can be formulated as follows:

$$\mathbf{d} = g(\mathbf{s}) + \boldsymbol{\varepsilon}, \quad (3.14)$$

where g is the forward operator, \mathbf{s} is the slowness field (inverse of velocity \mathbf{v}) of the modelled domain, $\boldsymbol{\varepsilon}$ is the observational noise and $\mathbf{d} = (d_1, \dots, d_N) \in \mathcal{R}^N$ with $N \geq 1$ is the measured first-arrival travel times between source-receiver pairs. We assign velocities to individual model parameters using $v = 0.06 + 0.02 \cdot (1 - \tilde{x})$, resulting in a continuous range of velocities between $[0.06, 0.08]$ m·ns⁻¹.

We consider a non-linear forward solver implemented in the pyGIMLi geophysical modelling library (Rücker *et al.*, 2017). In this implementation, the travel times are calculated on a mesh of nodes based on the Dijkstra's method, giving the shortest path between source-receiver

positions for a given slowness model. We use the Jacobian provided by pyGIMLi for a given source-receiver geometry and slowness model and calculate the travel times according to:

$$\mathbf{d}_{sim} = \mathbf{J}_g(\mathbf{s})\mathbf{s}, \quad (3.15)$$

where \mathbf{J}_g is the Jacobian matrix (also known as the sensitivity matrix) containing the length of the ray segment at each model cell for each travel time. Note that this Jacobian refers to the physical forward solver and not to the Jacobian of the IAF (eq. (3.2)). The accuracy of the simulated travel times can be improved by adding secondary nodes. In our examples, the Jacobian is re-computed for each slowness field using two secondary nodes. With the Jacobian acting as the forward operator (eq. (3.15)) we are able to comply with the automatic differentiation (auto-differentiation) requirements of machine-learning supported Python libraries (e.g., PyTorch and TensorFlow). Unfortunately, pyGIMLi objects are not supporting data storage using pickling which is a requirement when using most parallel computing Python libraries. Given this limitation, in this work the forward simulations are performed in a sequential manner.

3.2.5 Inversion in the latent space of a deep generative model with neural-transport

Our inversion framework combining NT with a DGM is composed of the methods described in previous subsections: IAF, VI and DGMs, where the forward response is required in order to compute the joint probability $p(\mathbf{m}, \mathbf{d})$. Both DGMs define a low-dimensional latent space involving uncorrelated variables with a well-defined prior (standard normal) that we choose to be in agreement with the base distribution of the IAF. As both DGMs are implemented in PyTorch, we use Pyro (Bingham *et al.*, 2019), a library for probabilistic programming built on Python and PyTorch, to train the IAF. In the following, we define $\mathbf{z}^{(0)}$ as a random variable within the latent space of the IAF that is drawn from a standard normal base distribution, we further define our target distribution q_ϕ on the latent space of the SGAN (or VAE) such that $\mathbf{m} = \mathbf{z}^{GAN}$ (or \mathbf{z}^{VAE}) and $\tilde{\mathbf{X}}$ as the high-dimensional, image-space parameters before slowness \mathbf{s} is assigned. For the remainder of this paper, we will refer to random samples drawn from the base distribution (and pushed forward to the variational distribution space) as particles. Each particle represents one model realization and has the same size as that of the latent space of the DGM in use. For the sake of conciseness, we will refer to variables from the target distribution \mathbf{z}^{GAN} (or \mathbf{z}^{VAE}) simply as \mathbf{z} and specify in the appropriate places to which generative model they belong.

To train the IAF, N_s particles are drawn from the base distribution $\mathbf{z}^{(0)} \sim q(\mathbf{z}^{(0)})$ and mapped through the invertible transformation of the IAF into the variational space $q_\phi(\mathbf{z})$ in which we approximate the posterior on the DGM's latent space \mathbf{z} . The N_s particles \mathbf{z} are then transformed into high-dimensional $\tilde{\mathbf{X}}$ -realizations through the generator $G(\cdot)$. Slowness \mathbf{s} is assigned to each pixel and the likelihood is computed for each of the N_s particles using the geophysical forward solver. We compute the logarithm of the joint distribution $\log p(\mathbf{z}, \mathbf{d}) = \log p(\mathbf{d}|\mathbf{z}) + \log p(\mathbf{z})$ (also referred to as the logarithmic form of the unnormalized posterior

$p(\mathbf{z}|\mathbf{d})$). Since we have a standard-normal prior (mean $\mu_z = 0$ and standard deviation $\sigma_z = 1$) on both the SGAN and VAE latent spaces, and further assume independent, identical and normally-distributed observational noise with zero mean and standard deviation of σ_d , we have

$$\log p(\mathbf{z}, \mathbf{d}) = -\frac{1}{2} \left(N_d \log(2\pi) + 2N_d \log(\sigma_d) + \sigma_d^{-2} \sum_{i=1}^{N_d} [d_i - g_i(G(\mathbf{z}))]^2 + N_z \log(2\pi) + 2N_z \log(\sigma_z) + \sigma_z^{-2} \sum_{i=1}^{N_z} z_i^2 \right), \quad (3.16)$$

where N_d is the number of data observations, N_z is the number of latent \mathbf{z} parameters and z_i is the i^{th} parameter in \mathbf{z} . Note that the log-likelihood is evaluated on forward simulations based on the high-dimensional $\tilde{\mathbf{X}}$ -space while the log-prior is evaluated on the low-dimensional SGAN (or VAE) latent parameters. The loss function $\mathcal{L}(\phi)$ can be calculated by using eq. (3.3) to evaluate $\log q_\phi(\mathbf{z})$:

$$\mathcal{L}(\phi) = \mathbb{E}_{\mathbf{z} \sim q} \left[\log \frac{p(\mathbf{z}, \mathbf{d})}{q_\phi(\mathbf{z})} \right] = \mathbb{E}_{\mathbf{z} \sim q} \left[\log \frac{p(\mathbf{z}, \mathbf{d})}{q(\mathbf{z}^{(0)}) \prod_{k=1}^K \left| \det \frac{df_k(\mathbf{z}^{(k-1)})}{d\mathbf{z}^{(k-1)}} \right|^{-1}} \right]. \quad (3.17)$$

The gradient of $\mathcal{L}(\phi)$ is computed through auto-differentiation. We consider $-\mathcal{L}(\phi)$ and perform stochastic gradient descent to update ϕ . A brief summary of the above routine appears in Figure 3.1 and Algorithm 2.

The architecture of the IAF can be adjusted in response to the level of complexity of the target distribution. *Hoffman et al.* (2019) used three stacked flows with two hidden layers each. We found that two sequential flows, each containing one hidden layer and a hidden dimensionality that is twice as large as the target distribution to be sufficient for our considered examples. Each flow is followed by a non-linear ReLU activation function providing the network with further flexibility (For detailed information about the architecture of the IAF see Appendix 3.6.1). During training the network parameters are optimized using ADAM with $\beta_{IAF,1} = 0.9$ and $\beta_{IAF,2} = 0.999$ and a learning rate of 0.01. In order to enable gradient calculation of the model parameters with respect to the DGM as part of the NT routine, we do not threshold the generated images to $[0, 1]$ (*Laloy et al.*, 2019).

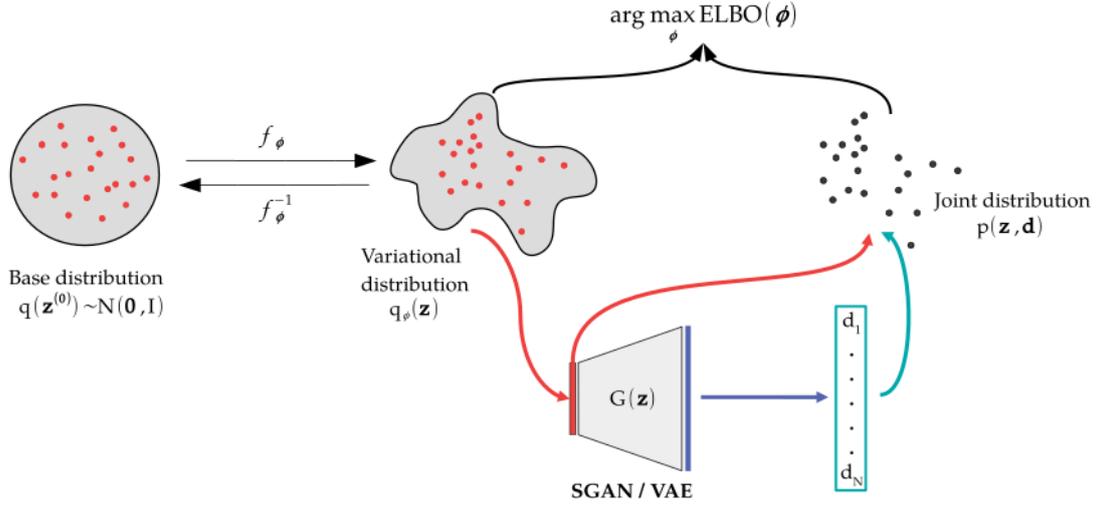


Figure 3.1: Illustration of one training iteration of neural-transport combined with deep generative models. The inferred model parameters (here represented in 2D as dots) are the low-dimensional latent parameters \mathbf{z} . The pre-trained generator G transform the latent parameters into their corresponding high-dimensional parameters in the image-space on which forward simulations are carried out to obtain the data vector \mathbf{d} . The IAF (represented by transformation f) parameters ϕ , are tuned during training to maximize the ELBO.

3.2.6 Performance assessment

We test NT in combination with each of the two considered DGMs using five different test models (Fig. 3.2) generated by the respective generator. Following *Lopez-Alvis et al. (2021)*, we refer to models generated by the SGAN with the abbreviation 'mg' and models generated by the VAE with 'mv'. It is seen that the SGAN provides images that are less blurry than those produced by the VAE. To assess the performance and quality of the approximate posterior $q_\phi(\mathbf{z})$ obtained from NT, we consider different statistical metrics. For each test model, we plot the mean and standard deviation image of the approximate posterior. The root-mean-squared error (RMSE) is computed on mean values of the latent parameters (RMSE_z), model parameters (RMSE_x) and data misfit (RMSE_d) at the last iteration. We rely on RMSE_d to determine if the approximate posterior has converged. The loss function for the IAF \mathcal{L} is used here as a complementary metric as we observed that the data fit can decrease even after \mathcal{L} becomes stable. We define two criteria that both need to be met to declare convergence: (1) for an iteration after which the RMSE_d value averaged over all particles equals that in the last 10% iterations of the algorithm; (2) the average $\text{WRMSE} = \sqrt{\frac{1}{N_d} \sum \left[\frac{d_i - g_i(G(\mathbf{z}))}{\sigma_i} \right]^2}$ (data misfit weighted by the standard deviation of the data noise) is less than 1.1. The former criterion ensures that the approximate posterior reached a stable solution while the latter prevents declaring convergence for models that are stuck in a local minimum with a data misfit that is poor. The similarity of the NT solution (mean value of the approximate posterior) to the true model in the high-dimensional space $\tilde{\mathbf{X}}$ is further assessed using the structural similarity index (SSIM; *Wang et al., 2004*). It measures the similarity between two images with respect to their structural information:

Algorithm 2: Bayesian inference using neural-transport and a deep generative model

```

1 set T (maximum number of iterations) and  $t = 0$ 
2 while ( $t < T$ ) do
3   Draw  $N_s$  particles (realizations of  $\mathbf{z}^{(0)}$ ) from the base distribution  $q(\mathbf{z}^{(0)})$ 
4    $\mathbf{z} \leftarrow \text{IAF}_\phi(\mathbf{z}^{(0)})$ 
5    $\tilde{\mathbf{X}} \leftarrow G(\mathbf{z})$ 
6   Assign slowness values to  $\tilde{\mathbf{X}}$  and compute the forward simulation  $g(\mathbf{s})$  to get simulated
   data  $\mathbf{d}$ 
7   Compute  $\mathcal{L}(\phi)$  and  $\nabla_\phi \mathcal{L}(\phi)$  using eq. (3.9), (3.16) and (3.17) and update  $\phi$  using
   stochastic gradient descent
8    $t = t + 1$ 
9 end
10 Function G( $\mathbf{z}$ )
11   Perform a series of transposed convolution layers with pre-trained weights
12   return  $\tilde{\mathbf{X}}$ 
13 end
14 Function IAF $_\phi(\mathbf{z}^{(0)})$ 
15    $\mathbf{z} = f_k \circ f_{k-1} \circ \dots \circ f_1(\mathbf{z}^{(0)})$ 
16   return  $\mathbf{z}$ 
17 end

```

$$SSIM(\mathbf{u}, \mathbf{v}) = \frac{(2\mu_{\mathbf{u}}\mu_{\mathbf{v}} + C_1)(2\sigma_{\mathbf{uv}} + C_2)}{(2\mu_{\mathbf{u}}^2 + \mu_{\mathbf{v}}^2 + C_1)(2\sigma_{\mathbf{u}}^2 + \sigma_{\mathbf{v}}^2 + C_2)}, \quad (3.18)$$

where \mathbf{u} and \mathbf{v} are sliding windows of size $M \times M$, each sub-samples its respective $[0, 1]$ normalized image. The values of the SSIM range between -1 and 1 , where 1 indicates perfectly matching images. Here we use $M = 7$, $C_1 = 0.01$ and $C_2 = 0.03$ as those values are commonly used (Wang *et al.*, 2004; Laloy *et al.*, 2021 and references therein).

Additionally, we assess the performance of the NT approach against the results obtained by MCMC. We use the differential evolution adaptive Metropolis (DREAM_(ZS)) algorithm (Ter Braak and Vrugt, 2008; Vrugt *et al.*, 2009; Laloy and Vrugt, 2012) to sample the posterior in the latent space of each considered DGM. In this MCMC algorithm, several chains evolve in parallel and jumps are proposed based on candidate points from an archive of past states. At each MCMC step and for each individual chain, a sample \mathbf{z}' proposed according to a symmetric proposal distribution is either accepted or rejected according to a Metropolis acceptance probability

$$p_{acc}(\mathbf{z}^{t-1}, \mathbf{z}') = \min\left(1, \frac{p(\mathbf{d}|\mathbf{z}')p(\mathbf{z}')}{p(\mathbf{d}|\mathbf{z}^{t-1})p(\mathbf{z}^{t-1})}\right). \quad (3.19)$$

If accepted, the chain moves to the proposed state, if rejected it remains at the current state. Convergence of each latent parameter is declared based on the Gelman-Rubin diagnostic (Gelman and Rubin, 1992) when $\hat{R} \leq 1.2$. We compare the approximate posterior PDFs of \mathbf{z} obtained from NT to those obtained from MCMC inversion. Furthermore, both posterior

distributions are also compared to the prior of the latent space. The distance between two PDFs is computed using the KL-divergence while their predictive power is assessed using the logarithmic scoring rule (LogS; *Good*, 1952). The LogS statistic is defined as $\log S(\hat{p}, \mathbf{y}) = -\log \hat{p}(\mathbf{y})$ where \mathbf{y} are the true values of the parameters of interest and \hat{p} is the PDF used to predict the probability of \mathbf{y} . A lower LogS indicates a more accurate prediction. Both the KL-divergence and LogS values are reported as mean values over all parameters.

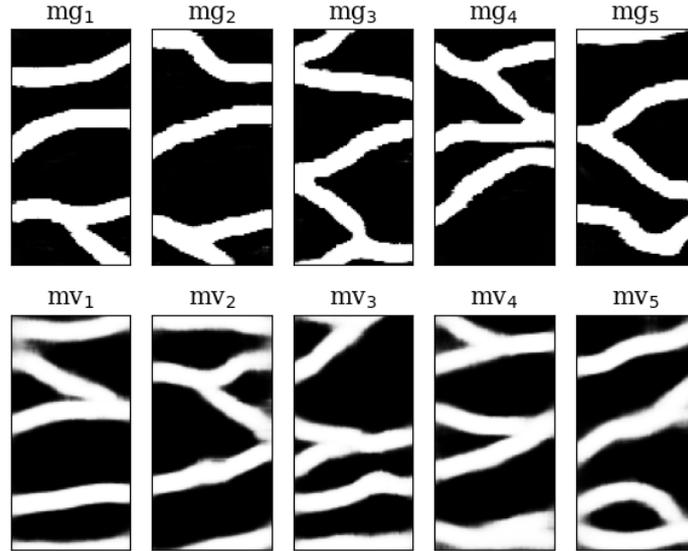


Figure 3.2: Reference models used for inversion. Models labeled with: 'mg' are test models generated by the SGAN and 'mv' are test models generated by the VAE.

3.3 Inversion results

We consider the inversion of synthetic data created using the forward solver described in section 3.2.4 that are contaminated with normally-distributed noise $\mathcal{N}(0, 1)$. We use 25 sources and 25 receivers resulting in 625 data points. Given results from the hyperparameter search described in Appendix 3.6.2, we use 20 particles to perform inversion with the SGAN and only a single particle when using VAE. After 250 iterations with 20 particles, the $RMSE_d$ of NT with SGAN is still decreasing towards the target value (Fig. 3.10a). Consequently, we increase the number of training iterations to 300 (6000 forward simulations) for the SGAN inversions. On the other hand, NT with VAE converges towards the target value in less than 1000 iterations with a single particle (Fig. 3.10d). Since we perform only one forward simulation per iteration we allow a maximum of 2000 training iterations for the VAE inversions. The learning rate in both types of inversions is set to 0.01. We run each NT-inversion scheme on a single CPU (AMD EPYC™ 7402) in a sequential manner due to the inability to distribute the forward response function on multiple CPUs (see section 3.2.4). It takes around 13 hours to run the VAE-based NT inversion considering 2000 training iterations and a single particle and around 40 hours to run the SGAN-based NT for 300 training iterations and 20 particles.

The computational effort is completely dominated by the calculation of the Jacobian of the physical forward solver (eq. (3.15)) at each iteration as it makes up 99% of the total computational time.

Figures 3.3 and 3.4 for SGAN and VAE, respectively, show the mean and standard deviation of the approximate posterior compared to the true model as well as the ELBO loss and $\text{RMSE}_{\mathbf{d}}$ during inversion. These figures are complemented by quantitative metrics in Table 3.1. For all of the mg models in Figure 3.3 that were obtained using SGAN as the DGM, the main features were reconstructed with the right number and location of the channels (Fig. 3.3b) and the largest uncertainty is located at the boundaries of the channels (Fig. 3.3c) as expected given results of previous studies (e.g., Zahner *et al.*, 2016). The inferred models mg_2 and mg_3 are of lower quality compared to the other ones with SSIM values of 0.71 and 0.76, while it is ≥ 0.85 for the other mg models. The inferred model for mg_2 has large uncertainty (around 0.3) on the upper left 2 m of the model and exhibits the largest data misfit (1.42 ns while it is ≤ 1.10 ns for the other models). It is possible that these estimates would have improved further with more training iterations.

The results obtained when using the VAE as DGM are both qualitatively (Fig. 3.4) and quantitatively (Table 3.1) much better. Table 3.1 suggests that all mv models are well reconstructed with all $\text{RMSE}_{\mathbf{d}} \leq 1.05$ ns and all $\text{SSIM} \geq 0.9$. The $\text{RMSE}_{\mathbf{z}}$ values are as low as 0.08 and none is higher than 0.37, indicating a good match between the inferred latent parameters and their true counterparts. These values are at least one order lower than the values obtained for the SGAN-based inversion (≥ 1.08). The $\text{RMSE}_{\mathbf{x}}$ is as low as 0.04 for the VAE models (mv_4) whereas the lowest value for the SGAN models is 0.10 (mg_1). Moreover, the mean standard deviation of \mathbf{X} , while highly dependent on the number of channel elements in the reference model, is consistently lower when the VAE is used as DGM (on average by $\sim 60\%$). Although there are noticeable differences in the bottom part of the inferred mv_1 and mv_2 models (last 2 m) compared to their references, these do not translate into large data misfits as these regions are not well constrained by the GPR rays. Furthermore, when we train the IAF with ten particles on the mv_1 model the reconstruction improves (see Appendix 3.6.3).

Three out of the five mg models and all mv models converged according to the criteria in subsection 3.2.6 (see Table 3.1). For the mg models, convergence occurs after 260 training iterations on average (with 20 particles), while for the mv models the stage at which convergence can be declared ranges between 477 and 1767 training iterations (with a single particle). Nonetheless, it can be seen in Figure 3.4 that by the 800th iteration the $\text{RMSE}_{\mathbf{d}}$ for all the models is below 1.1 ns. This is in agreement with Figure 3.5 where we take three exemplary latent parameters of mv_5 and plot their approximate PDF at different stages of the training process. We observe that after 500 training iterations the approximate density is close to the final density (2000 iterations) and after 1000 iterations the density becomes very similar to the final one for the first (Fig. 3.5a) and tenth (Fig. 3.5b) latent parameters.

After demonstrating that the SGAN- and particularly the VAE-based NT produce high-quality reconstructions of the true model, we assess now the corresponding approximate posteriors with respect to MCMC inversion ($\text{DREAM}_{(Z_S)}$) and the standard normal prior PDF. We run eight parallel MCMC chains for one test model of each DGM: mg_5 and mv_5 (these are also the models resulting in the lowest $\text{RMSE}_{\mathbf{z}}$). We limit the number of samples per chain to

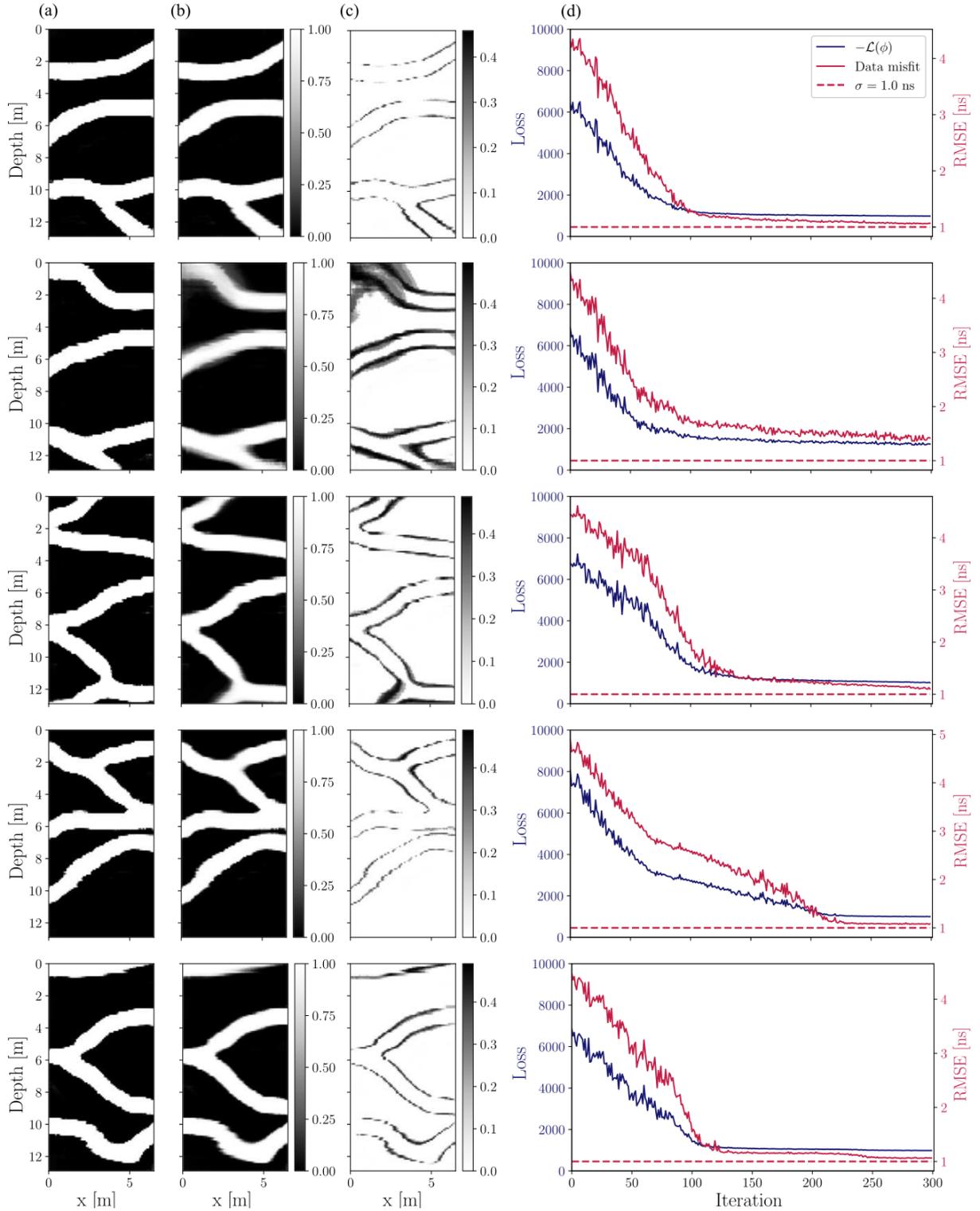


Figure 3.3: Inferred posterior distributions for different reference models generated by the SGAN using 20 particles and 300 training iterations. (a) True mg_1 - mg_5 models (b) mean posterior models obtained from NT and (c) posterior standard deviation in the model image space. (d) The ELBO loss and $RMSE_d$ in ns during training. The $RMSE_d$ curves represent the average values over all particles.

