Dimensionality reduction for geophysical inversion in highly structured subsurface environments

Jorge Lopez Alvis

A thesis presented for the degree of
Doctor of Philosophy (PhD)

Faculty of Sciences, Department of Geology
Ghent University
and
Faculty of Applied Sciences, Urban and Environmental Engineering
University of Liège

Promoters:
Prof. Frédéric Nguyen, University of Liège
Prof. Thomas Hermans, Ghent University
Jury:
Prof. Alain Dassargues, University of Liège
Prof. Philippe De Smedt, Ghent University
Prof. Majken Caroline Looms Zibar, University of Copenhagen
Dr. Eric Laloy, SCK-CEN
Dr. James Irving, University of Lausanne

March 2021

This work has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement number 722028 (ENIGMA ITN).

To my wife Lorena

Contents

Abstract

For highly structured subsurface, the use of strong prior information in geophysical inversion produces realistic models. Machine learning methods allow to encode or parameterize such models with a low dimensional representation. These methods require a large number of examples to learn such latent or intrinsic parameterization. By using deep generative models, inversion is performed in a latent space and resulting models display the desired patterns. However, the degree of nonlinearity for the generative mapping (which goes from latent to original representation) dictates how useful the parameterization is for tasks other than mere compression. After recognizing that changes in curvature and topology are the main cause of such nonlinearity, an adequate training for a variational autoencoder (VAE) is shown to allow the application of gradient-based inversion. Data obtained in highly structured subsurface may also be represented by low-dimensional parameterizations. Compressed versions of the data are useful for prior falsification because they allow modeling marginal probability distributions of structural parameters in a latent space. An objective way based on cross-validation is proposed to choose which compression technique retains information relevant to high-level structural parameters. Inversion and prior falsification using dimensionality reduction provide a computationally efficient framework to produce realistic models of the subsurface. This framework is successfully applied to a field dataset using a prior distribution assembled from distinct patterns resemble a realistic geological environment including deformation and intrafacies variability.

Sammenvating

Voor sterk gestructureerde ondergrond levert het gebruik van prior informatie bij geofysische inversie realistische modellen op. Machine learning-methoden maken het mogelijk om modellen te parametriseren met een lage dimensionale representatie. Deze methoden vereisen veel voorbeelden om de latente parametrisatie te leren. Door diepe generatieve modellen te gebruiken, wordt inversie uitgevoerd in een latente ruimte en modellen tonen de gewenste structurele patronen. De mate van niet-lineariteit van de generatieve mapping (die van de latente ruimte naar de originele representatie gaat) bepaalt echter hoe nuttig de parametrisatie is voor andere taken dan compressie. Na erkenning dat veranderingen in kromming en topologie de hoofdoorzaak zijn van dergelijke niet-lineariteit, wordt er aangetoond dat een adequate training voor een Variationele Autoencoder de gradiënt gebaseerde inversie mogelijk maakt. Gegevens die in een gestructureerde ondergrond verkregen zijn, kunnen ook worden weergegeven door laag-dimensionale parametrisatie. Gecomprimeerde versies van de gegevens zijn nuttig voor vervalsing procedures omdat ze het mogelijk maken marginale kansverdelingen van structurele parameters te modelleren. Er wordt een objectieve manier voorgesteld om te kiezen welke compressietechniek informatie vasthoudt voor bepaalde structurele parameters. Inversie en vervalsing met behulp van dimensionaliteitsreductie technieken bieden een rekenkundige efficiënte methode aan om realistische modellen van de ondergrond te produceren. Deze methode is met succes toegepast op een velddataset met behulp van een prior distributie samengesteld uit verschillende patronen die lijken op een realistische geologische omgeving, inclusief vervorming en variabiliteit binnen geologische facies.

Résumé

Pour des sous-sols hautement structurés, l'utilisation d'information a priori en inversion géophysique produit des modèles réalistes. Les méthodes de machine learning permettent de paramétrer ces modèles avec une représentation de faible dimension. Ces méthodes nécessitent cependant un grand nombre d'exemples pour apprendre une telle paramétrisation, appelée latente ou intrinsèque. En utilisant des modèles génératifs profonds (deep generative models), l'inversion est effectuée dans un espace latent et les modèles obtenus affichent les structures souhaitées. Cependant, le degré de non-linéarité de la fonction générative (qui va de la représentation latente à la représentation originale) dicte l'utilité du paramétrage pour des tâches autres que la simple compression. Après avoir reconnu que les changements de courbure et de topologie sont la cause principale de la non-linéarité, un entraînement adéquat pour un autoencodeur variationnel (VAE) est proposé pour permettre l'application de l'inversion basée sur le gradient. Les données obtenues dans un sous-sol hautement structuré peuvent également être représentées par des paramétrisations de faible dimension. Les versions compressées des données sont utiles pour la falsification de la distribution a priori car elles permettent de modéliser les distributions de probabilité marginales des paramètres structurels dans un espace latent. Une méthode objective basée sur la validation croisée est proposée pour choisir la technique de compression qui retient le maximum d'information relative aux paramètres structurels étudiés. L'inversion et la falsification préalable à l'aide de la réduction de dimensionnalité fournissent un cadre de calcul efficace pour produire des modèles réalistes du sous-sol. Ce cadre est appliqué avec succès à un ensemble de données de terrain en utilisant une distribution a priori assemblée à partir de modèles distincts,

ressemblant à un environnement géologique réaliste, y compris la déformation et la variabilité intrafacies.

Acknowledgments

During the last four years I felt immersed in a continuous learning experience both about life and science. I have to thank both my supervisors, Fred and Thomas, for this wonderful opportunity. They gave me their great support from the beginning by sharing their knowledge, encouraging interesting research directions and by doing a lot to make sure I was comfortable working and living in Belgium. They both pass me on their passion for geophysics and I'm particularly glad to have had the chance to join Thomas' research team at the beginning of his career as a professor, I hope he will keep showing the same dedication for both his research and students as he did for me. I thank also my coauthors for their trust and for allowing great collaborations, and the jury members for their valuable and constructive feedback.

I would like to thank my wife, Lorena, for being part of this achievement. I will remain forever grateful for her support, love, patience and all her joyful efforts to keep us motivated during this time. Although we both missed home, her smile was enough to stop me from being homesick and enjoy every moment. I would also like to thank my parents, my brother and the rest of my family for their support.

I'm also thankful to my colleagues at both research teams in Liège and Ghent for their support, interesting chats and for inviting me, the guy with scant field work, to learn and help in their field campaigns. I'm specially thankful to Richard for being a good friend and also for helping us adjust to life in Europe and to Itzel and Edmundo for their company and nice moments at the University. I also thank all the people from the ENIGMA ITN for their great dedication to the project and each of my 14 fellow PhD students for the great experiences both during learning

and outside it.

I would like to acknowledge the funding provided as part of ENIGMA ITN for the first three years of my research and two research scholarships by Ghent University and University of Liege provided during the last year.

Chapter 1

Introduction

1.1 Background

1.1.1 Geophysical inversion

Geophysical methods aim to provide a model of the subsurface (represented by a set of parameters) based on a set of sparse measurements sensing the spatial domain of interest. Obtaining a model from the measured data may be framed quantitatively as the solution of an inverse problem. Consider a survey or experiment for which a vector of noisy measurements $\mathbf{d} = (d_1, \ldots, d_Q)^T \in \mathbb{R}^Q$ of a physical process is available. A simplified description of the process may be expressed by a mathematical forward operator $\mathbf{f} : \mathbb{R}^N \rightarrow \mathbb{R}^Q$ that takes as input a subsurface model vector $\mathbf{m} = (m_1, \ldots, m_D)^T \in \mathbb{R}^N$ obtained by discretizing the spatial distribution of physical properties and outputs a simulated response $\mathbf{f}(\mathbf{m})$. Commonly, this operator is in the form of a set of partial differential equations (PDE) describing the process under study and is an approximation of the real process. These PDEs may be solved in different ways e.g. with analytical or numerical methods, some of which may imply additional approximations. As a result of these approximations and the use of noisy data, an error term $\boldsymbol{\eta}$ is added to the simulation to represent total uncertainty (in a probabilistic approach, the relation may be alternatively described by a conditional probability distribution as detailed in Appendix A). Then, the relation between the operator and the mea-

surements may be written as (see e.g. Aster et al., 2013):

$$\mathbf{d} = \mathbf{f}(\mathbf{m}) + \boldsymbol{\eta} \tag{1.1}$$

The corresponding inverse problem or inversion of Eq. (1.1), aims to obtain an estimation of the vector $\mathbf{m}$ from the (noisy) data $\mathbf{d}$. Deterministic inversion does so by optimizing a misfit or objective function $\gamma(\mathbf{m})$ that is usually given in the form of a distance function between simulated response $\mathbf{f}(\mathbf{m})$ and data $\mathbf{d}$, e.g. by the l_2 norm:

$$\gamma(\mathbf{m}) = \|\mathbf{f}(\mathbf{m}) - \mathbf{d}\|_2^2 = \sum_i (f_i(\mathbf{m}) - d_i)^2 \tag{1.2}$$

Using a probabilistic approach, under certain assumptions (see Appendix A), the estimation of maximum a posteriori values is described by the same optimization.

If the forward operator is linear and the original parameterization of $\mathbf{m}$ is used, the objective function in Eq. 1.2 is convex and methods for solving linear systems of equations are used (either direct or iterative methods). In this case, the inverse problem may still be ill-conditioned and require stabilization for its solution, e.g. through regularization (Aster et al., 2013). On the other hand, when either the forward operator is nonlinear or the model $\mathbf{m}$ is reparameterized nonlinearly, such gradient-based methods may face some difficulties in finding the optimal values for $\mathbf{m}$, i.e. they tend to get caught in local minima if the starting model is far from the optimum. Alternative global optimization methods such as simulated annealing or genetic algorithms may be used in this case. When applicable, however, gradient-based methods are generally more computationally efficient. In practice, gradient-based methods are useful when both the forward operator and the reparameterization are moderately nonlinear (as is detailed in Chapters 3 and 5). Gradient-based inversion requires the gradient $\nabla_{\mathbf{m}}\gamma(\mathbf{m})$ whose elements are:

$$[\nabla_{\mathbf{m}}\gamma(\mathbf{m})]_i = \frac{\partial\gamma(\mathbf{m})}{\partial m_i} \tag{1.3}$$

and are computed by considering Eq. (1.1) together with the chosen misfit.

When inversion is used to obtain a subsurface model some limitations must

be identified: (1) locations of sensors are restricted to on/above the surface of the ground and in boreholes, (2) measurements are often contaminated with noise and (3) in most cases one can only rely on an imperfect forward operator to mathematically simulate the measurements. On the one hand, from a deterministic point of view these limitations cause the inverse problem to be ill-posed and its solution to be non-unique (Aster et al., 2013). On the other hand, if a probabilistic approach is adopted the limitations increase the uncertainty in the solution, which is represented then by a probability distribution (Tarantola and Valette, 1982; Tarantola, 2005). Regardless of the point of view adopted, inversion results are more realistic when additional information regarding the structures in the subsurface is considered (Linde et al., 2015). Prior information may be either enforced by adding regularization or penalization terms in deterministic inversion or by considering a prior probability distribution in probabilistic inversion. When this prior information is limited, inversion is often done relying on relatively strong assumptions such as smoothness, sparsity or a covariance model for a Gaussian random field (Backus and Gilbert, 1967; Tikhonov and Arsenin, 1977; Franklin, 1970; Maurer et al., 1998). However, these assumptions strictly apply only when the subsurface structure is relatively simple and might thus lead to geologically unrealistic solutions when it is complex. When geophysical data is acquired for a highly structured subsurface (e.g. with high connectivity), an appropriate complex prior may be found that produces consistent structures but in general it is harder to use it for inversion since more specialized sampling is needed (Hu et al., 2001; Caers and Hoffman, 2006; Zahner et al., 2016). Figure 1.1 shows examples of models obtained with smooth regularization and a complex prior that imposes a structure consistent with given training patterns.

1.1.2 Prior information for highly structured subsurface environments

In order to accurately represent such highly structured subsurface, one may discretize the sensed domain using a high number of cells (or pixels), then a model $\mathbf{m}$ may be seen as a point in a high-dimensional model space $\mathbb{R}^N$ where N is the number of cells. However, given that only certain structures or patterns are

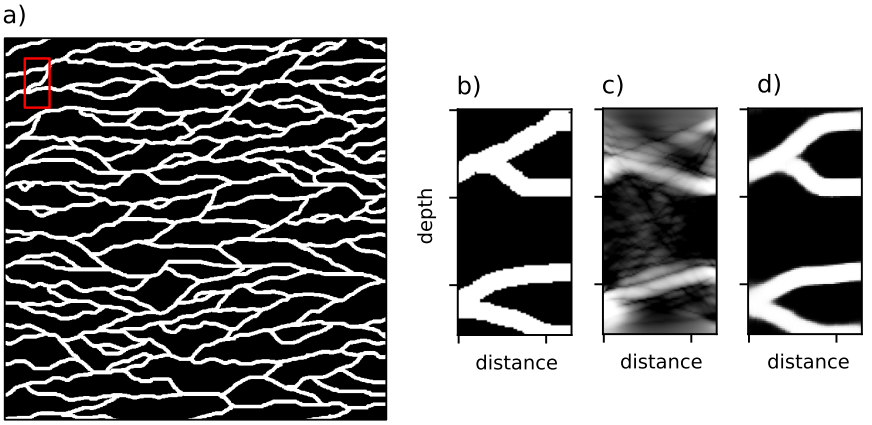


Figure 1.1: Inversion with prior information: (a) pattern samples for a highly structured subsurface in the form of a training image where the red rectangle shows the size of the domain of interest, (b) truth subsurface model, (c) inverted model with smooth regularization, and (d) inverted model with prior obtained from the training image.

expected in the subsurface (according to the prior assumptions), the intrinsic dimensionality of the model space is usually lower. In other words, the possible models lie on a subset $\mathcal{M}$ of $\mathbb{R}^N$. This assumption is known as the manifold hypothesis in machine learning literature (Fefferman et al., 2016). From a probabilistic point of view, this means that our prior distribution is defined only over $\mathcal{M}$. Although the sampling of models from the prior distribution on $\mathcal{M}$ may be done by using regularization (Lange et al., 2012), multiple-point statistics (Caers and Hoffman, 2006) and example texture synthesis (Zahner et al., 2016), recent advances in machine learning methods such as deep generative models (DGMs) represent an alternative to the former methods (Laloy et al., 2017; Mosser et al., 2018; Richardson, 2018). Note that the four above-mentioned strategies may be considered data-driven since they require a large number of training samples or patterns to approximate the prior on $\mathcal{M}$.

In the case of subsurface models, such training patterns may take the form of two- or three-dimensional training images (TIs). These TIs are representative of the structures formed by the different materials present in the subsurface. The TIs

may be: (1) designed or drawn by geologists who use their knowledge about the local geological environment (Park et al., 2013; Hermans et al., 2015), (2) directly digitized from photos of outcrops near the surveyed domain (Kessler et al., 2013), and/or (3) obtained from analogous or similar structures formed in other physical environments (Mariethoz and Kelly, 2011). When several patterns are possible for inversion, one may perform a first step where the probability of each pattern given the measured data is computed and then some of the patterns are potentially falsified if the probability is too low. This step is called prior falsification (Scheidt et al., 2018) and is done before any complex and costly inversion. In general, prior falsification helps to correctly represent uncertainty for field cases where two or more geological scenarios are deemed possible, even if they are not based on training images (e.g. when a geological scenario is modeled with a Gaussian field).

Since geophysical data $\mathbf{d}$ is obtained from a subsurface domain in the field which is conceptualized as a model, it may be approximated by a forward operator that is a function $f : \mathbb{R}^N \rightarrow \mathbb{R}^Q$ with Q denoting the size of the data vectors. If one considers only certain subsurface patterns, all the possible (noiseless) data points will be constrained to the map $f : \mathcal{M} \rightarrow \mathcal{D}$. Given the limited number of sensor locations and the spatial averaging of the measurement process, this data manifold $\mathcal{D}$ is a subset of $\mathbb{R}^Q$ and is generally of lower dimensionality than both $\mathcal{M}$ and $\mathbb{R}^Q$. Samples of $\mathcal{D}$ may be mainly obtained in two different ways. First, one may simply obtain a sample $\mathbf{m}$ of $\mathcal{M}$ and then use the forward operator to obtain a data sample $\mathbf{d} = f(\mathbf{m})$. Note that when the forward operator is computationally expensive, one may use a replacement or surrogate model instead of the forward operator. The second way is to directly learn the data manifold and then simply generate data samples from it without having to use the forward model (or a surrogate). Similar to the case of $\mathcal{M}$, data-driven strategies may be used to approximate $\mathcal{D}$. In this work, principal component analysis (PCA) or multidimensional scaling (MDS) are used to approximate $\mathcal{D}$ (Scheidt et al., 2018). While learning the data manifold $\mathcal{D}$ alone may not be directly useful for inversion, it is helpful in the previous step of prior consistency or prior falsification (Park et al., 2013; Hermans et al., 2015; Scheidt et al., 2015b).

The use of the term "data" in "data-driven" is not to be confused with the

geophysical data or measurements and simply refers to models that require large amounts of training samples whatever they might be, e.g. training patterns or images to learn the model prior distribution or data samples to learn the approximation of the data manifold. In fact, one of the main drivers of recent widespread use of machine learning is the development and availability of both hardware and software that is capable of processing such large amounts of training samples, e.g. algorithms specifically designed to take advantage of highly parallel computations on graphical processing units (GPUs) (Krizhevsky et al., 2017).

1.1.3 Dimensionality reduction and the latent space

Both DGMs for $\mathcal{M}$ and PCA or MDS for $\mathcal{D}$ approximate the corresponding manifolds in a low-dimensional space which is usually referred to as latent space (Bishop, 2006; Kingma and Welling, 2014). Formally, this is assumed an Euclidian space where the manifold is embedded (or immersed) (Shao et al., 2017; Arvanitidis et al., 2018; Chen et al., 2018). This space may be denoted as $\mathbb{R}^n$ for $\mathcal{M}$ and as $\mathbb{R}^q$ for $\mathcal{D}$, where $n << N$ and $q << Q$. Prior probability distributions may then be defined in such space. DGMs directly produce this prior distribution while for PCA and MDS one may use the training samples mapped in the latent space to estimate the probability density function by means of e.g. kernel density estimation (Park et al., 2013; Scheidt et al., 2018). Compared to other methods used to sample the prior distributions, the use of a latent space is advantageous because it provides an explicit representation of these prior distributions. However, the mapping to the latent space must be chosen in such a way that the low-dimensional representations of models and data are still useful for other purposes than merely compression. For instance, the mapping to the latent space obtained by DGMs may be chosen so that efficient gradient-based inversion is still possible (Laloy et al., 2019) or pre-processing may be done previous to PCA in order to selectively retain information related to certain aspects of the data or models for prior falsification (see Chapter 4).

The mapping to the latent space may be viewed as a dimensionality reduction operation or as obtaining a reparameterization with a data-driven sparse basis (Bora et al., 2017). In general, to achieve a lower dimensionality (or higher

compression) nonlinear mappings are required (Kramer, 1991). For instance, the mapping to the latent space resulting from application of PCA is linear, then compression without loss of accuracy is only possible if the manifold to be approximated is linear. However, if the manifold only slightly deviates from being linear, compression is still possible and the impact on the approximation of the manifold is minor. In contrast, the mapping to the latent space with DGMs is nonlinear (typically defined by a neural network) and therefore usually causes higher compression without significantly degrading the approximation of the manifold (Shao et al., 2017; Arvanitidis et al., 2018). In general, however, the more nonlinear the mapping the more samples are needed for learning such mapping. In summary, one must choose a dimensionality reduction strategy that is optimal for the application at hand, depending on (1) the number of available training samples for learning the mapping to the latent space, (2) the complexity or nonlinearity of the samples, and (3) the required accuracy in the approximation of the prior distribution which in turn depends on the objective of the dimensionality reduction, i.e. whether it is to be used for inversion or prior falsification.

Another useful interpretation of the setting described above comes from a adopting a probabilistic point of view. Inversion may be viewed as jointly considering all information available for the problem and then solve for the geophysical model vector. This may be explicitly represented by a probabilistic graphical model (see e.g. Bishop, 2006) which states the joint probability distribution of all variables with uncertainty and for which inference is done for the model $\mathbf{m}$. In this setting, using a latent space or reducing dimensions means replacing the original model vector $\mathbf{m}$ in the graphical model for another (denoted by $\mathbf{z}$ in this work) whose solution provides an approximation to that of the original model vector. The same may be applied to make inference for other variables in the graphical model (e.g. a variable representing different possible geological scenarios), which means obtaining marginals that may be useful for e.g. prior falsification. This substitution is either done to reduce computational cost (since evaluating integrals for the high-dimensional joint distribution is usually too expensive) or simply because there is no analytical form to express some prior distributions of the variables involved.

1.2 Objectives

The main objective of this thesis is to explore the use of dimensionality reduction methods for improving both inversion and prior falsification when geophysical data is acquired in a highly structured subsurface. In these conditions, standard deterministic inversions are failing to produce geologically realistic solutions, while probabilistic approaches are computationally too expensive to be applied in practice. To test the newly developed methodologies, both field and synthetic cross-borehole ground penetrating radar (GPR) traveltime data are considered but the outcomes of this work are applicable to other methods. The main objective is divided in three specific objectives:

1. Understanding the factors that limit the usefulness of DGMs to define a prior distribution for highly structured subsurface and testing if DGMs may be used successfully with gradient-based inversion (Chapter 3).

2. Proposing an objective way to select dimensionality reduction methods for prior falsification (Chapter 4).

3. Testing a framework that includes prior falsification using dimensionality reduction and a DGM as prior for inversion. This test includes validating the framework with field data and representing prior information as realistically as possible using an assembled prior, i.e. a prior including structures from different geological scenarios (Chapter 5).

1.3 Outline

This thesis is structured as follows. In Chapter 2, cross-borehole ground penetrating radar theory is given. This is necessary for a thorough understanding of the following chapters but may be skipped by a geophysicist familiar with the topic. In Chapter 3, an in-depth analysis on the use of DGMs to define a prior probability distribution for inversion is presented. Then, the use of a particular DGM called variational autoencoder (VAE) with an appropriate choice of training parameters is successfully proposed to define a prior distribution that allows

for gradient-based inversion by means of stochastic-gradient descent (SGD). This new framework is one of the first successful efficient geophysical inversion strategies based on DGM for non-linear problems. In Chapter 4, an objective way to select data-driven dimension reduction methods and some pre-processing techniques aimed at retaining only information relevant for prior falsification is presented. The proposed methodology is the first to propose a falsification procedure using ad-hoc features adapted for geophysical data while proposing an objective cross-validation procedure allowing to generalize the approach to any dimensionality reduction approach. Chapter 5 introduces a framework that combines both PCA-based prior falsification and a VAE to define an assembled prior distribution from different geological scenarios for gradient-based inversion. With this new framework it is possible to include perturbations of base patterns in the assembled prior obtained by deformation or intrafacies variability and also to estimate absolute velocity values. The framework is validated with a synthetic case and a field dataset. Finally, a general discussion and conclusions linking all the content in the thesis and giving some future perspectives are presented in Chapter 6. Figure 1.2 shows an overview of the proposed framework that highlights contributions of this thesis and provides a comparison with "traditional" inversion.

This thesis is based on three papers; one published, one submitted and one to be submitted in peer-review journals. The content of these papers is mainly presented in Chapters 3, 4 and 5. This content was edited with respect to the original versions so that repetition is limited and notation is consistent.

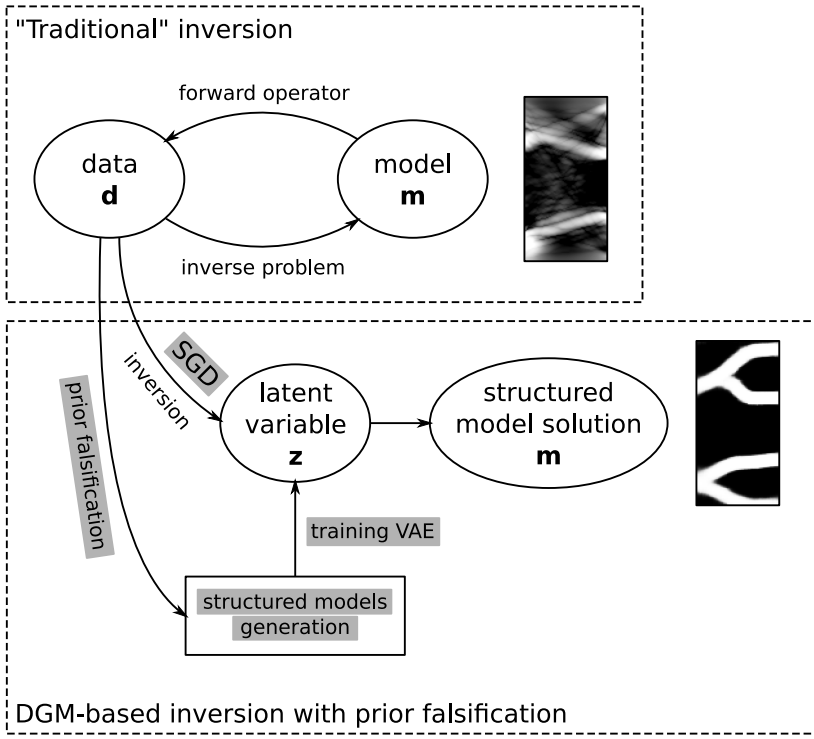


Figure 1.2: Diagram depicting an overview of the complete framework proposed in this work and the corresponding one for "traditional" inversion. Highlighted in gray are the main contributions of this thesis.

Chapter 2

Cross-borehole ground penetrating radar: theory and forward operator

The methods for inversion (Chapters 3 and 5) and prior falsification (Chapters 4 and 5) proposed in this thesis are tested for a particular type of geophysical data: traveltimes of the first arrival of electromagnetic waves from cross-borehole ground penetrating radar (GPR). This results in a specific spatial distribution of the sensitivity of the measurements which will influence the results as shown and discussed in the following chapters. In this chapter, an overview of the method and its corresponding geophysical forward operator f used to simulate the traveltimes of electromagnetic waves is presented together with the simplifying assumptions used to obtain such operator. For a more detailed review of GPR the reader is referred to Jol (2009) and Daniels (2004).

2.1 Principles of cross-borehole GPR

GPR uses a transmitter antenna to send an electromagnetic pulse into the subsurface and then records the signal that arrives at a receiver antenna. Sources and receivers may be located at the surface and/or in boreholes (Fig. 2.1). The

source pulse (also called wavelet) propagates through the subsurface and is scattered and attenuated by materials with different electromagnetic properties. The signal arriving at the receiver carries information both on the subsurface structure and its composition. For most subsurface materials, the magnetic permeability is very close to that of the vaccum (μ_0), therefore it is usual to only obtain electrical conductivity (σ) and permittivity (ϵ) from GPR data. The center frequency of the source pulse used in GPR is usually between 100 MHz and 4 GHz and is chosen depending on both the electromagnetic properties of the subsurface and the desired spatial resolution. For this frequency range, σ is related to attenuation while ϵ controls the wave velocity.

Data acquisition may be performed in either reflection or (direct) transmission modes, depending on the relative position of transmitter, receiver and the sensed region of the subsurface. Cross-borehole GPR refers to the case when source positions are located in one borehole and receivers positions are located in another borehole, i.e. borehole transmission mode (see Fig. 2.1a). When data for one source position is recorded in many or all receiver positions, the acquired data is referred to as multi-offset gather (MOG), otherwise, when only data from sources and receivers at the same depth is recorded, the dataset is referred to as zero-offset profile (ZOP).

The arriving signal is generally recorded only after a certain time since the source pulse is emitted. Such time lapse is expected to include mainly the signal coming from the domain of interest, possibly including multiple reflections and/or guided waves. These complete recorded signals are usually referred to as GPR traces or full-waveform data (Fig. 2.1b). In some cases, one may decide to work only with a subset of the full-waveform data for both computational and processing efficiency or when this subset is sufficient for the purpose of the survey. For instance, using only the traveltimes of the first arrivals of the waves (which are selected as shown in Fig. 2.1b and often represented as in Fig. 2.1c) one is able to obtain a model of the subsurface heterogeneity. However, while full-waveform data carries information on both electrical conductivity and permittivity, traveltimes only provide an estimate of the (wave) velocity distribution and also disregard information contained in later arrivals. Velocity (v) is related to permittivity by

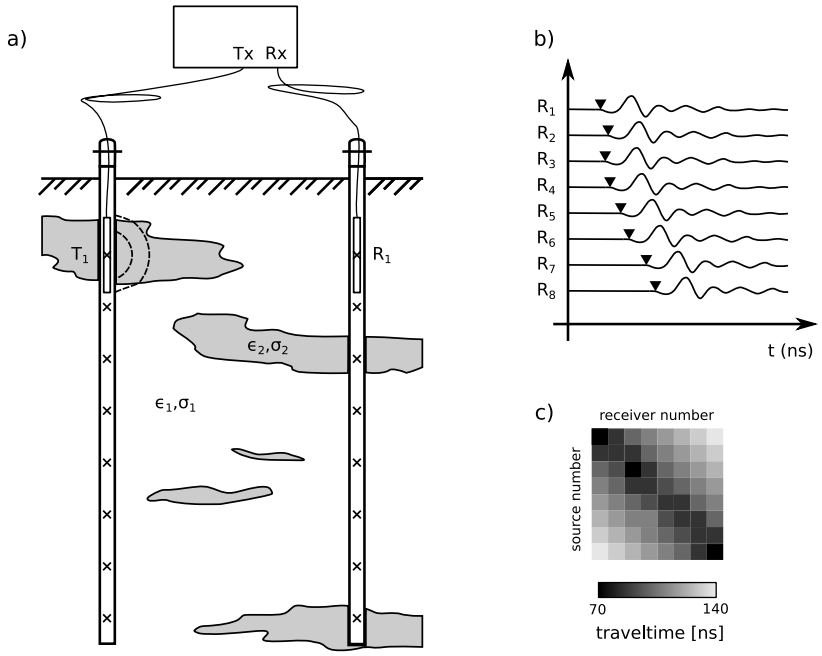


Figure 2.1: (a) Sketch of a cross-hole GPR field setup: console connected to transmitter (Tx) and receiver (Rx) antennas located in different boreholes. Antennas are shown in the position closest to the ground (denoted T_1 and R_1, respectively) and the $\times$'s show the all the positions for which data is acquired. Subsurface is composed of two different materials with different electromagnetic properties. Dashed lines depict the wavefront at two different times after the source pulse is emitted assuming $\epsilon_1 > \epsilon_2$. (b) Full-waveform data collected at all the receivers (subset of a MOG) for the first transmitter (T_1). Triangles mark the first arrivals. (c) Data matrix formed with the traveltimes for all sources and receivers.

Material	Conductivity, σ (mS/m)	Relative permittivity, ϵ_r
Air	0	1
Freshwater	0.1–10	78–88
Saltwater	4000	81–88
Clay (dry)	1–100	2–20
Clay (wet)	100–1000	15–40
Sand (dry)	10^{-4}–1	3–6
Sand (wet)	0.1–10	10–30

Table 2.1: Electromagnetic properties of some subsurface materials at 100 MHz. Values taken from Cassidy (2009).

$$v = \frac{1}{\epsilon\mu_0} \tag{2.1}$$

In this way, the wave will travel faster in materials with lower permittivity (as sketched by the wavefronts in Fig. 2.1a). In general, a model of permittivity is useful even if one aims to use the full-waveform data, e.g. the starting model for full-waveform inversion is usually obtained from a traveltime inversion.

The permittivity of materials is usually expressed with respect to that of the vacuum as:

$$\epsilon = \epsilon_r \epsilon_0 \tag{2.2}$$

where $\epsilon_0 = 8.854 \times 10^{-12}\, F \cdot m^{-1}$ and ϵ_r is the relative permittivity of the material. Permittivity in the subsurface is mainly related to water content because the relative permittivity of water for GPR frequencies is much higher than that of sediments or air (Table 2.1). Therefore permittivity may be approximated with a petrophysical relation by knowing e.g. the permittivity of the rock, the saturation and the porosity (Day-Lewis, 2005). For saturated or partially saturated subsurface, the main factors impacting the permittivity are then porosity and water retention capacity. They both are good indicators to distinguish subsurface media, e.g. in partially saturated subsurface, a (poorly sorted) glacial till usually has higher water retention due to clay content than a (well sorted) sand (as shown in Chapter 5).

2.2 Geophysical forward operator: electromagnetic wave traveltime

The propagation of electromagnetic waves is described by Maxwell's equations. These are a set of coupled PDE that describe the behavior of the electromagnetic field in space and time (Nabighian, 1987; Zhdanov, 2018). Under some assumptions, wave traveltimes may be computed with simplified equations (Chap. 3, Born and Wolf, 1980). Taking as starting point the spectral Maxwell's equations for harmonic fields:

$$\nabla \times \mathbf{E} = -j\omega\mu\mathbf{H} - \mathbf{J}_{ms} \tag{2.3}$$

$$\nabla \times \mathbf{H} = -j\omega\epsilon\mathbf{E} + \sigma\mathbf{E} + \mathbf{J}_{es} \tag{2.4}$$

$$\nabla \cdot \epsilon\mathbf{E} = \rho \tag{2.5}$$

where $\mathbf{E}$ is the electric field, $\mathbf{H}$ is the magnetic field, j is the imaginary unit, ω is the (angular) frequency, μ is the magnetic permeability, ϵ is the (dielectric) permittivity, σ is the electric conductivity, ρ is the electric charge density and $\mathbf{J}_{es}$ and $\mathbf{J}_{ms}$ denote electric and magnetic sources, respectively. Substituting for $\mathbf{H}$ in Eq. 2.3 one obtains (Chap. 5, Solimini, 2016):

$$\nabla\nabla \cdot \mathbf{E} - \nabla^2\mathbf{E} = \kappa^2\mathbf{E} \tag{2.6}$$

where κ is the propagation constant defined as $\kappa^2 = \omega^2\mu_0\epsilon$. Considering also that Eq. 2.5 in a neutral inhomogeneous material (for which $\rho = 0$), yields:

$$\nabla \cdot (\epsilon\mathbf{E}) = \mathbf{E} \cdot \nabla\epsilon + \epsilon\nabla \cdot \mathbf{E} = \rho = 0$$

from which:

$$\nabla \cdot \mathbf{E} = -\frac{\mathbf{1}}{\epsilon}\mathbf{E} \cdot \nabla\epsilon \tag{2.7}$$

Substituting Eq. 2.7 in 2.6, then the equation for the electric field is:

$$\nabla^2\mathbf{E} + \kappa^2\mathbf{E} + \nabla\left(\mathbf{E}\cdot\frac{\nabla\epsilon}{\epsilon}\right) = 0 \tag{2.8}$$

When either the spatial rate of variation of ϵ or $\kappa^2 \to \infty$, Eq. 2.8 is simplified to:

$$\nabla^2\mathbf{E} + \kappa^2(\mathbf{r})\mathbf{E} \simeq 0 \tag{2.9}$$

where $\mathbf{r}$ is a spatial position vector and denotes that the propagation constant changes with position. This approximation is usually referred to as geometrical optics or ray approximation (Born and Wolf, 1980; Solimini, 2016). This is a Helmholtz equation and its solution for the electric field in the inhomogeneous material may be approximated with the Luneburg-Kline asymptotic expansion (Courant and Hilbert, 1989):

$$\mathbf{E}(\mathbf{r}) = e^{-j\kappa_0\phi(\mathbf{r})}\sum_{m=0}^{\infty}\frac{\mathbf{E}_m(\mathbf{r})}{(j\kappa_0)^m} \tag{2.10}$$

where κ_0 is the propagation constant in vacuum, ϕ is the phase of the field (normalized by κ_0), $\mathbf{E}_m(\mathbf{r})$ are functions determined by the field equations and m is an index for the order of approximation. One can derive the eikonal equation in terms of the phase ϕ by substituting Eq. 2.10 in Eq. 2.9 and considering only zeroth order terms (m=0):

$$|\nabla\phi(\mathbf{r})|^2 = \left(\frac{\kappa(\mathbf{r})}{\kappa_0}\right)^{-2} \tag{2.11}$$

Then, writing Eq. 2.11 in terms of the traveltime τ:

$$|\nabla\tau(\mathbf{r})|^2 = v(\mathbf{r})^{-2} \tag{2.12}$$

where v is the wave (phase) velocity given by Eq. 2.1. After discretization, the model vector is then $\mathbf{m} = \mathbf{v}$ and the data vector is $\mathbf{d} = \boldsymbol{\tau}$.

The eikonal equation in Eq. 2.12 may be used to compute the traveltimes by defining the boundary condition $\tau = 0$ for the sources and the distribution of ϵ. Note that for GPR frequencies and sharp transitions between different subsurface materials, the ray approximation is usually not valid and an error is introduced, i.e.

scattering of low-frequency waves is not adequately modeled. However, the error introduced is often of the same order of magnitude to that of the measurement error (Hansen et al., 2014), thus using the eikonal equation to compute traveltimes provides generally sufficient accuracy. Moreover, solving the eikonal equation is computationally more efficient than solving equations that explicitly consider scattering e.g. a full-waveform simulation (Zelt and Chen, 2016) which expedites testing and uncertainty quantification.

