Categorical Spatial Data: A Bayesian Journey through Uncertainty Quantification, Generative Modeling, and Image Comparison

Oscar Ovanger

July 11, 2025

Preface

My PhD journey has been incredibly educational, both personally and academically. Looking back, there are many people I am deeply grateful to.

First and foremost, I want to thank my supervisors: Jo Eidsvik, Ragnar Hauge, Jacob Skauvold, and Ingrid Aarnes. You have been exceptional mentors throughout this journey. It was a fast and challenging transition from machine learning to geostatistics, and you have been thoughtful collaborators and academic partners every step of the way. Jo has been with me since my master's thesis, and I truly appreciate the dedication he has shown to my work ever since. I truly appreciate the time and effort all of you put into guiding me–there were many weekends when you reviewed my writing and provided detailed feedback over the course of several years. I remember the early meetings when the terminology felt completely foreign to me, but you were patient and taught me so much in a relatively short time. I'm also grateful that you gave me the creative freedom to explore problems I found interesting, go on multiple international conferences, and to pursue some unconventional models.

I know Ragnar still thinks I have a lot to learn when it comes to making and interpreting plots, but perhaps a PhD doesn't mean you've finished learning, and I find some comfort in that.

I'd also like to thank my office mate, Daesoo Lee, who shared an office with me for almost four years. It's been a true pleasure to discuss ideas, chat about everything and nothing, and eat cinnamon buns together nearly every Wednesday. I've also had the privilege of getting to know your wife Senida and daughter Ara and it's been lovely to share dinners and moments together over the years. We are co-authors on two of the papers in this thesis, and it was both rewarding and educational to work with you. It's rare to meet someone as efficient as you.

I'm also grateful to Michael Pyrcz, who hosted me at UT Austin in Texas. That collaboration was incredibly valuable and insightful. We co-wrote a paper that I'm very proud of, and working with you and getting to know the PhD students in your lab was a lot of fun. I especially remember a three-hour kayak trip we took, where I got completely sunburned, but it was worth it.

Finally, I want to thank my family and my partner. To my mom and dad,

thank you for supporting me throughout this journey. If you hadn't believed in me and pushed me to pursue this path, I wouldn't be here today. To my sister Liv, thank you for being so supportive over these past four years, and for the time we spent together in Trondheim. To my partner, Amber, someone I never would have met had I not pursued a PhD and done a research stay in Texas, thank you for being the most incredible support and conversation partner. You've been endlessly patient and have lifted me up whenever I doubted myself.

Thank you to the Norwegian Research Council for funding this PhD and to everyone at the Department of Mathematical Sciences for helping administer this and supporting my academic development.

Most of all, I leave this PhD motivated to keep learning, growing, and exploring new and exciting ideas. Thank you to everyone.

Oscar Ovanger, Trondheim, July 11, 2025

Contents

Part I

Background

Introduction

The Earth's subsurface harbors critical resources and opportunities: where to find oil, where to store CO_2, how groundwater flows, and where minerals accumulate. Yet we can only observe this hidden world through sparse, expensive measurements—drill holes that pierce the earth like needles sampling a vast tapestry. From these limited observations, we must reconstruct entire three-dimensional geological architectures that control billion-dollar decisions and environmental outcomes.

This thesis addresses a fundamental challenge in geosciences: how do we generate realistic models of subsurface geology that honor our sparse observations while capturing the uncertainty inherent in such systems? The problem is particularly acute when dealing with categorical geological variables—distinct rock types like sandstone, shale, and limestone that form complex spatial patterns critical for resource extraction and environmental management.

1.1 The Geostatistical Challenge

Consider a typical oil reservoir characterization problem. We have drilled perhaps a dozen wells across an area spanning several square kilometers, each well providing a one-dimensional profile of rock types encountered at different depths along the well path. Additionally, seismic surveys provide indirect information about rock properties across the entire reservoir volume, though with lower resolution and significant uncertainty in the rock type interpretation. From these sparse vertical lines of direct data and extensive but ambiguous seismic observations, we must infer the three-dimensional distribution of rock types throughout the entire reservoir volume—millions of grid cells, each assigned to one of several geological facies, with uncertainty statements. The challenge is compounded by several factors unique to geological systems:

Categorical Nature: Unlike temperature or pressure that vary continuously, rock types are discrete categories with sharp boundaries. A sandstone layer does not gradually transition into shale; the contact is abrupt. These sharp transitions control fluid flow—a connected sandstone channel can transport oil across kilometers, while a thin shale barrier can completely block flow.

Complex Spatial Patterns: Geological facies are not randomly distributed. They follow patterns dictated by ancient depositional processes—meandering rivers that created sinuous sand channels, storms that spread sheet-like sand bodies, or quiet seas that accumulated thick shale blankets. These patterns exhibit both large-scale trends and fine-scale variability that must be captured in our models.

Connectivity Criticality: In subsurface applications, connectivity matters more than local accuracy (Caers and Zhang, 2004). A model with incorrect connectivity, whether broken connections that exist in reality or spurious connections that do not, can lead to costly errors in decision-making, such as misplaced wells during oilfield development.

High Dimensionality: A modest $100 \times 100 \times 50$ grid with 5 rock types contains $5^{500,000}$ possible configurations—a space so vast that exhaustive exploration is impossible. We must find clever ways to navigate this space and identify configurations consistent with our observations.

This forces us to adopt sampling-based approaches that can efficiently explore the space of plausible models without attempting to evaluate every possible configuration.

1.2 The Bayesian Conditioning Framework

Bayesian conditioning provides a principled approach to this challenge by combining prior geological knowledge with observational data. The framework consists of three key components:

Prior Model $P(\mathbf{x})$: This probabilistic model encodes our understanding of how geological facies denoted $\mathbf{x}$ are spatially distributed based on depositional processes, analogues, and geological principles. Throughout this thesis, we explore various prior models—from simple statistical approaches like Truncated Gaussian Random Fields to complex rule-based models that explicitly encode stratigraphic principles.

Data Likelihood $P(\mathbf{d}|\mathbf{x})$: This observation model function describes how our data, denoted $\mathbf{d}$, relate to the true subsurface. Sometimes this relationship is deterministic (well data directly samples rock types), while other times it is probabilistic (seismic data provides indirect information about rock properties).

Posterior Distribution $P(\mathbf{x}|\mathbf{d})$: Bayes' rule combines prior and likelihood to define the posterior—all possible subsurface configurations consistent with both our geological understanding and observed data. The computational challenge lies in efficiently exploring this posterior to generate multiple plausible realizations that capture subsurface uncertainty.

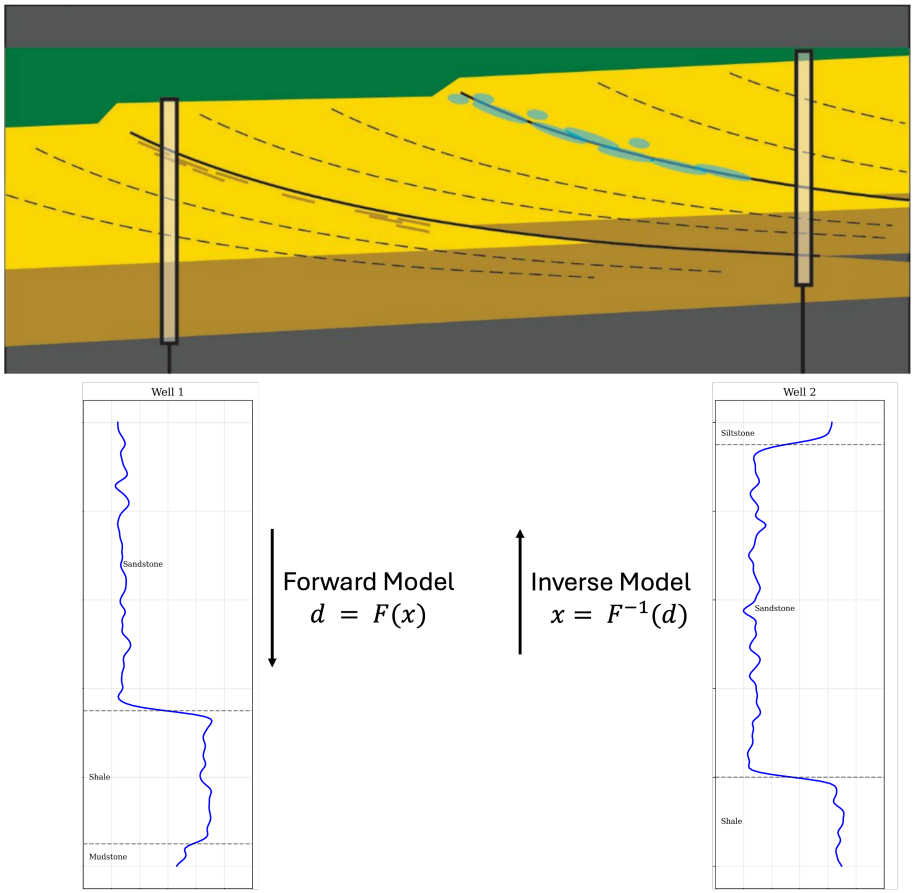


Figure 1.1: Geological conditioning problem. *Top*: Conceptual cross-section of a wave-dominated shore-face parasequence coloured by categorical facies (green = siltstone, yellow = sandstone, brown = shale, grey = mudstone). Dashed lines sketch a family of possible bed-set boundaries in the prior model, while the two vertical well bores (white) pierce the sequence and record lithology at discrete depths. *Bottom*: Gamma-ray logs from the two wells: high values indicate shale and silt, medium shale and low values sand. These logs are the data that must be honored. The Bayesian conditioning task is to generate an ensemble of facies realisations honoring both the prior distribution and the observed well data.

1.3 Computational Approaches and Challenges

The fundamental computational challenge in Bayesian conditioning is exploring the posterior distribution $P(\mathbf{x}|\mathbf{d})$. For categorical spatial models, this posterior often consists of disconnected regions in a vast discrete space, making traditional optimization methods ineffective. Two main computational paradigms have emerged:

Explicit Methods: These maintain the full Bayesian framework, using techniques like Markov Chain Monte Carlo (MCMC) to explore the posterior (Metropolis et al., 1953; Hastings, 1970). While theoretically elegant and asymptotically exact, these methods face challenges in high-dimensional spaces with complex constraints. Sophisticated strategies like tempering and blocking help navigate the space, but convergence can require millions of iterations with Markov chain updating.

Implicit Methods: These bypass explicit posterior computation by learning direct mappings from data to realizations. Multiple-point statistics (MPS) methods (Strebelle, 2002) pioneered this approach by using training images to capture complex patterns, replacing variogram-based models with direct pattern sampling. However, given machine learning's demonstrated superiority in pattern recognition tasks across domains, it is natural to expect neural networks to excel here as well. By training on thousands of prior samples and their corresponding data, deep learning models can provide near-instantaneous conditioning while potentially capturing more complex patterns than traditional MPS. Nevertheless, these methods may miss rare configurations and can generate geologically implausible results that violate fundamental constraints.

1.4 Evaluating Posterior Sampling Methods

A critical but often overlooked challenge in Bayesian conditioning is evaluating the quality of posterior samples. When multiple methods claim to solve the same conditioning problem—whether through MCMC, neural networks, or other computational approaches—how do we determine which performs better? For spatial categorical models, this requires careful consideration of what statistical properties matter most for the intended application.

The evaluation of posterior samples typically proceeds through a hierarchy of statistical comparisons:

First-order Statistics: The most basic level examines marginal distributions—do the generated samples maintain the expected proportions of each facies type from the prior model or training data? For instance, if our prior indicates 30% sandstone, 50% shale, and 20% limestone, do the conditional realizations preserve these proportions away from data locations? While nec-

essary for consistency, matching global proportions alone is insufficient. A random arrangement of facies with these exact proportions would fail to capture any meaningful geological structure.

Second-order Statistics: Spatial correlations capture how facies relate to their neighbors through metrics like variograms, connectivity functions, and transition probabilities. These statistics reveal whether sand bodies have appropriate dimensions, whether facies boundaries occur at realistic frequencies, and whether spatial continuity matches our geological understanding.

Higher-order Statistics: Complex spatial patterns often require statistics beyond pairwise correlations. Multiple-point statistics, cluster distributions, and morphological measures capture characteristics like channel sinuosity, object shapes, and hierarchical organization that define realistic geological architectures.

Feature Summary Statistics: Beyond individual statistics, composite metrics attempt to capture overall structural similarity. Generic structural features—such as object shapes, spatial arrangements, and multi-scale patterns—can provide resolution-invariant comparisons between images regardless of specific application. These metrics, originally developed in computer vision for assessing perceptual similarity, have found increasing use in geological applications where they complement domain-specific measures. Meanwhile, domain-specific features often matter most for practical decisions. In reservoir modeling, connected pore volume, breakthrough times, and flow-based metrics directly relate to the economic value of a model. The combination of generic structural metrics and application-driven statistics provides a more complete assessment of model adequacy, capturing both visual realism and functional performance.

The challenge lies not just in computing these statistics, but in determining which matter most for a given application and how to weight them appropriately. A method that exactly matches the variogram of the prior model or training data might fail to capture critical connectivity patterns. Conversely, matching complex features might come at the cost of basic statistical properties.

1.5 Thesis Contributions

This thesis advances Bayesian conditioning of geostatistical categorical models through five interconnected papers:

Paper I tackles the challenge of configuration uncertainty in deviated wells, where the same geological layer can be intersected multiple times, creating complex multimodal posteriors that traditional methods struggle to capture.

Papers II, III, and IV explore neural network approaches from complementary angles: Paper II introduces conditional diffusion models that honor

hard constraints without retraining; Paper III benchmarks these against classical geostatistical priors; Paper IV investigates the sampling distribution of vision transformers by directly analyzing their likelihood functions, revealing systematic biases in how these models capture spatial pattern.

Paper V develops new metrics for comparing spatial categorical images across different methods, addressing the fundamental challenge of validating models that operate at different scales and resolutions.

1.6 Thesis Overview

The remainder of Part I provides the technical foundation for understanding these contributions:

- Chapter 2 reviews spatial categorical models—the mathematical frameworks for representing geological facies distributions
- Chapter 3 examines Bayesian inference machinery, from classical MCMC to modern deep learning approaches
- Chapter 4 discusses image quality assessment metrics essential for validating and comparing different methods
- Chapter 5 connects these concepts to the specific innovations in each paper

Together, these chapters demonstrate how combining geological knowledge, sparse observational data, and advanced computational methods enables us to peer into the subsurface and make better decisions about our planet's hidden resources.

Spatial Categorical Models

The quest to represent spatial categorical fields—distinct rock types, soil classes, or land use categories—has driven decades of innovation in geostatistics and spatial modeling. From the simplest pixel-based approaches to sophisticated process simulations that mimic physical phenomena, the field has evolved along two competing axes: mathematical tractability versus geological realism.

Figure 2.1 illustrates this spectrum of spatial categorical models. On the left, Markov Random Fields (MRFs) represent the mathematical extreme—models defined through local conditional distributions that enable elegant theoretical analysis (Besag, 1974; Geman and Geman, 1984). These models, borrowed from statistical physics and image processing, characterize spatial dependence through neighborhood interactions, offering computational efficiency at the cost of limited geological expressiveness.

Moving rightward, Truncated Gaussian Random Fields (TGRFs) emerged as a practical compromise (Matheron, 1973). By thresholding continuous Gaussian fields, TGRFs inherit the well-understood covariance structure of Gaussian processes while producing categorical realizations. This approach, refined through plurigaussian extensions (Armstrong et al., 2011), became the workhorse of petroleum reservoir modeling, balancing mathematical tractability with the ability to reproduce essential geological features like spatial correlation and anisotropy.

The recognition that two-point statistics inadequately capture complex geological patterns motivated Multiple-Point Statistics (MPS) (Guardiano and Srivastava, 1993). Rather than relying on variograms, MPS methods scan training images to learn pattern distributions, then reproduce these patterns in simulations (Strebelle, 2002). This data-driven approach, exemplified by algorithms like SNESIM (Strebelle, 2002) marked a shift from parametric to example-based modeling.

Further right along the spectrum, object-based models directly place geometric primitives—channels, lobes, or fractures—according to geological rules (Deutsch and Wang, 1996). These models excel at reproducing specific geological architectures with known geometries but struggle with conditioning to dense data. Rule-based approaches generalize this concept, encoding ge-

ological knowledge through hierarchical rules and stratigraphic relationships (Pyrcz et al., 2009), offering a middle ground between geometric simplicity and process complexity.

At the far right, process-based models simulate the physical mechanisms of sediment transport, deposition, and erosion (Griffiths, 2001). These forward stratigraphic models produce the most geologically realistic results by solving governing equations, but their computational demands and parameter uncertainty limit practical application to conditioning problems.

The machine learning revolution has introduced a new dimension to this spectrum, with neural networks learning implicit representations from data (Laloy et al., 2019; Chan and Elsheikh, 2017). These ML-based approaches, shown below the traditional spectrum in Figure 2.1, promise to combine the ease of conditioning from the left with the realism from the right—though their position in this trade-off remains an active area of research.

Throughout this chapter we use a shared nomenclature for all spatial categorical models. We denote the facies variable on the spatial grid by $\mathbf{x} = [x_1, \ldots, xN]^T \in \mathbb{Z}^N_{[1,C]}$, where each x_i represents a categorical value at location index i, and C is the number of categories. Throughout our applications, we work with 2D square grids such that $n^2 = N$, where n is the number of cells in each row and column. Our primary objective is to establish a prior belief over this 2D grid, denoted by probability $P(\mathbf{x})$. While numerous choices exist for $P(\mathbf{x})$, we focus on those most relevant to the papers outlined in this thesis. This is not intended as a comprehensive review of all spatial categorical models, but rather a focused examination of the specific modeling approaches–TGRFs, MRFs, and rule-based geostatistical models–that we have employed and extended in our work.

Section 2.1 examines truncated Gaussian random fields, the industry standard that balances practicality with theory. Section 2.2 explores Markov random fields, which provide the mathematical foundation for many spatial models. Section 2.3 describes rule-based approaches that encode geological knowledge directly. While not exhaustive, these models represent the key prior distributions we employ and extend throughout our work.

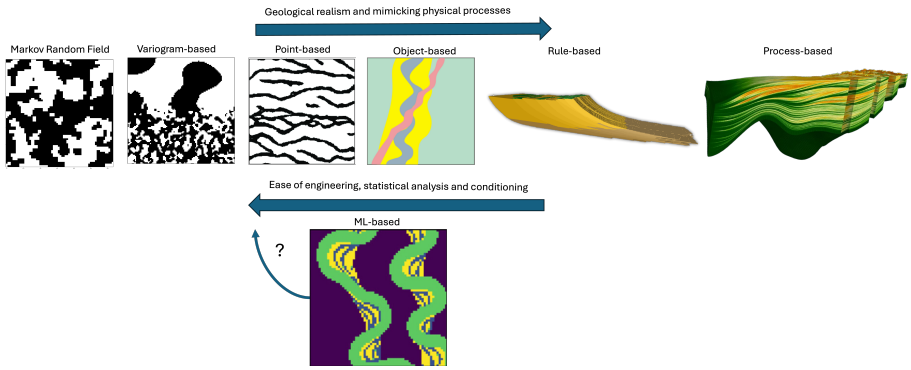


Figure 2.1: Conceptual spectrum of prior models for categorical facies simulation. Panels (left to right) illustrate: **(i)** Markov Random Field models (MRF); **(ii)** two–point statistical models (TGRF/MRF); **(iii)** multiple–point statistics (MPS)(Guardiano and Srivastava, 1993; Strebelle, 2002; Caers and Zhang, 2004); **(iv)** surface- or rule-based models; **(v)** object-based models (Deutsch and Wang, 1996); and **(vi)** process-based forward stratigraphic simulations.

2.1 Truncated Gaussian Random Fields

TGRFs emerge from GRFs, which are continuous-valued fields (Matheron, 1973; Armstrong et al., 2011). The truncation process transforms continuous fields into discrete categorical fields through thresholding operations.

2.1.1 Gaussian Random Fields: Foundation and Properties

A GRF is an infinite-dimensional Gaussian distribution characterized by two fundamental functions:

- The **mean function** $\mu(\mathbf{s})$, which specifies the expected value at any spatial location $\mathbf{s}$
- The **covariance function** $\Sigma(\mathbf{s}, \mathbf{s}')$, which describes the covariance between values at locations $\mathbf{s}$ and $\mathbf{s}'$

These continuous functions can theoretically be evaluated at any spatial location, enabling arbitrarily fine-grained discretization. However, computational constraints necessitate discretization onto finite grids of size N.

2.1.2 Stationarity

The statistical properties of GRFs are characterized by their stationarity:

- **Stationary Mean:** $\mu(\mathbf{s}) = \mu$ for all $\mathbf{s}$, meaning the expected value is constant across the field.

- **Stationary Covariance:** $\Sigma(\mathbf{s}, \mathbf{s}') = \Sigma(\mathbf{h})$ where $\mathbf{h} = |\mathbf{s} - \mathbf{s}'|$, meaning covariance depends only on the separation vector, not absolute position.

- **Non-stationary fields:** When either condition is violated, creating spatially varying statistical properties.

Figure 2.2 illustrates these concepts through three representative examples: (A) a field with non-stationary mean but stationary covariance, showing a clear spatial trend; (B) a field with zero mean but non-stationary covariance, exhibiting spatially varying correlation structures; and (C) an anisotropic field with direction-dependent covariance.

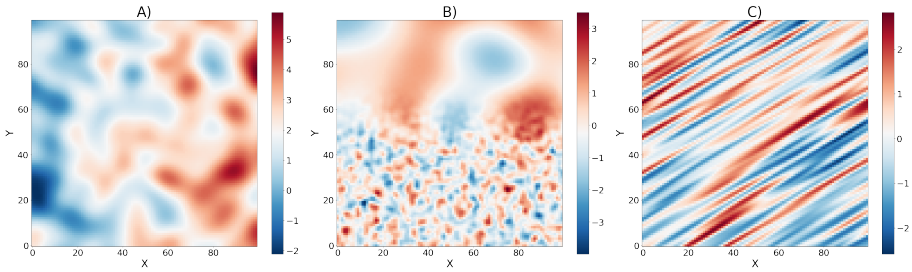


Figure 2.2: 3 realizations of Gaussian Random Fields with different parameterizations.

2.1.3 Covariance Functions and Spatial Structure

The covariance function $\Sigma(\mathbf{h})$ must satisfy mathematical constraints to ensure validity—specifically, it must yield a positive definite covariance matrix for any discretized field. Popular choices include:

- **Gaussian covariance:** $\Sigma(\mathbf{h}) = \sigma^2 \exp\left(-\frac{\|\mathbf{h}\|^2}{2\ell^2}\right)$ produces smooth, differentiable fields

- **Matérn covariance:** $\Sigma(\mathbf{h}) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{\|\mathbf{h}\|}{\ell}\right)^{\nu} K_\nu \left(\sqrt{2\nu}\frac{\|\mathbf{h}\|}{\ell}\right)$ offers tunable roughness through parameter ν (Rasmussen, 2004)

- **Exponential covariance:** $\Sigma(\mathbf{h}) = \sigma^2 \exp\left(-\frac{|\mathbf{h}|}{\ell}\right)$ generates more rugged, non-differentiable fields

Here, σ^2 represents the variance, ℓ the correlation length, and $\|\mathbf{h}\|$ the Euclidean distance. The correlation length ℓ controls the spatial extent of interactions between grid cells—larger values produce broader, more connected structures.

2.1.4 Probability Distribution of GRFs

When discretized onto a grid with N points, a GRF follows a multivariate Gaussian distribution. For a field $\mathbf{y} = [y_1, \ldots, y_N]^T$ with mean vector $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_N]^T$ and covariance matrix $\boldsymbol{\Sigma}$, the probability density function is:

$$P(\mathbf{y}) = \frac{1}{(2\pi)^{N/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right), \tag{2.1.1}$$

where $|\boldsymbol{\Sigma}|$ denotes the determinant of the covariance matrix. For stationary fields, $\mu_i = \mu$ for all i, and the covariance matrix elements are $\Sigma_{ij} = \Sigma(|\mathbf{s}_i - \mathbf{s}_j|)$.

This explicit form enables direct computation of probabilities and efficient sampling through standard multivariate Gaussian techniques, making GRFs attractive as priors in Bayesian conditioning problems.

2.1.5 Isotropy and Anisotropy

An **isotropic** field exhibits rotational invariance, where covariance depends only on the magnitude $\|\mathbf{h}\|$. In contrast, **anisotropic** fields have direction-dependent correlation structures, often parameterized through:

$$\Sigma(\mathbf{h}) = \sigma^2 f\left(\sqrt{\mathbf{h}^T\mathbf{A}\mathbf{h}}\right), \tag{2.1.2}$$

where $\mathbf{A}$ is an anisotropy matrix encoding preferential directions and correlation lengths and f represents a correlation function.

2.1.6 From GRFs to TGRFs: The Truncation Process

A TGRF transforms a continuous GRF into a categorical field through thresholding. For an C-ary TGRF with categories $\{1, 2, \ldots, C\}$, we define $(C-1)$ threshold values $t_1 < t_2 < \cdots < t_{C-1}$ such that:

$$x_i = k \quad \text{if} \quad t_{k-1} < y_i \leq t_k, \tag{2.1.3}$$

where y_i is the continuous GRF value at location i, with $t_0 = -\infty$ and $t_C = +\infty$.

While TGRFs are conceptually straightforward to generate, their probability distribution lacks a tractable closed form. For a categorical field $\mathbf{x} = [x_1, \ldots, x_N]^T$ where each $x_i \in \{1, 2, \ldots, C\}$, the probability requires integrating over all continuous GRF configurations that would yield this specific categorical pattern:

$$P(\mathbf{x}) = P(x_1 = k_1, x_2 = k_2, \ldots, x_N = k_N), \tag{2.1.4}$$

where each constraint $x_i = k_i$ translates to $t_{k_i-1} < y_i \leq t_{k_i}$ for the underlying GRF value y_i. This yields:

$$P(\mathbf{x}) = \int_{t_{k_1-1}}^{t_{k_1}} \cdots \int_{t_{k_N-1}}^{t_{k_N}} P(\mathbf{y})\, dy_1 \cdots dy_N. \tag{2.1.5}$$

If the underlying GRF were uncorrelated, this would factorize into a product of independent one-dimensional integrals, each easily computed as differences of cumulative distribution functions. However, the spatial correlation structure that makes TGRFs useful for modeling geological patterns also makes this integral intractable. The multivariate Gaussian distribution $P(\mathbf{y})$ with its full covariance matrix prevents factorization, leaving us with a genuine N-dimensional integral. Even for a modest 100×100 grid, we face a 10,000-dimensional integral with no closed-form solution.

The computational intractability of $P(\mathbf{x})$ has profound implications for Bayesian conditioning problems. While sampling from a TGRF prior is straightforward (generate GRF, then threshold), evaluating the prior probability of a given categorical configuration which is essential for MCMC acceptance ratios, typically requires approximation. This challenge has motivated several alternative conditioning approaches:

- Sequential Gaussian simulation with indicator kriging (Journel, 1998)
- Variational approximations using factorized distributions (Blei et al., 2017)
- Gibbs sampling of the underlying continuous field at conditioning locations (Armstrong et al., 2011)

The last approach represents the standard method for TGRF conditioning: rather than working with categorical variables directly, Gibbs sampling operates on the underlying continuous field. At each data location, values are drawn from the continuous field that, when thresholded, produce the observed category. Once these continuous values are fixed at data locations, the remainder of the field can be generated through conditional simulation (e.g., kriging), and the resulting continuous realization is then thresholded to produce the final categorical field. This elegant solution leverages the tractability of Gaussian conditioning while respecting the categorical constraints.

For a stationary GRF with mean μ and unit variance, the expected volume fractions (category proportions) are determined by:

$$P(X = k) = \Phi(t_k - \mu) - \Phi(t_{k-1} - \mu), \tag{2.1.6}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function.

This framework allows direct control over category proportions through threshold selection. For instance, setting $t_1 = \mu$ in a binary TGRF yields equal proportions $P(X = 1) = P(X = 2) = 0.5$, while adjusting thresholds creates biased fields favoring specific categories.

Figure 2.3 demonstrates how different GRF properties translate into categorical patterns when a single threshold is applied. The trending GRF produces

a binary field with a clear transition from one category to another, following the underlying spatial gradient. The non-stationary covariance GRF creates large connected regions where correlations are strong and fragmented patterns where correlations are weak. The anisotropic GRF generates elongated categorical boundaries aligned with the preferential correlation direction, resulting in banded structures reminiscent of geological layers.

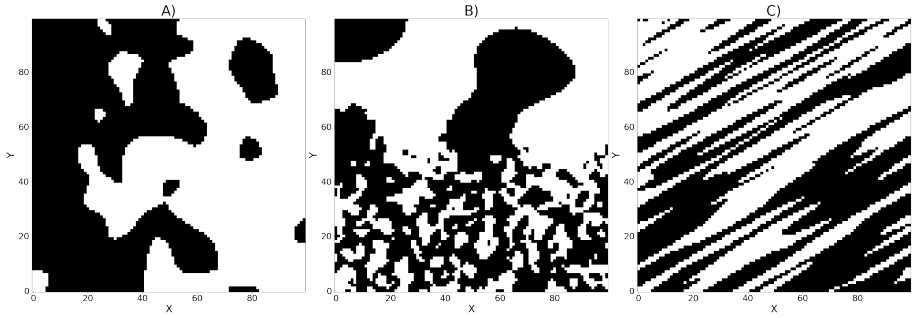


Figure 2.3: Binary TGRF realizations generated by thresholding the continuous GRFs from Figure 2.2. Black represents category 1, white represents category 2, with equal volume fractions (50% each). (A) Trending GRF, (B) Non-stationary covariance GRF, (C) Anisotropic GRF.