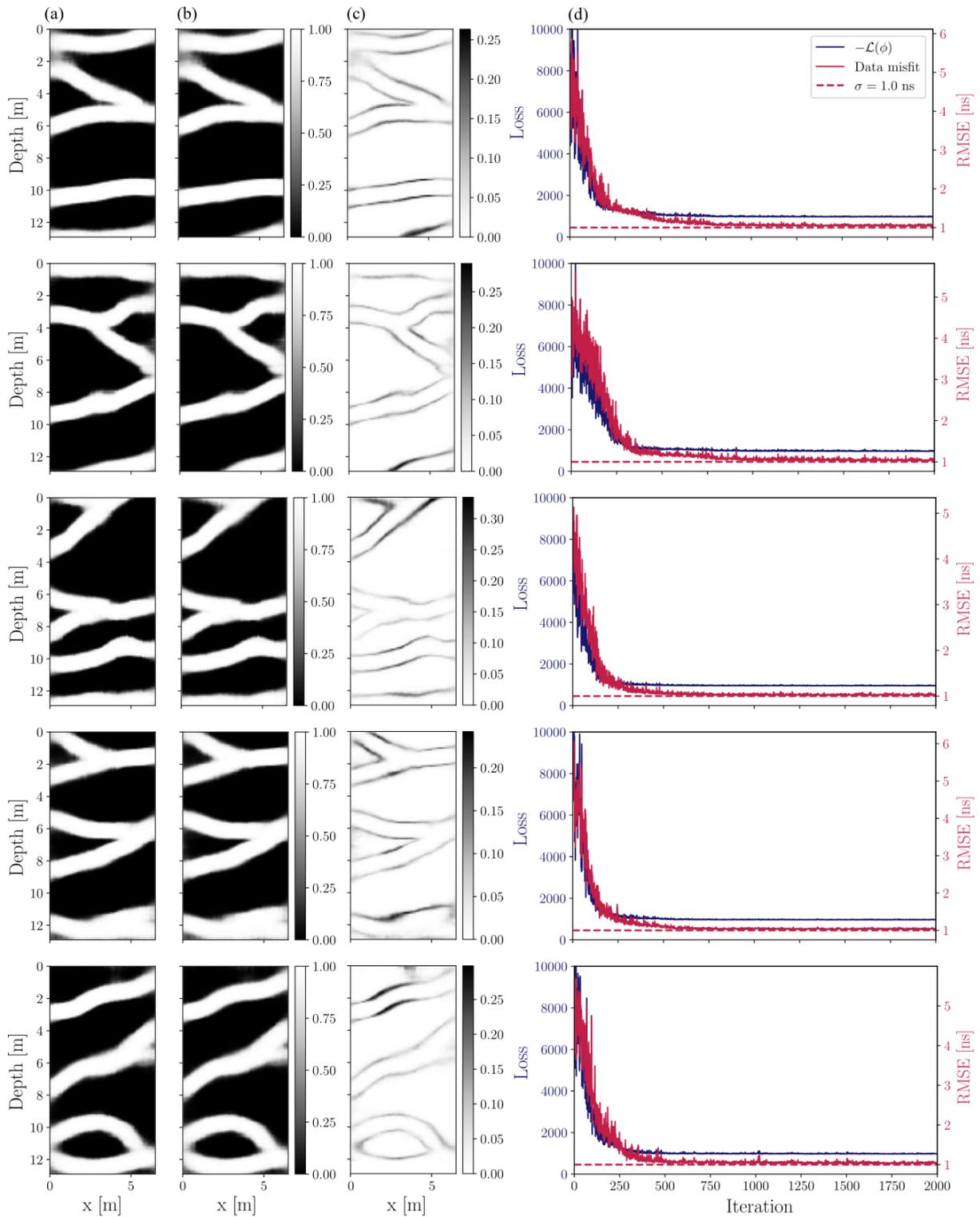


Figure 3.4: Inferred posterior distributions for different reference models generated by the VAE using one particle and 2000 training iterations. (a) True mv_1 - mv_5 models (b) mean posterior models obtained from NT and (c) posterior standard deviation in the model image space. (d) The ELBO loss and $RMSE_d$ in ns during training.

Table 3.1: Summary of the results obtained from inversion with neural-transport for various reference models and using either the SGAN (20 particles) or the VAE (one particle) as DGM. $\bar{S}(\mathbf{X})$ is the average standard deviation of the posterior $p(\mathbf{X}|\mathbf{d}, \mathbf{z})$ corresponding to samples from $q_\phi(\mathbf{z})$. $\text{RMSE}_{\mathbf{z}}$ is calculated on the mean latent space parameters of the resulting posterior while the $\text{RMSE}_{\mathbf{X}}$ is calculated in the model image space. $\text{RMSE}_{\mathbf{d}}$ is the data misfit RMSE value at the last iteration. The SSIM is calculated in the model image space on $[0, 1]$ models.

| DGM | Model | Converged (iteration #) | $\bar{S}(\mathbf{X})$ | RMSE | | | SSIM |
|------|-----------------|----------------------------|-----------------------|--------------|--------------|-------------------|------|
| | | | | \mathbf{z} | \mathbf{X} | \mathbf{d} [ns] | |
| SGAN | mg ₁ | 269 | 0.024 | 1.15 | 0.10 | 1.07 | 0.91 |
| | mg ₂ | - | 0.110 | 1.35 | 0.20 | 1.42 | 0.71 |
| | mg ₃ | - | 0.054 | 1.39 | 0.19 | 1.10 | 0.76 |
| | mg ₄ | 243 | 0.027 | 1.59 | 0.12 | 1.08 | 0.86 |
| | mg ₅ | 271 | 0.040 | 1.08 | 0.12 | 1.06 | 0.85 |
| VAE | mv ₁ | 1767 | 0.020 | 0.37 | 0.12 | 1.05 | 0.94 |
| | mv ₂ | 1467 | 0.017 | 0.18 | 0.06 | 1.04 | 0.96 |
| | mv ₃ | 477 | 0.024 | 0.15 | 0.09 | 1.04 | 0.90 |
| | mv ₄ | 496 | 0.020 | 0.08 | 0.04 | 1.04 | 0.97 |
| | mv ₅ | 1256 | 0.020 | 0.08 | 0.06 | 1.04 | 0.94 |

20 000 which represent a computation time of ~ 6 days on a 8-core workstation. For the SGAN-based MCMC inversion, 12 of the 15 parameters satisfy the \hat{R} criterion within this computational budget. For the VAE-based MCMC inversion, all 20 parameters converged within 8900 samples per chain (total of 71 200 samples). Given the two different convergence criteria for NT and MCMC, we have that the computational time required for the VAE-based MCMC inversion to converge to the posterior target is 7 times larger than that required by the VAE-based NT (56 times if the MCMC algorithm would not have been running in parallel). After 20 000 MCMC samples per chain, the $\text{RMSE}_{\mathbf{z}}$ s of the posterior means are 0.31 and 0.09 for the SGAN- and VAE-based MCMC inversions, respectively. The mean SSIM values in the model space that correspond to those posterior latent parameters are 0.91 (mg₅) and 0.95 (mv₅). In addition, the final $\text{RMSE}_{\mathbf{d}}$ averaged over all chains is about 1.03 ns for both DGM-based MCMC inversions. Compared to NT the MCMC achieves lower RMSE values and higher SSIM values when the SGAN is the DGM, but the performance is comparable for the VAE-based inversion.

For comparison, we plot the marginal prior and posterior latent distributions obtained by performing both NT and MCMC sampling for the mg₅ (SGAN) and mv₅ (VAE) true models (Figs. 3.6 and 3.7). The LogS of each PDF and the KL-divergence values between the various PDFs are provided in Table 3.2. The marginal posteriors obtained for the mg₅ model (Fig. 3.6) are considerably wider than those obtained for mv₅ (Fig. 3.7). This can be also observed in the range of the posterior standard deviation displayed in Figs. 3.3c and 3.4c. The posterior derived by the SGAN-based NT is often not centered around the true value and receives the highest LogS (7.66; see Table 3.2). Moreover, when the SGAN is used as DGM the KL-divergence value between NT and MCMC posteriors goes to infinity due to a minimal overlap. The estimates are more consistent when using the VAE. Here the latent posterior PDF (Fig.

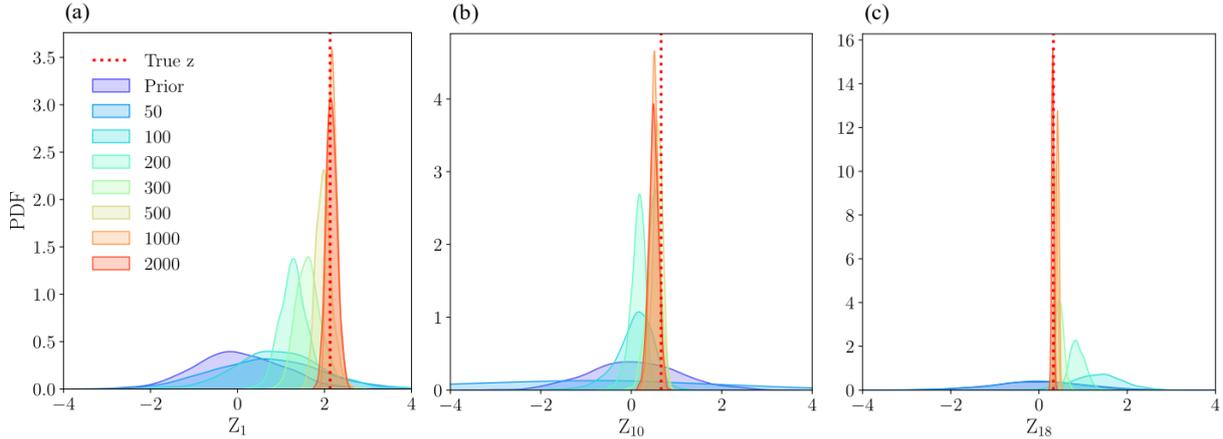


Figure 3.5: Estimation of the variational PDF describing the marginal posterior of the 1st, 10th and 18th latent parameters of the mv_5 model at various training iterations.

Table 3.2: Statistical summary of the posterior PDF in the latent space \mathbf{z} of mg_5 and mv_5 models obtained by neural-transport (NT) and MCMC with $DREAM_{(ZS)}$. Both are compared against the prior PDF of \mathbf{z} . The logS and KL-divergence are reported as the mean value over the \mathbf{z} parameters.

| | PDF (Q) | logS | $D_{KL}(Q P)$ | |
|--------|---------|-------|----------------|-----------|
| | | | MCMC (P) | Prior (P) |
| mg_5 | NT | 7.66 | inf | 3.63 |
| | MCMC | 0.11 | 0 | 1.48 |
| | Prior | 1.60 | - | 0 |
| mv_5 | NT | -1.29 | 0.19 | 2.84 |
| | MCMC | -1.19 | 0 | 2.55 |
| | Prior | 1.43 | - | 0 |

3.7) is either centered around or contains the true value for both the NT inversion and the MCMC inversion and posterior uncertainty derived by the NT is similar to that of the MCMC. Furthermore, the KL-divergence of the NT posterior from the MCMC posterior is relatively small (0.19), which indicates strong similarity between the two. The two VAE-based approximate posteriors also provide relatively similar LogS value with the NT posterior being slightly more accurate than the MCMC one (-1.29 for NT versus -1.19 for MCMC).

3.4 Discussion

Our results demonstrate that the presented NT approach works well with both SGAN and VAE in terms of reconstructing the true models in the image space. Moreover, the approximate posterior from the VAE-based inference provides a slightly better prediction (low LogS value) than MCMC as well as reliable uncertainty estimates with respect to the true latent space values (Fig. 3.7 and Table 3.2) at a much lower computational cost. Due to the invertible

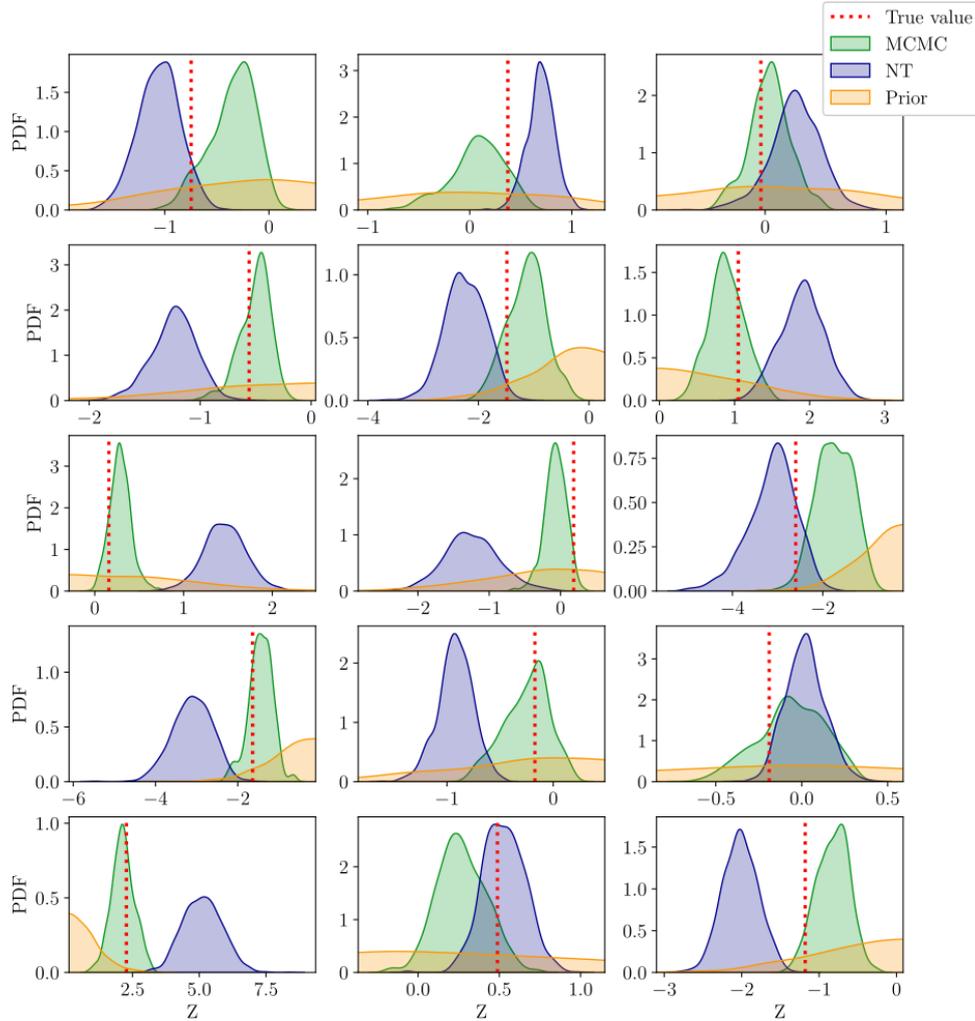


Figure 3.6: Approximate marginal posteriors on the latent space \mathbf{z} obtained with neural-transport (NT) and MCMC for model mg_5 in Fig. 3.2 as well as the prior on the latent space of the SGAN.

transformations of the IAF, in NT we can evaluate the approximate posterior analytically as well as draw random samples from it.

The differences in training (adversarial versus variational) and more so the differences in architecture (fully connected and convolutional layers in the VAE versus fully convolutional spatial GAN) between the two DGMs lead to transformations that vary in their degree of non-linearity. The SGAN transformation provides approximate latent posterior distributions that are both wider and less accurate than those obtained by the VAE-based inversions, thereby, indicating stronger dependencies between the SGAN parameters. The stronger correlation between the SGAN latent parameters can be explained by its spatial architecture, that is, its 2D latent space and fully convolutional layers. Although the approximate posterior in the latent space of the SGAN does not provide a good prediction of the true value, as seen from the $RMSE_z$ and $LogS$ values, the reconstruction of the actual model (in the original high-dimensional image space) is reasonable with SSIM values in the 0.71 - 0.91 range. This suggests that two latent vectors that lie far from each other may correspond to similar

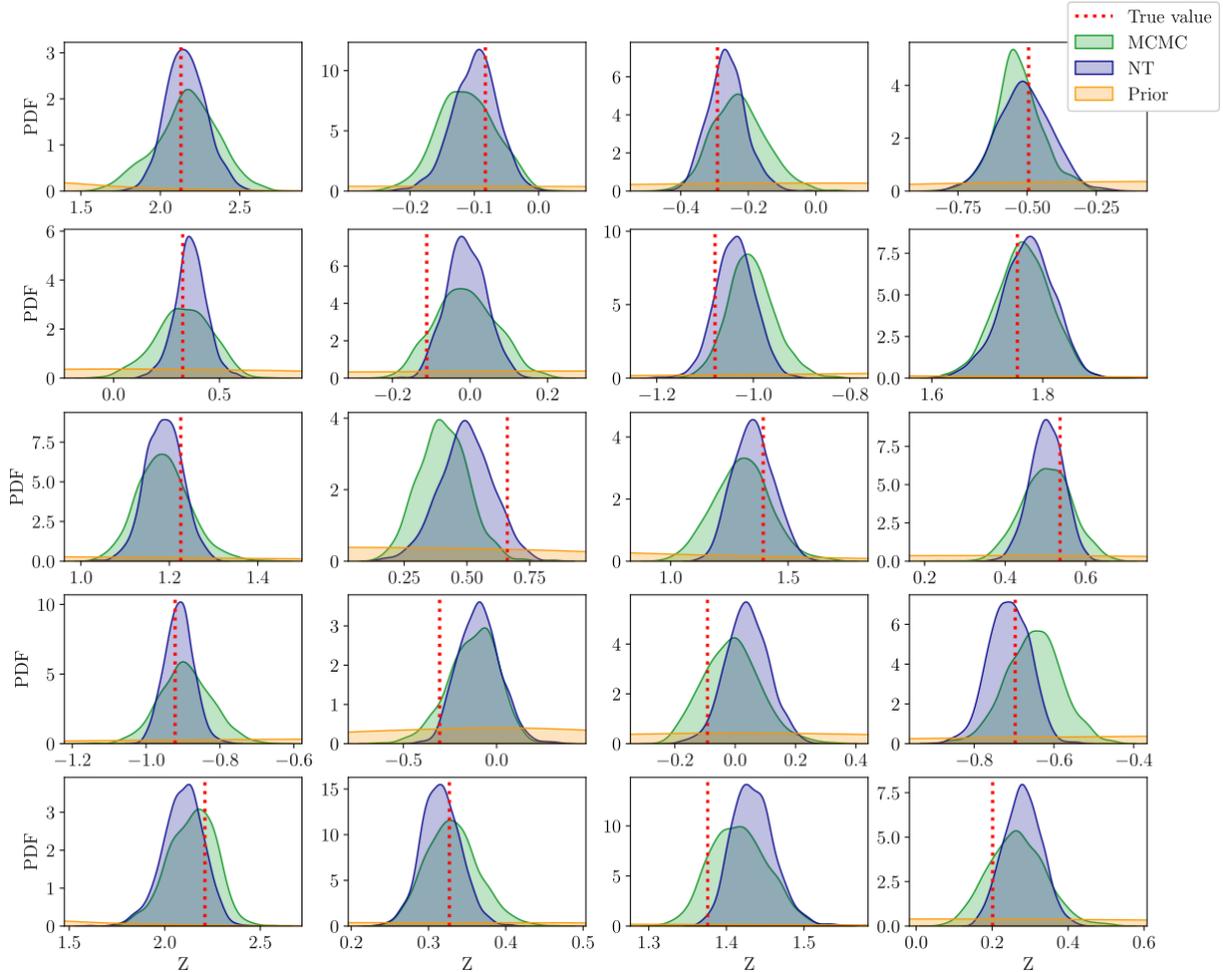


Figure 3.7: Approximate marginal posteriors on the latent space \mathbf{z} obtained with neural-transport (NT) and MCMC for model mv_5 in Fig. 3.2 as well as the prior on the latent space of the VAE.

realizations in the high-dimensional model space. This can be explained as an attempt of the generator to accommodate the change in topology between the latent space and the real manifold in the high-dimensional space (Lopez-Alvis *et al.*, 2021). Moreover, Lopez-Alvis *et al.* (2021) showed that the convexity of the misfit function in the latent space of the VAE can be controlled by choosing the right hyper-parameters during training. This is an important advantage, as Laloy *et al.* (2019) showed that the misfit function in the latent space of the SGAN is in fact a rough surface containing many local minima. Instead, the VAE is trained in such a way that both the changes induced in topology and the convexity of the misfit function are controlled. However, one must also keep in mind that this comes at the expense of generation accuracy and that realizations generated by the VAE are more lossy and, consequently, less sharp than those obtained from a GAN (Hou *et al.*, 2017; Bao *et al.*, 2022; see also our Figure 3.2).

We stress that variational inference is limited by the parameterization of the approximate distribution, hence, in some cases the solution might not converge to an appropriate approximation of the posterior if the parameterized distribution is not expressive enough. A

further improvement, for example in the case of the SGAN-based inversion, can be achieved by running traditional MCMC or Hamiltonian Monte Carlo (HMC; *Duane et al.*, 1987; *Neal*, 2011) within the latent space of the IAF (*Hoffman et al.*, 2019; *Papamakarios et al.*, 2021). In this setting, the normalizing flow is used to reparameterize the posterior distribution and the starting point for the MCMC sampler is the approximation resulting from training the IAF, thereby, providing a much shorter burn-in. Additionally, it provides a favorable sampling geometry from a standard normal which may improve MCMC mixing in multi-modal problems (*Nijkamp et al.*, 2020).

The results for the VAE-based inversion (Fig. 3.4) demonstrate that NT can be performed with a single particle. The SGAN-based inversion on the other hand requires more than a single particle and starts to perform well (measured in terms of data misfit only) when the number of particles is increased to 20 for the considered case studies (Fig. 3.10). This finding is consistent with those by *Laloy et al.* (2019) and *Lopez-Alvis et al.* (2021), where deterministic gradient-based inversions within the low-dimensional space of the SGAN was found to perform poorly due to the highly non-linear SGAN transformation and small-scale irregularities in the objective function. Increasing the number of particles allows for more regions in the latent/model space to be explored at each iteration, thereby providing a more robust gradient estimation. This is perhaps particularly important at the initial phase of inversion where a vast region of the prior is explored. As opposed to many other gradient-based methods, NT involves random sampling at each iteration, which makes it more robust and reduces the risk of getting stuck in a local minimum. Increasing the number of particles is also shown to result in earlier convergence, however, it comes at the cost of an increased computational expense as evolving one particle for one iteration involves a forward simulation and the calculation of its Jacobian. For instance, for the SGAN-based inversions with 20 particles it takes an average of 260 training iterations (among those who have reached convergence) to converge, which translates into 5200 forward simulations. In contrast, the maximum number of iterations needed for the single-particle VAE-based inversion to converge is 1767 forward simulations only (see Table 3.1). Those SGAN-based inversion cases which have not converged possibly require either more training iterations or more particles. Nevertheless, among the converged cases using either the VAE or the SGAN, the total number of forward simulations needed in the NT approach is always much lower than in MCMC. When compared based on their individual convergence criteria, the computational times required by MCMC and NT differ by a factor of 7 in favor of NT. This factor would be 56 if the eight MCMC chains were not evolved in parallel.

In our NT applications, the forward simulations of the particles are computed sequentially. However, the computational time when considering multiple particles can be significantly reduced by distributing the computations associated with individual particles over several processing units (preferably using one unit per particle). This option is available in NT-based inversions: transformations and forward simulations can be performed in parallel when using more than a single particle, given that the forward simulation is parallelizable. Note that auto-differentiation as performed by machine learning libraries such as PyTorch and TensorFlow requires the forward solver to be implemented in the library in use, or alternatively, that the gradients of the forward response are provided by the user (*Richardson*, 2018; *Laloy et al.*, 2019). As mentioned in subsection 3.2.5, to maintain a differentiable operation we do not

threshold the generated images to a binary value of 0 or 1 in neither the SGAN nor the VAE generations. This limitation might affect inversion performance when the inverted model is either binary or categorical.

Using DGMs results in a drastic reduction in the number of inferred parameters (here from 8385 to only 15 and 20 SGAN and VAE latent parameters, respectively) as well as realizations which honor the higher-order statistical features of the model as represented by a training image (*Laloy et al., 2017, 2018*). The NT mechanism on the other hand, leverages on gradient information, random drawing of particles and flexible parameterization of the approximate posterior distribution. Consequently, NT combined with a DGM forms an efficient and scalable approach for solving high-dimensional inverse problems. As discussed in section 3.3, most of the computational cost of the NT approach comes from the forward simulation and the largest updates to the IAF parameters occur at early training stages. Therefore, further improvement of NT efficiency could probably be gained by updating the Jacobian of the forward solver in eq. (3.15) less frequently as the inversion advances. Another option could be to set a large number of particles at the beginning of the inversion and gradually decrease it as updates to the IAF parameters are becoming smaller. We leave these two options for future studies. Additionally, our study was limited to channelized subsurface models and a weakly non-linear forward solver operating in a crosshole setting. Further research is required to assess the performance of this approach for different geomodels, physical models (e.g. fluid flow, wave-based reflection data) and 3D problems.

3.5 Conclusions

Neural-transport refers to the application of variational Bayes to train an IAF transformation, which maps samples from a simple base distribution into samples from an approximate posterior over the latent space of a DGM. We demonstrate that inferring a model in the latent space of either a SGAN or a VAE using neural-transport significantly reduces the number of forward simulations required compared to MCMC sampling. In this respect, DGMs play an important role in improving the efficiency and scalability of the NT approach when dealing with geophysical inverse problems that are generally high-dimensional. Our results are in agreement with previous works concerning deterministic inversions performed in the latent space of a SGAN or a VAE. Indeed, the VAE is found to provide a better reconstruction of the true model in both the low-dimensional latent and high-dimensional model image spaces; the agreement with the MCMC results were also excellent. NT combined with SGAN provides a reasonable estimation in the high-dimensional model space and much better results overall compared to previous results based on deterministic inversions. In contrast to MCMC, where the posterior is estimated based on an ensemble of samples, NT provides a closed-form solution of the approximate posterior such that it can be efficiently evaluated and sampled from. Performance of NT-based inversion could be further improved by combining it with MCMC sampling within the latent space of the NT, starting from the solution of the NT-based inversion.

3.6 Appendix

3.6.1 IAF design

The IAF is constructed as sequential flows with each of them producing a different distribution (see Figure 3.8a). Each flow involves an autoregressive network which takes as input either variables from the base distribution (if it is the first flow) or variables that are a result of the preceding flow. The output of the autoregressive network is a mean μ and logarithm of the standard deviation $\log(\sigma)$ (to prevent negative standard deviation as output, it is therefore exponentiated to get σ). To achieve the autoregressive property, the connections between layers are masked to ensure conditioning of variables only on those preceding them (see illustration in Figure 3.8b; *Germain et al., 2015; Papamakarios et al., 2017*). Before each flow the input is re-ordered (permutation) which has been shown to improve the training of such models (*Germain et al., 2015; Kingma et al., 2016*). The autoregressive network includes one hidden layer with $2n$ hidden units (neurons). To be able to represent all degrees of conditioning, the number of units in a hidden layer should be at least $n - 1$. Here we found that $2n$ units in the hidden layer performs slightly better than using n units (a default choice). We introduce a non-linearity to the transformation by applying a ReLU activation function to each flow. Results did not change significantly when other activation functions such as LeakyReLU or ELU were used and we found ReLU to work well for our purposes. Each additional flow in the current architecture introduces $6n^2 + 2$ trainable parameters therefore, the number of flows chosen was based on a consideration of complexity/performance versus computational effort that might vary for different types of models.