2.3 Numerical approximation of forward operator and computation of derivatives

Different numerical algorithms have been used to approximate the solution of the eikonal equation in Eq. 2.12. In this work, two different algorithms are applied: a shortest path method and a fast-marching method. Both rely on spatially discretizing the domain of interest in velocity (or permittivity) cells and both are more physically realistic compared to a linear straight-ray approach which neglects that the travel path depends on velocity heterogeneities. However, the two have different ways to control accuracy and also different ways to compute the derivatives needed for gradient-based inversion.

The shortest path (graph) method is based on Dijkstra's algorithm to compute the fastest path in a network of nodes. One may define the possible connections (or routes) between the nodes by different templates. In general, the higher the order of the template (more possible connections) the more accurate the traveltime computations but also the higher the computational demand. In this work, the algorithm proposed by Giroux and Larouche (2013) and implemented in PyGIMLi (Rücker et al., 2017) is used. This algorithm puts secondary nodes in the faces of velocity cells to define the template 2.2.

Gradient-based inversion may be done by linearizing the forward operator and obtaining the gradient of the l_2-norm objective function (1.3) as:

$$\nabla_{\mathbf{m}}\gamma(\mathbf{m}) = -\mathbf{J}(\mathbf{m})^T(\mathbf{d} - \mathbf{f}(\mathbf{m})) \tag{2.13}$$

where J is the $Q \times D$ Jacobian matrix:

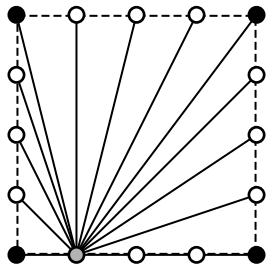


Figure 2.2: Nodes for shortest path method: a velocity cell (dashed line), nodes for velocity cells (black dots), secondary nodes (white dots), raypaths (solid lines) defining a template for the arrival node (gray dot) which is itself a secondary node.

$$[\mathbf{J}(\mathbf{m})]_{i,j} = \frac{\partial f_i(\mathbf{m})}{\partial m_j} \tag{2.14}$$

For the shortest path method, the elements of the Jacobian (which are derivatives) are computed by taking the length of the rays that are traced with the shortest path between each combination of source and receiver. As a result, only the cells traversed by at least one ray have a sensitivity different than zero. In Fig. 2.3b,d the sensitivity for all rays (with source-receiver offset less than 30 degrees) and for an individual ray is shown for a synthetic subsurface model (Fig. 2.3a). Notice that sensitivities are effectively focused only in rays whose paths are clearly controlled by the velocity heterogeneities.

The Fast-Marching method used in this work relies on a factorized version of the eikonal equation and the implementation of Treister and Haber (2016). The factorized equation helps to reduce the error induced by spatial discretization in the proximity of the sources. Fast-Marching methods use a heap sort algorithm and a finite-difference scheme to propagate a wave (or interface) front from the sources to the receivers. The same implementation allows one to efficiently compute the product $\mathbf{J}(\mathbf{m})^T(\mathbf{d} - \mathbf{f}(\mathbf{m}))$ which is a measure of sensitivity of traveltimes with respect to the velocity cells. This product is given by the solution of a triangular system exploiting the Fast-Marching sort order of the forward operator

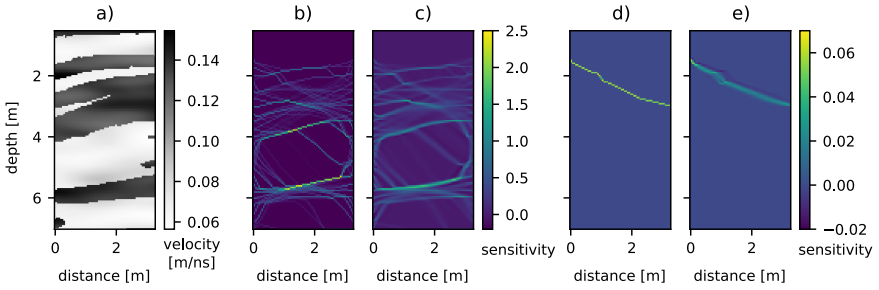


Figure 2.3: Sensitivity of forward operators: Velocity subsurface model (a), sum of sensitivity considering all combinations of sources and receivers for the shortest path method (b) and the fast marching method (c), sensitivity for a source at 1.5 m and a receiver at 3.0 m depth for the shortest path method (c) and the fast marching method (e).

(Treister and Haber, 2016). This means that one avoids computing individually each derivative of the Jacobian, as done in the shortest path method. The spatial sensitivity for the model in Fig. 2.3a for all sources and receivers and for an individual ray path is shown in Fig. 2.3c,e. Notice that due to the finite difference approximation and the size of the grid cells, the rays are not entirely focused in rays. This resembles the Fresnel zone that results when explicitly considering the finite-frequency of the waves: the traveltime is computed for a frequency whose wavelength is on the order of the node spacing (Zelt and Chen, 2016). Though this effect generally does not affect the accuracy of the forward operator, it is generally not possible to tune it to the center frequency of source wavelets.

Using the corresponding implementations, the computational time for both the forward simulation and the computation of the Jacobian product is about 10 times lower for the fast marching method compared to the shortest path method. The shortest path method was initially considered and used in Chapters 4 and 3. Since Chapter 5 required more extensive testing (i.e. more forward model simulations), the choice was made to switch to the fast marching method in order to reduce computational time.

2.4 Deterministic cross-borehole GPR tomography

For a nonlinear forward operator, the traditional deterministic inversion also referred to as tomography is usually done by adding a regularization term to the objective function in Eq. 1.2 which is then rewritten as:

$$\gamma(\mathbf{m}) = \|\mathbf{f}(\mathbf{m}) - \mathbf{d}\|_2^2 + \alpha\|\mathbf{L}\mathbf{m}\|_2^2 \tag{2.15}$$

where α is a regularization factor and $\mathbf{L}$ is a regularization (or roughening) operator. For instance, $\mathbf{L}$ may be chosen to be the finite-difference approximation of the second order spatial derivatives (Laplacian), then the regularization term penalizes solutions that are rough in terms of the second order derivatives, i.e. it favors smooth models. The gradient for Eq. 2.15 is then computed by:

$$\nabla_{\mathbf{m}}\gamma(\mathbf{m}) = -\mathbf{J}(\mathbf{m})^T(\mathbf{d} - \mathbf{f}(\mathbf{m})) + \alpha\mathbf{L}^T\mathbf{L}\mathbf{m} \tag{2.16}$$

which then might be directly optimized by using the gradient-descent method. However, for this objective function there are optimization methods with faster convergence such as the Gauss-Newton method. This method requires solving iteratively for $\Delta\mathbf{m}$ in (see e.g. Aster et al., 2013; Rücker et al., 2017):

$$(\mathbf{J}(\mathbf{m})^T\mathbf{J}(\mathbf{m}) + \alpha\mathbf{L}^T\mathbf{L})\Delta\mathbf{m} = \mathbf{J}(\mathbf{m})^T(\mathbf{d} - \mathbf{f}(\mathbf{m})) - \alpha\mathbf{L}^T\mathbf{L}\mathbf{m} \tag{2.17}$$

Such simple regularization terms may result in high-resolution and realistic models when the subsurface structure is well constrained by the data (i.e. high angular coverage between the boreholes, small transmitter/receiver spacing down each borehole, high-quality traveltime picks). However, when this is not the case (i.e. the dataset is not sufficiently informative) the inverted models tend to show structures that are not penalized by the chosen regularization. An example of a model obtained with regularization that favors smooth models is shown in Fig. 1.1c. In this case, however, the used regularization factor is relatively low (10^{-5}) and therefore some artifacts resulting from noise fitting are still visible (compare to the truth model in Fig. 1.1b). A higher regularization factor would have

resulted, however, in more blurry limits between the materials which is neither the case. This is a typical problem with prior information that is expressed by standard regularization choices which in most cases is not realistic enough to adequately represent highly structured subsurface.

Chapter 3

Reducing model dimension for inversion: deep generative models to represent highly structured spatial patterns[1]

When solving inverse problems in geophysical imaging, deep generative models (DGMs) may be used to enforce the solution to display highly structured spatial patterns which are supported by independent information (e.g. the geological setting) of the subsurface. In such case, inversion may be formulated in a latent space where a low-dimensional parameterization of the patterns is defined and where Markov chain Monte Carlo or gradient-based methods may be applied. However, the generative mapping between the latent and the original (pixel) representations is usually highly nonlinear which may cause some difficulties for inversion, especially for gradient-based methods. In this contribution we review the conceptual framework of inversion with DGMs and propose that this nonlinearity is caused mainly by changes in topology and curvature induced by the generative

[1]**Note:** The research presented in this chapter is based on: Lopez-Alvis, J., Laloy, E., Nguyen, F., and Hermans, T. (2020). Deep generative models in inversion: A review and development of a new approach based on a variational autoencoder. ArXiv:2008.12056 [Physics]. http://arxiv.org/abs/2008.12056. *Submitted to Computers and Geosciences.*

function. As a result, we identify a conflict between two goals: the accuracy of the generated patterns and the feasibility of gradient-based inversion. In addition, we show how some of the training parameters of a variational autoencoder, which is a particular instance of a DGM, may be chosen so that a tradeoff between these two goals is achieved and acceptable inversion results are obtained with a stochastic gradient-descent scheme. A series of test cases using synthetic models with channel patterns of different complexity and cross-borehole traveltime tomographic data involving both a linear and a nonlinear forward operator show that the proposed method provides useful results and performs better compared to previous approaches using DGMs with gradient-based inversion.

3.1 Introduction

A common task in the geosciences is to solve an inverse problem in order to obtain a model (or image) from a set of measurements sensing a heterogeneous spatial domain. When characterizing subsurface environments, the corresponding inverse problem is usually ill-posed yielding non-unique and potentially unstable solutions. This is mainly because the measurements do not provide sufficiently independent information on the distribution of subsurface properties. In such cases it is possible to constrain the solution to allow only certain spatial patterns. In practice, such patterns may be supported by independent (prior) information of the sensed domain (e.g. knowledge of the geological setting) and used with the aim of appropriately reconstructing heterogeneity. Classical regularization may be used to impose the model to be smooth or of minimum magnitude (Tikhonov and Arsenin, 1977) but in many cases this does not yield satisfactory results in areas poorly constrained by the data (Hermans et al., 2012; Caterina et al., 2014). Recently, the use of deep generative models (DGMs) to constrain the solution space of inverse problems has been proposed so that resulting models have specific spatial patterns (Bora et al., 2017; Laloy et al., 2017; Hand and Voroninski, 2018; Seo et al., 2019). DGMs can deal with realistic (natural) patterns which are not captured by classical regularization or random processes defined by second-order statistics (Linde et al., 2015). In this way, inversion with DGMs provides an alternative to inversion with either multiple-point geostatistics (MPS) (Caers

and Hoffman, 2006; González et al., 2008; Hansen et al., 2012; Linde et al., 2015; Rezaee and Marcotte, 2018) or example-based texture synthesis (ETS) (Zahner et al., 2016). While other methods exist that are also able to produce realistic models with inversion e.g. using plurigaussian fields (Armstrong et al., 2011; Liu and Oliver, 2005), they are usually not as flexible as DGMs, MPS or ETS in terms of the patterns they can generate.

All the previously mentioned methods generally rely on gridded representations for the models (i.e. by dividing the spatial domain in cells or pixels). They all require a large number of training examples of the desired patterns to work, which are usually provided as a large training image (or exemplar). However, the procedure for generating a model with MPS or ETS differs from that of DGMs. Both MPS and ETS build the models sequentially (i.e. pixel by pixel or patch by patch) either by directly sampling from the training image (Mariethoz et al., 2010) or by sampling from an empirical probability distribution that was previously obtained from the training image (Strebelle, 2002). In contrast, DGMs rely on a generative function and a low-dimensional reparameterization that follows a known probability distribution. The DGM is first trained with many examples of the desired patterns (e.g. many croppings of the training image) to obtain the generative function. A model is then generated by taking one sample from the low-dimensional probability distribution and passing it through the generative function. This low-dimensional reparameterization is often referred to as latent vector and the space where it is represented is called the latent space. Note that finding a low-dimensional representation is generally feasible for highly structured spatial patterns. The usual geometric argument for this statement is as follows: any gridded model may be represented as a vector in "pixel" space (a space where each pixel is one dimension) and when the models are restricted to those with certain spatial patterns, their vectors will take up only a subset of this pixel space. This subset usually defines a manifold of lower dimensionality than the pixel space (Fefferman et al., 2016) and the latent space is simply a low-dimensional space where such manifold is represented.

Most inversion methods require a perturbation step to search for models that fit the data but such a step is not straightforward to compute for highly structured patterns (Linde et al., 2015; Hansen et al., 2012). The latent space of DGMs pro-

vides a useful frame to compute a perturbation step (Laloy et al., 2017) or even a local gradient-descent direction (Laloy et al., 2019) which generally results in better exploration of the posterior distribution and/or faster convergence compared to inversion with MPS or ETS. So far, inversion with DGMs has been done successfully with regular MCMC sampling methods (Laloy et al., 2017, 2018). However, when applicable, gradient-based methods may be preferred given their lower computational demand. Gradient-based deterministic inversion with DGMs has been pursued with encouraging results (Richardson, 2018, Laloy et al., 2019), however, convergence to the true model was shown to be dependent on the initial model. In the framework of probabilistic inversion, MCMC methods that use the gradient to guide the sampling in the latent space have shown to be less prone to get trapped in local minima than gradient-based deterministic methods while they are also expected to reach convergence faster than regular MCMC (Mosser et al., 2018). A different inversion strategy that has also been applied successfully with DGMs and has a relatively low computational cost is the Ensemble Smoother (Canchumuni et al., 2019; Mo et al., 2020).

Recently, Laloy et al. (2019) studied the difficulties of performing gradient-based deterministic inversion with a specific DGM. They concluded that the nonlinearity of their generative function or "generator" (i.e. the mapping from the latent space to the pixel space) was high enough to hinder gradient-based optimization, causing the latter to often fail in finding the global minimum even when the objective function was known to be convex (in pixel space). In order to approximate manifolds of realistic patterns, most common DGMs involve (artificial) neural networks with several layers and nonlinear (activation) functions. For a specific subsurface pattern, the degree of nonlinearity of the generative function may be controlled mainly by its architecture and the way it is trained (Goodfellow et al., 2016). Regarding difference in training, two common types of DGMs can be distinguished: generative adversarial networks (GANs) (Goodfellow et al., 2014) and variational autoencoders (VAEs) (Kingma and Welling, 2014). VAEs training relies on a variational inference strategy where a DNN is used to approximate the required variational distribution. Such distribution is equivalent to a probabilistic encoder (see details in Section 3.2.3). GANs training is based on adversarial learning: the generator is trained together with a discriminator in

such a way that the models generated by the former are aimed to fool the latter. In both cases training generally takes the form of optimizing a loss function, but in the case of GANs one has to alternate between optimizing the generator and the discriminator. GANs and VAEs require specification of a probability distribution in the latent space and an architecture for the discriminator or encoder (respectively) in addition to the one for their generators. They might also require other parameters to be specified such as the weights on the different terms of the loss function. Frequently, some of these choices use default values, but generally all of them may affect the degree of nonlinearity of the generator (Rolinek et al., 2019).

Given all the possible choices to train the generator it is interesting to investigate whether one can find those that allow both for a good reproduction of the patterns and a good performance of less computationally demanding gradient-based inversion. In this chapter, we review some of the difficulties of performing inversion with DGMs and show how to obtain a well-balanced tradeoff between accuracy in patterns and applicability of gradient-based methods. In particular, we propose to use the training choice of a VAE as DGM and to select some of its parameters in order to achieve good results with gradient-based inversion. Then, we compare this to the training choice of a GAN that has been tested with gradient-based inversion in prior studies (Laloy et al., 2019; Richardson, 2018). Furthermore, we show that since the resulting VAE inversion is only mildly nonlinear, modified stochastic gradient-descent (SGD) methods are generally sufficient to avoid getting trapped in local minima and provide a better alternative than regular gradient-based methods while also retaining a low computational cost.

The remainder of this chapter is structured as follows. Section 3.2.1 explains DGMs and their conceptualization as approximating the real (pattern) manifold. In Section 3.2.2 the use of DGMs to represent prior information in inversion and the difficulties of performing gradient-based inversion are reviewed. Sections 3.2.3 and 3.2.4 show how to use a VAE and SGD to cope with some of the mentioned difficulties. Then, Section 3.3 shows some results of the proposed approach. Section 3.4 discusses the obtained results and points to some remaining challenges. Finally, Section 5.4 presents the conclusions of this chapter.

3.2 Methods

3.2.1 Deep generative models (DGM) to represent realistic patterns.

The term "deep learning" generally refers to machine learning methods that involve several layers of multidimensional functions. This general "deep" setting has been shown to allow for complex mappings to be accurately approximated by building a succession of intermediate (simpler) representations or concepts (Goodfellow et al., 2016). Consider, for instance, deep neural networks (DNNs) which are mappings defined by a composition of a set of (multidimensional) functions $\boldsymbol{\phi}_k$ as:

$$\mathbf{g}(\mathbf{x}) = (\boldsymbol{\phi}_L \circ \cdots \circ \boldsymbol{\phi}_2 \circ \boldsymbol{\phi}_1)(\mathbf{x}) \tag{3.1}$$

where $\mathbf{x}$ is a multidimensional (vector) input, $k = \{1, \ldots, L\}$ denotes the function (layer) index and composition follows the order from right to left. Furthermore, each $\boldsymbol{\phi}_k$ is defined as:

$$\boldsymbol{\phi}_k(\boldsymbol{\xi}) = \psi_k(\mathbf{A}_k\boldsymbol{\xi} + \mathbf{b}_k) \tag{3.2}$$

in which ψ_k is a (nonlinear) activation function, $\mathbf{A}_k$ is a matrix of weights, $\mathbf{b}_k$ is a vector of biases and $\boldsymbol{\xi}$ denotes the output of the previous function (layer) $\boldsymbol{\phi}_{k-1}$ for $k > 1$ or the initial input $\mathbf{x}$ for $k = 1$. Then, training the DNN involves estimating the values for all the parameters $\boldsymbol{\theta} = \{\mathbf{A}_k, \mathbf{b}_k \mid 1 \leq k \leq K\}$ where each $\mathbf{A}_k$ or $\mathbf{b}_k$ may be of different dimensionality depending on the layer. In practice, the number of parameters $\boldsymbol{\theta}$ for such models may reach the order of 10^6, therefore training is achieved by relying on autodifferentiation (see e.g. Paszke et al., 2017) and fast optimization techniques based on SGD (see e.g. Kingma and Ba, 2017), both usually implemented for and run in highly parallel (GPU) computing architectures.

A deep generative model (DGM) is a particular application of such deep methods (Salakhutdinov, 2015). In a DGM a set of training examples $\mathbf{M} = \{\mathbf{m}^{(i)} \mid 1 \leq i \leq T\}$ and a simple low-dimensional probability distribution $p(\mathbf{z})$ are used to learn a model $\mathbf{g}(\mathbf{z})$ that is capable of generating new samples of $\mathbf{m}$

(which are consistent with the training set) by using as input samples from $p(\mathbf{z})$. This can be written as:

$$\mathbf{m} = \mathbf{g}(\mathbf{z}),\ \mathbf{z} \sim p(\mathbf{z}) \tag{3.3}$$

where $\mathbf{g}(\mathbf{z})$ is referred to as the "generator" and $\mathbf{z}$ denotes a vector of latent variables or "code". While the training (and generated) samples $\mathbf{m}$ are usually represented in a high-dimensional space $\mathbb{R}^N$, the probability distribution $p(\mathbf{z})$ is defined in a low-dimensional space $\mathbb{R}^n$. The space $\mathbb{R}^N$ is often referred to as "ambient space" while the space $\mathbb{R}^n$ is called the "latent space". Fig. 3.1 shows a schematic representation of the general setting of DGMs with inversion where (a) and (c) show an ambient space with $N = 3$ and a latent space with $n = 2$. A typical application of DGMs is the generation of images (see e.g. Kingma and Welling, 2014; Goodfellow et al., 2014) for which the ambient space is just the pixel space. Gridded representations of subsurface models may be seen as two- or three-dimensional images of the subsurface.

The underlying assumption in DGMs is that real-world data are generally structured in their high-dimensional ambient space $\mathbb{R}^N$ and therefore have an intrinsic lower dimensionality—such assumption is known in machine learning literature as the manifold hypothesis (Fefferman et al., 2016) because it states that high-dimensional data usually lie on (or lie close to) a lower-dimensional manifold $\mathcal{M} \subset \mathbb{R}^N$. For instance, when studying a subsurface region it is usually assumed that geological processes gave it certain degree of structure then, to allow for a flexible base on which to represent the distribution of the different subsurface materials, the region is usually divided in homogeneous pixels (or cells). Such gridded representation "lives" in the high-dimensional pixel space (the ambient space) but since it has some structure there should be a lower dimensional space (the latent space) where the same distribution of subsurface materials might be represented. Technically, while both the latent space $\mathbb{R}^n$ and the manifold $\mathcal{M}$ are usually low-dimensional, they may differ in dimensionality and/or the manifold may only occupy a certain portion of the latent space (e.g. the shaded region in Fig. 3.1c). Manifolds are geometrical objects that have a topology and a curvature. A topology is the structure of a geometrical object that is preserved

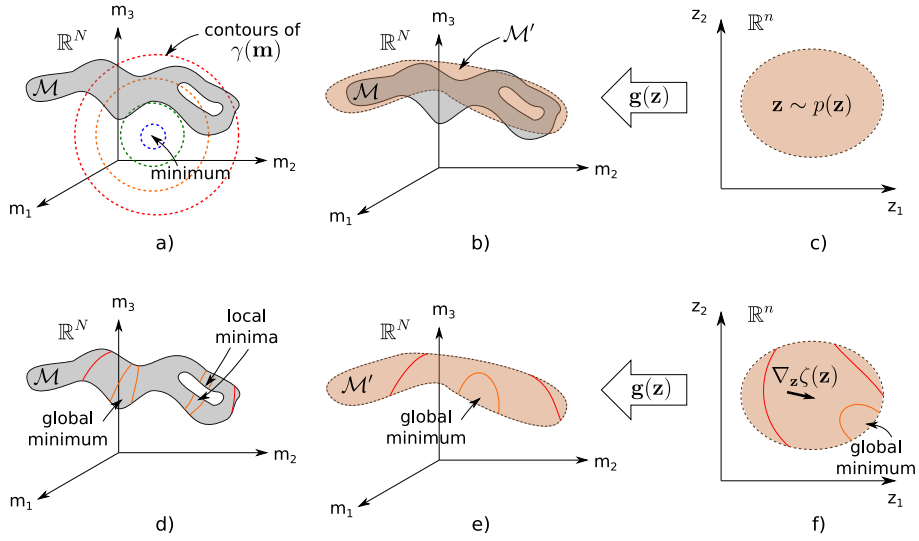


Figure 3.1: Sketch of the different parts involved in DGMs with inversion: approximation of the real manifold (a–c) and the impact of the approximated manifold in inversion (d–f). (a) Real manifold $\mathcal{M}$ and inversion's misfit function $\gamma(\mathbf{m})$ in ambient space $\mathbb{R}^N$. (b) Approximate manifold $\mathcal{M}'$ overlaying the real manifold. (c) Region of latent space $\mathbb{R}^n$ where the approximate manifold is implicitly defined by the probability distribution $p(\mathbf{z})$. (d) Misfit function contours intersected by the real manifold. (e) Misfit function contours intersected by approximate manifold. (f) Misfit function contours back-mapped onto the latent space and the related gradient $\nabla_{\mathbf{z}}\zeta(\mathbf{z})$ computed at one iteration.

under continuous deformations (e.g. stretching or bending). In other words, when a non-continuous operation such as gluing or tearing occurs the topology of the object changes. These changes may be described in terms of different topological properties such as compactness, connectedness and simple-connectedness. In this work, the concept of curvature is used to state that in general one starts with a "flat" domain in the latent space and then one has to curve it to fit the real manifold. In this way, the concept helps to understand where part of the nonlinearity of the generative function comes from. While formal definitions of curvature exist (e.g. Riemannian curvature as applied to smooth manifolds) they are not used in this work.

Considering the manifold assumption described above, a DGM may be regarded as a model to implicitly approximate the "real" manifold $\mathcal{M}$ by generating samples that closely follow such manifold, i.e. that lie on an approximate manifold $\mathcal{M}'$ (Fig. 3.1b). Samples of this approximate manifold are generated by sampling first from a simple probability distribution $p(\mathbf{z})$ in latent space (e.g. a normal or uniform distribution) and then passing them through the generator $\mathbf{g}(\mathbf{z})$. Since the probability distribution $p(\mathbf{z})$ defines indirectly a region (or subset) in latent space that generally has a different curvature and topology than the real manifold, the generator $\mathbf{g}(\mathbf{z})$ must be able to approximate both curvature and topology when mapping the samples of $p(\mathbf{z})$ to ambient space. This generally requires the generator to be a highly nonlinear function. As an instance, consider the case of certain spatial patterns whose real manifold is a highly curved surface with "holes" in ambient space and the (input) region defined by a uniform $p(\mathbf{z})$ is a (flat) plane in a two-dimensional latent space. Regarding their topological properties, one technically says that this plane is simply connected while the real manifold is not (see e.g. Kim and Zhang, 2019). Then, the generative function has to deform this plane in such a way as to approximate (or cover) the real manifold as close as possible. An important property of DGMs is that since a probability distribution in latent space is used, the sample "density" of such plane (and its mapping) also plays an important role. For instance, the generative function may approximate the "holes" of the real manifold by creating regions of very low density of samples when mapping to ambient space (to picture this one can imagine locally stretching a flexible material without changing its curvature). The com-

bined deformation needed to curve the plane and to "make" the holes causes the generative function to be highly nonlinear. Note that when considering a DGM that uses a DNN with rectified linear unit (ReLU) activation functions as generator $\mathbf{g}(\mathbf{z})$, it is also possible for $\mathbf{g}(\mathbf{z})$ to change topology of the input by "folding" transformations (Naitzat et al., 2020).

While one should always strive to accurately approximate the real manifold, since a finite set of training samples is used a tradeoff between accuracy and diversity in the generated samples may be a better objective. Indeed, the use of the prescribed probability distributions is done to continuously "fill" the space between the samples and therefore generate samples of a continuous manifold. Recent success—in terms of accuracy and diversity of generated samples—has been achieved with two DGMs that are based on deep neural networks (DNNs): generative adversarial networks (GANs) (Goodfellow et al., 2014) and variational autoencoders (VAEs) (Kingma and Welling, 2014). The generator $g(\mathbf{z})$ on both strategies is a mapping from low-dimensional input $\mathbf{z} \in \mathbb{R}^n$ to high-dimensional output $\mathbf{m} \in \mathbb{R}^N$. In contrast, the mappings corresponding to the discriminator and the encoder take high-dimensional inputs $\mathbf{m}$ and return low-dimensional outputs.

3.2.2 Gradient-based inversion with DGMs

DGMs may be used with inversion of subsurface data $\mathbf{d}$ to obtain geologically realistic spatial distributions of physical properties $\mathbf{m}$ (Laloy et al., 2017). While this is also possible with traditional deterministic inversion where a regularization term is added directly in Eq. (1.2) (i.e. in ambient space) to obtain models with the imposed structures that minimize the misfit (Lange et al., 2012; Caterina et al., 2014), DGMs are more flexible because they can simultaneously enforce different kind of patterns provided they are trained with samples of all such patterns (Bergmann et al., 2017). In the DGM setting, the low-dimensional samples $\mathbf{z}$ that input to the generator $\mathbf{g}(\mathbf{z})$ may be seen as defining a low-dimensional parameterization (or encoding) of realistic patterns $\mathbf{m}$ and therefore exploration of the set of feasible models may be done in the latent space $\mathbb{R}^n$, as long as the search is done within the region where the approximated manifold $\mathcal{M}'$ is defined

(depicted by shading in Fig. 3.1c).

Since the misfit $\gamma(\mathbf{m})$ is typically defined in ambient space $\mathbb{R}^N$ (e.g. in Fig. 3.1a), gradient-based inversion with DGMs may be seen as optimizing the intersection of $\gamma(\mathbf{m})$ with the approximate manifold $\mathcal{M}'$ (Fig. 3.1e). Such intersected misfit is mapped into the latent space (Fig. 3.1f) and may be expressed as $\gamma(\mathbf{g}(\mathbf{z}))$. Also note that when probability distributions $p(\mathbf{z})$ with infinite support are used (e.g. a normal distribution), one can guide the search in the latent space by adding controlling (regularization) terms to the mapped misfit (see e.g. Bora et al., 2017) and the resulting objective function may be written as:

$$\begin{aligned}\zeta(\mathbf{z}) &= \gamma(\mathbf{g}(\mathbf{z})) + \lambda R(\mathbf{z}) \\ &= \|\mathbf{f}(\mathbf{g}(\mathbf{z})) - \mathbf{d}\|^2 + \lambda R(\mathbf{z})\end{aligned} \tag{3.4}$$

where $R(z)$ is a regularization term defined in the latent space and λ is the corresponding regularization factor. A derivation of this objective function from a Bayesian point of view is presented in Appendix A. The goal of the regularization term is to make the search consistent with the selected probability distribution, i.e. optimization stays preferentially within the high-density regions in the latent space.

In practice, no exhaustive mapping has to be done and the gradient $\nabla_{\mathbf{z}}\zeta(\mathbf{z})$ is only computed for the points in latent space where optimization lands in each iteration (in Fig. 3.1f the gradient is represented for one iteration). The gradient $\nabla_{\mathbf{z}}\zeta(\mathbf{z})$ is computed by adding a derivative layer corresponding to $\nabla_{\mathbf{m}}\gamma(\mathbf{m})$ to the autodifferentation that was set up for $\mathbf{g}(\mathbf{z})$ while training the DGM (see e.g. Laloy et al., 2019). Such autodifferentiation setup may be seen as implicitly obtaining the Jacobian $\mathbf{J}(\mathbf{z})$ of size $N \times n$ whose elements are:

$$[\mathbf{J}(\mathbf{z})]_{i,j} = \frac{\partial g_i(\mathbf{z})}{\partial z_j} \tag{3.5}$$

Then, the gradient $\nabla_{\mathbf{z}}\zeta(\mathbf{z})$ is obtained from Eq. (3.4) by using the chain rule given by the product of Eqs. (1.3) and (3.5):

$$\begin{aligned}\nabla_{\mathbf{z}}\zeta(\mathbf{z}) &= \nabla_{\mathbf{z}}\gamma(\mathbf{g}(\mathbf{z})) + \lambda\nabla_{\mathbf{z}}R(\mathbf{z}) \\ &= \mathbf{J}(\mathbf{z})^T\nabla_{\mathbf{m}}\gamma(\mathbf{m}) + \lambda\nabla_{\mathbf{z}}R(\mathbf{z}) \end{aligned} \tag{3.6}$$

The latter may also be done implicitly by incorporating directly in the autodifferentiation framework, e.g. putting it on top of the so called computational graph (Richardson, 2018; Mosser et al., 2018).

Even when the considered misfit function $\gamma(\mathbf{m})$ is convex in ambient space $\mathbb{R}^N$ (as depicted by concentric contours in Fig. 3.1a), difficulties to perform gradient-based deterministic inversion may arise due to the generator $\mathbf{g}(\mathbf{z})$ (Laloy et al., 2019). We propose that such difficulties arise because the generator (1) is highly nonlinear and (2) changes the topology of the input region defined by $p(\mathbf{z})$. Both of these properties often cause distances (between samples) in latent space to be significantly different than distances in ambient space. Consider again the example of a real manifold that is a highly curved surface with "holes" in it and a uniform distribution $p(\mathbf{z})$ is used as input to the generator, then the latter might be able to approximate both the curvature and the holes at the cost of increasing nonlinearity and/or changing topology. When considering this backwards—e.g. when mapping the misfit function $\gamma(\mathbf{m})$ in the latent space—the approximation of both high curvature and differences in topology often translate in discontinuities or high nonlinearities because a continuous mapping onto the uniform distribution is enforced. This results in high curvature being effectively "flattened" and holes effectively "glued", both of which cause distances to be highly distorted. In this work, we will call a generator "well-behaved" when it is only mildly nonlinear and preserves topology.

Both the generator's nonlinearity and its ability to change topology, may be controlled by two factors: (1) the generator architecture (type and size of each layer and total number of layers) and (2) the way it is trained (including training parameters). If the goal is to perform gradient-based inversion with DGMs, one should try to preserve convexity of $\gamma(\mathbf{m})$ as much as possible when mapping it to the latent space as $\gamma(\mathbf{g}(\mathbf{z}))$ while not degrading the generator's ability to reproduce the desired patterns. To aid in preserving such convexity, we propose to

enforce the generator $\mathbf{g}(\mathbf{z})$ to be well-behaved. This means that the generator will approximate the real manifold $\mathcal{M}$ with a manifold $\mathcal{M}'$ with a moderate curvature and whose topology is the same as the region defined in latent space by $p(\mathbf{z})$. By enforcing a moderate curvature manifold, local oscillations that may give rise to local minima (as those shown in Fig. 3.1d) but only have minimum impact in pattern accuracy are avoided in the approximate manifold $\mathcal{M}'$ (the local minima are no longer present in Fig. 3.1e). In turn, when the generator is encouraged to preserve topology no more local minima should arise in $\mathbb{R}^n$ than the ones resulting from intersecting $\gamma(\mathbf{m})$ with the approximate manifold $\mathcal{M}'$ in $\mathbb{R}^N$ (note e.g. there is one local minima in both Fig. 3.1e,f). The latter is in line with the proposal of Falorsi et al. (2018), where they argue that for the purpose of representation learning (which basically means learning encodings that are useful for other tasks than just generative modeling) the mapping should preserve topology.

GANs often produce highly nonlinear generators that do not preserve topology, which may result in challenging inversion in the latent space. Laloy et al. (2018) provide an example of how architecture of a GAN is set to obtain a relatively well-behaved generator $\mathbf{g}(\mathbf{z})$. They propose to use a model called spatial generative adversarial network (SGAN) (Jetchev et al., 2017) that enforces different latent variables to affect different local regions in the ambient space. Their architecture results in a high compression (lower dimensionality of the latent space) and controls nonlinearity which allowed them to successfully perform MCMC-based inversion in the latent space. However, gradient-based deterministic inversion performed with the same DGM was shown to be highly dependent on the initial model (Laloy et al., 2019) pointing towards the existence of local minima. In addition, since training GANs is a rather complicated procedure where one has to find a balance between the performance of the generator and the discriminator, there is no straightforward way in which to modify such training to control nonlinearity. In this work we aim for robust gradient-based inversion in latent space by considering a VAE, the other predominant type of DGM, since its training may be tuned to produce a well-behaved generator.

3.2.3 VAE as DGM for inversion

A VAE is the model resulting from using a reparameterized gradient estimator for the evidence lower bound while applying (amortized) variational inference to an autoencoder, i.e. an architecture involving an encoder and a decoder which are both (possibly deep) neural networks (Kingma and Welling, 2014; Zhang et al., 2018). To train a VAE one uses a dataset $\mathbf{M} = \{\mathbf{m}^{(i)} \mid 1 \leq i \leq T\}$ where each $\mathbf{m}^{(i)}$ is a sample (e.g. an image) with the desired patterns and then maximizes the sum of the evidence (or marginal likelihood) lower bound of each individual sample. The evidence lower bound for each sample can be written as (Kingma and Welling, 2014)

$$\mathcal{L}(\theta, \vartheta; \mathbf{m}^{(i)}) = \mathcal{L}^m + \mathcal{L}^z \tag{3.7}$$

with

$$\mathcal{L}^m = \mathbb{E}_{q_\vartheta(\mathbf{z}|\mathbf{m}^{(i)})}[\log(p_\theta(\mathbf{m}^{(i)}|\mathbf{z})] \tag{3.8}$$

and

$$\mathcal{L}^z = -D_{KL}(q_\vartheta(\mathbf{z}|\mathbf{m}^{(i)})||p(\mathbf{z})) \tag{3.9}$$

where $\mathbf{z}$ refers to the codes or latent vectors, $p_\theta(\mathbf{m}|\mathbf{z})$ is the (probabilistic) decoder, $q_\vartheta(\mathbf{z}|\mathbf{m})$ is the (probabilistic) encoder, $\mathbb{E}$ denotes the expectation operator, D_{KL} denotes the Kullback-Leibler distance and, θ and ϑ are the parameters (weights and biases) of the DNNs for the decoder and encoder, respectively.