Extension to multiple categories requires additional thresholds. Figure 2.4 shows three-category fields where the choice of two thresholds controls both the volume fractions and the spatial arrangement of categories. The trending field maintains smooth transitions between all three categories following the underlying gradient. The non-stationary field exhibits scale-dependent behavior—large homogeneous regions transition to small fragmented patches as the correlation length varies. The anisotropic field produces layered structures with elongated boundaries, creating patterns often observed in sedimentary geological systems.

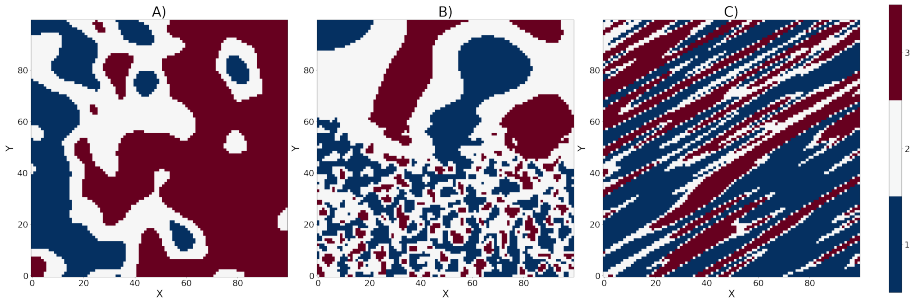


Figure 2.4: Three-category TGRF realizations using two thresholds on the GRFs from Figure 2.2. Blue (category 1), white (category 2), and red (category 3) with varying volume fractions. (A) Trending GRF (20% blue, 30% white, 50% red), (B) Non-stationary GRF (30% blue, 50% white, 20% red), (C) Anisotropic GRF (50% blue, 20% white, 30% red).

2.1.7 Extensions: Pluri-Gaussian Simulation

While TGRFs impose a natural ordering on categories through thresholding, **Pluri-Gaussian Simulation** (PGS) relaxes this constraint by utilizing multiple independent GRFs (Loc'h and Renard, 1992; Armstrong and Loc'h, 1994; Gustafson and Caers, 1997). This approach enables direct transitions between any pair of categories, better representing geological scenarios where, for example, sandstone and limestone can be adjacent without requiring an intermediate category. Another important aspect of pluri-Gaussian simulation is that it allows the different fields to have different structures, such that the facies and facies transitions can have different geometries. For instance, one GRF might have long-range correlations to control large-scale geological features, while another has short-range correlations to model local heterogeneities, providing flexibility in representing complex geological architectures.

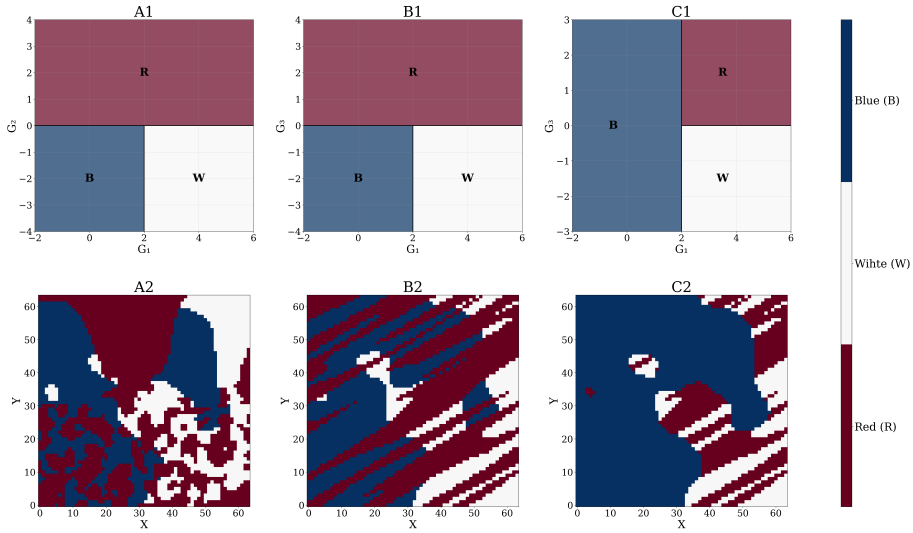


Figure 2.5: Pluri-Gaussian Simulation (PGS) examples. Top row (A1, B1, C1): Transition maps in Gaussian space showing how combinations of two independent GRFs (G_1, G_2) or (G_1, G_3) are mapped to three categories: Red (R), White (W), and Blue (B). **Bottom row (A2, B2, C2):** Corresponding PGS realizations demonstrating different spatial connectivity patterns. A2 results from combining the GRFs in Panel A) and B) from Figure 2.2 with the facies transition map in panel A1. B2 combines the GRFs in Panel A) and C) from Figure 2.2 with facies transition map in panel B1. C2 combines the same GRFs as B2 with the facies transition map from C1.

Figure 2.5 illustrates the PGS methodology through three different transition map configurations. Each configuration demonstrates how combinations of two independent GRFs can be partitioned to create categorical fields with equal volume fractions while allowing flexible spatial connectivity patterns that would be impossible with single-GRF truncation methods.

TGRFs and PGS have proven valuable for modeling categorical spatial structures across multiple disciplines. In petroleum reservoir modeling, they characterize sandstone-shale sequences where connectivity controls hydrocarbon flow paths and recovery efficiency. For groundwater studies, they delineate permeable units within clay sequences, critical for water resource management and contamination assessment. In CO_2 storage applications, they characterize caprock integrity where seal continuity determines long-term storage security. Agricultural applications include mapping soil types and crop classifications where spatial correlation affects yield patterns and farming strategies.

2.2 Markov Random Fields

While TGRFs model spatial dependence via an underlying continuous process, an alternative is to encode dependency directly on the grid. Markov Random Fields (MRFs) provide this complementary viewpoint. A MRF is a set of random variables having the Markov property which can be described on an undirected graph (Kindermann and Snell, 1980; Besag, 1974). In our case with $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$, we can let each variable x_i represent a node in a graph, and the connections between them describe dependence relations. This means that if we have no edge between two nodes x_j and x_k, they are independent given all other variables:

$$x_j \perp x_k | \mathbf{x}_{V\setminus\{j,k\}}, \tag{2.2.1}$$

where $\mathbf{x}_{V\setminus\{j,k\}}$ denotes all variables, except x_j, x_k. Since we are working with 2D grids, the edges are typically between cells that are in close proximity. For example, we can choose neighbors within a certain Manhattan distance on the grid. If we work on a 10×10 grid and pick neighbors of Manhattan distance 1, this creates a graph structure where each interior cell connects to its four orthogonal neighbors. With Manhattan distance 2, the connectivity increases significantly where each cell connects not only to its immediate neighbors but also to diagonal cells and those two steps away in cardinal directions, creating a denser graph with stronger spatial smoothing.

An important concept in graph theory is a clique. A clique is defined as a set of nodes such that there is an edge between any two nodes in the set. The maximal cliques are those cliques that cannot be extended by adding another node while maintaining the fully connected property. For the simplest case of first-order neighborhood, cliques are pairwise interactions. A second-order neighborhood construction is visualized in Figure 2.6 alongside it is maximal clique potentials in Figure 2.7 that are invariant under rotation and inversion (interchange 0s and 1s).

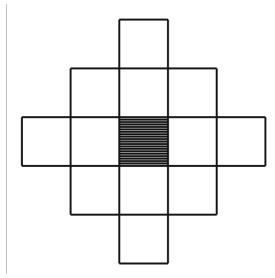


Figure 2.6: Second-order neighborhood structure in MRF construction.

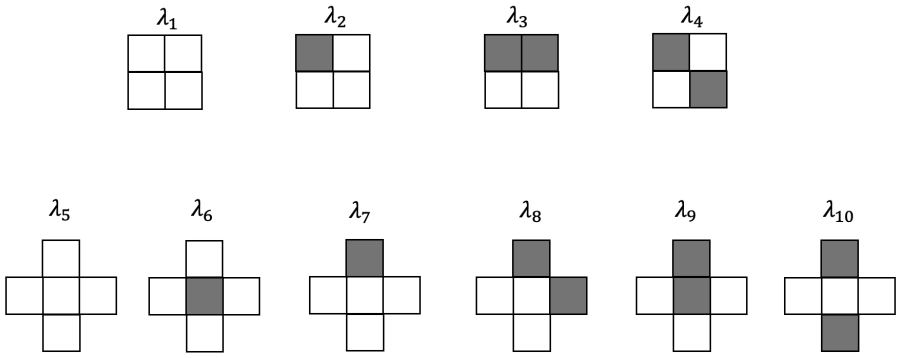


Figure 2.7: Maximal clique potentials in MRF model.

An important result is the Hammersley-Clifford theorem, which tells us that we can represent the joint probability mass function of any MRF as:

$$P(\mathbf{x}) = \frac{1}{Z} \exp\left(- \sum_{\Lambda \in \mathcal{C}} V_\Lambda(\mathbf{x}_\Lambda) \right), \tag{2.2.2}$$

where $V_\Lambda(\mathbf{x}_\Lambda)$ is the potential function of clique Λ, $\mathcal{C}$ is the set of all cliques and Z is the normalization constant. This allows us to decompose the field into individual terms over cliques where the log-probability can be expressed as the sum of log-potentials. This means that all we have to do to make a valid MRF is to define a set of cliques and their corresponding potentials (Besag, 1974; Tjelmeland and Besag, 1996). We denote the λ-values in Figure 2.7 the maximal clique potentials.

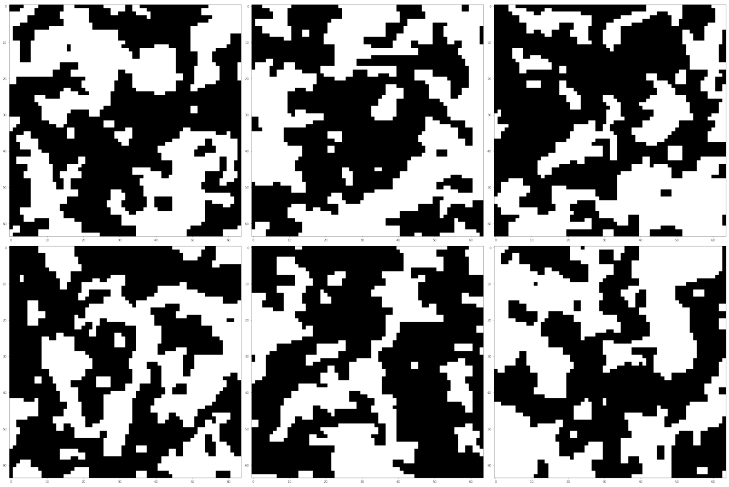


Figure 2.8: Markov Random Field realizations using a 12-template neighborhood structure.

Figure 2.8 shows examples of MRF realizations using the neigbhourhood structure in Figure 2.6. MRFs have the advantage over TGRFs that the probability distribution function can be made completely tractable. However, they are often more difficult to work with in practice because they require us to define many components: all cliques and potentials, making it sometimes difficult to control which geometries the prior model produces. Despite this complexity, MRFs provide explicit control over local spatial dependencies and have found applications in image processing, spatial statistics, and categorical field modeling.

2.3 Rule-based Models

Figure 2.1 illustrates the spectrum of geostatistical modeling approaches. The TGRFs and MRFs discussed in previous sections occupy the far left of this spectrum, prioritizing mathematical simplicity and computational efficiency. In this thesis, particularly in Papers I and II, we also work with rule-based models that occupy a middle position, offering enhanced geological realism while maintaining computational feasibility for conditioning problems.

Rule-based models encode geological knowledge through explicit rules governing spatial relationships between facies. Unlike the statistical approaches of TGRFs and MRFs, these models directly incorporate physical and stratigraphic principles to generate geologically plausible realizations. Consider a simple example of sequential construction for a shallow marine environment:

1. **Define base surface**: Generate initial topography $z_0(u, v)$ representing the depositional surface, where (u, v) denote horizontal coordinates

2. **Apply sea level**: For current sea level ℓ (meters above datum), classify locations based on water depth and assign facies $x_{u,v}$:

   - Deep marine: where $\ell - z_0(u, v) > d_{\text{deep}} \rightarrow$ deposit shale ($x_{u,v} = 1$)
   - Shallow marine: where $d_{\text{shallow}} < \ell - z_0(u, v) \leq d_{\text{deep}} \rightarrow$ deposit sandstone ($x_{u,v} = 2$)
   - Subaerial: where $\ell - z_0(u, v) \leq d_{\text{shallow}} \rightarrow$ no deposition ($x_{u,v} = 0$)

3. **Update surface**: $z_1(u, v) = z_0(u, v) + \Delta z(u, v)$ based on deposition/erosion rules, where Δz represents sediment thickness

4. **Iterate**: Repeat for changing sea levels and sediment supply over time steps $t = 1, ..., T$

This sequential process builds a 3D realization layer by layer, with each decision conditioned on previous states. The prior probability emerges implicitly from the rule cascade:

$$P(\mathbf{x}) = P(z_0) \prod_{t=1}^{T} P(\ell_t|\ell_{t-1}) P(\mathbf{x}_t|z_{t-1}, \ell_t) P(z_t|z_{t-1}, \mathbf{x}_t), \tag{2.3.1}$$

where $\mathbf{x}_t$ represents the facies field at time t, z_t is the topographic surface, and ℓ_t is the sea level. However, this probability cannot be evaluated directly because the rule-based transitions $P(\mathbf{x}_t|z_{t-1}, \ell_t)$ are defined algorithmically rather than analytically. We can sample from this distribution by forward simulation, but computing the probability of a specific realization would require tracking all possible paths that could lead to that configuration—a generally intractable problem.

Rule-based approaches have shown promising results in specific applications. In fluvial systems, event-based models simulate channel migration and avulsion to create realistic channel-belt architectures (Bridge and Leeder, 1992; Pyrcz et al., 2009). Object-based models place geometric shapes representing channels or lobes according to geological rules (Howell et al., 2008). Process-based forward modeling simulates physical equations governing sediment transport and deposition (Borgomano et al., 2020). However, it is important to note that while these methods can generate geologically realistic models, their application to conditioning problems remains challenging and computationally intensive compared to the widespread industrial use of TGRF and MPS methods.

The implicit nature of $P(\mathbf{x})$ in rule-based models presents unique challenges for Bayesian conditioning. The fundamental challenge with rule-based models

is that while generating unconditional realizations is straightforward—simply run the forward simulation—conditioning on data is exceptionally difficult. Proposals must respect the entire rule hierarchy, often requiring complex MCMC schemes that modify underlying parameters rather than facies directly. This asymmetry between easy unconditional generation and difficult conditioning makes rule-based models ideal candidates for a different approach: generate many unconditional realizations as training data for neural networks that can then learn to perform conditioning efficiently. This strategy, explored in Papers I and II, leverages the geological realism of rule-based models while sidestepping their conditioning challenges.

Bayesian Conditioning

Having established the spatial categorical models that serve as our priors in Chapter 2, we now address the central computational challenge: transforming a prior $P(\mathbf{x})$ into a posterior $P(\mathbf{x}|\mathbf{d})$ that honors observational data. This conditioning problem, which incorporates sparse localized measurements into spatial models, has been fundamental to geostatistics since its inception.

Sequential simulation methods provided early practical approaches for conditioning. Sequential Indicator Simulation (Journel, 1983) visited each unsampled location sequentially, drawing from locally estimated conditional distributions using indicator kriging. By transforming categorical variables into binary indicators and applying kriging to each, SIS could honor exact data while approximately reproducing spatial statistics. Despite widespread adoption, the method struggles to reproduce complex multi-point patterns and suffers from order-dependence artifacts (Emery, 2004).

Simulated annealing, introduced to geostatistics by Deutsch (1995), reformulated conditioning as an optimization problem. Starting from a random realization, the algorithm iteratively proposes modifications and accepts or rejects based on an objective function combining data misfit and pattern reproduction. The temperature parameter, gradually decreased during optimization, controls the trade-off between exploration and convergence. While capable of handling complex objective functions, simulated annealing provides only a single "optimal" realization rather than sampling from the posterior distribution.

The introduction of MCMC methods to geostatistics marked a paradigm shift. Early work by Farmer (1987) and Hegstad et al. (1994) demonstrated MCMC's potential, with Tjelmeland and Besag (1996) providing a comprehensive framework for categorical MRF models. Unlike previous approaches, MCMC provides samples from the true posterior distribution without requiring analytical expressions or approximations. The Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) constructs a Markov chain whose stationary distribution equals the target posterior, enabling rigorous Bayesian conditional sampling. Despite theoretical elegance, MCMC methods face practical challenges: slow convergence in high dimensions, difficulty traversing multimodal posteriors, and computational costs that scale poorly with model

size (Hansen et al., 2012).

Deep learning opened entirely new possibilities for conditioning. Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) learn to generate realistic realizations through adversarial training, with conditional variants (cGANs) incorporating data constraints (Mirza and Osindero, 2014). Early geostatistical applications demonstrated GANs' ability to reproduce complex geological patterns (Chan and Elsheikh, 2017; Mosser et al., 2017), but training instability and mode collapse limited their reliability for accurately representing the posterior sample space (Dupont et al., 2018).

Variational Autoencoders (VAEs) (Kingma and Welling, 2014) offered a more stable alternative by learning probabilistic mappings between realizations and latent representations. Conditioning could be performed by optimizing latent codes to match data constraints (Laloy et al., 2019). However, the Gaussian assumptions underlying VAEs often produce overly smooth realizations, failing to capture sharp facies boundaries critical in categorical models (Canchumuni et al., 2019).

More recent developments have addressed these limitations. Diffusion models (Ho et al., 2020; Song et al., 2021) achieve state-of-the-art generation quality through iterative denoising, with various conditioning strategies enabling flexible data integration (Lee et al., 2025). Vision Transformers (Dosovitskiy et al., 2020) leverage self-attention mechanisms to capture long-range dependencies while providing explicit access to conditional probabilities, facilitating uncertainty quantification (Yan et al., 2021).

It is relatively straightforward to generate unconditional realizations from complex geological priors, whether TGRFs, object-based models, or process-based simulations. However, conditioning these realizations on data typically requires custom MCMC implementations tailored to each prior model. If neural networks can learn to perform universal conditioning, meaning mapping from any data configuration to appropriate realizations, this opens entirely new possibilities for using sophisticated geological priors that would otherwise be impractical to condition. This vision motivates our exploration of neural approaches alongside classical MCMC, particularly in Paper IV where we investigate whether transformers can truly capture the complex distributions needed for geological modeling.

3.1 Markov Chain Monte Carlo

MCMC methods provide the theoretical foundation for rigorous Bayesian conditional sampling. The key insight is that we can construct a Markov chain whose stationary distribution equals our target posterior $P(\mathbf{x}|\mathbf{d})$, enabling sampling without computing the intractable normalization constant.

While we present MCMC using categorical field notation $\mathbf{x}$ for consistency, it is important to note that many geostatistical applications perform MCMC in a parameter space that then maps to the categorical field. Object-based models update object parameters (location, size, orientation) rather than grid cells directly. Truncated Gaussian methods perform MCMC on the underlying continuous Gaussian field, which is then thresholded to produce categories. The principles remain the same, but the state space and proposal mechanisms differ substantially.

For spatial categorical models, we seek samples from:

$$P(\mathbf{x}|\mathbf{d}) = \frac{P(\mathbf{d}|\mathbf{x})P(\mathbf{x})}{\sum_{\mathbf{x}'} P(\mathbf{d}|\mathbf{x}')P(\mathbf{x}')}. \tag{3.1.1}$$

The foundation of MCMC is the detailed balance condition. For a Markov chain with transition kernel $p(\mathbf{x}^*|\mathbf{x})$, detailed balance requires:

$$P(\mathbf{x}|\mathbf{d})\,p(\mathbf{x}^*|\mathbf{x}) = P(\mathbf{x}^*|\mathbf{d})\,p(\mathbf{x}|\mathbf{x}^*). \tag{3.1.2}$$

This condition ensures that the target posterior $P(\mathbf{x}|\mathbf{d})$ is the stationary distribution of the Markov chain—if we run the chain long enough, samples will be drawn from the desired distribution.

The Metropolis-Hastings algorithm constructs a transition kernel by decomposing it into a proposal distribution $q(\mathbf{x}^*|\mathbf{x})$ and an acceptance probability $\alpha(\mathbf{x}, \mathbf{x}^*)$:

$$p(\mathbf{x}^*|\mathbf{x}) = q(\mathbf{x}^*|\mathbf{x})\alpha(\mathbf{x}, \mathbf{x}^*) \quad \text{for } \mathbf{x}^* \neq \mathbf{x}. \tag{3.1.3}$$

The key insight is choosing α to satisfy detailed balance while maximizing acceptance rates (Algorithm 3.1).

Algorithm 3.1 Metropolis-Hastings Algorithm

Initialize $\mathbf{x}^{(0)}$ to satisfy data constraints
for $t = 1$ to T **do**
    Select location i to update (randomly or systematically)
    Propose new value $x_i^* \sim q(x_i^* | x_i^{(t-1)})$
    Form $\mathbf{x}^*$ where $x_j^* = x_j^{(t-1)}$ for all $j \neq i$
    Calculate acceptance ratio:

$$\alpha(\mathbf{x}^{(t-1)}, \mathbf{x}^*) = \min\left(1, \frac{P(\mathbf{x}^*|\mathbf{d})\, q(x_i^{(t-1)} | x_i^*)}{P(\mathbf{x}^{(t-1)}|\mathbf{d})\, q(x_i^* | x_i^{(t-1)})}\right)$$

    Draw $u \sim \text{Uniform}(0, 1)$
    if $u < \alpha(\mathbf{x}^{(t-1)}, \mathbf{x}^*)$ **then**
        Accept: $\mathbf{x}^{(t)} = \mathbf{x}^*$
    else
        Reject: $\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)}$
    end if
end for

The specific form of the acceptance ratio in Algorithm 3.1 is precisely designed to satisfy the detailed balance equation. For single-site updates where states differ only at location i, this choice of $\alpha(\mathbf{x}^{(t-1)}, \mathbf{x}^*)$ ensures the Markov chain converges to the target posterior distribution.

The proposal distribution $q(x_i^* | x_i)$ for individual sites critically affects efficiency. For categorical fields, a simple but inefficient approach proposes new categories uniformly at random. More sophisticated proposals leverage problem structure—proposing geologically plausible modifications or using gradient information where available.

Gibbs sampling provides an important special case where proposals are always accepted. The key idea is then to sample each variable from its conditional distribution given all others:

$$x_i^{(t)} = x_i^* \sim P(x_i | \mathbf{x}_{-i}^{(t-1)}, \mathbf{d}), \tag{3.1.4}$$

where $\mathbf{x}_{-i}$ denotes all variables except x_i. For MRF priors with hard data constraints at locations $\mathcal{D}$, the conditional distribution becomes:

$$P(x_i | \mathbf{x}_{-i}, \mathbf{d}) \propto \begin{cases} \exp\left(-\sum_{\Lambda \in \mathcal{C}_i} V_\Lambda(\mathbf{x}_\Lambda)\right) & \text{if } i \notin \mathcal{D} \\ \mathbb{I}[x_i = d_i] & \text{if } i \in \mathcal{D}, \end{cases} \tag{3.1.5}$$

where $\mathcal{C}_i$ is the set of all cliques containing location i, and V_Λ are the clique potentials.

Despite theoretical guarantees, MCMC faces significant practical challenges. Convergence can require millions of iterations for complex models, multimodal posteriors trap chains in local modes, and each iteration requires evaluating the likelihood and prior—computationally expensive for sophisticated geological models. While advances like parallel tempering (Earl and Deem, 2005) and adaptive proposals (Haario et al., 2001) help, the tension remains: geological realism requires complex priors that make MCMC increasingly difficult.

3.2 Diffusion Models

Diffusion models transform the conditioning problem into a denoising task, learning to iteratively refine noisy samples into valid realizations that honor data constraints. The approach rests on a surprisingly simple principle: if we can learn to reverse a gradual noising process, we can generate new samples (Ho et al., 2020).

The forward process progressively adds Gaussian noise to data. Starting with a clean image $\mathbf{x}_0$, we add small amounts of noise at each timestep t according to a variance schedule β_t (typically increasing from $\beta_1 \approx 0.0001$ to $\beta_T \approx 0.02$). After many steps, the original structure is completely destroyed, leaving only Gaussian noise. Mathematically, the forward diffusion process q is defined as

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\big(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\big), \tag{3.2.1}$$

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\big(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\big), \tag{3.2.2}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ is the cumulative product of α.

The second equation allows us to sample $\mathbf{x}_t$ directly from $\mathbf{x}_0$ by:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \tag{3.2.3}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is standard Gaussian noise.

The reverse process learns to denoise step by step. A neural network ϵ_θ (where θ denotes the network parameters) is trained to predict the noise ϵ that was added to create $\mathbf{x}_t$ from $\mathbf{x}_0$. Given a noisy image $\mathbf{x}_t$, the network estimates this noise component, allowing us to take a small step back toward the clean image. The training objective is simply to minimize the difference between the true noise added and the network's prediction:

$$\mathcal{L} = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}[||\epsilon - \epsilon_\theta(\mathbf{x}_t, t)||^2]. \tag{3.2.4}$$

Correspondingly, the parameterized reverse transition is modeled as

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right), \sigma_t^2 \mathbf{I}\right), \tag{3.2.5}$$

where σ_t^2 is typically set to β_t or learned concurrently with ϵ_θ.

For geological applications, working directly in pixel space is inefficient. Geological images contain complex heterogeneity and long-range dependencies—channels that snake across the entire domain, layer boundaries that extend for kilometers. Latent diffusion models (LDMs) address this by first learning a compressed representation of the geological patterns (Rombach et al., 2022). Ideally, an autoencoder learns a compact latent representation that retains the dominant geological structures while filtering out redundant details—although some fine-scale information can still be lost in practice. Performing diffusion in this latent space substantially reduces computational cost and, when the encoding is sufficiently faithful, still allows the model to capture complex geological patterns.

The elegance of conditioning emerges through cross-attention mechanisms. As shown in Figure 3.1, conditioning data $\mathbf{d} = \mathbf{x}_c$ is encoded into a latent representation $\mathbf{z}_c$ and incorporated at each denoising step through cross-attention layers. This allows the model to "pay attention" to the conditioning data throughout the generation process, naturally steering the denoising toward realizations that honor the observations. The same trained model can handle arbitrary conditioning patterns—vertical wells, deviated wells, or scattered observations—without retraining.

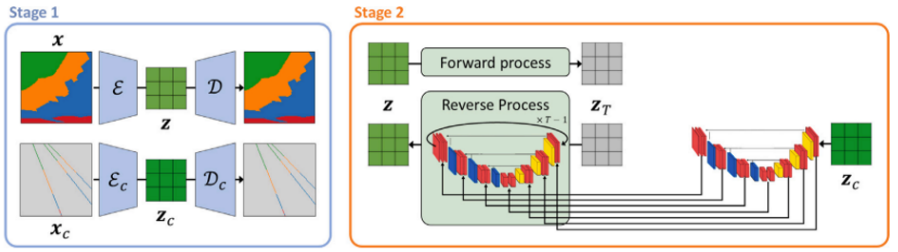


Figure 3.1: Two-stage training for conditional LDMs. Stage 1 trains autoencoders to compress spatial categorical fields into an efficient latent representation where geological structures are preserved. Stage 2 performs diffusion in this latent space, with cross-attention mechanisms incorporating conditioning information $\mathbf{z}_c$ at each denoising step. The latent space representation is particularly effective for geological images due to their inherent structure and long-range dependencies.

Alternative conditioning strategies exist, each with different trade-offs. Replacement methods directly enforce hard constraints by substituting known values after each denoising step. Gradient-based guidance (Dhariwal and Nichol, 2021) modifies the denoising direction using the gradient of the data likelihood. These approaches trade off between exact data matching and maintaining the learned distributional properties.

However, diffusion models face their own challenges. Training requires extensive computational resources and large datasets. The iterative denoising process, typically 50-1000 steps, makes generation slower than single-forward-pass methods. Most critically for categorical data, the Gaussian noise assumption may not be ideal for discrete variables, requiring careful adaptation. While these models excel at capturing visual patterns, ensuring they properly represent the full posterior distribution, including rare but important patterns while also excluding impossible patterns, remains an open challenge.

3.3 Vision Transformers

Vision Transformers (ViTs) represent a fundamentally different approach to spatial modeling by treating images as sequences of patches (Dosovitskiy et al., 2020). Originally devised for natural language processing (Vaswani et al., 2023), they have recently shown strong potential for spatial categorical data because the self-attention mechanism can, in principle, capture long-range dependencies—one of the key ingredients behind the success of large language models. Nonetheless, a ViT does not automatically recover every geological nuance; its fidelity ultimately depends on model capacity, training data diversity, and how well the patch representation aligns with the underlying structures.

This patch-based reformulation enables powerful attention mechanisms while also granting something rarely available in convolutional architectures: explicit access to autoregressive conditional probabilities.

The architecture divides the spatial field into patches and models the joint distribution autoregressively:

$$P(\mathbf{x}|\mathbf{d}) = P(x_1|\mathbf{d}), P(x_2|x_1, \mathbf{d}), \ldots, P(x_N|x_1, \ldots, x_{N-1}, \mathbf{d}). \tag{3.3.1}$$

At first glance, this factorization seems intractable—the number of possible configurations of $x_1, \ldots, x_{N-1}$ far exceeds any reasonable training set. The key insight is that transformers share parameters across all positions, allowing the network to represent each conditional distribution $P(x_i|x_1, \ldots, x_{i-1}, \mathbf{d})$ with a single set of weights.