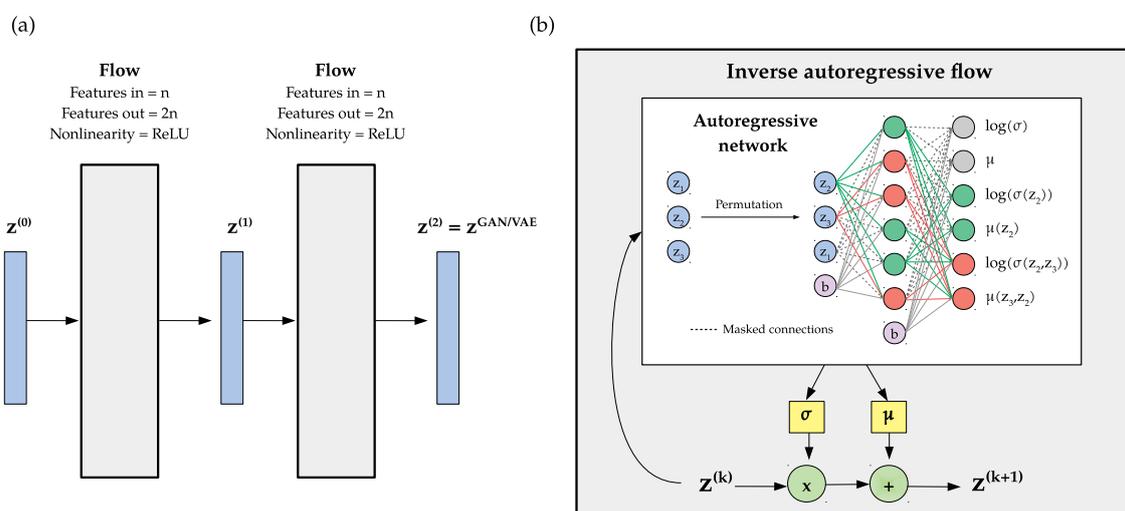


Figure 3.8: Schematic drawing of the IAF architecture. (a) General workflow of the IAF with two flows corresponding to one intermediate distribution. The number of neurons depends on the number of input features n . (b) An illustration of an autoregressive network in which connections are masked to honor the autoregressive property.

3.6.2 Hyperparameter calibration

Once the architecture (i.e. number of flows, number of layers etc.) of the IAF is fixed, there are two main algorithmic variables that may affect the final results: (1) number of particles N_s and (2) learning rate. To determine proper values for these variables and test the robustness of the approach to different choices, we perform a hyperparameter search and show the results on models mg_1 (for the SGAN) and mv_3 (for the VAE) in Figure 3.2. We first test the NT routine with the SGAN and VAE using learning rates of: 0.1, 0.05, 0.01, 0.005, 0.001 and one particle. The curves in Figure 3.9 represent the RMSE of the data misfit $RMSE_d$ during the NT-training for the different learning rates. For both the SGAN and VAE it is found that a learning rate of 0.01 gives the fastest and most stable convergence towards the target misfit of 1 ns corresponding to the standard deviation of the noise added to the synthetic data. A higher learning rate results in either instability or convergence to a sub-optimal solution, while a lower learning rate results in a slower convergence.

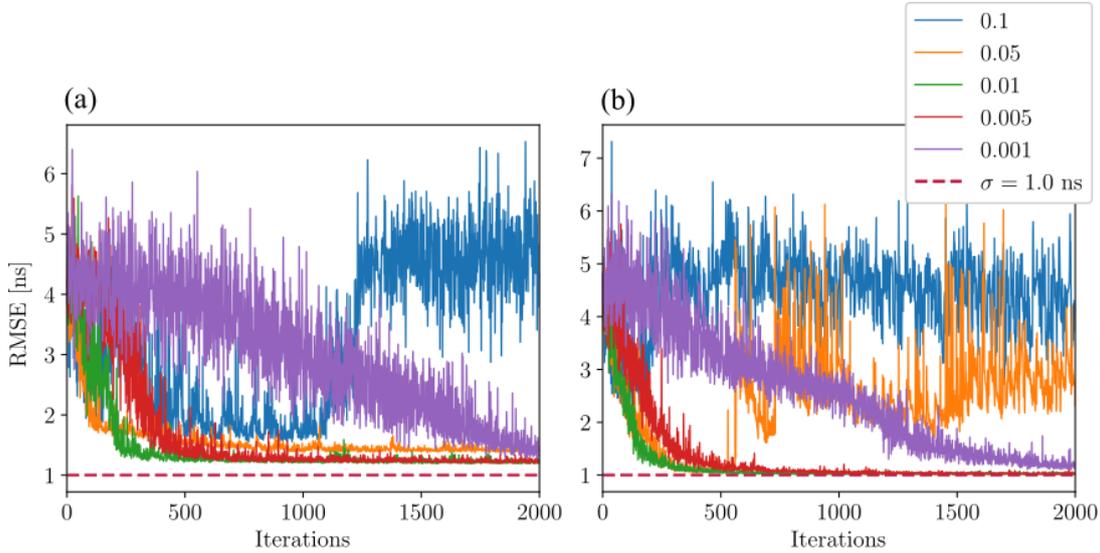


Figure 3.9: Average $RMSE_d$ value over model particles during inference as a function of the learning rate using (a) SGAN and (b) VAE as DGM.

To evaluate how many particles N_s to use, we fix the learning rate at the optimal value of 0.01. We then test the NT using 1, 5, 10 and 20 particles. We compare the $RMSE_d$ averaged over the particles during NT-training and plot them as a function of the number of training (gradient-descent) iterations (Figs. 3.10a and c) and the number of forward simulations (Figs. 3.10b and d). Increasing the number of particles leads to more stable and earlier convergence with respect to the number of iterations. However, increasing the number of particles also induces a higher computational demand. When considering the $RMSE_d$ with respect to the number of forward simulations as in Figure 3.10d, it becomes clear that the VAE optimization performed using one particle only provides the best trade-off (at least if not considering parallelization), as the target misfit is then reached at the lowest computational cost. In contrast, the SGAN benefits from a larger number of particles as we found that using 20 particles reduces the risk of getting stuck in a local minima (Fig. 3.10b) and in most cases

it brings the RMSE_d closer to the target misfit despite the fewer gradient-descent iterations assigned to it. This behavior is likely due to the higher degree of non-linearity of the SGAN, for which a higher number of particles provides more robustness with respect to local features in the misfit function when updating the parameters of the autoregressive network.

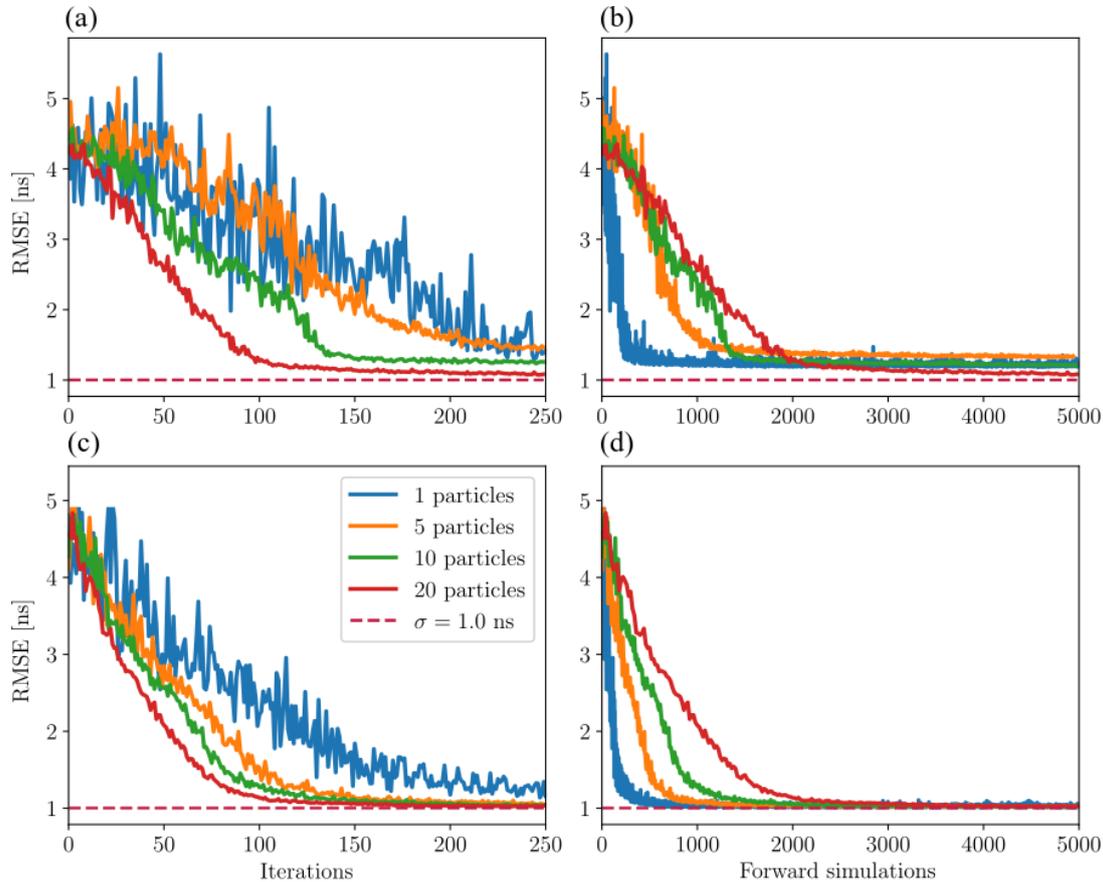


Figure 3.10: Average RMSE_d values during the NT inversion as a function of the number of (a) and (c) iterations and (b) and (d) forward simulations with SGAN (a,b) and VAE (c,d) as DGM. The different curves correspond to different number of particles used to estimate the ELBO and its gradients at each NT iteration.

3.6.3 Supplementary results

The NT-inversion with the mv_1 model performed relatively poorly at the lower boundary when using one particle only (Figs. 3.4a-d). Indeed, the mean model (Fig. 3.4b) and the standard deviation (Fig. 3.4c) suggest that the true model is not part of the posterior. By extending the number of particles to ten, we find that the posterior mean is much closer to the true model (compare Figs. 3.11a-b) and the standard deviation is higher implying a better exploration of the posterior. The lower region of high standard deviation maps well to the interface between the lower channel and the background matrix. This suggests that adding more particles can also be beneficial when using VAE as DGM.

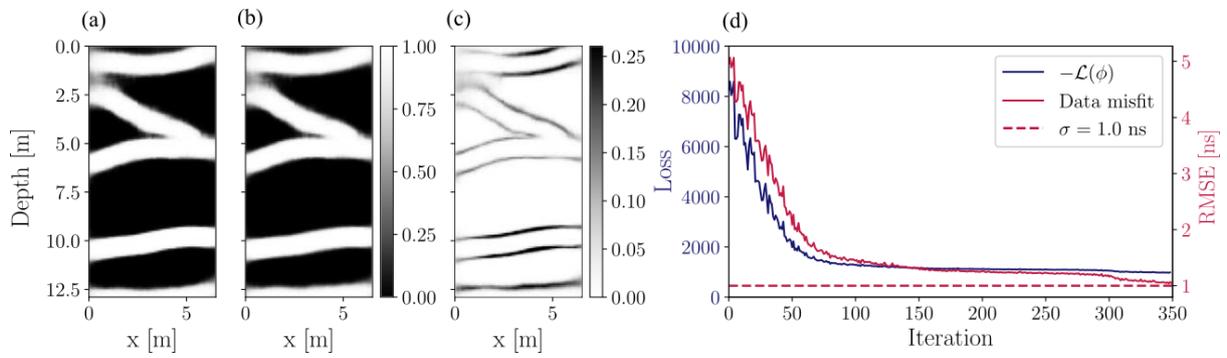


Figure 3.11: (a) True mv_1 model (b) mean posterior models obtained from NT using ten particles and (c) posterior standard deviation in the model image space. (d) The ELBO loss and $RMSE_d$ in ns during training.

Chapter 4

Conditioning of multiple-point statistics simulations to indirect geophysical data

Shiran Levy, Lea Friedli, Grégoire Mariéthoz, Niklas Linde

Submitted to *Computers & Geosciences*¹

¹Levy, S., Friedli, L., Mariéthoz, G. and Linde, N. Conditioning of multiple-point statistics simulations to indirect geophysical data. Submitted to *Computers & Geosciences*.

Abstract

We introduce a new methodology providing geostatistical realisations honouring both indirect geophysical data and complex prior knowledge described by a training image. It uses a multiple-point statistical (MPS) simulation algorithm to iteratively build up a realisation pixel-by-pixel starting from an empty grid. During each simulation step, the MPS algorithm generates multiple proposals from the conditional prior and one of them is selected proportionally to an approximate likelihood accounting for the indirect geophysical data. The posterior distribution is approximated by simulating many complete field realisations (sequentially or in parallel). In our demonstrations of the method, we consider synthetic geophysical data obtained from crosshole ground-penetrating radar first-arrival simulations. We test our approach, which we name Indirect Data Conditional Simulations (IDCS), with both multi-Gaussian priors and linear physics for which analytical posteriors are available as well as for more complex priors and non-linear physics. We assess its accuracy against a Gibbs sampler employed within an extended Metropolis framework. The IDCS method is inherently approximate due to the use of a finite training image, a finite number of MPS candidates at each simulation step and the need to approximate intractable likelihood functions. Nevertheless, the results demonstrate that the posterior approximations obtained by IDCS are often comparable to those obtained with a Markov chain Monte Carlo method, with IDCS being at least one order of magnitude faster. While the method performs the best when the underlying physics is modelled as a linear response, we provide encouraging preliminary results considering a non-linear physical response.

4.1 Introduction

Multiple-point statistics (MPS) is a non-parametric family of methods that rely on sequential simulations to produce geostatistical realisations honouring higher-order spatial statistics present in so-called training images (TI; *Guardiano and Srivastava, 1993; Strebelle, 2002; Zhang, 2006; Mariethoz and Caers, 2014*). These methods rely on sequential assignment of parameter values to points on a simulation grid. This process entails scanning the training image and contrasting the patterns within it with the patterns surrounding the simulated point on the simulation grid, using various distance metrics. They are widely used for applications in geology, hydrogeology, remote sensing and reservoir engineering to obtain model realisations with the right spatial statistics while also honouring available data points (e.g. borehole information) or volume (linear averages, *Straubhaar et al., 2016*) measurements. Even if deep generative models offer highly competitive approaches to provide unconditional realisations (*Laloy et al., 2017, 2018*), MPS algorithms are still far superior in accounting for hard conditioning data (*Zhang, 2015; Hansen et al., 2018; Straubhaar and Renard, 2021*).

Sequential geostatistical simulations, including MPS, are commonly constrained to hard data such as well measurements. Multiple-point statistics methods can additionally be utilized as a post-processing tool to refine a deterministic, smoothness-constrained solution obtained from a non-linear inversion. In such a setting, the resolution of the model realisations is enhanced (*Lochbühler et al., 2014; Linde et al., 2015b*), but without any guarantees that

the resulting realisations honour the original geophysical data. Other algorithms such as the Blocking Moving Window (BMW), introduced by *Alcolea and Renard (2010)*, constrains MPS simulations to both hard data and connectivity information and while it introduces correlation with soft data (e.g. geophysical data) through a secondary training image, it does not impose it as a constraint.

Geophysical inversion using MPS algorithms generally involves Markov chain Monte Carlo (MCMC) methods (*Mariethoz et al., 2010b; Hansen et al., 2012*). At each model proposal step, an MPS algorithm performs sequential Gibbs sampling in which a subset of randomly chosen model cells (*Mariethoz et al., 2010b*) or a randomly selected patch (*Hansen et al., 2012*) within the model domain are re-simulated. The model proposals generated by the MPS algorithm are consistent with the patterns of the training image and conditional on the cell values that have not been re-simulated. The acceptance probability is given by the ratio of the likelihoods of the proposed model and the previous model in the chain. This extended Metropolis method (*Mosegaard and Tarantola, 1995*) will eventually locate the posterior and sample proportionally to it. However, it is often very slow in practice, as geostatistical re-simulation and forward simulation times are non-negligible, and there is often a need to perform millions of MCMC iterations before the posterior is sampled sufficiently. The latter is a result of very high correlation in the sampled MCMC states, implying that very long runs are needed to draw a sufficient number of independent samples (*Ruggeri et al., 2015*).

In the context of linear forward problems, *Hansen et al. (2006)* introduced a method for conditioning sequential simulations to noisy indirect data of mixed support (point- and volume-support). This method allows for the incorporation of geophysical measurements into the simulation process. In their implementation the mean and covariance of the posterior probability density function (PDF) is obtained by solving a kriging system with an a-priori mean and covariance as well as support volumes related to the physical response. In a subsequent step, posterior realisations are generated through sequential simulation using the kriging mean and covariance. This method was later extended by *Hansen and Mosegaard (2008)* to accommodate non-Gaussian marginal prior distributions, however, it captures only two-point spatial statistics, and its ability to faithfully reproduce the prior is restricted by the kriging process. Applying the concept of averages over support volumes to MPS, *Straubhaar et al. (2016)* showed how simulations can be constrained to indirect geophysical data and the multiple-point statistics of a conditional prior. In their method, MPS candidate values for a simulated location are accepted according to an accumulated error considering the target value (mean value obtained from the data), a tolerance range and the mean over the support volume in the simulation grid. This method however, does not sample the posterior, as the likelihood used does not account for the error statistics and is based on arbitrary tolerance values.

In this paper, we propose an approach enabling fast geostatistical simulations honouring geophysical constraints under a linear system response and explore its possible extension to non-linear responses. This approach can, for instance, be applied to potential-field methods such as gravity, magnetics and self-potential when prior knowledge is best represented by higher-order statistics (e.g. *Jensen et al., 2022*). Our approach involves gradually constructing an MPS realisation starting from an empty simulation grid or, if hard data is available, with the known local data values. Each simulated value is selected based on geostatistical constraints

considering spatial patterns created by the already informed values, as well as constraints offered by the geophysical data. Incorporating the latter constraint would normally involve calculating a likelihood by marginalising over the distributions of the uninformed model parameters (grid cells), something that is computationally impractical. Instead, we draw at each simulation step k conditional samples with the MPS algorithm and accept one of them proportionally to an approximate likelihood. In our likelihood approximation, we estimate the uninformed model parameter values (mean and covariance) using kriging. Once the simulation grid is fully informed, it can be seen as a draw from an approximate posterior distribution. Conducting multiple independent simulations allows the estimation of an approximate posterior distribution.

Our approach is faster than sequential Gibbs sampling within MCMC, as simulations are built up conditionally to the data at each simulation step and no re-simulation steps are performed. Moreover, the approach can be easily parallelized since each full simulation can be performed independently of other simulations. Nonetheless, the method is approximate due to three factors: (1) the training image is finite, (2) the likelihood distribution is approximated in each simulation step using a limited number of MPS proposals and (3) the uninformed model parameters are assumed to be normally-distributed when approximating the likelihood. To assess the impacts of these approximations on the simulation results, we first consider a training image depicting a multivariate Gaussian field for which the posterior is known analytically. We then consider more complex continuous and binary channelised training images for which comparisons are made in terms of computational effort and accuracy, with respect to a sequential Gibbs sampler. Finally, we introduce an extension of our approach to non-linear physical responses and show preliminary results.

The paper is organised as follows: Section 2 provides a detailed explanation of the theory behind our proposed approach; Section 3 details the metrics and the comparative approach used to assess the quality of the results; Section 4 presents the results obtained when considering a linear physical solver as well as initial results for a non-linear solver; Section 5 discusses the results, highlighting the limitations, advantages and possible future developments. Finally, in Section 6 we conclude the study.

4.2 Methods

Our proposed method allows conditioning MPS simulations to both point data (e.g., facies) and indirect data (geophysical measurements). In our implementation of the method, we rely on QuickSampling (*Gravey and Mariethoz, 2020*) as the MPS algorithm, and indirect geophysical data given by crosshole ground-penetrating radar (GPR) simulations. In the following subsections, we provide a detailed description of our method (section 4.2.1) focusing on the approximation of the likelihood and how to perform a fast update of the kriging mean and covariance. We then proceed by giving a concise description of the QuickSampling algorithm (section 4.2.2) and the considered forward response (section 4.2.3).

4.2.1 Bayesian formulation for conditional sequential simulation

Our method begins with an empty simulation grid $S(x)$, where x denotes the location in the grid. If any (hard) conditioning points are known, they are assigned before the simulation begins. The simulation path \mathbf{p} (order of simulated locations) is generated randomly to maximise the variability of the realisations. The variable $\boldsymbol{\theta}$ is used here as both the simulated property field (model parameters) and the function mapping the location in the grid to the property value. At each simulation step, we distinguish between three types of (model) parameters: "informed" parameters, denoted as $\boldsymbol{\theta}_I$, corresponding to values that were simulated in previous simulation steps or are related to hard data, "simulated" parameters, denoted as θ_S , corresponding to the value of the cell that is simulated in the current step of the algorithm and "uninformed" parameters $\boldsymbol{\theta}_U$, corresponding to the value of empty grid cells that have not yet been simulated. While $\boldsymbol{\theta}_I$ and $\boldsymbol{\theta}_U$ refer to the values in the informed \mathbf{x}_I and uninformed \mathbf{x}_U locations, respectively, θ_S refers to the value in the simulated cell ($\theta_S = \boldsymbol{\theta}(x_S)$). In each simulation step, one of k candidate values proposed for location x_S , that are denoted here as \mathcal{Q}_{MPS} , is chosen proportionally to an unnormalized posterior that is conditional on observed data \mathbf{d} and previously informed parameters $\boldsymbol{\theta}_I$:

$$p(\theta_S|\mathbf{d},\boldsymbol{\theta}_I) \propto p(\mathbf{d}|\theta_S,\boldsymbol{\theta}_I)p(\theta_S|\boldsymbol{\theta}_I). \quad (4.1)$$

The conditional prior $p(\theta_S|\boldsymbol{\theta}_I)$ cannot be computed explicitly, and instead we rely on an MPS algorithm to sample from it. In general, MPS algorithms generate samples from a conditional distribution that preserves higher-order statistics among multiple data points. These algorithms scan the training image, comparing its patterns to that of a defined neighbourhood in the simulation grid $S(x)$. However, our methodology is not confined to MPS and it is adaptable to any algorithm capable of generating multiple samples from a conditional prior.

The simulated data \mathbf{d}^{sim} , resulting from the forward response $g(\boldsymbol{\theta})$, depends on the whole property field with the response contribution arising from the uninformed parameters $\boldsymbol{\theta}_U$ being unknown:

$$\mathbf{d}^{sim} = g(\theta_S,\boldsymbol{\theta}_I,\boldsymbol{\theta}_U). \quad (4.2)$$

This poses a problem as the corresponding likelihood in Eq. (4.1), which is a marginalised likelihood over all uninformed parameters $\boldsymbol{\theta}_U$:

$$p(\mathbf{d}|\theta_S,\boldsymbol{\theta}_I) = \int p(\mathbf{d}|\theta_S,\boldsymbol{\theta}_I,\boldsymbol{\theta}_U)p(\boldsymbol{\theta}_U|\theta_S,\boldsymbol{\theta}_I)d\boldsymbol{\theta}_U, \quad (4.3)$$

is intractable in most cases. To circumvent this problem, we derive below an approximation of the likelihood by estimating the distribution of the uninformed $\boldsymbol{\theta}_U$ parameters conditional on informed $\boldsymbol{\theta}_I$ and simulated θ_S parameters.

Estimation of uninformed quantities and likelihood approximation

To estimate the distribution of the uninformed parameters $\boldsymbol{\theta}_U$ as needed by our likelihood approximation, we rely on kriging-based interpolation (Matheron, 1963). Kriging assumes a mean $m(\mathbf{x})$ and a stationary covariance $C(x_i, x_j)$ function describing the correlation between locations x_i and x_j that are separated by some distance. To build the covariance model, kriging relies on theoretical variograms (Oliver and Webster, 1990). Here we use simple kriging, in which the mean of the property of interest is assumed to be known and we estimate the values of uninformed locations based on conditioning to informed and simulated (MPS candidates) locations (Chilès and Desassis, 2018).