In order to maximize the evidence lower bound in Eq. (3.7), its gradient with respect to both θ and ϑ is required, however, this is generally intractable and therefore an estimator is used. This estimator is based on a so called reparameterization trick of the random variable $\widetilde{\mathbf{z}} \sim q_\vartheta(\mathbf{z}|\mathbf{m})$ which uses an auxiliary noise $\boldsymbol{\epsilon}$. In the case of a VAE, the encoder is defined as a multivariate Gaussian with diagonal covariance:

$$q_\vartheta(\mathbf{z}|\mathbf{m}) = \mathcal{N}(\mathbf{h}_\vartheta(\mathbf{m}), \mathbf{u}_\vartheta(\mathbf{m}) \cdot I_n) \tag{3.10}$$

where $\mathbf{h}_\vartheta(\mathbf{m})$ and $\log \mathbf{u}_\vartheta(\mathbf{m})$ are modeled with DNNs and I_n is a $n \times n$ identity matrix. Then, the encoder and the auxiliary noise $\boldsymbol{\epsilon}$ are used in the following way during training (Kingma and Welling, 2014)

$$\widetilde{\mathbf{z}} = \mathbf{h}_\vartheta(\mathbf{m}) + \mathbf{u}_\vartheta(\mathbf{m}) \odot \boldsymbol{\epsilon}, \ \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon}) \tag{3.11}$$

where $\odot$ denotes an element-wise product. Often Eq. (3.9) has an analytical solution, then only Eq. (3.8) is approximated with the estimator as (Kingma and Welling, 2014)

$$\widetilde{\mathcal{L}}^m = \frac{1}{M} \sum_{j=1}^{M} \log(p_\theta(\mathbf{m}^{(i)}|\widetilde{\mathbf{z}}^{(i,j)})) \tag{3.12}$$

where $\widetilde{\mathbf{z}}^{(i,j)} = \mathbf{h}_\vartheta(\mathbf{m}^{(i)}) + \mathbf{u}_\vartheta(\mathbf{m}^{(i)}) \odot \boldsymbol{\epsilon}^{(j)}$, $\boldsymbol{\epsilon}^{(j)} \sim p(\boldsymbol{\epsilon})$ and M is the number of samples used for the estimator. Further, if we set the decoder $p_\theta(\mathbf{m}|\mathbf{z})$ as a multivariate Gaussian with diagonal covariance structure, then

$$p_\theta(\mathbf{m}|\mathbf{z}) = \mathcal{N}(\mathbf{g}_\theta(\mathbf{z}), \mathbf{v}_\theta(\mathbf{z}) \cdot I_N) \tag{3.13}$$

where $\mathbf{g}_\theta(\mathbf{z})$ and $\log \mathbf{v}_\theta(\mathbf{z})$ are modeled with DNNs and I_N is a $N \times N$ identity matrix. In this work, we consider only the mean of the decoder $p_\theta(\mathbf{m}|\mathbf{z})$ which is just the (deterministic) generator $\mathbf{g}_\theta(\mathbf{z})$. Then, the corresponding loss function may be written as

$$\widetilde{\mathcal{L}}^m = \frac{1}{M} \sum_{j=1}^{M} \|\mathbf{g}_\theta(\widetilde{\mathbf{z}}^{(i,j)}) - \mathbf{m}^{(i)}\|^2 \tag{3.14}$$

The described setting allows for the gradient to be computed with respect to both θ and ϑ and then stochastic gradient descent is used to maximize the lower bound in Eq. (3.7). In the rest of this work, we drop the subindex θ in $\mathbf{g}(\mathbf{z})$ to simplify notation and also because once the DGM is trained, the parameters θ do not change, i.e. they are fixed for the subsequent inversion.

As previously mentioned, it is often possible to analytically integrate the Kullback-Leibler distance in Eq. (3.9). In this work, we consider that $p(\mathbf{z})$ and $q_\vartheta(\mathbf{z}|\mathbf{m})$ are both Gaussian therefore Eq. (3.9) may be rewritten as (Kingma and

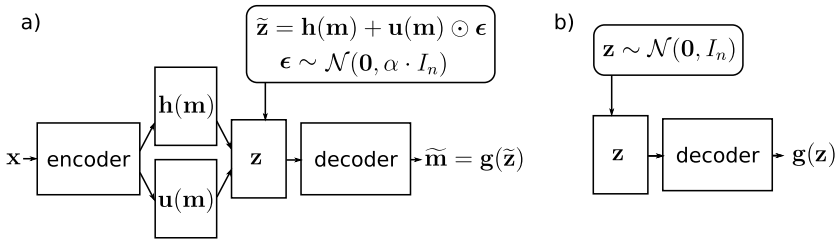


Figure 3.2: A diagram for a VAE: (a) steps needed for training and (b) steps needed for generation.

Welling, 2014):

$$\mathcal{L}^z = \frac{1}{2}\sum_{i=1}^{d}(1 + \log((u_i)^2) - (h_i)^2 - (u_i)^2) \tag{3.15}$$

where the sum is done for the n output dimensions of the encoder.

Note that the term in Eqs. (3.8), (3.12) and (3.14) may be interpreted as a reconstruction term that causes the outputs of the encode-decode operation to look similar to the training samples, while the term in Eqs. (3.9) and (3.15) may be considered a regularization term that enforces the encoder $q_\vartheta(\mathbf{z}|\mathbf{m})$ to be close to a prescribed distribution $p(\mathbf{z})$. In practice, one may add a weight to the second term (Higgins et al., 2017) of the lower bound as:

$$\widetilde{\mathcal{L}}(\theta, \vartheta; \mathbf{m}^{(i)}) = \widetilde{\mathcal{L}}^m + \beta\mathcal{L}^z \tag{3.16}$$

to prevent samples to be encoded far from each other in the latent space, which may cause overfitting of the reconstruction term and degrade the VAE's generative performance. The overall process of training and generation for a VAE is depicted in Fig. 3.2.

Note that in setting up the VAE one has to choose: (1) the architectures of the encoder and decoder, (2) the probability distribution $p(\mathbf{z})$, (3) the noise distribution $p(\boldsymbol{\epsilon})$ and (4) the regularization weight β. As mentioned in Section 3.2.2, these choices may impact the nonlinearity of the generator and its ability to preserve topology, which in turn affect the mapping of the data misfit function

$\gamma(\mathbf{m})$ in latent space and possibly diminish the performance of inversion methods. While different choices in the architecture and probability distribution $p(\mathbf{z})$ may aid in obtaining a well-behaved generator, they are generally not straightforward and highly problem dependent. Therefore in this work we focus on the other two possible controls, the distribution $p(\boldsymbol{\epsilon})$ and the regularization weight β, since they provide the simplest means of improving nonlinearity issues.

The effect of the regularization weight β is such that when increased the encoded training samples tend to lie closer to the prescribed probability distribution $p(\mathbf{z})$. Then, one may picture the transformation of the encoder as taking the low-dimensional approximate manifold in the ambient space and charting it (e.g. by bending, stretching and even folding) into the region defined by $p(\mathbf{z})$ in the latent space and the generator as the transformation undoing such charting. While the effect of β in a VAE is relatively easy to understand, the effect of the noise distribution $p(\boldsymbol{\epsilon})$ is not so straightforward. First, note that the typical choice of a diagonal noise as $p(\boldsymbol{\epsilon}) = \mathcal{N}(\mathbf{0}, \alpha \cdot I_n)$ where α denotes a constant variance (frequently set to $\alpha = 1.0$) is usually done for tractability or computational convenience (Kingma and Welling, 2014; Rolinek et al., 2019). However, it has been proposed recently that the choice of a diagonal noise has an impact on a property called disentanglement (Rolinek et al., 2019). Such disentanglement basically means that different latent directions control different independent characteristics of the training (or generated) samples. They explain that a diagonal $p(\boldsymbol{\epsilon})$ might induce an encoding that preserves local orthogonality of the ambient space. In this work, we argue that the choice of a diagonal $p(\boldsymbol{\epsilon})$ (which is usually done only for computational convenience) might be useful in producing a well-behaved generator.

In order to visualize the joint effect of α and β, Fig. 3.3 shows a synthetic example where samples in a two-dimensional ambient space lie close to a rotated "eight-shaped" manifold (Fig. 3.3a). In addition, to study the impact on inversion, a convex data misfit function $\gamma(\mathbf{m})$ in the same space (created synthetically with a negative isotropic Gaussian function) is shown in Fig. 3.3b. The latent space is also chosen two-dimensional for visualization purposes but recall that for a real case the dimensionality of the latent space is usually much lower than the one of the ambient space. Then, Fig. 3.4 considers nine different combinations for the

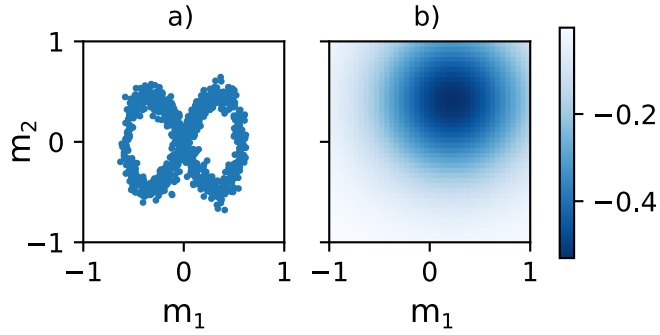


Figure 3.3: Synthetic example of two-dimensional "eight-shaped" manifold: (a) training samples lying close to the manifold, and (b) synthetic misfit function $\gamma(\mathbf{m})$.

values of α and β to show how the (nonlinear) generator $\mathbf{g}(\mathbf{z})$ maps a region of the latent space (denoted by the z-axes in the first three rows) into the ambient space (denoted by the x-axes in the last three rows) in order to approximate the manifold in Fig. 3.3a. To visualize the deformation caused by the generator, an orthogonal grid in the z-axes and its mapping into the x-axes (a deformed grid) are shown (both on the left of each inset). The corresponding encoded training samples are shown in red in the z-axes (left of each inset) and their reconstruction (resulting from the operation of encode-decode) is shown also in red in the x-axes (right of each inset), where also the original training samples are shown (in blue) to assess the accuracy of reconstruction. Samples obtained from a Gaussian distribution with a unitary diagonal covariance $p(\mathbf{z})$ are shown in the z-axes in orange (left of each inset), while their generator-mapped values are shown also in orange in the x-axes (right of each inset). Finally, the mapping of the data misfit function in Fig. 3.3b into the latent space is shown in the z-axes (right of each inset).

It is worth mentioning a few effects visible in the illustrative example of Figs. 3.3 and 3.4. First, note that increasing α seems to cause the grid to be more "rigid" locally (grid lines tend to intersect more at right angles) while going through the generator which may in turn help in preserving topology and controlling non-linearity (e.g. compare the deformation of the grids for different values of α for $\beta = 0.01$), and more importantly, in preserving the convexity of the data mis-

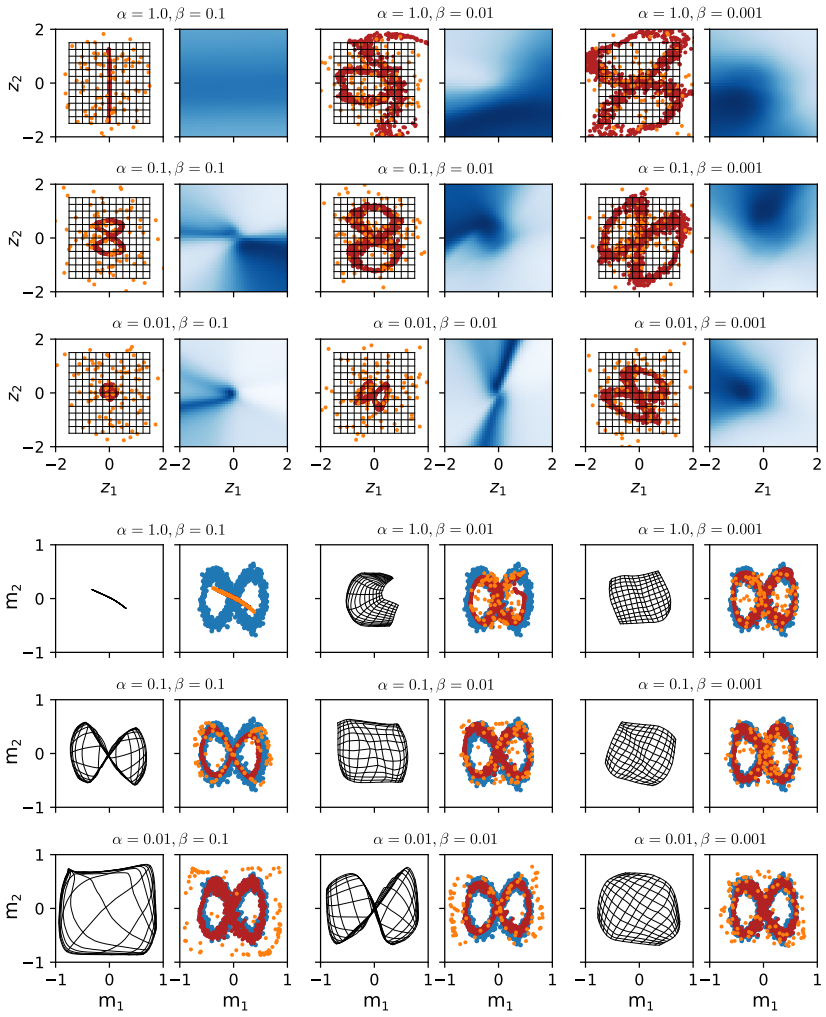


Figure 3.4: Mapping a region of the latent space by the generator $\mathbf{g}(\mathbf{z})$ and mapping of the misfit function $\gamma(\mathbf{m})$ to the latent space with different values for α and β. The first three rows (z-axes) depict the latent space where each case shows: (left frame) orthogonal grid (black), encoded training samples (red) and generated samples (orange); (right frame) misfit function mapped in latent space (blue). The last three rows (x-axes) depict the ambient space where each case shows: (left frame) the same grid but mapped by the generator; (right frame) training (blue), reconstructed (red) and generated samples (orange).

fit function in the latent space (the mapped misfit function using $\alpha = 0.1$ and $\beta = 0.01$ has a single global minimum, while the misfit function for $\alpha = 0.01$ and $\beta = 0.01$ has two minima in latent space). Also note that both α and β should be set in order to not cause a significant degradation in: (1) the reconstruction of the patterns, e.g. the cases of $\alpha = 1.0$ with both $\beta = 0.1$ and $\beta = 0.01$ show that the "eight-shape" is not completely reconstructed (seen in red samples not fully overlaying the blue samples in x-axes), or (2) the similarity of the encoded samples to the prescribed distribution $p(\mathbf{z})$, e.g. the case of $\alpha = 0.01$ and $\beta = 0.1$ shows that encoded samples (red dots in z-axes) are too concentrated (lower variance) and therefore far from the prescribed normal distribution with unit variance (orange dots in z-axes). In this case, the intermediate values ($\alpha = 0.1$ and $\beta = 0.01$) seem to provide the best choice in terms of reconstruction of the patterns, generative accuracy and convexity of the misfit function in latent space. Cases with ($\alpha = 1.0, \beta = 0.001$) and ($\alpha = 0.1, \beta = 0.001$) also have good performance but show two minor defects: (1) a bit higher number of generated samples over the "holes" (orange dots in x-axes) which would translate into higher number of inaccurate patterns, and (2) a higher number of encoded samples (red dots in z-axes) in low-density regions which means the misplaced training patterns will be harder to generate.

In summary, a generator $\mathbf{g}(\mathbf{z})$ that preserves topology and contains nonlinearity is the best choice for gradient-based inversion in the latent space because it preserves convexity of the objective function. Note, however, that if the topology of the probability distribution $p(\mathbf{z})$ is different to the one of the real manifold $\mathcal{M}$, this strategy may result in approximate manifolds $\mathcal{M}'$ that do not account for all topological differences—e.g. that partially cover holes of the real one (see e.g. Fig. 3.1b)—and therefore might produce models that have non-accurate patterns when sampling from $p(\mathbf{z})$. We argue that the two training parameters α and β of a VAE may be chosen in order for the latter issue to not be severe, i.e. the generated patterns do not deviate too much from the training patterns, while still approximately preserving convexity of the objective function in the latent space.

To test our proposed method we implement a VAE in PyTorch (Paszke et al., 2017) and use training samples cropped from a "training image" which is large enough to have many repetitions of the patterns at the cropping size—a require-

ment similar in MPS. For our synthetic case, we use the training image of 2500 $\times$ 2500 pixels from Laloy et al. (2018) and the cropping size is chosen to fit the setting of our synthetic experiment (explained in detail in Sec. 3.2.5). Fig. 3.5a shows a patch of the training image and the position of the three (cropped) training samples shown Fig. 3.5b. Three generated samples from our proposed VAE trained with such croppings are shown Fig. 3.5c. Notice that the output of the generator is continuous (to allow for computation of gradients for training and inversion) with values between 0 and 1, and is later transformed to velocity values by a linear relation. For comparison, Fig. 3.5d shows three samples generated with the SGAN proposed by Laloy et al. (2019). Patterns of generated samples in Fig. 3.5c are not completely accurate comparing to those of the training image or the SGAN—they might display e.g. some breaking channels and smoothed edges (notice their output is also continuous but looks almost categorical). As mentioned above, this is expected for our proposed VAE because the approximate manifold fills some holes of the real manifold and may have less curvature. Also, the average proportion of channels from models generated from the VAE is a bit higher (0.36) than that of the training image (0.27). However, we argue that such inaccuracies may not cause significant error while performing inversion in practice because an informative dataset will generally make the inversion land in appropriate models (given the prescribed patterns were selected correctly). More importantly, in contrast to the SGAN, a modified gradient-based inversion (such as that presented in Sec. 3.2.4) will generally find a consistent minimum when applied with our proposed VAE regardless of the initial model.

3.2.4 Stochastic gradient descent with decreasing step size

Note that even when topology is preserved and nonlinearity is contained, the data misfit function in the latent space might still present some local minima. Using our proposed VAE approach in the synthetic case study, the resulting misfit function seems to have the shape of a global basin of attraction with some local minima of less amplitude. To deal with such remaining local minima we propose to use a SGD method instead of regular gradient-based optimization.

SGD methods are commonly used in training machine learning models to

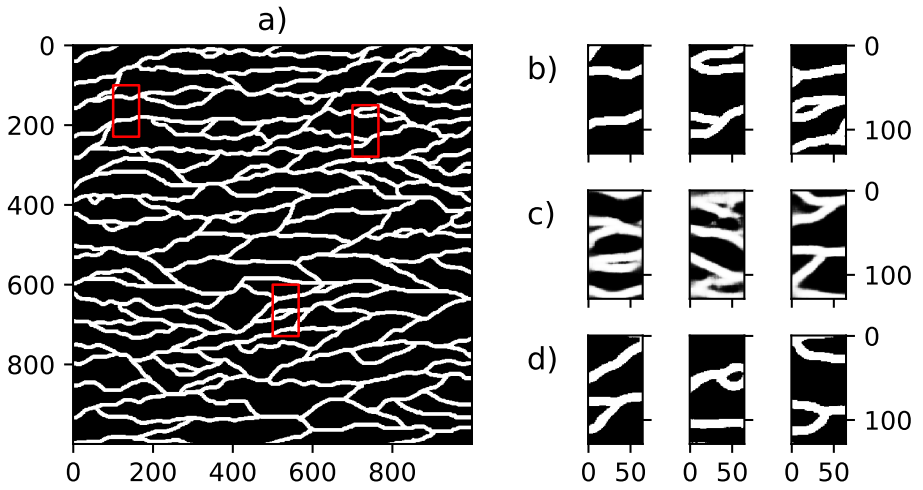


Figure 3.5: (a) A 1000 $\times$ 1000 patch of the training image of Laloy et al. (2018), (b) cropped training samples whose location in (a) is shown red, (c) generated samples from our proposed VAE, and (d) generated samples from the SGAN proposed by Laloy et al. (2018).

cope with large datasets (e.g. Kingma and Ba, 2017) and it has also been shown they are able to find minima that are useful in terms of generalization (Smith and Le, 2018). They essentially use an estimator for the gradient of the objective function computed only with a batch of the data. Such estimator is used in each gradient descent iteration and may be written for the case of inversion in the latent space as:

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \ell \cdot \nabla_{\mathbf{z}} \zeta(\mathbf{z})_k \tag{3.17}$$

where k denotes the iteration index, ℓ is the step size (or learning rate) and the gradient estimator $\nabla_{\mathbf{z}} \zeta(\mathbf{z})_k$ is computed by using Eq. (3.6) for a data batch (i.e. a subset of $\mathbf{d}$) which is different for each k-th iteration but of constant size b. Relying on such estimator makes SGD methods less likely to get trapped in local minima when the objective function has the shape of a global basin of attraction mentioned above (Kleinberg et al., 2018).

Recently, it has been proposed that using SGD may be seen as optimizing a smoothed version of the objective function obtained by convolving it with the

gradient "noise" resulting from batching (Kleinberg et al., 2018). The degree of noise (and therefore the degree of smoothness) is controlled by the ratio of the learning rate to the batch size ℓ/b (Chaudhari and Soatto, 2018; Smith and Le, 2018). Therefore if we choose to decrease the value of ℓ (while keeping b constant) as the optimization progresses we might be able to achieve lower misfit values i.e. get sufficiently close to the global minimum. This may be implemented by using:

$$\ell_{k+1} = c_\ell \cdot \ell_k \tag{3.18}$$

where a constant value of $c_\ell < 1.0$ and a starting value ℓ_0 must be chosen. In practice, the method may be further improved by also decreasing the controlling (regularization) term in Eqs. (3.4) and (3.6) in order to prevent that large initial steps diverge from the region of the latent space where the manifold is defined (Bora et al., 2017; Luo et al., 2015). Then, similarly to ℓ this may be done as:

$$\lambda_{k+1} = c_\lambda \cdot \lambda_k \tag{3.19}$$

again a constant $c_\lambda < 1.0$ and a starting value λ_0 must be selected.

The combined effect of simultaneously decreasing ℓ and λ is illustrated in Fig. (3.6) for a simple synthetic problem in a two-dimensional ($n = 2$) latent space $\mathcal{R}^n$. The misfit term (i.e. first term of Eq. (3.4)) of the synthetic problem is shown in Fig. 3.6a. Assuming that $p(\mathbf{z})$ is a normal distribution $\mathcal{N}(\mathbf{0}, I_n)$, we propose a specific regularization term $R(\mathbf{z})$ that will preferentially stay in the regions of higher mass (where most samples are located). This is done by radially constraining the search space by means of a χ-distribution, i.e. the regularization term is written as:

$$R(\mathbf{z}) = (\|\mathbf{z}\| - \mu_\chi)^2 \tag{3.20}$$

where μ_χ is the mean for a χ-distribution with n degrees of freedom. We refer to this strategy as "ring" regularization since for a two-dimensional latent space it enforces inversion to preferentially stay within a region with the shape of a ring. Dashed lines in Fig. 3.6a denote this mean together with the 16- and 84-th

percentiles. In general, this is especially useful for higher dimensionalities where most of the mass of a normal distribution is far from its center (Domingos, 2012). Then, Eq. (3.4) may be rewritten as:

$$\zeta(\mathbf{z}) = \|\mathbf{f}(\mathbf{g}(\mathbf{z})) - \mathbf{d}\|^2 + \lambda(\|\mathbf{z}\| - \mu_\chi)^2 \tag{3.21}$$

and correspondingly Eq. (3.6) may be expressed as:

$$\nabla_\mathbf{z}\zeta(\mathbf{z}) = \mathbf{J}(\mathbf{z})^T\nabla_\mathbf{m}\gamma(\mathbf{m}) + 2\lambda\mathbf{z}\left(1 - \frac{\mu_\chi}{\|\mathbf{z}\|}\right) \tag{3.22}$$

As mentioned above, this gradient is often computed simply by adding a layer to the autodifferentiation of the generator. One optimization instance for a random initial model is shown in Fig. 3.6b, while the behavior of the misfit and $\|\mathbf{z}\|$ is shown in Fig. 3.6c,d. Notice the rather "noisy" inversion trajectory, but also its ability to escape local minima. The effect of decreasing ℓ is seen in Fig. 3.6c by the decreasing of the oscillations amplitude as the optimization progresses, while the effect of decreasing λ is noticeable in Fig. 3.6d by the progressive shifting of $\|\mathbf{z}\|$ away from μ_χ.

The strategy described above and stated by Eq. (3.21) is generally applicable to DGMs that use an independent normal distribution as its probability distribution $p(\mathbf{z})$ and whose generator is well-behaved. In this chapter, we consider a VAE whose training parameters β and $p(\boldsymbol{\epsilon})$ are chosen so that it results in a mildly nonlinear inversion for which such SGD strategy is generally useful.

3.2.5 Inverse problem: traveltime tomography

To test our proposed method and compare it with a previous instance of inversion with a DGM, we consider an identical setting to that used in Laloy et al. (2019). Such setting considers a dataset of crosshole ground penetrating radar (GPR) traveltime tomography. To obtain a subsurface model $\mathbf{m} \in \mathbb{R}^N$ this method relies in contrasts of electromagnetic wave velocity which is related to moisture content and therefore to porosity for saturated media. The tomographic array considers a transmitter antenna in one borehole and a receiver antenna in the other, each of which is moved to different positions and a vector of measurements $\mathbf{d} \in \mathbb{R}^Q$ is

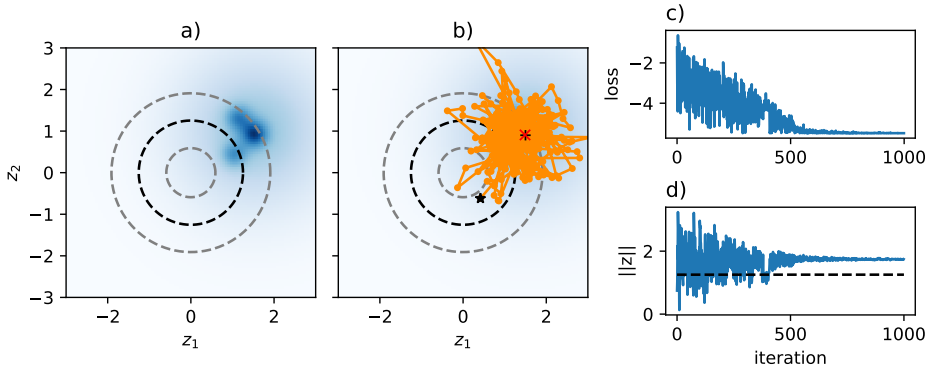


Figure 3.6: Regularized gradient-based inversion in a synthetic two-dimensional latent space: (a) misfit (blue) and mean of χ-distribution (black dashed) together with 16- and 84-th percentiles (gray dashed), (b) the same setting of (a) with an overlay of an instance of optimization (trajectory in orange) for a random initial model (black '$\star$'), showing also final model (red '$\times$') and true model (black '$+$'), (c) misfit vs. iteration number, and (d) norm of $\mathbf{z}$ vs. iteration number. Dashed line in (d) corresponds to the norm of the radius defined by the mean of the χ-distribution.

obtained by taking the traveltime of the wave's first arrival for each transmitter-receiver combination. We assume that the sensed physical domain is a 6.5×12.9 m plane (i.e. the two-dimensional region between the boreholes) and is discretized in 0.1×0.1 m cells of constant velocity to represent spatial heterogeneity (i.e. a representation of $N = 65 \times 129 = 8385$ cells is obtained). We consider a binary subsurface (e.g. composed of two materials with different porosity) with respective wave velocities of 0.06 and 0.08 m ns^{-1}. Measurements are taken every 0.5 m in depth (the first being at 0.5 m and the last at 12.5 m) resulting in a dataset of $Q = 625$ traveltimes. Note that though this model provides a good learning tool and a rather challenging test case, it is unrealistic, e.g. subsurface environments usually contain many more materials and further variability within each them. For one instance of our synthetic case, we add normal independent noise $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \cdot I_Q)$ where σ^2 is the noise variance and I_Q is a 625×625 identity matrix. In the case a different noise distribution is used, one needs to add a weight matrix to the misfit term in Eq. (3.4) so that inversion takes such distribution into account, e.g. when different data points have different magnitudes for the noise,

inversion should put more weight on those that are less affected by noise.

Similarly to Laloy et al. (2019), we first consider a fully linear forward operator $\mathbf{f}$ for which raypaths are always straight, i.e. independent of the velocity spatial distribution. For this case Eq. (1.1) may be rewritten as:

$$\mathbf{d} = \mathbf{F}\mathbf{m} + \boldsymbol{\eta} \tag{3.23}$$

where $\mathbf{F}$ is a matrix of dimension $Q \times N$ in which a certain row contains the length of the raypath in each cell of the model for a certain transmitter-receiver combination. The corresponding gradient of the misfit $\nabla_{\mathbf{m}}\gamma(\mathbf{m})$ to be used in Eq. (3.22) for the solution of the inversion is:

$$\nabla_{\mathbf{m}}\gamma(\mathbf{m}) = -2\mathbf{F}^T(\mathbf{d} - \mathbf{F}\mathbf{m}) \tag{3.24}$$

We also consider the case of a more physically realistic nonlinear forward operator $\mathbf{f}$ (see Eq. (1.1)) for which raypaths are not straight. In particular, we consider a shortest path (graph) method which uses secondary nodes to improve the accuracy of the simulated traveltimes as proposed by Giroux and Larouche (2013) and implemented in PyGIMLi (Rücker et al., 2017). For this case, when inversion with Eq. (3.22) is pursued, we linearize the forward operator $\mathbf{f}$ in order to compute the gradient:

$$\nabla_{\mathbf{m}}\gamma(\mathbf{m}) = -\mathbf{S}(\mathbf{m})^T(\mathbf{d} - \mathbf{f}(\mathbf{m})) \tag{3.25}$$

where $\mathbf{S}(\mathbf{m})$ is the $Q \times N$ Jacobian matrix of the forward operator whose elements are:

$$[\mathbf{S}(\mathbf{m})]_{i,j} = \frac{\partial f_i(\mathbf{m})}{\partial m_j} \tag{3.26}$$

The elements of the Jacobian $\mathbf{S}(\mathbf{m})$ are computed by the shortest path method and also represent lengths of raypaths. In contrast to the linear case, these have to be recomputed in every iteration. Both the nonlinear forward operator and the need for recomputing the Jacobian result in higher computational cost compared to the linear operator.

The method proposed in Sec. 3.2.4 to perform gradient-based inversion with a VAE should work for the linear forward operator because the nonlinearity in the inverse problem arises only due to the nonlinearity of the generator $\mathbf{g}(\mathbf{z})$ which is moderate when the latter is well-behaved. However, since the considered nonlinear forward operator in Eq. (3.25) is only mildly nonlinear (when contrast in velocities is not extreme), the same method may also provide good inversion results for this operator.

3.3 Results

3.3.1 Training of VAE

As previously mentioned, our proposed method relies on a VAE whose training parameters are selected in order to improve gradient-based inversion. The training samples are the croppings detailed in Sec. 3.2.3 whose dimensionality is $N = 8325$ and we consider a latent code of dimensionality $n = 20$. Different values for n were tested and $n = 20$ was chosen because higher values did not significantly improve the reconstruction of the training samples but did have a negative impact on the accuracy of the generated patterns (for this, generated patterns were assessed visually from a set of generated models such as those shown in Fig. 3.5c). Moreover, since the value of n also impacts the diversity of the generated patterns (i.e. how much they depart from training patterns), $n = 20$ provided a trade-off where patterns display sufficient diversity but still resemble those of the training image. The probability distribution $p(\mathbf{z})$ is an independent multinormal distribution $\mathcal{N}(\mathbf{0}, I_n)$ with I_n an identity matrix of size 20×20. The architecture of the encoder and the decoder includes 4 convolutional layers, 2 fully-connected layers and instance normalization is used between each layer. The VAE has around 4.5 million parameters in total (weights and biases), which is a typical number for convolutional neural networks (further details may be consulted in the associated code). In order to show their impact on our proposed method, α and β are set to span three orders of magnitude. Table 3.1 shows the values of α and β that were used and their impact in the data RMSE of the linear case explained below. This means that nine different VAEs are trained for this test. Each VAE is trained

	$\beta = 10^4$	$\beta = 10^3$	$\beta = 10^2$
$\alpha = 1.0$	2.551	1.765	2.940
$\alpha = 0.1$	3.116	1.763	3.756
$\alpha = 0.01$	2.747	1.937	2.701

Table 3.1: Sum of average data RMSE for the cases mc_1, mc_2 and mc_3. The average is computed for 100 initial models for each case.

by maximizing the lower bound in Eq. (3.16) using 10^5 iterations and batches of 100 random croppings in each iteration (a GeForce RTX 2060 GPU was used in which training took $\sim$2 hours). In the following, we first test the impact of α and β on inversion with a linear forward model. Then, we select the VAE with the best training parameters to study the impact of the different factors added in our approach (such as regularization and data batching) and make a comparison with methods from previous studies. Finally, we present some results of our approach using a mildly nonlinear forward operator.

3.3.2 Case with a linear forward model

In this section, we consider the linear operator in Eq. (3.23) and assess the performance of our proposed DGM inversion approach: using VAEs trained as detailed above and SGD with both decreasing step size and regularization to optimize Eq. (3.21). We aim to show that, when appropriate values of α and β are chosen, this approach is robust regarding its convergence to the global minimum and therefore assess its performance by using 100 different initial models. To test this, we considered three different true models (with different degrees of complexity) that were cropped from the training image and not considered during the VAE's training (models mc_1, mc_2 and mc_3 in first row of Fig. 3.7). Table 3.1 shows the inversion data RMSE obtained for all combinations of α and β that were tested for this linear forward operator (the average data RMSE values are simply summed for the three true models). These results are consistent with our explanation in Fig. 3.4, since both α and β have a noticeable impact on inversion performance. It is interesting to note that the values yielding the lowest data RMSE (α=0.1 and β=1000) are not those typically used in previous studies (α=1.0 and $\beta < 100$). Also, the impact of α seems to be lower compared to β.

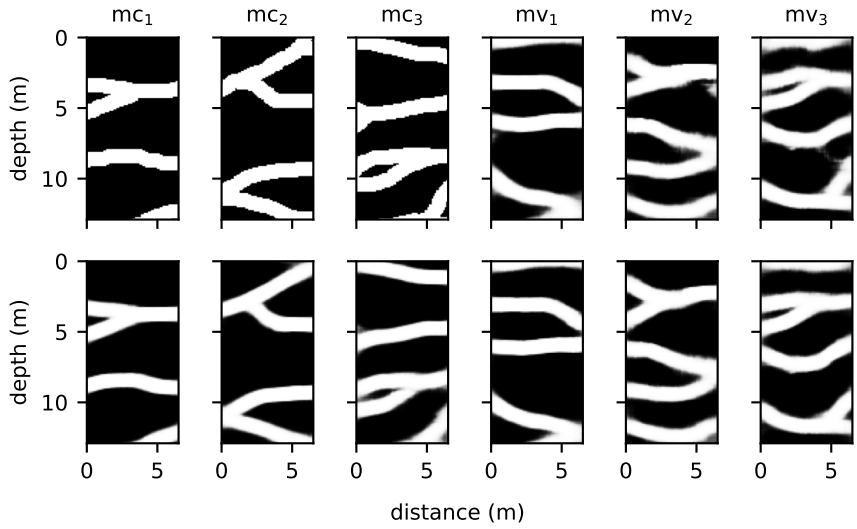


Figure 3.7: Truth models (first row): cropped from training image (denoted by "mc") and generated from trained VAE (denoted by "mv"). Corresponding models resulting from encode-decode of truth models (second row). Subindex indicates level of complexity, with "1" being the least complex.

To further assess our approach and to compare with previous studies, only the VAE with the values yielding the lowest data RMSE is considered in the remainder of this section. The main differences of our proposed approach with the method of Laloy et al. (2019) are in the type of DGM and the optimization strategy. We make a comparison with their method and also to other base cases listed in Table 3.2 to show the impact of each factor involved in our approach. As denoted by the columns of this table, the different cases consider: (1) VAE and SGAN as DGMs, (2) SGD and Adam (Kingma and Ba, 2017) as stochastic optimizers, (3) data batching for computing the gradient $\nabla_{\mathbf{z}}\zeta(\mathbf{z})$, which basically means using SGD when batching and using (regular) gradient-descent when not batching, (4) regularization in the latent space, with "origin" being the one proposed in Bora et al. (2017) and "ring" the one proposed herein, and (5) decreasing of the step size (or learning rate). Our proposed approach is then labeled as "VSbrd". We also show the chosen values for the step size ℓ and its decreasing factor c_ℓ when applicable—for these cases the values of λ = 10.0 and c_λ = 0.999 are used. When data batching is used, the batch size b is 25 (of a total of 625) and is sampled with no replacement, then the whole dataset is used every 25 iterations (i.e. with 120 epochs, the total is 120 $\times$ 25 = 3000 iterations). The number of iterations for the cases with no data batching is also set to 3000. For our synthetic cases, once the DGMs are trained, there is no need for GPU acceleration to perform inversion, so all inversions were done in CPU. Compared to MCMC methods used in previous studies where the number of forward model evaluations was between 96,000 and 200,000 (Laloy et al., 2017, 2018) the computational cost is herein significantly reduced. Note that we also compare against the approach in Laloy et al. (2019), where SGAN is used as DGM and Adam (gradient-descent with adaptive moments) are used to optimize the resulting objective function—this case is labeled "SAnnn" in Table 3.2. The difference in computational time for this case (17.3 s) and our proposed method (7.3 s) was minor. We also consider the case where we apply our proposed SGD to the same SGAN (labeled as "SSbnd"). For both of these cases instead of regularization we use stochastic clipping in the latent space (Laloy et al. 2018, 2019) because a uniform $p(\mathbf{z})$ with finite support is used.