During training, illustrated in Figure 3.2, the model employs a masked-prediction strategy. Random patches from unconditional prior samples (e.g., TGRF realizations) are masked, and the network learns to predict these masked patches given only the visible ones. This is achieved by minimizing the categorical cross-entropy loss between predicted and true patch values, computed only at masked positions. The self-attention mechanism enables the network to dynamically focus on relevant parts of the visible field, learning spatial patterns and dependencies rather than memorizing specific configurations. Consequently, after exposure to thousands of geological realizations, a well-trained

transformer can often generate plausible patterns for configurations it has never seen, indicating that it has internalized broad spatial relationships—but it may still misrepresent rare or extremely fine-scale features.

For conditioning, the autoregressive structure provides natural flexibility. During inference, the model generates patches sequentially while respecting data constraints. Data-aligned patches are simply fixed during generation. For partial observations, we adjust the output probabilities:

$$\tilde{p}(x_i = k|\mathbf{x}_{<i}, \mathbf{d}) = \begin{cases} 0 & \text{if } k \text{ conflicts with } \mathbf{d}, \\ \frac{p(x_i=k|\mathbf{x}_{<i})}{\sum_{j\in\mathcal{V}} p(x_i=j|\mathbf{x}_{<i})} & \text{otherwise,} \end{cases} \tag{3.3.2}$$

where $\mathbf{x}_{<i} = (x_1, ..., x_{i-1})$ and $\mathcal{V}$ is the set of valid categories given constraints.

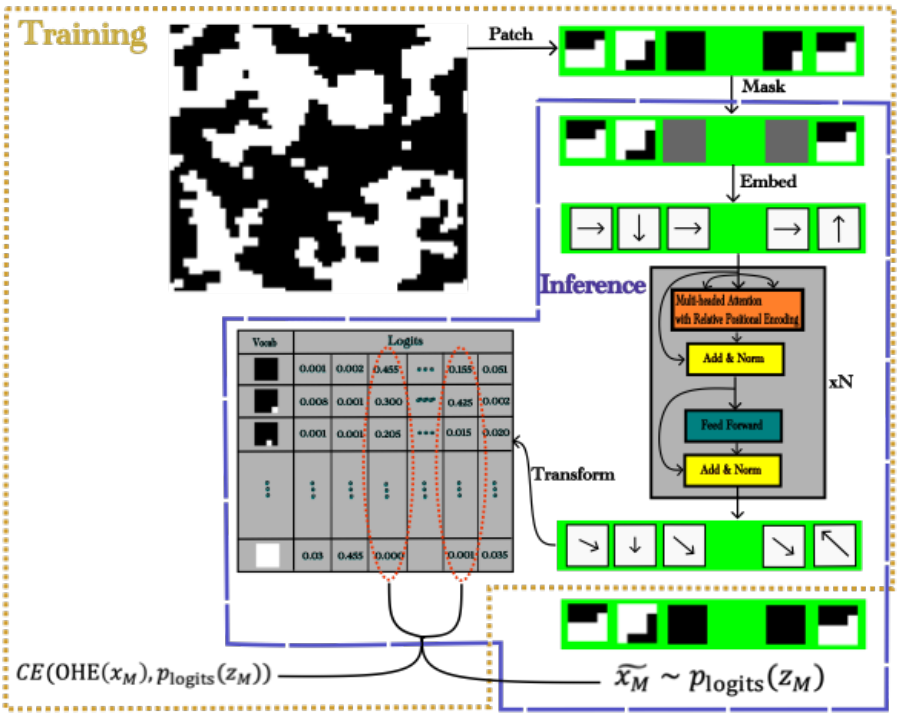


Figure 3.2: Vision Transformer architecture for spatial categorical modeling. During training (top), random patches are masked and the model learns to predict them from visible context. During inference (bottom), the model generates patches autoregressively while respecting data constraints through logit adjustment.

The explicit probability modeling distinguishes ViTs from other generative approaches. We can evaluate exact likelihoods, compare realizations probabilistically, and quantify uncertainty through the conditional distributions. This

transparency is particularly valuable for understanding model behavior, as explored in Paper IV.

The primary limitation remains computational scaling. Attention mechanisms require $O(N^2)$ operations where N is the sequence length, constraining applicable grid sizes. For the 64×64 grids used in our experiments, this is manageable, but scaling to millions of cells remains challenging. However, if transformers can learn universal conditioning strategies that generalize across different prior types, the investment in training could enable conditioning of geological models that would otherwise require prohibitive custom MCMC development. This potential for universal conditioning rather than raw computational speed represents the true promise of neural approaches for geostatistical modeling.

Image Quality Assessment

The evaluation of spatial categorical models presents a fundamental challenge: how do we quantify the similarity between complex spatial patterns? Categorical fields require metrics that capture the essence of spatial structure—the connectivity of sand bodies, the sinuosity of channels, or the clustering of similar facies. The development of appropriate comparison metrics has paralleled the evolution of geostatistical modeling itself, progressing from simple marginal statistics to sophisticated measures that attempt to capture perceptual and geological realism.

Early geostatistical practice relied heavily on first-order statistics—volume fractions or category proportions—to validate models (Journel, 1983). While necessary for ensuring basic consistency, these marginal measures proved woefully inadequate for distinguishing between radically different spatial arrangements that happened to share similar proportions. This limitation drove the adoption of two-point statistics, particularly the variogram, as the primary tool for characterizing spatial structure (Matheron, 1973; Krige, 1951). The indicator variogram for categorical data (Journel, 1983) enabled comparison of spatial correlation structures, though it too failed to capture many important patterns.

The introduction of multiple-point statistics marked a recognition that geological patterns require higher-order characterization (Guardiano and Srivastava, 1993). Methods for extracting and comparing pattern frequencies—whether through template matching (Arpat and Caers, 2007), cluster analysis (Scheidt and Caers, 2009), or wavelet decomposition (Gloaguen and Dimitrakopoulos, 2009)—attempted to capture the multi-scale, multi-point nature of geological structures. Yet the curse of dimensionality limited these approaches to relatively small templates or specific pattern types.

Recent years have seen a proliferation of feature-based and perceptual metrics borrowed from computer vision. The Structural Similarity Index (SSIM) (Wang et al., 2003), originally designed for image quality assessment, found applications in comparing geological realizations. Connectivity metrics (Renard and Allard, 2013) targeted specific aspects of geological importance. Meanwhile, the machine learning community contributed learned metrics based on deep feature extraction (Mosser et al., 2017), promising more holistic compar-

isons aligned with human perception.

This chapter reviews the hierarchy of image comparison methods available for spatial categorical fields, examining their strengths, limitations, and appropriate domains of application. Section 4.1 addresses first-order statistics and their fundamental limitations. Section 4.2 explores second-order methods, particularly variogram-based approaches. Section 4.3 examines higher-order statistics and pattern-based metrics. Section 4.4 discusses composite and feature-based measures that attempt to provide more complete characterizations. Rather than seeking a single "best" metric, we argue for using multiple complementary measures that together span the space of relevant structural properties.

4.1 First-Order Statistics

First-order statistics summarize the marginal distribution of pixel values without considering spatial relationships. For a spatial categorical field $\mathbf{x} = [x_1, ..., x_N]^T$ with $x_i \in \{1, 2, ..., C\}$, the most fundamental measures are the category proportions:

$$p_k = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(x_i = k), \tag{4.1.1}$$

where $\mathbb{I}(\cdot)$ is the indicator function. For a binary field ($C = 2$) this reduces to the mean $\mu = p_2$, because the two proportions must sum to 1. The variance

$$\sigma^2 = \mu(1 - \mu),$$

is functionally determined by μ and therefore provides no additional information. For images with multiple categories, mean and variance of category indices are often not meaningful, and the full proportion vector $\mathbf{p} = [p_1, ..., p_C]$ should be used instead.

Figure 4.1 illustrates the fundamental limitation of first-order statistics through three different spatial models: a TGRF, a MRF, and a fluvial MPS. The TGRF and MRF have nearly identical first-order statistics ($\mu = 0.41$ vs 0.43, $\sigma^2 = 0.24$ vs 0.25) yet exhibit distinctly different spatial structures—the TGRF shows smooth, blob-like patterns with irregular boundaries, while the MRF displays more blocky, regular structures. This demonstrates that first-order statistics cannot distinguish between fundamentally different spatial organizations. The fluvial MPS, with its characteristic channel features, has noticeably different first-order statistics ($\mu = 0.25$, $\sigma^2 = 0.19$), yet even knowing these values tells us nothing about the elongated, connected channel structures that define this realization. Whether the first-order statistics are similar or different, they fail to capture the spatial patterns that often matter most in applica-

tions—connectivity, object shapes, and correlation structures remain invisible to these marginal measures.

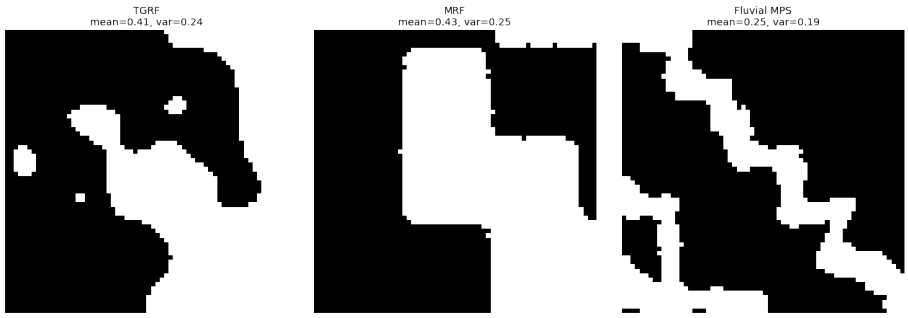


Figure 4.1: Three different spatial models with similar first-order statistics. Each realization is shown in binary form with its mean volume fraction (μ) and variance ($\sigma^2 = \mu(1-\mu)$). Despite similar marginal distributions, the spatial patterns differ dramatically.

When comparing two categorical fields, the Mean Squared Error (MSE) provides a pixel-wise measure of difference:

$$\text{MSE}(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2. \tag{4.1.2}$$

For binary fields where $x_i, y_i \in \{0, 1\}$, this simplifies to the proportion of mismatched pixels:

$$\text{MSE}_{\text{binary}}(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(x_i \neq y_i). \tag{4.1.3}$$

For general categorical data where category labels are arbitrary, treating them as numerical values in MSE calculations can be misleading. A more appropriate measure is the categorical MSE:

$$\text{MSE}_{\text{categorical}}(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{C} (\mathbb{I}(x_i = k) - \mathbb{I}(y_i = k))^2. \tag{4.1.4}$$

This expands the comparison to indicator functions for each category, avoiding arbitrary numerical assignments. For normalized comparisons between different image pairs, the MSE is often reported as a percentage of the maximum possible error, which for categorical data equals 2 (when all pixels differ).

Another useful measure for comparing categorical distributions is the Jensen-Shannon Divergence (JSD) (Lin, 1991), which provides a symmetric measure of similarity between two probability distributions:

$$\text{JSD}(\mathbf{p}, \mathbf{q}) = \frac{1}{2} D_{KL}(\mathbf{p} || \mathbf{m}) + \frac{1}{2} D_{KL}(\mathbf{q} || \mathbf{m}), \tag{4.1.5}$$

where $\mathbf{m} = \frac{1}{2}(\mathbf{p} + \mathbf{q})$ is the average distribution and D_{KL} is the Kullback-Leibler divergence. Unlike MSE, JSD is bounded between 0 and 1 and provides a true metric for probability distributions (Lin, 1991).

While MSE provides a quantitative measure of pixel-wise differences, it shares the fundamental limitation of all first-order statistics: it cannot distinguish between images with different spatial structures but similar pixel distributions. A random permutation of an image will have high MSE compared to the original, even though both share identical statistical properties. This limitation motivates higher-order approaches that consider spatial relationships (Moran, 1950; Leuangthong et al., 2004).

4.2 Second-Order Statistics

Many spatial priors, most notably GRFs and TGRFs, are completely specified by their mean and covariance and are therefore second-order models. Second-order statistics capture pairwise spatial relationships between pixels. In geostatistics, the variogram is the primary tool for characterizing spatial correlation:

$$\gamma(\mathbf{h}) = \frac{1}{2|N(\mathbf{h})|} \sum_{(i,j)\in N(\mathbf{h})} |x_i - x_j|^2, \tag{4.2.1}$$

where $N(\mathbf{h})$ is the set of pixel pairs separated by lag vector $|\mathbf{h}|$, and $|N(\mathbf{h})|$ is the number of such pairs. For categorical data, the indicator variogram is often used, defined as $\gamma_k(\mathbf{h}) = \frac{1}{2|N(\mathbf{h})|} \sum_{(i,j)\in N(\mathbf{h})} |\mathbb{I}(x_i = k) - \mathbb{I}(x_j = k)|^2$.

While the variogram is the traditional geostatistical tool, the full covariance matrix provides an alternative representation of second-order spatial relationships. For a spatial field with locations $\mathbf{s}_1, ..., \mathbf{s}_N$, the covariance matrix Σ has entries:

$$\Sigma_{ij} = \mathrm{Cov}(x(\mathbf{s}_i), x(\mathbf{s}_j)). \tag{4.2.2}$$

The variogram and covariance are intrinsically linked through the relationship:

$$\gamma(\mathbf{h}) = \Sigma(0) - \Sigma(\mathbf{h}), \tag{4.2.3}$$

where $\Sigma(0)$ is the variance and $\Sigma(\mathbf{h})$ is the covariance at lag $|\mathbf{h}|$. This connection means that stationary fields with known variance can be fully characterized by either representation.

Figure 4.2 shows the isotropic empirical variograms for our three example realizations. The fluvial MPS (red) exhibits rapid decay, reflecting its narrow channel widths—pixels become decorrelated over short distances due to the alternating channel and background structure. In contrast, the TGRF (blue) and MRF (green) share more closely two-point correlations, with slower decay indicating larger-scale spatial structures. This demonstrates both the power and

limitation of second-order statistics: while they successfully discriminate the channelized model from the others, they cannot distinguish between the TGRF and MRF very well despite their visually distinct patterns.

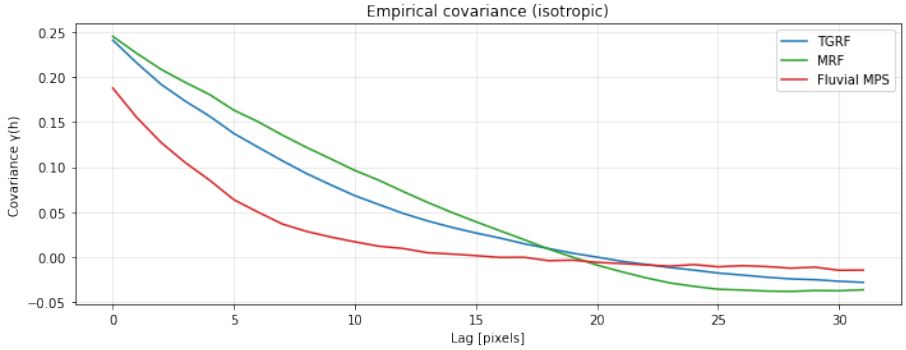


Figure 4.2: Isotropic empirical variograms of the three realizations from Figure 4.1. TGRF (blue), MRF (green), and fluvial MPS (red).

Comparing variograms between images can be done through the sum of squared differences across all lag distances: $D_{\text{var}}(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{h}} |\gamma_{\mathbf{x}}(\mathbf{h}) - \gamma_{\mathbf{y}}(\mathbf{h})|^2$. While variograms capture important spatial structure and are widely used in geostatistical applications, they cannot distinguish between patterns that differ in their higher-order properties. For instance, a checkerboard pattern, randomly placed squares of the same size, and certain arrangements of stripes can all produce identical variograms despite their vastly different visual appearance and connectivity properties. This limitation becomes particularly important when dealing with complex geological structures that require higher-order statistics to characterize adequately (Honarkhah and Caers, 2010; Boisvert et al., 2010; Tan et al., 2014).

4.3 Higher-Order Statistics

To capture more complex spatial patterns, higher-order statistics examine configurations of multiple pixels simultaneously. The most common approach uses template scanning with n-point histograms, where a template $\mathcal{T}$ defines a local neighborhood configuration, typically the 3×3 or 5×5 region around a central pixel.

By scanning the entire image, we count occurrences of specific patterns:

$$f_{\mathcal{T}}(\mathbf{p}) = \frac{1}{N_{\mathcal{T}}} \sum_{i \in \mathcal{I}_{\mathcal{T}}} \mathbb{I}(\mathbf{x}_{\mathcal{T}_i} = \mathbf{p}), \tag{4.3.1}$$

where $\mathcal{I}_{\mathcal{T}}$ is the set of all valid template center locations (i.e., locations where

the template fits entirely within the image), $\mathbf{x}_{\mathcal{T}_i}$ is the configuration at location i, $\mathbf{p}$ is a specific pattern, and $N_{\mathcal{T}} = |\mathcal{I}_{\mathcal{T}}|$ is the total number of template locations.

The computational challenge is severe: for a 3×3 binary template, there are $2^9 = 512$ possible patterns. Figure 4.3 shows the relative frequencies of the twenty most common patterns for each of our three example realizations. The pattern indices (P0, P1, ...) correspond to the 512 possible binary configurations, sorted in descending order of probability independently for each image—meaning P0 represents the most frequent pattern in each realization, but these may be different actual configurations.

The pattern distributions reveal structural differences invisible to lower-order statistics. Both the TGRF and MRF concentrate their probability mass on blob-like, isotropic patterns—configurations where pixels of the same value cluster together without preferential direction. In stark contrast, the fluvial MPS favors elongated templates aligned with channel directions, reflecting the anisotropic nature of channel systems. These differences in local pattern spectra expose the higher-order structural contrasts that distinguish these models, demonstrating why variogram-based approaches fail to differentiate between TGRF and MRF despite their distinct visual characteristics.

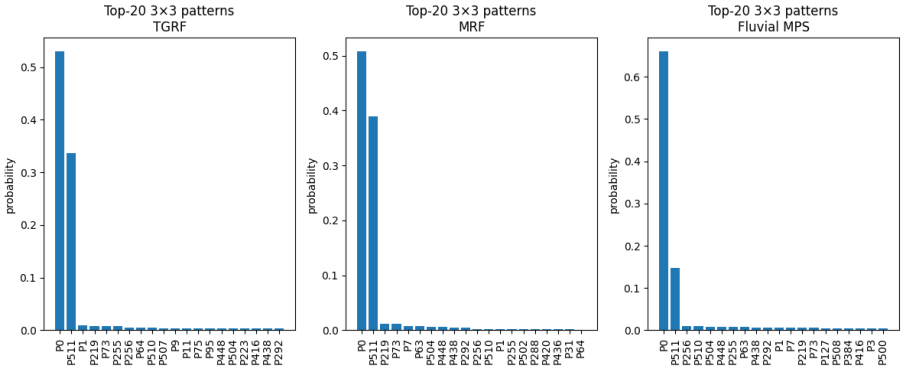


Figure 4.3: Relative frequency of the twenty most common 3×3 binary templates in each realization. Pattern indices are sorted independently for each image in descending order of frequency.

Beyond template-based approaches, other higher-order statistics include third-order spatial cumulants, which capture three-point interactions and can reveal asymmetries in spatial patterns invisible to second-order methods (Mustapha and Dimitrakopoulos, 2010). These statistics are particularly valuable for distinguishing between processes that generate similar two-point correlations but differ in their higher-order spatial relationships.

For larger templates or multi-category data, the pattern space becomes intractable. This has motivated several dimensionality reduction approaches,

including pattern clustering using techniques like multi-dimensional scaling or hierarchical clustering to group similar patterns, adaptive templates that select template size based on local entropy or information content, and filter banks that apply predefined filters such as Gabor or wavelets to extract specific features. Despite these advances, template-based methods remain computationally intensive and can be difficult to interpret, particularly when dealing with large-scale structures that extend beyond the template size (Boisvert et al., 2010; Mariethoz and Caers, 2014; Tahmasebi, 2018). No low-dimensional statistic can represent all information in an image class; the sheer diversity of valid realizations precludes such compression. In practice the goal is a useful rather than complete summary.

4.4 Feature Summary Statistics

Modern approaches to image comparison often combine multiple statistics into composite metrics that attempt to capture perceptual or structural similarity. While these methods recognize that no single statistic can adequately characterize the complexity of spatial categorical fields, they also represent a fundamental trade-off: condensing multidimensional information into scalar values inevitably discards potentially important distinctions. This dimensionality reduction can be viewed both positively as providing interpretable summary measures and critically, as oversimplifying complex spatial relationships that might be better represented by multiple complementary statistics.

The Structural Similarity Index (SSIM) (Wang et al., 2003) represents one of the most successful composite metrics, combining luminance, contrast, and structure comparisons:

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_{\mathbf{x}}\mu_{\mathbf{y}} + c_1)(2\sigma_{\mathbf{xy}} + c_2)}{(\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2 + c_1)(\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2 + c_2)}, \tag{4.4.1}$$

where μ_x, μ_y is the mean pixel intensity of image x and y respectively, $\sigma_{\mathbf{xy}}$ is the covariance between images, and c_1, c_2 are stability constants.

Originally designed for continuous-valued images in computer vision applications, SSIM has gained widespread adoption due to its alignment with human perceptual judgments. Designed for pixel-level correspondence in natural images, SSIM struggles when absolute location is irrelevant but internal structure matters, for example recognising a cat regardless of where it appears in the frame, or identifying channelised facies in geology. Its lack of rotational and translational invariance, together with the requirement that images share the same resolution, make SSIM ill-suited for many categorical tasks (Brunet et al., 2012; Sampat et al., 2009).

Recognizing these limitations, numerous variants have been proposed. Multi-Scale SSIM (MS-SSIM) (Wang et al., 2003) addresses the single-scale limitation by computing SSIM at multiple resolutions and combining the results, providing better alignment with human perception across different viewing distances. Complex Wavelet SSIM (CW-SSIM) (Sampat et al., 2009) achieves limited translation invariance through complex wavelet transforms, making it more robust to small spatial shifts.

Despite these improvements, fundamental challenges remain. All SSIM-based metrics assume pixel-wise correspondence and struggle with non-aligned comparisons. For geological applications where patterns matter more than absolute positions—such as comparing channel systems or facies distributions—these limitations motivate alternative approaches that focus on structural properties rather than pixel-level similarity. This recognition has driven the development of distribution-based metrics, morphological measures, and the resolution-invariant approaches explored in Paper V of this thesis.

Summary of Contributions

The PhD work have resulted in five papers that are summarized next. In addition to this, contributions have been presented at the following international conferences:

- IAMG 2022, Nancy, France
- IAMG 2023, Trondheim, Norway
- Petroleum Geostatistics 2023, Porto, Portugal
- Geostats 2024, Azores, Portugal

There have further been seminar presentations both at NTNU and during my 6-month visit at UT Austin.

Implementation and data science are also important elements of 21st century statistics, and contributions in this regard are posted on repositories on GitHub: https://github.com/OscarOvanger.

5.1 Paper 1: Addressing Configuration Uncertainty in Well Conditioning for a Rule-Based Model

Authors: Oscar Ovanger, Jo Eidsvik, Jacob Skauvold, Ragnar Hauge, Ingrid Aarnes
Published in: *Mathematical Geosciences* (2024) 56:1763–1788
DOI: 10.1007/s11004-024-10144-7

Motivation

Non-vertical wells often intersect the same bedset more than once, so the sequence of bedsets recorded in the log is ambiguous. Treating that sequence, called the well configuration, as fixed can underestimate uncertainty. The paper shows how to model configuration ambiguity explicitly within a simple shore-face bedset model for categorical facies variables.

Approach

Bedsets are stacked sequentially and controlled by progradation and aggradation parameters. For any set of well picks the algorithm lists all admissible configurations, evaluates a likelihood that mixes equality (intersection) and inequality (interval) constraints, and applies a Laplace approximation for the continuous parameters together with Monte Carlo sampling over configurations. Conditional realisations of bedsets and their facies configurations are obtained by averaging over configurations weighted by their posterior probability.

Contributions

The study (i) formulates a probabilistic model that separates continuous geology parameters from discrete configurations, (ii) derives a closed-form mixed-constraint likelihood that can be computed bedset by bedset, (iii) introduces a Laplace–Monte Carlo routine that gives configuration probabilities without exhaustive sampling, and (iv) provides reference results that can be used to test future conditioning algorithms.

Findings

In a synthetic three-bedset example the method recovers the true configuration distribution; the most likely path appears about 56% of the time, and low-probability paths are still represented. Ignoring configuration uncertainty makes bedset-boundary variance and well-length statistics too narrow, which could bias flow predictions.

Future work

Extending the approach to many bedsets, multiple wells and full 3-D grids will need path-pruning or heuristic search. Adding erosion, pinch-out and variable shoreline trajectories would improve realism. Coupling the configuration engine with downstream facies or flow simulators is a natural next step.

5.2 Paper 2: Latent Diffusion Model for Conditional Reservoir Facies Generation

Authors: Daesoo Lee, Oscar Ovanger, Jo Eidsvik, Erlend Aune, Jacob Skauvold, Ragnar Hauge
Published in: *Computers & Geosciences* (2025)
DOI: 10.1016/j.cageo.2024.105750

Motivation

Generative adversarial networks have been used for facies modelling, yet they often struggle with stable training and accurate honouring of well data. Diffusion models have shown reliable synthesis in computer vision, so we test whether a LDM can improve conditioning accuracy for facies grids.

Approach

The study trains a two-stage LDM on 5 000 synthetic 2-D facies images of size 128×128. Stage 1 learns an encoder–decoder that compresses one-hot facies grids to a latent space. Stage 2 trains a denoising network that maps latent noise to conditional samples. Well observations are injected through a cross-attention layer, and an extra loss term penalises any change to observed cells. A conditional GAN with the same dataset and conditioning scheme serves as baseline.

Contributions

The study introduces a diffusion-based workflow for conditional facies generation. It adds a simple preservation loss that directly targets conditioning accuracy. Additionally, it provides an open baseline comparison against a re-implemented conditional GAN.

Findings

The LDM keeps more than 99.9 % of conditioning cells intact, while the GAN misses about ten percent. First- and second-order statistics of the diffusion samples match those of the training set, and training in latent space fits on a single GPU for the grid size tested.

Future work

The current test case is limited to 2-D shoreline facies with a single grid resolution. Next steps include extending the method to 3-D models, larger grids

and more varied conditioning patterns, as well as evaluating flow-dependent metrics.

5.3 Paper 3: A Statistical Study of Latent Diffusion Models for Geological Facies Modeling

Authors: Oscar Ovanger, Daesoo Lee, Jo Eidsvik, Ragnar Hauge, Jacob Skauvold, Erlend Aune
Published in: *Mathematical Geosciences* (2025)
DOI: 10.1007/s11004-025-10178-5

Motivation

LDMs can generate realistic facies images, yet little is known about how their sample distribution compares with traditional geostatistical priors. This study benchmarks an LDM against a TGRF reference, moving the evaluation beyond visual inspection to first-, second- and higher-order statistics.

Approach

The LDM is trained on 5000 synthetic 128×128 facies grids produced by a TGRF. Two data sets are used: (i) a shoreface case with a vertical trend and one well column of conditioning data; (ii) a laterally heterogeneous case without trend, generated with three different covariance kernels and forty random conditioning points. For each case the study draws 1 000 unconditional and conditional samples from the LDM and from the exact TGRF posterior. Metrics include cell-wise probabilities, volume fractions, Jensen–Shannon divergence, transiograms, sub-grid pattern counts and third-order cumulants.

Contributions

The study provides a reproducible multi-metric framework for testing generative models of categorical geology. It quantifies how an LDM trained on TGRF data departs from the reference in both unconditional and conditional settings. The work highlights the role of training data complexity and conditioning pattern in model performance.

Findings

In the shoreface case the LDM reproduces mean trends and transiograms but narrows volume-fraction variance and slightly underrepresents rare sub-grid patterns. Conditional shoreface samples fail to honour one or two well cells in 6% of realisations. In the laterally heterogeneous case the LDM overestimates the middle facies in the Matérn experiment and misses at least one conditioning point in 85% of samples. Third-order cumulants reveal larger spread than

TGRF for exponential covariance but smaller for Matérn, indicating sensitivity to spatial smoothness.

Future work

Improving data preservation for scattered observations, extending the method to 3-D grids and larger images, and mitigating variance loss caused by latent compression are identified as next steps. The benchmark code and data sets enable direct comparison for future architectures.

5.4 Paper 4: Statistical Properties of Binary-Image Posterior Vision Transformer Samples

Authors: Oscar Ovanger, Jo Eidsvik, Ragnar Hauge, Jacob Skauvold
To be submitted

Motivation

Exact sampling for categorical Markov-random-field posteriors is computationally expensive. ViTs offer fast autoregressive generation and an explicit likelihood, but their statistical fidelity for conditional sampling is unknown. The paper tests how well a ViT can reproduce the posterior of a binary MRF conditioned on sparse observations.

Approach

A ViT with two encoder layers is trained by masked-token reconstruction on 5 462-token representations of 64×64 binary images drawn from an MRF. Conditioning pixels are handled by logit truncation; two patch-fill orders (Manhattan and inverse Manhattan) and two temperatures ($\tau = 1.0, 0.9$) are compared. For each variant the study generates one thousand conditional images and evaluates them against exact Variable Elimination Algorithm (VEA) samples using log-likelihoods, Jensen–Shannon divergence, PointSSIM, transiograms and third-order cumulants.