To estimate the uninformed grid points $\boldsymbol{\theta}_U$, we assume that both our property field $\boldsymbol{\theta}$ and the observational noise follow a normal distribution. Note that this Gaussian assumption on the property field is only made to approximate the likelihood (Eq. (4.3)), while the candidates are provided by draws from MPS-based priors. Given a multivariate Gaussian field with the following prior and likelihood distributions

$$\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta) \quad (4.4)$$

$$\mathbf{d}|\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{G}\boldsymbol{\theta}, \boldsymbol{\Sigma}_d), \quad (4.5)$$

there exist an analytical solution both for the likelihood $p(\mathbf{d}|\boldsymbol{\theta})$ and posterior $p(\boldsymbol{\theta}|\mathbf{d})$ distributions (see Appendix 4.7.1). In Eq. (4.4), $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$ represent the mean and covariance matrix of the model parameters. The term $\mathbf{G}\boldsymbol{\theta}$ in Eq. (4.5), where \mathbf{G} is the linear forward operator of the physical response, corresponds to the expected value of the data and $\boldsymbol{\Sigma}_d$ is the covariance matrix of the data errors. We express $\boldsymbol{\theta} = (\theta_S, \boldsymbol{\theta}_I, \boldsymbol{\theta}_U) = (\boldsymbol{\theta}(x_S), \boldsymbol{\theta}(\mathbf{x}_I), \boldsymbol{\theta}(\mathbf{x}_U))$ and $\boldsymbol{\theta}_c = (\theta_S, \boldsymbol{\theta}_I)$ for which $\theta_S \in \mathcal{Q}_{MPS}$ is provided by an MPS algorithm and re-write the prior distribution as the following multivariate distribution

$$\begin{pmatrix} \boldsymbol{\theta}_c \\ \boldsymbol{\theta} \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} \boldsymbol{\mu}_{\theta_c} \\ \boldsymbol{\mu}_\theta \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{\theta_c} & \boldsymbol{\Sigma}_{\theta_c\theta} \\ \boldsymbol{\Sigma}_{\theta\theta_c} & \boldsymbol{\Sigma}_\theta \end{pmatrix} \right), \quad (4.6)$$

where $\boldsymbol{\mu}_{\theta_c}$ and $\boldsymbol{\mu}_\theta$ are the prior means for parameters $\boldsymbol{\theta}_c$ and $\boldsymbol{\theta}$, respectively. Furthermore, $\boldsymbol{\Sigma}_{\theta_c}$ and $\boldsymbol{\Sigma}_\theta$ are the covariance matrices whose (i, j) entries are the covariances between the i -th and j -th value of $\boldsymbol{\theta}_c$ or $\boldsymbol{\theta}$, respectively, and $\boldsymbol{\Sigma}_{\theta_c\theta}$ refers to the covariance matrix consisting of the covariance values between $\boldsymbol{\theta}_c$ and $\boldsymbol{\theta}$. We can then calculate the conditional distribution $\boldsymbol{\theta} | (\boldsymbol{\theta}_c = \boldsymbol{\theta}_c^*) \sim \mathcal{N}(\boldsymbol{\mu}_{\theta|\theta_c^*}, \boldsymbol{\Sigma}_{\theta|\theta_c^*})$ as follows (Prince, 2012):

$$\tilde{\boldsymbol{\mu}}_\theta = \boldsymbol{\mu}_\theta + \boldsymbol{\Sigma}_{\theta\theta_c} \boldsymbol{\Sigma}_{\theta_c}^{-1} (\boldsymbol{\theta}_c^* - \boldsymbol{\mu}_{\theta_c}), \quad (4.7)$$

$$\tilde{\boldsymbol{\Sigma}}_\theta = \boldsymbol{\Sigma}_\theta - \boldsymbol{\Sigma}_{\theta\theta_c} \boldsymbol{\Sigma}_{\theta_c}^{-1} \boldsymbol{\Sigma}_{\theta_c\theta}. \quad (4.8)$$

For informed and simulated parameters θ_c , their entries in the kriging mean $\tilde{\boldsymbol{\mu}}_\theta$ are their value before kriging and their corresponding entries in the kriging covariance matrix $\tilde{\boldsymbol{\Sigma}}_\theta$ are zero. The multiplication $\boldsymbol{\Sigma}_{\theta\theta_c} \boldsymbol{\Sigma}_{\theta_c}^{-1}$ yields the kriging weights that provide the necessary information for interpolating from known grid points (at locations \mathbf{x}_I and x_S) to unknown points (at location \mathbf{x}_U).

At each simulation step, we consider k candidate values $\theta_S \in \mathcal{Q}_{MPS}$ and obtain k kriging means and a single covariance matrix (see Figure 4.1 for illustration). The likelihood of each candidate value is estimated using the kriging mean and covariance as (*Bishop and Nasrabadi, 2006*):

$$\mathbf{d}|\boldsymbol{\theta} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_L, \tilde{\boldsymbol{\Sigma}}_L) \quad (4.9)$$

$$\tilde{\boldsymbol{\mu}}_L = \mathbf{G}\tilde{\boldsymbol{\mu}}_\theta \quad (4.10)$$

$$\tilde{\boldsymbol{\Sigma}}_L = \boldsymbol{\Sigma}_d + \mathbf{G}\tilde{\boldsymbol{\Sigma}}_\theta\mathbf{G}^T. \quad (4.11)$$

For each candidate value and corresponding kriging mean, we calculate the forward response to generate $\tilde{\boldsymbol{\mu}}_L$ (Eq. (4.10)). Additionally, we incorporate the kriging error into the error covariance of the likelihood $\tilde{\boldsymbol{\Sigma}}_L$ (Eq. (4.11)). Normally, in the case of uncorrelated data errors, this matrix is given by a diagonal matrix $\boldsymbol{\Sigma}_d$ with the variance of the observational noise on its diagonal. Finally, the Gaussian likelihood can be approximated as:

$$p(\mathbf{d}|\theta_S, \boldsymbol{\theta}_I) \approx \frac{1}{\sqrt{(2\pi)^{N_d}|\tilde{\boldsymbol{\Sigma}}_L|}} \exp\left(-\frac{1}{2}[\mathbf{d} - \tilde{\boldsymbol{\mu}}_L]^T \tilde{\boldsymbol{\Sigma}}_L^{-1} [\mathbf{d} - \tilde{\boldsymbol{\mu}}_L]\right), \quad (4.12)$$

where N_d is the size of the observed data and $|\tilde{\boldsymbol{\Sigma}}_L|$ is the determinant of $\tilde{\boldsymbol{\Sigma}}_L$. The value assigned to the simulated location is drawn proportionally to the approximate likelihoods of the k proposed values. This is achieved by drawing randomly from the cumulative distribution function (CDF) of the approximate likelihoods.

To estimate the covariance structure of the Gaussian prior distribution in Eq. (4.6), we employ the GSTools Python library (*Müller et al., 2022*) that automatically fits a theoretical variogram to samples from the training image. Based on these samples, which we limit to 30 000, GSTools provides the standard deviation σ , integral scale ℓ in two directions and shape parameter ν of the fitted model. In this paper, the term "training image" refers to either a complete image or a portion of an image used as a template for the simulation process.

Fast update of conditional mean and covariance

To gain computational efficiency by avoiding re-computing Eqs. (4.7)-(4.8) at each simulation step, we adopt the fast kriging update approach introduced by *Emery (2009)* and later extended by *Chevalier et al. (2014, 2015)*. *Chevalier et al. (2015)* used this approach to assimilate new observation points to sequential simulations. Their technique enables a fast update of the kriging mean $\tilde{\boldsymbol{\mu}}_\theta$ and the kriging covariance $\tilde{\boldsymbol{\Sigma}}_\theta$ given new conditioning data points. Instead of calculating the conditional mean and covariance from scratch at each simulation step, we simply perturb the previous estimate given the newly simulated value. To maintain consistency, we adopt the general notation used in the previous subsection and express the kriging update equations as a function of the simulation step, denoted as t . In this notation, $\boldsymbol{\theta}$ becomes $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\theta}_c^{(t)} = \boldsymbol{\theta}(\mathbf{x}_c^{(t)}) = (\boldsymbol{\theta}_I^{(t)}, \boldsymbol{\theta}_S^{(t)}) = (\boldsymbol{\theta}(\mathbf{x}_I^{(t)}), \boldsymbol{\theta}(x_S^{(t)}))$. The conditional distribution then becomes $\boldsymbol{\theta}^{(t)} | (\boldsymbol{\theta}_c^{(t)} = \boldsymbol{\theta}_c^{*(t)}) \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_\theta^{(t)}, \tilde{\boldsymbol{\Sigma}}_\theta^{(t)})$.

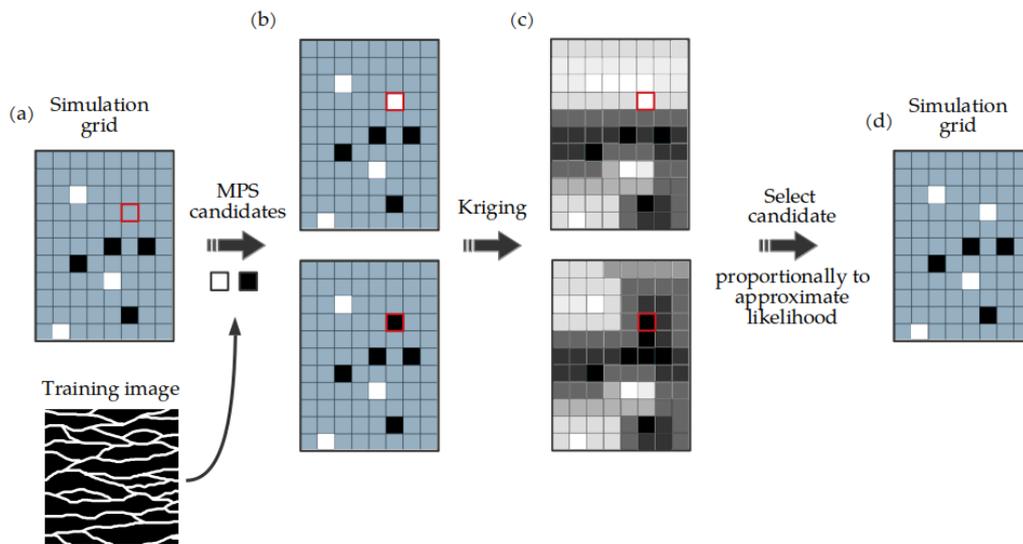


Figure 4.1: Schematic illustration of one IDCS simulation step for a binary model. (a) The simulation grid where the simulated location is marked by a red square, informed locations are marked as either white or black filled squares and uninformed locations are marked with blue. (b) The MPS algorithm proposes $k = 2$ candidates sampled from the training image that are conditional on the pattern in the simulation grid. (c) Kriging is then used to estimate uninformed grid cells in order to calculate an approximate likelihood given the geophysical data. (d) One candidate is drawn proportionally to the approximate likelihood and assigned to the simulated location.

Let us express $\Sigma_\theta = (\sigma_\theta(x_i, x_j))_{1 \leq i, j \leq n}$ and $\tilde{\Sigma}_\theta = (\tilde{\sigma}_\theta(x_i, x_j))_{1 \leq i, j \leq n}$ and re-write Eqs. (4.7)-(4.8) with respect to step t and the location x ,

$$\tilde{\boldsymbol{\mu}}_\theta^{(t)}(x) = \boldsymbol{\mu}_\theta(x) + \boldsymbol{\sigma}_\theta(\mathbf{x}_c^{(t)}, x)^T \boldsymbol{\sigma}_\theta(\mathbf{x}_c^{(t)}, \mathbf{x}_c^{(t)})^{-1} (\boldsymbol{\theta}_c^{(t)} - \boldsymbol{\mu}_\theta(\mathbf{x}_c^{(t)})), \quad (4.13)$$

$$\tilde{\boldsymbol{\sigma}}_\theta^{(t)}(x_i, x_j) = \boldsymbol{\sigma}_\theta(x_i, x_j) - \boldsymbol{\sigma}_\theta(\mathbf{x}_c^{(t)}, x_i)^T \boldsymbol{\sigma}_\theta(\mathbf{x}_c^{(t)}, \mathbf{x}_c^{(t)})^{-1} \boldsymbol{\sigma}_\theta(\mathbf{x}_c^{(t)}, x_j). \quad (4.14)$$

Using the fast update, $\tilde{\boldsymbol{\mu}}_\theta^{(t)}(x)$ is computed by perturbing the kriging mean resulting from the previous step $\tilde{\boldsymbol{\mu}}_\theta^{(t-1)}(x)$, according to the difference between the value of the MPS candidate $\theta_S^{(t)}$ and the value at location x_S in $\tilde{\boldsymbol{\mu}}_\theta^{(t-1)}$:

$$\tilde{\boldsymbol{\mu}}_\theta^{(t)}(x) = \tilde{\boldsymbol{\mu}}_\theta^{(t-1)}(x) + \tilde{\boldsymbol{\sigma}}_\theta^{(t-1)}(x_S^{(t)}, x)^T (\tilde{\boldsymbol{\sigma}}_\theta^{(t-1)}(x_S^{(t)}, x_S^{(t)})^{-1} (\theta_S^{(t)} - \tilde{\boldsymbol{\mu}}_\theta^{(t-1)}(x_S^{(t)})), \quad (4.15)$$

where $\tilde{\boldsymbol{\sigma}}_\theta^{(t-1)}(x_S^{(t)}, x)^T (\tilde{\boldsymbol{\sigma}}_\theta^{(t-1)}(x_S^{(t)}, x_S^{(t)})^{-1}$ represents the kriging weight. The update to the conditional covariance is based on the same kriging weight

$$\tilde{\boldsymbol{\sigma}}_\theta^{(t)}(x_i, x_j) = \tilde{\boldsymbol{\sigma}}_\theta^{(t-1)}(x_i, x_j) - \tilde{\boldsymbol{\sigma}}_\theta^{(t-1)}(x_S^{(t)}, x_i)^T (\tilde{\boldsymbol{\sigma}}_\theta^{(t-1)}(x_S^{(t)}, x_S^{(t)})^{-1} \tilde{\boldsymbol{\sigma}}_\theta^{(t-1)}(x_S^{(t)}, x_j). \quad (4.16)$$

Once the conditional mean and covariance are updated, they are plugged into Eqs. (4.10) and (4.11). For a summary of our full conditioning algorithm, see Algorithm 3. The algorithm describes a single run of our method which we refer to as Indirect Data Conditional Simulations (IDCS).

4.2.2 QuickSampling algorithm

In the implementation of our method, we use QuickSampling (QS) which is a computationally efficient pixel-based MPS algorithm that in contrast to many other pixel-based MPS algorithms, does not store conditional distributions (Strebelle, 2002; Straubhaar et al., 2011) or rely on threshold criteria for choosing a candidate (Mariethoz et al., 2010a). In this algorithm, the training image, denoted as T , is scanned to find a close match to the pattern around the simulated location x_S in the simulation grid S . The pattern is represented by values and their relative positions to x_S . At each simulation step a neighbourhood in S , denoted as N , is considered; $N(x)$ is centred around the currently simulated grid cell and contains within a specified radius the locations in the simulation grid that are previously informed (either previously simulated or conditioning data points).

In QS, the cross-correlation between $N(x)$ and T is calculated to generate a dissimilarity (mismatch) map E :

$$E(T, N(x)) \propto \mathcal{F}^{-1} \left\{ \sum_{i \in I} \sum_{j \in J} \overline{\mathcal{F}\{\mathbb{1}(T_i) \circ f_j(T_i)\}} \circ \mathcal{F}\{w_i \circ \mathbb{1}(x_i) \circ g_j(N(x)_i)\} \right\}, \quad (4.17)$$

where \mathcal{F} and \mathcal{F}^{-1} are the fast Fourier transform and its inverse, $\overline{\mathcal{F}\{\}}$ is the conjugate and \circ indicates an element-wise multiplication. Furthermore, w is a weighting matrix and it can be set to assign different weights as a function of the distance from x_S . The indicator function $\mathbb{1}$

Algorithm 3: Indirect Data Conditional Simulations (IDCS) with a general MPS algorithm

```
1 Input: simulation grid  $S(x)$  (either empty or informed by hard conditioning data),  
   training image  $T$ , simulation path  $\mathbf{p}$ , observed data  $\mathbf{d}$ , number of candidates  $k$  and  
   algorithm-specific MPS parameters  $\mathbf{b}$ .  
2 Output: fully informed grid with property field  $\theta$   
3 set simulation step  $t = 1$   
4 for  $t \leq \text{size}(\mathbf{p})$  do  
5    $x_S = p_t$   
6   Function  $MPS(\theta, T, x_S, k, \mathbf{b})$   
7     Sample candidate values from  $T$  that are conditional on the  $\theta_I$  around  
     location  $x_S$  in the simulation grid.  
8     return  $\mathcal{Q}_{MPS} = \{\theta_S^1, \dots, \theta_S^k\}$   
9   if  $t=1$  then  
10    | Compute  $\tilde{\mu}_\theta$  and  $\tilde{\Sigma}_\theta$  using Eqs. (4.6)-(4.8) for all  $k$  candidates  
11  else  
12    | Update  $\tilde{\mu}_\theta$  and  $\tilde{\Sigma}_\theta$  using Eqs. (4.15)-(4.16) for all  $k$  candidates  
13  end  
14   $\tilde{\mu}_L, \tilde{\Sigma}_L \leftarrow$  compute Eqs. (4.10)-(4.11) for all  $k$  candidates  
15  Approximate  $p(\mathbf{d}|\theta) \leftarrow$  compute Eq. (4.12) for all  $k$  candidates  
16  Calculate cumulative distribution function (CDF) of  $k$  likelihoods  $p(\mathbf{d}|\theta)$  and draw  
   one value from  $\mathcal{Q}_{MPS}$   
17  Populate grid cell at location  $x_S$  with the selected candidate value  
18   $t = t + 1$   
19 end
```

equals 1 at informed grid cells and 0 everywhere else, that is,

$$\mathbb{1}(x) = \begin{cases} 1, & \text{if } N(x) \text{ is informed} \\ 0, & \text{otherwise} \end{cases}. \quad (4.18)$$

The variables f_j and g_j represent components of a series of decomposed functions that depend on the distance metric used (see *Gravey and Mariethoz (2020)* for more information). In the original implementation, candidates are sorted in increasing order of mismatch and the simulated value is sampled proportionally to a probability determined by a user-defined rank k_{rank} . In addition to k_{rank} , the QS algorithm requires a user-defined parameter n that determines the number of neighbouring locations around x_S , and effectively the size of the neighbourhood $N(x)$ to be considered for MPS conditioning. In our implementation, the QS algorithm functions solely as a means to sample k conditional draws from the prior that are evaluated using an approximate likelihood. Therefore, the QS parameter k_{rank} is replaced in our implementation by k_{cand} which represents the number of candidates provided by the QS algorithm ($k = k_{cand}$).

4.2.3 Forward response

To test IDCS, we consider a crosshole geometry in which GPR antennas placed in two boreholes are used to send and receive electromagnetic signals and the travel-times between different source and receiver pairs are measured. In this setting, $\boldsymbol{\theta} = \mathbf{s}$, where \mathbf{s} is the slowness field (inverse of the velocity \mathbf{v}). Specifically, we use a ray-based formulation in which the travel-time t_{ray} is an integration of slowness s over the ray path l :

$$t_{ray} = \int s(l) dl. \quad (4.19)$$

The aforementioned physical response can be written in a general form as

$$\mathbf{d} = g(\mathbf{s}) + \boldsymbol{\varepsilon}, \quad (4.20)$$

where $g(\cdot)$ is the forward operator projecting the parameters \mathbf{s} in the model space into a vector \mathbf{d} in the data space and the process typically involves some type of error $\boldsymbol{\varepsilon}$. Here we consider only uncorrelated, randomly distributed Gaussian (measurement) noise under $\boldsymbol{\varepsilon}$.

Linear physical response

Considering linear physics, Eq. (4.19) can be simplified into $t_{ray} \approx \sum_i l_i \cdot s_i$ and the response becomes a matrix-vector multiplication operation

$$\mathbf{d}^{sim} = \mathbf{G}\mathbf{s}, \quad (4.21)$$

where \mathbf{G} (also referred to as the sensitivity matrix) contains the ray length in each grid cell considering only a straight path between the source and receiver. The simulated data \mathbf{d}^{sim} are represented in a vector containing each source-receiver response in the form of travel-times.

Non-linear physical response

When dealing with a non-linear physical response, an approximation to Eq. (4.20) is required in order to calculate Eqs. (4.10)-(4.11). This is achieved by linearising the forward operator $g(\mathbf{s})$ around a given subsurface model to obtain the sensitivity matrix (Jacobian). In general, the Jacobian represents the gradient around the point of linearisation and, therefore, the forward response is calculated with respect to a reference point \mathbf{s}_0 :

$$\mathbf{d}^{sim} = g(\mathbf{s}_0) + \mathbf{J}(\mathbf{s}_0)(\mathbf{s} - \mathbf{s}_0), \quad (4.22)$$

where $\mathbf{J}(\mathbf{s}_0)$ is the Jacobian calculated for the subsurface model \mathbf{s}_0 . In travel-time tomography the forward operator can be replaced by the Jacobian to calculate the physical response $\mathbf{d}^{sim} = \mathbf{J}\mathbf{s}$, where \mathbf{J} is the Jacobian given the slowness field \mathbf{s} . In this case, \mathbf{G} in Eqs. (4.10) and (4.11) is simply replaced with \mathbf{J} to obtain:

$$\tilde{\boldsymbol{\mu}}_L = \mathbf{J}\tilde{\boldsymbol{\mu}}_\theta \quad (4.23)$$

$$\tilde{\boldsymbol{\Sigma}}_L = \boldsymbol{\Sigma}_d + \mathbf{J}\tilde{\boldsymbol{\Sigma}}_\theta\mathbf{J}^T. \quad (4.24)$$

4.3 Comparative approach and quality assessment criteria

4.3.1 Sequential Gibbs sampling

To assess the quality and performance of the IDCS method when no analytical solution is available, we compare it against results obtained with the extended Metropolis algorithm (Mosegaard and Tarantola, 1995) using a sequential Gibbs sampler (Hansen et al., 2012; Cordua et al., 2012).

The extended Metropolis algorithm allows exploring the posterior PDF when dealing with a prior distribution of arbitrary complexity which cannot be quantified but from which samples can be drawn. In this algorithm, the acceptance or rejection of a model proposal $\boldsymbol{\theta}'$ is determined by the acceptance probability $P_{accept} = \min\left(1, \frac{p(\mathbf{d}|\boldsymbol{\theta}')}{p(\mathbf{d}|\boldsymbol{\theta}^{(t)})}\right)$, where $\boldsymbol{\theta}^{(t)}$ represents the current model. If accepted $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}'$ otherwise $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)}$. Gibbs sampling takes a given realisation $\boldsymbol{\theta}$ and at each iteration computes the conditional distribution at a random position x_i

$$p(\boldsymbol{\theta}(x_i)|\boldsymbol{\theta}(x_1), \dots, \boldsymbol{\theta}(x_{i-1}), \boldsymbol{\theta}(x_{i+1}), \dots, \boldsymbol{\theta}(x_N)). \quad (4.25)$$

A value for $\boldsymbol{\theta}(x_i)$ is then drawn from the conditional distribution forming the new realisation. Sequential Gibbs sampling combines sequential simulations with the Gibbs sampler such that one can generate realisations from the conditional distribution.

In this study, we use QS to generate conditional model proposals and proceed to estimate the posterior distribution using the extended Metropolis with each chain being initialised by an unconditional MPS simulation. At each MCMC step, a random subset of the model domain is re-simulated while being conditioned on the remaining part of the domain. The size of the subset is adapted during the first 2000 MCMC steps, within the burn-in period, using the

parameter δ . After burn-in, the value of δ remains constant to maintain detailed balance of the Markov chain. The parameter δ represents half of the side-lengths of a square centred at a grid point chosen at random. The size of δ is used to control the step length of the sequential Gibbs sampler, where a small value leads to a high acceptance rate with highly correlated chains and larger values lead to lower acceptance rates but less correlated chains (Hansen *et al.*, 2012). During burn-in, the value of δ is adjusted every 20 iterations according to

$$\delta_{new} = \delta * \frac{\bar{P}_{acc}}{P_{target}}, \quad (4.26)$$

with the aim of maintaining a reasonable acceptance rate (Gelman *et al.*, 1996; Cordua *et al.*, 2012). The variable \bar{P}_{acc} is the average acceptance rate between adjustment steps and P_{target} is the target acceptance rate, which we set to 0.3. We use the Gelman-Rubin diagnostic (Gelman and Rubin, 1992), which compares the variance between the independent chains and within the chains, to assess the convergence of the MCMC chains to a stationary distribution for each of the model parameters. Convergence is declared when $\hat{R} \leq 1.2$ for all considered parameters (grid cell values).

4.3.2 Performance metrics

To determine the optimal algorithmic setting and to assess the quality of the resulting posterior realisations, we use the structural similarity index (SSIM; Wang *et al.*, 2004), which is calculated with respect to the reference subsurface model, and the weighted root-mean-squared error (WRMSE) which is calculated with respect to the synthetic data. The SSIM metric indicates the structural similarity between two images. It is defined as:

$$SSIM(\mathbf{u}, \mathbf{v}) = \frac{(2\mu_{\mathbf{u}}\mu_{\mathbf{v}} + C_1)(2\sigma_{\mathbf{uv}} + C_2)}{(2\mu_{\mathbf{u}}^2 + \mu_{\mathbf{v}}^2 + C_1)(2\sigma_{\mathbf{u}}^2 + \sigma_{\mathbf{v}}^2 + C_2)}, \quad (4.27)$$

where \mathbf{u} and \mathbf{v} are $M \times M$ sliding windows of their respective $[0, 1]$ normalised image, $\mu_{\mathbf{u}}$ 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} , $\sigma_{\mathbf{uv}}$ is the covariance between \mathbf{u} and \mathbf{v} and C_1 and C_2 are constants. We use 7×7 windows and set $C_1 = 0.01$ and $C_2 = 0.03$. The SSIM score ranges from -1 to 1 , where 1 indicates perfectly matching images. It is reported as a mean value across all posterior realisations. The data fit is evaluated with respect to the standard deviation of the observational noise σ during the simulation using the WRMSE

$$WRMSE = \sqrt{\frac{1}{N_d} \sum_{i=1}^{N_d} \left[\frac{d_i - d_i^{sim}}{\sigma_i} \right]^2}, \quad (4.28)$$

between the observed data \mathbf{d} associated with the reference model and the simulated data \mathbf{d}^{sim} corresponding to posterior realisations. The reported WRMSE is the final value averaged over all simulations. A WRMSE value close to one indicates an appropriate data fit while values larger than one indicate that the data residuals are too large compared to the data noise. In addition to those two metrics, we show for each test case the model realisation for

Table 4.1: True and estimated training image statistics. Estimated values are based on a variogram fitted to 30 000 samples drawn from a training image of size 500×500 pixels.

| TI | True | | | | Estimated | | |
|--|-----------|-------|-----------------|--------------------|--------------|-------|--------------------|
| | l_x/l_y | ν | μ [ns/m] | σ [ns/m] | l_x/l_y | ν | σ [ns/m] |
| Gaussian | 10/5 | 1 | 14.332 | 0.827 | 10.569/5.597 | 1.26 | 0.811 |
| Connected high-conductivity structures | - | - | 14.338 | 0.831 | 9.458/9.657 | 0.92 | 0.806 |
| Binary channels | - | - | 13.580 | 1.826 | 27.220/6.421 | 1.30 | 1.889 |

which the lowest root-mean-squared error (RMSE) with respect to the reference subsurface model was obtained.