We consider 6 different true subsurface models to assess our method and com-

Case	DGM	GD	Batching	Reg.	Dec.	ℓ	c_ℓ
VSnnn	VAE	SGD	no	none	no	1e-4	-
VSbnn	VAE	SGD	yes	none	no	1e-4	-
VSbod	VAE	SGD	yes	origin	yes	1e-2	0.95
VSbrd	VAE	SGD	yes	ring	yes	1e-2	0.95
SAnnn*	SGAN	Adam	no	none	no	1e-2	-
SSbnd	SGAN	SGD	yes	none	yes	1e-3	0.95

Table 3.2: Configuration of our proposed approach (VSbrd) and the base cases for comparison. The case marked with * corresponds to the one considered by (Laloy et al., 2019). "Reg." stands for regularization and "Dec." for decreasing.

pare with the base cases: (1) the set of three models cropped directly from the training image described above and (2) a set of three models obtained by generating from the trained VAE. Both sets include models with three different degrees of complexity. These truth models are shown in the first row of Fig. 3.7 where "mc" refers to the first set, "mv" refers to the second set and the degree of complexity is denoted by a subscript, where "1" denotes least complex and "3" most complex. The second set (mv) is similar to the one used by Laloy et al. (2019) to test the performance of their setup, only in their case the models were generated from a SGAN instead of a VAE. For each one of these truths, we generate synthetic data applying the forward operator $\mathbf{F}$ and use these data to perform gradient-based inversion for each case in Table 3.2.

We first consider no added noise to the synthetic dataset, hence after inversion the data misfit should be close to zero for inverted models that are sufficiently close to the global minimum. To define a threshold for this data misfit beyond which inverted models are "accepted", we use the RMSE between these synthetic data and data obtained by applying the forward operator on models resulting from passing the truth models through a VAE's encoding-decoding (these models are shown in the second row of Fig. 3.7 and the corresponding values for the threshold are shown in Table 3.3). This is done because we found the encode-decode reconstructed models to be visually very similar to the truth models (compare first and second rows of Fig. 3.7) and also show a low model RMSE when compared to them. The model RMSE is computed as the difference of pixel values (previous to transforming to velocity values, so they have values between

	data RMSE (ns)	model RMSE (-)
mc_1	0.724	0.112
mc_2	0.854	0.133
mc_3	1.395	0.176
mv_1	0.749	0.097
mv_2	1.380	0.146
mv_3	1.436	0.145

Table 3.3: Data RMSE (ns) of encode-decode operation used to define thresholds (for the linear forward operator) and corresponding model RMSE.

	VSnnn	VSbnn	VSbod	VSbrd	SAnnn	SSbnd	VSbrd (noise)
mc_1	91	33	100	100	0	0	100
mc_2	86	59	100	100	0	0	100
mc_3	91	35	100	100	0	0	100
mv_1	91	30	100	100	0	0	100
mv_2	95	71	100	100	0	0	100
mv_3	98	77	100	100	0	0	100

Table 3.4: Number of accepted inversions (using 100 different initial models) according to the defined threshold.

0 and 1) between truth model and the encode-decode model and shown Table 3.3. Once such a threshold is defined for each truth model, gradient-based inversion is run for the same 100 initial models for all cases in Table 3.2. Note that no convergence criteria were set in order to compare to all base cases (some cases such as "SAnnn" do not allow for easily defining such criteria) but in practice it is possible to set them for our proposed approach (VSbrd) in terms of a minimal change in either step size and/or data misfit. This also means that for some cases (including our proposed VSbrd) the 3000 iterations may not be necessary for all truths and all initial models. Results for the number of accepted inverted models are shown in Table 3.4 while the corresponding mean of the misfit (expressed as RMSE) for the 100 inversions is shown in Table 3.5.

As seen in Table 3.4, given our defined threshold: (1) the cases where VAE and SGD with decreasing step were used (VSbod and VSbrd) resulted in all inverted models being accepted, (2) the cases where SGAN was used (SAnnn and SSbnd) resulted in all models being rejected, and (3) the cases where VAE and

	VSnnn	VSbnn	VSbod	VSbrd	SAnnn	SSbnd	threshold	VSbrd (noise)
mc_1	0.536	1.169	0.551	0.434	4.538	3.988	0.724	0.501
mc_2	0.832	1.518	0.626	0.541	5.266	4.495	0.854	0.583
mc_3	0.908	1.543	0.853	0.788	3.298	3.775	1.395	0.827
mv_1	0.296	1.418	0.353	0.055	3.952	4.226	0.749	0.259
mv_2	0.568	1.286	0.618	0.078	4.161	5.251	1.380	0.268
mv_3	0.557	0.854	0.232	0.036	4.591	5.537	1.436	0.256

Table 3.5: Mean RMSE (ns) of inversions using 100 different initial models and defined threshold for accepting models.

non-decreasing step size SGD was used (VSnnn and VSbnn) resulted in some inverted models being accepted. Note also that using SGD (data batching) without a decreasing step size (VSbnn) results in less accepted models compared to GD (VSnnn), highlighting the importance of our proposed decreasing step size and regularization. As shown in Table 3.5 a higher mean RMSE is related to a lower number of accepted models. Furthermore, Table 3.5 shows that there is a general improvement caused by our proposed regularization compared to the one from Bora et al. (2017).

Examples of inverted models obtained for the different cases in Table 3.2 using the cropped truth with moderate complexity (mc_2) are shown in Fig. 3.8. Here, truth models are shown in Fig. 3.8a while Fig. 3.8b shows one example of an accepted model for cases that have at least one (VSnnn, VSbnn, VSbod and VSbrd). Similarly, Fig. 3.8c shows one example of a rejected model for applicable cases (VSnnn, VSbnn, SAnnn and SSbnd). Finally, the corresponding data RMSE vs. iteration number plots are shown in Fig. 3.8d (in blue for accepted models and red for rejected ones) and corresponding model RMSE plots are shown in Fig. 3.8e. Note both the higher similarity with the truth model (i.e. note the low model RMSE and compare models in Fig. 3.8b,c with those in Fig. 3.8a) and the lower RMSE for accepted models. Also, examples of inverted models for our proposed approach (VSbrd) using all the truths are shown in Fig. 3.9b, together with plots of RMSE vs. iteration number (Fig. 3.9d) and norm of $\mathbf{z}$ vs. iteration number (Fig. 3.9e). For cropped truths (mc) it seems that visual similarity decreases and final data RMSE of inverted models increases as complexity increases, whereas for generated truths they seem independent of complexity. Notice the overshoot

in $\|\mathbf{z}\|$ in the initial iterations and its eventual convergence close to μ_χ as defined in Eq. (3.20).

To study the effect of noise for our proposed approach (VSbrd), we added noise with a standard deviation σ = 0.25 ns to the synthetic traveltime data. Corresponding results are shown in the rightmost column of Tables 3.4 and 3.5 and in Fig. 3.9c (with corresponding data RMSE and $\mathbf{z}$ norm plots in Fig. 3.9d,e). The threshold in this case is set equal to the one for the noise-free case plus σ and when using it all inverted models with our proposed approach are accepted. It is also worth noticing the relative robustness of the method to noise, as shown by the corresponding mean misfit values in Table 3.5 that indicate no significant overfitting, i.e. the mean misfit values are close to the noise-free threshold plus σ even if no traditional regularization was used. The latter means that optimizing in the latent space of the DGM is effectively constraining the inverted models to display the prescribed patterns. A higher value of σ = 1.0 ns was also tested which produced similar results (not shown).

3.3.3 Case with a nonlinear forward model

After showing that our proposed method works with the linear forward operator for the synthetic case considered, we now test its performance with a nonlinear forward operator. For inversion, the general form of Eq. (1.1) is used and the gradient in the latent space given in Eq. (3.22) is computed using Eq. (3.25). As mentioned in Sec. 3.2.5, we consider a shortest path method to solve for the traveltime for which we use 3 secondary nodes added to the edges of the velocity grid. Note that the Jacobian $\mathbf{S}(\mathbf{m})$ in Eq. (3.25) has to be recomputed at every iteration. Given the higher computational demand for inversion with the nonlinear forward operator and since it was already shown to be the best performing approach for the linear forward operator, we only test our proposed approach VSbrd with all the truths and for a single initial model (Fig. 3.10). This was done both without noise and with noise added using the same standard deviation σ = 0.25 as in the linear operator scenario. We select the following values for the required inversion parameters: $\ell = 0.1$, $c_\ell = 0.8$, $\lambda = 1.0$ and $c_\lambda = 0.99$. The total number of iterations is 750 with data batching of size 25 similar to the linear case. Note that

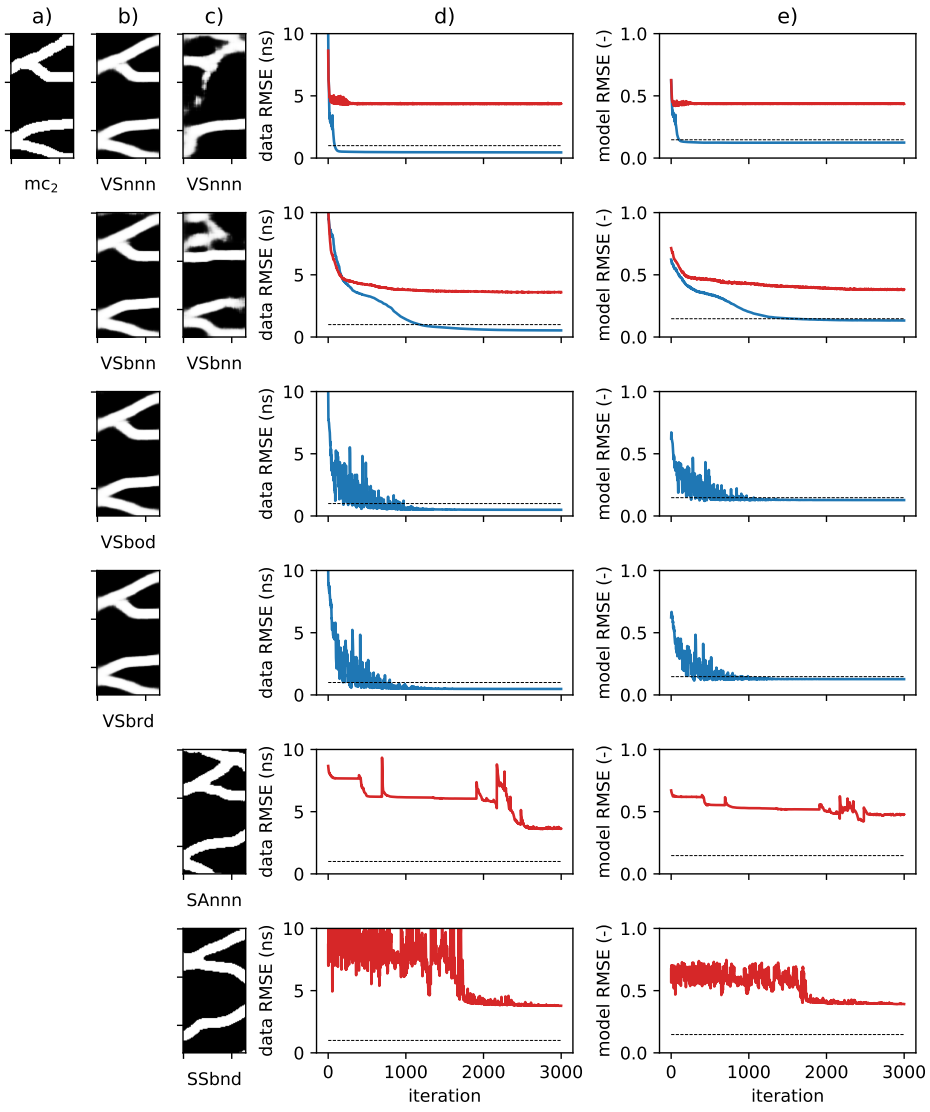


Figure 3.8: Examples of inverted models for mc_2 truth for all cases in Table 3.2: (a) truth model, (b) accepted models according to defined threshold, (c) rejected models, (d) data RMSE vs. iterations plots (blue for accepted models and red for rejected models and dashed line indicates defined threshold) and (e) model RMSE vs. iterations plots (dashed line indicates model RMSE for encode-decode operation).

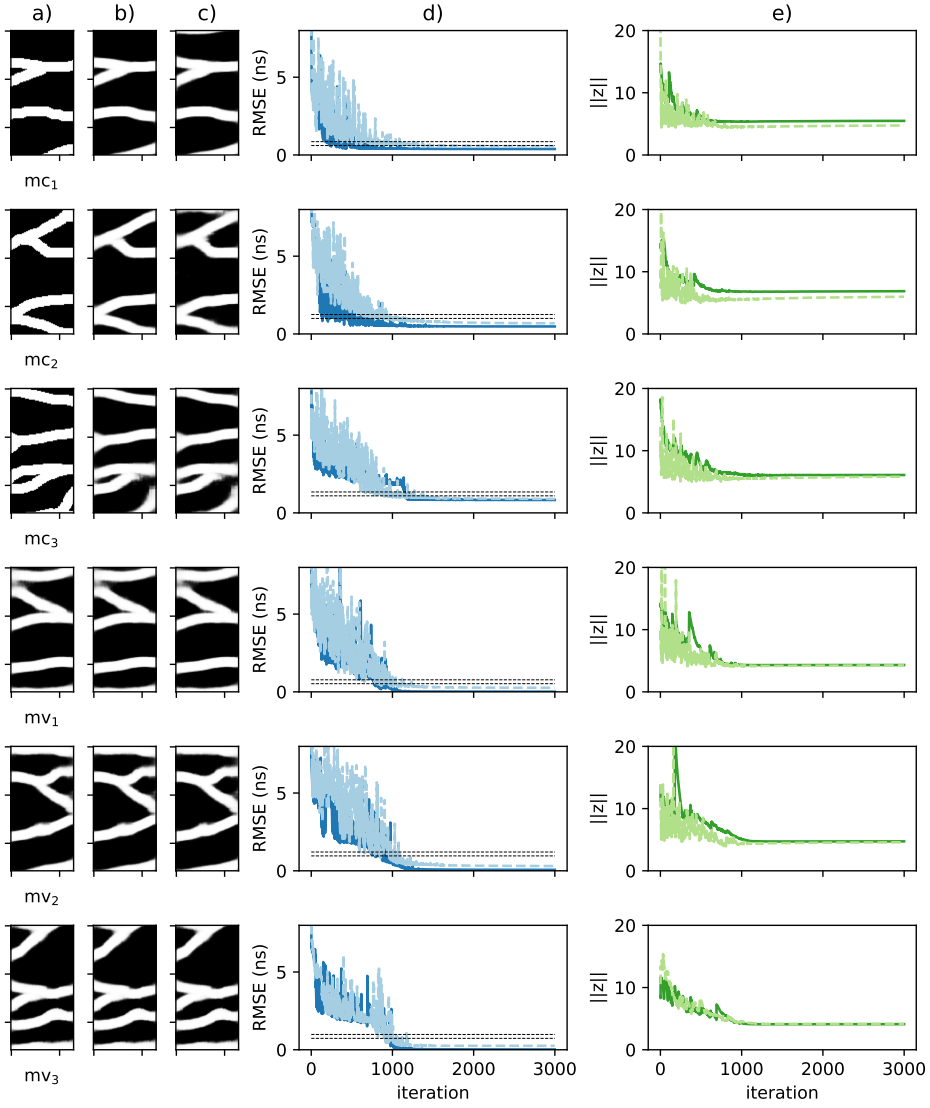


Figure 3.9: Examples of gradient-based inversion using our proposed approach (VSbrd) for all truth models and the linear forward operator: (a) truth models, (b) inverted models with no added noise, (c) inverted models with added noise, (d) RMSE vs. iterations plots (no noise case in dark blue and noise case in light blue; lower dashed line indicates the defined threshold while upper dashed line is threshold plus $\sigma = 0.25$), and (e) norm of $\mathbf{z}$ vs. iterations plots (no noise case in dark green and noise case in light green).

to further reduce the number of iterations required for inversion we use a lower c_ℓ compared to the linear case, but the decreasing in Eq. (3.18) is only done every 5 iterations. This may cause the method to converge to the global minimum with lower probability, however it seems to still be high enough since all of the inversions with no added noise are very similar to the truth models. Also, using the threshold obtained by encoding-decoding the truth models (now computed with the nonlinear forward operator) all inverted models are accepted (these models are shown in Fig. 3.10b). When considering added noise, results are similar but inversion seems to converge to the global minimum with slightly lower probability (6 out of 8 inversions are accepted) and accepted models are shown in in Fig. 3.10c. The behavior of the misfit during optimization (Fig. 3.10d) is similar to the linear case, although oscillations of a slightly higher amplitude are still visible in the last iterations (mainly due to the lower number of iterations). To partially solve the latter issue, we take as inverted model the model with lowest misfit and not the one for the final iteration (these are the models shown in Fig. 3.10b,c). The plot of the norm of $\mathbf{z}$ vs. iterations in Fig. 3.10e shows a similar behavior to the linear case, although there seems to be more oscillations in $\|\mathbf{z}\|$ during initial iterations.

3.4 Discussion

For both our toy example in Fig. 3.4 and our synthetic case for the linear forward operator (Table 3.1), results show that α and β have an impact on inversion. While in both cases the value yielding the lowest data RMSE for α is 0.1, β spans a larger range of values. This occurs mainly because α is coupled to the imposed unit variance of $p(\mathbf{z})$, since larger values of α tend to place samples further apart in the latent space and therefore make it inconsistent with $p(\mathbf{z})$. In contrast, due to the nature of the VAE training loss function in Eq. (3.16), β depends on the dimensionality of both the training samples (N) and the latent vectors (n). In order to have more comparable values between studies, Higgins et al. (2017) proposed normalizing β as $\beta' = \beta \times n/N$. In our synthetic case, the value of β=1000 yields β'=2.4 which is still high compared to what Higgins et al. (2017) found for an optimal disentangling (for n=20) or to what Laloy et al. (2017) used in their

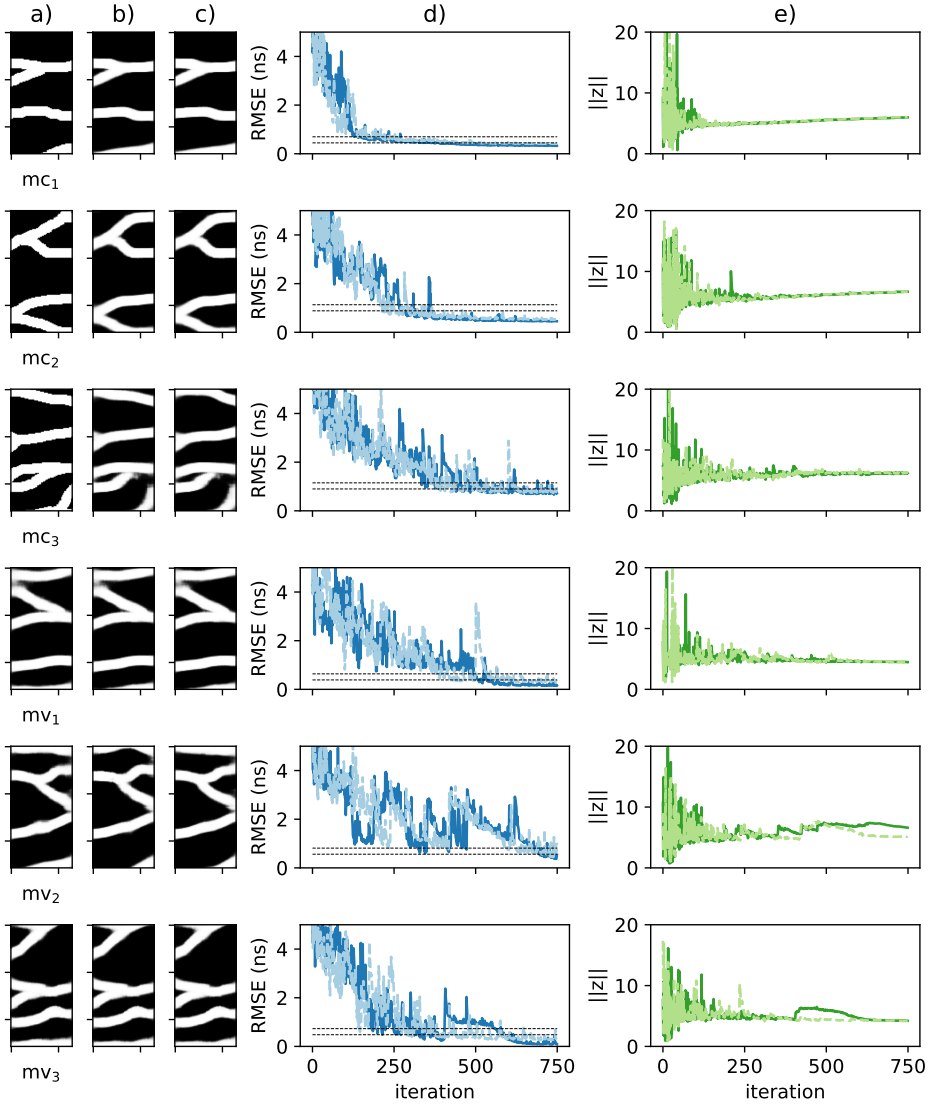


Figure 3.10: Examples of gradient-based inversion using our proposed approach (VSbrd) for all truth models and the nonlinear forward operator: (a) truth models, (b) inverted models with no added noise, (c) inverted models with added noise, (d) RMSE vs. iterations plots (no noise case in dark blue and noise case in light blue; lower dashed line indicates the defined threshold while upper dashed line is threshold plus $\sigma = 0.25$), and (c) norm of $\mathbf{z}$ vs. iterations plots (no noise case in dark green and noise case in light green).

study (both around β'=0.1). This may be related to the fact that these studies focused either on disentangling or on generative accuracy for selecting β, instead of inversion performance as done in here. Note that normalized values of β are the most appropriate to provide guidelines for future studies. Our results suggest that setting β'>1.0 may be useful for inversion, but further testing with different kinds of patterns is still required to support this.

In order to select SGD parameters in our proposed approach, we suggest looking jointly at the behavior of the misfit and norm of $\mathbf{z}$. For instance, if a certain number of iterations is desired for computational reasons, we suggest choosing first ℓ and c_ℓ that produce a behavior of the misfit similar to that in Fig. 3.9d, i.e. oscillations of high amplitude at the beginning and then progressive attenuation of the oscillations in such a way that at the end they are negligible. Note, however, that inversion may have to be run a few times because divergence may occur during initial iterations (this is easily seen in the value of $\|\mathbf{z}\|$ taking values far from μ_χ). Once ℓ and c_ℓ are chosen, the selection of λ and c_λ is done only to prevent divergence, this may be achieved by looking for a behavior similar to that in Fig. 3.9e. An initial overshoot in $\|\mathbf{z}\|$ is normal (and even necessary) since the method is exploring more rapidly the latent space, however, it should eventually converge to a value close to μ_χ.

The results for gradient-based inversion using our proposed approach point to a (possible) conflict between the accuracy of the reproduced patterns and the feasibility of gradient-based inversion with DGMs. As mentioned above, this is due to a non-convex objective function in latent space resulting from the generator's nonlinearity and its induced changes in topology. In this work, we argue that nonlinearity and changes in topology might be safely controlled by selecting certain values of α and β while training a VAE in order to improve performance of gradient-based inversion. We empirically show the validity of this statement by considering different values of α and β for our linear case study. In general (for inversion with DGMs), this implies that a tradeoff between generative accuracy and a well-behaved generator may be found. The latter statement also supports our assumption regarding the "holes" of the real manifold for the case of channel patterns (as mentioned in Section 3.2.3): when approximating the real manifold using a VAE with a well-behaved generator, the approximate manifold will tend

to fill the holes and therefore produce breaking channels. While the generator's nonlinearity was already identified by Laloy et al. (2019) as a potential factor for hindering gradient-based inversion, its causes (curvature and topology of the real manifold) and the possible induced changes in topology have not been previously explained as factors in degrading the performance of gradient-based inversion in the latent space (to the authors' knowledge).

In general, good performance of DNNs for some tasks is usually associated with their ability to change topology (Naitzat et al., 2020). However, when one wants to use the latent variables or codes of DGMs for further tasks and not just for generation, these changes in topology might become an issue. For instance, we interpret the misfit "jumps" seen in gradient-based inversion with SGAN (as seen in Fig. 3.8c for case SAnnn) as resulting from the "gluing" or "collapsing" in latent space of holes in the real manifold—either caused by an induced change in topology or a high nonlinearity in the SGAN generator. Some studies have even suggested that if one wants to obtain useful geometric interpretations in the latent space (e.g. to perform interpolation), the activation functions should be restricted to ones that are smooth (Shao et al., 2017; Arvanitidis et al., 2018), that means e.g. not using the ReLU activation function that is generally recognized to result in faster learning. In contrast, in this work we do consider ReLU activation functions but control the changes in topology by means of a combination of α and β, whether this might nullify the advantages of ReLU is still an open question. Note however that, in general, control of induced changes in topology and high nonlinearities (as in our proposed approach) might be useful for any inversion method that relies in the concept of a neighborhood (e.g. MCMC and ensemble smoothers).

Besides its good performance for gradient-based inversion, a further advantage of our approach when compared to the previous approaches is that when the data used for inversion is not sufficiently informative, regularization in the latent space might be used to constrain to the most common patterns with our regularization term in Eq. (3.20). This statement provides an interesting paradigm where regularization in latent space might be seen as a flexible way to incorporate complex regularization. In contrast, a disadvantage of our proposed approach is that GANs in general result in higher generative accuracy (all generated pat-

terns look more similar to those in the training image). However, as previously mentioned this may negatively affect inversion performance, at least for gradient-based methods. Also, as may be noticed in the relation between the data misfit and the degree of complexity for cropped truths, a limiting factor in using our VAE is its inability to produce new highly complex patterns. Nevertheless, this lack of innovation (or sample diversity) is generally present in other methods and may be even more severe for regular GANs, where the phenomenon is known as mode collapse. Recently, different ways to control such mode collapse in VAEs and GANs have been proposed (Metz et al., 2017; Salimans et al., 2016).

Regarding the SGD optimization method proposed, we must note that similar results might be obtained with a MCMC method where information about the gradient is taken into account. For example, Mosser et al. (2018) use a Metropolis-adjusted Langevin method which basically follows a gradient-descent and adds some noise to the step. However, the noise added to the gradient step in our approach is different—SGD noise has been shown to be approximately constant but anisotropic (Chaudhari and Soatto, 2018). Another possible alternative to our method is to use Riemannian optimization, which is possible when the DGM approximate manifold is smooth. Although it is possible to compute the direction of the gradient by using the pullback Riemannian metric, which may be obtained as suggested by e.g. Shao et al. (2017); Chen et al. (2018); Arvanitidis et al. (2018), it is not straightforward to compute the step because it would have to be along a geodesic curve instead of a straight path and such geodesics are computationally demanding to obtain.

Finally, we acknowledge that in order to be applied for a variety of field conditions, our proposed method needs to be extended to: (1) handle multiple materials (i.e. not a binary subsurface), (2) consider further variability inside each material, (3) estimate the velocity values directly (i.e. not assume they are known as was done above), (4) consider larger domains, and (5) condition to observed values of materials (e.g. in wells). To address the first two points and since DGMs are not restricted to categorical outputs, the VAE could simply be trained using samples with continuous outputs, however, the accuracy of the patterns may not be as good as in the binary case for a training image of the same size (Laloy et al., 2018). If one chooses to approximate the subsurface with a multi-categorical

output, a different and more consistent loss function for the training such as the cross-entropy loss may give better results. Regarding the estimation of velocity values, a simple way to achieve this for binary models would be to include two extra parameters in inversion by assuming a linear relationship to shift and scale the output of the VAE. A similar approach may be used for multi-categorical or continuous outputs, although its usefulness may be more limited since the scaling and shifting operations do not significantly change the contrasts between the materials from those of the DGM outputs. When the spatial domain being studied is large or, more specifically, when it has many repetitions of the patterns, our method would require a very large training image which is generally difficult to obtain. A possible solution for this is to use an architecture that is more efficient for repetitive patterns. For instance, one may propose a spatial VAE (similar to the spatial GAN) which relies on 2D or 3D tensors instead of vectors as latent variables. Of course one then would need to test that efficient inversion (e.g. gradient-based) is still possible with such an architecture. Finally, conditioning to direct material observations may be achieved by adding a term to the inversion objective function in Eq. (3.4), although it has been shown that this does not produce perfect fitting to such observations (Laloy et al., 2017, 2018) so further study in this topic is required.

3.5 Conclusions

In this chapter both the impact and the causes of nonlinearity on inversion with DGMs are studied and a conflict between generated pattern accuracy and feasibility of gradient-based inversion is identified. Also, an approach based on a VAE as DGM and a modified stochastic gradient descent method for optimization is proposed to address such conflict. We show that two training parameters of the VAE (the weight factor β and the variance α of the encoder's noise distribution $p(\boldsymbol{\epsilon})$) may be chosen in order to obtain a well-behaved generator $\mathbf{g}(\mathbf{z})$, i.e. one that is mildly nonlinear and approximately preserves topology when mapping from latent space to ambient space. This helps in maintaining the convexity of the misfit function in the latent space and therefore improves the behavior of gradient-based inversion. We highlight changes in topology which have not been

previously identified as impacting the convexity of the inversion objective function. In contrast to prior studies where gradient-based inversion was used, our approach converges to the neighborhood of the global minimum with very high probability for both a linear forward operator and a mildly nonlinear forward operator with and without noise. We argue that when using DGMs in inversion, a tradeoff may be found where inverted models are close enough to the prescribed patterns while low cost gradient-based inversion is still applicable. Indeed, our proposed approach finds such tradeoff and produces inverted models with significant similarity to the training patterns and a sufficiently low data misfit.

Chapter 4

Reducing data dimension for prior falsification: feature extraction from data acquired in a highly structured subsurface[1]

Spatial heterogeneity is a critical issue in the management of water resources. However, most studies do not consider uncertainty at different levels in the conceptualization of the subsurface patterns, for example using one single geological scenario to generate an ensemble of realizations. In this paper, we represent the spatial uncertainty by the use of hierarchical models in which higher-level parameters control the structure. Reduction of uncertainty in such higher-level structural parameters with observation data may be done by updating the complete hierarchical model, but this is, in general, computationally challenging. To address this, methods have been proposed that directly update these structural parameters by means of extracting lower dimensional representations of data called data features that are informative and applying a statistical estimation technique

[1]**Note:** The research presented in this chapter is based on: Lopez-Alvis, J., Hermans, T., and Nguyen, F. (2019). A cross-validation framework to extract data features for reducing structural uncertainty in subsurface heterogeneity. *Advances in Water Resources*, 133, 103427. https://doi.org/10.1016/j.advwatres.2019.103427

using these features. The difficulty of such methods, however, lies in the choice and design of data features, i.e. their extraction function and their dimensionality, which have been shown to be case-dependent. Therefore, we propose a cross-validation framework to properly assess the robustness of each designed feature and make the choice of the best feature more objective. Such framework aids also in choosing the values for the parameters of the statistical estimation technique, such as the bandwidth for kernel density estimation. We demonstrate the approach on a synthetic case with cross-hole ground penetrating radar traveltime data and two higher-level structural parameters: discrete geological scenarios and the continuous preferential orientation of channels. With the best performing features selected according to the cross-validation score, we successfully reduce the uncertainty for these structural parameters in a computationally efficient way. While doing so, we also provide guidelines to design features accounting for the level of knowledge of the studied system.

4.1 Introduction

Modeling subsurface systems requires accounting for uncertainty in many tasks such as reserve estimation, process understanding, decision making or water resources forecasting (Scheidt et al., 2018). To consider explicitly different sources of uncertainty, probabilistic approaches are often used (Tarantola and Valette, 1982; Tarantola, 2005) and allow to easily integrate any type of data or prior knowledge. In the Earth sciences, spatial heterogeneity is of utmost importance but its uncertainty is often not properly represented leading to over-simplifications of subsurface systems (Xu and Valocchi, 2015) and biased predictions made from such systems (Hermans et al., 2018).

Hydrogeological modelling is often hierarchical (Feyen and Caers, 2006; Tsai and Elshall, 2013; Comunian et al., 2016), in the sense that, based on available data, hydrogeologists first speculate on the nature of the depositional system (e.g., marine, deltaic or fluvial) and on global characteristics of the deposits (orientation or size of the structures) leading to the definition of different scenarios that serve as the basis for further modeling. Within each scenario, more specific spatial uncertainty rules can be defined. Each geological scenario might be expressed

by its own training image or variogram model depicting the spatial uncertainty. Despite growing efforts made to model realistic prior geological information (see Linde et al., 2015, for a review), a single main structure is often considered which may underestimate the uncertainty or bias models if the structure is wrong (Linde et al., 2006). As an example, Hermans et al. (2015) demonstrated that the posterior distribution of hydrofacies constrained to electrical resistivity tomography and pumping data was dependent on the training image used and that ignoring the uncertainty on the depositional systems led to a biased solution. A possible strategy to avoid these problems is to consider hyperparameters—i.e., higher level parameters having their own prior probability distributions—leading to a so-called Bayesian hierarchical model (Gelman et al., 2014). Such hyperparameters may include the range of a variogram, the choice of training image or even the width of channels in a specific training image. These hierarchical problems have been addressed outside a Bayesian framework (see e.g. Khaninezhad and Jafarpour, 2014; Golmohammadi and Jafarpour, 2016, in the context of geological scenario identification), but in doing so, the uncertainty in the results is generally not quantified.

Within a Bayesian framework such an hierarchical model is then represented by a joint probability distribution involving hyperparameters, parameters and data, increasing the dimensionality of the joint space and making exploration more computationally demanding. Two different general approaches can be used to perform inference (i.e. updating uncertainty given some data) in such hierarchical models: (1) one-step methods where inference on the complete model (i.e., on both hyperparameters and parameters) is done in a single step, and (2) two-step methods where inference is done first for the hyperparameters and then the results are used to obtain the uncertainty on the parameters.

One-step approaches can be formulated by directly applying Markov chain Monte Carlo (MCMC) (e.g. Vrugt et al., 2009) to the complete hierarchical model. MCMC are sampling techniques that can cope with high-dimensional parameters. However, they must be modified to account for the hierarchical structure by changing the equations for the probability of acceptance of the proposal distribution (e.g. Malinverno, 2002) which may not be straightforward for all types of hyperparameters. Modifying one hyperparameter such as the training image,

for example, impacts the model in its whole, potentially leading to completely different likelihood, which is not desirable for convergence in MCMC. Only very recent advances have made possible the exploration of such complex joint spaces. In this regard, Arnold et al. (2019) and Demyanov et al. (2019) presented a framework based on the definition of a metric space for the geological scenario and a combination of global optimization and resampling, to approximate a thorough MCMC.

Two-step approaches are based on the factorization of the joint posterior distribution in the product of the posterior of the parameters given the hyperparameter and the (marginal) posterior of the hyperparameter, and perform inference separately for each factor (Neuman, 2003; Khodabakhshi and Jafarpour, 2013; Park et al., 2013). However, the factor corresponding to the (marginal) posterior of the hyperparameter involves a multidimensional integral which may be computationally demanding (Neuman, 2003). The focus of this Chapter is in the computation of the (marginal) posterior of the hyperparameter, i.e. only the first step in a two-step approach while solving the complete inverse problem for the hierarchical model.

Regarding computational demand, it has been argued that two-step may be more efficient than one-step approaches because of their ability to discard or falsify certain values with a relatively cheap method (that does not require inference of the parameters) in the first step (e.g. Park et al., 2013; Hermans et al., 2015). This may be specially advantageous when considering a high number of discrete values or a continuous range of the hyperparameter. However, it is also possible that one-step methods, when designed to be efficient (e.g. Demyanov et al., 2019), could quickly discard values of the hyperparameter that are not consistent with data. Park et al. (2013) make a comparison of their method, a two-step approach, with rejection sampling (which is used as a one-step method) and show that their method provides similar results with less computations of the forward model. However, rejection sampling is a very expensive method and, to the authors' knowledge, a comparison against more favorable one-step approaches has not been done yet. Such a comparison is, nevertheless, outside of the scope of the paper.

Different ways to handle the hyperparameter factor in a two-step approach

have been proposed, especially within the context of Bayesian model averaging (BMA) (Hoeting et al., 1999). When the hyperparameter is discrete, the factorization strategy mentioned above is equivalent to applying BMA for the parameters. In BMA, the aforementioned multidimensional integral is usually approximated by using a Gaussian distribution for the parameter dimensions in the likelihood. This Gaussian distribution is centered on the maximum-likelihood parameters and computed for each value of the hyperparameter (the so-called Laplace approximation), and it is a common approach when using BMA in hydrogeology (Neuman, 2003; Ye et al., 2004; Li and Tsai, 2009). Therefore, the Laplace approximation requires the classical inverse problem to be solved once for each value of the hyperparameter (and would require more involved sampling in the case of continuous hyperparameters). Moreover, to be a higher-order approximation, it requires the evaluation of the Hessian with respect to the parameters. Both the maximum-likelihood estimation and the Hessian may require a significant number of simulations using a computationally expensive numerical model. For this reason, some studies (Li and Tsai, 2009; Tsai and Elshall, 2013) have relied on the fact that, when the number of data becomes large relative to the number of parameters, the Laplace approximation can be simplified and computed using the Bayesian information criterion (BIC) (Raftery, 1995), which does not require evaluation of the Hessian. However, this may not be the case for most problems in Earth sciences, where parameters are usually high-dimensional and data is sparse. Khodabakhshi and Jafarpour (2013) used the same factorization within a sequential Monte Carlo approach, where the hyperparameter factor is first computed with a mixture model and then used for adaptive sampling in an Ensemble Kalman Filter to update the parameters. However, since their method is embedded in the sequential approach, the hyperparameter factor cannot be computed separately, i.e. the hyperparameter factor at the final time cannot be computed without updating the parameters at each time step.