Contributions

The study introduces a ViT workflow for flexible conditional sampling with exact likelihood evaluation. It provides a multi-metric benchmark that pairs ViT outputs with ground-truth VEA realisations. The work diagnoses how sampling order and temperature affect marginal probabilities, correlations and higher-order structure.

Findings

ViT samples honour all conditioning pixels and capture large-scale patterns, yielding PointSSIM scores close to VEA. They are shifted toward lower MRF log-likelihoods and show a bias for the white class, traced to token-level probability miscalibration. Covariance analysis reveals over-extended correlations near observations and undersmoothing elsewhere; temperature sharpening alleviates one issue while worsening the other. Third-order cumulants are near zero, signalling loss of multi-pixel interactions present in the exact posterior.

Future work

Reducing token-level bias, exploring smaller tokens or hybrid convolutional neural network–ViT encoders, and extending the method to larger grids and multi-class facies are suggested next steps. A key open question is whether modified training objectives can balance conditioning accuracy with correct spatial correlations.

5.5 Paper 5: PointSSIM – A Low-Dimensional, Resolution-Invariant Image Metric

Authors: Oscar Ovanger, Ragnar Hauge, Jacob Skauvold, Michael J. Pyrcz, Jo Eidsvik
Posted on: arXiv (2025)[1]
Under review for: *IEEE Transactions on Image Processing*

Motivation

Pixel-wise metrics such as MSE and SSIM depend on identical image resolution and exact alignment, which limits their use for geological patterns that appear at many scales. The paper introduces PointSSIM, a metric designed to compare binary images of different sizes and orientations by focusing on structural rather than pixel correspondence.

Approach

Each image is converted to a marked point process. Anchor points are found as locally adaptive maxima of the minimal distance transform; their positions, radii and object labels form a compact representation. Four rotation- and scale-insensitive summary measures—anchor count, area coverage, anchor points per object and spatial variance irregularity—are computed, and a simple distance in this four-dimensional space defines the similarity score.

Contributions

The study defines a resolution-invariant structural metric for binary images grounded in mathematical morphology. It presents an efficient algorithm that reduces an $n_r n_c$-pixel grid to a handful of summary values, making large-scale comparisons fast. The work publishes code and five benchmark data sets covering geological and synthetic patterns.

Findings

PointSSIM separates image classes more clearly than MSE, SSIM and MS-SSIM and keeps within-class scores close to one. Tests on the same facies patterns rendered at 256^2, 512^2 and 1024^2 show minimal drift, confirming practical resolution invariance. The metric is most reliable when images contain enough anchor points; very large, homogeneous objects can reduce its sensitivity.

[1] `https://arxiv.org/abs/2501.01234`

Future work

Planned extensions include multi-facies and greyscale support, alternative point marks that capture curvature or texture, and the use of PointSSIM as a structural loss term when training generative models.

Conclusion and Discussion

This thesis has explored Bayesian conditioning for spatial categorical models through five papers spanning classical geostatistics and modern deep learning. Each contribution sheds light on different facets of this complex problem while revealing both progress made and fundamental limitations that remain.

Paper I highlights the importance of configuration ambiguity in well conditioning—a phenomenon where multiple geological interpretations honor the same well data. By explicitly modeling which observations belong to which geological bodies, the simulation-based approach exposes how such ambiguity can dominate posterior uncertainty in reservoir characterization.

Paper III systematically evaluates the LDM introduced in Paper II against its geostatistical training data. The results show that, although the model reproduces large-scale patterns and visual characteristics, it underestimates variability and remains sensitive to training-data composition. Architectural modifications tailored to geology (Paper II) markedly improve preservation of conditioning data, underscoring the value of domain-specific model design.

Paper IV and V emphasise that quantitative evaluation must complement visual inspection. PointSSIM introduces a resolution-invariant similarity metric, while the analysis of ViTs uncovers systematic biases in their generated patterns. These studies motivate a broader discussion of image comparison metrics for geological data—a theme that cuts across the entire thesis.

On the limits of image comparison metrics. Complex spatial structures cannot be condensed into a single scalar without losing information. Connectivity, morphology, and scale-dependent features matter differently across tasks such as flow simulation, well-placement optimisation, or facies mapping. Hence, no universal metric will suffice. Instead, effective assessment requires suites of complementary metrics that jointly span the structural attributes of interest: first-order proportions for basic consistency, variograms or connectivity functions for spatial correlation, higher-order statistics for morphology, and domain-specific measures linked to engineering objectives.

Most established metrics were designed for binary images and need careful adaptation for multi-facies or volumetric data. Computational burdens grow

rapidly in three dimensions, where feature extraction and comparison become far costlier. Future research should therefore pursue (i) metric collections that cover distinct aspects of structure, (ii) principled ways to select the most relevant subset for a given application, and (iii) efficient algorithms to compute these metrics on large 3D grids. Integrating such tools with generative models, as demonstrated throughout this thesis, is essential for rigorous validation of Bayesian conditioning workflows.

Trade-offs among modelling paradigms. Throughout the thesis we observe a consistent triad—theoretical guarantees, computational efficiency, and flexibility. MCMC methods deliver theoretical rigour and flexibility at high computational cost; explicit geostatistical priors offer clarity and efficiency but limited flexibility; neural networks provide flexibility and speed yet sacrifice strong guarantees. No approach simultaneously achieves all three, mandating application-driven method selection.

The contrast between implicit and explicit representations further clarifies these trade-offs. Classical models explicitly encode spatial structure via mathematical forms, whereas neural networks learn implicit representations from data. Our experiments show that when trained on samples from explicit priors, neural networks struggle to reproduce the full variability of their training sets, suggesting challenges in learning higher-order statistics.

Across all approaches, the core difficulty remains: incorporating sparse observations while preserving realistic spatial patterns. Whether manifesting as configuration uncertainty, data preservation errors, or distributional biases, the tension between honouring data and retaining prior statistics persists.

Practical implications. For practitioners, this work underscores the need for application-specific method selection and comprehensive validation. When rigorous uncertainty quantification is paramount, MCMC remains preferable despite its cost. Where flexibility or non-standard constraints dominate, neural methods hold promise, but the systematic biases revealed here demonstrate that visual realism alone is insufficient—metric-based evaluation is indispensable.

Future directions. Several grand challenges extend beyond this thesis. Realistic applications demand 3D models with millions of cells, straining both classical and neural approaches. Physical constraints should ideally regularise categorical models but are difficult to encode. Ensuring ensembles accurately reflect posterior uncertainty—not just plausibility—remains an open question.

Hybrid methods combining MCMC rigour with neural speed (e.g., neural proposal distributions) offer a promising path forward. Developing standardised benchmarks and evaluation protocols, including the multi-metric suites

proposed above, would enable fair comparison across methods. Finally, moving from purely statistical to causal models of geological processes may enhance both interpretability and fidelity.

In summary, the intersection of classical geostatistics and modern machine learning presents rich opportunities tempered by significant challenges. Progress will require integrating statistical fundamentals, domain expertise, and computational innovation. While this thesis clarifies some strengths and weaknesses of existing approaches, many questions remain—chief among them, how to better preserve statistical properties in implicit models and how to devise conditioning methods that honour both data and prior knowledge. Addressing these questions will demand continued cross-disciplinary collaboration to advance our capacity to model the subsurface with confidence.

Bibliography

Armstrong, M., Galli, A., Beucher, H., Loc'h, G., Renard, D., Doligez, B., Eschard, R., and Geffroy, F. (2011). Plurigaussian simulations in geosciences.

Armstrong, M. and Loc'h, G. L. (1994). Plurigaussian kriging and simulation. *Proceedings of Geostatistics Troia '92*.

Arpat, G. B. and Caers, J. (2007). Conditional simulation with patterns. *Mathematical Geology*, 39(2):177–203.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B*, 36(2):192–236.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Boisvert, J. B., Pyrcz, M. J., and Deutsch, C. V. (2010). Multiple point metrics to assess categorical variable models. *Natural Resources Research*, 19(3):165–175.

Borgomano, J., Lanteaume, C., Leonide, P., Fournier, F., Montaggioni, L. F., and Masse, J.-P. (2020). Quantitative carbonate sequence stratigraphy: Insights from stratigraphic forward models. *AAPG Bulletin*, 104(5):1115–1142.

Bridge, J. S. and Leeder, M. R. (1992). A simulation model of alluvial stratigraphy. *Sedimentology*, 39(5):851–885.

Brunet, D., Vrscay, E. R., and Wang, Z. (2012). On the mathematical properties of the structural similarity index. *IEEE Transactions on Image Processing*, 21(4):1488–1499.

Caers, J. and Zhang, T. (2004). Multiple–point geostatistics: A quantitative vehicle for integrating geologic analogs into multiple reservoir models. *AAPG Bulletin*, 88(6):975–992.

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.

Chan, S. and Elsheikh, A. H. (2017). Parametrization and generation of geological models with generative adversarial networks. *arXiv preprint arXiv:1708.01810*.

Deutsch, C. V. (1995). *Annealing Techniques Applied to Reservoir Modeling and the Integration of Geological and Engineering (well Test) Data*. UMI.

Deutsch, C. V. and Wang, L. (1996). Hierarchical object–based simulation of fluvial reservoirs. In *Proceedings of the Sixth International Geostatistics Congress*, pages 1–12.

Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020). An image is worth 16×16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Dupont, E., Zhang, T., Tilke, P., Liang, L., and Bailey, W. (2018). Generating realistic geology conditioned on physical measurements with generative adversarial networks. *arXiv preprint arXiv:1802.03065*.

Earl, D. J. and Deem, M. W. (2005). Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916.

Emery, X. (2004). Properties and limitations of sequential indicator simulation. *Stochastic Environmental Research and Risk Assessment*, 18(6):414–424.

Farmer, C. (1987). The generation of stochastic fields of reservoir parameters with specified geostatistical distributions.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):721–741.

Gloaguen, E. and Dimitrakopoulos, R. (2009). Two-dimensional conditional simulations based on the wavelet decomposition of training images. *Mathematical Geosciences*, 41(6):679–701.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*.

Griffiths, P. (2001). Forward stratigraphic modeling of deltaic deposits. *Sedimentology*, 48:203–222.

Guardiano, F. and Srivastava, R. M. (1993). Multivariate geostatistics: Beyond bivariate moments. pages 133–144.

Gustafson, G. and Caers, J. (1997). Plurigaussian simulation in petroleum reservoir modeling. *Petroleum Geoscience*, 3(1):23–32.

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive metropolis algorithm. *Bernoulli*, 7(2):223–242.

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.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109.

Hegstad, B. K., Omre, H., Tjelmeland, H., and Tyler, K. (1994). Stochastic simulation and conditioning by annealing in reservoir description. *Geostatistical Simulations (proceedings)*. Often cited as Hegstad et al. (1994).

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*.

Honarkhah, M. and Caers, J. (2010). Stochastic simulation of patterns using distance-based pattern modeling. *Mathematical Geosciences*, 42(5):487–517.

Howell, J. A., Skorstad, A., MacDonald, A., Fordham, A., Flint, S., Fjellvoll, B., and Manzocchi, T. (2008). Sedimentological parameterization of shallow-marine reservoirs. *Petroleum Geoscience*, 14(1):17–34.

Journel, A. G. (1983). Nonparametric estimation of spatial distributions. *Mathematical Geology*, 15(3):445–468.

Journel, A. G. (1998). Sequential indicator simulation with local probability. *Mathematical Geology*, 30(7):735–748.

Kindermann, R. and Snell, J. L. (1980). *Markov Random Fields and Their Applications*. American Mathematical Society.

Kingma, D. P. and Welling, M. (2014). Auto–encoding variational bayes. *International Conference on Learning Representations*.

Krige, D. G. (1951). A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6):119–139.

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.

Lee, D., Ovanger, O., Eidsvik, J., Aune, E., Skauvold, J., and Hauge, R. (2025). Latent diffusion model for conditional reservoir facies generation. *Computers Geosciences*, 194:105750.

Leuangthong, O., McLennan, J. A., and Deutsch, C. V. (2004). Minimum acceptance criteria for geostatistical realizations. *Natural Resources Research*, 13(3):131–141.

Lin, J. (1991). Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Loc'h, G. L. and Renard, D. (1992). Stochastic simulation of lithofacies: Implementation in saffron. In *Proceedings of the 5th International Geostatistics Congress*.

Mariethoz, G. and Caers, J. (2014). *Multiple-Point Geostatistics: Stochastic Modeling with Training Images*. John Wiley & Sons, Chichester.

Matheron, G. (1973). *The Theory of Regionalized Variables and its Applications*. École des Mines de Paris.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets.

Moran, P. A. P. (1950). Notes on continuous stochastic phenomena. *Biometrika*, 37(1–2):17–23.

Mosser, L., Dubrule, O., and Blunt, M. J. (2017). Reconstruction of three-dimensional porous media using generative adversarial neural networks. *Physical Review E*, 96(4):043309.

Mustapha, H. and Dimitrakopoulos, R. (2010). A new approach for geological pattern recognition using high-order spatial cumulants. *Computers Geosciences*, 36(3):313–334.

Pyrcz, M. J., Boisvert, J. B., and Deutsch, C. V. (2009). Alluvsim: A program for event-based stochastic modeling of fluvial depositional systems. *Computers & Geosciences*, 35(8):1671–1685.

Rasmussen, C. E. (2004). *Gaussian Processes in Machine Learning*, pages 63–71. Springer Berlin Heidelberg, Berlin, Heidelberg.

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

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High–resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695.

Sampat, M. P., Wang, Z., Gupta, S., Bovik, A. C., and Markey, M. K. (2009). Complex wavelet structural similarity: A new image similarity index. *IEEE Transactions on Image Processing*, 18(11):2385–2401.

Scheidt, C. and Caers, J. (2009). Representing spatial uncertainty using distances and kernels. *Mathematical Geosciences*, 41(4):397–419.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score–based generative modeling through stochastic differential equations. *International Conference on Learning Representations*.

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

Tahmasebi, P. (2018). Multiple point statistics: A review. In *Handbook of Mathematical Geosciences: Fifty Years of IAMG*, pages 577–602. Springer.

Tan, X., Tahmasebi, P., and Caers, J. (2014). Comparing training-image based algorithms using an analysis of distance. *Mathematical Geosciences*, 46(2):149–169.

Tjelmeland, H. and Besag, J. (1996). Markov random fields with higher-order interactions. *Scandinavian Journal of Statistics*, 23(4):465–492.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.

Wang, Z., Simoncelli, E. P., and Bovik, A. C. (2003). Multi-scale structural similarity for image quality assessment. In *Proceedings of the 37th IEEE Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1398–1402.

Yan, Z., Zhang, H., Picard, D., and Chang, Y. (2021). Transformer-based attention networks for continuous pixel-wise prediction. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16269–16279.

Part II

Scientific Papers

Addressing Configuration Uncertainty in Well Conditioning for a Rule-Based Model

Oscar Ovanger, Jo Eidsvik, Jacob Skauvold, Ragnar Hauge and Ingrid Aarnes

Mathematical Geosciences Volume 56, pages 1763–1788, (2024)

Mathematical Geosciences (2024) 56:1763–1788
https://doi.org/10.1007/s11004-024-10144-7

SPECIAL ISSUE

Addressing Configuration Uncertainty in Well Conditioning for a Rule-Based Model

Oscar Ovanger[1] · Jo Eidsvik[1] · Jacob Skauvold[2] · Ragnar Hauge[2] · Ingrid Aarnes[2]

Received: 14 February 2023 / Accepted: 13 April 2024 / Published online: 1 June 2024

Abstract
Rule-based reservoir models incorporate rules that mimic actual sediment deposition processes for accurate representation of geological patterns of sediment accumulation. Bayesian methods combine rule-based reservoir modelling and well data, with geometry and placement rules as part of the prior and well data accounted for by the likelihood. The focus here is on a shallow marine shoreface geometry of ordered sedimentary packages called bedsets. Shoreline advance and sediment build-up are described through progradation and aggradation parameters linked to individual bedset objects. Conditioning on data from non-vertical wells is studied. The emphasis is on the role of 'configurations'—the order and arrangement of bedsets as observed within well intersections in establishing the coupling between well observations and modelled objects. A conditioning algorithm is presented that explicitly integrates uncertainty about configurations for observed intersections between the well and the bedset surfaces. As data volumes increase and model complexity grows, the proposed conditioning method eventually becomes computationally infeasible. It has significant potential, however, to support the development of more complex models and conditioning methods by serving as a reference for consistency in conditioning.

Keywords Geomodelling · Object model · Well conditioning · Reservoir model · Configuration · Rule-based model

1 Introduction

In a sedimentary system, a geobody corresponds to a single depositional event. In a wave-dominated shallow marine environment, this is usually represented by a bedset. A probabilistic model for bedsets involves the shapes and dimensions of these geo-

Oscar Ovanger
oscar.ovanger@ntnu.no

[1] Norwegian University of Science and Technology, Trondheim, Norway

[2] Norwegian Computing Centre, Oslo, Norway

bodies through parameters with a specified probability distribution. The initial model is updated when incorporating well data, using Bayes' theorem.

Geobodies can span extensive areas, and a single geobody may appear in multiple well observations, especially for non-vertical wells. Therefore, to determine from well logs whether two observations come from the same geobody, one must consider various configurations, or ways in which bodies and observations could be linked. This is a hard problem in geomodelling (see Fig. 1).

Deutsch and Wang (1996) argue that this problem is so difficult that approaches based on rejection sampling are needed to get it right. However, Hauge et al. (2007) show that a carefully crafted Markov chain Monte Carlo algorithm can provide samples that are a good approximation to the true posterior, including for properties of objects that intersect one or more wells.

To test the quality of the sampling, Hauge et al. (2007) use the principle of double expectation. The key insight is that if realisations are generated from a prior model and synthetic data are generated from the realisations, before finally a conditional realisation is generated given these data, then any statistical property should follow the same distribution whether it is computed from the prior samples or from the conditional samples. A model that satisfies this criterion can be used to sample out configuration probabilities in a manner comparable to the well correlation analysis of Bertoncello et al. (2013) or Wingate et al. (2016). Wang et al. (2018) suggest another optimisation-based approach for conditioning in object-based models.

This article focuses on directly computing the probability of a given configuration of geobodies in well observations. Although an exact answer cannot be computed, the proposed approximate method performs well in a double expectation test. As is shown in Sect. 4, conditioning becomes relatively easy once a distribution over configurations is in place. Rather than using a complex conditioning scheme to sample out configuration probabilities, this approach builds on complex computation of configuration probabilities to create a direct conditioning scheme.

Geobody modelling approaches can be divided into two main classes: conventional object models and rule-based models. Object models were introduced by Bridge and Leeder (1979), and have typically been used to model fluvial systems (Viseur et al. 1998; Holden et al. 1998; Keogh et al. 2007). Troncoso et al. (2022) study a sequential object placing model with a particle filter solution for conditioning this kind of model. In these object models, each geobody is modelled as more or less independent of the others.

In rule-based models (Cojan et al. 2005; Pyrcz et al. 2015; Parquer et al. 2017; Rongier et al. 2017), the geobodies respond to each other, typically through stacking rules. These models are generally sequential, mimicking the depositional process by adding geobodies in geological order.

Object-based models for facies in a stratigraphical stacking of beds have also been considered. For example, Manzocchi and Walsh (2023) study analytic expressions for proportions and amalgamation ratios of foreground and background facies. Studies of sequence simulation and conditioning include the work of Allard et al. (2021) who use Markov chain Monte Carlo to simulate latent Gaussian models of sequences conditional on thicknesses in vertical well logs, and the surface-based geological models of Titus et al. (2021), who train neural networks on geological model inputs

and facies types along wells. Taking an approach based on the ensemble Kalman filter, Skauvold and Eidsvik (2018) updated geological process model realisations given well log information in a sequential, bottom-up fashion.

Most work on well conditioning of geobody models, and in particular work that considers the allocation of geobodies conditional on wells, has been done on object models (Deutsch and Wang 1996; Seifert and Jensen 2000; Hauge et al. 2007, 2017). Although well conditioning of rule-based models has received some attention (Bertoncello et al. 2013; Wingate et al. 2016; Jo and Pyrcz 2020), no definitive conditioning methodology has yet emerged. The model considered here is a simple rule-based bedset model. It will be demonstrated that configuration probabilities for this model can be estimated, and that these resulting probabilities can be used to generate samples from a distribution that closely approximates the true posterior distribution.

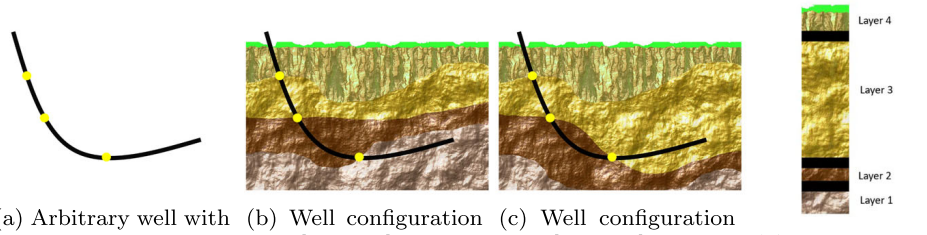
Section 2 describes the main components of the bedset model and completes the well configuration concept with more detail. Section 3 describes the statistical models for bedsets and for well data. In Sect. 4, the conditioning problem is made precise. Then an algorithm to estimate configuration probabilities is presented, and a conditioning approach based on that algorithm is laid out. Section 5 details a simulation study that uses the algorithm. Section 6 concludes the article by summarising findings and pointing out possible directions for further research.

2 Problem Description

Bedsets are three-dimensional objects bounded above and below by surfaces. The top of one bedset is the base of the next, and keeping track of one surface per bedset is sufficient for a complete representation. In this work, all illustrations and simulation experiments will be in two dimensions for simplicity. Considering the process of stacking m bedset objects, initial bathymetry is defined by $z_0(x)$, for lateral coordinate x in a domain $\mathcal{D} \subset \mathbb{R}^1$. Bedset boundaries $z_1, ..., z_m$ are placed above each other sequentially, assuming no erosion. The geometry of the bedset system is mainly controlled by how far the system builds out (progradation) and up (aggradation) with the addition of each bedset. The pair of progradation and aggradation parameters of the ith bedset is denoted by $\boldsymbol{\theta}_i$. At a larger scale, a set of stacked bedsets bounded by marine flooding surfaces is a parasequence (Catuneanu et al. 2009; Boggs 2014).

Data from a well, denoted by $\boldsymbol{d}$, are collected at specific coordinates. A running assumption here is that these data capture the transitions between different bedsets, but the bedset indices of these transitions are not revealed and hence the bedset order of the well is not known. If one assumes that bedset age increases monotonically with depth everywhere, then a vertical well uniquely constrains the order of intersected bedsets. However, if the well is not vertical, then the order is ambiguous even with this assumption. The order is still the sequence of indices referring to the layers (bedsets) visited along the well trajectory (in the 'down' direction), but this order is not necessarily monotonic. This sequence of indices is here called a configuration and denoted by $\boldsymbol{c}$.

To clarify the concept of a well configuration in this context, consider a parabola-shaped well (Fig. 1a) going through stratigraphic sequences. From the well data, three



(a) Arbitrary well with 3 intersections (b) Well configuration $c_1 = [4, 3, 2, 1]$ (c) Well configuration $c_2 = [4, 3, 2, 3]$ (d) Bedset labels.

Fig. 1 Example of two different well configurations c

Table 1 Main variables and their description

Variable	Description
z_i	Elevation to top surface of ith bedset
$z = (z_1, \ldots, z_m)$	Elevation to all bedset top surfaces
$\boldsymbol{\theta}_i$	Progradation and aggradation of ith bedset
$\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m)$	Progradation and aggradation of all bedsets
$\boldsymbol{d}$	Well data of locations along well path and bedset intersections
$\boldsymbol{c}$	Configurations of bedsets along well path

flooding surfaces or abrupt changes in the subsurface are identified. Given a specific object-model, this indicates two transitions from one object to another. Notably, this well information does not reveal the number of distinct objects visited or their sequence. However, one can compile a list of objects that may have been visited by the well. A configuration is one such list of object transitions. Figure 1 shows two distinct scenarios. In one, the parabola-shaped well has configuration $\boldsymbol{c}_1 = [4, 3, 2, 1]$ (Fig. 1b), while the configuration is $\boldsymbol{c}_2 = [4, 3, 2, 3]$ in Fig. 1c. In the former scenario, the well intersects with three objects, whereas it only intersects with two in the latter. Often, in real-world situations, the flow between objects is limited, while the flow within an object is efficient, given that the permeability within the object is high. Under such conditions, the well in the first scenario would interact with more objects, potentially optimising flows. Essentially, the well data alone do not reveal the configuration, yet predictions of surface depths and flow behaviour may depend sensitively on it. Therefore, acknowledging the existence of multiple potential configurations and exploring methods to model them can be crucial.

The notation is summarised in Table 1.

The statistical model for the bedset geometry is explained in Sect. 3.1, while the configurations and well data models are described in Sect. 3.2. In Figs. 3, 9, 10 and 11 depth is used as the axis-label to represent the distance from the top surface of the most elevated bedset to the bottom of the bathymetry. The depth-axis is normalised to the range of 0 to 100.

3 Statistical Model for Bedsets, Well Data, and Well Configurations

3.1 A Model of Bedsets

The focus of this paper is on a rule-based model (Pyrcz and Deutsch 2014; Jo and Pyrcz 2020) for a wave-dominated shoreface environment (see e.g. Eide et al. 2014, 2015). In such systems, sediments originally supplied by rivers are reworked by waves and distributed alongshore. The constant reworking gives rise to an upward-coarsening grain size profile that follows wave-energy. On top is a sandy shoreface with excellent reservoir properties. Below the fair-weather wavebase there is an offshore transition zone which is an interbedding of mud and sand. Below the storm wavebase there is only background sedimentation of offshore mud.

New bedsets are believed to form when gradual deposition is disturbed, such as by an abrupt small-scale rise in relative sea level or a river avulsion. The boundary surfaces that separate older bedsets from younger ones are the result of such sedimentary supply interruptions. Graham et al. (2015a, b) show that low-permeability layers between bedsets can significantly affect fluid flow in a reservoir, and that adequate representation of this heterogeneity can be a key requirement for accurate flow simulations.

Here, stacking behaviour within a single parasequence is considered. This system follows well-understood rules, conveniently represented by a model that stacks bodies outwards and upwards, where bedset objects represent deposition events. Here, the modelling assumption is that bedsets can be recognised in the field and that they have a meaningful modelling scale. Another assumption is that the bedset boundaries follow the same mean shape. Figure 3 illustrates multiple stacked bedsets building up a parasequence. In addition to the bedset boundaries of main interest, the within-bedset facies transitions from sand ($\nu = 1$) to shale ($\nu = 2$) are also illustrated in this display.

The geometry of new bedsets are mainly driven by the basal profile and the cumulative aggradation α_i and progradation π_i parameters, $i = 1, \ldots, m$, which control vertical and horizontal components of the shoreline position. Assuming no erosion, these parameters are assumed to be positive. The entire parameter vector is denoted by $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m)$, where

$$\boldsymbol{\theta}_i = \begin{cases} (\log \pi_i, \log \alpha_i), & i = 1 \\ (\log(\pi_i - \pi_{i-1}), \log(\alpha_i - \alpha_{i-1})), & i \geq 2. \end{cases} \tag{1}$$

The incremental progradation and aggradation between two subsequent bedsets are considered to be random variables with joint probability density function (PDF) $p(\boldsymbol{\theta}_i)$, $i = 1, \ldots, m$. Parameters for different layers are assumed to be independent, so that $p(\boldsymbol{\theta}) = \prod_{i=1}^{m} p(\boldsymbol{\theta}_i)$. To allow negative correlation between aggradation and progradation, a common bivariate Gaussian model is used for each $\boldsymbol{\theta}_i$

$$\boldsymbol{\theta}_i = \begin{bmatrix} \log(\pi_i - \pi_{i-1}) \\ \log(\alpha_i - \alpha_{i-1}) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_\pi \\ \mu_\alpha \end{bmatrix}, \begin{bmatrix} \sigma_\pi & \rho_{\pi_\alpha} \\ \rho_{\alpha_\pi} & \sigma_\alpha \end{bmatrix} \right). \tag{2}$$

A Markov assumption is applied for the bedset boundaries model

$$p(z|\boldsymbol{\theta}) = \prod_{i=1}^{m} p(z_i|z_{i-1}, \boldsymbol{\theta}_i). \tag{3}$$

For the initial bathymetry, $z_0(x) = \exp\left(-\frac{x-\pi_0}{\xi}\right)$, where ξ controls the slope of the bathymetry and is set to 100 here. Using a sequential structure, and given α_i and π_i, the PDF of $\boldsymbol{z}_i$ is represented by a Gaussian random field (GRF). The mean $\boldsymbol{\mu}_i$ is a function of $\boldsymbol{\theta}_i$

$$\boldsymbol{\mu}_i = \mathbf{1}\alpha_i + \exp\left(-\frac{\mathbf{x} - \mathbf{1}\pi_i}{\xi}\right) \tag{4}$$

where $\mathbf{1}$ is a vector of ones acting on the aggradation α_i and the progradation π_i of bedset i. The residual covariance matrix for each bedset boundary is denoted by $\boldsymbol{\Sigma}$. It is the same for each bedset. With the Markovian structure, the conditional distributions are

$$[z_i|z_{i-1}, \boldsymbol{\theta}_i] \sim N(\boldsymbol{\mu}_i + \boldsymbol{T}\boldsymbol{\Sigma}^{-1}(z_{i-1} - \boldsymbol{\mu}_{i-1}), \boldsymbol{\Sigma} - \boldsymbol{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{T}). \tag{5}$$

The inter-bedset covariance matrix $\boldsymbol{T}$ in (5) is calculated using an autocorrelation parameter, $|\rho| < 1$, that measures how closely neighbouring bedset surfaces covary. This parameter is then multiplied by the residual covariance matrix $\boldsymbol{\Sigma}$.