4.4 Results

We consider three different training images: (1) a multivariate Gaussian field, (2) connected high-conductivity structures and (3) binary channels, each corresponding to a full stationary image of size 2500×2500 pixels representing an area of 250×250 m. Rather than using the entire available image as a training image, we select a smaller section of 50×50 m serving as the training image for the QS algorithm and variogram fitting. The three training images are shown in Figure 4.2 and their true and estimated covariance model parameters (standard deviation, integral scale and shape parameter) are reported in Table 4.1. The multivariate Gaussian image was generated using the fast Fourier transform Moving Average (FFT-MA; *Ravalec et al., 2000*) method with an exponential model ($\nu = 1$). After generation, it was re-scaled to have a mean of $0.07 \text{ m}\cdot\text{ns}^{-1}$ and a standard deviation of $0.004 \text{ m}\cdot\text{ns}^{-1}$. To generate an image with connected high-conductivity structures, we use the transformation in *Zinn and Harvey (2003)*. This field manipulation transforms an isotropic random Gaussian field of mean zero and unit standard deviation to a field in which high values are connected. After the transformation, the image is re-scaled to have a mean of $0.07 \text{ m}\cdot\text{ns}^{-1}$ and a standard deviation of $0.004 \text{ m}\cdot\text{ns}^{-1}$. The image with binary channels is taken from *Zahner et al. (2016)* and velocity values of $[0.06, 0.08] \text{ m}\cdot\text{ns}^{-1}$ are assigned to the channels and surrounding matrix material, respectively.

Following preliminary tests to determine the QS parameters (see Appendix 4.7.2), we use $n = 10$ for the continuous models and $n = 25$ for the binary one as they result in the best simulation quality. To mitigate the potential risk of sampling unfavourable candidates in the presence of a finite and possibly small training image, we set $k_{cand} = 100$ for both continuous and binary models. When performing MCMC inversion, we use the original implementation of QS and set $k_{rank} = 1.2$ (represents a probability rather than the number of candidates) and $n = 30$ as those values lead to a good simulation quality (*Gravey and Mariethoz, 2020; Meerschman et al., 2013*). Throughout the IDCS simulations and the MCMC inversion the QS parameters remain constant.

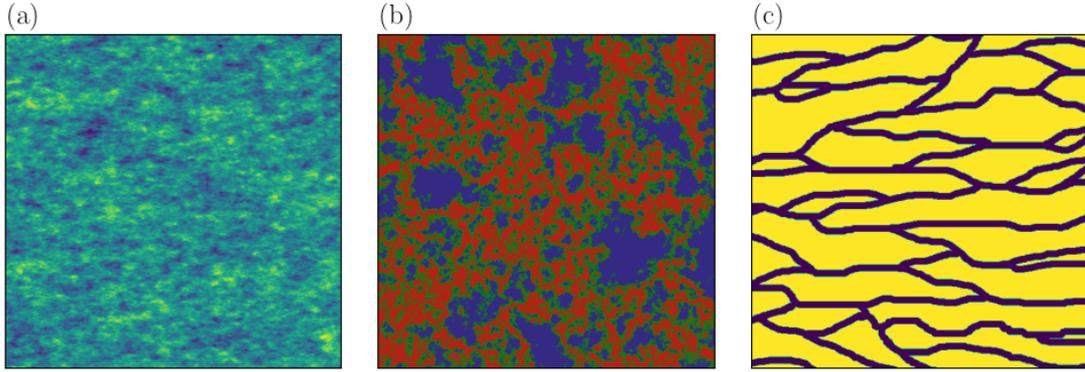


Figure 4.2: Training images for the various tested models. 500×500 pixels (50×50 m) section of (a) anisotropic, random Gaussian field, (b) isotropic field with connected high-conductivity structures generated by applying the transformation in *Zinn and Harvey (2003)* and (c) binary channels.

We consider a model domain of size 5×10 m with 0.1 m discretisation yielding a total of 5000 model parameters. The forward response is computed given two boreholes separated by a distance of 5 m. The borehole on the right side of the domain contains 25 source locations and the borehole on the left contains 25 receiver locations. The antennas are located between 0.2 and 9.8 m depth with 0.4 m separating subsequent antenna positions. Ray-paths between source-receiver pairs that exceed angles of $\pm 50^\circ$ to the horizontal are filtered out and are not considered during inversion. Consequently, the number of data points is 515. The reference model (synthetic truth) is cropped from a portion of the full image that remains unused during the simulation process and the corresponding synthetic observed data of all case studies are contaminated with normally distributed noise with mean zero and standard deviation of 1 ns.

4.4.1 Linear physics

We first show the results obtained from IDCS simulations considering different subsurface models and a linear physical response.

Multivariate random Gaussian field

We perform 100 independent IDCS runs (each with a different simulation path) given the noise-contaminated synthetic data corresponding to the reference model in Figure 4.3a. Since we deal with a multi-Gaussian property field and linear physics, we can compute the analytical solution (see Appendix 4.7.1) of the posterior distribution and use it for comparison. The element-wise mean and standard deviation of the analytical solution and the approximate posterior obtained by IDCS are displayed in Figure 4.3. The mean obtained from running 100 IDCS simulations (Fig. 4.3c) is almost identical to the analytical mean (Fig. 4.3b) and the three IDCS posterior samples, representing the best and worst data fit (Fig. 4.3d and 4.3e, respectively) as well as the closest matching subsurface model realisation (Fig. 4.3f),

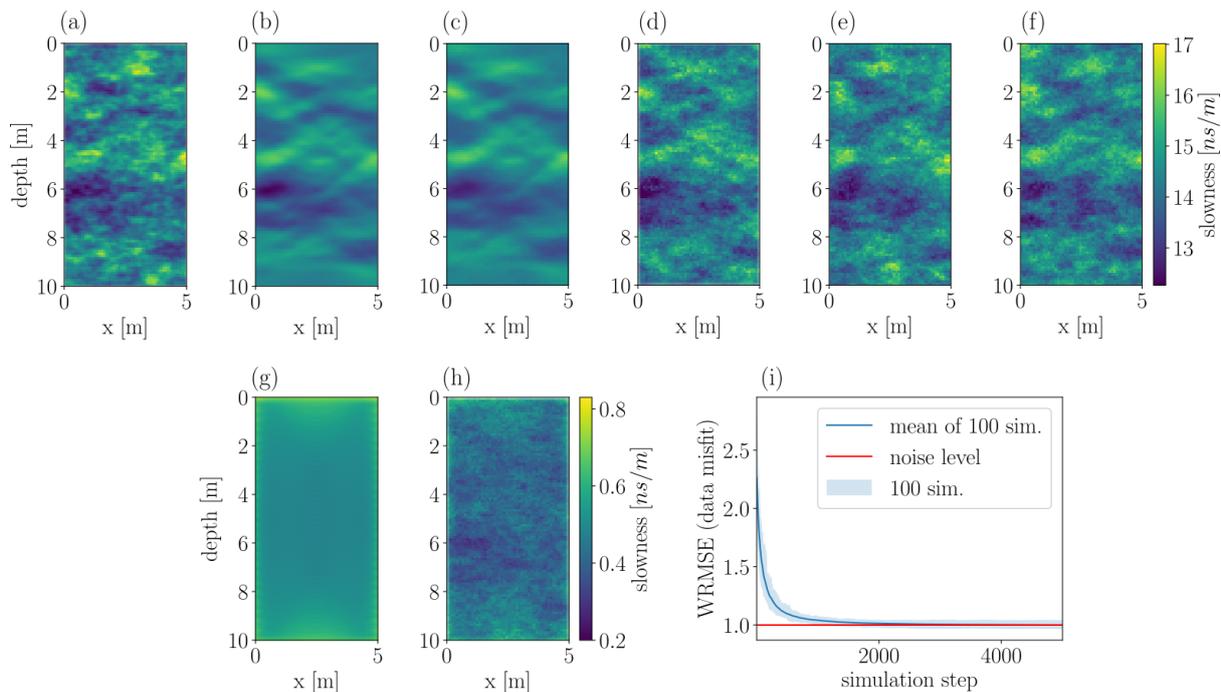


Figure 4.3: IDCS results for a random multivariate Gaussian field and a linear forward solver. (a) Reference model, the (b) analytical and (c) approximate (IDCS) posterior pixel-wise mean, where the latter is computed on 100 IDCS realisations. IDCS realisations with the (d) lowest and (e) highest data WRMSE while (f) is the IDCS realisation with the lowest model RMSE. Pixel-wise standard deviation of the (g) analytical and (h) approximate (IDCS) posteriors. (i) The WRMSE curves of 100 IDCS runs as well as their mean.

are all reproducing the patterns in the reference model. The standard deviation calculated on the conditional realisations (Fig. 4.3h) underestimates (by 17% on average) that of the analytical solution (Fig. 4.3g). This underestimation is likely a consequence of using a finite training image and only 100 IDCS runs. As the simulations are conditioned on the observed data, we expect the data misfit to gradually decrease during the simulation to a WRMSE of one, representing realisations that fit the data to the noise level. This behaviour is confirmed by our results in that the median WRMSE among the 100 realisations is 1.00 (Table 4.2) and it is already around 1.01 after simulating 1850 grid cells (Fig. 4.3i).

Connected high-conductivity structures

We now perform 100 independent IDCS runs given the noise-contaminated synthetic data corresponding to the reference model in Figure 4.4a. Since no analytical solution is available for this case, we compare the results against eight independent MCMC chains (see section 4.3.1). We provide computational resources to permit the maximal performance of each method, namely, one CPU per chain for MCMC (eight in total) and one CPU per simulation for the conditional MPS simulations (100 in total). Both methods are executed on a cluster that is equipped with AMD EPYC™ 7402 CPUs. It takes around 100 minutes to run 100 conditional QS simulations in parallel and it took around 26 hours to perform 20 000 MCMC steps (per chain, in total 160 000 samples). For the MCMC, we used a re-simulated sub-domain with a maximum size of 11×11 cells ($\delta = 5$) resulting in acceptance rate of 31% on average. The

Table 4.2: Summary of IDCS results for linear and non-linear physical responses. The mean SSIM as well as the median data WRMSE computed on 100 IDCS runs for the three types of models.

| Physics | TI type | SSIM | WRMSE |
|------------|--|------|-------|
| | Gaussian | 0.50 | 1.00 |
| Linear | Connected high-conductivity structures | 0.48 | 1.02 |
| | Binary channels | 0.86 | 1.02 |
| Non-linear | Gaussian | 0.46 | 1.03 |
| | Connected high-conductivity structures | 0.44 | 1.05 |
| | Binary channels | 0.55 | 2.44 |
| | Binary channels (linearised Jacobian) | 0.60 | 2.56 |

MCMC chains did not converge after 20 000 iterations and the \hat{R} -values range from 1.27 to 8.42 with the median value being 4.54.

The reference model contains connected high-value features with different orientations and those aligned vertically typically present challenges in terms of identifiability in a crosshole setting. This is seen in the posterior mean of both MCMC samples (Fig. 4.4b) and the IDCS realisations (Fig. 4.4c). While the features that are horizontally-oriented are present in the estimated posterior mean obtained from the IDCS, the vertically-oriented ones are unresolved leading to a slightly lower SSIM than in the previous test case (0.48 versus 0.50, see Table 4.2). This is not a method-specific problem as similar SSIM values are observed for the MCMC posterior samples (on average 0.48). The IDCS realisations exhibit higher standard deviation (Fig. 4.4h) than those observed in the MCMC samples (Fig. 4.4g). In both methods, the highest standard deviation is observed where high-slowness features are present or at the locations where they are poorly resolved. This is also evident from the notable variability observed in these locations in the independent IDCS realisations displayed in Figures 4.4d-f. The median WRMSE among the IDCS realisations is 1.02, indicating an overall appropriate data fit and suggesting that the vertical features are not well constrained by the available data.

Binary channels

The last test case with linear physics is a binary reference model with channel-like structures (Figure 4.5a). No analytical solution is available and we compare again the IDCS results against those obtained from eight MCMC chains. While the computation cost of the IDCS runs is the same as in the previous example, it took 72 hours to perform 20 000 MCMC steps (per chain). The longer computational times are a result of a larger re-simulated sub-domain with a maximum size of 17×17 cells ($\delta = 8$) leading to an average acceptance rate of 24%. As in the previous test case, the MCMC chains did not converge after 20 000 iterations and the \hat{R} -values range from 1.00 to 28.92 with the median value being 1.36.

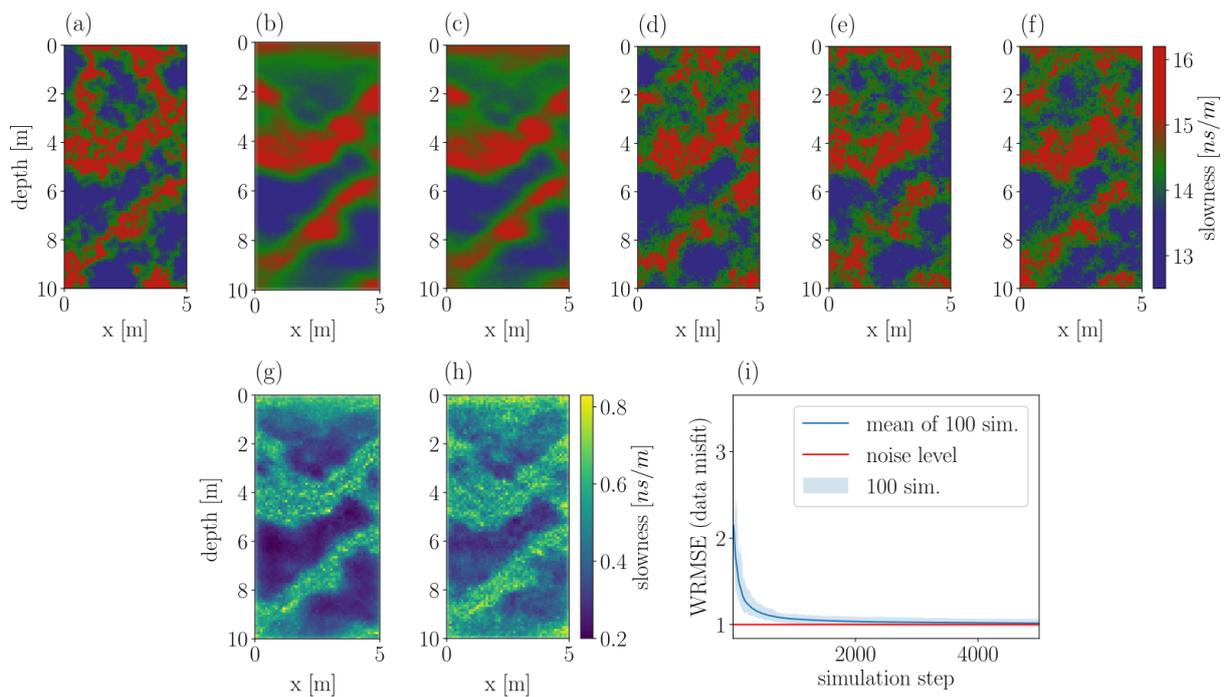


Figure 4.4: IDCS results for the isotropic field with connected high-conductivity structures and a linear forward solver. (a) Reference model, posterior pixel-wise mean computed on (b) MCMC samples and (c) 100 IDCS realisations. IDCS realisations with the (d) lowest and (e) highest data WRMSE, while (f) is the IDCS realisation with the lowest model RMSE. Pixel-wise standard deviations of the posterior approximated by (g) MCMC samples and (h) IDCS realisations. (i) The WRMSE curves of 100 IDCS runs as well as their mean.

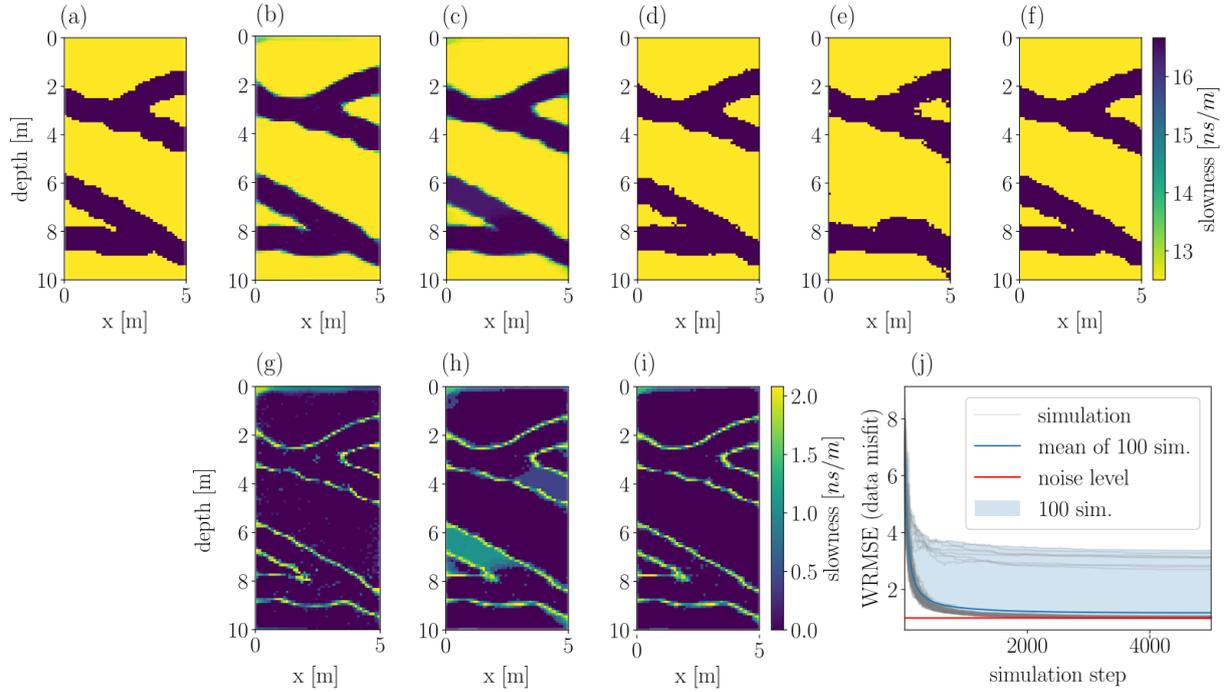


Figure 4.5: IDCS results for the binary channelised field and a linear forward solver. (a) Reference model, posterior pixel-wise mean computed on (b) MCMC samples and (c) 100 IDCS realisations. IDCS realisations with the (d) lowest and (e) highest data WRMSE, while (f) is the IDCS realisation with the lowest model RMSE. Pixel-wise standard deviations of the posterior approximated by (g) MCMC samples, (h) IDCS realisations and (i) IDCS realisations excluding outliers. (j) The WRMSE curves of 100 IDCS runs as well as their mean.

The posterior mean of the MCMC inversion (Figs. 4.5b) and the approximate posterior mean of the IDCS (Fig. 4.5c) reconstruct the channel features well. In accordance with *Zahner et al.* (2016), the highest uncertainty is concentrated around the boundaries of the channels (Figs. 4.5g and 4.5h). While most IDCS realisations correctly reproduce the channels in the reference model (e.g. Figs. 4.5d and 4.5f), seven IDCS realisations do not locate channel material around 6 – 8 m and one misses it around 3.5 – 4.5 m (e.g. Fig. 4.5e). These eight realisations have significantly higher WRMSE values (2.70 – 3.37) than the remaining realisations (average WRMSE of 1.02, see Fig. 4.5j). These artefacts appear at an early stage when matrix material is incorrectly placed where a channel should be, thus constraining subsequent simulation steps such that the resulting data misfit is high. Nonetheless, the aberrant simulations are easily distinguishable from the other simulations and can be regarded as outliers. One approach is to calculate the Inter Quartile-Range statistic (IQR; *Ter Braak and Diks, 2009*) during the simulation and to discard those simulations that are indicated as outliers. This can be done either during the simulation (to avoid redundant computation) or as a post-processing step. After simulating approximately 23% of the grid (corresponding to 1138 grid cells), the IQR statistics identify the eight aberrant simulations as outliers. If these simulations are excluded, the uncertainty becomes comparable to the ones obtained from the MCMC posterior samples (Fig. 4.5i).

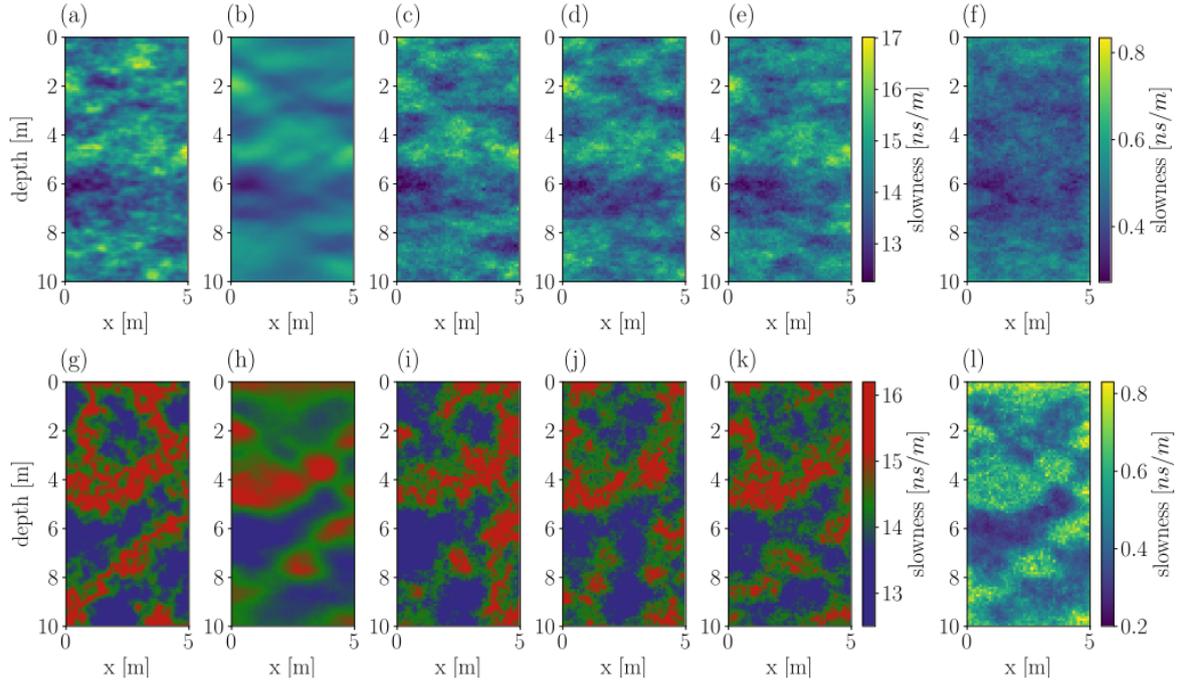


Figure 4.6: IDCS results given reference models (a) and (g) and a non-linear forward solver. (b) and (h) Means of 100 IDCS realisations, (c) and (i) realisations with the lowest WRMSE, (d) and (j) realisations with the highest WRMSE, (e) and (k) realisations with the lowest model RMSE and (f) and (l) the standard deviations of 100 IDCS realisations.

4.4.2 Non-linear physics

In this section we present results obtained when running IDCS with a non-linear forward response (see Section 4.2.3). As the calculation of the Jacobian is generally expensive, instead of calculating \mathbf{J} for each MPS candidate it is computed based on the kriging mean, given the informed grid cells at step t : $\theta_I^{(t)}$. Accordingly, the number of Jacobian updates reduces to the number of grid cells to be simulated. To calculate the forward response, we use the pyGIMLi geophysical modelling library (Rücker *et al.*, 2017) to calculate the shortest path between a source and receiver pair given a slowness model. The accuracy of the forward response depends on the number of secondary nodes on the edges of the grid cell, allowing for more ray angles. Here we limit the number of secondary nodes used to compute the Jacobian to two in order to avoid too long computation times.

Running the IDCS with the aforementioned non-linear forward response takes 21 hours on average. The mean of the approximate posterior for the multivariate Gaussian case (Fig. 4.6b) is similar to the reference model (Fig. 4.6a) and the standard deviation of the 100 realisations (Fig. 4.6f) is similar to the linear-physics case. The isotropic field with connected high-conductivity structures results in similar posterior mean and standard deviation (Figs. 4.6h and 4.6l, respectively) as with linear physics, however, structures are less connected and are more patchy (Figs. 4.6i-k). In both types of subsurface models, the SSIM has slightly reduced values from 0.50 and 0.48 to 0.46 and 0.44 for the multivariate Gaussian and connected high-conductivity structures, respectively (see Table 4.2). Although the WRMSE is close to 1 in both cases, it increased by 0.03 compared to the WRMSE reached with linear physics.

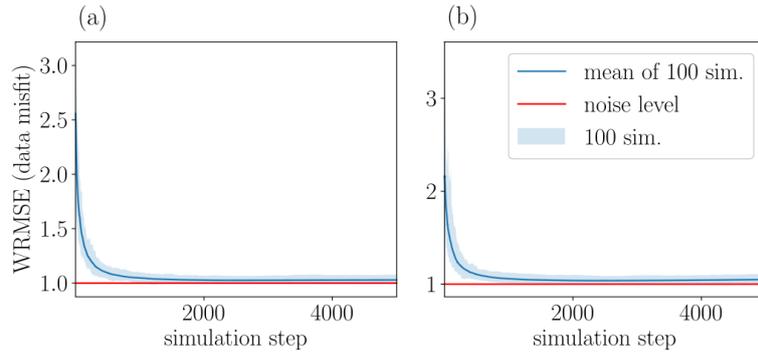


Figure 4.7: Data WRMSE curves during the IDCS run given the reference model in (a) Figure 4.6a and (b) Figure 4.6g. The WRMSE is calculated at each simulation step using a non-linear forward solver.

In contrast to the continuous test cases, the application to the binary channels model yields a substantial decrease in the quality of the posterior approximation in comparison with the linear physics. The SSIM reduced from 0.86 to 0.55 and the WRMSE increased from 1.02 to 2.38. While the mean of the IDCS posterior (Fig. 4.8b) captures the channel structure, it is excessively smooth. Additionally, the individual realisations are of lower quality compared to those obtained with linear physics. This can be also observed in a large uncertainty on the boundaries as well as inside the channels (Fig. 4.8f). Correspondingly, the WRMSE curves in Figure 4.9a are scattered and none of the IDCS realisations fit the data to the noise level. The underlying approximations (Gaussianity, continuity, single Jacobian update for all candidates) together with a higher level of non-linearity intensifies the aberrant simulations problem already observed for the linear physics case in Figure 4.5 when considering this training image.