Considering the same factorization of the joint distribution, an alternative method to obtain the (marginal) posterior of the hyperparameter was proposed by Park et al. (2013) that copes with the disadvantages of the Laplace approximation but also computes the hyperparameter factor separately. In other words, their method works for low numbers of data (where BIC does not apply), retains

a low computational demand and does not require previous inference on the parameters. Instead of aiming for a point-by-point match of data, a feature match would result in a similar (marginal) posterior distribution for the hyperparameter according to the authors. This is equivalent to approximating the data manifold described in Chapter 1. Therefore, feature extraction techniques are needed to reduce the dimensions of data low enough so that statistical techniques (e.g. kernel density estimation) may be applied to a low number of Monte Carlo samples to approximate the (marginal) joint distribution of the hyperparameter and the features. This distribution is then evaluated at the features of the observed data to obtain the (marginal) posterior of the hyperparameter. No maximum-likelihood estimation of the parameters for each value of the hyperparameter or Hessian evaluation is needed, as opposed to the Laplace approximation. Feature extraction techniques may incur in some computational time depending on their complexity, but this is generally negligible compared to evaluations of the numerical model. Park et al. (2013) presented an example where they generate Monte Carlo samples of the joint distribution by numerical simulations of reservoir flow data, then disregard parameter dimensions and apply data dimension reduction together with kernel density estimation to approximate the (marginal) posterior distribution of the geological scenario (which is the hyperparameter in their case). As mentioned above, they showed that the method yields results similar to rejection sampling. Hermans et al. (2015) applied it for one discrete hyperparameter but with two different types of data: hydraulic heads and electrical resistivity tomography. Scheidt et al. (2015b) extended the approach to estimate the posterior distribution of a continuous hyperparameter. Scheidt et al. (2015a) follow the same approach but deal with seismic data and a wavelet-based method to reduce dimensions of this data. A major difficulty of this approach is that choosing between the different ways to extract features is not straightforward, and an objective assessment of all the possible choices of features is lacking. Moreover, applying the techniques involves some additional specific parameters whose values are not straightforward to optimize.

In this paper, we define and systematically compare the efficiency of a new range of features for the application of the Park et al. (2013) framework with geophysical data. As part of the features definition, we propose a cross-validation

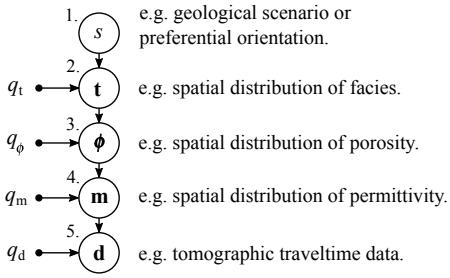


Figure 4.1: Graphical model for the proposed Bayesian hierarchical model. s stands for the structural parameter, $\mathbf{t}$ is the field of geological facies, $\boldsymbol{\phi}$ is the field of a physical property, $\mathbf{m}$ is the field of a geophysical property, $\mathbf{d}$ is the geophysical data. The q's are the fixed variables required at each step. On the right side, examples of each variable.

method to select the best feature and the parameters required by the framework that is based on performance scores of the newly designed data features.

We illustrate the proposed approach using near-surface geophysical data to derive posterior probability distributions of one discrete and one continuous hyperparameter.

4.2 Methodology

4.2.1 Hierarchical probabilistic model sampling

To deal with multi-level uncertainty problems typically present in Earth sciences we propose to build a Bayesian hierarchical model to explicitly consider the relations between all parameters and data. The probabilistic model considered in this study can be represented as the directed acyclic graph (DAG; Bishop, 2006) shown in Fig. 4.1, where each random variable is represented by an open node and relations of conditional dependency are represented by directed arrows.

In this graphical model, the hyperparameter s at the top controls the structure of spatial parameters in lower levels, therefore we will refer to it as structural parameter. Geophysical data or observations are in the lowest level of this

model. Indeed, as implied by the conditional relations of the graph and given its spatially-distributed nature, geophysical data provide a means to reduce uncertainty in spatial heterogeneity. Our objective is to compute the posterior distribution of structural parameters given the geophysical data $p(s|\mathbf{d})$, which can be obtained by considering the corresponding marginal distribution $p(s, \mathbf{d})$ of the joint probability distribution $p(s, \mathbf{t}, \boldsymbol{\phi}, \mathbf{m}, \mathbf{d})$. The DAG implies that the joint probability distribution can be factorized as

$$p(s, \mathbf{t}, \boldsymbol{\phi}, \mathbf{m}, \mathbf{d}) = p(s)p(\mathbf{t}|s)p(\boldsymbol{\phi}|\mathbf{t})p(\mathbf{m}|\boldsymbol{\phi})p(\mathbf{d}|\mathbf{m}) \tag{4.1}$$

where s stands for the structural parameter, $\mathbf{t}$ is the field of geological facies, $\boldsymbol{\phi}$ is the field of a physical property, $\mathbf{m}$ is the field of a geophysical property, $\mathbf{d}$ is the geophysical data and $p(\cdot|\cdot)$ expresses a conditional probability distribution.

To approximate the joint distribution from Eq. (4.1) we use Monte Carlo sampling. Since our probabilistic model is represented by a DAG we can obtain samples of the joint distribution by ancestral sampling, i.e., sampling following an order determined by the arrows in Fig. 4.1. Hence, when sampling a certain node, all nodes pointing to it (termed parent nodes) must be already sampled. Fixed variables that may be required in each sampling step are usually represented as black dots, e.g. the specified noise level in the data used in the last step is included in q_d (Fig. 4.1). Once the samples of the joint distribution are obtained we disregard parameters (or dimensions) other than the ones in the marginal of interest, $p(s, \mathbf{d})$. In our implementation, each step is given by (for numbering, refer to Fig. 4.1):

1. The structural parameter, s, is sampled from either a uniform distribution or a discrete uniform distribution.

2. The geological heterogeneity is represented by a spatially discretized facies field, $\mathbf{t}$. In our case, we consider this field as generated by a stochastic process, such that sampling from $p(\mathbf{t}|s)$ gives a categorical random field defined either by multiple-point statistics or truncated sequential gaussian simulation.

3. The physical property field $\boldsymbol{\phi}$ requires a probability distribution $p(\boldsymbol{\phi}|\mathbf{T} =$

$\mathbf{t}$). If no uncertainty is assumed at this step, then only a relation that assigns a value of the physical property to each facies is used.

4. The geophysical property field $\mathbf{m}$ is obtained by using a petrophysical relation which may also be formulated as a probability distribution $p(\mathbf{m}|\mathbf{\Phi} = \boldsymbol{\phi})$.

5. Finally, the geophysical data $\mathbf{d}$ is the result of a geophysical forward operator $g(\mathbf{m})$ and formulated as $p(\mathbf{d}|\mathbf{M} = \mathbf{m})$. Note that this is just the likelihood function defined in the non-hierarchical inverse problem of geophysical data.

Performing ancestral sampling N times according to the DAG of this model (Fig. 4.1)—i.e., sampling sequentially each conditional probability of the factorized joint distribution in Eq. (4.1)—will output N samples of the joint probability distribution.

4.2.2 Designing data features to inform on structural parameters

Given the described process to sample the hierarchical probabilistic model, we notice the data $\mathbf{d}$ are dependent not only on the higher-level structural parameters s but also on intermediate-level parameters. Here, we design features $h(\mathbf{d})$ from the data $\mathbf{d}$ to retain information related only to the structural parameters s and to reduce the dimensionality of the problem. As mentioned in the Introduction, this reduced dimensionality is required to make the use of statistical techniques—such as kernel density estimation (KDE)—computationally tractable. This implies we will approximate the marginal distribution as $p(s|\mathbf{d}) \approx p(s|h(\mathbf{d}))$. We will consider feature extraction as any function $h(\mathbf{d})$ that maps $\mathbf{d}$ from a space of dimension N_d (the number of data points) to a lower dimensional space of dimension N_h—this would also entail function compositions, e.g. $h(\mathbf{d}) = \psi \circ \xi = \psi(\xi(\mathbf{d}))$ where ψ and ξ are functions. The vector of features will be denoted as $\mathbf{h} = h(\mathbf{d})$ and is of dimension N_h. Ideally, feature extraction of data should (1) retain all information regarding the structural parameter (be informative), and (2) disregard information not related to it (dimension reduction).

A first approach for feature extraction consists in using dimension reduction techniques, or so-called data-driven approaches also referred to as continuous latent variables (Bishop, 2006), which aim to retain as much variability of the original data as possible but with a low dimensional representation of the data. In our study, we consider principal component analysis (PCA) and multidimensional scaling (MDS). PCA is based on the eigendecomposition of the data covariance matrix—the eigenvectors represent orthogonal directions following an order of maximum variability and the corresponding eigenvalues state the magnitude of this variability. By disregarding eigenvectors, PCA can be used as a linear dimension reduction method (it is only based on rotation and scaling operations). MDS takes dissimilarities (or distances) between data samples as input and then maps these samples in a lower-dimensional space by approximating the original distances. This may be achieved by optimizing a so-called stress function—a method which is referred to as Scaling by MAjorizing of a COmplicated Function (SMACOF) (De Leeuw and Heiser, 1980). In this way, MDS works as a non-linear dimension reduction method. When using MDS, one can also choose distance functions that are more suited to state the dissimilarity of interest (Scheidt and Caers, 2009). Note that in practice, mapping back to the original distribution is not exact because we disregard some information by considering only the first components of a decomposition for PCA or by retaining only relative distance between samples for MDS.

A second approach consists in designing $h(\mathbf{d})$ to extract specifically information linked to the structural parameters s using domain knowledge, leading to the so-called insight-driven features (Morzfeld et al., 2018). For instance, Hermans et al. (2015) applied an insight-driven approach favoring a combination of inversion and multidimensional scaling (MDS) to extract relevant features for the geological scenario, while Scheidt et al. (2015a) used a wavelet transform on seismic reflection data in combination with an L^2-norm distance as insight-driven feature to update different uncertain geological parameters. Since these functions depend on the specific combination of structural parameter s and data $\mathbf{d}$, they will be detailed in the following sections. As previously mentioned, in our case $\mathbf{d}$ are ground-penetrating radar (GPR) traveltime data collected in cross-borehole tomographic mode. Note that this could also apply to seismic traveltime.

In this Chapter, when using insight-driven features we always consider their combination with data-driven approaches, i.e. we apply first an insight-driven approach and then use data-driven techniques to further reduce dimensionality while retaining most information (this further reduction in dimensions is to enable the application of kernel density estimation as will be explained below). As a result, all of our features may be considered within the so-called metric space modelling (Park et al., 2013; Scheidt and Caers, 2009). Whether using a data-driven approach or a composition of insight-driven and data-driven approaches, we will refer to the number of dimensions after feature extraction as N_h.

Extracting data features for discrete geological scenario

While considering the uncertainty of different geological scenarios formulated as a discrete structural parameter s, we propose extracting features from tomographic data in six different ways summarized in Table 4.1. The first two approaches use dimension reduction techniques (PCA and MDS) on the traveltime directly, and the remaining use dimension reduction techniques but only after an initial insight-driven transformation. The third and fourth approaches transform the data using a histogram and the last two rely on an inverse transform of the data (tomogram).

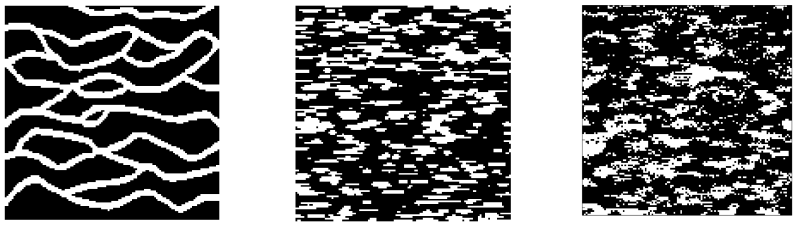
The targeted discrete structural parameter s is implicitly linked to the connectivity of the medium, i.e. each scenario implies the use of a geostatistical algorithm with defined inputs that is expected to produce different degrees of connectivity (Figs. 4.2 and 4.3a). The histogram transformation for cases PCA_h and MDS_h was chosen because differences in connectivity are expected to cause different distributions of traveltimes. For example, if the system is well-connected, the histogram of the traveltimes will show high values for the bins in faster traveltimes and also a multi-modal distribution. This can be observed on Fig. 4.3c (top and bottom). Indeed, the ray paths follow complex patterns for different source-receiver offsets which may be described as the ray "jumping" from one high velocity to another high velocity object, if a high number of jumps occurs the histogram of traveltimes will tend to be smooth, on the other hand if a low number of jumps occurs the histogram will display multi-modality. To estimate

Case	Insight-driven	Data-driven	distance
PCA_t	-	PCA	-
MDS_t	-	MDS	euclidian
PCA_h	histogram	PCA	-
MDS_h	histogram	MDS	Jensen-Shannon
MDS_v	smooth inversion	MDS	euclidian
MDS_c	smooth inversion and connectivity	MDS	euclidian

Table 4.1: Different feature extraction cases proposed for geological scenario from top to bottom: PCA on data (PCA_t); MDS with euclidian distance on data (MDS_t); PCA on histograms of traveltime data (PCA_h); MDS with a Jensen-Shannon distance function on histograms of traveltime data (MDS_h); MDS with euclidian distance on geophysical images obtained by regularized inversion of traveltime data (MDS_v); MDS with euclidian distance on connectivity curves obtained from geophysical images (MDS_c).

the distance between two histograms or probability distributions, we used the Jensen-Shannon distance. As suggested by (Scheidt et al., 2015b), the Jensen-Shannon distance (or the square root of the Jensen-Shannon divergence) is an appropriate metric to measure the distance between two probability distributions or, as in our case, their approximations in the form of histograms. We note that the choice of metric must be made in order to better discriminate the parameter of interest by means of the features so far extracted from the data.

Connectivity may also be quantified if one has access to the knowledge of the spatial distribution of the facies which, in the case of geophysical traveltime data, can be easily approximated using a deterministic inversion (Fig. 4.3d). To quantify the connectivity, we used the Euler characteristic curve in case MDS_c (Renard and Allard, 2013) by thresholding the inverted velocities in 100 steps (see Fig. 4.3e). In other words, we obtain the range of velocity values on each inverted "image" and divide it in 100 intervals, then use the upper bound of each interval to get a binary "image" (i.e. all values lower than the upper bound are set to 1 and the remaining to 0) and compute the Euler characteristic for each of these binary images. The result is then a 100-dimensional vector that is a discrete version of the Euler characteristic curve. The Euler characteristic is a topological characteristic and for binary images is equal to the number of objects



(a) (b) (c)

Figure 4.2: Three different geological scenarios considered: (a) and (b) are training images used for multiple-point geostatistics simulations, and (c) is a realization of a truncated Gaussian simulation with its anisotropic variogram fitted to the training image in (a).

(or clusters) minus the number of holes in such objects (Renard and Allard, 2013). For comparison, we also used directly the inverted images in case MDS_ν.

Extracting data features for continuous channel orientation

The targeted continuous structural parameter s is again linked to the connectivity of the system but here quantified by the orientation of the connectivity rather than the degree of connectivity. In this case, we propose four feature extraction cases (Table 4.2) in addition to the dimension reduction techniques: two based on what we call "oriented averages" of traveltime and two based on tomograms (inverted velocities).

The oriented averages in cases PCA_a and MDS_a were proposed to inform on the orientation of the channel by computing the average of traveltime data in all possible orientations of source-receiver combinations. For the oriented averages in our synthetic setup (described below) we get 37 orientations, hence a vector of 37 insight-driven features.

The Radon transform in the MDS_R case is a line integral transform that is equivalent to a linear tomography taken at constant offsets and in a series of directions (Durrani and Bisset, 1984). It has been used to extract orientation information of images (see e.g. Aydin and Caers, 2013) and we compute it considering eight different directions $\{0, \pi/6, \pi/4, \pi/3, \pi/2, 2\pi/3, 3\pi/2, 4\pi/3\}$ in radians.

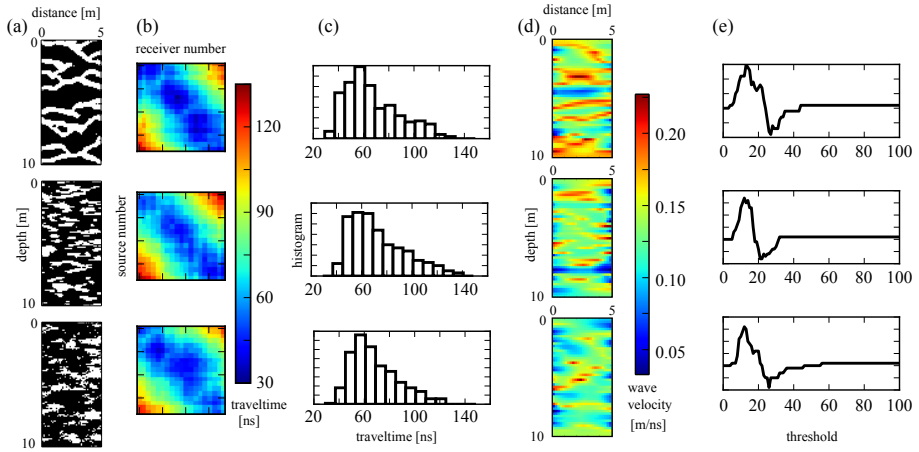


Figure 4.3: Realizations of geological facies with proposed features for the discrete structural parameter: (a) facies samples, (b) simulated traveltimes (reciprocal data not shown), (c) histogram of simulated traveltimes, (d) deterministic inversion of simulated traveltimes, and (e) Euler characteristic curves.

Case	Insight-driven	Data-driven	distance
PCA_t	-	PCA	-
MDS_t	-	MDS	euclidian
PCA_a	oriented averages	PCA	-
MDS_a	oriented averages	MDS	euclidian
MDS_v	smooth inversion	MDS	euclidian
MDS_R	smooth inversion and Radon transform	MDS	euclidian

Table 4.2: Different feature extraction cases proposed for preferential orientation. From top to bottom: PCA on data (PCA_t); MDS with euclidian distance on data (MDS_t); PCA on oriented averages of traveltime data (PCA_a); MDS with euclidian distance on oriented averages of traveltime data (MDS_a); MDS with euclidian distance on geophysical images obtained by regularized inversion (MDS_v); MDS with euclidian distance on a Radon transform of geophysical images (MDS_R).

For comparison, we also used directly the inverted images in case MDS_ν.

4.2.3 KDE and cross-validation approach

We chose to apply kernel density estimation (KDE) to approximate the marginal distributions $p(s, h(\mathbf{d}_{obs}))$ and $p(\mathbf{d}_{obs})$ using the features samples (obtained by applying the transformations of the previous sections to the Monte Carlo data samples) to compute the posterior $p(s|h(\mathbf{d}_{obs}))$. Heteroscedasticity may arise due to the nature of the structural parameter or be induced by the transformations of the feature extraction. To handle such heteroscedasticity, we use an adaptive version of KDE that is based on clustering of the samples similar to the ones proposed by Park et al. (2013) and Scheidt et al. (2015b). As a result, our implementation takes the following form

$$\begin{aligned} p(s|h(\mathbf{d}_{obs})) &= \frac{p(s, h(\mathbf{d}_{obs}))}{p(\mathbf{d}_{obs})} \\ &= \frac{\sum_{i=1}^{N_c} \sum_{s_j, \mathbf{d}_j \in C^{(i)}} K_{H_s}^{(i)}(s - s_j) K_{H_h}^{(i)}(h(\mathbf{d}_{obs}) - h(\mathbf{d}_j))}{\sum_{i=1}^{N_c} \sum_{s_j, \mathbf{d}_j \in C^{(i)}} K_{H_h}^{(i)}(h(\mathbf{d}_{obs}) - h(\mathbf{d}_j))} \end{aligned} \tag{4.2}$$

where N_c is the number of clusters used in the clustering algorithm, $C^{(i)}$ refers to the i-th cluster from the set $\{C^{(i)}|i = 1, ..., N_c\}$, s_j and $\mathbf{d}_j$ are the values for the structural parameter and the data for the j-th sample, therefore the index $j = \{1, ..., N\}$, $K_H^{(i)}(\cdot)$ refers to a scaled kernel function with corresponding bandwidths H_s for the structural parameter and H_h for the data whose values depend on which cluster $C^{(i)}$ they belong to, and $\mathbf{d}_{obs}$ is the observed data. Further details on adaptive kernel density estimation and our particular implementation are presented in the Appendix. What is important to note here is that the bandwidths H_s and H_h are parameters controlling the shape or "smoothing" of the distribution in the joint space $p(s, h(\mathbf{d}))$ and they are implicitly given by N_c.

As previously mentioned regarding the possible heteroscedastic character of the posterior distribution of the structural parameter, adaptive KDE was chosen because it (1) accounts for the degree of uncertainty as a function of the structural parameter s and (2) adjusts the error model in the feature space (i.e. non

Gaussian). In the latter case, the noise model for the data is no longer valid for the features. Instead of handling this using "perturbed" observations (Hermans et al., 2016; Morzfeld et al., 2018), adaptive KDE can deal with this directly because it works for heteroscedastic and multimodal distributions.

At this point we should note that our methodology results in three main degrees of freedom, namely the number of Monte Carlo samples N (section 4.2.1), the number of dimensions N_h after feature extraction and dimension reduction (section 4.2.2) and the number of clusters N_c (this section and the Appendix) used in the adaptive KDE. Because the evaluation of the numerical model is usually the most computationally demanding step, a low value of samples N should be chosen. Then, N_h and N_c should be chosen so that the method performs optimally. To choose this optimum, we propose a leave-one-out cross-validation approach with two different scores depending on the type of structural parameter being estimated. For discrete parameters, N_h and N_c can be fixed by using the number of correct classifications obtained by assigning the scenario with the highest (marginal) posterior probability at the data sample. In case of equal number of correct classifications, we take the mean of all the (marginal) posterior probabilities of the correctly classified scenarios, termed here as ℓ_d, and pick the one with the highest value (Hermans et al., 2015). For continuous parameters, the proposed cross-validation approach is based on a likelihood score defined by (Habbema et al., 1974) as

$$\ell_c = \frac{1}{n} \sum_{j=1}^{n} \ln p_{-i}(N_h, N_c) \tag{4.3}$$

where p_{-i} stands for the leave-one-out estimate of the conditional distribution $p(s|h(\mathbf{d}_i))$, i.e., the probability value computed at the i-th point without considering the same point in the adaptive KDE.

We compare our cross-validation approach to the silhouette index proposed by Scheidt et al. (2015b), in a simple one-dimensional example (Fig. 4.4) of applying adaptive KDE when the data error model is Gaussian and we aim to estimate its probability density but we can only work with features (e.g. a nonlinear feature, like the exponential $h(x) = e^x$ in Fig. 4.4) as in our approach.

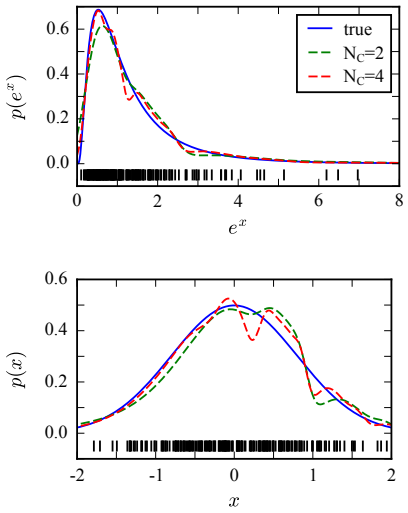


Figure 4.4: Comparison of the adaptive KDE for a non-linearly transformed space when the number of clusters is chosen by silhouette index (N_c=2) and by cross-validation (N_c=4). Vertical markers in the lower part denote the samples used to approximate the probability distribution. Both plots show the same samples, with the upper one representing an exponential transformation of the values in the lower one.

Since it is not easy to visually discern which curve gives a better approximation, we used the Jensen-Shannon distance (see Section 4.2.2) to measure the distance with respect to the true distribution. Results show that the adaptive KDE would better approximate the original error model in this feature space when the number of clusters, $N_c = 4$, is estimated through cross-validation, which generally produces an optimal bias-variance tradeoff, instead of the result $N_c = 2$ obtained with the silhouette index (Fig. 4.4). In our case, the probability distribution to be approximated is $p(s, h(d))$ instead of $p(e^x)$ and $p(s, d)$ would be in place $p(x)$. Another advantage of cross-validation is that it can always be applied since we can always generate the necessary Monte Carlo samples. Given high-dimensional and more complex distributions, we expect the use of cross-validation will be more beneficial.

4.3 Reducing structural uncertainty using features of GPR traveltime on a synthetic model

4.3.1 Model set-up

A synthetic case is presented in this section to demonstrate the proposed methodology using GPR traveltime as data $\mathbf{d}$. The spatial domain is a vertical section between two boreholes separated 5 m from each other and whose depth is 20 m (4.3). As in the usual tomographic survey, data is generated by considering the sources are in one borehole while receivers are located in the other. Afterwards, reciprocal data is simulated by placing sources in the borehole where receivers were firstly placed and vice-versa. Vertical separation of both the receivers and sources is constant and equal to 0.5 m. We consider 19 sources and 19 receivers (and the same number for reciprocal data) where the first position of the receivers/sources is 0.5 m from surface and last is 19.5 m.

In our specific synthetic demonstration we study two cases, one including a discrete structural parameter and the other a continuous one, for which we describe steps 1 to 5 in Fig. 4.1. An outline of the hierarchical sampling for the discrete structural parameter is presented in Fig. 4.5. Note that steps 3 to 5 are common for both types of structural parameters.

Step 1. The discrete structural parameter $s \in \{s_1, s_2, s_3\}$ denotes three different geological scenarios, represented by three different geostatistical models: two multiple-point geostatistics models with different training images and one truncated gaussian field model (Fig. 4.2). Each row in Fig. 4.5 corresponds to a value of s. The prior $p(s)$ is a discrete uniform distribution and we consider this implicitly by using 50 samples of each value $\{s_1, s_2, s_3\}$ for a total of 150 samples. These 150 samples are used in the following steps.

The continuous structural parameter is the preferential orientation of the geological patterns (channels in our case) and its range is $s \in (0, \pi)$. 200 samples were obtained from a uniform distribution with range $(0, \pi)$. This range was chosen because the training image used (see realizations in Fig. 4.9a) has a rotational symmetry of order two, i.e. data realizations from s and $s + \pi$ can be considered coming both from s only.

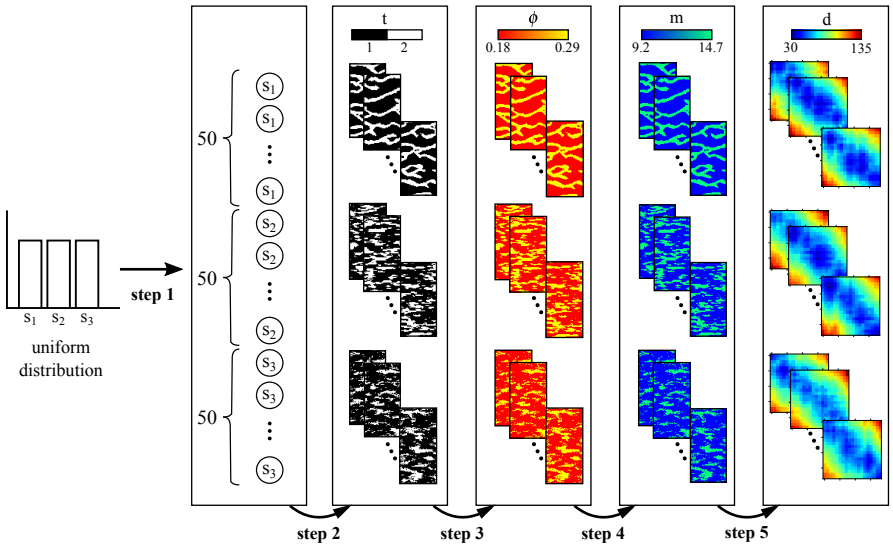


Figure 4.5: Sketch of the hierarchical sampling process for the geological scenario where s is the geological scenario index, t is the facies index, ϕ is the porosity, m is the relative dielectric permittivity and d are electromagnetic wave traveltimes in nanoseconds plus added Gaussian noise with $\sigma = 1.4\ ns$. Steps 1 through 5 are described in the text.

Step 2. In the discrete case, we obtain facies samples $\mathbf{t}$ from s_1 and s_2 by means of multiple-point geostatistics sequential simulations (two first rows for the t column in Fig. 4.5), considering training images ti_1 and ti_2 (Fig. 4.2), respectively. The $\mathbf{t}$ samples from s_3 are generated by truncated sequential gaussian simulation, whose anisotropic spherical variogram was obtained by fitting to the training image ti_1 (last row for the t column in Fig. 4.5). The samples of the continuous s, are used as input to generate samples of $\mathbf{h}$ by multiple-point geostatistics simulations using training image ti_1.

Step 3. The porosity $\boldsymbol{\phi}$ is given by a constant mapping $\mathbf{q}(\mathbf{t})$ of the facies and the probability distribution can be expressed as

$$p(\boldsymbol{\phi}|\mathbf{T}=\mathbf{t}) = \delta(\boldsymbol{\phi} - \mathbf{q}(\mathbf{t})) \tag{4.4}$$

where δ is the delta function and

$$\mathbf{q}(\mathbf{t}) = \begin{cases} q_1 & t = 1 \\ q_2 & t = 2 \end{cases} \tag{4.5}$$

where $q_1 = 0.18$ and $q_2 = 0.29$ are porosity values for two different geological facies. This amounts to assigning a porosity value for each facies (the ϕ column in Fig. 4.5), but we choose to express it as a conditional probability to be consistent with the Bayesian hierarchical model, where uncertainty may be included at this step to consider e.g. intrafacies variability.

Step 4. We choose a mixing model named CRIM (Birchak et al., 1974) to transform the porosity field into a dielectric permittivity field (the m column in Fig 4.5). Such transformation is denoted by $\mathbf{r}(\boldsymbol{\phi})$ and the corresponding probability distribution is

$$p(\mathbf{m}|\boldsymbol{\Phi}=\boldsymbol{\phi}) = \delta(\mathbf{m} - \mathbf{r}(\boldsymbol{\phi})) \tag{4.6}$$

again δ is the delta function and

$$\mathbf{r}(\boldsymbol{\phi}) = ((1-\boldsymbol{\phi})\sqrt{\epsilon_s} + \boldsymbol{\phi}\sqrt{\epsilon_w})^2 \tag{4.7}$$

where $\epsilon_s = 3$ is the permittivity of the solid grains and $\epsilon_w = 81$ is the

permittivity of water. In this way, the facies $t = 1$ will have lower permittivity (therefore, higher electromagnetic wave velocity) than the facies $t = 2$.

Step 5. Numerical modeling of the electromagnetic wave traveltime is done by a ray-path approximation model, as implemented in PyGIMLi's Refraction module (Rücker et al., 2017). Note this approximation reduces computational demand compared to full-waveform simulation. Interestingly, within a feature-based framework, traveltime data can be seen as a first feature extraction step from the full-waveform data. The corresponding probability distribution is

$$p(\mathbf{d}|\mathbf{M} = \mathbf{m}) \sim \mathcal{N}(f(\mathbf{m}), I\sigma^2) \tag{4.8}$$

where $\mathcal{N}$ stands for a multivariate normal distribution, I is an identity matrix of size N_d, $f(\cdot)$ is the geophysical forward operator given by the numerical model mentioned above, and $\sigma = 1.4\ ns$ states the magnitude of independent normally-distributed noise in the geophysical data. Simulated traveltimes data are shown in data arrays (where columns represent the receiver index and rows the source index) in the d column of Fig. 4.5. Note that uncertainty was not considered in steps 4 and 5 here but could easily be included.

We generate samples of the (marginal) joint distribution $p(s, \mathbf{d})$ by following steps 1 to 5 and disregarding the parameter dimensions.

4.3.2 Results for a discrete structural parameter

We extract features of traveltime data to approximate the posterior distribution of the structural parameters $p(s|\mathbf{d}) \approx p(s|h(\mathbf{d}))$ according to the six different cases mentioned in Section 4.2.2. Fig. 4.3 shows one realization for each value of the discrete structural parameter, the simulated traveltime data and the corresponding insight-driven features.

Cross-validation was used to select the number of dimensions, N_h, and the number of clusters, N_c, for each one of these cases. We restricted to values $N_h \leq 10$ and $N_c \leq 15$ since the number of samples needed to obtain a good estimate with KDE beyond this bound would be too high. The cross-validation score used was the number of correctly classified realizations, i.e. an integer between 0 and 150, recalling we generated 50 samples for each value of the discrete

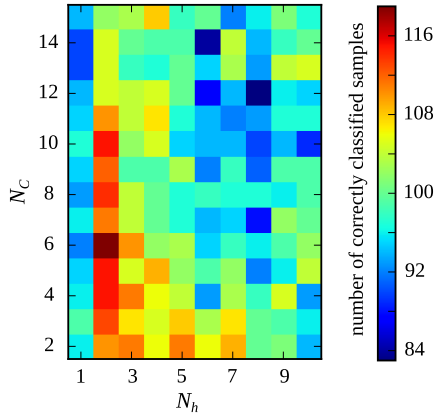


Figure 4.6: Cross-validation matrix for the case MDS_h of the discrete structural parameter (geological scenario).

structural parameter. Fig. 4.6 shows the cross-validation matrix obtained for the case MDS_h where we can see there is an optimum choice of N_h and N_c that is within our chosen search limits for both parameters. In the cross-validation matrix, we see a counterbalancing of N_h and N_c: within the bound $N_h \leq 8$, the classification maxima for increasing N_h generally correspond to lower values of N_c. Since the same number of samples is considered, this may be explained because lower values of N_c mean the adaptive KDE is using wider bandwidths when going into higher dimensional spaces, effectively covering more space in the density estimation than with a higher N_c. However, the effect of N_h is stronger and leads to better classification, which is also an indication of a properly chosen feature extraction to reduce dimensions. For this reason, in case of the same performance, we rather choose the combination where N_h is lower. Note also that an arbitrary chosen combination of N_c and N_h could easily lead to a significantly lower performance of the approach, highlighting the need to optimize the choice of those degrees of freedom.

Fig. 4.7 shows the MDS mapping applied to the histograms of traveltime data in the low dimensional feature space for $N_h = 2$ (the optimum selected by our approach). Points are approximately separated according to the three values of

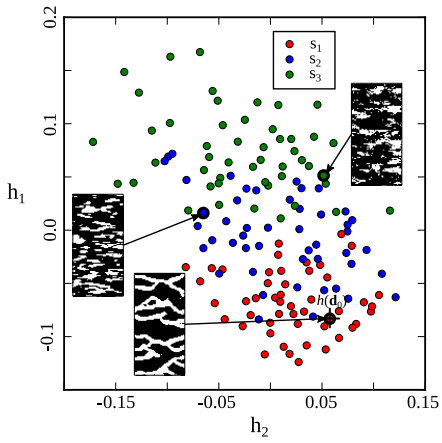


Figure 4.7: MDS applied on histograms of traveltime data. Examples of realizations for each value of the discrete structural parameter s are shown.

the discrete structural parameter s which means the features are informative on this structural parameter. The joint probability distribution $p(s, h(\mathbf{d})) = p(s, \mathbf{h})$ obtained through adaptive KDE is shown also for this case MDS_h (Fig. 4.8a). The estimation of the posterior probability of the structural parameter for one data sample d_0 with known true value of $s = s_1$, equivalent to one computation of the leave-one-out cross-validation is shown in Fig. 4.8b, where we see the method correctly gets the value s_1 as the most likely for d_0. We also note here that the probability of s_3 is very close to zero. If d_0 were measured geophysical data, this geological scenario would be falsified and could be left out of further analysis (e.g. inversion for spatial parameters).

For the other cases, a complete visualization is difficult due to the higher dimensionality of N_h but a summary of the results are shown in Table 4.3. Some cases show a higher number of correctly classified samples (66% correctly classified for the worst case and 80% for the best one) but with different values for N_h and N_c. Also, the values of mean updated probability ℓ_d are higher for certain cases but to a lesser degree than for the number of correct classifications. The best performing strategy is the composition of MDS on histograms of traveltime (MDS_h). This means our proposed insight-driven feature has indeed aided to some extent in retaining information only on the structural parameter

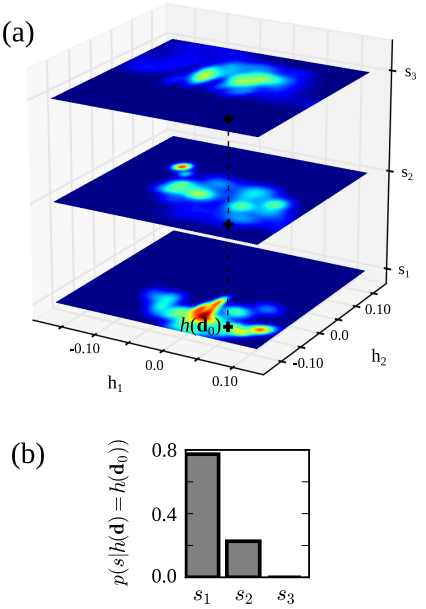


Figure 4.8: (a) Joint probability distribution $p(s, \mathbf{h})$ for the case of MDS on histograms of traveltime data. The '+' denotes one realization $\mathbf{d_0}$ when the discrete parameter $s = s_1$ and is the same shown in Fig. 4.7. (b) The posterior probability of the structural parameter s obtained by cross-validation when $\mathbf{d} = \mathbf{d_0}$.