Overall, the joint PDF $p(z_1, ..., z_m|\boldsymbol{\theta})$ is defined via a product sequence of PDFs with one component per bedset. The marginal PDF of these bedset geometries are obtained by integrating out the parameters

$$p(z) = \int p(z|\boldsymbol{\theta})p(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}. \tag{6}$$

Algorithm 1 lists the steps involved in producing a Monte Carlo (MC) sample from this prior distribution. Previous research on structural models of horizons employing Gaussian random fields (GRFs) was conducted by Abrahamsen (1993) and Goff (2000), in which horizons are sequentially sampled. Nonetheless, in these studies, the configurations of observations for conditional simulations are predetermined.

Algorithm 1 Producing a prior sample of $\boldsymbol{\theta}, \boldsymbol{z}$

procedure DRAWPRIORSAMPLE($\boldsymbol{\mu}_{\boldsymbol{\theta}}$, $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$, $\boldsymbol{\Sigma}$, $\boldsymbol{T}$, m)
  for $i = 1 : m$ **do**
    $\boldsymbol{\theta}_i \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}})$
    $z_i \sim \mathcal{N}(\boldsymbol{\mu}_i + \boldsymbol{T}\boldsymbol{\Sigma}^{-1}(z_{i-1} - \boldsymbol{\mu}_{i-1}), \boldsymbol{\Sigma} - \boldsymbol{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{T})$
  end for
  return $\{\boldsymbol{\theta}_i, z_i : i = 1, \ldots, m\}$
end procedure

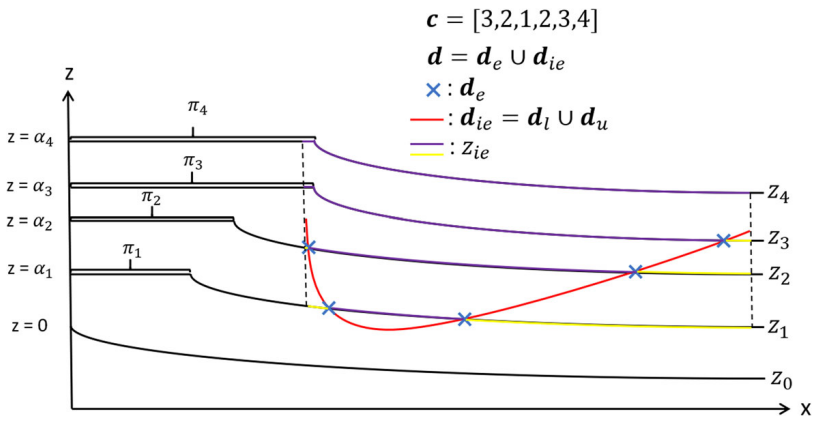


Fig. 2 A simplified figure of the set-up with the data nomenclature used throughout this work. z_{ie} is the part of the bedset boundaries subject to inequality constraints. The purple segment is strictly above the well, and the yellow part is strictly below the well. The blue crosses are where the well and the bedset boundaries intersect and thus $z_{\text{e}} = \boldsymbol{d}_{\text{e}}$

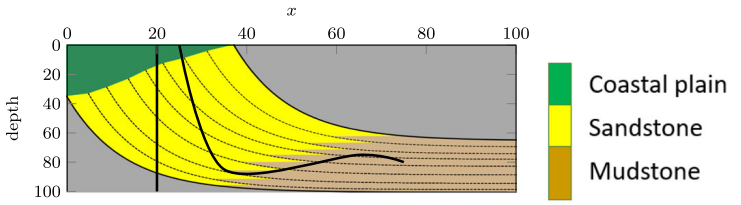


Fig. 3 Vertical versus non-vertical well data. A well configuration is defined by the indices of bedset transitions. This is straightforward for vertical wells. It becomes more involved for non-vertical wells

Bedsets can be subdivided into different architectural elements. As shown in Fig. 3, each bedset can be split into an all-sand shoreface part and an offshore transition zone part of interbedded sandstone and shale (Eide et al. 2014). Next, porosity, permeability, and other properties can be filled in with the desired level of heterogeneity. These downstream modelling steps are important and relevant, but the scope of this article is limited to geometric modelling of bedset boundaries.

3.2 Well Data, Configurations, and Their Significance

Well configurations are defined by transitions between bedsets. In Fig. 3, a sequence of bedsets is intersected by two vertical wells (left) and a non-vertical well (right).

The configuration of well data is $\boldsymbol{c} = [5, 4, 3, 2, 1]$ for the vertical case. The configuration of well data in the non-vertical well is $\boldsymbol{c} = [6, 5, 4, 3, 2, 3, 4, 5, 6, 5]$. Note that there are eight bedsets in the figure. The coastal plain is not a bedset, thus it is not a part of the configurations.

To simplify the presentation, this study focuses only on a single well. The well is assumed to have known coordinates. Hence, the well configuration $\boldsymbol{c}$ is determined by the bedset boundaries $\boldsymbol{z}$. The probability density $p(\boldsymbol{c}|\boldsymbol{z})$ is then a Dirac δ-function. The marginal probabilities $p(\boldsymbol{c})$ can be approximated by sampling multiple times

from $p(\boldsymbol{z})$ and then counting the configurations. The MC sampling procedure for well configuration distribution involves the steps listed in Algorithm 2. Here, n_r denotes the number of MC samples. The probability of a particular configuration $\boldsymbol{c}$ is approximated by the number of MC samples having that configuration, divided by n_r. Parameters σ_z^2, η represent variance and smoothness of the GRF.

Algorithm 2 Monte Carlo approximation of the well configuration distribution.

procedure ApproximateConfigurationDistributionByMonteCarloSampling(n_r)
  initialise $i \leftarrow 1, \mathcal{C} \leftarrow \emptyset$
  while $i < n_r$ **do**
    Sample $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_m \sim N(\boldsymbol{\mu}_\theta, \Sigma_\theta)$
    $\boldsymbol{z}_1, ..., \boldsymbol{z}_m | \boldsymbol{\theta} \sim N(\boldsymbol{\mu}_{\boldsymbol{z}}(\boldsymbol{\theta}), \Sigma_{\boldsymbol{z}}(\sigma_z^2, \eta, \rho))$
    Determine $\boldsymbol{c}$ based on how the bedsets intersect the well.
    $\mathcal{C} \leftarrow \mathcal{C} \cup \boldsymbol{c}$
    $i \leftarrow i + 1$
  end while
end procedure

The set of possible well configurations depends on the number of bedsets in the model and the number of observed bedset boundaries. As the well goes from one bedset to another, the new bedset must be one of the neighbours of the previous one. If the previous bedset happens to be the top or bottom bedset, then there is only one bedset that can be entered. Possible well configurations for a given number of observed bedset boundaries are illustrated in Fig. 4, where m represents the mth bedset according to the model, counting from the bottom up. Every left-to-right path is a possible configuration. For example, $\boldsymbol{c} = [m, m-1, m-2, m-1, m, m-1]$ is one of the 10 possible configurations with six measurements.

The data $\boldsymbol{d}$ include well picks of bedset boundary surfaces, when the model is known. Such well transition data could, for instance, be extracted from large changes in the gamma-ray log or a related log of shale/sand proportions. Bubnova et al. (2020) used clustering techniques of sand proportions data to detect sedimentary transitions in log data. The bedset picks are here assumed to be measurements of the z-coordinate of the bedset boundary at the (x, z)-locations where they are made. For vertical wells, these observed intersection points are the only data constraints. A non-vertical well imposes mixed constraints on the bedset boundary surfaces:

- Equality constraints: Observed intersections in a bedset layer i put equality constraints on the surface $\boldsymbol{z}_i$ which must have specific values at these well locations, denoted $\boldsymbol{d}_{\text{e}}$.
- Inequality constraints: Well intervals that come before, between or after intersection point observations impose inequality constraints on the two adjacent bedset surfaces. A well interval lying inside bedset i would act as an upper bound for $\boldsymbol{z}_{i-1}$ and as a lower bound for $\boldsymbol{z}_i$. The upper and lower bounds are denoted $\boldsymbol{d}_u$ and $\boldsymbol{d}_l$, respectively, and the notation $\boldsymbol{d}_{\text{ie}}$ is used to represent the union of the upper and lower bounds.

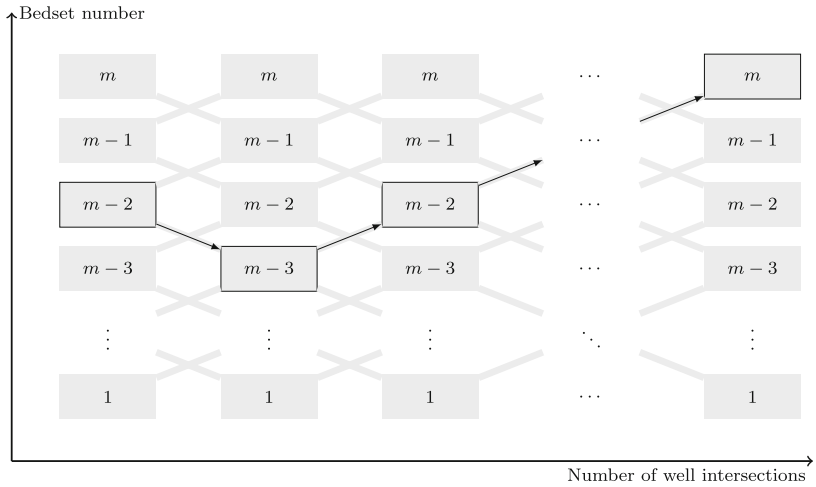


Fig. 4 Possible configurations of layers observed in a well. Nodes are labelled with layer indices. Each rightward transition from one column to the next represents an observed surface intersection of the well. When passing an intersection point a well cannot stay in the same layer, but must move to a neighbouring bedset. The indicated path through the lattice corresponds to the configuration $c = [m-2, m-3, m-2, \ldots, m]$. Configurations can have any number of picks, including zero. In that case, it is identified by a single index. The configuration still specifies which layer the well is in

Assuming n_{d_i} observations $\boldsymbol{d}_i$ of bedset boundary i, let $\boldsymbol{H}_i \in \mathbb{R}^{n_{d_i} \times n_x}$, where n_x is the horizon grid size, be the design matrix that maps points from the geometric model to the data space, that is, $\boldsymbol{d}_i = \boldsymbol{H}_i \boldsymbol{z}_i$. This gives the following distribution of the equality data

$$\boldsymbol{d}_i | \boldsymbol{c}, \boldsymbol{\theta}_i \sim \mathcal{N}\left(\boldsymbol{H}_i \boldsymbol{\mu}_{z_i|z_{i-1}}, \boldsymbol{H}_i \boldsymbol{\Sigma}_{z_i|z_{i-1}} \boldsymbol{H}_i^T\right), \tag{7}$$

where $\boldsymbol{\mu}_{z_i|z_{i-1}}$ is the conditional mean given the previous surface in Eq. (5). Under mixed-constraints, the likelihood function becomes

$$p(\boldsymbol{d}|\boldsymbol{c}, \boldsymbol{\theta}) = p(\boldsymbol{z}_\mathrm{e} = \boldsymbol{d}_\mathrm{e}(\boldsymbol{c}), \boldsymbol{d}_\ell(\boldsymbol{c}) \leq \boldsymbol{z}_\mathrm{ie} \leq \boldsymbol{d}_\mathrm{u}(\boldsymbol{c}) | \boldsymbol{c}, \boldsymbol{\theta}), \tag{8}$$

where $\boldsymbol{z}_\mathrm{e}$ and $\boldsymbol{z}_\mathrm{ie}$ contain components of $\boldsymbol{z}$ that are subject to equality and inequality constraints, respectively. The influence of the configuration enters through the vectors $\boldsymbol{d}_\ell$ and $\boldsymbol{d}_\mathrm{u}$ of lower and upper bounds. Assimilating the mixed constraints sequentially, starting with the equality constraints, Eq. (8) can be rewritten as

$$\begin{aligned} p(\boldsymbol{d}|\boldsymbol{c}, \boldsymbol{\theta}) &= P(\boldsymbol{d}_\ell(\boldsymbol{c}) \leq \boldsymbol{z}_\mathrm{ie} \leq \boldsymbol{d}_\mathrm{u}(\boldsymbol{c}) | \boldsymbol{z}_\mathrm{e} = \boldsymbol{d}_\mathrm{e}(\boldsymbol{c}), \boldsymbol{c}, \boldsymbol{\theta}) p(\boldsymbol{d}_\mathrm{e}(\boldsymbol{c}) | \boldsymbol{c}, \boldsymbol{\theta}) \\ &= \left[F_{\boldsymbol{z}_\mathrm{ie}|\boldsymbol{z}_\mathrm{e}=\boldsymbol{d}_\mathrm{e}(\boldsymbol{c}),\boldsymbol{\theta}}(\boldsymbol{d}_\mathrm{u}(\boldsymbol{c})) - F_{\boldsymbol{z}_\mathrm{ie}|\boldsymbol{z}_\mathrm{e}=\boldsymbol{d}_\mathrm{e}(\boldsymbol{c}),\boldsymbol{\theta}}(\boldsymbol{d}_\ell(\boldsymbol{c}))\right] f(\boldsymbol{d}_\mathrm{e}(\boldsymbol{c}) | \boldsymbol{\theta}, \boldsymbol{c}), \end{aligned} \tag{9}$$

where the latter is the PDF of the equality constraints while $F_{\boldsymbol{z}_\mathrm{ie}|\boldsymbol{z}_\mathrm{e}=\boldsymbol{d}_\mathrm{e}(\boldsymbol{c}),\boldsymbol{\theta}}(\boldsymbol{z}_\mathrm{ie})$ is the multivariate cumulative distribution function (CDF) of $\boldsymbol{z}_\mathrm{ie}$ after conditioning the GRF on the equality constraints. If the elements of $\boldsymbol{z}_\mathrm{ie}$ given $\boldsymbol{z}_\mathrm{e} = \boldsymbol{d}_\mathrm{e}(\boldsymbol{c})$ and $\boldsymbol{\theta}$ are weakly

correlated, they can be treated as independent, giving the element-wise factorisation

$$p(\boldsymbol{d}|\boldsymbol{c},\boldsymbol{\theta}) = f(\boldsymbol{d}_{\mathrm{e}}(\boldsymbol{c})|\boldsymbol{\theta},\boldsymbol{c}) \prod_{z\in\boldsymbol{z}_{\mathrm{ie}}} \left[F_{z|\boldsymbol{z}_{\mathrm{e}}=\boldsymbol{d}_{\mathrm{e}}(\boldsymbol{c}),\boldsymbol{\theta}}(d_{\mathrm{u}}(\boldsymbol{c})) - F_{z|\boldsymbol{z}_{\mathrm{e}}=\boldsymbol{d}_{\mathrm{e}}(\boldsymbol{c}),\boldsymbol{\theta}}(d_{\ell}(\boldsymbol{c}))\right], \quad (10)$$

where d_{ℓ} and d_{u} are the elements of $\boldsymbol{d}_{\ell}$ and $\boldsymbol{d}_{\mathrm{u}}$ that go along with each $z \in \boldsymbol{z}_{\mathrm{ie}}$.

When evaluating Eq. (10), it is further useful to consider one surface at a time, and one inequality observation at a time. For a given surface, each element of $\boldsymbol{z}_{\mathrm{ie}}$ will be bounded above or below, but never both. When there is only a lower bound, let $d_{\mathrm{u}} \to \infty$. The first CDF term then becomes equal to 1. Similarly, when there is only an upper bound, let $d_{\ell} \to -\infty$. The second CDF term is then zero. It is also convenient to let $\mathcal{L}$ and $\mathcal{U}$ stand for the two disjoint lower-bounded and upper-bounded subsets of $\boldsymbol{z}_{\mathrm{ie}}$. When all this is put together, Eq. (10) becomes

$$p(\boldsymbol{d}|\boldsymbol{c},\boldsymbol{\theta}) = f(\boldsymbol{d}_{\mathrm{e}}(\boldsymbol{c})|\boldsymbol{\theta},\boldsymbol{c}) \prod_{z\in\mathcal{L}} \left[1 - F_{z|\boldsymbol{z}_{\mathrm{e}}=\boldsymbol{d}_{\mathrm{e}}(\boldsymbol{c}),\boldsymbol{\theta}}(d_{\ell}(\boldsymbol{c}))\right] \prod_{z\in\mathcal{U}} F_{z|\boldsymbol{z}_{\mathrm{e}}=\boldsymbol{d}_{\mathrm{e}}(\boldsymbol{c}),\boldsymbol{\theta}}(d_{\mathrm{u}}(\boldsymbol{c})). \quad (11)$$

Algorithm 3 lists the steps in the procedure for evaluating the likelihood $p(\boldsymbol{d}|\boldsymbol{\theta},\boldsymbol{c})$. First the log-likelihood variable is initialised with a value of zero. Then, in lines 2, 3, and 4, relevant indices are identified. It is only necessary to consider surfaces that are adjacent to at least one point on the well trajectory. In line 5, it loops over all layers that could potentially be updated. For each surface, a decision is made concerning the kind of constraint to apply. There are always inequality constraints, and if the current layer is intersected by the well, then there are also equality constraints. In the case of pure inequality constraints (lines 8 and 9), the cumulative log-likelihood variable is updated by evaluating the appropriate CDF (see Eq. 11). In the mixed constraints case (lines 12–15), the equality constraints are considered first, as the surface is conditioned to them. In this case, both the PDF and CDF of the surface distribution need to be evaluated to update the log-likelihood with the information from all constraints.

Algorithm 3 Evaluating $p(\boldsymbol{d}|\boldsymbol{c},\boldsymbol{\theta})$

1: **procedure** EVALUATELIKELIHOOD($\boldsymbol{d}$, $\boldsymbol{c}$, $\boldsymbol{\theta}$)
2: Initialise $\log p = 0$
3: $k_{\mathrm{start}} \leftarrow \min(\boldsymbol{c}) - 1$: Index of last surface below well.
4: $k_{\mathrm{end}} \leftarrow \max(\boldsymbol{c}) + 1$: Index of first surface above well.
5: **for** $k = k_{\mathrm{start}} : k_{\mathrm{end}}$ **do**
6: $\log p \leftarrow \log p + \log p(\boldsymbol{d}_k|\boldsymbol{c},\boldsymbol{\theta})$
7: **end for**
8: **return** $\log p$
9: **end procedure**

4 Conditioning Problem

The goal is to condition (i) the placement of each bedset boundary $\boldsymbol{z}$, (ii) the model parameters $\boldsymbol{\theta}$, and (iii) the configuration $\boldsymbol{c}$, given the observations $\boldsymbol{d}$ made in the wells. By using different variants of Bayes' rule, components of the full posterior models are used to form an approximate solution. First, the full posterior PDF of the geometry $\boldsymbol{z}$ is studied, next the PDF $p(\boldsymbol{\theta}|\boldsymbol{c}, \boldsymbol{d})$ of the model parameters is approximated, and then the posterior distribution for the well configuration $\boldsymbol{c}$ is assessed. Finally, the pieces are combined in a compact algorithm.

4.1 Conditioning of Geometries

The density $p(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{c}, \boldsymbol{d})$ is multivariate Gaussian, and by standard expressions for the conditional multivariate Gaussian distribution of equation (5), the mean and covariance of bedset boundaries can be obtained (see e.g. Chapter 4 in Cressie and Wikle (2015)).

The idea is to let the data constrain both $\boldsymbol{c}$ and $\boldsymbol{\theta}$, and write the posterior distribution of $\boldsymbol{z}$ as

$$p(\boldsymbol{z}|\boldsymbol{d}) = \sum_{\boldsymbol{c}} \int p(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{c}, \boldsymbol{d}) p(\boldsymbol{\theta}|\boldsymbol{c}, \boldsymbol{d}) p(\boldsymbol{c}|\boldsymbol{d}) \, \mathrm{d}\boldsymbol{\theta}. \tag{12}$$

4.2 Parameter Estimation

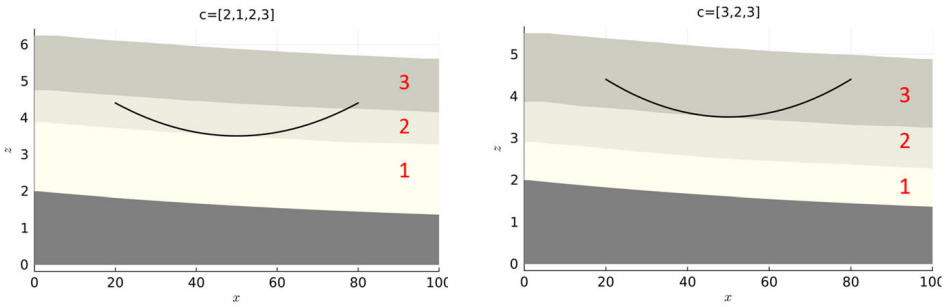
The posterior distribution of model parameters $\boldsymbol{\theta}$ can be assessed from well observations of bedsets. This involves finding the following PDF

$$p(\boldsymbol{\theta}|\boldsymbol{c}, \boldsymbol{d}) = \frac{p(\boldsymbol{d}|\boldsymbol{c}, \boldsymbol{\theta}) p(\boldsymbol{c}, \boldsymbol{\theta})}{p(\boldsymbol{c}, \boldsymbol{d})}. \tag{13}$$

This posterior has no closed form expression, but a reasonable substitute is a local Gaussian approximation (GA) such that $\boldsymbol{\theta}|\boldsymbol{c}, \boldsymbol{d} \approx \mathcal{N}(\boldsymbol{\theta}_{\mathrm{MAP}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}_{\mathrm{MAP}}})$, where

$$\begin{gathered} \hat{\boldsymbol{\theta}}_{\mathrm{MAP}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left[-\log p(\boldsymbol{\theta}|\boldsymbol{c}, \boldsymbol{d}) \right], \\ \text{and} \\ \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}_{\mathrm{MAP}}} = \left[-\nabla^2 \log p(\hat{\boldsymbol{\theta}}_{\mathrm{MAP}}|\boldsymbol{c}, \boldsymbol{d}) \right]^{-1}. \end{gathered} \tag{14}$$

This is a normal distribution fitted to the mode of the log-posterior, and using the curvature at this mode (negative Hessian of the log-posterior) as precision matrix (inverse covariance matrix). This PDF is denoted by $\hat{p}_{\mathrm{GA}}(\boldsymbol{\theta}|\boldsymbol{c}, \boldsymbol{d})$. In this approximation, an assumption is that $\boldsymbol{c}$ and $\boldsymbol{\theta}$ are independent a priori, so that $p(\boldsymbol{c}, \boldsymbol{\theta}) = p(\boldsymbol{c}) p(\boldsymbol{\theta})$. The mode $\hat{\boldsymbol{\theta}}_{\mathrm{MAP}}$ in Eq. (14) is hence found by minimising the sum of the negative log-prior for $\boldsymbol{\theta}$ and the negative log-likelihood $p(\boldsymbol{d}|\boldsymbol{c}, \boldsymbol{\theta})$. The Nelder–Mead algorithm (see e.g.



(a) Realization with well configuration $\boldsymbol{c} = [2, 1, 2, 3]$.

(b) Realization with well configuration $\boldsymbol{c} = [3, 2, 3]$.

Fig. 5 Two realisations following parameters in Table 2 and simulation in Algorithm 1. Here, $\boldsymbol{c} = [3, 2, 3]$ is the most likely configuration given the parameters, and the realisation in Fig. 5a is the one that the posterior parameters in Fig. 6 are conditioned on

Mogensen and Riseth 2018) is used to locate the mode. The Hessian is evaluated by second-order central difference (Abramowitz 1972).

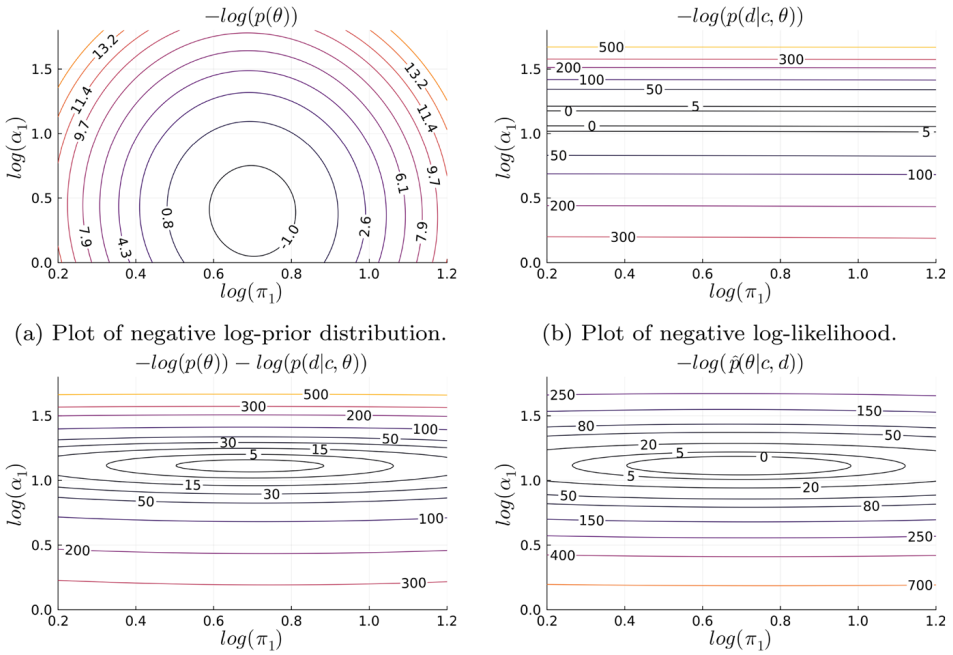
As an example of a typical approximation situation, conditioning results from one synthetic well observation are presented next. The well observation is simulated by creating a realisation following Algorithm 1 and extracting the well data. The realisation is illustrated in Fig. 5a. Figure 6 shows the negative log-prior, log-likelihood, sum of log-prior and log-likelihood, and approximate log-posterior for $\log(\pi_1)$ and $\log(\alpha_1)$, the log-progradation and log-aggradation of the bottommost bedset. In each case, the negative logarithm of the PDF or likelihood is shown, so that lower values correspond to more likely outcomes. It is evident from the contour shapes of the different functions, in particular the display in (c) versus the prior display in (a), that the data provide more information about $\log(\alpha_1)$ than $\log(\pi_1)$. The approximate log-posterior presented in display (d) resembles the evaluation conducted in the sum of log-prior and log-likelihood as shown in display (c). Given that the sum of the log-prior and log-likelihood is not normalised, an expected outcome would be a separation by a constant. This expectation aligns with observations indicating that the approximate log-posterior is approximately half of the sum. Hence, the Gaussian assumption at the mode fits rather well here.

4.3 Conditional Well Configuration Distribution

The conditional distribution $p(\boldsymbol{c}|\boldsymbol{d})$ over possible well configurations can be represented via known or approximated conditional distributions. First, the PDF $p(\boldsymbol{\theta}, \boldsymbol{c}, \boldsymbol{d})$ can be factorised in two different ways, namely

$$p(\boldsymbol{\theta}, \boldsymbol{c}, \boldsymbol{d}) = p(\boldsymbol{d}|\boldsymbol{c}, \boldsymbol{\theta})p(\boldsymbol{c}, \boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{d}, \boldsymbol{c})p(\boldsymbol{c}|\boldsymbol{d})p(\boldsymbol{d}). \tag{15}$$

Solving for $p(\boldsymbol{c}|\boldsymbol{d})$ in this expression and using the approximation in Eq. (14) gives



(a) Plot of negative log-prior distribution.

(b) Plot of negative log-likelihood.

(c) Plot of sum of negative log-prior and negative log-likelihood distribution.

(d) Plot of approximate negative log-posterior using Laplace Approximation.

Fig. 6 Plot of distributions for parameters $\boldsymbol{\theta} = [\log(\pi_1), \log(\alpha_1)]$, where the data and configuration are based on a simulation from Algorithm 1 with configuration $\boldsymbol{c} = [2, 1, 2, 3]$

$$p(\boldsymbol{c}|\boldsymbol{d}) = \frac{p(\boldsymbol{d}|\boldsymbol{\theta}, \boldsymbol{c})p(\boldsymbol{c}, \boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{c}, \boldsymbol{d})p(\boldsymbol{d})} \propto \frac{p(\boldsymbol{d}|\boldsymbol{\theta}, \boldsymbol{c})p(\boldsymbol{c}, \boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{c}, \boldsymbol{d})}$$
$$\hat{p}(\boldsymbol{c}|\boldsymbol{d}) \approx \frac{p(\boldsymbol{d}|\boldsymbol{\theta}, \boldsymbol{c})p(\boldsymbol{c}, \boldsymbol{\theta})}{\hat{p}_{\text{GA}}(\boldsymbol{\theta}|\boldsymbol{c}, \boldsymbol{d})}, \tag{16}$$

where the latter equation constructs an approximation by plugging in the PDF of the GA for $\boldsymbol{\theta}$ in the denominator of the expression. Note that $p(\boldsymbol{c}, \boldsymbol{\theta}) = p(\boldsymbol{c}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ here, and $p(\boldsymbol{c}|\boldsymbol{\theta})$ is approximated by MC sampling from the prior model for bedset boundaries. Evaluating Eq. (16) requires that a value of $\boldsymbol{\theta}$ be chosen and inserted. Even though the relation holds for any choice of $\boldsymbol{\theta}$, the relative error of the expression is minimised by using the mode $\hat{\boldsymbol{\theta}}_{\text{MAP}}$. Equation (16) is a version of the Laplace's approximation (see e.g. Rue et al. 2009).