To improve the approximation we run the IDCS for the same observed data, but with a constant Jacobian linearised around the realisation in Figure 4.8c, that is, the realisation corresponding to the lowest WRMSE. This further run adds around 100 minutes of computation to the total computation time (see Section 4.4.1) as the forward operator remains constant during the simulation and no update is performed. The posterior approximation is overall improved as characterized by an increase in SSIM to an average of 0.60. Moreover, the mean of the posterior (Fig. 4.8g) becomes better defined and the uncertainty within the channels is reduced (Fig. 4.8k). The channel feature within the 6 – 8 m range still presents a significant degree of uncertainty. Nevertheless, this specific feature posed a challenge even in the linear case, as can be seen in Figure 4.5h. The WRMSE curves in Figure 4.9b are calculated using the Jacobian linearised around the realisation in Figure 4.8c. While these exhibit an overall reduction in the WRMSE, the real WRMSE of the final realisations (with the Jacobian being computed for each individual realisation) has increased from 2.44 in the first run to 2.56 in the second run (Table 4.2).

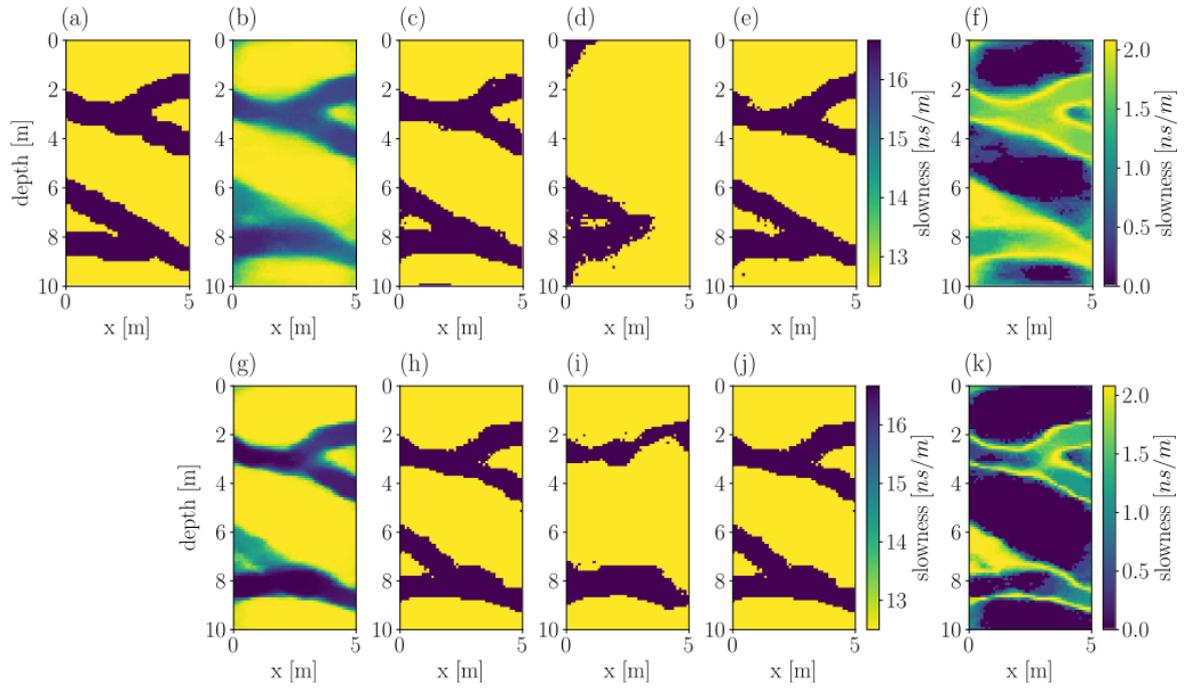


Figure 4.8: IDCS results for the binary channelled subsurface model in (a) and a non-linear forward solver considering (b)-(f) IDCS runs with the Jacobian updated according to the kriging mean and (g)-(k) considering subsequent IDCS runs with a constant Jacobian corresponding to the realisation in (c). (b) and (g) Means of 100 IDCS realisations, (c) and (h) realisations with the lowest WRMSE, (d) and (i) realisations with the highest WRMSE, (e) and (j) realisations with the lowest model RMSE and (f) and (k) the standard deviations of 100 IDCS realisations.

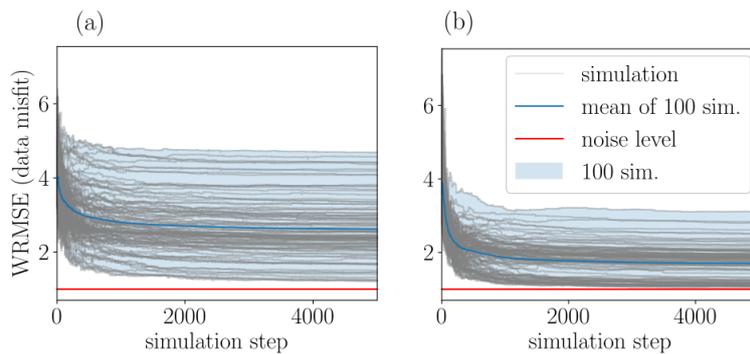


Figure 4.9: Data WRMSE curves during the IDCS run given the reference model in Figure 4.8a. The WRMSE in (a) is calculated at each simulation step using the linearised Jacobian around the kriging mean while in (b) the WRMSE is calculated using a constant Jacobian corresponding to the realisation in Figure 4.8(c).

4.5 Discussion

The proposed IDCS method successfully approximates the posterior distribution when considering an ensemble of simulations (e.g., 100), given linear physics, for both Gaussian and non-Gaussian reference models. The IDCS runs are able to provide posterior approximations that are comparable in quality to those obtained when performing MCMC, but at a much lower computational cost. This applies also to the binary test case, which poses a greater challenge due to the values being discrete, while the likelihood approximation assumes continuity and a Gaussian distribution. As a consequence of this discrepancy between the nature of the model and the estimation method, a small fraction of the IDCS runs introduce artefacts in the early stages, as matrix material is erroneously placed in locations where channel material exists in the reference model. Fortunately, these outlier simulations are easily distinguishable and can be removed at an early stage or in a post-processing step using statistical metrics for dispersion, such as the IQR.

In comparison with the block data method of *Straubhaar et al.* (2016), our method works by estimating unknown parameters and sampling MPS candidates according to an approximate likelihood, eliminating the need for precise knowledge of the ray paths. Hence, when considering non-linear physical responses and provided that the forward response is differentiable and can be linearised, the Jacobian can be used to obtain the kriging error and a first-order approximation of the forward response. Results for both continuous reference models suggest that the IDCS is able to approximate the posterior rather well even when the forward response is non-linear, with only slight deterioration in SSIM and WRMSE metrics compared to the linear cases. In the binary subsurface case, the issue with aberrant simulations seen in the linear case is worsened likely due to a combination of the Gaussian approximation and a poor coverage of the channels by ray paths (see Appendix 4.7.3). A possible improvement could be gained by using a different likelihood approximation for binary or categorical model parameters. The results from conducting a second run of IDCS simulations, where we linearise the Jacobian around the best data-fitting realisation from the initial run, show improvement of the posterior approximation.

The results obtained by IDCS runs are inherently approximate due to the finite training image (prior distribution), the limited number of candidate values considered at each simulation step, and the approximation of the intractable likelihood function. Testing of the influence of the number of candidates (k) and the number of neighbours (n) (see Appendix 4.7.2) suggests that the quality of the simulation and the data fit is less sensitive to changes in n than in k . For a finite training image with a fixed size, large n (50 and above for a 500×500 pixels training image) can potentially lead to pattern degradation and the generation of artefacts. This is due to the limited number of distinct patterns available in the training image. On the other hand, the choice of k represents a trade-off between structure and data fit. For large k , the algorithm is forced to sample more values with decreasing pattern similarity and some of them will be accepted by the algorithm as they might lead to sufficiently low data misfit values. This means that the optimal choice of n and k depends on the size of the training image and the diversity of its patterns. Given that the algorithm is vectorized with respect to k , increasing k does not introduce additional computation time. However, the computation time increases quadratically with the size of the training image (*Gravey and Mariethoz, 2020*).

The computational cost of the IDCS algorithm does not scale linearly with the size of the model domain. Three factors come into play: the increase in the number of simulation steps (linear effect), the need to multiply a larger covariance matrix (Eq. (4.11)) (non-linear effect) and the need for a larger training image (non-linear effect). In our examples, the approximation of the likelihood for the linear case is responsible for 97% of the computation time of a single simulation step. Out of the time it takes to approximate the likelihood and return a simulated value, 76% is spent on computing $\tilde{\Sigma}_L$ (Eq. (4.11)), 11% on the (linear) forward response (for all candidates using vectorized operations), 2% on the likelihood function (Eq. (4.12)) and the rest on various small operations. This suggests that modern matrix multiplication algorithms (e.g. *Nowak et al.*, 2003) could help enhance the efficiency of our approach and mitigate the impact of the matrix multiplication operation in Eq. (4.11). When compared with MCMC, IDCS is at least an order of magnitude faster depending on the test example. The number of forward simulations required in a single IDCS runs depends linearly on the number of cells to simulate and on the k candidates. In contrast, the computation time of MCMC depends on the number of chains and the number of MCMC steps needed to converge, which is unknown before running the inversion. Using Gibbs sampling, the computational time is also influenced by the size of the re-simulated domain. When the physical response is non-linear the number of Jacobian updates during the IDCS is equal to the number of grid cells to be simulated (as described in Section 4.4.2). Thus, IDCS provides a more predictable and efficient alternative compared to MCMC inversions provided that the inevitable approximations are acceptable. Furthermore, as simulations are independent, the number of simulations that are running simultaneously scale with the number of available processing units (either CPUs or GPUs). A further reduction in the computational time can be achieved by conditioning the simulation on indirect data only up to a stage where the data fit curve stabilises and changes in the data misfit are small (e.g. around 2000 steps in Figure 4.7). At this stage, all necessary large-scale features are present to which the rest of the simulation is constrained. This is exemplified in Figure 2 in *Laloy et al.* (2016), which suggests that when re-simulated parameters are distributed throughout the model domain, a large fraction of the domain has to be re-simulated (50% and above) to obtain significant large differences in the likelihood (and as a result in the data RMSE). The effect of such an approach on the results would need to be tested but the potential reduction in the computational time is substantial.

The IDCS method is suitable when a conceptual model of the subsurface is available in terms of a training image and the physical response is either linear or can be linearised. Examples with linear physics include: tomographic problems where parameters are integrated along straight lines such as muon (*Rosas-Carbajal et al.*, 2017) and x-ray tomography, as well as potential-field applications such as gravity, magnetics (*Blakely*, 1996) and the self-potential method (*Revil and Jardani*, 2013). The method can of course also be used, as in this example, when a linear physics assumption might be acceptable as in GPR amplitude inversion (*Jensen et al.*, 2022). Some additional improvement could probably be gained by using a preferential path strategy (*Hansen et al.*, 2018; *Jóhannsson and Hansen*, 2023). This approach prioritises the simulation of locations that are highly constrained by the available data, such as those traversed by multiple rays and thereby, might decrease the risk of artefacts. In challenging scenarios where conditional MPS simulations struggle to fit the data, particularly in cases involving categorical models, the approximation can be improved by using the MPS condi-

tional realisations as an initial solution for MCMC chains. For instance, one can run multiple conditional MPS simulations and use the realisations that fit the data best to initialise the MCMC chains. By doing so, we effectively shorten the burn-in period and possibly speed up convergence compared to using MCMC only.

4.6 Conclusions

We have introduced a novel approach for conditioning multiple-point statistics simulations to geophysical data represented as linear averages over the model domain. These linear averages are either constant during the simulation (linear physics) or varies as the simulation is built up (non-linear physics). Our method, named IDCS, is stochastic in nature and offers an efficient framework for approximating the posterior distribution by performing many simulation runs in parallel. The conditioning of the geophysical data is performed, for each simulated grid cell, by drawing k conditional values from the prior and accepting one of them proportionally to a kriging-based approximation of the intractable likelihood. In non-linear problem settings, the forward response has to be linearised, leading to a first-order approximation of the likelihood. Considering crosshole ground-penetrating radar data, the method was found to successfully approximate the posterior distribution for three subsurface models: multivariate Gaussian, connected high-conductivity structures, and binary channel. Its main practical limitation is that the computational time scales non-linearly with the size of the model domain due to operations involving the covariance matrix. Nonetheless, for the model size tested in this paper, IDCS was found to be one to two orders of magnitude faster than MPS-based MCMC inversion for a posterior approximation of similar quality. Possible directions for future work include more sophisticated approaches to estimate the intractable likelihood, enhancing the efficiency of IDCS by exploring more sophisticated matrix multiplication techniques and use more elaborate simulation strategies (i.e. ending data conditioning when the data misfit is sufficiently low).

4.7 Appendix

4.7.1 Analytical posterior PDF for a multi-Gaussian field

Under the assumptions of linear physics and a Gaussian prior $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta)$, where $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$ are the mean and covariance of the property field of interest $\boldsymbol{\theta}$, there exists an analytical expression for the posterior PDF $p(\boldsymbol{\theta}|\mathbf{d})$. Considering normally distributed observational noise, the likelihood can be expressed as follows:

$$p(\mathbf{d}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{d}; \mathbf{G}\boldsymbol{\theta} + \mathbf{b}, \boldsymbol{\Sigma}_d). \quad (4.29)$$

A closed-form expression for the posterior is then obtained by (*Bishop and Nasrabadi, 2006*)

$$p(\mathbf{d}) = \mathcal{N}(\mathbf{d}; \mathbf{G}\boldsymbol{\mu}_\theta + \mathbf{b}, \boldsymbol{\Sigma}_d + \mathbf{G}\boldsymbol{\Sigma}_\theta\mathbf{G}^T) \quad (4.30)$$

$$p(\boldsymbol{\theta}|\mathbf{d}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\Sigma} \{ \mathbf{G}^T \boldsymbol{\Sigma}_d^{-1} (\mathbf{d} - \mathbf{b}) + \boldsymbol{\Sigma}_\theta^{-1} \boldsymbol{\mu}_\theta \}, \boldsymbol{\Sigma}), \quad (4.31)$$

where $\boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_\theta^{-1} + \mathbf{G}^T \boldsymbol{\Sigma}_d^{-1} \mathbf{G})^{-1}$.

4.7.2 Choice of QS parameters

There are two main hyper-parameters in the QS implementation used herein: (1) k_{cand} the number of candidates proposed by QS, sorted in ascending order of mismatch (in the original implementation of *Gravey and Mariethoz (2020)*, k_{rank} is a rank that represents the probability of sampling the sorted candidates) and (2) n the number of informed grid cells around the simulated location on which to calculate the misfit map. To determine the appropriate values for k_{cand} and n , we run several IDCS simulations for different k_{cand} and n values. We compare the different runs with respect to the SSIM and the WRMSE, as well as by visually inspecting the different realisations.

Although large values are usually recommended for n in MPS simulations (≥ 30 ; *Gravey and Mariethoz, 2020; Meerschman et al., 2013*), we found that n has little effect on the data fit and that better simulation quality (based on visual appearance and SSIM values) is achieved for small n (10) when the model parameters are continuous. For $n \geq 25$ the quality of the simulation decreases significantly and the realisations become noisy. For the binary channels model, a balance between the quality of the simulation (reflected in better visual appearance having less artefacts) and good model fit is found for $n = 25$. This difference in the optimal n between the continuous and binary models may be the result of the features' different characteristic sizes. It can be seen in Table 4.1 that the binary training images have greater correlation lengths, thus, the larger the radius within which neighbours contain relevant information.

As the value of k_{cand} increases, the WRMSE decreases and approaches 1.00 (see Table 4.3). With a larger k_{cand} , there is a larger chance that one of the proposals have a high likelihood. In most of the tested models, the model's SSIM values generally show improvement when k_{cand} is increased to 100. Further increasing k_{cand} to 500 enhances the SSIM only for the Gaussian model, possibly due to the greater variety of patterns and values present in a continuous Gaussian training image. It is important to note that raising k_{cand} too much can introduce undesired artefacts. This occurs because the algorithm is forced to generate more candidates, which given a finite training image, inevitably leads to a decrease in their quality.

Additionally, *Gravey and Mariethoz (2020)* indicated that using a weighting kernel can improve the quality of the QS simulation. As previously mentioned tests suggest that shorter distances are more important in continuous models, we performed tests with the weighting kernel $w = e^{-\alpha \|d\|_2}$ on the connected high-conductivity structures model, where d is the distance from the simulated pixel, α is the kernel parameter and $\|\cdot\|_2$ is the Euclidean distance. This

Table 4.3: Average model SSIM and data WRMSE for 50×50 pixels simulation given different k values and training image size of 500×500 ; these are calculated on 10 different simulations.

| k | n | TI type | SSIM | WRMSE |
|-----|-----|--|------|-------|
| 10 | 10 | Gaussian | 0.49 | 1.02 |
| 25 | 10 | | 0.49 | 1.01 |
| 50 | 10 | | 0.51 | 1.00 |
| 100 | 10 | | 0.51 | 1.00 |
| 500 | 10 | | 0.52 | 1.00 |
| 10 | 10 | Connected high- conductivity structures | 0.47 | 1.04 |
| 25 | 10 | | 0.56 | 1.02 |
| 50 | 10 | | 0.50 | 1.02 |
| 100 | 10 | | 0.54 | 1.02 |
| 500 | 10 | | 0.49 | 1.01 |
| 10 | 25 | Channels | 0.66 | 4.14 |
| 25 | 25 | | 0.84 | 1.81 |
| 50 | 25 | | 0.89 | 1.05 |
| 100 | 25 | | 0.89 | 1.05 |
| 500 | 25 | | 0.89 | 1.04 |

Table 4.4: Average model SSIM and data WRMSE of 10 simulations given a connected high-conductivity structures model of size 50×50 pixels for different α values, $k = 100$ and $n = 10$.

| α | SSIM | WRMSE |
|----------|------|-------|
| 0 | 0.54 | 1.02 |
| 0.03 | 0.53 | 1.02 |
| 0.3 | 0.48 | 1.03 |

kernel gives more weight to closer neighbours as the α increases. Nonetheless, this type of kernel did not lead to improvements for our considered examples (see Table 4.4).

4.7.3 Sensitivity matrix: binary channelised subsurface model

When attempting to approximate the posterior of the binary channels subsurface model (Fig. 4.10a) using linear physics, we notice anomalous simulations. This could be attributed to the Gaussian approximation involved in computing the kriging mean and covariance to approximate the likelihood. This problem worsens in the presence of non-linear physics, as the Gaussian approximation is combined with the insufficient coverage of the channels by the rays (Fig. 4.10b), leading to inadequate constraints from the available data and as a consequence to an uninformative likelihood for these grid cells.

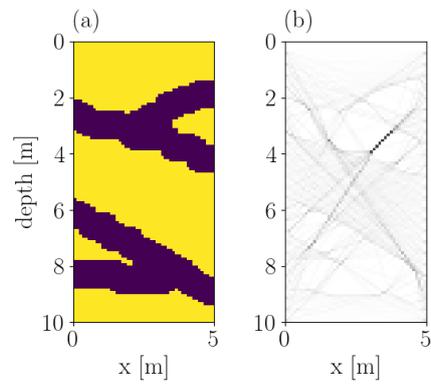


Figure 4.10: True sensitivity associated with the binary channels subsurface model in Section 4.4.2. (a) is the reference model and (b) is the ray path given the reference model.

Chapter 5

Conclusions and outlook

5.1 Conclusions

Sampling techniques used to estimate intractable posterior probability distributions for high-dimensional problems often suffer from long computation times. Given the significant impact of the repeated computation of the forward response on computation time, as it is necessary to compare model proposals, there are several ways to reduce the computational burden. It can be achieved by either simplifying the forward model, thereby reducing the time required for computing one forward response, or by the overall number of forward response computations needed. This thesis offers several approaches that can mitigate the computational burden and optimise the overall efficiency of the inversion process. These approaches adopt techniques from the fields of deep learning and geostatistics. They are either integrated into probabilistic, well-established algorithms like Metropolis-Hastings, allowing for the incorporation of modelling errors, or serve as powerful inversion tools outside standard geophysical inversion frameworks.

Surrogate models offer a cost-effective and simplified means to compute the forward response. The utilisation of computationally-inexpensive forward models becomes particularly significant when repetitive forward evaluations are necessary. Nonetheless, one must account for the error arising due to simplifications. In chapter 2, we introduced an approach that enables the use of a surrogate model as a substitute to a more accurate yet computationally intensive forward model. In our tests we used a simple straight-ray solver to model a crosshole GPR travel-time tomography. This solver is significantly faster than its high-fidelity alternative solvers: Eikonal and FDTD. To avoid bias and over-confident solutions we account for the modelling error due to the simplified physics through a generative adversarial network. Generating modelling errors through a deep generative model offer two key advantages. First, it eliminates the need for assumptions regarding the statistics of the error model, as the DGM is trained on a collection of modelling error realisations that represent the differences between the high-fidelity and surrogate models. Second, it enables the encoding of modelling errors within a low-dimensional space, thereby avoiding the inclusion of numerous additional inferred parameters. In comparison to the results obtained from running a MCMC inversion without accounting for modelling errors, or alternatively, accounting for the errors by inflating the likelihood error covariance, our approach yielded solutions

that demonstrated low bias and better data fit. Moreover, as the modelling error is inferred during inversion, one obtains a posterior representation of the modelling error for a given experiment.

Efficient optimisation together with low-dimensional parameterization can effectively and efficiently locate the solution to the inversion problem, hence, reduce the number of forward responses required and consequently the overall computation time. In chapter 3, we train inverse autoregressive flows through variational Bayesian inference to infer the posterior distribution of the low-dimensional latent space encoded by a deep generative model. This approach uses random sampling and gradient optimisation to approximate the posterior on the latent space of either a GAN or a VAE. The reduction in the number of inferred parameters coupled with efficient gradient-based optimisation led to a speedup of seven times compared to MCMC inversion. Although deterministic gradient-based approaches were unsuccessful in performing inversion that involves the highly-nonlinear GAN transformation, training the inverse autoregressive flows led to a successful reproduction of the reference model. While the posterior approximations in the low-dimensional latent space of the GAN were broad and did not encompass the true value, the posterior in the low-dimensional latent space of the VAE not only included the true value but also exhibited comparable uncertainty quantification as obtained from MCMC sampling.

The use of MPS simulations offers an efficient means of generating geologically-realistic subsurface models that can be conditioned on known (hard) data points. In chapter 4, a novel methodology was presented to condition MPS simulations on indirect (linear) geophysical data, allowing for the approximation of the posterior distribution without the need for time-consuming sampling frameworks. Our synthetic case-studies involve three distinct subsurface models with their priors being sampled using the QS algorithm. The QS simulations were conditioned in a sequential manner on GPR crosshole tomography data generated using a straight-ray solver. By considering each realisation obtained from the conditional QS simulation as a draw from the posterior distribution, we were able to approximate the posterior distribution using multiple realisations. The posterior approximations obtained through this approach were found to be on par with approximations obtained via a Gibbs sampler, but with at least one order of magnitude difference in computation time in favour of the conditional QS simulations. This approach has potential uses in field applications, where a linear forward model adequately represents the physical process and where two-point statistics fail to provide a suitable representation of the subsurface model.

The latter two methods also provide parallelism capabilities that scale well with available computational resources, allowing for the simultaneous execution of multiple realisations. Moreover, all three approaches use either deep generative models or MPS to sample the prior. Such representations ensure geologically-realistic subsurface models and reduces the search space compared to more general prior models.

5.2 Limitations and outlook

The approaches introduced in this thesis offer substantial improvements in terms of computational cost; however, it is essential to acknowledge their limitations. The approach proposed in Chapter 2 was able to account for modelling errors that are a result of using a simplified forward solver. It is important to note that numerical models are abstractions of real physical processes, and as such, they introduce a certain level of error into their predictions. The errors accounted for in our approach do not capture the discrepancy between the forward solver and the true physics, that is for the most part impractical to accurately quantify. This means that solutions obtained from inverting real field data are inevitably biased, even when the discrepancy between high- and low-fidelity solvers is perfectly represented. Furthermore, the subsurface model and the modelling error are learned by two separate GANs, implying that any prior correlations are ignored. Our approach could be extended by encoding the pairing of subsurface model and model error into a shared latent space on which inversion is performed. From the perspective of surrogate modelling, physics-informed neural networks (*Raissi et al., 2019*) offer a compelling advantage in obtaining efficient and accurate surrogate models (*Song et al., 2021a; Rasht-Behesht et al., 2022*). These networks incorporate the governing equations of the system as additional constraints during their optimisation process. By doing so, they seamlessly integrate domain-specific physics knowledge, leading to more robust and reliable predictions.

Incorporating highly-nonlinear transformations, such as those learned in DGMs, to represent model and model error parameters can present challenges for deterministic inversion algorithms and may result in convergence issues in probabilistic algorithms, as observed in Section 2.3.2. While the neural-transport approach (Chapter 3) has shown success in handling highly nonlinear transformations, future research could explore the use of normalising flows with surjective transformations or multi-scale architectures and investigate their applicability to geophysical inverse problems (*Das et al., 2019; Nielsen et al., 2020*). In the multi-scale architecture for generative flows, the importance of each dimension (an input parameter) is taken into account based on its contribution to the overall log-probability of the target density. This leads to a dimension factorization process, where certain dimensions are factored out, while others undergo more flow layers. Although this architecture does not provide a dimensionality reduction in the strict sense, it makes generative flows computationally efficient. On the other hand, surjective transformations overcome the limitation of requiring equal dimensional sizes in flow models. Surjective transformations are deterministic in one direction and stochastic in its inverse, therefore, enabling changes in dimensionality between the input and output variables. Both of these approaches eliminate the need for an intermediate DGM that is trained separately from the flow-based model and make flow-based generative models even more computationally efficient.