	N_h	N_c	$class$	ℓ_c
PCA_t	8	3	99	0.61
MDS_t	6	14	100	0.66
PCA_h	3	3	108	0.61
MDS_h	2	6	119	0.69
MDS_v	2	7	102	0.62
MDS_c	6	4	109	0.65

Table 4.3: Cross-validation results for discrete structural parameter s where $class$ refers to the number of correct classifications.

s. The connectivity-based approach (MDS_c) does not perform better than the data-driven approach. However, it is more discriminating than the tomograms (MDS_v). Those approaches are less effective in terms of computational demand, since they require both a deterministic inversion and computation of the Euler characteristic curves for each realization. This result might appear counter-intuitive as imaging is generally appealing for the human eyes and a common result of geophysical exploration. However, inversion can be considered as a feature extraction of data leading to loss of information related to the regularization operator. We note, however, that these results are related to the type of data (cross-hole GPR traveltime, in our case) and might differ for other data or even other acquisition setups. For instance, surface ERT data has been shown to be extremely sensitive to shallow resistivity structure hence a possible strategy is to extract features from the geophysical image rather than directly from the data or to develop more appropriate insight-driven features (Hermans et al., 2015).

We note a small improvement on the classification scores between PCA, a linear dimension reduction method, and MDS, a non-linear dimension reduction method. This may be explained as MDS being able to account for some non-linearity in the relation of the structural parameter with the data. Also, we see that a higher dimensionality is chosen (through cross-validation) for PCA in comparison with MDS, which may be because both methods are able to retain similar information but with different N_h.

4.3.3 Results for a continuous structural parameter

As previously mentioned six different cases are considered in which both data-driven and a composition of insight-driven with data-driven features are used (section 4.2.2).

The number of clusters N_c and the number of dimensions N_h was selected according to cross-validation using the minimum value for the score of Eq. (4.3) (third column in Table 4.4). Again, we restricted to $N_h \leq 10$ and $N_c \leq 15$. The chosen number of dimensions for the case PCA_t is $N_h = 3$ so, in order to represent the complete space where the method is applied, we would have to use three dimensions. However, for visualization purposes, we use the first two and show the distribution of realizations of features of the data (Fig. 4.9a). Here, the insets display four samples of the corresponding geological facies for which the simulated data and the PCA features were obtained. We clearly see that points are arranged according to the value of the structural parameter s which means that they are informative of it. Moreover, the distribution of samples reveals that the obtained features are probably linearly related to the structural parameter since they plot close to a circle and orientation is circular (i.e. periodic). Indeed, if we take this into account and plot the orientation versus the angle formed by the two features we see a linear trend (Fig. 4.9b). The scatter plot reveals a small degree of heteroscedasticity for this specific dataset (higher variance around 0.25π and 0.75π and lower variance around 0.5π and 0) which is also present for the other cases (MDS_t, PCA_a and MDS_a). However, due the small number of samples (200), this may not be statistically significant, therefore the process was repeated with 500 samples where the change in spread as a function of the orientation is clearer (not shown). This means cross-borehole GPR data is more discriminative in angles close to $0°/180°$ and $90°$ and is less discriminative for angles close to $45°$ and $135°$. This could be physically explained by the fact that changes in the length of the wave path through low velocity zones are greater when the angles are close to $0°/180°$ or $90°$. Further analysis is required to validate this conclusion, e.g. prove that the chosen dimension reduction techniques did not affect the results.

For the case PCA_t, Fig. 4.10a shows the distribution of features of the

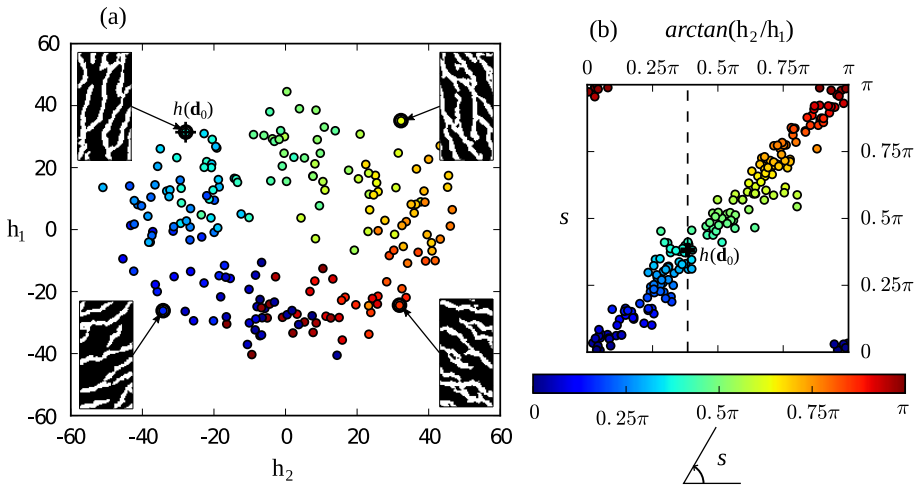


Figure 4.9: (a) PCA applied on traveltime data for continuous structural parameter s. (b) Same samples as (a) but computing the angle formed by the two features and plotting versus true orientation. Colors are true values for the structural parameter s (preferential orientation). $\mathbf{d_0}$ denotes a particular sample taken out during cross-validation and the dashed line denotes the position in the feature axis for this sample.

data together with the continuous structural parameter s and Fig. 4.10b shows the marginal distributions of the corresponding three-dimensional joint probability distribution $p(s, h_1, h_2)$. In order to apply the adaptive KDE to this circular parameter the bandwidth for the structural parameter dimension was computed in a transformed space (i.e. a two-dimensional space with $x = sin(s)$ and $y = cos(s)$) and the periodicity was accounted for by means of replication of samples in the boundaries (Silverman, 1986). For the other three cases, the method works similarly but its application is harder to visualize given the high number of dimensions N_h selected.

Since we are dealing with a continuous parameter, the posterior probability distribution is also continuous. The process of building this distribution is depicted in Fig. 4.10 and the resulting posterior probability distribution for a certain value $\mathbf{d_0}$—taking its value out in the adaptive KDE while performing leave-one-out cross-validation—is shown in Fig. 4.11. We clearly see that the posterior contains the true value and it is sharply peaked around it which means the method is correctly estimating the structural parameter s. Given that the prior distribution was uniform, the achieved reduction of uncertainty is on the order of 75%.

A summary of the obtained results is shown in Table 4.4 which indicates the best performing case is MDS_t, but it is not far from PCA_t. The similar results of these two cases mean there is no clear advantage in using a non-linear dimension reduction method and may be explained by the mostly linear relation between the structural parameter and the data (as shown by Fig. 4.9b). We also see that data-driven approaches applied alone perform better than their compositions with insight-driven features, which are used in the last four cases. This means that our chosen insight-driven features provide no better strategy to retain information on the structural parameters s than the data-driven approaches by themselves. This may be explained to some extent by the fact that both PCA and MDS were designed to explicitly search for continuous parameters (also termed continuous latent variables) that explain variability in the data (Bishop, 2006), and not discrete parameters as the ones in the last section. Also, in this case working with the geophysical images gives the worst results and this was not improved by the chosen insight-driven feature (Radon transform).

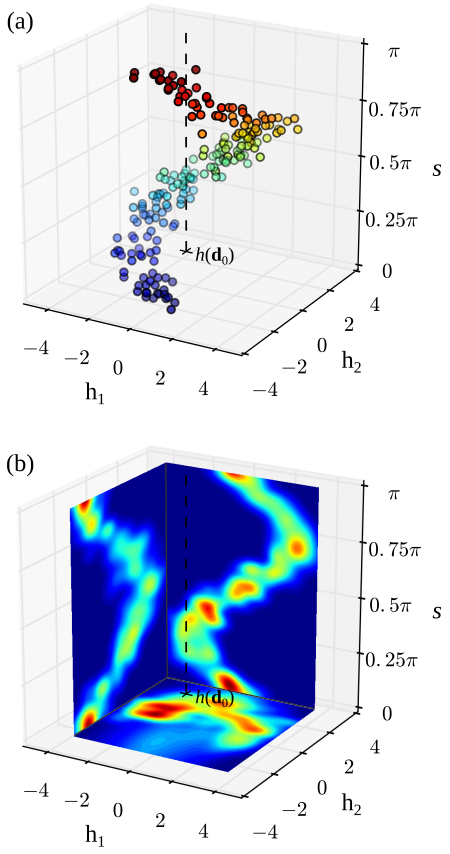


Figure 4.10: PCA applied on traveltime data showing structural parameter s as third dimension (a) colors are the same as in Fig. 4.9. Marginal distribution resulting from the application of adaptive KDE (b). The '+' denotes the sample $\mathbf{d_0}$ taken out during cross-validation and is the same as the one referenced in Figs. 4.9 and 4.11. The dashed line shows the conditioning to $\mathbf{d_0}$ in $p(s|\mathbf{d_0})$, therefore highlights the direction along which the adaptive KDE is applied.

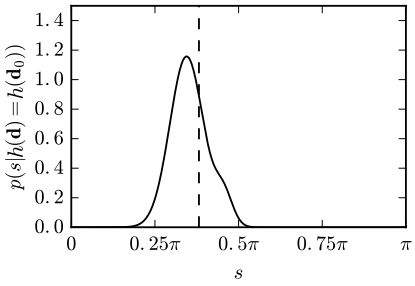


Figure 4.11: Posterior probability for one sample $\mathbf{d_0}$ taken out during cross-validation computed using features $h(\mathbf{d})$ obtained with PCA directly applied to traveltime data. The vertical dashed line denotes the true value of the sample.

	N_h	N_c	ℓ_c
PCA_t	3	5	-0.270
MDS_t	4	6	-0.257
PCA_a	3	7	-0.364
MDS_a	5	5	-0.328
MDS_v	2	5	-0.614
MDS_R	4	5	-0.675

Table 4.4: Cross-validation results for continuous structural parameter s.

We must note that our data is highly sensitive to the preferential orientation therefore the dimensions explaining most data variability are indeed related to the chosen structural parameter. When this is not the case, insight-driven features may prove more useful. Finally, it is worth mentioning that insight-driven features are easier to propose when the parameter of interest is discrete, since the expected effect on data can be investigated in a finite number of scenarios.

4.4 Conclusions

In this Chapter we provide a novel framework to design and assess data features in the approach proposed by Park et al. (2013)—an approach to reduce the structural parameter uncertainty—making it more objective and readily applicable. Our results show that the design and relative success of data features on which the

approach is based is case-dependent, which may therefore challenge the robustness of the approach. Since cross-validation can always be applied, our proposed framework relies on its use to make an objective assessment of the features and the additional degrees of freedom brought by the method.

To illustrate the different choices of feature extraction methods, these were analyzed according to whether they are data-driven only or based on insight about the relation between the data and the structural parameter. In the presented synthetic cases, cross-validation identified the defined insight-driven features as more successful to retrieve the posterior (marginal) probability distribution of a discrete structural parameter (the geological scenario) than for a continuous one (the preferential orientation). Similarly, data-driven approaches performed better for the orientation according to the cross-validation scores and we argue that this is mainly because a significant part of data variability is explained by this structural parameter. We also found that, for the synthetic cases considered in this Chapter, there is not much difference in using a data-driven linear dimension reduction method (such as principal component analysis), in comparison to a nonlinear one (such as multidimensional scaling), other than the former will generally require more dimensions to achieve a similar performance. As an additional result, some useful ways to extract features were proposed when reducing the uncertainty of the geological scenario and the preferential orientation using geophysical tomographic data. All these outcomes may prove useful in the general context of multi-level uncertainty in the Earth sciences. An interesting result of our investigations is that, although geophysical data are often used to produce images of the subsurface through inversion, using the inversion as an insight-driven feature is not necessarily a good approach to reduce the uncertainty on structural parameters. The data themselves can be more informative.

When using data-driven feature extraction techniques, we considered mainly the dimensions that explain most of the variability in the data. It may be interesting for future studies to consider also combinations of different dimensions (maybe excluding the ones explaining most variability) to see if they are more informative on structural parameters, hence provide a better estimation for the structural uncertainty. This may prove especially useful when the structural parameter does not have a major impact on data variability. In the same regard, this

suggests using supervised dimension reduction techniques could be beneficial.

Chapter 5

Reducing data and model dimension: prior falsification followed by inversion with an assembled prior[1]

Prior information regarding subsurface patterns may be used in geophysical inversion to obtain realistic subsurface models. Field experiments require sufficiently diverse patterns to accurately estimate the spatial distribution of geophysical properties in the sensed subsurface domain. A variational autoencoder (VAE) provides a way to assemble all patterns deemed possible in a single prior distribution. Such patterns may include those defined by different base training images and also their perturbed versions, e.g. those resulting from operations such as erosion/dilation, local deformation and intrafacies variability. Once the VAE is trained, inversion may be done in the latent space which ensures that inverted models have the patterns defined by the assembled prior. Inversion with both a synthetic and a field case of cross-borehole GPR traveltime data shows that using the VAE assembled prior performs as good as using the VAE trained on the

[1]**Note:** The research presented in this chapter is based on: Lopez-Alvis, J., Nguyen, F., Hermans, T. and Looms, M. (2021). Geophysical inversion using a variational autoencoder to model an assembled spatial prior uncertainty. *To be submitted.*

pattern with the best fit, but it has the advantage of lower computation cost and more realistic prior uncertainty. Moreover, the synthetic case shows an adequate estimation of most small scale structures. Estimation of absolute values of wave velocity is also possible by assuming a linear velocity model and including two additional parameters in the inversion.

5.1 Introduction

As detailed in Chapter 1, geophysical inversion estimates the values of the spatial model parameters by combining information regarding the model itself, the measured data and a forward operator, which gives a relation between model parameters and data by describing approximately the physical process by which the data arose. When data does not provide sufficiently independent information about the distribution of subsurface properties, inversion relies on regularization to stabilize the solution (Backus and Gilbert, 1967; Tikhonov and Arsenin, 1977) but this inherently biases the solution towards an a priori constraint which may not be realistic and therefore may hinder the use of the model for certain applications. If information regarding spatial patterns of the subsurface is available it may be used together with measured data in order to improve model realism (Tarantola and Valette, 1982). This information is typically obtained from independent knowledge about the subsurface structure, e.g. outcrops which are representative of the local geology (Linde et al., 2015). To integrate this information with measured data, the patterns must be described by techniques that account for their spatial nature. This has been generally achieved by using traditional geostatistical techniques, which usually provide more realistic models than classical regularization by means of imposing a covariance structure (Franklin, 1970; Maurer et al., 1998). The choice of geostatistical technique depends on both the complexity of the spatial patterns and the information content of the measured data (Mariethoz, 2018). In general, it is recognized that multiple-point geostatistics (MPS) is more suited to reproduce highly-connected spatial structures than covariance-based (or Gaussian random field) methods (Strebelle, 2002; Journel and Zhang, 2007). Recently, deep generative models (DGMs) have been proposed as an alternative to MPS to reproduce such complex spatial patterns (Laloy et al.,

2017; Chan and Elsheikh, 2019).

MPS and DGMs rely on a gridded (pixel) representation for generating high-resolution spatial realizations. An Euclidian space $\mathbb{R}^N$ may be assumed for this representation where N is the number of pixels, then models may be seen as points in a high-dimensional model space. Since the spatial patterns are restricted, however, the set of possible models will not cover the whole model space. This subset may be stated by a prior probability distribution (Tarantola and Valette, 1982). While both MPS and DGMs are able to approximate such prior distribution and generate new samples with patterns similar to those contained in a training dataset (e.g. a large training image, TI), DGMs present some advantages for inversion. First, contrary to MPS which either saves the number of occurrences of patterns (Strebelle, 2002; Straubhaar et al., 2011) or queries them directly from the TI (Mariethoz et al., 2010), DGMs build a continuous prior probability distribution from which spatial realizations of the patterns are generated. This continuous probability distribution means that DGMs may provide (1) more diverse patterns, i.e. they generate models whose patterns are not necessarily contained in the training image, effectively interpolating between training samples, (2) a direct continuous perturbation step while exploring the model space (Laloy et al., 2017) and (3) the possibility of assembling different kinds of patterns in a single prior probability distribution (Bergmann et al., 2017). Second, given certain conditions, DGMs may also allow for gradient information (of the objective function) to be used in inversion which may substantially reduce the computational cost (Laloy et al., 2019; Mosser et al., 2018; Lopez-Alvis et al., 2020). This is typically not available for inversion with MPS, for which other ways of exploring the model space have been used (Hu et al., 2001; Caers and Hoffman, 2006; Hansen et al., 2012; Linde et al., 2015).

There were two main advances that allowed for DGMs to be applicable to high-resolution images: (1) neural networks that preserve complex spatial information, and (2) inference algorithms that are able to train instances of these networks that specifically include a continuous probability distribution within their layers. A common type of neural network that fulfills the first point are (deep) convolutional neural networks (CNNs) (Fukushima, 1980; LeCun et al., 1989). CNNs are widely used in image processing and computer vision and have shown

to be able to process highly complex spatial patterns (Krizhevsky et al., 2017). DGMs may use CNNs as their generative mapping and therefore produce new high-resolution samples with the training spatial patterns (Radford et al., 2016). Given the high-dimensionality of the model space, the training of such models was only possible with the introduction of inference algorithms that were able to cope with such high-dimensionality. Two main algorithms are currently used to train DGMs: amortized variational inference (Kingma and Welling, 2014; Zhang et al., 2018) and adversarial learning (Goodfellow et al., 2014). The former gives rise to variational autoencoders (VAEs) while the latter produces generative adversarial networks (GANs).

Both VAEs and GANs may be used to generate samples that display the training patterns by sampling from a n-dimensional probability distribution (where typically $n << N$). However, when used for inversion, the concern is not only on pattern accuracy but also on the feasibility of efficiently exploring the possible models that fit the data, or in Bayesian terms, efficiently integrating model prior information with the measured data by means of the forward operator (Mosser et al., 2018; Laloy et al., 2019; Canchumuni et al., 2019). It was recently argued that with certain choice of parameters VAEs may control both the degree of non-linearity and the topological changes of their generative mapping, which in turn allows the gradient to be used in a computationally efficient inversion (Lopez-Alvis et al., 2020). Such choice of parameters is also useful in controlling the diversity of samples: instead of only generating samples very close to the training samples, the probability distribution expands or covers larger regions between the samples what can counterbalance the lack of diversity or finite nature of the training image.

This improved diversity may be useful when the goal is to generate a prior probability distribution which is assembled from different types of patterns (e.g. different TIs), including the case when base patterns are perturbed by operations such as deformation, erosion-dilation and intrafacies variability. This may be advantageous for field data because it increases the number of possible patterns in the subsurface which leads to a better representation of model prior information or uncertainty. However, an important step before considering the different transformed patterns is to check their consistency with the observed data, i.e. if they

are likely to have generated the measurements. This may be framed as a prior consistency check or falsification step (Park et al., 2013; Hermans et al., 2015; Scheidt et al., 2018).

In this Chapter, DGMs are used to impose spatial patterns during geophysical inversion. In particular, the ability of VAEs to build an assembled prior from different base TIs and their perturbed versions is tested. The impact of such assembled prior for modeling the subsurface is assessed by making use of gradient-based inversion for both synthetic and field cases of cross-borehole ground penetrating radar (GPR) traveltime data. The corresponding prior consistency check is done in both cases for all TIs considered. It is worth noticing that the current study constitutes one of the first efforts to apply DGM-based inversion to a field dataset. Also, in contrast to previous studies (Laloy et al., 2017, 2018; Mosser et al., 2018; Lopez-Alvis et al., 2020) the values of the geophysical parameter (wave velocity) are assumed unknown and included in inversion by means of a linear model. The remaining of this Chapter is structured as follows. In section 5.2, an outline of the proposed framework including the underlying theory of VAEs and their use within gradient-based inversion is presented. In this section, the prior consistency step and the field data used to test the framework are also described. Section 5.3 presents and discusses results of the proposed approach: first, a synthetic case that mimics the field case is introduced and then results of the field case are presented. In this section, the relation of the proposed framework with previous studies is also highlighted and suggestions for future work are given. Finally, concluding remarks of this Chapter are presented in Section 5.4.

5.2 Methods

The framework proposed in this Chapter may be summarized as follows:

1. Define a realistic generative model for the subsurface spatial patterns as prior distribution. The generative model may include operations that transform some base patterns such as erosion/dilation, local deformation and intrafacies variability.

2. Check consistency of the defined prior, this may include falsifying some of

the patterns.

3. If the prior is consistent, train the VAE with samples from the generative model. Once trained, the VAE works as an assembled prior, i.e. it is able to generate patterns similar to the training patterns including those transformed by the defined operations.

4. Perform gradient-based inversion in the latent space of the VAE.

All of the methods and concepts required in each of the previous steps are detailed in the following sections.

5.2.1 Variational autoencoder: approximating a complex probability distribution

A variational autoencoder (VAE) may be classified as a deep generative model (DGM). A DGM is a type of probabilistic model that relies on a relatively simple probability distribution $p(\mathbf{z})$ to approximate a more complex one $p(\mathbf{m})$ by passing the samples from the former through a (usually nonlinear) mapping, e.g. a neural network (Dayan et al., 1995; Uria et al., 2014). This mapping is referred to as the generative mapping $\mathbf{g}_\theta(\mathbf{z})$ and may be represented more generally by a conditional distribution $p_\theta(\mathbf{m}|\mathbf{z})$ where θ denotes the parameters of the mapping, e.g. the weights of the neural network. Here, $\mathbf{m}$ is defined in the original model space $\mathbb{R}^N$ while $\mathbf{z}$ is defined in a space $\mathbb{R}^n$. The space $\mathbb{R}^n$ is usually referred to as the latent space and $\mathbf{z}$ is called the code or latent vector. In general, samples $\mathbf{m}$ exhibit some order or structure which means they are confined to a subset $\mathcal{M} \subset \mathbb{R}^N$. This assumption is known as the "manifold hypothesis" (Fefferman et al., 2016) and means that in general it should be possible to define $\mathbb{R}^n$ with $n < N$, for which n is at minimum the dimension of the subset (or manifold) $\mathcal{M}$. This also means that the probability distribution $p(\mathbf{m})$ only needs to be defined over $\mathcal{M}$.

Assuming a large dataset $\mathbf{M} = \{\mathbf{m}^{(i)}\}_{i=1}^{P}$ containing P samples from the complex probability distribution $p(\mathbf{m})$ is available, DGMs are trained by estimating the parameters θ of the generative mapping given a fixed $p(\mathbf{z})$. In this way, one is able to generate new samples similar to those of the training dataset

$\mathbf{M}$ by sampling from $p(\mathbf{z})$ and passing through the generative mapping, i.e. sampling according to $p(\mathbf{z})p_\theta(\mathbf{m}|\mathbf{z})$. However, when the training samples $\mathbf{m}^{(i)}$ are high-dimensional, non-standard inference methods are required to efficiently estimate the parameters θ of the generative mapping. VAEs use a neural network as generative mapping and rely on amortized variational inference to estimate its parameters (Kingma and Welling, 2014; Rezende et al., 2014). This inference technique requires another mapping to approximate a recognition (or variational) probability distribution $q_\vartheta(\mathbf{z}|\mathbf{m})$. In this way the generative mapping may take the output of the recognition mapping as input and vice-versa, which resembles a neural network architecture known as autoencoder (Kramer, 1991), with the generative mapping as decoder and the recognition mapping as encoder. In this Chapter the choices proposed by Kingma and Welling (2014) regarding the probability distributions involved in a VAE are followed. The resulting framework for the VAE is detailed in Section 3.2.3. In the rest of this work, we drop the subindex θ in $\mathbf{g}(\mathbf{z})$ to simplify notation and also because once the DGM is trained, the parameters θ do not change, i.e. they are fixed for the subsequent inversion.

Note that the training dataset $\mathbf{M}$ may contain different kinds of patterns which allow the VAE to effectively learn what is here termed an assembled prior, i.e. a continuous prior distribution which generates not only patterns similar to those in the training set but also those corresponding to the transitions between the training patterns. Bergmann et al. (2017) propose a similar idea for GANs. One may also picture this process as changing or substituting the original (probabilistic) generative model by the VAE, i.e. the latent variables now include jointly the effects of the original variables (Fig. 5.1).

5.2.2 Convolutional neural networks for spatial representation

The VAE described in the last section may be used with any kind of neural network architecture. In order to be classified as a DGM, however, it must rely on a deep architecture. The term "deep" means that the mapping, in this case a neural network, is actually composed by many layers of functions, which in turn create as many intermediate representations (also known as hidden layers). In other words, mappings are built sequentially where the inputs for the current layer come

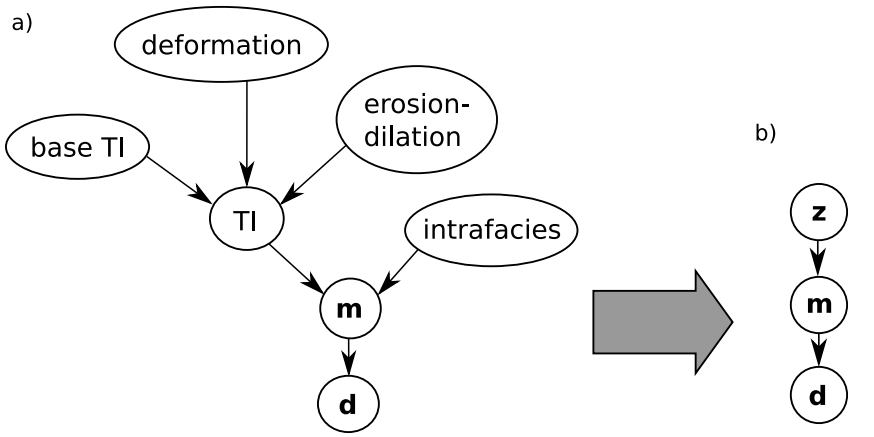


Figure 5.1: Probabilistic graphical models for: (a) the original variables which is used to train the VAE and (b) the latent variables (i.e. including the VAE) which is used for inversion. $\mathbf{m}$, $\mathbf{d}$ and $\mathbf{z}$ refer to the model, data and latent vectors, respectively.

from the outputs of the previous layer (the first layer being the original input). In general, neural networks tend to work better if features from the original inputs are first constructed. The intermediate mappings of a deep neural network may be seen as progressively building more high-level features avoiding the need for tailored feature extraction (Bengio, 2009; Goodfellow et al., 2016).

Convolutional neural networks (CNNs) are one of the most widely used deep architectures for images (LeCun et al., 2015). They provide a very general template which is able to preserve spatial information of the inputs. They do so by defining each layer mapping as a convolution with a set of kernels (or filters), where the output of the mapping may be seen as a stack whose number of channels (or levels) is equal to the number of filters used. The weights of the filters are actually the weights of the neural network, and therefore are estimated during training. In this architecture, while the input may have only one channel (i.e. it is a one level stack) the intermediate representations have several (depending on the number of filters of the previous convolutional layer) therefore the kernels that are applied to them have a corresponding number of channels (i.e. the convolu-

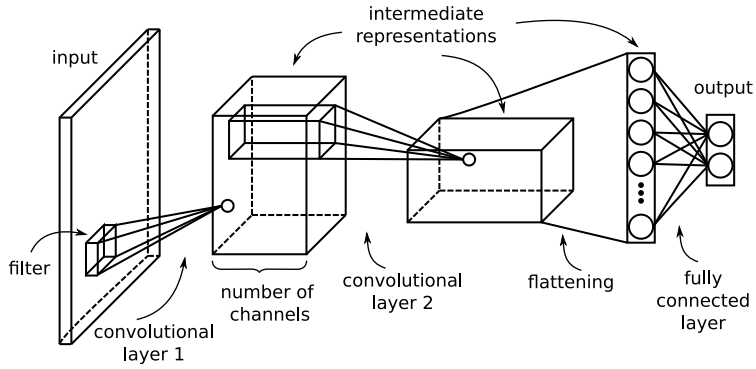


Figure 5.2: A sketch of a CNN depicting the general architecture and its main components.

tion is applied across all channels). It has been shown that this architecture allows neurons or units in the first layers to mostly represent simple features (such as single edges) while units in subsequent layers may represent more complex features (corners or gridded patterns) (Zeiler and Fergus, 2013). CNNs usually contain fully-connected layers which serve as a bottleneck in the architecture. A sketch of a CNN with two convolutional layers and a fully-connected layer (applied after flattening the output of the second convolutional layer) is shown in Fig. 5.2 where the general architecture and its main components are highlighted.

In this work we consider a VAE in which both encoder and decoder (see Fig. 3.2) are based on CNNs. The encoder has an architecture similar to that of Fig. 5.2 (but with different number of layers, size of kernels, and so on) while the decoder has a reversed architecture (as would be obtained by flipping from left to right the one in Fig. 5.2). An important parameter that controls the complexity (and diversity) of the patterns is the dimensionality n of the latent vector. In this work, $n = 40$ was chosen by testing a set of increasing values ($n = 20, 40$ and 60) whose range was based on previous studies for similar patterns (see Chapter 3) and selecting the one that provides accurate reconstruction of the training samples without degrading the similarity of the generated patterns (this was assessed by

visualizing a set of randomly generated models). We found, e.g. that $n = 60$ provides only a slight improvement in reconstruction of the training samples but causes a significant degradation of generated patterns.

5.2.3 Inversion of traveltime data using a VAE as prior

As mentioned above, a VAE using CNNs provides a powerful tool to represent complex probability distributions. Therefore if one has a large dataset containing examples of spatial patterns, the VAE allows to approximate complex prior probability distributions in the context of geophysical inversion. Following the derivation of Section 3.2.2, inversion is done by minimizing the objective function in Eq. 3.4 with $R(\mathbf{z}) = \|\mathbf{z}\|^2$ (Bora et al., 2017) and whose gradient is computed according to Eqs. 3.6 and 3.25 as:

$$\nabla_{\mathbf{z}}\zeta(\mathbf{z}) = -\mathbf{S}(\mathbf{z})^T(\,\mathbf{J}(\mathbf{m})^T(\mathbf{d} - \mathbf{f}(\mathbf{m}))\,) + 2\lambda\mathbf{z} \tag{5.1}$$

In this Chapter, we illustrate the proposed approach with a cross-borehole GPR traveltime field dataset. In order to approximate the propagation of waves, a forward operator that relies on the eikonal equation:

$$|\nabla\tau|^2 = v^{-2} \tag{5.2}$$

is used, where τ denotes the traveltime and v is the velocity of the subsurface materials. A numerical solution is typically required, where after discretization one obtains the forward operator that relates the vector of traveltimes $\mathbf{d} = \boldsymbol{\tau}$ to the slowness (which is the reciprocal of velocity) vector $\mathbf{m} = \mathbf{v}^{-1}$ in Eq. A.1. The Fast-Marching method and a factorized version of the eikonal equation are used herein (Treister and Haber, 2016). The factorized equation helps to reduce the error induced by spatial discretization in the proximity of the sources. It is important to note that this forward operator may still result in noticeable error when used for field data since effects related to the finite-frequency or scattering are not considered. When a proper discretization is chosen and a moderate velocity contrast is assumed, the magnitude of this error is comparable to the one of measurement error (Hansen et al., 2014) which should allow for data misfit error

only a bit higher than with more realistic operators. Though, a non-negligible bias remains which must be considered when analyzing inversion results. The same implementation allows one to efficiently compute the product $\mathbf{J}(\mathbf{m})^T(\mathbf{d}-\mathbf{f}(\mathbf{m}))$ which is given by the solution of a triangular system exploiting the Fast-Marching sort order of the forward operator (Treister and Haber, 2016). The choice of such forward operator is motivated by the need to keep computational demand low, as inversions usually require a significant amount of both forward simulations and the above sensitivity product.

In contrast to previous studies where synthetic cases assumed that the mean velocity values in each facies were known (Laloy et al., 2017, 2018; Mosser et al., 2018; Canchumuni et al., 2019), here the inversion of these velocity (or slowness) values is done by assuming a linear model that shifts and scales the spatial models obtained from the VAE according to $\mathbf{v} = w_1 + w_2\,\mathbf{m}$. This is helpful for field cases since typically there is uncertainty in these values. The inversion will then include two extra parameters (w_1 and w_2). If these parameters are assumed independent of the latent vector $\mathbf{z}$, one may compute the gradient of the objective function with respect to them:

$$\frac{\partial \zeta(\mathbf{w})}{\partial w_i} = \nabla_{\mathbf{v}}\gamma(\mathbf{v})\frac{\partial \mathbf{v}}{\partial w_i} \tag{5.3}$$

where $\nabla_{\mathbf{v}}\gamma(\mathbf{v})$ is given by Eq. 3.25 but computed using the values of $\mathbf{v}$ instead of $\mathbf{x}$. For the two w_i parameters we have:

$$\frac{\partial \mathbf{v}}{\partial w_1} = \mathbf{1},\ \frac{\partial \mathbf{v}}{\partial w_2} = \mathbf{m} \tag{5.4}$$

Similarly, the first term on the right of Eq. 3.6 should now be computed using $\mathbf{v}$ instead of $\mathbf{m}$. Strictly, this term should also include a derivative with respect to $\mathbf{m}$, however, this is a constant and it has no impact since the step size of the optimization is scaled in every iteration. Since these two parameters cause a stronger impact on traveltime values than the latent variables, their step is multiplied by a factor equal to 10^{-4} to make the inversion stable.

In this Chapter, stochastic gradient descent (SGD) and Eq. 5.1 are used for optimization of the objective function (Lopez-Alvis et al., 2020). SGD provides

two main advantages: (1) it is less prone to get trapped in local minima, especially if the objective function has the shape of a global basin of attraction, and (2) the computational cost of each iteration is reduced by only simulating a subset of the data (also called a data batch). Decreasing of the step size (or learning rate) is also employed as it has been shown to further help in reaching the neighborhood of the global minimum (Kleinberg et al., 2018).

5.2.4 Checking the prior consistency

When the inversion described above is applied to a field case, it is important to check that the chosen prior is consistent with the data (Scheidt et al., 2018). Further, when considering an assembled prior, this check may allow to falsify some of the patterns before training the VAE, potentially improving the accuracy of the generated patterns and/or allowing for a lower dimensionality to be used for the latent space. This prior consistency or falsification step is done using the original generative model (Fig. 5.1a). The method applied here relies on approximating the marginal conditional distribution with respect to the TI as $p(\mathbf{d}|TI) \approx p(\mathbf{d}^*|TI)$ where $\mathbf{d}^*$ refers to a lower-dimensional or compressed version of the data $\mathbf{d}$. Here, a number of samples from each TI and their corresponding simulations (using the forward operator) are obtained, then principal component analysis (PCA) is used to perform the dimensionality reduction. The conditional $p(\mathbf{d}^*|TI)$ is then approximated with adaptive kernel density estimation (KDE) (Park et al., 2013). Finally, the value of $p(\mathbf{d}^*|TI)$ at the observed data is compared to the probability density value at the 99 percent confidence region of a multivariate Gaussian distribution with the same dimension as $\mathbf{d}^*$. If the density value at the observed data is lower than the density value of the multivariate Gaussian, the TI is falsified or deemed inconsistent with the data. As recommended in Chapter 4, when you do not have insight in to how the different structural parameters will impact the data, the best strategy is to rely on data-driven dimensionality reduction. Since here, different factors (erosion/dilation, intrafacies, deformation) generate variability between the different TIs, it is difficult to come up with insight-driven ways to reduce the data dimensionality therefore PCA, as a data-driven strategy, was used.

The use of the conditional $p(\mathbf{d}|TI)$ in this chapter is done to define a quantitative threshold for inconsistent TIs, whereas in Chapter 4 it is assumed that data is in the range of the structural parameter (e.g. it was necessarily generated by one of the three TIs considered which means a uniform prior $p(TI)$ is assumed). The issue with the latter case for checking consistency is that it will provide a conditional probability $p(TI|\mathbf{d})$ that integrates to one (e.g. $p(TI_1|\mathbf{d}) + p(TI_2|\mathbf{d}) + p(TI_3|\mathbf{d}) = 1$) even if the measured data is in a low probability density zone. So, in the case the measured data lands in low density zones for all the TIs, one would have no way of knowing that probably all the TIs should be discarded. In contrast, the conditional $p(\mathbf{d}|TI)$ provides a quantity that will depend on the probability density in the position of the observed data point and therefore it is possible to define a threshold by comparing with e.g. the confidence region of a normal distribution with the same dimensionality (as proposed above). The difference lies in the fact that the procedure in Chapter 4 is really a prior updating step where the one in this chapter is simply a prior consistency check.

5.2.5 Training VAE with realistic patterns based on an outcrop

The size of the spatial domain to be modeled was selected according to the region sensed by the acquisition setup (see details below). A uniform cell discretization of 5 cm was chosen to model high-resolution details. Although CNNs may be set to the desired dimensions by selecting the correct size for the filters, stride and padding, one could also consider a slightly larger size and then crop the cells outside the domain since they do not affect the data misfit. In this Chapter, some cells close to the surface are retained even if they are outside the sensed volume because they allow a qualitative assessment of the effect of the prior pattern information in the absence of data. Therefore, the spatial domain was discretized by $65 \times 129 = 8385$ cells, corresponding to a 3.25 m $\times$ 6.45 m section.