The procedure is summarised in Algorithm 4. To find the probability of each configuration given the data, there is a loop over every configuration permitted by the data (line 2 of Algorithm 4). The possible configurations depend on the number of bedsets m and number of observations in the well, as shown in Fig. 4.

Algorithm 4 Well configuration probability approximation

1: **procedure** FINDWELLCONFIGURATIONPROBABILITIES($\mathcal{C}$, $\boldsymbol{d}$)
2: **for** $\boldsymbol{c} \in \mathcal{C}$ **do**
3: Find $\hat{\boldsymbol{\theta}}_{\text{MAP}}$ by maximising $p(\boldsymbol{\theta})p(\boldsymbol{d}|\boldsymbol{c}, \theta)$
4: Find $p(\boldsymbol{c}|\hat{\boldsymbol{\theta}}_{\text{MAP}}) = \int p(\boldsymbol{c}, \boldsymbol{z}|\hat{\boldsymbol{\theta}}_{\text{MAP}})\, \mathrm{d}\boldsymbol{z}$ by Monte Carlo integration
5: Prepare Gaussian approximation $\hat{p}_{\text{GA}}(\boldsymbol{\theta}|\boldsymbol{c}, \boldsymbol{d})$
6: Compute $\hat{p}(\boldsymbol{c}|\boldsymbol{d}) = \left[p(\boldsymbol{d}|\boldsymbol{\theta}, \boldsymbol{c})p(\boldsymbol{c}, \boldsymbol{\theta})/\hat{p}_{\text{GA}}(\boldsymbol{\theta}|\boldsymbol{c}, \boldsymbol{d})\right]_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{MAP}}}$
7: **end for**
8: **return** $\{p(\boldsymbol{c}|\boldsymbol{d}) : \boldsymbol{c} \in \mathcal{C}\}$
9: **end procedure**

4.4 Remarks About the Algorithm

Algorithm 4 is used to compute the conditional probability of all well configurations given the well data. An important step in this algorithm is to approximate the posterior for progradation and aggradation parameters, conditional on the data and the well configuration using the GA defined in Eq. (14). For each $\boldsymbol{c}$, $\hat{\boldsymbol{\theta}}_{\text{MAP}}$ is found in line 3 of Algorithm 4, and is used in line 6. In line 4 of Algorithm 4, $p(\boldsymbol{c}|\hat{\boldsymbol{\theta}}_{\text{MAP}})$ is evaluated by MC sampling. This is done as described in Algorithm 2, conditional on the parameter $\hat{\boldsymbol{\theta}}_{\text{MAP}}$ in line 3. Note that this step of approximating $p(\boldsymbol{c}|\boldsymbol{\theta})$ is not performed in the optimisation algorithm to find the GA, because doing so for every value of $\boldsymbol{\theta}$ in the optimisation steps would drastically increase the running time. Furthermore, the joint $p(\boldsymbol{c}, \boldsymbol{\theta}, \boldsymbol{d})$ is dominated by the likelihood factor $p(\boldsymbol{d}|\boldsymbol{c}, \boldsymbol{\theta})$, which means that the influence of $p(\boldsymbol{c}|\boldsymbol{\theta})$ on the posterior of $\boldsymbol{\theta}$ is small in comparison. Thus, it is reasonable to ignore this factor in optimisation.

The solution for determining the posterior model of the bedset boundaries is given by Eq. (12). Averaging is done over well configurations using MC samples from the GA of progradation and aggradation parameters $\boldsymbol{\theta}$.

5 Simulation Study

This section presents a simulation study to examine the effect of the well configuration approach (Algorithm 4). The simulation experiments test whether the proposed approach can reproduce true well configurations simulated from the prior model. This is achieved by generating multiple synthetic bedset geometries, well configurations and well datasets, and comparing the results of the suggested methodology with the truth. Recall that the well trajectory is assumed fixed and known, so the generated subsurface models are the only stochastic element. In Fig. 5 are two realisations generated from the prior model and with the fixed well locations used in this simulation study, displaying two distinct well configurations.

A goal of the simulation study is to compare the predictions obtained by the suggested approach with predictions obtained using a single, fixed configuration. In the comparison, prior configuration results are checked with expected posterior results. Mathematically, by double expectation, this means that $p(\boldsymbol{c}) = E_{\boldsymbol{d}}[p(\boldsymbol{c}|\boldsymbol{d})]$. In practice, the probabilities and expectations are computed by MC sampling.

Table 2 Symbol, description and value of initial parameters

Symbol	Description	Value
m	Number of bedsets	3
$\boldsymbol{\mu}_{\boldsymbol{\theta}}$	Expected value of $\boldsymbol{\theta}$	$[0.7, 0.4]^T$
$\Sigma_{\boldsymbol{\theta}}$	Covariance matrix of $\boldsymbol{\theta}$	$\begin{bmatrix} 0.01 & -0.001 \\ -0.001 & 0.1 \end{bmatrix}$
σ_z^2	Matérn variance parameter	0.001
η	Matérn smoothness parameter	1
ρ	Matérn length parameter	10
h	Central difference scheme step size	0.01
N	Number of MC-runs to approximate $p(\boldsymbol{c}\|\boldsymbol{\theta})$	1,000
n_r	Num. simulations in MC-approx. of $p(\boldsymbol{c})$	100,000
n_a	Num. simulations in MC-approx. of $p(\boldsymbol{c}\|\boldsymbol{d})$	10,000

5.1 Study Design

Sampling from the prior well configuration distribution $p(\boldsymbol{c})$ is done by Algorithm 2. The posterior sampling is done using Monte Carlo samples of data and computing the well configuration for each data sample. Hence, for each simulated replicate $\boldsymbol{d}^b$, $b = 1, \ldots, n_a$, Algorithm 4 estimates the posterior probabilities $p(\boldsymbol{c}|\boldsymbol{d}^b)$ for each outcome $\boldsymbol{c}$. Using double expectation, the marginal distribution $p(\boldsymbol{c})$ is then approximated by

$$\hat{p}(\boldsymbol{c}) = \int \hat{p}(\boldsymbol{c}|\boldsymbol{d}) p(\boldsymbol{d}) \, \mathrm{d}\boldsymbol{d} \approx \frac{1}{n_a} \sum_{b=1}^{n_a} p(\boldsymbol{c}|\boldsymbol{d}^b). \tag{17}$$

Increasing n_a reduces the Monte Carlo sampling error. By double expectation, samples from $\hat{p}(\boldsymbol{c})$ in equation (17) are expected to yield a similar distribution to $p(\boldsymbol{c})$ if the approximate conditioning approach performs well. This assumes that the MC sample approximations are accurate enough for reliable interpretation. The level of MC noise will be quantified in the results.

The number of samples n_r in Algorithm 2 and n_a in Eq. (17) are not necessarily the same. Posterior sampling is much more demanding than the prior sampling, and the Monte Carlo approximation could also converge more slowly. Recall that $\boldsymbol{d}$ includes both inequality and equality constraints, as described in Sect. 4.3.

Table 2 lists the initial model parameters used in the simulations. These values were chosen for geological realism at the authors' discretion.

5.2 Results

Figure 7 displays histograms of the simulated and approximated well configuration distribution. The results show that the approximation is reasonably good since the

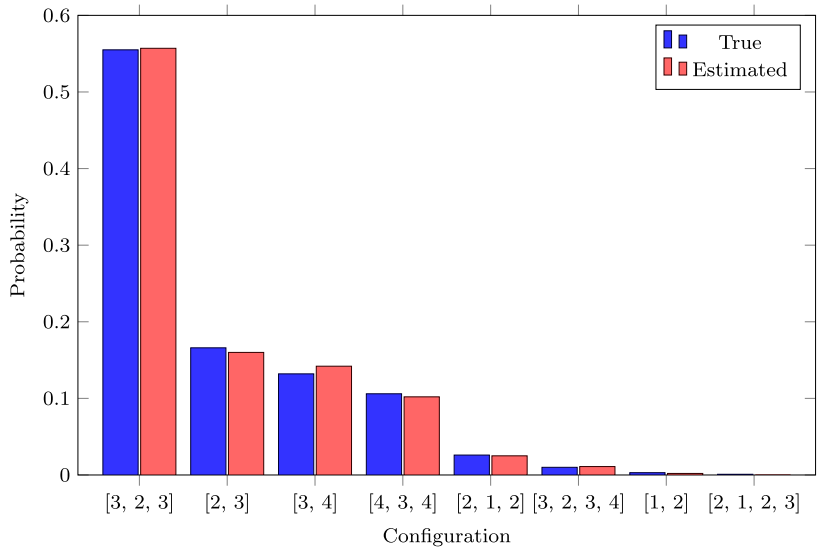


Fig. 7 Simulated well configuration probability $p(\boldsymbol{c})$ and approximated well configuration probability $\hat{p}(\boldsymbol{c})$. The prior distribution is displayed in blue, while the approximate well configuration is in red. The numerical values are also given in Table 3

distributions are very similar. Minor differences due to approximation error and MC error are expected.

Table 3 gives configuration frequencies using simulations following Algorithm 2 and approximations using Algorithm 4. Approximate 95% confidence intervals for the probabilities are also shown. These are based on the assumption that the frequency of each configuration follows a binomial distribution with a large population size, so the binomial distribution is approximated reasonably accurately by the normal distribution. Hence, the fraction $\hat{p}$ of a given configuration approximately follows an $\mathcal{N}(p, p(1-p)/n_r)$ distribution, where p is the true underlying probability of that configuration. An approximate 95% confidence interval for the configuration probability is then

$$\left[\hat{p} - \Phi^{-1}(0.025)\sqrt{\frac{\hat{p}(1-\hat{p})}{n_r}},\ \hat{p} + \Phi^{-1}(0.025)\sqrt{\frac{\hat{p}(1-\hat{p})}{n_r}}\right]. \tag{18}$$

Here, $\Phi(\cdot)$ refers to the Gaussian distribution.

Standard methods for deriving confidence intervals are not applicable for configuration probability estimates generated by Algorithm 4, due to the non-binomial distribution of these configurations. Unlike binary outcomes in simulations, each simulation here has varied probabilities for different configurations. This variation is illustrated through histograms showing the probability distribution for each configuration, as presented in Fig. 8. Specifically, Fig. 8a, d, and g demonstrate consistently low probabilities for certain configurations, while Fig. 8h, e show consistently high probabilities for others. This indicates a relative likelihood of these configurations in

Table 3 Probability and 95% confidence interval of every configuration using Monte Carlo sampling (Algorithm 2) with 10^6 iterations and using the approximate approach (Algorithm 4) with 10^5 iterations

$\boldsymbol{c}$ configuration	$p(\boldsymbol{c})$	$\hat{p}(\boldsymbol{c})$
[3, 2, 3]	0.555 (0.551, 0.559)	0.557 (0.541, 0.573)
[2, 3]	0.166 (0.159, 0.173)	0.160 (0.142, 0.178)
[3, 4]	0.132 (0.125, 0.139)	0.142 (0.125, 0.159)
[4, 3, 4]	0.106 (0.103, 0.109)	0.102 (0.092, 0.112)
[2, 1, 2]	0.026 (0.024, 0.028)	0.025 (0.020, 0.030)
[3, 2, 3, 4]	0.01	0.011
[1, 2]	0.003	0.002
[2, 1, 2, 3]	0.0009	0.000
[4, 3, 2, 3, 4]	0.0002	0.000

simulations. For configurations not clearly falling into high or low probability categories, their occurrence in simulations is less predictable. In cases where probabilities are neither 0 nor 1, the variance in configuration frequency is less than in binomial scenarios. Therefore, the binomial distribution can serve as an upper limit for the 95% confidence interval. This approach is documented in Table 3 (right). However, this method is less effective for extremely rare configurations, and in such cases, uncertainty margins are not provided. It is observed that the estimated values $\hat{p}(\boldsymbol{c})$ and $p(\boldsymbol{c})$ are within the 95% upper confidence limit, validating the effectiveness of the approximate conditioning algorithm.

To illustrate the effect of the configuration uncertainty, the probability of being in a given bedset at a given grid cell is computed under two different conditions. In both cases, the observation $\boldsymbol{d}$ is fixed. In one case the probability is computed by summing over the conditional distribution of the well configurations similar to Eq. (12). In the second case, only the most probable configuration $\hat{\boldsymbol{c}}$ is used in the prediction, without any sum accounting for the posterior uncertainty in the well configuration. The probability of a point $\boldsymbol{x}$ being in a bedset B, computed in these two ways can then be written as

$$p_1(\boldsymbol{x} \in B) = \sum_{\boldsymbol{c} \in \mathcal{C}} \int I(\boldsymbol{x} \in B|z)\, p(z|\boldsymbol{c}, \boldsymbol{d})\, p(\boldsymbol{c}|\boldsymbol{d})\, \mathrm{d}z, \tag{19}$$

and

$$p_2(\boldsymbol{x} \in B) = \int I(\boldsymbol{x} \in B|z)\, p(z|\hat{\boldsymbol{c}}, \boldsymbol{d})\, \mathrm{d}z. \tag{20}$$

Here, $\boldsymbol{x}$ is a point in the x-z coordinate system used in this paper. Hence, p_1 is the approximate bedset point probability when summing over the full configuration distribution, while p_2 represents the approximate bedset point probability by fixing the configuration at $\hat{\boldsymbol{c}}$. Both expressions are assessed by MC sampling, averaging over z samples.

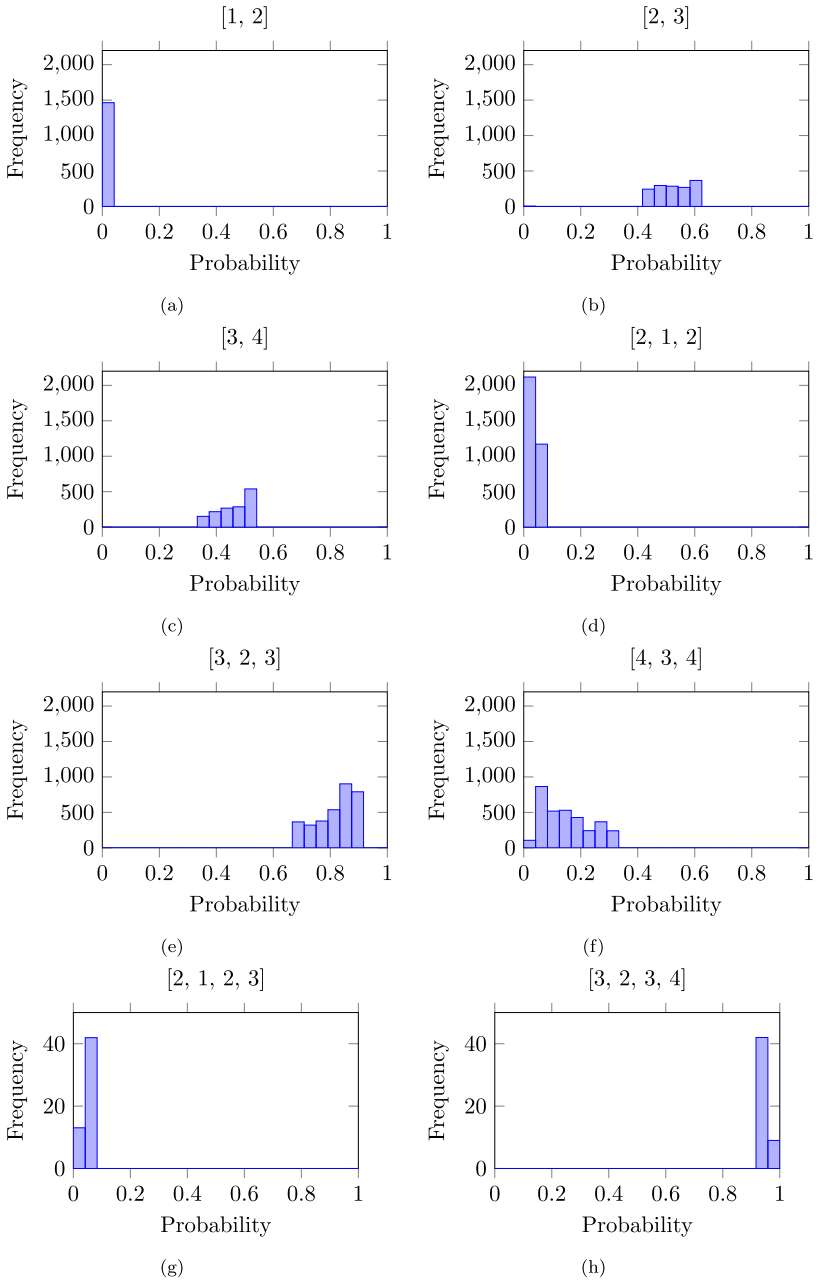


Fig. 8 Frequencies of estimated probabilities $p(\boldsymbol{c})$, produced by Algorithm 4 over repeated runs. The vertical scale is the same for the first six panels, while the bottom two plots have a much shorter y-axis

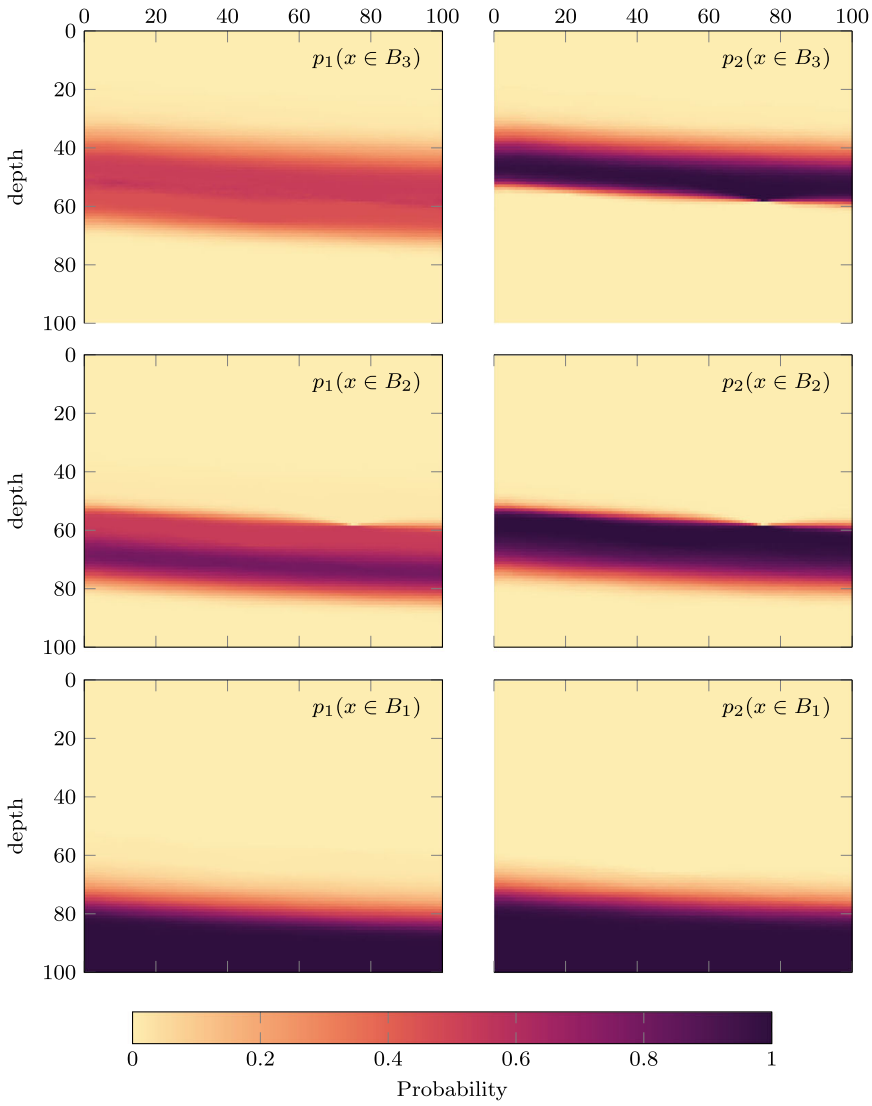


Fig. 9 Illustration of the point probabilities of bedsets. The first column shows the probabilities when summing over all possible configurations with one observed intersection. The second column shows the probabilities when fixing $\boldsymbol{c} = [2, 3]$, the most likely single intersection-configuration

Figures 9 and 10 display results where colours indicate the probability of observing specific bedsets at various locations, with each figure focusing on a single bedset. The left columns in these figures represent the full configuration distribution, while the right columns show fixed configurations. Rows in each figure correspond to different bedsets. For instance, the bottom left figure in Fig. 9 shows the probability for the bottom bedset using the full distribution. Notably, probabilities in p_1 (Eq. 19) exhibit higher variance compared to p_2 (Eq. 20), as p_2 does not include all configurations, resulting in less variability representation in bedset positions.

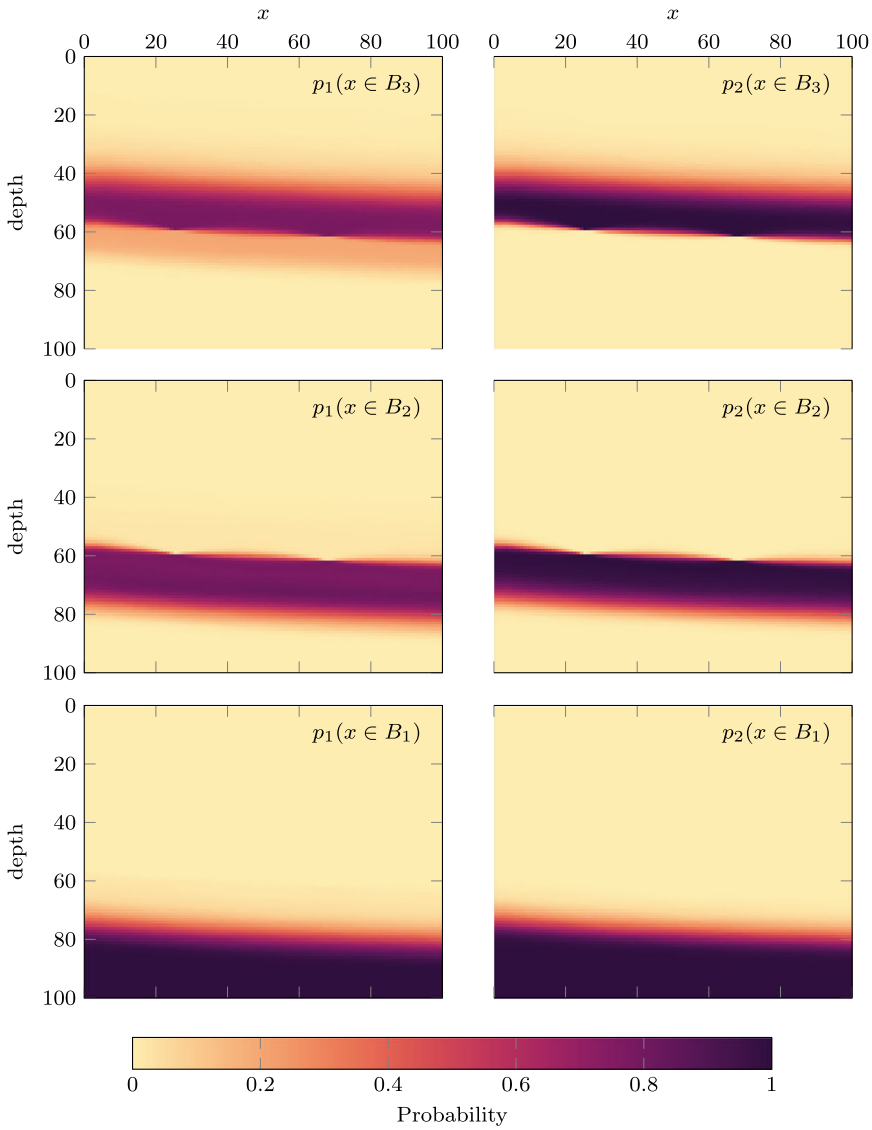
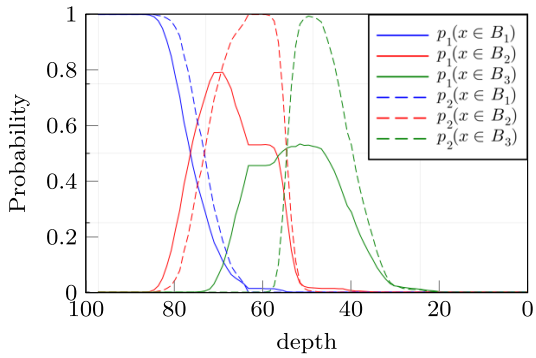
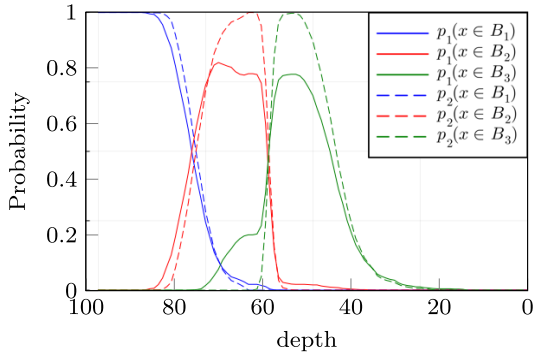


Fig. 10 Illustration of the point probabilities of bedsets. The first column shows the probabilities when summing over all possible configurations with intersections. The second column shows the probabilities when fixing the configuration to [3, 2, 3], the most likely configuration with two intersections

Figure 11 shows cross sections of the bedset point probabilities by fixing the x-coordinate to $x = 50$. Figure 11a, b illustrate both probabilities for all three bedsets with one and two well bedset transition observations, respectively. The probabilities p_1 display flatter curves, representing the increased uncertainty in bedset position when averaging over configurations. Also, the overlap between curves is greater for probabilities p_1, meaning that multiple bedsets can occupy the same space for different configurations.



(a) Illustration of probabilities in Fig. 9, when fixing $x = 50$.



(b) Illustration of probabilities in Fig. 10, when fixing $x = 50$.

Fig. 11 Illustration of point probabilities of bedsets, when fixing $x = 50$

Bedset information can be very important for reservoir flow. Although this study refrains from direct flow simulation, it is interesting to study the lengths of well trajectory segments lying within each bedset as a proxy for flow behaviour. The significance of these well lengths stems from the fact that flow differs both between and within bedsets. In evaluating well lengths, it is essential to consider the well configurations. For instance, a well crossing three bedset transitions can exhibit various configurations, each leading to distinct lengths.

Figure 12 illustrates the well length distributions for the case of two bedset transitions. Similar to what was done for the point probabilities, the length distributions are shown both for the mode configuration $c = [3, 2, 3]$ and for the weighted sum across all possible configurations. Key observations include:

- For bedset 2, the mode configuration shows a unimodal distribution with a mode at around $x = 52$ with a constant rise in density from the left. Weighting over all configurations, there is probability 0.154 of having 0 well length, which is the contribution from configuration $\boldsymbol{c} = [4, 3, 4]$; this gives a lower overall density for the remaining well lengths.
- For bedset 3, the mode configuration reveals a unimodal distribution with a mode centred roughly at $x = 8$. The average across configurations also has some probability for zero-well length due to the configuration $\boldsymbol{c} = [2, 1, 2]$ of prob-

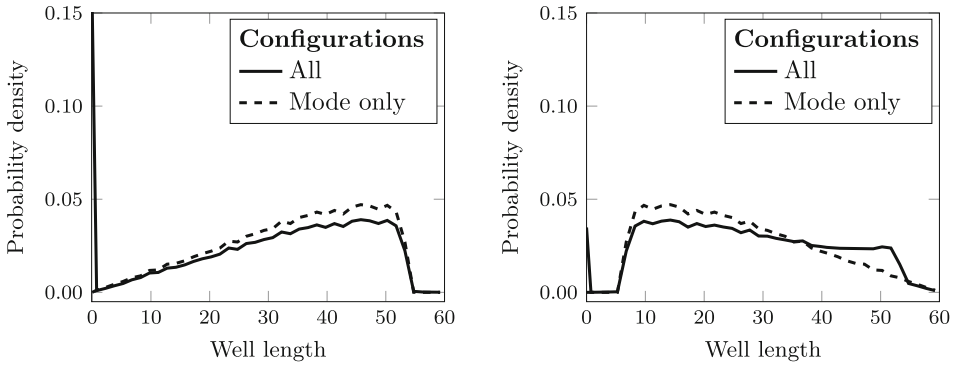


Fig. 12 PDFs of well lengths within bedsets 2 and 3 for three bedset transition observations in well. Plots are both for the mode configuration $c = [3, 2, 3]$ and weighted over all configurations

ability 0.035. The PDF also shows a flatter curve, which is the contribution from $\boldsymbol{c} = [4, 3, 4]$, giving larger well lengths.

These differences in the distributions might have profound implications for flow simulations. In this situation, one risks ignoring the high probability of diminished zero extraction from bedsets 2 and 3, and potentially higher extraction from bedset 3.

5.3 Discussion

Figures 9 and 10 highlight some of the issues of existing conditional models. Merely assuming a well configuration risks under-representing the variance of conditional objects. Moreover, Fig. 12 shows that this might have an effect on flow simulation as well. In Fig. 9 which showcases the scenario with a single bedset boundary intersection ($\boldsymbol{c} = [2, 3]$), the under-representation of variance is more distinct than in Fig. 10. This occurs because configurations with more data (here two bedset intersections rather than just one) have a smaller variance than configurations with one bedset intersection. Thus, the modelling assumption that some possible well configurations can be safely ignored clearly depends on the well configuration distribution. In some cases, when the prior model parameters are very strict, or when the equality constraints contain a lot of information, this assumption can lead to an excellent approximation of the true distribution. However, with sparse data or a diffuse prior, or both, it is a questionable assumption at best.