Other promising directions for geophysical inversions, is to implement DGMs as a full inversion framework (*Laloy et al., 2021; McAilely and Li, 2021*), where the reconstructed model is the output given input data. The accuracy of any framework that relies on DGMs, however, depends on the generalisation and goodness of the trained model and as highlighted in Section 1.1, it heavily relies on the availability of data, which in geophysics, and geoscience in general (*Karpatne et al., 2018*), might not always be readily available or easy to compute.

Future efforts should, therefore, be dedicated to increasing data availability in geophysics and further improve the capabilities of DGMs to learn high-quality representations of spatial models and physical processes.

Although gradient-based variational optimisation techniques such as in Chapter 3 are efficient, they require a fully differentiable forward response and prior. In the case of DGMs, this imposes a limitation on subsurface models with strongly varying features, such as in channelised models. This limitation can have a strong impact in some applications, for instance modelling fluid flow. On the contrary, differentiating the forward model can be accomplished either by implementing the forward solver within a framework that supports automatic differentiation (*Margossian, 2019*) or by manually performing the differentiation by, for instance, the adjoint-state method (*Plessix, 2006*). Manually differentiating the forward response can be laborious and challenging, particularly for highly complex multi-physics forward models. Furthermore, many legacy forward solvers are typically implemented in programming languages in which automatic differentiation is not readily available. Future research aimed at developing machine learning-based surrogate models capable of accurately capturing the forward response holds great potential in providing practical solutions. These models are fully differentiable and are implemented in libraries supporting automatic-differentiation, making them easily integratable into inversion schemes that necessitate model differentiation.

The approach introduced in Chapter 4 was so far tested only on cases where the forward response can be described in terms of linear physics. Nevertheless, the application of this method would offer significant advantages for nonlinear forward responses, which typically incur a higher computational burden when utilised within a sampling framework. A possible partial extension to nonlinear problems would involve a linearization of the forward response. Alternatively, one can adopt the idea of surrogate modelling with covariance inflation of the likelihood error (*Hansen et al., 2014*), thereby, reducing the computational cost and eliminating the need to linearize the forward response. The simulations in our case studies, were performed with random simulation paths. It is possible that using preferential paths that visit locations that are better constrained by the data first, might help improve the quality of the model realisations. A prominent limitation in our approach is the non-linear growth in computation as the size of the simulation grid increases. This is attributed to the covariance matrix multiplication, which implies that the method remains computationally favourable only in 2D grids and up to a certain grid size. Future advancements in matrix multiplication algorithms hold the potential to reduce the computational cost associated with the size of the covariance matrix, thereby enhancing the applicability of this approach to larger grids.

The methodologies presented in Chapter 3 and 4 could serve as preliminary approximations to MCMC sampling. By initially approximating the posterior using these efficient techniques, we can initialise the MCMC chains with samples from these approximate posteriors and further improve the posterior approximation. One notable advantage of such an approach is that it allows us to start MCMC sampling with posterior samples that are likely to have high probability (*Hoffman et al., 2019*), thereby significantly reducing the time required to adequately sample the posterior distribution. As a result, such a hybrid approach strikes a balance between computational efficiency and accurate posterior representation. Future research could explore different strategies to build such hybrid approaches.

All approaches introduced in this thesis were tested within a GPR crosshole setting. In order to further assess the performance of these approaches and their generalisation, they should also be tested and evaluated for different geophysical problems, for instance, seismics, gravity and magnetics and eventually tested on real data. One major drawback of DGMs in that context is that they are specifically trained for a particular problem, and any modification to the prior or the experimental design would necessitate re-training the DGM, which can be a time-consuming process in itself. Additionally, these generative models have to be general enough to be able to accurately represent real data. Possible avenues for further research can be built on recent developments in generative models, such as, training a single DGM on multiple TIs, learning multiple textures and using conditional GANs to generate environment specific realisations (*Mirza and Osindero, 2014; Bergmann et al., 2017; Lopez-Alvis et al., 2022*).

Bibliography

- Alcolea, A., and P. Renard (2010), Blocking moving window algorithm: Conditioning multiple-point simulations to hydrogeological data, *Water Resources Research*, 46(8), W08511.
- Alumbaugh, D. L., and G. A. Newman (2000), Image appraisal for 2-D and 3-D electromagnetic inversion, *Geophysics*, 65(5), 1455–1467.
- Arjovsky, M., S. Chintala, and L. Bottou (2017), Wasserstein Generative Adversarial Networks, in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 214–223, PMLR.
- Azevedo, L., and A. Soares (2017), *Geostatistical Methods for Reservoir Geophysics*, Springer.
- Ba, J., M. A. Erdogdu, M. Ghassemi, T. Suzuki, S. Sun, D. Wu, and T. Zhang (2019), Towards characterizing the high-dimensional bias of kernel-based particle inference algorithms, 2nd Symposium on Advances in Approximate Bayesian Inference. Vancouver, Canada.
- Bagtzoglou, A. C., and J. Atmadja (2005), Mathematical methods for hydrologic inversion: The case of pollution source identification, *Water Pollution: Environmental Impact Assessment of Recycled Wastes on Surface and Ground Waters; Engineering Modeling and Sustainability*, pp. 65–96.
- Bao, J., L. Li, and A. Davis (2022), Variational autoencoder or generative adversarial networks? a comparison of two deep learning methods for flow and transport data assimilation, *Mathematical Geosciences*, 54(6), 1017–1042.
- Barrash, W., and T. Clemo (2002), Hierarchical geostatistics and multifacies systems: Boise Hydrogeophysical Research Site, Boise, Idaho, *Water Resources Research*, 38(10), 1196.
- Bergen, K. J., P. A. Johnson, M. V. de Hoop, and G. C. Beroza (2019), Machine learning for data-driven discovery in solid earth geoscience, *Science*, 363(6433), eaau0323.
- Bergmann, U., N. Jetchev, and R. Vollgraf (2017), Learning texture manifolds with the periodic spatial gan, *arXiv preprint arXiv:1705.06566*.
- Bingham, E., J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. A. Szerlip, P. Horsfall, and N. D. Goodman (2019), Pyro: Deep universal probabilistic programming, *J. Mach. Learn. Res.*, 20, 28:1–28:6.
- Bishop, C. M., and N. M. Nasrabadi (2006), *Pattern Recognition and Machine Learning*, Springer.

- Blakely, R. J. (1996), *Potential Theory in Gravity and Magnetic Applications*, Cambridge university press.
- Blei, D. M., A. Kucukelbir, and J. D. McAuliffe (2017), Variational inference: A review for statisticians, *Journal of the American Statistical Association*, 112(518), 859-877.
- Bond-Taylor, S., A. Leach, Y. Long, and C. G. Willcocks (2022), Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11), 7327-7347.
- Born, M., and E. Wolf (2013), *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, Elsevier.
- Bosch, M., T. Mukerji, and E. F. Gonzalez (2010), Seismic inversion for reservoir properties combining statistical rock physics and geostatistics: A review, *Geophysics*, 75(5), 75A165-75A176.
- Braak, C. J. T. (2006), A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces, *Statistics and Computing*, 16, 239-249.
- Brooks, S., A. Gelman, G. Jones, and X.-L. Meng (2011), *Handbook of Markov Chain Monte Carlo*, CRC press.
- Brunetti, C., and N. Linde (2018), Impact of petrophysical uncertainty on Bayesian hydrogeophysical inversion and model selection, *Advances in Water Resources*, 111, 346-359.
- Brunetti, C., N. Linde, and J. A. Vrugt (2017), Bayesian model selection in hydrogeophysics: Application to conceptual subsurface models of the South Oyster Bacterial Transport Site, Virginia, USA, *Advances in Water Resources*, 102, 127-141.
- Brunetti, C., M. Bianchi, G. Pirot, and N. Linde (2019), Hydrogeological model selection among complex spatial priors, *Water Resources Research*, 55(8), 6729-6753.
- Brynjarsdóttir, J., and A. O-Hagan (2014), Learning about physical parameters: The importance of model discrepancy, *Inverse Problems*, 30(11), 114007.
- Calvetti, D., O. Ernst, and E. Somersalo (2014), Dynamic updating of numerical model discrepancy using sequential sampling, *Inverse Problems*, 30(11), 114019.
- Canchumuni, S. W., A. A. Emerick, and M. A. C. Pacheco (2019), Towards a robust parameterization for conditioning facies models using deep variational autoencoders and ensemble smoother, *Computers & Geosciences*, 128, 87-102.
- Chan, S., and A. H. Elsheikh (2019), Parametric generation of conditional geological realizations using generative neural networks, *Computational Geosciences*, 23(5), 925-952.
- Chevalier, C., D. Ginsbourger, and X. Emery (2014), Corrected kriging update formulae for batch-sequential data assimilation, in *Mathematics of Planet Earth*, edited by E. Pardo-Igúzquiza, C. Guardiola-Albert, J. Heredia, L. Moreno-Merino, J. J. Durán, and J. A. Vargas-Guzmán, pp. 119-122, Springer Berlin Heidelberg.

- Chevalier, C., X. Emery, and D. Ginsbourger (2015), Fast update of conditional simulation ensembles, *Mathematical Geosciences*, 47(7), 771–789.
- Chevitarese, D., D. Szwarcman, R. M. D. Silva, and E. V. Brazil (2018), Seismic facies segmentation using deep learning, *AAPG Annual and Exhibition*.
- Chilès, J.-P., and N. Desassis (2018), Fifty years of kriging, *Handbook of Mathematical Geosciences: Fifty Years of IAMG*, pp. 589–612.
- Cordua, K. S., T. M. Hansen, and K. Mosegaard (2012), Monte Carlo full-waveform inversion of crosshole GPR data using multiple-point geostatistical a priori information, *Geophysics*, 77(2), H19–H31.
- Cowles, M. K., and B. P. Carlin (1996), Markov chain Monte Carlo convergence diagnostics: a comparative review, *Journal of the American Statistical Association*, 91(434), 883–904.
- Cui, T., C. Fox, and M. J. O’Sullivan (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.
- Das, H. P., P. Abbeel, and C. J. Spanos (2019), Likelihood contribution based multi-scale architecture for generative flows, *arXiv preprint arXiv:1908.01686*.
- Dejtrakulwong, P., T. Mukerji, and G. Mavko (2012), Using kernel principal component analysis to interpret seismic signatures of thin shaly-sand reservoirs, in *SEG Technical Program Expanded Abstracts 2012*, pp. 1–5, Society of Exploration Geophysicists.
- Deutsch, C. V. (1992), *Annealing Techniques Applied to Reservoir Modeling and the Integration of Geological and Engineering (Well Test) Data*, stanford university.
- Dijkstra, E. W. (1959), A note on two problems in connexion with graphs, *Numerische Mathematik*, 1, 269–271.
- Dinh, L., D. Krueger, and Y. Bengio (2014), NICE: Non-linear independent components estimation, *arXiv preprint arXiv:1410.8516*.
- Dinh, L., J. Sohl-Dickstein, and S. Bengio (2016), Density estimation using real NVP, *arXiv preprint arXiv:1605.08803*.
- Dramsch, J. S. (2020), 70 years of machine learning in geoscience in review, *Advances in Geophysics*, 61, 1–55.
- Duane, S., A. D. Kennedy, B. J. Pendleton, and D. Roweth (1987), Hybrid Monte Carlo, *Physics Letters B*, 195(2), 216–222.
- Dumoulin, V., and F. Visin (2016), A guide to convolution arithmetic for deep learning, *arXiv preprint arXiv:1603.07285*.
- Dupont, E., T. Zhang, P. Tilke, L. Liang, and W. Bailey (2018), Generating realistic geology conditioned on physical measurements with generative adversarial networks, *arXiv preprint arXiv:1802.03065*.

- Emery, X. (2009), The kriging update equations and their application to the selection of neighboring data, *Computational Geosciences*, 13(3), 269.
- Ernst, J. R., A. G. Green, H. Maurer, and K. Holliger (2007), Application of a new 2D time-domain full-waveform inversion scheme to crosshole radar data, *Geophysics*, 72(5), J53–J64.
- Feng, R., D. Grana, and N. Balling (2021), Variational inference in Bayesian neural network for well-log prediction, *Geophysics*, 86(3), M91–M99.
- Friel, N., and A. N. Pettitt (2008), Marginal likelihood estimation via power posteriors, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3), 589–607.
- Friel, N., and J. Wyse (2012), Estimating the evidence—a review, *Statistica Neerlandica*, 66(3), 288–308.
- Gallet, A., S. Rigby, T. Tallman, X. Kong, I. Hajirasouliha, A. Liew, D. Liu, L. Chen, A. Hauptmann, and D. Smyl (2022), Structural engineering from an inverse problems perspective, *Proceedings of the Royal Society A*, 478(2257), 20210526.
- Gelman, A. (1992), Iterative and non-iterative simulation algorithms, *Computing Science and Statistics*, 24, 433–438.
- Gelman, A., and D. B. Rubin (1992), Inference from iterative simulation using multiple sequences, *Statistical Science*, 7(4), 457–472.
- Gelman, A., G. O. Roberts, W. R. Gilks, et al. (1996), Efficient Metropolis jumping rules, *Bayesian Statistics*, 5(599-608), 42.
- Gelman, A., J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin (2013), *Bayesian Data Analysis*, CRC press.
- Geman, S., and D. Geman (1984), Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6), 721-741.
- Germain, M., K. Gregor, I. Murray, and H. Larochelle (2015), Made: Masked autoencoder for distribution estimation, in *International Conference on Machine Learning*, vol. 37, edited by F. Bach and D. Blei, pp. 881–889, PMLR.
- Giannakis, I., A. Giannopoulos, and C. Warren (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.
- Giroux, B., and B. Larouche (2013), Task-parallel implementation of 3D shortest path raytracing for geophysical applications, *Computers & Geosciences*, 54, 130–141.
- Gómez-Hernández, J. J., and X.-H. Wen (1998), To be or not to be multi-gaussian? a reflection on stochastic hydrogeology, *Advances in Water Resources*, 21(1), 47–61.
- Good, I. J. (1952), Rational decisions, *Journal of the Royal Statistical Society: Series B (Methodological)*, 14(1), 107-114.

- Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014), Generative adversarial nets, *Advances in Neural Information Processing Systems*, 27.
- Goodfellow, I., Y. Bengio, and A. Courville (2016), *Deep Learning*, MIT Press, <http://www.deeplearningbook.org>.
- Gravey, M., and G. Mariethoz (2020), Quicksampling v1. 0: a robust and simplified pixel-based multiple-point simulation approach, *Geoscientific Model Development*, 13(6), 2611–2630.
- Gray, R. M. (2011), *Entropy and Information Theory*, Springer Science & Business Media.
- Green, P. J. (1995), Reversible jump Markov chain Monte Carlo computation and Bayesian model determination, *Biometrika*, 82(4), 711–732.
- Griewank, A., and A. Walther (2008), *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, SIAM.
- Guardiano, F. B., and R. M. Srivastava (1993), Multivariate geostatistics: beyond bivariate moments, in *Geostatistics Tróia'92: Volume 1*, edited by A. Soares, pp. 133–144, Springer.
- Guo, J., Y. Li, M. W. Jessell, J. Giraud, C. Li, L. Wu, F. Li, and S. Liu (2021), 3D geological structure inversion from Noddy-generated magnetic data using deep learning methods, *Computers & Geosciences*, 149, 104701.
- Haario, H., E. Saksman, and J. Tamminen (2001), An adaptive Metropolis algorithm, *Bernoulli*, pp. 223–242.
- Haario, H., M. Laine, A. Mira, and E. Saksman (2006), DRAM: efficient adaptive MCMC, *Statistics and Computing*, 16, 339–354.
- Hammersley, J. M., and D. C. Handscomb (1964), General principles of the Monte Carlo method, in *Monte Carlo Methods*, pp. 50–75, Springer Netherlands.
- Hansen, T., K. Mosegaard, and K. Cordua (2010), Sampling informative/complex a priori probability distributions using gibbs sampling assisted by sequential simulation, in *Proceedings of the 14th Annual Conference of the International Association for Mathematical Geoscience*.
- Hansen, T. M., and K. Mosegaard (2008), VISIM: Sequential simulation for linear inverse problems, *Computers & Geosciences*, 34(1), 53–76.
- Hansen, T. M., A. G. Journel, A. Tarantola, and K. Mosegaard (2006), Linear inverse Gaussian theory and geostatistics, *Geophysics*, 71(6), R101–R111.
- Hansen, T. M., K. S. Cordua, and K. Mosegaard (2012), Inverse problems with non-trivial priors: efficient solution through sequential Gibbs sampling, *Computational Geosciences*, 16(3), 593–611.
- Hansen, T. M., K. S. Cordua, B. H. Jacobsen, and K. Mosegaard (2014), Accounting for imperfect forward modeling in geophysical inverse problems - Exemplified for crosshole tomography, *Geophysics*, 79(3), H1-H21.

- Hansen, T. M., K. Mosegaard, K. S. Cordua, et al. (2018), Multiple point statistical simulation using uncertain (soft) conditional data, *Computers & Geosciences*, 114, 1–10.
- Hastings, W. K. (1970), Monte Carlo sampling methods using Markov chains and their applications, *Biometrika*, 57(1), 97-109.
- Heusel, M., H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter (2017), GANs trained by a two time-scale update rule converge to a local Nash equilibrium, in *Advances in Neural Information Processing Systems*, vol. 30, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, pp. 6629–6640, Curran Associates, Inc.
- Hinton, G. E., and R. R. Salakhutdinov (2006), Reducing the dimensionality of data with neural networks, *Science*, 313(5786), 504–507.
- Hitchcock, D. B. (2003), A history of the Metropolis-Hastings algorithm, *The American Statistician*, 57(4), 254–257.
- Hoffman, M., P. Sountsov, J. V. Dillon, I. Langmore, D. Tran, and S. Vasudevan (2019), Neutralizing bad geometry in Hamiltonian Monte Carlo using neural transport, *arXiv preprint arXiv:1903.03704*.
- Hoffman, M. D., and A. Gelman (2014), The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo, *Journal of Machine Learning Research*, 15, 1351–1381.
- Hoffman, M. D., D. M. Blei, C. Wang, and J. Paisley (2013), Stochastic variational inference, *Journal of Machine Learning Research*, 14, 1303–1347.
- Hou, X., L. Shen, K. Sun, and G. Qiu (2017), Deep feature consistent variational autoencoder, in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1133–1141.
- Hu, G., Z. Hu, J. Liu, F. Cheng, and D. Peng (2020), Seismic fault interpretation using deep learning-based semantic segmentation method, *IEEE Geoscience and Remote Sensing Letters*, 19, 1–5.
- Hu, L., and T. Chuginova (2008), Multiple-point geostatistics for modeling subsurface heterogeneity: A comprehensive review, *Water Resources Research*, 44(11), W11413.
- Hu, Q., D. Grana, and K. A. Innanen (2023), Feasibility of seismic time-lapse monitoring of CO₂ with rock physics parametrized full waveform inversion, *Geophysical Journal International*, 233(1), 402–419.
- Hubbard, S. S., and Y. Rubin (2000), Hydrogeological parameter estimation using geophysical data: a review of selected techniques, *Journal of Contaminant Hydrology*, 45(1-2), 3–34.
- Hunziker, J., E. Laloy, and N. Linde (2019), Bayesian full-waveform tomography with application to crosshole ground penetrating radar data, *Geophysical Journal International*, 218(2), 913–931.
- Irving, J., and R. Knight (2006), Numerical modeling of ground-penetrating radar in 2-D using MATLAB, *Computers & Geosciences*, 32(9), 1247-1258.

- Irving, J. D., and R. J. Knight (2005), Effect of antennas on velocity estimates obtained from crosshole GPR data, *Geophysics*, 70(5), K39–K42.
- Jankovic, I., M. Maghrebi, A. Fiori, and G. Dagan (2017), When good statistical models of aquifer heterogeneity go right: The impact of aquifer permeability structures on 3d flow and transport, *Advances in Water Resources*, 100, 199–211.
- Jensen, B. B., T. M. Hansen, L. Nielsen, K. S. Cordua, N. Tuxen, A. Tsitonaki, and M. C. Looms (2022), Accounting for modeling errors in linear inversion of crosshole ground-penetrating radar amplitude data: Detecting sand in clayey till, *Journal of Geophysical Research: Solid Earth*, 127(10), e2022JB024666.
- Jetchev, N., U. Bergmann, and R. Vollgraf (2016), Texture synthesis with spatial generative adversarial networks, *arXiv preprint arXiv:1611.08207*.
- Jin, Z. L., Y. Liu, and L. J. Durlofsky (2020), Deep-learning-based surrogate model for reservoir simulation with time-varying well controls, *Journal of Petroleum Science and Engineering*, 192, 107273.
- Jóhannsson, Ó. D., and T. M. Hansen (2023), Multiple-point statistics and non-colocational soft data integration, *Computers & Geosciences*, 172, 105280.
- Jol, H. M. (2008), *Ground Penetrating Radar Theory and Applications*, elsevier.
- Jordan, M. I., Z. Ghahramani, T. S. Jaakkola, and L. K. Saul (1999), An introduction to variational methods for graphical models, *Machine Learning*, 37(2), 183–233.
- Joshi, D., A. K. Patidar, A. Mishra, A. Mishra, S. Agarwal, A. Pandey, B. K. Dewangan, and T. Choudhury (2021), Prediction of sonic log and correlation of lithology by comparing geophysical well log data using machine learning principles, *GeoJournal*, pp. 1–22.
- Kaipio, J., and E. Somersalo (2007), Statistical inverse problems: Discretization, model reduction and inverse crimes, *Journal of Computational and Applied Mathematics*, 198(2), 493–504.
- Kalos, M. H., and P. A. Whitlock (2009), *Monte Carlo Methods*, John Wiley & Sons.
- Karpatne, A., G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar (2017), Theory-guided data science: A new paradigm for scientific discovery from data, *IEEE Transactions on Knowledge and Data Engineering*, 29(10), 2318–2331.
- Karpatne, A., I. Ebert-Uphoff, S. Ravela, H. A. Babaie, and V. Kumar (2018), Machine learning for the geosciences: Challenges and opportunities, *IEEE Transactions on Knowledge and Data Engineering*, 31(8), 1544–1554.
- Kennedy, M. C., and A. O’Hagan (2001), Bayesian calibration of computer models, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3), 425–464.
- Kim, Y., and N. Nakata (2018), Geophysical inversion versus machine learning in inverse problems, *The Leading Edge*, 37(12), 894–901.

- Kingma, D. P., and J. Ba (2014), ADAM: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*.
- Kingma, D. P., and P. Dhariwal (2018), GLOW: Generative flow with invertible 1x1 convolutions, *Advances in Neural Information Processing Systems*, 31.
- Kingma, D. P., and M. Welling (2014), Auto-encoding variational Bayes, *arXiv preprint arXiv:1312.6114*.
- Kingma, D. P., T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling (2016), Improved variational inference with inverse autoregressive flow, *Advances in Neural Information Processing Systems*, 29, 4743–4751.
- Kobyzev, I., S. J. Prince, and M. A. Brubaker (2021), Normalizing flows: An introduction and review of current methods, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11), 3964-3979.
- Konaté, A. A., H. Pan, H. Ma, X. Cao, Y. Yevenyo Ziggah, M. Oloo, and N. Khan (2015), Application of dimensionality reduction technique to improve geophysical log data classification performance in crystalline rocks, *Journal of Petroleum Science and Engineering*, 133, 633-645.
- Köpke, C., J. Irving, and A. H. Elsheikh (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., J. Irving, and D. Roubinet (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.
- Kucukelbir, A., D. Tran, R. Ranganath, A. Gelman, and D. M. Blei (2017), Automatic differentiation variational inference, *Journal of Machine Learning Research*, 18(1), 430–474.
- Kullback, S., and R. A. Leibler (1951), On information and sufficiency, *The Annals of Mathematical Statistics*, 22(1), 79–86.
- Laloy, E., and J. A. Vrugt (2012), High-dimensional posterior exploration of hydrologic models using multiple-try DREAM_(ZS) and high-performance computing, *Water Resources Research*, 48(1), W01526.
- Laloy, E., N. Linde, D. Jacques, and G. Mariethoz (2016), Merging parallel tempering with sequential geostatistical resampling for improved posterior exploration of high-dimensional subsurface categorical fields, *Advances in Water Resources*, 90, 57-69.
- Laloy, E., R. Héroult, J. Lee, D. Jacques, and N. Linde (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., R. Héroult, D. Jacques, and N. Linde (2018), Training-image based geostatistical inversion using a spatial generative adversarial neural network, *Water Resources Research*, 54(1), 381–406.

- Laloy, E., N. Linde, C. Ruffino, R. Hérault, G. Gasso, and D. Jacques (2019), Gradient-based deterministic inversion of geophysical data with generative adversarial networks: Is it feasible?, *Computers & Geosciences*, 133, 104333.
- Laloy, E., N. Linde, and D. Jacques (2021), Approaching geoscientific inverse problems with vector-to-image domain transfer networks, *Advances in Water Resources*, 152, 103917.
- Lange, K., J. Frydendall, K. S. Cordua, T. M. Hansen, Y. Melnikova, and K. Mosegaard (2012), A frequency matching method: solving inverse problems by use of geologically realistic prior information, *Mathematical geosciences*, 44, 783–803.
- Lazaratos, S. K., and B. P. Marion (1997), Crosswell seismic imaging of reservoir changes caused by CO₂ injection, *The Leading Edge*, 16(9), 1300–1308.
- Le, H., and A. Borji (2017), What are the receptive, effective receptive, and projective fields of neurons in convolutional neural networks?, *arXiv preprint arXiv:1705.07049*.
- Lelièvre, P. G., D. W. Oldenburg, and N. C. Williams (2009), Integrating geological and geophysical data through advanced constrained inversions, *Exploration Geophysics*, 40(4), 334–341.
- Lelièvre, P. G., C. G. Farquharson, and C. A. Hurich (2012), Joint inversion of seismic traveltimes and gravity data on unstructured grids with application to mineral exploration, *Geophysics*, 77(1), K1–K15.
- Levy, S., J. Hunziker, E. Laloy, J. Irving, and N. Linde (2022), Using deep generative neural networks to account for model errors in Markov chain Monte Carlo inversion, *Geophysical Journal International*, 228(2), 1098–1118.
- Linde, N., P. Renard, T. Mukerji, and J. Caers (2015a), Geological realism in hydrogeological and geophysical inverse modeling: A review, *Advances in Water Resources*, 86, 86–101.
- Linde, N., T. Lochbühler, M. Dogan, and R. L. Van Dam (2015b), Tomogram-based comparison of geostatistical models: Application to the Macrodispersion Experiment (MADE) site, *Journal of Hydrology*, 531, 543–556.
- Linde, N., D. Ginsbourger, J. Irving, F. Nobile, and A. Doucet (2017), On uncertainty quantification in hydrogeology and hydrogeophysics, *Advances in Water Resources*, 110, 166–181.
- Liu, Q., and D. Wang (2016), Stein variational gradient descent: a general purpose Bayesian inference algorithm, *Advances in Neural Information Processing Systems*, 29, 2378–2386.
- Lochbühler, T., G. Pirot, J. Straubhaar, and N. Linde (2014), Conditioning of multiple-point statistics facies simulations to tomographic images, *Mathematical Geosciences*, 46, 625–645.
- Lopez-Alvis, J., E. Laloy, F. Nguyen, and T. Hermans (2021), Deep generative models in inversion: The impact of the generator’s nonlinearity and development of a new approach based on a variational autoencoder, *Computers & Geosciences*, 152, 104762.