The training patterns used to train the VAE are constructed by a hierarchical model that allows for the transformation of an initial set of TIs (Fig. 5.1a). The sensed subsurface was assumed to be mainly composed by two different materials: till and sand. Two initial object-based TIs (BTI_1 and BTI_2) were built according to information on local geology and a quantitative analysis of an outcrop close

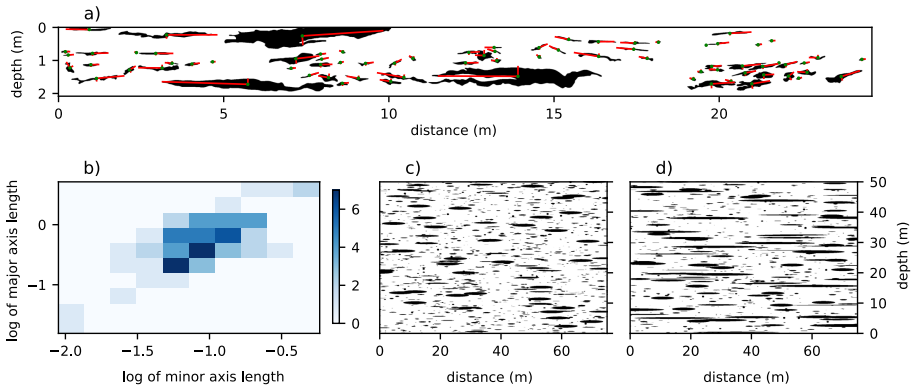


Figure 5.3: (a) Digitized outcrop from Kessler et al. (2013) showing sand bodies in black, background till in white, the axes of fitted ellipses for the sand bodies in red and centers of the ellipses in green. (b) Two-dimensional histogram of the major and minor axes lengths of the ellipses fitted in the outcrop. 1500×1000 pixel croppings of base TIs: (c) BTI_1 and (d) BTI_2.

to the investigated cross-borehole section (Kessler et al., 2013). These two TIs were mainly chosen because there is uncertainty in the presence of sand sheets (the most elongated sand bodies) in the sensed region: they were not present in the outcrop used in the analysis but they were present in other outcrops. All of the sand bodies were assumed to be approximated with ellipses of different sizes and eccentricity (Fig. 5.3a). For this, the statistical distribution of the major and minor axes of the sand bodies was approximated from the outcrop by a two-dimensional histogram (Fig. 5.3b). Then, BTI_1 is directly constructed by sampling ellipses sizes according to the histogram, placing them randomly in the domain (overlapping is allowed to partially account for the more complex shapes) while maintaining a facies proportion similar to the one in the outcrop which is 0.17 (Fig. 5.3c). BTI_2 is built similarly but includes the sand sheets (Fig. 5.3d) for which the size distribution was based on the one reported by Kessler et al. (2012). The size of these TIs was chosen in order to include many repetitions of the patterns for the target size to be simulated (65×129), therefore TIs with a size of 4762×4762 are used.

To account for more diverse and realistic shapes for the sand bodies (as those

seen in the outcrop) two main transformations were applied to the initial TIs: erosion/dilation and local deformation. Erosion/dilation here refers to the image morphological operation for which pixels are removed/added to the limits of objects by setting a pixel to the minimum/maximum over all pixels in a neighborhood centered at that pixel (Soille, 2004). Though erosion/dilation may refer to either of the two facies, here we will refer to the ones of the sand bodies to avoid confusion. One step for dilation and one for erosion was done using a neighborhood which is 6×2 pixels. The local deformation was done by a piecewise affine transformation (van der Walt et al., 2014) which requires defining a uniform grid of nodes and a corresponding mesh by Delaunay triangulation. Then, the positions of the nodes were perturbed according to two Gaussian random fields (one for the x- and one for the y-coordinates) and finally a local affine transformation is done to the pixels inside each triangle of the original mesh in order to fit the new deformed mesh. Deformation was applied with two different amplitudes in the perturbation of the grid, resulting in two different levels of deformation. Considering all the combinations of erosion-dilation and deformation (including the ones with no erosion-dilation and zero deformation) a total of nine different cases or modified TIs for each base TI are built. The patterns of each of the nine modified TIs obtained from BTI_2 are shown in Fig. 5.4. The size of each of these modified TIs is a bit smaller (4722×4722) than for the base TIs since cropping was needed in the edges after deformation.

Finally, intrafacies variability was considered by means of using Gaussian field simulations with different means and anisotropy for each facies: they both use a Gaussian covariance function with correlation length of 1.0 m but the channels facies uses an anisotropy factor of 0.2 and a mean of 0.35 (prior to transforming to velocity values) while the background facies uses a factor 0.25 and a mean of 0.7. This was done following a "cookie cutter" approach where each of the simulations is only set in pixels with the corresponding facies value. Values were log-transformed in order to prevent negative values. This step is done after the sample is cropped from the modified TI to train the VAE to allow more variability in the patterns. The overall hierarchical model from where training samples for the VAE are taken is shown in Fig. 5.1a.

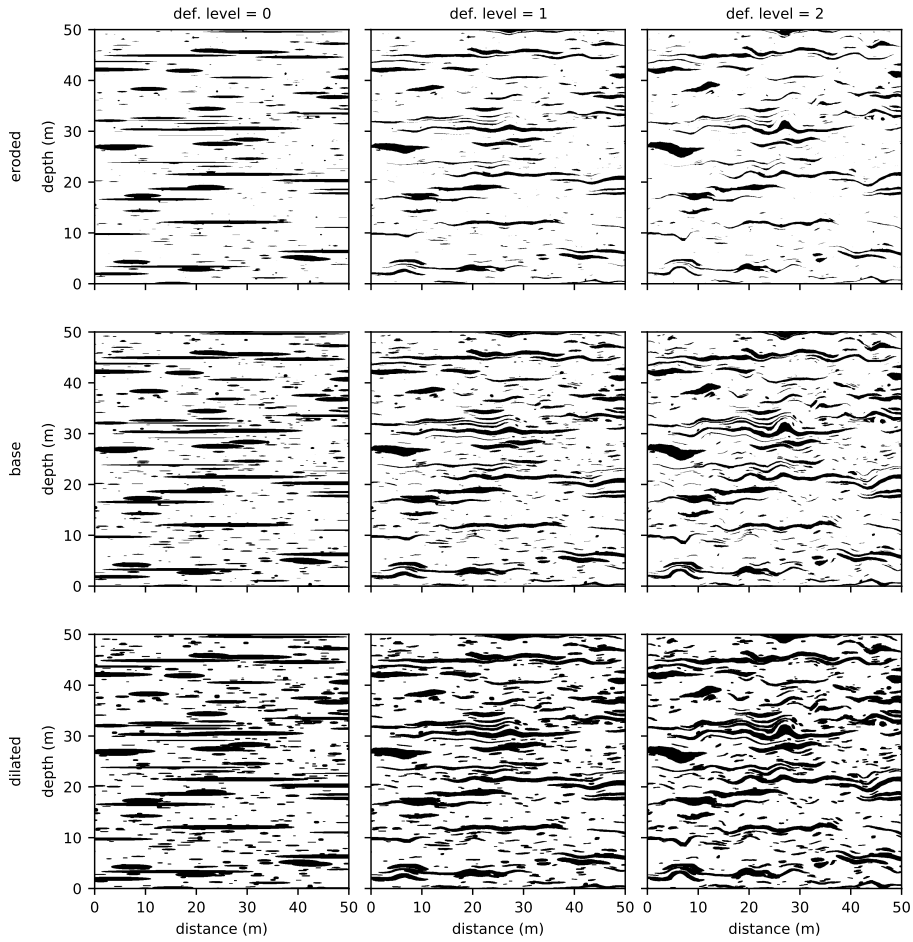


Figure 5.4: 1000 × 1000 croppings of each of the nine modified TIs for the base TI including the sand sheets (BTI_2).

5.2.6 Field site and data description

The field site is located at the Kallerup gravel pit, Denmark. The local geology is composed by a glacial till with several elongated sand bodies embedded (Kessler et al., 2012). Till is composed by particle sizes from clay to gravel, while sand bodies have a more narrow grain size distribution. Further, shapes of the sand bodies display varying degrees of deformation characteristic of basal till. This type of geology results in highly contrasting subsurface, as may be seen in Fig. 5.3a. After the data was acquired, the field site was excavated which allows to compare with inversion results, at least qualitatively (Larsen et al., 2016; Bording et al., 2019).

The field dataset is the cross-borehole traveltime data presented by Looms et al. (2018). Measurements were collected with 100 MHz borehole antennas and a PulseEKKO system (Sensors & Software, ON, Canada). The two boreholes are located 3.25 m apart and are 8 m deep. Data was acquired forming a multi-offset gather (MOG) with all source positions in one borehole and receiver positions in the other. Spacing for both sources and receivers was 0.25 m and data was collected from 1.0 m to 7.0 m deep, for a total of 625 traces. First arrivals were picked with a semi-automatic procedure (Looms et al., 2018). Data for sources and receivers with depth less than 1.5 m were removed to avoid error from refraction at the air-ground interface. For similar reasons, since the boreholes are located in the unsaturated zone, data offsets with angles > 30 degrees were not considered to avoid error from borehole refraction. Estimated measurement error is 0.47 ns while average traveltime is 41.5 ns.

Although in this work the full waveform data was not considered, it is important to mention that there is further information content in such data that may be exploited to give a more constrained characterization of the subsurface. For instance, Looms et al. (2018) present inversion results of full-waveform data that provide also a distribution of (electrical) conductivity. This allows for certain structures to be identified even if they do not have a contrasting permittivity, e.g. their inverted model (Figure 2 in Looms et al., 2018) shows a region at about 5 m depth with slightly higher conductivity than the background while there is no noticeable contrast in permittivity for the same region.

To assess the performance of our proposed inversion, a synthetic case is first analyzed with the same acquisition settings than those of the field data. A synthetic model was built with the same statistical distribution of BTI_2 but with a higher proportion of sand to till proportion (0.32) and different degree of deformation (an amplitude just in the middle between 1 and 2 in Fig. 5.4). The model was cropped from a TI of the same size as the ones used for training but its random spatial realization was different, i.e. the ellipses and its positions were randomly set, therefore one should expect different patterns may be present than those in the TI used for training. Then, synthetic data was generated using the forward operator and Gaussian noise with the same magnitude as the error estimated for the field data was added (0.47 ns). Note that in this case, there is no error due to the forward operator. In this way, the synthetic case should provide an idea of how performant is the inversion with VAE in obtaining patterns that deviate from the ones used for training.

5.3 Results

5.3.1 Training the VAE and prior consistency check

The VAE for the assembled prior is trained by randomly selecting from any of the 18 modified TIs, then randomly sampling a cropped piece (with the appropriate size of the spatial domain) and adding the intrafacies variability. Examples of the cropped samples are shown in Fig. 5.5a. Note that the color scale is with respect to the model variable $\mathbf{m}$ but prior to its transformation to velocity values. The VAE was implemented and trained using PyTorch (Paszke et al., 2017). The training used a total of $P = 10^7$ cropped samples and took around ~ 4.5 hrs on a Nvidia GPU RTX 2060 (~ 3 hrs without the intrafacies). Note that deformation and erosion-dilation may have been done directly while feeding the samples to train the VAE (similar to the intrafacies), however, this would have likely resulted in prohibiting computational time (while erosion-dilation is typically fast, the local deformation is generally much slower). Once trained, samples are generated according to the graphical model in Fig. 5.1b (following the process defined by Fig. 3.2). A few examples of random samples generated from the trained VAE

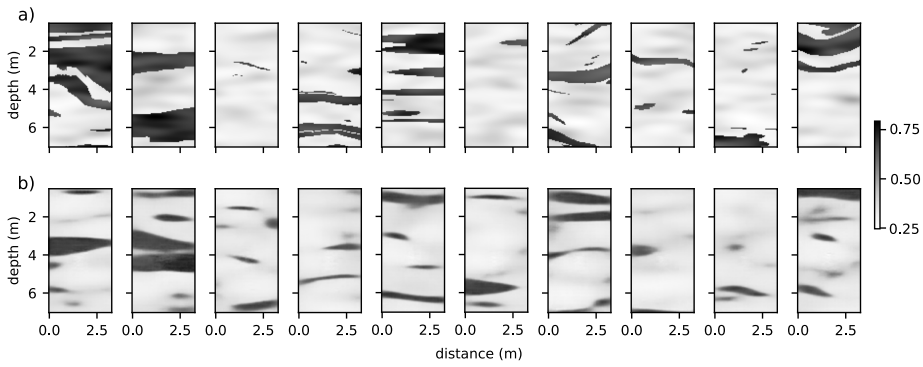


Figure 5.5: Examples of training samples (a) and samples generated from the trained VAE (b). The colorscale is with respect to the model variable $\mathbf{m}$ prior to its transformation to velocity values.

are shown in Fig. 5.5b, these are samples from the assembled prior distribution approximated by the VAE. Also, a VAE is trained for each individual TI to make a comparison with the assembled prior.

The prior consistency check is performed for both the synthetic and field data. For this, 300 model samples (generated as in Fig. 5.1a) and their corresponding forward simulations are obtained for each training image. Then, the first three PCA components of these simulations and the data are used to compute the value of $p(\mathbf{d}^*|TI)$. The first three components were considered because they account for about 84 percent of data variability (explained variance). The density value at the contour of the 99 percent confidence region of a three-dimensional multivariate Gaussian distribution is equal to 2.2×10^{-4}, so any TI with a conditional density value lower than this is deemed non-consistent or very unlikely to have generated the data. Fig. 5.6a,c shows the $p(\mathbf{d}^*|TI)$ for each TI. For both the synthetic case and the field case, all TIs show a conditional probability above the defined threshold, i.e. none of the TIs is falsified. Note that for the field data, TIs 3 and 5 are very close to the threshold. An additional visual check for these two TIs is performed by plotting of the data point together with the simulated data points (not shown), which confirms that the data point is in a low density region but it is still likely to be produced by each of the two TIs.

The VAE based generated patterns may fail to adequatly represent the pat-

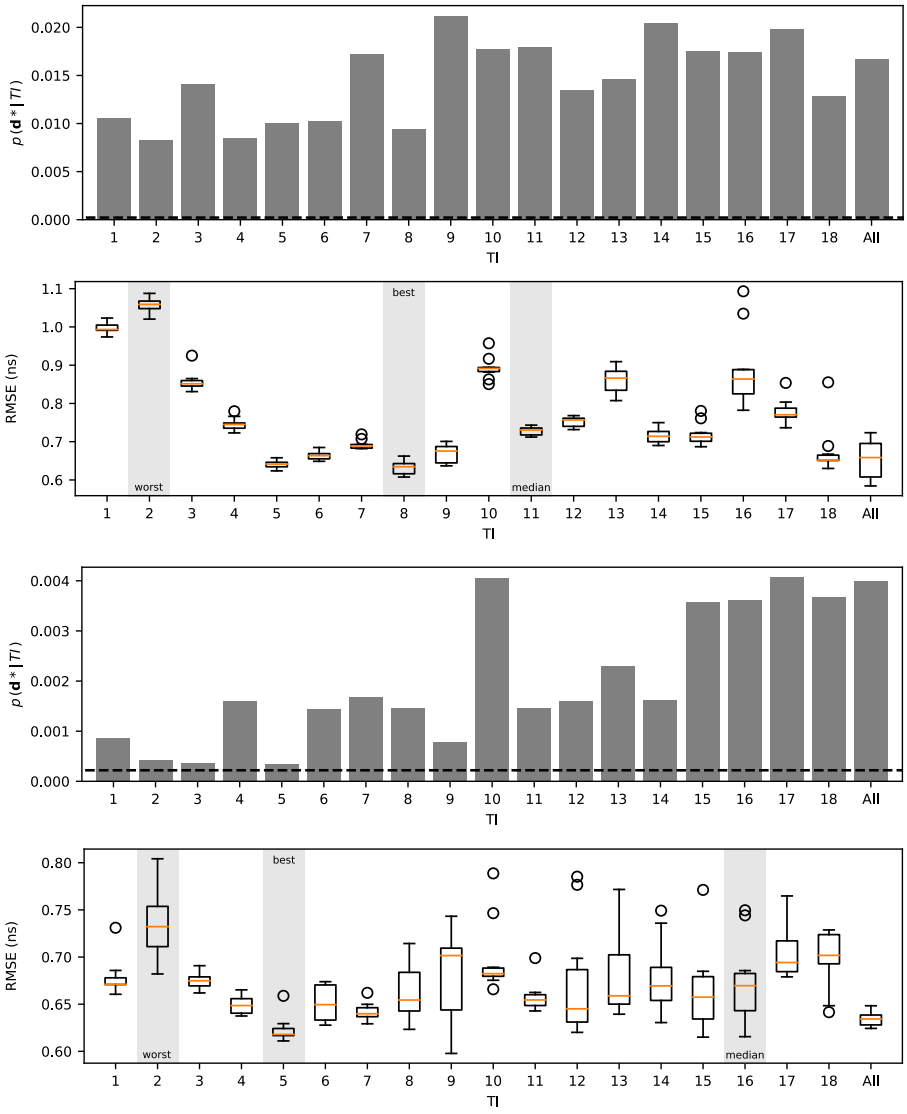


Figure 5.6: Prior falsification results and RMSE boxplots for synthetic (a,b) and field case (c,d) for individual priors (VAEs trained on each of the 18 TIs) and the assembled prior (labeled "All"). Dashed line in (a) and (c) is the threshold for falsification. Shaded areas in (b) and (d) indicate best, median and worst performing individual prior in terms of mean data RMSE.

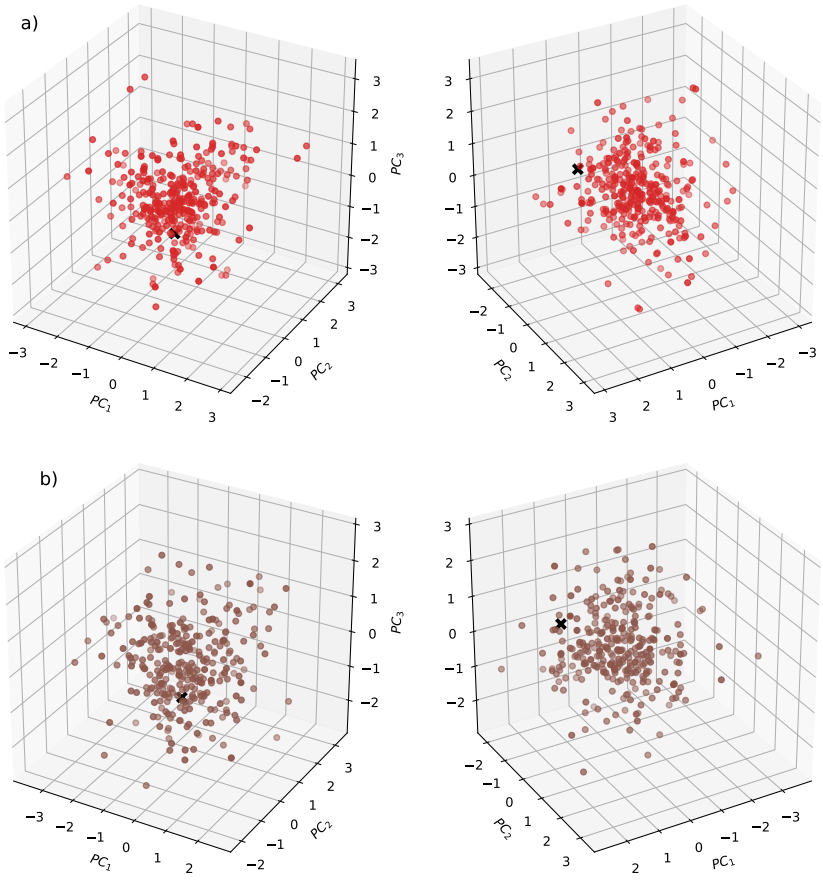


Figure 5.7: Principal components of simulated data and field data for TI_3 (a) and TI_5 (b). Simulated data is in colored dots and field data is denoted by the '$\times$' symbol.

terns of heterogeneity encountered in the field for three main reasons: (1) sufficiently similar patterns are not included during training, (2) patterns are filtered or simplified by the VAE, and (3) the diversity of the patterns was not sufficient to simulate new consistent patterns. In general, these three reasons play a role to different degrees. The first is unavoidably present in any study that aims to use information from nearby outcrops or local geology to constrain the subsurface patterns in the sensed domain. However, this may be partially accounted for by considering different base patterns and their perturbed versions (obtained by morphological operations and local transformations) which may all be attributed to a similar environment. Note, however, that this strategy will not add new materials (lithologies). The prior consistency step may indicate if the VAE fails due to the first reason: the ability of the proposed patterns to generate the data may be checked before training the VAE. The effect arising from the second reason is directly related to generative accuracy and is captured in Fig. 5.5 for example, in that the generated samples seem to have filtered out patterns with very high curvature. Finally, the third reason, which is somewhat tied to the first, is related to how the VAE is able to interpolate between training patterns. This may be checked by visualizing a set of training images as in Fig. 5.5 and also making a latent traversal as shown in Fig. 5.8, which makes steps along two of the dimensions of the latent space and fixes the rest. This should also be supplemented by an assessment of how much the generated patterns depart from the training samples while retaining consistent patterns. In recent work (Lopez-Alvis et al. (2020) and Chapter 3) it was shown that VAEs are able to deviate from training patterns while still preserving realistic patterns through breaking continuous channels from the original training image. There have been some recent efforts to quantitatively measure diversity in DGMs (Lucic et al., 2018; Sajjadi et al., 2018) however, it remains an open question whether useful departures (such as the breaking channels) would be adequately captured by these measures. In summary, the proposed approach is not intended to generate perfectly accurate patterns but to allow the generated patterns to deviate from training patterns in order to both improve diversity and fit the data without compromising the patterns' realism.

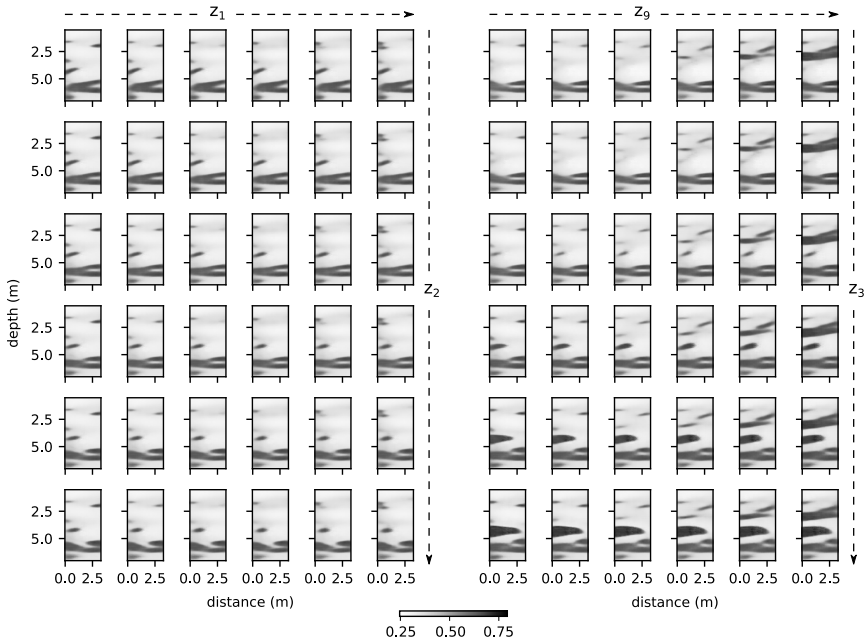


Figure 5.8: Examples of VAE latent traversals (stepping in two latent dimensions while keeping the rest fixed) for: latent dimensions z_1 and z_2 (left) and latent dimensions z_9 and z_3 (right). The colorscale is with respect to the model variable $\mathbf{m}$ prior to its transformation to velocity values.

5.3.2 SGD-based inversion of synthetic data with VAE as prior

Once the VAE is trained, the assembled prior may be used directly in inversion to impose the diverse patterns. It is worth noticing that the latent parameters $\mathbf{z}$ of the VAE have effectively substituted the parameters related to the original hierarchical model (the substitution is denoted by the grey arrow in Fig. 5.1). The latent parameter distribution now includes all the discrete and intractable operations (i.e. different base TIs, erosion-dilation, deformation and intrafacies variability) in a continuous and searchable space. This allows for optimization to be performed by continuously stepping in the latent space. Moreover, such steps can take advantage of the gradient (as detailed in Sec. 5.2.3) which generally would not be the case if one sought to directly estimate the original parameters.

The results of our proposed inversion approach are first assessed using the synthetic data presented above. Fig. 5.9a,b,c shows the real synthetic model, an inverted model with traditional smooth regularization and a VAE inverted model (for one randomly chosen starting model). The smooth inversion is done with a low regularization factor (10^{-9}), so it mainly represents the information content of the data and therefore is prone to artefacts due to noise (e.g. ray artefacts in Fig. 5.9b). In contrast, due to the use of strong prior information, the VAE inverted model is artefact-free (note that this is usually also the case for inversion using MPS). For the model in Fig. 5.9c, the behavior of the data misfit (RMSE), the Euclidian distance between the current model and the real model, the norm of $\mathbf{z}$ and the velocity parameters as the inversion progresses are shown in Fig. 5.9d-g, respectively. The norm of $\mathbf{z}$ is useful to check that the algorithm does not diverge from the prior. This is because the prior $p(\mathbf{z})$ is multivariate Gaussian $\mathcal{N}(\mathbf{0}, I_n)$, then models consistent with the prior should not be far from the origin and also models with the most common patterns should be centered according to a χ-distribution with d degrees of freedom. Since we are using SGD which is a stochastic optimization method, inversion is done for 10 different starting models. Inversion results for three different starting models are shown in Fig. 5.9h-j. To assess the impact of the assembled prior compared to VAEs trained on individual TIs, inversion is done also for each of the individual cases. Considering 10 different starting models for each case, the mean and stardard deviation of

data RMSE, norm of $\mathbf{z}$, and velocity parameters are computed. These values are shown in Table 5.1 for the best, median and worst individual TIs in terms of mean RMSE together with those of the assembled prior. Boxplots of the data RMSE for all individual TIs and the assembled prior are shown in Fig. 5.6b. Notice that the mean data RMSE for the assembled prior (0.655) is only slightly higher than the magnitude of the added noise.

To analyze the impact of prior information (as represented by the VAE) on inversion results, one must also consider how much information content is provided by the data, i.e. how much the data constrains the posterior distribution. In this work, the cross-hole traveltime dataset is considered informative enough to produce relatively similar inverted models, however, since a high resolution model is desired, the choice of prior information (and the way it is imposed) still causes noticeable variations in inverted models. The inversion results for the synthetic case in Fig. 5.9c show that although reconstruction is not perfect, the method is able to identify most of the structural characteristics of the real model. The inverted model is noticeably better than traditional smooth inversion (Fig. 5.9b), which shows higher data RMSE and slightly less connected sand bodies and from which it is not possible to identify small features (at 5 m depth in the right and close to 7 m depth on the left in Fig. 5.9a). On the other hand, both inversion methods miss a low velocity structure (at 3 m depth on the left of Fig. 5.9) and most of the intrafacies variability. The VAE-SGD inversion even locally biases the model in order to account for the lack of intrafacies variability (note a more pronounced bend of the lower part of the sand body at $\sim$ 4 m depth to make up for a low velocity intrafacies zone). Most likely this comes from the fact that the model is not exactly within the prior. Since, no error in the forward operator model is introduced for the synthetic case, the RMSE value higher than the noise level indicates that deviations in the inverted model are mainly due to the prior, whose accuracy slightly degrades due to a joint effect of the three reasons mentioned in Sec. 5.3.1. The synthetic case shows that our proposed inversion still provides useful results even when the patterns of the real model differ slightly from those of the TIs used to train the VAE.

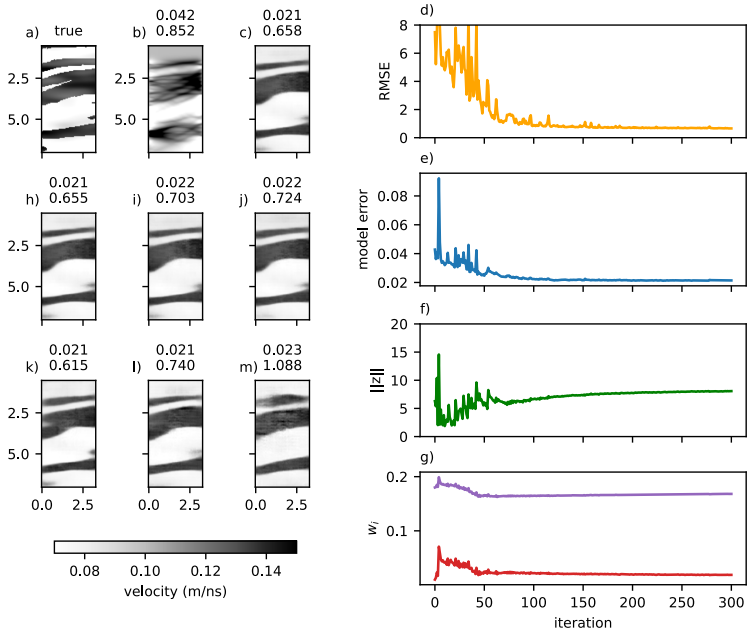


Figure 5.9: Inversion results for the synthetic case: (a) True model, (b) smooth inverted model, (c) VAE-SGD inverted model for one random starting model using the assembled prior. For the model in (c), the values in each iteration for: data RMSE (d), model RMSE (e), norm of $\mathbf{z}$ (f) and linear velocity parameters (g). VAE-SGD inverted models for three different starting models using the assembled prior (h,i,j). VAE-SGD inverted models for prior with individual TIs using one random starting model: best (k), median (l) and worst (m) in terms of RMSE (see Fig. 5.6b). For all inverted models, model RMSE and data RMSE are shown at the top.

TI	data RMSE	$\|\mathbf{z}\|$	v_1	v_2
		synthetic case		
All	0.655 ± 0.050	8.004 ± 0.309	0.017 ± 0.005	0.17 ± 0.007
best	0.632 ± 0.017	7.767 ± 0.124	0.019 ± 0.001	0.166 ± 0.002
median	0.728 ± 0.011	8.325 ± 0.072	0.017 ± 0.001	0.171 ± 0.001
worst	1.058 ± 0.018	10.097 ± 0.326	0.015 ± 0.001	0.175 ± 0.003
		field case		
All	0.634 ± 0.008	5.342 ± 0.244	0.031 ± 0.001	0.157 ± 0.004
best	0.623 ± 0.013	5.194 ± 0.124	0.029 ± 0.001	0.157 ± 0.004
median	0.674 ± 0.041	5.155 ± 0.294	0.033 ± 0.003	0.148 ± 0.010
worst	0.732 ± 0.035	5.371 ± 0.214	0.031 ± 0.001	0.150 ± 0.004

Table 5.1: Mean and standard deviation values of inversions using 10 different initial models. The "TI" column indicates best, median and worst in terms of data RMSE from all 18 TIs.

5.3.3 SGD-based inversion of field data with VAE as prior

Inversion for field data is done similarly to the synthetic case. The smooth regularization inverted model and the VAE inverted model are shown in Fig. 5.10a and b, respectively. The behavior of RMSE, norm of $\mathbf{z}$ and velocity parameters during optimization is shown in Fig. 5.10c-e. The RMSE follows a behavior consistent with the chosen SGD scheme: an initial phase with very large oscillations followed by a more stable decreasing behavior. The behavior of the norm of $\mathbf{z}$ indicates that during the initial phase the search covers very large range of radial distances from the origin while for the end it is constrained to small radial changes. VAE inverted models with different initial starting models are shown in Fig. 5.10f-h. Again, to check if assembling the prior from many different TIs is advantageous, we compare it with the results of using the individual TIs. The mean and standard deviation values for the final data RMSE, norm of $\mathbf{z}$, and the velocity parameters are computed from 10 inversions with different initial models (Table 5.1). Boxplots of the data RMSE for inversions with VAEs trained with each of the TIs and the assembled prior are shown in Fig. 5.6d. The models inverted for one starting model with the TIs corresponding to the best, median and worst average data RMSE are shown in Fig. 5.10i-k.

Inversion results for the field data (Fig. 5.10) show a behavior very similar

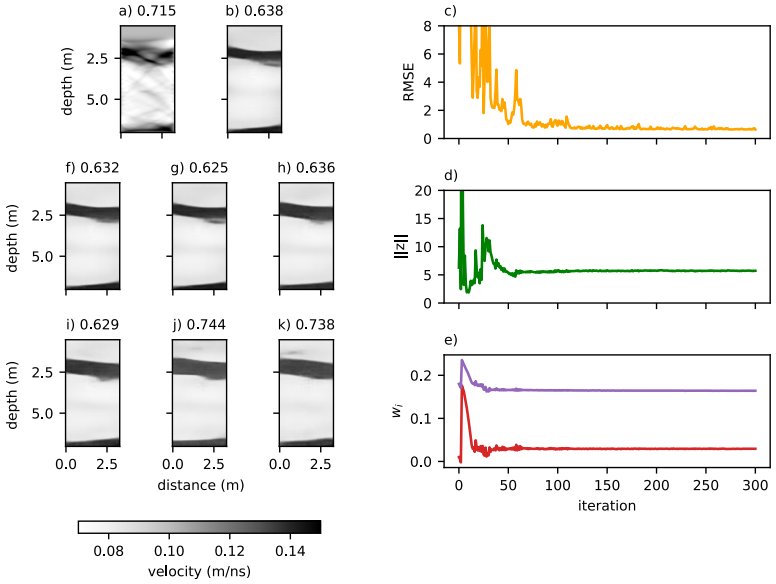


Figure 5.10: Inversion results for the field case: (a) smooth inverted model, (b) VAE-SGD inverted model for one random starting model using the assembled prior. For the model in (b), the values in each iteration for: data RMSE (c), norm of $\mathbf{z}$ (d) and linear velocity parameters (e). VAE-SGD inverted modes for three different starting models using the assembled prior (f,g,h). VAE-SGD inverted models for prior with individual TIs using one random starting model: best (i), median (j) and worst (k) in terms of RMSE (see Fig. 5.6d). For all inverted models, data RMSE is shown at the top.

to the synthetic case. However, the inverted model indicates a simpler structure when compared to the synthetic case. This is consistent with evidence from the excavation and even inclination trends of both the upper sand and lower sand bodies seem to match those observed in excavated profiles close to the GPR sensed domain (Larsen et al., 2016; Bording et al., 2019). Regarding the performance of the assembled prior for inversion, Table 5.1 shows that training the VAE with all the TIs at the same time performs better than the median individual TI and results in approximately equal values of average RMSE compared to inversion with the best individual TI. This indicates that it may be better to build an assembled wide prior than to consider many TIs individually for inversion (Hermans et al., 2015). Note that results of the best individual TI have slightly lower values of RMSE. This may be partially explained by the fact that a constant dimensionality $n = 40$ for the latent vector is used. A better strategy might be to slightly increase n when more diversity in the patterns is considered. Note also that the prior falsification step gives a rather low probability value for the best performing inversion case (see TI_5 in Fig. 5.6c,d). This may be caused by: (1) the low number of samples used for the prior falsification (300 forward runs for each TI) and (2) the enhanced diversity caused by the VAE, i.e. even if the patterns in TI_5 did not produced sufficiently similar patterns to those giving rise to the field data, the VAE trained with this TI does produce such patterns. The assembled prior also has the advantage of a lower computational demand: one does not have to train a VAE and do the inversion for each individual TI. In the presented field case, for instance, the computational demand is 18 times higher if the TIs are considered individually. Moreover, prior uncertainty tends to be larger in field cases therefore a wider prior distribution, such as the one modeled by the VAE with all the TIs, is preferable. This wider prior distribution may indeed help in reducing bias arising when highly informative prior information is used.

It is interesting to contrast the mechanism by which the VAE generates new samples of the patterns to equivalent mechanisms in MPS. While the departure of new patterns from training patterns in a VAE depends mainly in training parameters such as regularization weights α and β which in turn impact the approximation of the continuous prior in model space, MPS may control the diversity of patterns by relaxing the conditioning, e.g. by changing the number of condi-

tioning pixels or by defining distances to the conditioning event. Further study of this relation should enlighten under which circumstances it is better to use either of these strategies to produce more diverse patterns or even if it is possible to combine them to better represent prior uncertainty in the most realistic way possible (see e.g. Bai and Tahmasebi, 2020). On the other hand, the problem of using multiple TIs for MPS seems to have received little attention (Silva and Deutsch, 2012; Scheidt et al., 2016) perhaps because most studies focus on discrete aspects (e.g. different depositional environments) rather than continuous aspects as in this study (i.e. deformation, erosion-dilation and intrafacies variability). In some cases, however, one should be able to frame inversion problems for subsurface models in terms of continuous variables (e.g. two depositional environments may have transitional environments between them), so further study of this subject may prove beneficial.

In this Chapter we considered a normal multivariate Gaussian distribution to model the prior in latent space (i.e. as input to the generative function of the VAE), however, other types of distributions may also be used, e.g. a Gaussian mixture model (Makhzani et al., 2015). These other types of distributions may provide two main advantages: (1) they may produce more accurate patterns, and (2) they are more directly related to the prior distribution in model space and therefore cause less nonlinearity and/or topological changes. However, sampling from these distributions in latent space is not as straightforward as for a multivariate Gaussian. This means that one would have to rely on either different regularization terms in latent space or more advanced (but potentially more computationally demanding) ways of sampling.