The explicit calculation of $p(\boldsymbol{c}|\boldsymbol{d})$ works only when the combinatorics involved are within reasonable limits. In practice, this means that the number of possible well configurations cannot be too large. As shown in Fig. 4, the possible configurations can be represented as paths in a graph. When the number of observations grows large and there are many bedsets, the number of paths increases rapidly (Sloane 2014). For example, when $m = 8$ and $n = 12$, there are 13,884 possible configurations. Computationally, looping through all combinations in Algorithm 4 quickly becomes infeasible. Thus, extending the applicability of the approach described in this article to situations

with more observations or bedsets will require a way to eliminate infeasible paths to improve the combinatorics. In the scientific literature, various methods have been explored to mitigate uncertainty in well data, particularly for sampling stratigraphic configurations. The study referenced as (Lallier et al. 2012) employs correlation rules to align well logs of stratigraphic sequences. This correlation is based on the premise that two wells can be matched if they exhibit not only similar diagenetic characteristics but also comparable well log patterns. On the other hand, Baville et al. (2022) introduces the use of Dynamic Time Warping (DTW) for correlating stratigraphic sequences across wells. This approach hinges on the assumption that stratigraphic sequences within a well should have synchronous ages.

Although these methods share certain similarities with the research presented in this paper, our focus diverges significantly. We aim to determine the probabilities of various stratigraphic sequences within a single well log, grounded on specific prior assumptions. In contrast, the studies in Lallier et al. (2012) and Baville et al. (2022) primarily concentrate on correlating well logs to infer similarities in depositional environments among observed units.

6 Closing Remarks

6.1 General Thoughts

In this article, a novel method for conditioning a surface-based model to non-vertical wells has been presented. This approach differs from conventional conditioning methods in emphasising the role of well configurations, the coupling between observations and modelled objects. Whereas configurations are typically treated as implicit parts of models, or as byproducts of conditioning procedures, the probabilistic model and conditioning algorithm discussed here represents configurations explicitly and makes them a core component of the conditioning process. The overarching idea is that generating conditional realisations is hard to do in the general case, but becomes relatively easy if a configuration is known. Once the configuration concept has been made explicit, the wider conditioning problem is, in a certain sense, reduced to the specific problem of configuration conditioning.

6.2 Limitations and Area of Applicability

There are some clear limitations to the method presented here, mainly in the number of possible configurations that can be handled. Each evaluation of $p(\boldsymbol{c}|\boldsymbol{d})$ is costly, and the number of evaluations increases linearly with the number of possible configurations. Furthermore, as the number of possible configurations increases, the MC sampling of $p(\boldsymbol{c}|\boldsymbol{\theta}_{\mathrm{MAP}})$ becomes less stable, requiring more iterations. Thus, the cost per evaluation also increases.

Another limiting factor is the ability to find a good estimate for $\boldsymbol{\theta}_{\mathrm{MAP}}$. As more data and parameters are added, the optimisation of the posterior distribution gets

increasingly difficult. The quality of the Gaussian approximation for parameters could also get worse.

However, in a simple setting, such as the one described here, the approach works very well. This makes it suitable as a reference tool for more complex algorithms trying to handle larger data sets. It validates implementations in terms of reproducing correct configuration probabilities, and more complex algorithms must agree with the one presented here on small data sets.

6.3 Further Work

The scope of this article is limited to a single non-vertical well. A natural extension of this work is to allow multiple non-vertical wells. This would require mitigating or overcoming some of the limitations described in Sect. 6.2. Another natural direction for further investigation is the introduction of additional parameters in the prior model to enable representation of more complex geological structures. For example, allowing bedsets to erode the underlying deposit, have negative aggradation or progradation, and to pinch out rather than extend to the edge of the model domain. A more complex prior model would give more realistic posterior realisations. Such a refinement would make the model more applicable to real data.

For the challenge of conditioning geologically realistic facies models, one must in this situation connect the bedset models with a facies model in each bedset. For this task, there is currently much interesting work in developing machine learning approaches for the conditioning problem here such as generative adversarial networks (GANs); see, for example, Song et al. (2021) and Feng et al. (2022) or diffusion models (Lee et al. 2023).

Acknowledgements We acknowledge support from the Norwegian Research Council project GEOPARD grant 319951 and the SFI Centre for Geophysical Forecasting grant 309960.

Funding Open access funding provided by NTNU Norwegian University of Science and Technology (incl St. Olavs Hospital - Trondheim University Hospital)

Declarations

Conflict of interest The authors have no conflict of interest. This work has not been submitted elsewhere.

References

Abrahamsen P (1993) Bayesian kriging for seismic depth conversion of a multi-layer reservoir. In: Soares A (ed) Geostatistics Tróia 92. Quantitative geology and geostatistics, vol 5. Springer, Dordrecht. https://doi.org/10.1007/978-94-011-1739-5_31

Abramowitz M, Stegun IA (1972) Handbook of mathematical functions with formulas, graphs, and mathematical tables, vol 55. National Bureau of standards applied mathematics series. Tenth Printing, ERIC, New York

Allard D, Fabbri P, Gaetan C (2021) Modeling and simulating depositional sequences using latent gaussian random fields. Math Geosci 53:469–497

Baville P, Apel M, Hoth S, Knaust D, Antoine C, Carpentier C, Caumon G (2022) Computer-assisted stochastic multi-well correlation: sedimentary facies versus well distality. Mar Petrol Geol 135:105371

Bertoncello A, Sun T, Li H, Mariethoz G, Caers J (2013) Conditioning surface-based geological models to well and thickness data. Math Geosci 45:873–893

Boggs S (2014) Principles of sedimentology and stratigraphy. Pearson Education Limited, London

Bridge JS, Leeder MR (1979) A simulation model of alluvial stratigraphy. Sedimentology 26(5):617–644

Bubnova A, Ors F, Rivoirard J, Cojan I, Romary T (2020) Automatic determination of sedimentary units from well data. Math Geosci 52(2):213–231

Catuneanu O, Abreu V, Bhattacharya J, Blum M, Dalrymple R, Eriksson P, Fielding CR, Fisher W, Galloway W, Gibling M et al (2009) Towards the standardization of sequence stratigraphy. Earth Sci Rev 92(1–2):1–33

Cojan I, Fouché O, Lopéz S, Rivoirard J (2005) Process-based reservoir modelling in the example of meandering channel. Geostatistics Banff 2004:611–619

Cressie N, Wikle CK (2015) Statistics for spatio-temporal data. Wiley, New York

Deutsch CV, Wang L (1996) Hierarchical object-based stochastic modeling of fluvial reservoirs. Math Geol 28:857–880

Eide CH, Howell J, Buckley S (2014) Distribution of discontinuous mudstone beds within wave-dominated shallow-marine deposits: star point sandstone and Blackhawk formation, Eastern Utah. AAPG Bull 98(7):1401–1429

Eide CH, Howell JA, Buckley SJ (2015) Sedimentology and reservoir properties of tabular and erosive offshore transition deposits in wave-dominated, shallow-marine strata. EAGE/Geological Society of London, Book Cliffs

Feng R, Grana D, Mukerji T, Mosegaard K (2022) Application of bayesian generative adversarial networks to geological facies modeling. Math Geosci 54(5):831–855

Goff JA (2000) Simulation of stratigraphic architecture from statistical and geometrical characterizations. Math Geol 32:765–786

Graham GH, Jackson MD, Hampson GJ (2015a) Three-dimensional modeling of clinoforms in shallow-marine reservoirs: part 1. Concepts and application. AAPG Bull 99(06):1013–1047

Graham GH, Jackson MD, Hampson GJ (2015b) Three-dimensional modeling of clinoforms in shallow-marine reservoirs: part 2. Impact on fluid flow and hydrocarbon recovery in fluvial-dominated deltaic reservoirs. AAPG Bull 99(06):1049–1080

Hauge R, Holden L, Syversveen AR (2007) Well conditioning in object models. Math Geol 39:383–398

Hauge R, Vigsnes M, Fjellvoll B, Vevle ML, Skorstad A (2017) Object-based modeling with dense well data. Geostat Valencia 2016:557–572

Holden L, Hauge R, Skare Ø, Skorstad A (1998) Modeling of fluvial reservoirs with object models. Math Geol 30:473–496

Jo H, Pyrcz MJ (2020) Robust rule-based aggradational lobe reservoir models. Nat Resour Res 29:1193–1213

Keogh KJ, Martinius AW, Osland R (2007) The development of fluvial stochastic modelling in the Norwegian oil industry: a historical review, subsurface implementation and future directions. Sed Geol 202(1–2):249–268

Lallier F, Caumon G, Borgomano J, Viseur S, Fournier F, Antoine C, Gentilhomme T (2012) Relevance of the stochastic stratigraphic well correlation approach for the study of complex carbonate settings: application to the Malampaya buildup (Offshore Palawan, Philippines). In: Advances in carbonate exploration and reservoir analysis. Geological Society of London

Lee D, Ovanger O, Eidsvik J, Aune E, Skauvold J, Hauge R (2023) Latent diffusion model for conditional reservoir facies generation. arXiv preprint arXiv:2311.01968

Manzocchi T, Walsh DA (2023) Vertical stacking statistics of multi-facies object-based models. Math Geosci 55(4):461–496
Mogensen PK, Riseth AN (2018) Optim: a mathematical optimization package for Julia. J Open Source Softw 3(24):615
Parquer MN, Collon P, Caumon G (2017) Reconstruction of channelized systems through a conditioned reverse migration method. Math Geosci 49(8):965–994
Pyrcz MJ, Deutsch CV (2014) Geostatistical reservoir modeling. Oxford University Press, Oxford
Pyrcz MJ, Sech RP, Covault JA, Willis BJ, Sylvester Z, Sun T (2015) Stratigraphic rule-based reservoir modeling. Bull Can Pet Geol 63(4):287–303
Rongier G, Collon P, Renard P (2017) A geostatistical approach to the simulation of stacked channels. Mar Pet Geol 82:318–335
Rue H, Martino S, Chopin N (2009) Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. J Roy Stat Soc Ser B (Stat Methodol) 71(2):319–392
Seifert D, Jensen J (2000) Object and pixel-based reservoir modeling of a braided fluvial reservoir. Math Geol 32:581–603
Skauvold J, Eidsvik J (2018) Data assimilation for a geological process model using the ensemble Kalman filter. Basin Res 30(4):730–745
Sloane NJA (2014) A handbook of integer sequences. Academic Press, Cambridge
Song S, Mukerji T, Hou J (2021) GANsim: conditional facies simulation using an improved progressive growing of generative adversarial networks (GANs). Math Geosci 53:1413–1444
Titus Z, Heaney C, Jacquemyn C, Salinas P, Jackson M, Pain C (2021) Conditioning surface-based geological models to well data using artificial neural networks. Comput Geosci 26:779–802
Troncoso A, Freulon X, Lantuéjoul C (2022) Sequential simulation of a conditional Boolean model. Math Geosci 54(2):389–411
Viseur S, Shtuka A, Mallet JL (1998) New fast, stochastic, Boolean simulation of fluvial deposits. In: SPE annual technical conference and exhibition. SPE, pp 697–709
Wang YC, Pyrcz MJ, Catuneanu O, Boisvert JB (2018) Conditioning 3D object-based models to dense well data. Comput Geosci 115:1–11
Wingate D, Kane J, Wolinsky M, Sylvester Z (2016) A new approach for conditioning process-based geologic models to well data. Math Geosci 48:371–397

Latent diffusion model for conditional reservoir facies generation

Daesoo Lee, Oscar Ovanger, Jo Eidsvik, Erlend Aune, Jacob Skauvold and Ragnar Hauge

Submitted

Computers & Geosciences 194 (2025) 105750

Contents lists available at ScienceDirect

Computers and Geosciences

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

Research paper

Latent diffusion model for conditional reservoir facies generation

Daesoo Lee [a,*], Oscar Ovanger [a], Jo Eidsvik [a], Erlend Aune [a,c,d], Jacob Skauvold [b], Ragnar Hauge [b]

[a] Norwegian University of Science and Technology, Alfred Getz' vei 1, 7034 Trondheim, Norway
[b] Norwegian Computing Centre, Gaustadalléen 23A, 0373 Oslo, Norway
[c] BI Norwegian Business School, Nydalsveien 37, 0484 Oslo, Norway
[d] HANCE, Ålesundsgata 3D, 0470 Oslo, Norway

ARTICLE INFO

Dataset link: https://figshare.com/articles/dataset/Dataset_used_in_Latent_Diffusion_Model_for_Conditional_Reservoir_Facies_Generation/26892868?file=48931588

Keywords:
Machine learning
Diffusion models
Facies generation
Conditional sampling
Statistical learning

ABSTRACT

Creating accurate and geologically realistic reservoir facies based on limited measurements is crucial for field development and reservoir management, especially in the oil and gas sector. Traditional two-point geostatistics, while foundational, often struggle to capture complex geological patterns. Multi-point statistics offers more flexibility, but comes with its own challenges related to pattern configurations and storage limits. With the rise of Generative Adversarial Networks (GANs) and their success in various fields, there has been a shift towards using them for facies generation. However, recent advances in the computer vision domain have shown the superiority of diffusion models over GANs. Motivated by this, a novel Latent Diffusion Model is proposed, which is specifically designed for conditional generation of reservoir facies. The proposed model produces high-fidelity facies realizations that rigorously preserve conditioning data. It significantly outperforms a GAN-based alternative. Our implementation on GitHub: github.com/ML4ITS/Latent-Diffusion-Model-for-Conditional-Reservoir-Facies-Generation

1. Introduction

Creating accurate and geologically realistic reservoir facies predictions based on limited measurements is a critical task in development and production of oil and gas resources. It is also very relevant in connection with CO_2 storage, where one makes decisions about injection strategies to manage leakage risk and ensure safe long-term operations. In both contexts, key operational decisions are based on realizations of stochastic reservoir models. Through the use of multiple realizations, one can go beyond point-wise prediction of facies, and additionally quantify spatial variability and correlation. This gives better descriptions of the relevant heterogeneity.

When generating facies realizations, one must honor both geological knowledge and reservoir-specific data. A wide range of stochastic models have been proposed to solve this problem. A good overview can be found in the book by Pyrcz and Deutsch (2014). There are variogram-based models, where the classical concept of a variogram-based Gaussian field (see for instance Cressie, 2015) is combined with a discretization scheme to generate facies. Then there are more geometric approaches, such as object models or process-mimicking models, where facies are described as geometric objects with an expected shape and uncertainty. Of particular interest here are multiple-point models, which use a training image to generate a pattern distribution, and then generate samples following this distribution.

Multiple-point models are very flexible, and allow for complex interactions between any number of facies. But as the method fundamentally hinges on storing pattern counts, there are strict limitations due to memory. In practice, only a limited number of patterns can be handled, leading to restrictions in pattern size and a demand for stationarity of patterns. Furthermore, the simulation algorithm has clear limitations in its ability to reproduce the patterns, so a realization will typically contain many patterns not found in the initial database, leading to unwanted geometries (Zhang et al., 2019). Limitations like these have led to the adoption of models such as generative adversarial networks (GANs, Goodfellow et al., 2020). In recent years, GANs have gained substantial attention for the conditional generation of realistic facies while retaining conditional data in a generated sample, see e.g. Chan and Elsheikh (2019), Zhang et al. (2019), Azevedo et al. (2020), Pan et al. (2021), Song et al. (2021), Zhang et al. (2021), Yang et al. (2022), Razak and Jafarpour (2022) and Hu et al. (2023).

We frame stochastic facies modeling as a conditional generation problem in machine learning. This view is motivated by the observation that in some existing methods for reservoir modeling, generating unconditional realizations is comparatively easy, and the difficulty increases sharply as one moves to generating conditional realizations. The principal idea of this paper is to exploit this difficulty gap by using easily generated unconditional realizations as training data for a

* Corresponding author.
E-mail address: daesoo.lee@ntnu.no (D. Lee).

https://doi.org/10.1016/j.cageo.2024.105750
Received 25 January 2024; Received in revised form 6 September 2024; Accepted 27 October 2024
Available online 1 November 2024

Fig. 1. Illustration of our conditional reservoir generation problem in which the generative model stochastically samples a realistic reservoir (right) given the limited measurements (left). The regions with no information are denoted in gray.

machine learning model. Crucially, this model will learn not only how to reproduce features seen in the training realizations, but also how to honor conditioning data. A model successfully trained in this way can generate conditional realizations given previously unseen conditioning data. Fig. 1 illustrates our conditional generation problem.

Recent studies in computer vision have demonstrated the superiority of diffusion models over GANs in terms of generative performance (Dhariwal and Nichol, 2021; Rombach et al., 2022; Ho et al., 2022; Kim et al., 2022). As a result, diffusion models are state-of-the-art for image generation, while the popularity of GANs has diminished due to limitations including convergence problems, mode collapse, generator-discriminator imbalance, and sensitivity to hyperparameter selection. Latent diffusion models (LDMs) are a type of diffusion model in which the diffusion process occurs in a latent space rather than in pixel space (Rombach et al., 2022). LDMs have become popular because they combine computational efficiency with good generative performance.

Motivated by the progress made with diffusion models on computer vision and image processing tasks, this work proposes a novel LDM, specifically designed for the generation of conditional facies realizations in a reservoir modeling context. Its appeal lies in the ability to strictly preserve conditioning data in the generated realizations. To the authors' knowledge, this is the first work to adopt a diffusion model for conditional facies generation.

Experiments were carried out using a dataset of 5000 synthetic 2D facies realizations to evaluate the proposed diffusion model against a GAN-based alternative. The diffusion model achieved robust conditional facies generation performance in terms of fidelity, sample diversity, and the preservation of conditional data, while the GAN-based model struggled with multiple critical weaknesses.

To summarize, the contributions of this paper are:

- the adoption of a diffusion model for conditional facies generation,
- a novel LDM, designed to preserve observed facies data in generated samples,
- conditional facies generation with high fidelity, sample diversity, and robust preservation.

In Section 2, we describe GANs and background information for the LDMs. In Section 3, we present our suggested methodology for conditional facies realizations with LDMs. In Section 4, we show experimental results of our method applied to a bedset model with stacked facies, including the comparison with GANs. In Section 5, we summarize and discuss future work.

2. Background on generative models

2.1. Generative adversarial network for conditional image generation

GANs were a breakthrough innovation in the field of generative AI when they emerged in 2014. The core mechanism of GANs involves two neural networks, a generator and a discriminator, engaged in a sort of cat-and-mouse game. The generator aims to mimic the real data, while the discriminator tries to distinguish between real and generated data. Through iterative training, the generator improves its ability to create realistic data, and the discriminator becomes more adept at identifying fakes.

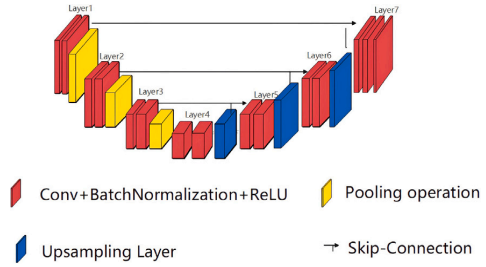


Fig. 2. Illustration of the U-Net architecture (Cai et al., 2022), where Conv denotes a convolutional layer. U-Net is a convolutional neural network architecture, featuring an encoder (first half of U-Net) and decoder (second half) structure with skip connections that allow for the transfer of spatial information across layers, which in turn enables precise localization and high-resolution output.

Conditional Generative Adversarial Networks (CGANs) were proposed by Mirza and Osindero (2014) in the same year as the GAN. The CGAN was designed to guide the image generation process of the generator given conditional data such as class labels and texts as auxiliary information. Since then, CGANs have been further developed to perform various tasks. Among these, Isola et al. (2017) stands out from the perspective of conditional facies generation, proposing a type of CGAN called Pixel2Pixel (Pix2Pix), which has become a popular GAN method for image-to-image translation. Pix2Pix works by training a CGAN to learn a mapping between input images and output images from different distributions. For instance, the input could be a line drawing, and the output a corresponding color image. The mapping can be realized effectively with the help of the U-Net architecture (Ronneberger et al., 2015), illustrated in Fig. 2.

Image-to-image translation is directly relevant to conditional facies generation because the input can be facies observations on a limited subset of the model domain, and the output can be a complete facies model. This is the typical situation when the goal is to generate 2D or 3D facies realizations from sparse facies observations at the well locations.

2.2. GANs for conditional facies generation

Dupont et al. (2018) were the first to adopt a GAN for conditional facies generation, overcoming the limitations of traditional geostatistical methods by producing varied and realistic geological patterns that honor measurements at data points. However, the latent vector search required to ensure a match with the conditioning data makes the sampling process inefficient. Chan and Elsheikh (2019) introduced a second inference network that enables the direct generation of realizations conditioned on observations, thus providing a more efficient conditional sampling approach. Zhang et al. (2019) introduced a GAN-based approach to generate 3D facies realizations, specifically focusing on complex fluvial and carbonate reservoirs. Their paper clearly demonstrated the superiority of GAN over MPS for this application. Azevedo

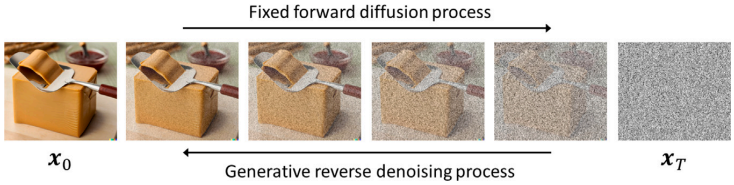


Fig. 3. Illustration of the principle of a diffusion process. The diffusion modeling mainly consists of (1) forward process (noising) and (2) reverse process (denoising). The noising process begins with a data sample and incrementally adds Gaussian noise over multiple time steps to convert it into a Gaussian noise sample; conversely, the denoising process iteratively refines this Gaussian noise sample back into a data-like sample, guided by a neural network trained specifically for this denoising task.

et al. (2020) used GANs in a similar way, but the evaluation of its conditional generation makes this study different from others. Where GANs from other studies typically condition on multiple sparse points, the paper considered conditioning on patches and lines. Because such shapes typically involve a larger region than multiple sparse points, their conditional setup is more difficult, which is demonstrated in experiments. Pan et al. (2021) used Pix2Pix, adopting the U-Net architecture. It takes a facies observation and noise as input, and then stochastically outputs a full facies realization. Notably, the preservation of conditional data in a generated sample was shown to be effective due to the U-Net architecture that enables precise localization. A paper by Zhang et al. (2021) is concurrent with and methodologically similar to Pan et al. (2021) as both articles propose a GAN built on U-Net. However, the U-Net GAN of Zhang et al. (2021) has an additional loss term to ensure sample diversity, which simplifies the sampling process. Subsequently, many studies have sought to improve conditional facies generation using GANs, working within the same or similar frameworks as the studies mentioned above (Song et al., 2021; Yang et al., 2022; Razak and Jafarpour, 2022; Hu et al., 2023).

The main difference between the current study and previous research is the type of generative model employed, specifically the choice of a diffusion model over a GAN. This also leads to a specific network architecture used to enable conditioning. Another difference is that whereas much earlier work is done in a top-down view, we focus on a vertical section. A consequence of this is that we get a different structure for the conditioning data. In a vertical section, well data become paths, giving connected lines of cells with known facies. In the top-down view, wells appear as scattered individual grid cells with known facies.

2.3. Denoising Diffusion Probabilistic Model (DDPM)

Ho et al. (2020) represented a milestone for diffusion model-based generative modeling. DDPMs offer a powerful framework for generating high-quality image samples from complex data distributions. At its core, a DDPM leverages the principles of diffusion processes to model a data distribution. It operates by iteratively denoising a noisy sample and gradually refining it to generate a realistic sample as illustrated in Fig. 3. This denoising process corresponds to the reverse process of a fixed Markov process of a certain length.

A DDPM employs a denoising autoencoder, denoted by $\epsilon_\theta(\boldsymbol{x}_t, t); t = 1, \ldots, T$. The denoising autoencoder gradually refines the initial noise $\boldsymbol{x}_T$ to generate a high-quality sample $\boldsymbol{x}_0$ that closely resembles the target data distribution. A U-Net is used for the denoising autoencoder since its architecture provides effective feature extraction, preservation of spatial details, and robust performance in modeling complex data distributions (Baranchuk et al., 2022).

Training: Prediction of noise in $\boldsymbol{x}_t$. DDPM training consists of two key components: the non-parametric forward process and the parameterized reverse process. The former component represents the gradual addition of Gaussian noise. In contrast, the reverse process needs to be learned to predict noise $\boldsymbol{\epsilon}$ in $\boldsymbol{x}_t$. Its loss function is defined by

$$L_{\text{DDPM}} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t}\left[\|\boldsymbol{\epsilon} - \epsilon_\theta(\boldsymbol{x}_t, t)\|_2^2\right], \tag{1}$$

where ϵ_θ denotes the denoising autoencoder with parameters $\boldsymbol{\theta}$. Eq. (1) measures the discrepancy between the noise and the predicted noise by the denoising autoencoder. While Eq. (1) defines a loss function for unconditional generation, the loss for conditional generation is specified by

$$L_{\text{DDPM},c} = \mathbb{E}_{\boldsymbol{x},c,\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t}\left[\|\boldsymbol{\epsilon} - \epsilon_\theta(\boldsymbol{x}_t, t, c)\|_2^2\right], \tag{2}$$

where c denotes conditional data such as texts or image class, and in our situation, the observed facies classes in wells. Typically, L_{DDPM} and $L_{\text{DDPM},c}$ are both minimized during training, to allow both unconditional and conditional generation. For details, see Ho and Salimans (2021).

Sampling via learned reverse process. The forward diffusion process is defined as $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\mathbf{I})$ where β_t is called a variance schedule and $1 \geq \beta_T > \beta_1 \geq 0$. Equivalently, it can be written $\boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}_{t-1}$ with $\boldsymbol{\epsilon}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We further reformulate the equation with respect to $\boldsymbol{x}_{t-1}$ and it becomes

$$\boldsymbol{x}_{t-1} = (\boldsymbol{x}_t - \sqrt{\beta_t}\boldsymbol{\epsilon}_{t-1})/\sqrt{1-\beta_t} = (\boldsymbol{x}_t - \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1})/\sqrt{\alpha_t}, \tag{3}$$

where $\alpha_t = 1 - \beta_t$. Then we can go backwards, sampling $\boldsymbol{x}_0$ from $\boldsymbol{x}_T$ by recursively applying Eq. (3) for $t = T, \ldots, 2, 1$.

2.4. Latent Diffusion Model (LDM)

LDMs extend DDPMs by introducing a diffusion process in a latent space. The main idea of LDMs is illustrated in Fig. 4, aligning with the overview of our proposed method for conditional facies generation as depicted in Fig. 5. In the common situation, data are typically text or images as indicated to the far right in Fig. 4. In our setting, the conditional data are facies observations along a few well paths in the subsurface.

Compared with a DDPM, an LDM has two additional components: encoder $\mathcal{E}$ and decoder D. The encoder transforms $\boldsymbol{x}$ into a latent representation, $\boldsymbol{z} = \boldsymbol{z}_0 = \mathcal{E}(\boldsymbol{x})$, while the decoder reconstructs $\boldsymbol{z}$ to produce $\tilde{\boldsymbol{x}}$. Importantly, the encoding and decoding processes involve downsampling and upsampling operations, respectively. The encoder and decoder are trained so that $\tilde{\boldsymbol{x}}$ is as close as possible to $\boldsymbol{x}$. This is ensured by minimizing a reconstruction loss between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$. Notably, the forward and backward processes are now taking place in the latent space, therefore $\boldsymbol{z}_T$ denotes a Gaussian noise sample. In Fig. 4, $\mathcal{E}_c$ denotes an encoder for conditional data. The encoded conditional data is fed into the reverse process for conditioning the generation process.

The main advantage of LDMs over DDPMs is computational efficiency. The encoder $\mathcal{E}$ compresses high-dimensional data $\boldsymbol{x}$ into a lower-dimensional latent space represented via latent variable $\boldsymbol{z}$. This dimensionality reduction significantly reduces the computational cost, making LDM more feasible to be trained on local devices. However, a trade-off exists between computational efficiency and sample quality. Increasing the downsampling rate of $\mathcal{E}$ increases the computational efficiency but typically results in a loss of sample quality, and vice versa.

Training of LDMs adopts a two-staged modeling approach (Van Den Oord et al., 2017; Chang et al., 2022). The first stage (stage 1) is

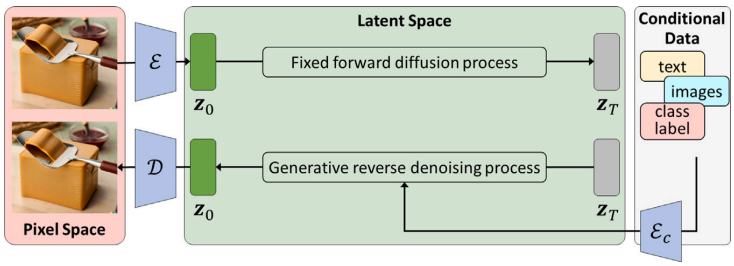


Fig. 4. Overview of LDM. The encoder $\mathcal{E}$ and decoder $\mathcal{D}$ enable data compression, enabling the forward and reverse processes to operate in a reduced-dimensional space. This eases the task of learning prior and posterior distributions and improves computational efficiency. In addition, conditional data can be fed into the reverse process, enabling conditional generation.