- Lopez-Alvis, J., F. Nguyen, M. Looms, and T. Hermans (2022), Geophysical inversion using a variational autoencoder to model an assembled spatial prior uncertainty, *Journal of Geophysical Research: Solid Earth*, 127(3), e2021JB022581.
- Luengo, D., L. Martino, M. Bugallo, V. Elvira, and S. Särkkä (2020), A survey of Monte Carlo methods for parameter estimation, *EURASIP Journal on Advances in Signal Processing*, 2020(1), 1–62.
- Maasackers, J. D., D. J. Jacob, M. P. Sulprizio, T. R. Scarpelli, H. Nesser, J. Sheng, Y. Zhang, X. Lu, A. A. Bloom, K. W. Bowman, et al. (2021), 2010–2015 north american methane emissions, sectoral contributions, and trends: a high-resolution inversion of gosat observations of atmospheric methane, *Atmospheric Chemistry and Physics*, 21(6), 4339–4356.
- Malinverno, A., and V. A. Briggs (2004), Expanded uncertainty quantification in inverse problems: Hierarchical Bayes and empirical Bayes, *Geophysics*, 69(4), 1005–1016.
- Mamalakis, A., I. Ebert-Uphoff, and E. A. Barnes (2022), Neural network attribution methods for problems in geoscience: A novel synthetic benchmark dataset, *Environmental Data Science*, 1, e8.
- Margossian, C. C. (2019), A review of automatic differentiation and its efficient implementation, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(4), e1305.
- Mariethoz, G. (2018), When should we use multiple-point geostatistics?, *Handbook of Mathematical Geosciences: Fifty Years of IAMG*, pp. 645–653.
- Mariethoz, G., and J. Caers (2014), *Multiple-Point Geostatistics: Stochastic Modeling with Training Images*, John Wiley & Sons.
- Mariethoz, G., P. Renard, and J. Straubhaar (2010a), The direct sampling method to perform multiple-point geostatistical simulations, *Water Resources Research*, 46(11), W11536.
- Mariethoz, G., P. Renard, and J. Caers (2010b), Bayesian inverse problem and optimization with iterative spatial resampling, *Water Resources Research*, 46(11), W11530.
- Matheron, G. (1963), Principles of geostatistics, *Economic Geology*, 58(8), 1246–1266.
- McAliley, W. A., and Y. Li (2021), Machine learning inversion of geophysical data by a conditional variational autoencoder, in *First International Meeting for Applied Geoscience & Energy*, pp. 1460–1464, Society of Exploration Geophysicists.
- Meerschman, E., G. Pirot, G. Mariethoz, J. Straubhaar, M. Van Meirvenne, and P. Renard (2013), A practical guide to performing multiple-point statistical simulations with the Direct Sampling algorithm, *Computers & Geosciences*, 52, 307–324.
- Meju, M. A. (1994), *Geophysical Data Analysis: Understanding Inverse Problem Theory and Practice*, Society of Exploration Geophysicists.
- Menke, W. (2015), Review of the generalized least squares method, *Surveys in Geophysics*, 36, 1–25.

- Menke, W. (2018), *Geophysical Data Analysis: Discrete Inverse Theory*, Academic press.
- Metropolis, N., and S. Ulam (1949), The Monte Carlo method, *Journal of the American Statistical Association*, 44(247), 335–341.
- Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953), Equation of state calculations by fast computing machines, *The Journal of Chemical Physics*, 21(6), 1087–1092.
- Milledge, D. G., S. N. Lane, A. L. Heathwaite, and S. M. Reaney (2012), A Monte Carlo approach to the inverse problem of diffuse pollution risk in agricultural catchments, *Science of the Total Environment*, 433, 434–449.
- Mirza, M., and S. Osindero (2014), Conditional generative adversarial nets, *arXiv preprint arXiv:1411.1784*.
- Miyato, T., T. Kataoka, M. Koyama, and Y. Yoshida (2018), Spectral normalization for generative adversarial networks, in *International Conference on Learning Representations*.
- Mosavi, A., P. Ozturk, and K.-w. Chau (2018), Flood prediction using machine learning models: Literature review, *Water*, 10(11), 1536.
- Mosegaard, K., and M. Sambridge (2002), Monte Carlo analysis of inverse problems, *Inverse Problems*, 18(3), R29.
- Mosegaard, K., and A. Tarantola (1995), Monte Carlo sampling of solutions to inverse problems, *Journal of Geophysical Research: Solid Earth*, 100(B7), 12431–12447.
- Mosser, L., O. Dubrule, and M. J. Blunt (2018), Conditioning of generative adversarial networks for pore and reservoir scale models, in *80th EAGE Conference and Exhibition 2018*, 1, pp. 1–5.
- Mosser, L., O. Dubrule, and M. J. Blunt (2020), Stochastic seismic waveform inversion using generative adversarial networks as a geological prior, *Mathematical Geosciences*, 52(1), 53–79.
- Müller, S., and L. Schüler (2020), GeoStat-Framework/GSTools. Zenodo. <https://doi.org/10.5281/zenodo.1313628>.
- Müller, S., L. Schüler, A. Zech, and F. Heße (2022), GSTools v1. 3: a toolbox for geostatistical modelling in Python, *Geoscientific Model Development*, 15(7), 3161–3182.
- Neal, R. M. (2001), Annealed importance sampling, *Statistics and Computing*, 11, 125–139.
- Neal, R. M. (2011), MCMC using Hamiltonian dynamics, *Handbook of Markov Chain Monte Carlo*, 2(11), 113–162.
- Nielsen, D., P. Jaini, E. Hoogeboom, O. Winther, and M. Welling (2020), Survae flows: Surjections to bridge the gap between vaes and flows, *Advances in Neural Information Processing Systems*, 33, 12685–12696.

- Nijkamp, E., R. Gao, P. Sountsov, S. Vasudevan, B. Pang, S.-C. Zhu, and Y. N. Wu (2020), Learning energy-based model with flow-based backbone by neural transport MCMC, *arXiv preprint arXiv:2006.06897*.
- Nowak, W., S. Tenkleve, and O. A. Cirpka (2003), Efficient computation of linearized cross-covariance and auto-covariance matrices of interdependent quantities, *Mathematical Geology*, 35, 53–66.
- Oldenburg, D., and D. Pratt (2007), Geophysical inversion for mineral exploration: A decade of progress in theory and practice, in *Proceedings of Exploration 07: Fifth Decennial International Conference on Mineral Exploration*, vol. 7, edited by B. Milkereit, pp. 61–95.
- Oliver, M. A., and R. Webster (1990), Kriging: a method of interpolation for geographical information systems, *International Journal of Geographical Information System*, 4(3), 313–332.
- Papamakarios, G., T. Pavlakou, and I. Murray (2017), Masked autoregressive flow for density estimation, *Advances in Neural Information Processing Systems*, 30.
- Papamakarios, G., E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan (2021), Normalizing flows for probabilistic modeling and inference, *Journal of Machine Learning Research*, 22(57), 1–64.
- Parker, R. L. (1994), *Geophysical Inverse Theory*, vol. 1, Princeton university press.
- Peterson, J. E., Jr (2001), Pre-inversion corrections and analysis of radar tomographic data, *Journal of Environmental & Engineering Geophysics*, 6(1), 1–18.
- Pirot, G., N. Linde, G. Mariethoz, and J. H. Bradford (2017), Probabilistic inversion with graph cuts: Application to the Boise Hydrogeophysical Research Site, *Water Resources Research*, 53(2), 1231–1250.
- Plessix, R.-E. (2006), A review of the adjoint-state method for computing the gradient of a functional with geophysical applications, *Geophysical Journal International*, 167(2), 495–503.
- Podvin, P., and I. Lecomte (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.
- Prince, S. J. (2012), *Computer Vision: Models, Learning, and Inference*, Cambridge University Press.
- Puzyrev, V., and A. Swidinsky (2021), Inversion of 1d frequency-and time-domain electromagnetic data with convolutional neural networks, *Computers & Geosciences*, 149, 104681.

- Raissi, M., P. Perdikaris, and G. E. Karniadakis (2019), Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics*, 378, 686–707.
- Ramgraber, M., R. Weatherl, F. Blumensaat, and M. Schirmer (2021), Non-Gaussian parameter inference for hydrogeological models using Stein variational gradient descent, *Water Resources Research*, 57(4), e2020WR029339.
- Rammy, M. H., A. H. Elsheikh, and Y. Chen (2019), Quantification of prediction uncertainty using imperfect subsurface models with model error estimation, *Journal of Hydrology*, 576, 764–783.
- Rasht-Behesht, M., C. Huber, K. Shukla, and G. E. Karniadakis (2022), Physics-informed neural networks (pinns) for wave propagation and full waveform inversions, *Journal of Geophysical Research: Solid Earth*, 127(5), e2021JB023120.
- Ravalec, M. L., B. Noetinger, and L. Y. Hu (2000), The FFT moving average (FFT-MA) generator: An efficient numerical method for generating and conditioning Gaussian simulations, *Mathematical Geology*, 32(6), 701–723.
- Reading, A. M., M. J. Cracknell, D. J. Bombardieri, and T. Chalke (2015), Combining machine learning and geophysical inversion for applied geophysics, *ASEG Extended Abstracts, 2015(1)*, 1–5.
- Revil, A., and A. Jardani (2013), *The Self-Potential Method: Theory and Applications in Environmental Geosciences*, Cambridge University Press.
- Rezende, D., and S. Mohamed (2015), Variational inference with normalizing flows, in *International Conference on Machine Learning*, vol. 37, edited by F. Bach and D. Blei, pp. 1530–1538, PMLR.
- Richardson, A. (2018), Generative adversarial networks for model order reduction in seismic full-waveform inversion, *arXiv preprint arXiv:1806.00828*.
- Ripley, B. D. (2009), *Stochastic Simulation*, John Wiley & Sons.
- Rizzuti, G., A. Siahkoohi, P. A. Witte, and F. J. Herrmann (2020), Parameterizing uncertainty by deep invertible networks: An application to reservoir characterization, in *SEG International Exposition and Annual Meeting*, p. D031S057R006.
- Robert, C. P., G. Casella, and G. Casella (1999), *Monte Carlo Statistical Methods*, vol. 2, Springer.
- Roberts, G. O., and J. S. Rosenthal (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., R. L. Tweedie, et al. (1996), Exponential convergence of Langevin distributions and their discrete approximations, *Bernoulli*, 2(4), 341–363.

- Rosas-Carbajal, M., K. Jourde, J. Marteau, S. Deroussi, J.-C. Komorowski, and D. Gibert (2017), Three-dimensional density structure of La Soufrière de Guadeloupe lava dome from simultaneous muon radiographies and gravity data, *Geophysical Research Letters*, 44(13), 6743–6751.
- Rücker, C., T. Günther, and F. M. Wagner (2017), pyGIMLi: An open-source library for modelling and inversion in geophysics, *Computers & Geosciences*, 109, 106–123.
- Ruggeri, P., J. Irving, and K. Holliger (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.
- Russell, B. (2019), Machine learning and geophysical inversion—a numerical study, *The Leading Edge*, 38(7), 512–519.
- Sagar, D., Q. Cheng, and F. Agterberg (2018), *Handbook of Mathematical Geosciences: Fifty Years of IAMG*, Springer Nature.
- Sambridge, M., P. Rickwood, N. Rawlinson, and S. Sommacal (2007), Automatic differentiation in geophysical inverse problems, *Geophysical Journal International*, 170(1), 1–8.
- Sambridge, M., A. Jackson, and A. P. Valentine (2022), Geophysical inversion and optimal transport, *Geophysical Journal International*, 231(1), 172–198.
- Scheidt, C., L. Li, and J. Caers (2018), *Quantifying Uncertainty in Subsurface Systems*, vol. 236, John Wiley & Sons.
- Scheiter, M., A. Valentine, and M. Sambridge (2022), Upscaling and downscaling Monte Carlo ensembles with generative models, *Geophysical Journal International*, 230(2), 916–931.
- Seillé, H., and G. Visser (2020), Bayesian inversion of magnetotelluric data considering dimensionality discrepancies, *Geophysical Journal International*, 223(3), 1565–1583.
- Si, X., Y. Yuan, F. Ping, Y. Zheng, and L. Feng (2020), Ground roll attenuation based on conditional and cycle generative adversarial networks, in *SEG 2019 Workshop: Mathematical Geophysics: Traditional vs Learning, Beijing, China, 5-7 November 2019*, pp. 95–98, Society of Exploration Geophysicists.
- Siahkoobi, A., M. Louboutin, and F. J. Herrmann (2019), The importance of transfer learning in seismic modeling and imaging, *Geophysics*, 84(6), A47–A52.
- Skilling, J. (2006), Nested sampling for general Bayesian computation, *Bayesian Analysis*, 1(4), 833–860.
- Solonen, A., P. Ollinaho, M. Laine, H. Haario, J. Tamminen, and H. Järvinen (2012), Efficient MCMC for climate model parameter estimation: parallel adaptive chains and early rejection, *Bayesian Analysis*, 7(2), 1–22.
- Song, C., T. Alkhalifah, and U. B. Waheed (2021a), Solving the frequency-domain acoustic vti wave equation using physics-informed neural networks, *Geophysical Journal International*, 225(2), 846–859.

- Song, S., T. Mukerji, and J. Hou (2021b), GANSim: Conditional facies simulation using an improved progressive growing of generative adversarial networks (GANs), *Mathematical Geosciences*, 53(7), 1413–1444.
- Song, S., T. Mukerji, and J. Hou (2021c), Bridging the gap between geophysics and geology with generative adversarial networks, *IEEE Transactions on Geoscience and Remote Sensing*, 60, 1–11.
- Straubhaar, J., and P. Renard (2021), Conditioning multiple-point statistics simulation to inequality data, *Earth and Space Science*, 8(5), e2020EA001515.
- Straubhaar, J., P. Renard, G. Mariethoz, R. Froidevaux, and O. Besson (2011), An improved parallel multiple-point algorithm using a list approach, *Mathematical Geosciences*, 43, 305–328.
- Straubhaar, J., P. Renard, and G. Mariethoz (2016), Conditioning multiple-point statistics simulations to block data, *Spatial Statistics*, 16, 53–71.
- Strebelle, S. (2002), Conditional simulation of complex geological structures using multiple-point statistics, *Mathematical Geology*, 34, 1–21.
- Strebelle, S., and N. Remy (2005), Post-processing of multiple-point geostatistical models to improve reproduction of training patterns, in *Geostatistics banff 2004*, edited by O. Leuangthong and C. V. Deutsch, pp. 979–988, Springer Netherlands.
- Subramanian, A. K., and N. Y. Chong (2019), Mean spectral normalization of deep neural networks for embedded automation, in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pp. 249–256.
- Sun, A. Y., B. R. Scanlon, Z. Zhang, D. Walling, S. N. Bhanja, A. Mukherjee, and Z. Zhong (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.
- Tahmasebi, P. (2018), Multiple point statistics: a review, *Handbook of Mathematical Geosciences*, pp. 613–643.
- Tang, M., Y. Liu, and L. J. Durlofsky (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., and B. Valette (1982a), Generalized nonlinear inverse problems solved using the least squares criterion, *Reviews of Geophysics*, 20(2), 219–232.
- Tarantola, A., and B. Valette (1982b), Inverse Problems= Quest for Information, *Journal of Geophysics*, 50(1), 159–170.

- Ter Braak, C., and C. Diks (2009), Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling, *International Journal of Nonlinear Sciences and Numerical Simulation*, 10(3), 273–290.
- Ter Braak, C. J., and J. A. Vrugt (2008), Differential Evolution Markov Chain with snooker updater and fewer chains, *Statistics and Computing*, 18(4), 435–446.
- Tieleman, T., and G. Hinton (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.
- Tikhonov, A. (1963), Resolution of ill-posed problems and the regularization method (in russian), in *Dokl. Akad. Nauk SSSR*, vol. 151, pp. 501–504.
- Toms, B. A., E. A. Barnes, and I. Ebert-Uphoff (2020), Physically interpretable neural networks for the geosciences: Applications to earth system variability, *Journal of Advances in Modeling Earth Systems*, 12(9), e2019MS002002.
- Tripathy, R. K., and I. Bilonis (2018), Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification, *Journal of Computational Physics*, 375, 565–588.
- Ulyanov, D., A. Vedaldi, and V. Lempitsky (2016), Instance normalization: The missing ingredient for fast stylization, *arXiv preprint arXiv:1607.08022*.
- Urozayev, D., B. Ait-El-Fquih, I. Hoteit, and D. Peter (2021), A reduced-order variational Bayesian approach for efficient subsurface imaging, *Geophysical Journal International*.
- Valentine, A. P., and M. Sambridge (2023), Emerging directions in geophysical inversion, *Applications of Data Assimilation and Inverse Problems in the Earth Sciences*, 5, 9.
- Virieux, J., and S. Operto (2009), An overview of full-waveform inversion in exploration geophysics, *Geophysics*, 74(6), WCC1–WCC26.
- Vrugt, J. A., C. J. Ter Braak, M. P. Clark, J. M. Hyman, and B. A. Robinson (2008), Treatment of input uncertainty in hydrologic modeling: Doing hydrology backward with Markov chain Monte Carlo simulation, *Water Resources Research*, 44(12), W00B09.
- Vrugt, J. A., C. J. Ter Braak, H. V. Gupta, and B. A. Robinson (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., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli (2004), Image quality assessment: from error visibility to structural similarity, *IEEE Transactions on Image Processing*, 13(4), 600–612.
- Wegener, M., and G. R. Amin (2019), Minimizing greenhouse gas emissions using inverse dea with an application in oil and gas, *Expert Systems with Applications*, 122, 369–375.

- Wilt, M., and D. Alumbaugh (2003), Oil field reservoir characterization and monitoring using electromagnetic geophysical techniques, *Journal of Petroleum Science and Engineering*, 39(1-2), 85–97.
- Wold, S., K. Esbensen, and P. Geladi (1987), Principal component analysis, *Chemometrics and Intelligent Laboratory Systems*, 2(1-3), 37–52.
- 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 A. J. Valocchi (2015), A Bayesian approach to improved calibration and prediction of groundwater models with structural error, *Water Resources Research*, 51(11), 9290–9311.
- Xu, T., A. J. Valocchi, M. Ye, and F. Liang (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.
- Yongkai, A., Y. Xueman, L. Wenxi, H. Qian, and Z. Zaiyong (2022), An improved Bayesian approach linked to a surrogate model for identifying groundwater pollution sources, *Hydrogeology Journal*, 30(2), 601–616.
- Yu, S., and J. Ma (2020), Data-driven geophysics: from dictionary learning to deep learning, *arXiv preprint arXiv:2007.06183*.
- Yu, S., and J. Ma (2021), Deep learning for geophysics: Current and future trends, *Reviews of Geophysics*, 59(3), e2021RG000742.
- Zahner, T., T. Lochbühler, G. Mariethoz, and N. Linde (2016), Image synthesis with graph cuts: a fast model proposal mechanism in probabilistic inversion, *Geophysical Journal International*, 204(2), 1179–1190.
- Zahura, F. T., J. L. Goodall, J. M. Sadler, Y. Shen, M. M. Morsy, and M. Behl (2020), Training machine learning surrogate models from a high-fidelity physics-based model: Application for real-time street-scale flood prediction in an urban coastal community, *Water Resources Research*, 56(10), e2019WR027038.
- Zhang, C., X. Song, and L. Azevedo (2021a), U-net generative adversarial network for subsurface facies modeling, *Computational Geosciences*, 25, 553–573.
- Zhang, R., R. Zen, J. Xing, D. M. S. Arsa, A. Saha, and S. Bressan (2020), Hydrological process surrogate modelling and simulation with neural networks, in *Advances in Knowledge Discovery and Data Mining*, edited by H. W. Lauw, R. C.-W. Wong, A. Ntoulas, E.-P. Lim, S.-K. Ng, and S. J. Pan, pp. 449–461, Springer International Publishing.
- Zhang, T. (2006), *Filter-Based Training Pattern Classification for Spatial Pattern Simulation*, Stanford University.
- Zhang, T. (2015), MPS-driven digital rock modeling and upscaling, *Mathematical Geosciences*, 47(8), 937–954.

- Zhang, X., and A. Curtis (2020a), Seismic tomography using variational inference methods, *Journal of Geophysical Research: Solid Earth*, 125(4), e2019JB018589.
- Zhang, X., and A. Curtis (2020b), Variational full-waveform inversion, *Geophysical Journal International*, 222(1), 406-411.
- Zhang, X., and A. Curtis (2021), Bayesian full-waveform inversion with realistic priors, *Geophysics*, 86(5), A45-A49.
- Zhang, X., M. A. Nawaz, X. Zhao, and A. Curtis (2021b), An introduction to variational inference in geophysical inverse problems, in *Inversion of Geophysical Data, Advances in Geophysics*, vol. 62, edited by C. Schmelzbach, pp. 73–140, Elsevier.
- Zhang, Z.-d., and T. Alkhalifah (2020), High-resolution reservoir characterization using deep learning-aided elastic full-waveform inversion: The North Sea field data example, *Geophysics*, 85(4), WA137–WA146.
- Zhao, X., A. Curtis, and X. Zhang (2022), Bayesian seismic tomography using normalizing flows, *Geophysical Journal International*, 228(1), 213–239.
- Zhdanov, M. S. (2015), *Inverse Theory and Applications in Geophysics*, vol. 36, Elsevier.
- Zheng, Q., L. Zeng, and G. E. Karniadakis (2020), Physics-informed semantic inpainting: Application to geostatistical modeling, *Journal of Computational Physics*, 419, 109676.
- Zheng, Y., Q. Zhang, A. Yusifov, and Y. Shi (2019), Applications of supervised deep learning for seismic interpretation and inversion, *The Leading Edge*, 38(7), 526–533.
- Zhu, W., K. Xu, E. Darve, and G. C. Beroza (2021), A general approach to seismic inversion with automatic differentiation, *Computers & Geosciences*, 151, 104751.
- Zinn, B., and C. F. Harvey (2003), When good statistical models of aquifer heterogeneity go bad: A comparison of flow, dispersion, and mass transfer in connected and multivariate gaussian hydraulic conductivity fields, *Water Resources Research*, 39(3), 1051.

Shiran Levy

✉ shiran.levy@unil.ch

🆔 Shiran Levy

🌐 ShiLevy

Education

- 2019 – 2023 📖 **Ph.D. Earth Sciences**, University of Lausanne, Lausanne, Switzerland
Thesis title: *Efficient Bayesian inversion for geophysical applications: Leveraging deep learning and geostatistical methods*
Director: Prof. Niklas Linde
- 2016 – 2018 📖 **M.Sc. Applied Geophysics**, ETH Zürich, TU Delft, RWTH Aachen
Thesis title: *Fault state estimation in subduction zones using a particle filter with time-lagged particle generation*
Director: Dr. Femke Vossepoel
- 2012 – 2015 📖 **B.Sc. Earth Sciences**, The Hebrew University of Jerusalem, Jerusalem, Israel
Thesis title: *Northern Levant's four major geological processes during Permian to Pleistocene*
Director: Prof. Dov Avigad

Research Publications

- 1 Levy, S., L. Friedli, G. Mariéthoz, and N. Linde (2023). “Conditioning of multiple-point statistics simulations to indirect geophysical data”. *Under review*.
- 2 Levy, S., E. Laloy, and N. Linde (2023). “Variational Bayesian inference with complex geostatistical priors using inverse autoregressive flows”. *Computers & Geosciences* 171, p. 105263.
- 3 Meles, G. A., M. Amaya, S. Levy, S. Marelli, and N. Linde (2023). “Efficient Bayesian travel-time tomography with geologically-complex priors using sensitivity-informed polynomial chaos expansion and deep generative networks”. *Under review*.
- 4 Levy, S., J. Hunziker, E. Laloy, J. Irving, and N. Linde (2022). “Using deep generative neural networks to account for model errors in Markov chain Monte Carlo inversion”. *Geophysical Journal International* 228.2, pp. 1098–1118.

Conferences

- April 2023 📖 **Sequential multiple-point statistics simulations conditioned on arithmetic averages**
EGU General Assembly
- August 2022 📖 **Efficient inversion with complex geostatistical priors using neural transport**
21st Annual Conference of the International Association for Mathematical Geosciences
- June 2022 📖 **Efficient inversion with complex geostatistical priors using normalizing flows and variational inference**
14th international conference on geostatistics for environmental applications
- April 2022 📖 **Efficient inversion with complex geostatistical priors using normalizing flows and variational inference**
EGU General Assembly

Conferences (continued)

December 2020  **Accounting for model errors using deep generative neural networks and Markov chain Monte Carlo inversion**
AGU Fall Meeting

Workshops and summer schools

January 2021  **Deep learning workshop**
PhD School Water-Earth Systems - University of Neuchâtel, Neuchâtel, Switzerland

July 2021  **Flow and transport in porous & fractured media**
5th Summer School - L'Institut d'Etudes Scientifiques de Cargèse, Cargèse, France

Skills

Languages  English (fluent), Hebrew (native), German (B1), Italian (A2), French (A1-A2)

Coding  Python, MATLAB, C++, Bash script, Java, L^AT_EX

Misc.  Linux, Windows, high-performance computing, Git