5.4 Conclusions

When prior information is expressed by a set of TIs and their perturbed versions, a VAE may be used to approximate a prior distribution that effectively assembles all the possible spatial patterns. The perturbations may include operations such as erosion/dilation, local deformation and intrafacies variability which result in a set of patterns that represent similar geological environments. The VAE is capable of producing patterns that deviate from training patterns but remain realistic,

therefore increasing pattern diversity. The cross-borehole GPR traveltime synthetic case demonstrates that inversion with SGD in the latent space of the VAE is able to obtain a realistic model while remaining computationally efficient. Even though the final misfit is higher than the noise level, most structural features are correctly inverted. By assuming a linear velocity model (two additional parameters), the absolute values of velocity may be also estimated in the inversion. This setting allows for inversion using a VAE as prior to be successfully applied to a field dataset. Results from the field case show a realistic inverted model with misfit only slightly higher than the estimated noise. Moreover, a comparison of VAEs trained on individual TIs and the VAE trained with all the TIs at the same time, shows that the latter performs as good as the best individual TIs but has the advantage of lower computational demand and a more adequate (wider) prior uncertainty, which in turn may reduce bias from highly informative prior information. Finally, future work may include extending the proposed method to handle more general distributions in the latent space or using it in combination with MPS to improve the accuracy and diversity of patterns.

Chapter 6

General discussion and conclusions

In this thesis it has been shown that a compressed representation of complex geological structures and data allows for all relevant information to be preserved in order to perform either inversion or prior falsification.

In Chapter 3, it was shown that an appropriately chosen DGM may be used together with efficient inversion: a VAE with certain values of regularization and SGD-based optimization. In this chapter the induced changes in both curvature and topology of the manifold defined in latent space are identified as the main causes of the nonlinearity of the generative mapping. Moreover, a way to control such nonlinearity through the VAE training parameters is presented which allows gradient-based optimization of an objective function in the latent space.

Chapter 4 presented an objective strategy to select data dimension reduction in order to preserve information related to high-level structural parameters which allows to falsify or update the marginal distribution of such parameters prior to any inversion. Here it was shown that both data-driven and insight-driven dimension reduction are useful for prior falsification of structural parameters, the latter being more easily applied to discrete parameters.

Dimension reduction for both data and model was used in Chapter 5 in order to perform inversion after prior falsification for a field case in complex geological deposits. In this chapter, the methodology of Chapter 3 was further developed

to include perturbations of base patterns of the spatial heterogeneity (such as intrafacies variability) and velocity estimation, a prerequisite for efficient use of the algorithm for field data.

One important contribution of this work was to test and demonstrate the applicability of the proposed methods for both prior falsification and inversion in a complete framework that is validated with field data and benchmarked against traditional inversion. In Chapter 5 such a framework was presented in which a realistic prior was built by assembling different base patterns and perturbing them to resemble patterns in a geological environment described as deformed basal till. The transformations applied to the base patterns included deformation, erosion/dilation and intrafacies variability. These are only a few examples of the possible transformations that may be included with DGM-based priors. A prior falsification or consistency step was first performed in order to check that the defined patterns are consistent with the measured data. Prior distributions were approximated with VAEs for each individual pattern and for all patterns taken together (called assembled prior). More diverse patterns and similar data fit were obtained when comparing SGD-based inversion results of the assembled prior to those of a prior with the pattern of best data fit. In general, this indicates that the framework is useful to obtain models for highly structured subsurface using geophysical data and has sufficient flexibility to image patterns not learnt from the prior, what is a desirable feature for deterministic inversion.

Data driven strategies have been shown to be useful given that sufficient training examples are available. In general, the stronger the assumptions the less training examples are needed, e.g. PCA assumes linear dimensionality reduction and therefore requires a relatively low number of training samples, however, if the underlying manifold is nonlinear some information will be lost with such assumption. Recent computational and algorithmic advances in machine learning have resulted in widespread use of such data driven strategies and placed them as computationally efficient alternatives against traditional methods, such as inversion with MPS. Another advantage of machine learning methods is the explicit representation of the prior probability distribution in latent space, which allows for a more straightforward way to search the model or data spaces. This comes, however, with the restriction that the generative mapping should be moderately

nonlinear, otherwise the latent space will be a very challenging representation of the original spaces and further processing such as inversion or prior falsification will be considerably hindered. This indicates that generally there is a compromise between the accuracy of the representation and the usefulness of such representation for inversion or prior falsification.

These data driven strategies rely on multi-level representations: in DGMs these are built directly during training given the defined architecture while for geophysical data dimension reduction some pre-processing aimed at preserving only the information relevant to certain structural parameters was applied. Such insight driven pre-processing (described in Chapter 4) is necessary since the desired compression is expected to be useful to update the structural parameter marginal distribution and not the model posterior distribution which would generally require all data information content. Interpreting these multi-level representations as probabilistic graphical models shows that dimensionality reduction is equivalent to adding a latent representation to the graphical model. Then, inference is performed on this new latent variable and then the result is mapped back to the original variable if needed. This is a very general procedure and is available to quantify uncertainty at any desired level: in this thesis it was shown for both prior falsification which is applied at higher levels (global parameters) and for inversion which is applied at lower levels (local parameters).

It is interesting to note that the understanding gained by an in-depth review of the conceptual framework of DGMs with inversion (Chapter 3) allowed to both support and propose a framework that is applicable to field data (Chapter 5). Indeed, results indicate that inversion based on a VAE with appropriate training parameters allowed for more diverse patterns which in turn helped to obtain realistic models that fit the data. Failing to recognize that this diversity is important for field data and proposing a framework that does not account for this could lead to biased models.

6.1 Outlook for future work

A major aim of this work was to advance the state-of-the-art in the topic of inversion with DGMs. It was only in recent years when deep learning started to be

applicable using modern GPUs that inversion with DGMS was seen as a viable alternative to inversion with MPS (Laloy et al., 2017). Indeed, as mentioned in the Introduction, while some studies have presented synthetic cases of inversion using DGMs, many important concepts that impact performance of inversion still need to be studied in more detail. As an attempt to improve our understanding of inversion with DGMs, the role of different training parameters and the reasons why these parameters may be impacting inversion was discussed in depth in this work (Chapter 3). While the machine learning community is already tackling some of these concepts (e.g. Bora et al., 2017; Naitzat et al., 2020), the geosciences community should also engage in this process since subsurface datasets may involve both new challenges and opportunities.

An important extension to the presented work would be the estimation of full the posterior and not only the maximum likelihood values through probabilistic inversion; notice that controlling the nonlinearity in the generative mapping benefits not only maximum likelihood estimations but also Monte Carlo methods such as Markov Chain Monte Carlo (Laloy et al., 2017, 2018; Mosser et al., 2018) because e.g. a uni-modal distribution is less likely to be represented by a multi-modal distribution in latent space. However, the more exhaustive exploration of the model space required for such methods may still represent a major limitation for problems where the evaluation of the forward operator is computationally expensive.

Recent work in modeling and sampling conditional distributions with DGMs may be useful for both prior falsification and inversion (Kingma et al., 2014; Engel et al., 2018). One may, for instance, fix a certain latent dimension that is related to a desired feature (e.g. a structure in the center of the domain) and then perform inversion with this conditional prior distribution.

Future work may also include the use of DGMs together with other geostatistical methods such as MPS in order to improve conditioning to direct observations of the subsurface materials (Bai and Tahmasebi, 2020). This may also help to increase the mechanisms to generate more diverse samples. One may for example, change the size of the MPS simulation template to allow samples to deviate from the pattern in the training images.

Regarding the architecture of DNNs, while convolutional neural networks

have been remarkably useful for preserving spatial information, some recent work in neural networks using so called transformer architectures have shown potential for both natural language processing and images (Parmar et al., 2018). This architecture is based on a mechanism called self-attention and provides a different generative mapping that may be more useful for inversion.

Although in this work cross-hole GPR data was considered to benchmark the proposed methods, it is important to highlight that these methods are general and may be applied with any other type of geophysical data. In this way, e.g. one could exploit the higher information content in full-waveform data to further constrain the subsurface and/or identify structures that have a contrasting conductivity even if they have similar permittivity. Some of the methods might require tuning when applied to other types of data (e.g. the selection of insight-driven features in Chapter 4) but others are general enough that would only require changing the forward model and possibly the data error model (e.g. the SGD inversion with VAE in Chapter 3).

Appendix A

Bayesian inversion with VAE

Following a Bayesian approach, inversion may be considered as the conjunction of information regarding the model, the measured data and their relation given by a forward operator (Tarantola and Valette, 1982). The latter relation may be expressed as:

$$\mathbf{d} = \mathbf{f}(\mathbf{m}) \tag{A.1}$$

where $\mathbf{d}$ is a Q-dimensional vector representing the data and $\mathbf{f} : \mathbb{R}^N \rightarrow \mathbb{R}^Q$ is the geophysical forward operator. Since both the measurements and the forward operator typically have some error, the relation in Eq. A.1 may be represented with a conditional probability distribution $p(\mathbf{d}|\mathbf{m})$. Then, inversion is stated as:

$$p(\mathbf{m}|\mathbf{d}) = k\, p(\mathbf{d}|\mathbf{m})\, p(\mathbf{m}) \tag{A.2}$$

where $p(\mathbf{m}|\mathbf{d})$ is the posterior distribution, $p(\mathbf{m})$ is the model prior distribution, $p(\mathbf{d}|\mathbf{m})$ is termed the likelihood function and k is a proportionality constant.

When the prior distribution is approximated with a VAE, inversion may be restated in terms of the latent vector $\mathbf{z}$ as:

$$
\begin{aligned}
p(\mathbf{m}, \mathbf{z}|\mathbf{d}) &= k\, p(\mathbf{d}|\mathbf{m})\, p(\mathbf{z})\, p(\mathbf{m}|\mathbf{z}) \\
p(\mathbf{z}|\mathbf{d}) &= k\, p(\mathbf{z}) \int p(\mathbf{d}|\mathbf{m})\, p(\mathbf{m}|\mathbf{z})\, d\mathbf{m}
\end{aligned} \tag{A.3}
$$

where $p(\mathbf{z})$ is the latent prior distribution and $p(\mathbf{m}|\mathbf{z})$ is the generative mapping (or decoder), as defined in Section 5.2.1. Further, as mentioned above when only considering the mean of the decoder then $p(\mathbf{m}|\mathbf{z}) = \delta(\mathbf{m} - \mathbf{g}(\mathbf{z}))$ and Eq. A.3 may be written as:

$$
\begin{aligned}
p(\mathbf{z}|\mathbf{d}) &= k\, p(\mathbf{z}) \int p(\mathbf{d}|\mathbf{m})\, \delta(\mathbf{m} - \mathbf{g}(\mathbf{z}))\, d\mathbf{m} \\
&= k\, p(\mathbf{z})\, p(\, \mathbf{d}|\mathbf{g}(\mathbf{z})\,)
\end{aligned} \tag{A.4}
$$

Eq. A.4 may be used to solve an inverse problem in which a VAE (or some other DGM) is used to state the prior model distribution. For instance, one may apply Markov chain Monte Carlo to Eq. A.4 and get the posterior distribution of the latent variables (Laloy et al., 2017, 2018). When appropriate values to train the VAE are used (see Section 5.2.1), $\mathbf{g}$ is expected to be only mildly nonlinear. If we further assume that $\mathbf{f}$ is also mildly nonlinear and that errors in the data (with respect to forward predictions) are independent and Gaussian, the likelihood $p(\, \mathbf{d}|\mathbf{g}(\mathbf{z})\,)$ will be approximately independent and Gaussian (Holm-Jensen and Hansen, 2019). Given these conditions, minimizing the objective function $\zeta(\mathbf{z})$ in Eq. 3.4 should provide a good approximation for maximum likelihood model parameters.

Appendix B

Adaptive kernel density estimation

The standard (non-adaptive) equation for kernel density estimation that would apply for our case is (Scheidt et al., 2018)

$$p(s|h(\mathbf{d}_{obs})) = \frac{p(s, h(\mathbf{d}_{obs}))}{p(\mathbf{d}_{obs})} = \frac{\sum_{j=1}^{N} K_{H_s}(s - s_j) K_{H_h}(h(\mathbf{d}_{obs}) - h(\mathbf{d}_j))}{\sum_{j=1}^{N} K_{H_h}(h(\mathbf{d}_{obs}) - h(\mathbf{d}_j))} \tag{B.1}$$

where the involved variables are the same as in Eq. (4.2) but here no clustering is defined, therefore no separate summation for each cluster is needed and the bandwidths H_s and H_h for the scaled kernel functions are the same for all the N Monte Carlo samples. The expected value of Eq. (B.1) is also referred to as the Nadaraya-Watson model or kernel regression (Bishop, 2006).

In general, the bandwidth H refers to the width of the kernel that is used to approximate the distributions and for the multivariate case it is a $Q \times Q$ matrix, where Q is the number of dimensions of the variable. Different kernel functions may be used to do this approximation (Silverman, 1986), in our case we chose the multivariate independent Gaussian kernel.

$$K_H(\mathbf{x}) = (2\pi)^{-Q/2} |H|^{-1/2} e^{-\frac{1}{2}\mathbf{x}^T H^{-1}\mathbf{x}} \tag{B.2}$$

where Q is the number of dimensions of $\mathbf{x}$ and H is a diagonal matrix. As suggested by Park et al. (2013) and Scheidt et al. (2015b), we used clustering in order to make the KDE bandwidth H adaptive. This requires the specification of the number N_c of clusters and results in narrow bandwidths where the density of points is high and wide bandwidths where density is low. We used k-means clustering on the feature space and each sample is assigned a bandwidth H for both the features and the structural parameter according to which cluster it belongs to. The value of the bandwidth H (a diagonal matrix) within each cluster is computed by means of Silverman's rule of thumb (Silverman, 1986) as

$$(H_{ii})^{1/2} = \frac{4}{Q+2}^{\frac{1}{Q+4}} n^{\frac{-1}{Q+4}} \sigma_i \tag{B.3}$$

where n denotes the number of samples and may be different for each cluster, and σ_i is the standard deviation in the i-th dimension in the same cluster. In this way, the control on the bandwidth is implicit on the number of clusters N_c. Applying KDE with this adaptive approach is expressed in Eq. (4.2). There H_s and H_h are computed using the same clusters and have dimensions 1×1 and $N_h \times N_h$, respectively.

Bibliography

Armstrong, M., Galli, A., Beucher, H., Loc'h, G., Renard, D., Doligez, B., Eschard, R., and Geffroy, F. (2011). *Plurigaussian Simulations in Geosciences*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Arnold, D., Demyanov, V., Rojas, T., and Christie, M. (2019). Uncertainty Quantification in Reservoir Prediction: Part 1—Model Realism in History Matching Using Geological Prior Definitions. *Mathematical Geosciences*, 51(2):209–240.

Arvanitidis, G., Hansen, L. K., and Hauberg, S. (2018). Latent Space Oddity: on the Curvature of Deep Generative Models. *arXiv:1710.11379 [stat]*. arXiv: 1710.11379.

Aster, R., Borchers, B., and Thurber, C. (2013). *Parameters estimation and inverse problems*. Academic press, 2nd edition edition.

Aydin, O. and Caers, J. (2013). Image transforms for determining fit-for-purpose complexity of geostatistical models in flow modeling. *Computational Geosciences*, 17(2):417–429.

Backus, G. E. and Gilbert, J. F. (1967). Numerical Applications of a Formalism for Geophysical Inverse Problems. *Geophysical Journal of the Royal Astronomical Society*, 13(1-3):247–276. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-246X.1967.tb02159.x.

Bai, T. and Tahmasebi, P. (2020). Hybrid geological modeling: Combining

machine learning and multiple-point statistics. *Computers & Geosciences*, 142:104519.

Bengio, Y. (2009). Learning Deep Architectures for AI. *Found. Trends Mach. Learn.*, 2(1):1–127. Place: Hanover, MA, USA Publisher: Now Publishers Inc.

Bergmann, U., Jetchev, N., and Vollgraf, R. (2017). Learning Texture Manifolds with the Periodic Spatial GAN. *arXiv:1705.06566 [cs, stat]*. arXiv: 1705.06566.

Birchak, J. R., Gardner, C., Hipp, J. E., and Victor, J. M. (1974). High dielectric constant microwave probes for sensing soil moisture. *Proceedings of the IEEE*, 62(1):93–98.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Information science and statistics. Springer, New York.

Bora, A., Jalal, A., Price, E., and Dimakis, A. G. (2017). Compressed Sensing using Generative Models. *arXiv:1703.03208 [cs, math, stat]*. arXiv: 1703.03208.

Bording, T. S., Fiandaca, G., Maurya, P. K., Auken, E., Christiansen, A. V., Tuxen, N., Klint, K. E. S., and Larsen, T. H. (2019). Cross-borehole tomography with full-decay spectral time-domain induced polarization for mapping of potential contaminant flow-paths. *Journal of Contaminant Hydrology*, 226:103523.

Born, M. and Wolf, E. (1980). *Principles of Optics*. Pergamon, sixth edition edition.

Caers, J. and Hoffman, T. (2006). The Probability Perturbation Method: A New Look at Bayesian Inverse Modeling. *Mathematical Geology*, 38(1):81–100.

Canchumuni, S. W., Emerick, A. A., and Pacheco, M. A. C. (2019). Towards a robust parameterization for conditioning facies models using deep variational autoencoders and ensemble smoother. *Computers & Geosciences*, 128:87–102.

Cassidy, N. J. (2009). Chapter 2 - Electrical and Magnetic Properties of Rocks, Soils and Fluids. In Jol, H. M., editor, *Ground Penetrating Radar Theory and Applications*, pages 41 – 72. Elsevier, Amsterdam.

Caterina, D., Hermans, T., and Nguyen, F. (2014). Case studies of incorporation of prior information in electrical resistivity tomography: comparison of different approaches. *Near Surface Geophysics*, 12:451–465.

Chan, S. and Elsheikh, A. H. (2019). Parametrization and generation of geological models with generative adversarial networks. *arXiv:1708.01810 [physics, stat]*. arXiv: 1708.01810.

Chaudhari, P. and Soatto, S. (2018). Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. *arXiv:1710.11029 [cond-mat, stat]*. arXiv: 1710.11029.

Chen, N., Klushyn, A., Kurle, R., Jiang, X., Bayer, J., and van der Smagt, P. (2018). Metrics for Deep Generative Models. *arXiv:1711.01204 [cs, stat]*. arXiv: 1711.01204.

Comunian, A., De Micheli, L., Lazzati, C., Felletti, F., Giacobbo, F., Giudici, M., and Bersezio, R. (2016). Hierarchical simulation of aquifer heterogeneity: implications of different simulation settings on solute-transport modeling. *Hydrogeology Journal*, 24(2):319–334.

Courant, R. and Hilbert, D. (1989). Application of the Calculus of Variations to Eigenvalue Problems. In *Methods of Mathematical Physics*, pages 397–465. John Wiley & Sons, Ltd. Section: 6 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9783527617210.ch6.

Daniels, D. J., editor (2004). *Ground Penetrating Radar*. Radar, Sonar & Navigation. Institution of Engineering and Technology.

Day-Lewis, F. D. (2005). Applying petrophysical models to radar travel time and electrical resistivity tomograms: Resolution-dependent limitations. *Journal of Geophysical Research*, 110(B8).

Dayan, P., Hinton, G. E., Neal, R. M., and Zemel, R. S. (1995). The Helmholtz Machine. *Neural Computation*, 7(5):889–904. Publisher: MIT Press.

De Leeuw, J. and Heiser, W. J. (1980). Multidimensional scaling with restrictions on the configuration. In *Multivariate Analysis*, volume V, pages 501–522, Amsterdam, the Netherlands. North Holland Publishing Company.

Demyanov, V., Arnold, D., Rojas, T., and Christie, M. (2019). Uncertainty Quantification in Reservoir Prediction: Part 2—Handling Uncertainty in the Geological Scenario. *Mathematical Geosciences*, 51(2):241–264.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78.

Durrani, T. S. and Bisset, D. (1984). The Radon transform and its properties. *Geophysics*, 49(8):1180–1187.

Engel, J., Hoffman, M., and Roberts, A. (2018). Latent Constraints: Learning to Generate Conditionally from Unconditional Generative Models. *ArXiv*, abs/1711.05772.

Falorsi, L., de Haan, P., Davidson, T. R., De Cao, N., Weiler, M., Forré, P., and Cohen, T. S. (2018). Explorations in Homeomorphic Variational Auto-Encoding. *arXiv:1807.04689 [cs, stat]*. arXiv: 1807.04689.

Fefferman, C., Mitter, S., and Narayanan, H. (2016). Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049.

Feyen, L. and Caers, J. (2006). Quantifying geological uncertainty for flow and transport modeling in multi-modal heterogeneous formations. *Advances in Water Resources*, 29(6):912–929.

Franklin, J. N. (1970). Well-posed stochastic extensions of ill-posed linear problems. *Journal of Mathematical Analysis and Applications*, 31(3):682–716.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2014). *Bayesian Data Analysis*. Texts in Statistical Science Series. Chapman and Hall/CRC, third edition edition.

Giroux, B. and Larouche, B. (2013). Task-parallel implementation of 3D shortest path raytracing for geophysical applications. *Computers & Geosciences*, 54:130–141.

Golmohammadi, A. and Jafarpour, B. (2016). Simultaneous geologic scenario identification and flow model calibration with group-sparsity formulations. *Advances in Water Resources*, 92:208–227.

González, E. F., Mukerji, T., and Mavko, G. (2008). Seismic inversion combining rock physics and multiple-point geostatistics. *GEOPHYSICS*, 73(1):R11–R21.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Networks. *arXiv:1406.2661 [cs, stat]*. arXiv: 1406.2661.

Habbema, J. D. F., Hermans, J., and Van den Broek, K. (1974). A stepwise discrimation analysis program using density estimation. In *Proceedings in Computational Statistics*, Vienna. Physica Verlag.

Hand, P. and Voroninski, V. (2018). Global Guarantees for Enforcing Deep Generative Priors by Empirical Risk. *arXiv:1705.07576 [cs, math]*. arXiv: 1705.07576.

Hansen, T. M., Cordua, K. S., Jacobsen, B. H., and Mosegaard, K. (2014). Accounting for imperfect forward modeling in geophysical inverse problems — Exemplified for crosshole tomography. *GEOPHYSICS*, 79(3):H1–H21. _eprint: https://doi.org/10.1190/geo2013-0215.1.

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

Hermans, T., Nguyen, F., and Caers, J. (2015). Uncertainty in training image-based inversion of hydraulic head data constrained to ERT data: Workflow and case study. *Water Resources Research*, 51(7):5332–5352.

Hermans, T., Nguyen, F., Klepikova, M., Dassargues, A., and Caers, J. (2018). Uncertainty Quantification of Medium-Term Heat Storage From Short-Term Geophysical Experiments Using Bayesian Evidential Learning. *Water Resources Research*, 54(4):2931–2948.

Hermans, T., Oware, E., and Caers, J. (2016). Direct prediction of spatially and temporally varying physical properties from time-lapse electrical resistance data. *Water Resources Research*, 52(9):7262–7283.

Hermans, T., Vandenbohede, A., Lebbe, L., Martin, R., Kemna, A., Beaujean, J., and Nguyen, F. (2012). Imaging artificial salt water infiltration using electrical resistivity tomography constrained by geostatistical data. *Journal of Hydrology*, 438-439:168–180.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-VAE: Learning basic visual concepts with a constrained variational framework. page 13.

Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. (1999). Bayesian Model Averaging: A Tutorial. *Statistical Science*, page 22.

Holm-Jensen, T. and Hansen, T. M. (2019). Linear Waveform Tomography Inversion Using Machine Learning Algorithms. *Mathematical Geosciences*.

Hu, L. Y., Blanc, G., and Noetinger, B. (2001). Gradual deformation and iterative calibration of sequential stochastic simulations. *Mathematical Geology*, 33(4):475–489.

Jetchev, N., Bergmann, U., and Vollgraf, R. (2017). Texture Synthesis with Spatial Generative Adversarial Networks. *arXiv:1611.08207 [cs, stat]*. arXiv: 1611.08207.

Jol, H. M. (2009). *Ground Penetrating Radar: Theory and Applications*. Elsevier, Amsterdam.

Journel, A. and Zhang, T. (2007). The Necessity of a Multiple-Point Prior Model. *Mathematical Geology*, 38(5):591–610.

Kessler, T., Klint, K., Nilsson, B., and Bjerg, P. (2012). Characterization of sand lenses embedded in tills. *Quaternary Science Reviews*, 53:55–71.

Kessler, T. C., Comunian, A., Oriani, F., Renard, P., Nilsson, B., Klint, K. E., and Bjerg, P. L. (2013). Modeling Fine-Scale Geological Heterogeneity-Examples of Sand Lenses in Tills. *Groundwater*, 51(5):692–705.

Khaninezhad, M. M. and Jafarpour, B. (2014). Prior model identification during subsurface flow data integration with adaptive sparse representation techniques. *Computational Geosciences*, 18(1):3–16.

Khodabakhshi, M. and Jafarpour, B. (2013). A Bayesian mixture-modeling approach for flow-conditioned multiple-point statistical facies simulation from uncertain training images. *Water Resources Research*, 49(1):328–342.

Kim, J. and Zhang, B.-T. (2019). Data Interpolations in Deep Generative Models under Non-Simply-Connected Manifold Topology. *arXiv:1901.08553 [cs, stat]*. arXiv: 1901.08553.

Kingma, D. P. and Ba, J. (2017). Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*. arXiv: 1412.6980.

Kingma, D. P., Rezende, D. J., Mohamed, S., and Welling, M. (2014). Semi-Supervised Learning with Deep Generative Models. *arXiv:1406.5298 [cs, stat]*. arXiv: 1406.5298.

Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*. arXiv: 1312.6114.

Kleinberg, R., Li, Y., and Yuan, Y. (2018). An Alternative View: When Does SGD Escape Local Minima? *arXiv:1802.06175 [cs]*. arXiv: 1802.06175.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

Laloy, E., Hérault, R., Jacques, D., and Linde, N. (2018). Training-Image Based Geostatistical Inversion Using a Spatial Generative Adversarial Neural Network. *Water Resources Research*, 54(1):381–406.

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

Laloy, E., Linde, N., Ruffino, C., Hérault, R., Gasso, G., and Jacques, D. (2019). Gradient-based deterministic inversion of geophysical data with generative adversarial networks: Is it feasible? *Computers & Geosciences*, 133:104333.

Lange, K., Frydendall, J., Cordua, K. S., Hansen, T. M., Melnikova, Y., and Mosegaard, K. (2012). A Frequency Matching Method: Solving Inverse Problems by Use of Geologically Realistic Prior Information. *Mathematical Geosciences*, 44(7):783–803.

Larsen, T. H., Palstrøm, P., and Larsen, L. (2016). Kallerup Grusgrav: Kortlægning af sandlinser i moræneler. Technical Report 3641400021, Orbicon A/S.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.*, 1(4):541–551. Place: Cambridge, MA, USA Publisher: MIT Press.

Li, X. and Tsai, F. T.-C. (2009). Bayesian model averaging for groundwater head prediction and uncertainty analysis using multimodel and multimethod. *Water Resources Research*, 45(9).

Linde, N., Finsterle, S., and Hubbard, S. (2006). Inversion of tracer test data using tomographic constraints. *Water Resources Research*, 42(4).

Linde, N., Renard, P., Mukerji, T., and Caers, J. (2015). Geological realism in hydrogeological and geophysical inverse modeling: A review. *Advances in Water Resources*, 86:86–101.

Liu, N. and Oliver, D. S. (2005). Ensemble Kalman filter for automatic history matching of geologic facies. *Journal of Petroleum Science and Engineering*, 47(3):147–161.

Looms, M. C., Klotzsche, A., van der Kruk, J., Larsen, T. H., Edsen, A., Tuxen, N., Hamburger, N., Keskinen, J., and Nielsen, L. (2018). Mapping sand layers in clayey till using crosshole ground-penetrating radar. *GEOPHYSICS*, 83(1):A21–A26.

Lopez-Alvis, J., Laloy, E., Nguyen, F., and Hermans, T. (2020). Deep generative models in inversion: a review and development of a new approach based on a variational autoencoder. *arXiv:2008.12056 [physics]*. arXiv: 2008.12056.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bousquet, O. (2018). Are GANs Created Equal? A Large-Scale Study. *arXiv:1711.10337 [cs, stat]*. arXiv: 1711.10337.

Luo, X., Stordal, A. S., Lorentzen, R. J., and Nævdal, G. (2015). Iterative Ensemble Smoother as an Approximate Solution to a Regularized Minimum-Average-Cost Problem: Theory and Applications. *SPE Journal*, 20(05):962–982.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2015). Adversarial Autoencoders. *arXiv:1511.05644 [cs]*. arXiv: 1511.05644.

Malinverno, A. (2002). Parsimonious Bayesian Markov chain Monte Carlo inversion in a nonlinear geophysical problem. *Geophysical Journal International*, 151(3):675–688.

Mariethoz, G. (2018). When Should We Use Multiple-Point Geostatistics? In Daya Sagar, B., Cheng, Q., and Agterberg, F., editors, *Handbook of Mathematical Geosciences: Fifty Years of IAMG*, pages 645–653. Springer International Publishing, Cham.

Mariethoz, G. and Kelly, B. F. J. (2011). Modeling complex geological structures with elementary training images and transform-invariant distances: MPS WITH ELEMENTARY TRAINING IMAGES. *Water Resources Research*, 47(7).

Mariethoz, G., Renard, P., and Straubhaar, J. (2010). The Direct Sampling method to perform multiple-point geostatistical simulations: PERFORMING MULTIPLE-POINTS SIMULATIONS. *Water Resources Research*, 46(11).

Maurer, H., Holliger, K., and Boerner, D. (1998). Stochastic regularization: Smoothness or similarity. *Geophysical Research Letters*, 25(15):2889–2892.

Metz, L., Poole, B., Pfau, D., and Sohl-Dickstein, J. (2017). Unrolled Generative Adversarial Networks. *arXiv:1611.02163 [cs, stat]*. arXiv: 1611.02163.

Mo, S., Zabaras, N., Shi, X., and Wu, J. (2020). Integration of Adversarial Autoencoders With Residual Dense Convolutional Networks for Estimation of Non-Gaussian Hydraulic Conductivities. *Water Resources Research*, 56(2).

Morzfeld, M., Adams, J., Lunderman, S., and Orozco, R. (2018). Feature-based data assimilation in geophysics. *Nonlinear Processes in Geophysics*, 25(2):355–374.

Mosser, L., Dubrule, O., and Blunt, M. J. (2018). Stochastic seismic waveform inversion using generative adversarial networks as a geological prior. *arXiv:1806.03720 [physics, stat]*. arXiv: 1806.03720.

Nabighian, M. N. (1987). *Electromagnetic Methods in Applied Geophysics: Volume 1, Theory*. Society of Exploration Geophysicists.

Naitzat, G., Zhitnikov, A., and Lim, L.-H. (2020). Topology of deep neural networks. *arXiv:2004.06093 [cs, math, stat]*. arXiv: 2004.06093.

Neuman, S. P. (2003). Maximum likelihood Bayesian averaging of uncertain model predictions. *Stochastic Environmental Research and Risk Assessment (SERRA)*, 17(5):291–305.

Park, H., Scheidt, C., Fenwick, D., Boucher, A., and Caers, J. (2013). History matching and uncertainty quantification of facies models with multiple geological interpretations. *Computational Geosciences*, 17(4):609–621.

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., and Tran, D. (2018). Image Transformer. *arXiv:1802.05751 [cs]*. arXiv: 1802.05751.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch. page 4.

Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv:1511.06434 [cs]*. arXiv: 1511.06434.

Raftery, A. E. (1995). Bayesian Model Selection in Social Research. *Sociological Methodology*, 25:111–163.

Renard, P. and Allard, D. (2013). Connectivity metrics for subsurface flow and transport. *Advances in Water Resources*, 51:168–196.

Rezaee, H. and Marcotte, D. (2018). Calibration of categorical simulations by evolutionary gradual deformation method. *Computational Geosciences*, 22(2):587–605.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *arXiv:1401.4082 [cs, stat]*. arXiv: 1401.4082.

Richardson, A. (2018). Generative Adversarial Networks for Model Order Reduction in Seismic Full-Waveform Inversion. *arXiv:1806.00828 [physics]*. arXiv: 1806.00828.

Rolinek, M., Zietlow, D., and Martius, G. (2019). Variational Autoencoders Pursue PCA Directions (by Accident). *arXiv:1812.06775 [cs, stat]*. arXiv: 1812.06775.

Rücker, C., Günther, T., and Wagner, F. M. (2017). pyGIMLi: An open-source library for modelling and inversion in geophysics. *Computers & Geosciences*, 109:106–123.

Sajjadi, M. S. M., Bachem, O., Lucic, M., Bousquet, O., and Gelly, S. (2018). Assessing Generative Models via Precision and Recall. *arXiv:1806.00035 [cs, stat]*. arXiv: 1806.00035.

Salakhutdinov, R. (2015). Learning Deep Generative Models. *Annual Review of Statistics and Its Application*, 2(1):361–385.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved Techniques for Training GANs. *arXiv:1606.03498 [cs]*. arXiv: 1606.03498.

Scheidt, C. and Caers, J. (2009). Representing Spatial Uncertainty Using Distances and Kernels. *Mathematical Geosciences*, 41(4):397–419.

Scheidt, C., Fernandes, A. M., Paola, C., and Caers, J. (2016). Quantifying natural delta variability using a multiple-point geostatistics prior uncertainty model: DELTA VARIABILITY AND GEOSTATISTICS. *Journal of Geophysical Research: Earth Surface*, 121(10):1800–1818.

Scheidt, C., Jeong, C., Mukerji, T., and Caers, J. (2015a). Probabilistic falsification of prior geologic uncertainty with seismic amplitude data: Application to a turbidite reservoir case. *Geophysics*, 80(5):M89–M12.

Scheidt, C., Li, L., and Caers, J. (2018). *Quantifying Uncertainty in Subsurface Systems*. Number 236 in Geophysical Monograph Series. John Wiley and Sons & American Geophysical Union, Hoboken, NJ & Washington D.C.

Scheidt, C., Tahmasebi, P., Pontiggia, M., Da Pra, A., and Caers, J. (2015b). Updating joint uncertainty in trend and depositional scenario for reservoir exploration and early appraisal. *Computational Geosciences*, 19(4):805–820.

Seo, J. K., Kim, K. C., Jargal, A., Lee, K., and Harrach, B. (2019). A Learning-Based Method for Solving Ill-Posed Nonlinear Inverse Problems: A Simulation Study of Lung EIT. *SIAM Journal on Imaging Sciences*, 12(3):1275–1295.

Shao, H., Kumar, A., and Fletcher, P. T. (2017). The Riemannian Geometry of Deep Generative Models. *arXiv:1711.08014 [cs, stat]*. arXiv: 1711.08014.

Silva, D. A. and Deutsch, C. V. (2012). Multiple Point Statistics with Multiple Training Images. *CCG Annual Report*, 14:8.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Monographs on Statistics and Applied Probability. Chapman and Hall, London.

Smith, S. L. and Le, Q. V. (2018). A Bayesian Perspective on Generalization and Stochastic Gradient Descent. *arXiv:1710.06451 [cs, stat]*. arXiv: 1710.06451.

Soille, P. (2004). *Morphological Image Analysis*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Solimini, D. (2016). *Understanding Earth Observation*, volume 23 of *Remote Sensing and Digital Image Processing*. Springer International Publishing.

Straubhaar, J., Renard, P., Mariethoz, G., Froidevaux, R., and Besson, O. (2011). An Improved Parallel Multiple-point Algorithm Using a List Approach. *Mathematical Geosciences*, 43(3):305–328.

Strebelle, S. (2002). Conditional simulation of complex geological structures using multiple-point statistics. *Mathematical Geology*, 34(1):1–21.

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

Tarantola, A. and Valette, B. (1982). Inverse problems = quest for information. *Journal of Geophysics*, 50(3):159–170.

Tikhonov, A. N. and Arsenin, V. I. A. (1977). *Solutions of ill-posed problems*. Winston.

Treister, E. and Haber, E. (2016). A fast marching algorithm for the factored eikonal equation. *Journal of Computational Physics*, 324:210–225. arXiv: 1607.00973.

Tsai, F. T.-C. and Elshall, A. S. (2013). Hierarchical Bayesian model averaging for hydrostratigraphic modeling: Uncertainty segregation and comparative evaluation. *Water Resources Research*, 49(9):5520–5536.

Uria, B., Murray, I., and Larochelle, H. (2014). A Deep and Tractable Density Estimator. *arXiv:1310.1757 [cs, stat]*. arXiv: 1310.1757.

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., and contributors, t. s.-i. (2014). scikit-image: image processing in Python. *PeerJ*, 2:e453.

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

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

Ye, M., Neuman, S. P., and Meyer, P. D. (2004). Maximum likelihood Bayesian averaging of spatial variability models in unsaturated fractured tuff. *Water Resources Research*, 40(5).

Zahner, T., Lochbühler, T., Mariethoz, G., and Linde, N. (2016). Image synthesis with graph cuts: a fast model proposal mechanism in probabilistic inversion. *Geophysical Journal International*, 204(2):1179–1190.

Zeiler, M. D. and Fergus, R. (2013). Visualizing and Understanding Convolutional Networks. *arXiv:1311.2901 [cs]*. arXiv: 1311.2901.

Zelt, C. A. and Chen, J. (2016). Frequency-dependent traveltime tomography for near-surface seismic refraction data. *Geophysical Journal International*, 207(1):72–88.

Zhang, C., Butepage, J., Kjellstrom, H., and Mandt, S. (2018). Advances in Variational Inference. *arXiv:1711.05597 [cs, stat]*. arXiv: 1711.05597.

Zhdanov, M. S. (2018). *Foundations of Geophysical Electromagnetic Theory and Methods*. Elsevier, second edition edition.