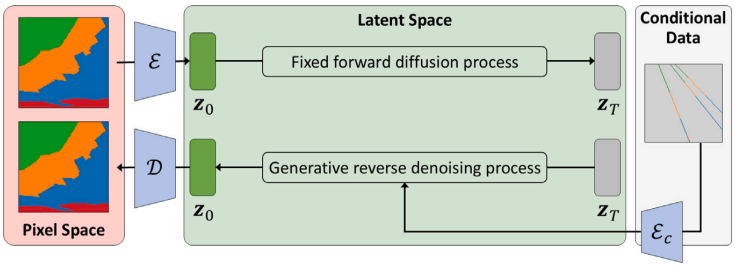


Fig. 5. Overview of our proposed method. Our method can be regarded as an adapted version of LDM to effectively handle the categorical input and allow maximal preservation of conditional facies data in generated facies while maintaining the high fidelity of generated facies.

for learning the compression and decompression of $\boldsymbol{x}$ by training $\mathcal{E}$ and $\mathcal{D}$, and the second stage (stage 2) is for learning the prior and posterior distributions by training ϵ_θ.

In stage 1, $\boldsymbol{x}$ is encoded into $\boldsymbol{z}$ and decoded back into the data space. The training of $\mathcal{E}$ and $\mathcal{D}$ is conducted by minimizing the following reconstruction loss:

$$\|\boldsymbol{x} - D(\mathcal{E}(\boldsymbol{x}))\|_2^2. \tag{4}$$

In stage 2, the denoising autoencoder ϵ_θ is trained to learn prior and posterior distributions, while $\mathcal{E}$ and D are set to be untrainable (frozen). This involves minimizing

$$L_{\text{LDM}} = \mathbb{E}_{\mathcal{E}(\boldsymbol{x}),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t}\left[\|\boldsymbol{\epsilon} - \epsilon_\theta(\boldsymbol{z}_t, t)\|_2^2\right], \qquad \text{Prior training,} \tag{5}$$

$$L_{\text{LDM},c} = \mathbb{E}_{\mathcal{E}(\boldsymbol{x}),c,\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t}\left[\|\boldsymbol{\epsilon} - \epsilon_\theta(\boldsymbol{z}_t, t, c)\|_2^2\right], \qquad \text{Posterior training.} \tag{6}$$

The recent work that proposed DALLE-2 (Ramesh et al., 2022) empirically found that predicting $\boldsymbol{z}_0$ instead of $\boldsymbol{\epsilon}$ results in better training. We adopt the same approach for better training and methodological simplicity of our conditional sampling. Hence, we in stage 2 instead minimize

$$L_{\text{LDM}}(\boldsymbol{z}, g_\theta) = \mathbb{E}_{\mathcal{E}(\boldsymbol{x}),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t}\left[\|\boldsymbol{z}_0 - g_\theta(\boldsymbol{z}_t, t)\|_2^2\right], \tag{7}$$

$$L_{\text{LDM},c}(\boldsymbol{z}, c, g_\theta) = \mathbb{E}_{\mathcal{E}(\boldsymbol{x}),c,\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t}\left[\|\boldsymbol{z}_0 - g_\theta(\boldsymbol{z}_t, t, c)\|_2^2\right], \tag{8}$$

where g_θ is a denoising autoencoder that predicts $\boldsymbol{z}_0$ instead of $\boldsymbol{\epsilon}$. Then sampling in the latent space can be formulated as $q(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, \boldsymbol{z}_0) = \mathcal{N}(\boldsymbol{z}_{t-1}; \tilde{\mu}_t(\boldsymbol{z}_t, \boldsymbol{z}_0), \tilde{\beta}_t\mathbf{I})$ where $\tilde{\mu}_t(\boldsymbol{z}_t, \boldsymbol{z}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{z}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{z}_t$, $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$, and $\bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$. Equivalently, we have

$$\boldsymbol{z}_{t-1} = \tilde{\mu}_t(\boldsymbol{z}_t, \boldsymbol{z}_0) + \sqrt{\tilde{\beta}_t}\boldsymbol{\epsilon}. \tag{9}$$

Then we can sample $\boldsymbol{z}_0$ from $\boldsymbol{z}_T$ by recursively applying

$$\boldsymbol{z}_{t-1} = \tilde{\mu}_t(\boldsymbol{z}_t, g_\theta(\boldsymbol{z}_t, t)) + \sqrt{\tilde{\beta}_t}\boldsymbol{\epsilon}. \tag{10}$$

3. Methodology

We here propose our LDM method tailored for conditional reservoir facies generation with maximal preservation of conditional data. We first describe the differences between image generation and reservoir facies generation that are important to be considered in the design of our method, and then outline the proposed method.

3.1. Differences between image generation and reservoir facies generation

There are several distinct differences between the image generation problem and the reservoir facies generation problem that pose challenges in employing an LDM for reservoir facies generation:

Input types. In image generation, an input image is considered continuous and one has $\boldsymbol{x} \in \mathbb{R}^{3\times H\times W}$ where 3, H, and W denote RGB channels, height, and width, respectively. In the reservoir facies generation, however, the input is categorical $\boldsymbol{x} \in \mathbb{Z}_2^{F\times H\times W}$ where $\mathbb{Z}_2 \in \{0, 1\}$, F denotes the number of facies types, and each pixel, denoted by $\boldsymbol{x}_{:hw} = \boldsymbol{x}_{fhw} \forall f = 1, 2, \ldots, F$, is a one-hot-encoded vector where 1 for a corresponding facies type index, 0 otherwise. The conditional data of $\boldsymbol{x}$, notated as $\boldsymbol{x}_c$ has a dimension of $(F + 1 \times H \times W)$. It has one more dimension than $\boldsymbol{x}$ for indicating a masked region.

Properties of conditional data. The domains of conditional data in LDM are often different from the target domain. It can for instance be a text prompt, which is among the most common conditional domains. In the conditional reservoir facies generation, unlike common applications of LDMs, the conditional domain corresponds to the target domain. Importantly, its conditional data $\boldsymbol{x}_c$ is spatially aligned with $\boldsymbol{x}$.

Strict requirement to preserve conditional data in generated sample. In an LDM, the conditioning process has cross-attention (Vaswani et al., 2017) between the intermediate representations of the U-Net and the representation of conditional data obtained with $\mathcal{E}_c$. One way of viewing this is that the encoded conditional data is mapped to the U-Net as

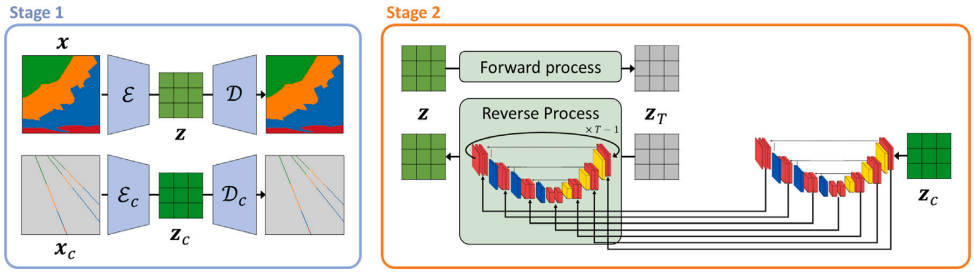


Fig. 6. Overview of the training process of our proposed method. It consists of two subsequent training stages: stage 1 for learning to compress and decompress data, and stage 2 for learning the conditional denoising process. Two U-Nets are used in stage 2. One is for the denoising process and the other is for extracting the intermediate representations of data latent vector z_c. After completing the training process, the generation of a new sample (sampling process) involves the denoising of variable latent vector z_T into $z_0 = z$, using Eq. (10), followed by its decoding into the data space, which is expressed as $D(z)$.

auxiliary information. However, an LDM has a caveat in the conditional generation — that is, its conditioning mechanism is not explicit but rather implicit. To be more specific, conditional data is provided to the denoising autoencoder, but the denoising process is not penalized for insufficiently honoring the conditional data. As a result, LDMs are often unable to fully preserve conditional data in the generated sample but rather only capture the context of conditional data. The limitation has been somewhat alleviated using classifier-free guidance (Ho and Salimans, 2021), but the problem still persists. Our problem with facies generation requires precise and strict preservation of conditional data in the generated data. In other words, facies measurements in wells should be retained in the predicted facies realization. Therefore, an explicit conditioning mechanism needs to be incorporated.

3.2. Proposed method

Our proposed method, tailored for conditional reservoir facies generation, is based on LDMs, leveraging its computational efficiency and resulting feasibility. The suggested method addresses several key aspects, including proper handling of the categorical input type, effective mapping of conditional data to the generative model, and maximal preservation of conditional data through a dedicated loss term for data preservation. Fig. 6 presents the overview of the training process of our proposed method.

Stage 1. has two pairs of encoder and decoder, trained to compress and decompress $\boldsymbol{x}$ and $\boldsymbol{x}_c$, respectively. The first pair is $\mathcal{E}$ and D for the unconditional part, and the second pair is $\mathcal{E}_c$ and D_c for the conditional part. Here, $\mathcal{E}$ compresses $\boldsymbol{x}$ to $\boldsymbol{z}$, while $\mathcal{E}_c$ compresses $\boldsymbol{x}_c$ to $\boldsymbol{z}_c$. Because $\boldsymbol{x}$ and $\boldsymbol{x}_c$ are spatially aligned, we use the same architectures for $\mathcal{E}$ and $\mathcal{E}_c$, and D and D_c. Furthermore, our input is categorical as $\boldsymbol{x} \in \mathbb{Z}_2^{F\times H\times W}$ and $\boldsymbol{x}_c \in \mathbb{Z}_2^{F+1\times H\times W}$. Therefore, we cannot naively use the stage 1 loss of LDM in Eq. (4). We tackle the limitation by reformulating the task as a classification task instead of a regression. Hence, our loss function in stage 1 is based on the cross-entropy loss function and it is formulated as:

$$L_{\text{stage1}} = \mathbb{E}_{\boldsymbol{x},h,w}\left[-\sum_f \boldsymbol{x}_{fhw}\log \text{softmax}(D(\mathcal{E}(\boldsymbol{x}))_{fhw}) - \sum_f (\boldsymbol{x}_c)_{fhw}\log \text{softmax}(D_c(\mathcal{E}_c(\boldsymbol{x}_c))_{fhw})\right] \tag{11a}$$

$$= \mathbb{E}_{\boldsymbol{x},h,w}\left[-\sum_f \boldsymbol{x}_{fhw}\log \tilde{\boldsymbol{x}}_{fhw} - \sum_f (\boldsymbol{x}_c)_{fhw}\log (\tilde{\boldsymbol{x}}_c)_{fhw}\right] \tag{11b}$$

$$= CE(\boldsymbol{x},\tilde{\boldsymbol{x}}) + CE(\boldsymbol{x}_c,\tilde{\boldsymbol{x}}_c) \tag{11c}$$

$$= L_{\text{recons}}(\boldsymbol{x},\mathcal{E},D) + L_{\text{recons}}(\boldsymbol{x}_c,\mathcal{E}_c,D_c), \tag{11d}$$

where CE denotes a cross-entropy loss function and L_{recons} denotes a reconstruction loss function.

Stage 2. is dedicated to learning prior and posterior distributions via learning the reverse denoising process. The learning process involves two important perspectives: (1) effective mapping of $\boldsymbol{z}_c$ to the denoising autoencoder g_θ to enable the conditional generation and (2) maximal preservation of conditional data in the generated data.

To achieve the effective mapping of $\boldsymbol{z}_c$ to g_θ, we employ two U-Nets with the same architecture to process $\boldsymbol{z}$ and $\boldsymbol{z}_c$, respectively. The first U-Net is the denoising autoencoder g_θ and the second U-Net is denoted g_ϕ for extracting multi-level intermediate representations of $\boldsymbol{z}_c$. In the conditional denoising process, the intermediate representations of $\boldsymbol{z}_c$ are mapped onto those of $\boldsymbol{z}_t$ obtained with g_θ. This multi-level mapping enables a more effective conveyance of $\boldsymbol{z}_c$ which in turn results in better preservation of conditional data in the generated facies realizations. The multi-level mapping is possible because $\boldsymbol{x}$ and $\boldsymbol{x}_c$ are spatially aligned, and equivalently for $\boldsymbol{z}$ and $\boldsymbol{z}_c$ with their intermediate representations from the U-Nets.

To achieve maximal preservation of conditional data, we explicitly tell the generative model to preserve $\boldsymbol{x}_c$ in the generated sample by introducing the following loss:

$$L_{\text{preserv}} = CE(\boldsymbol{x}_c,\hat{\boldsymbol{x}}_c), \tag{12}$$

where $\hat{\boldsymbol{x}}_c$ represents a softmax prediction of $\boldsymbol{x}_c$ and is a subset of $\hat{\boldsymbol{x}}$ in which $\hat{\boldsymbol{x}} = \text{softmax}(D(\hat{\boldsymbol{z}}))$ and $\hat{\boldsymbol{z}} \sim p_\theta(\boldsymbol{z}|\boldsymbol{z}_t, g_\phi(\boldsymbol{z}_c))$. Here, $p_\theta(\boldsymbol{z}|\boldsymbol{z}_t, g_\phi(\boldsymbol{z}_c))$ denotes the conditional probabilistic generative denoising process to sample $\hat{\boldsymbol{z}}$, given $\boldsymbol{z}_t$ and $g_\phi(\boldsymbol{z}_c)$.

Finally, our loss function in stage 2 is defined by

$$L_{\text{stage2}} = \left\{p_{\text{uncond}} L_{\text{LDM}}(\boldsymbol{z}, g_\theta) + (1-p_{\text{uncond}}) L_{\text{LDM,c}}(\boldsymbol{z}, g_\phi(\boldsymbol{z}_c), g_\theta)\right\} + L_{\text{preserv}}, \tag{13}$$

where p_{uncond} is a constant probability of unconditional generation, typically assigned a value of either 0.1 or 0.2 (Ho and Salimans, 2021).

4. Experiments

Our dataset comprises 5000 synthetic reservoir facies samples. The generating facies model is motivated by data from shoreface deposits in wave-dominated shallow-marine depositional environments. For details about the geological modeling assumptions about bedset stacking and facies sampling, see Appendix A. The data samples are partitioned into training (80%) and test datasets (20%). In our experiments, we assess the effectiveness of our proposed version of an LDM for both conditional and unconditional facies generation. Furthermore, we present a comprehensive comparative analysis of our diffusion model against a GAN-based approach. Specifically, we adopt the U-Net GAN from Zhang et al. (2021) due to its similar conditional setup to ours and because it has shown good performance in terms of fidelity and sample diversity in the conditional generation of binary facies. For the details of our diffusion model and U-Net GAN, see Appendices B and C, respectively.

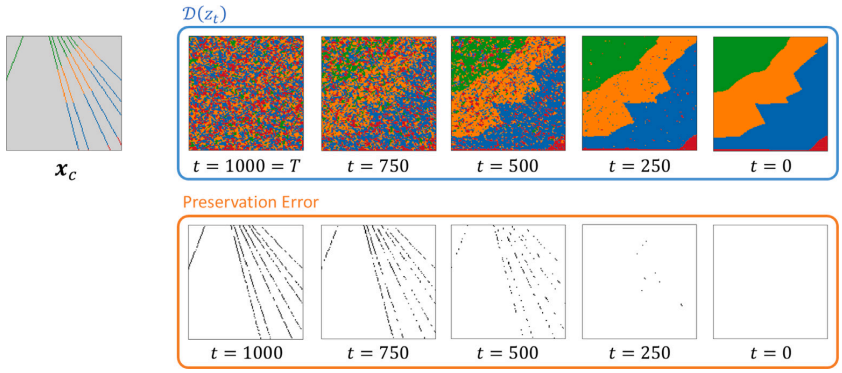


Fig. 7. Visualization of transitions in the conditional denoising process. The denoising autoencoder sequentially denoises latent variable vector z_T to z_0, conditioned on the encoded x_c, in the sampling process. Each z_t in the process can decoded and visualized to gain a better understanding of the conditional denoising process. We present $D(z_t)$ at the denoising steps of 1000, 750, 500, 250, and 0, where $T = 1000$ in this setup. The preservation error indicates the degree of accuracy with which the conditional data is retained within the generated data. Pixels colored in black indicate the error.

4.1. Conditional facies generation by the proposed diffusion model

In the sampling process, the denoising autoencoder iteratively performs denoising to transition z_T (Gaussian noise) into z_0 with each step being conditioned on the encoded conditional data. To provide a granular insight into the progressive denoising process, we present a visual example of transitions of the conditional denoising process in Fig. 7. (Additional examples are presented in Fig. 13 in Appendix E.) At the beginning of the denoising process ($t = 1000 = T$), z_t is initially composed of random Gaussian noises. Consequently, $D(z_t)$ also represents noise, resulting in a significant preservation error. However, as the denoising steps progress towards $t = 0$, the generated facies gradually become more distinct and recognizable while the preservation error becomes smaller.

In Fig. 7, x_c is sourced from the test dataset, and we visualize the most probable facies types within $D(z_t)$. It is important to emphasize that $D(z_t)$ belongs to the space $\mathbb{Z}_2^{F\times H\times W}$, where the most probable facies type corresponds to the channel f with the highest value. The results demonstrate the effectiveness of the denoising process of our method. We notice the gradual improvement in the fidelity of the generated facies and the preservation error, eventually producing realistic facies that honor the conditional data.

The denoising process is stochastic, therefore various facies can be generated given x_c. In Fig. 8, multiple instances of conditionally-generated facies are showcased for different x_c. Each row in this display hence represents multiple realizations of facies models, given the well facies data (second column of each row).

The results highlight the efficacy of our diffusion model in capturing the posterior and sample diversity while adhering to given constraints. In particular, the conditional generation can be notably challenging, especially when there is a substantial amount of conditional data to consider (*e.g.,* the third row in Fig. 8). However, our diffusion model demonstrates its capability to honor the conditional data while generating realistic facies faithfully. This capability facilitates the quantification of uncertainty associated with the generated facies, providing valuable insights for decision-makers in making informed decisions. With the bedset model, the well data contains much information about the transition zone from one facies type to another. This information clearly constrains the variability in the conditional samples, and there is not so much variability within the samples in a single row compared with the variability resulting from different well configurations and facies observations in the wells.

4.2. Conditional facies generation by GAN and its limitations

We next show results of using a GAN on this reservoir facies generation problem. As commonly observed in the GAN literature, we experienced a high level of instability in training with a U-Net GAN. Fig. 9 presents the training history of the U-Net GAN. First, the gap between the generator and discriminator losses becomes larger as the training progresses. This indicates that it suffers from the generator-discriminator imbalance problem. Second, the discriminator loss eventually converges, while the generator loss diverges towards the end of the training. This exhibits the divergent loss problem and results in a complete failure of the generator. Third, the loss for preserving x_c is unstable and non-convergent due to the unstable training process of the GAN. Lastly, the sample diversity loss indicates better diversity when the loss value is lower and vice versa. Throughout the training process, the diversity loss remains high until shortly before around 780 epochs, then the generator fails and starts producing random images. The failure leads to a decrease in the diversity loss. It indicates that the GAN model struggles to capture a sample diversity while retaining good generative performance.

Fig. 10 shows conditionally-generated samples using U-Net GAN at different training steps. The unstable training process can be seen in the generated samples. For instance, we observe a noticeable improvement in the quality of generated facies up to the 400 training epoch. However, from the 500 epoch, the quality continues to decline until the generated samples are barely recognizable. Generally, the GAN model appears to face challenges in concurrently maintaining high fidelity, preserving conditional data, and achieving sample diversity, therefore failing to capture the posterior.

Fig. 11 presents multiple instances of generated facies conditioned on different x_c using the U-Net GAN. The generator at the training epoch of 400 is used to generate the samples for its better performance than the generators at the other epochs. While showing more consistency than that of Fig. 10, the results still show that the generated samples have low fidelity, considerable deviations from the ground truths, and a lack of sample diversity due to the mode collapse. Furthermore, the generated samples exhibit a considerable sum of preservation errors, indicating the incapability to retain the conditional data. Overall, the results demonstrate that the GAN model fails to capture the posterior.

Table 1 specifies the preservation error rates of our proposed diffusion model and the U-Net GAN. The preservation error rate is defined as in Box I

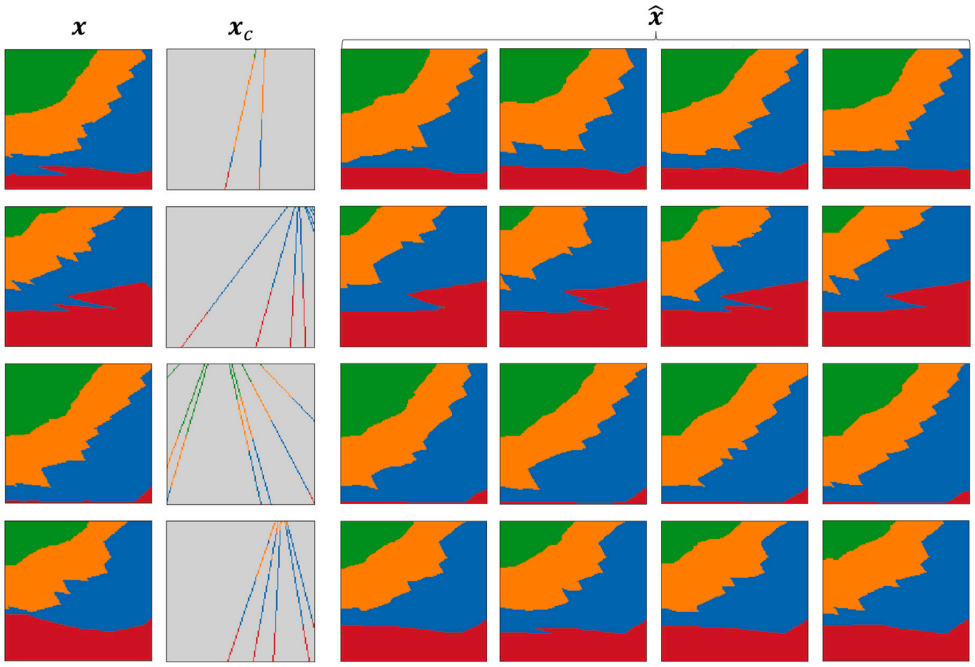


Fig. 8. Examples of multiple instances of generated facies conditioned on different conditional data x_c using our diffusion model. The first and second columns represent x (ground truth) and x_c, respectively, from the test dataset, and the remaining columns represent the conditionally generated facies. It is important to emphasize that the preservation error maps are omitted here because conditional sample $\hat{x}$ does not carry any preservation error here.

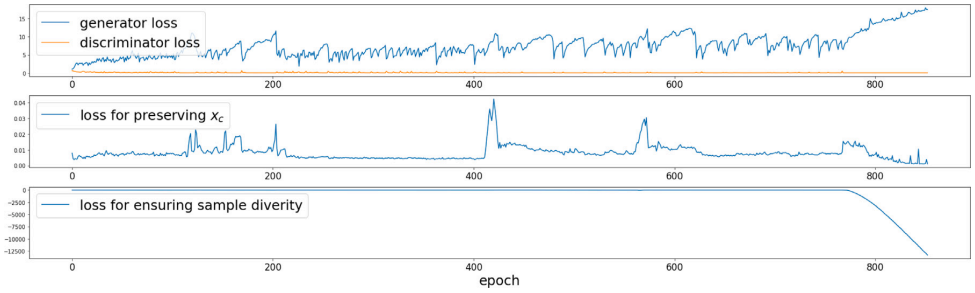


Fig. 9. Training loss history of U-Net GAN. The training process exhibits issues such as non-convergence, imbalance between the generator and discriminator, and divergent loss.

$$\text{Preservation error rate} = \frac{\text{number of different pixels between } x_c \text{ and argmax } \hat{x}_c \text{ for the valid pixels in } x_c}{\text{number of valid pixels in } x_c},$$

Box I.

where the preservation error rate of zero indicates perfect preservation. The results demonstrate that our diffusion model achieves the near-perfect preservation (*i.e.,* only 0.04% of conditional data fails to be retained) and it significantly outperforms the U-Net GAN in retaining conditional data, surpassing it by a factor of approximately 255 times.

To better illustrate the mode collapse phenomenon in U-Net GANs in comparison to our proposed diffusion model, Fig. 12 shows a comparison between the prior distributions of the training and test datasets along with the prior distribution predicted by our diffusion model and that of the U-Net GAN. To visualize the prior distributions of the training and test datasets, we first employ an argmax operation on x across the channel dimension. This operation results in argmax $x \in$

Table 1
Preservation error rates of our diffusion model and U-Net GAN on the test dataset.

	Our diffusion model	U-Net GAN
Preservation error rate	**0.0004**	0.1022

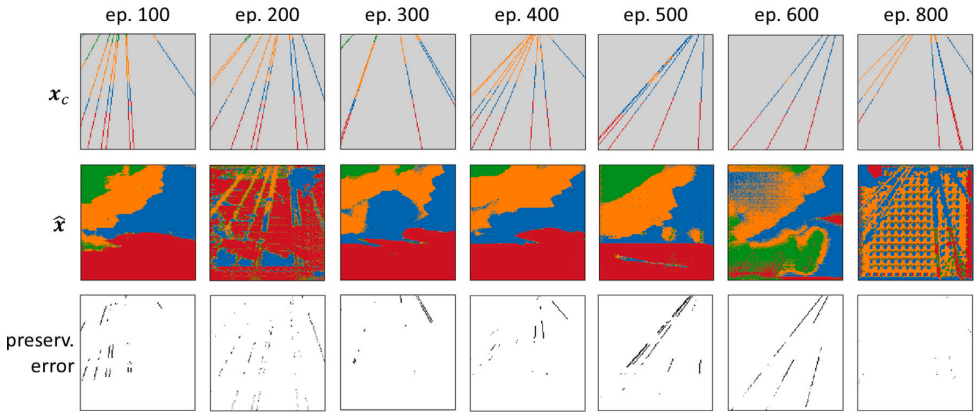


Fig. 10. Visualization of conditionally-generated facies samples by U-Net GAN at different training steps with the preservation error map. Here, ep. denotes epoch, and variable x and conditioning data x_c are from the test dataset.

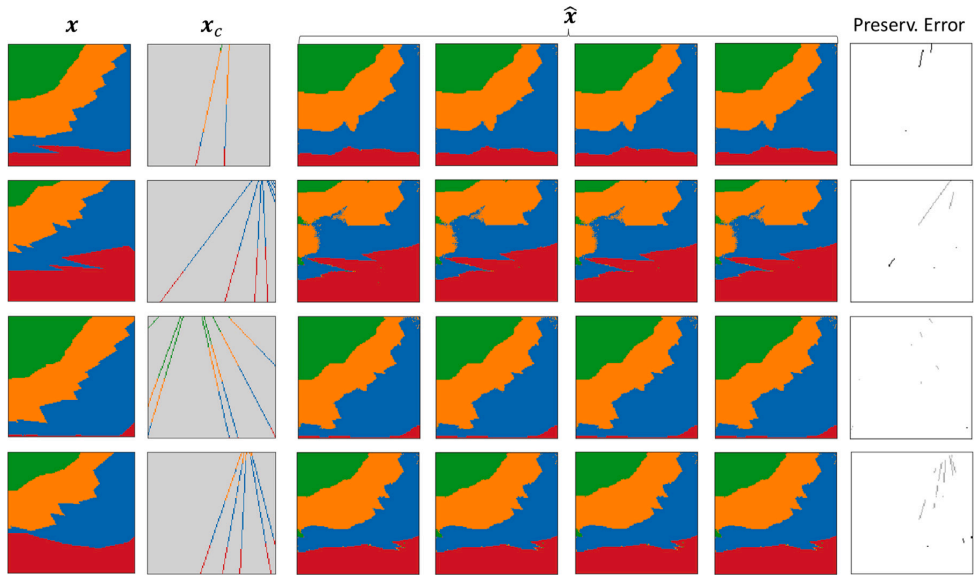


Fig. 11. Examples of multiple instances of generated facies conditioned on different data x_c using U-Net GAN The first and second columns represent x (ground truth) and x_c, respectively, from the test dataset, $\hat{x}$ denotes the conditionally generated facies, and the last column shows the preservation error maps. It is important to highlight that we are showcasing a total of four distinct generated samples. Nevertheless, they appear to be identical, primarily as a result of the mode collapse phenomenon that occurs during the GAN training. Because the generated facies are identical, their corresponding preservation error maps are also identical. Hence, we present a single preservation error map on the right-hand side.

$\mathbb{R}^{H\times W}$ that contains integer values. Subsequently, we calculate the average of argmax x for all instances of x within the training or test dataset. The same visualization procedure is applied to visualize the prior distributions predicted by our diffusion model and U-Net GAN, with the only difference being the application of an argmax operation to generated facies data. For the U-Net GAN, its generator at the 400 training epochs is used (same as above). The prior distribution contains four main distinct colors (green, orange, blue, red) depending on facies types, and darker colors indicate high likelihood and vice versa. These results clearly demonstrate our diffusion model's capability to accurately capture the prior distribution, whereas the U-Net GAN faces substantial challenges in this regard due to the mode collapse, leading to severe underestimation of the variability in the generated samples.

In Fig. 12 we also show the Jensen–Shannon (JS) divergence of the generated samples compared with the true model. Similar to the Kullback–Leibler divergence, but non-symmetric and finite (between 0 and 1), the JS divergence here measures the probabilistic difference between the generated and training samples. Clearly, the divergence is much smaller for the LDM here, while the GAN gets very large JS divergence (close to 1) at some of the facies boundaries. Even though it is less prominent than for the GAN, the divergence for LDM shows

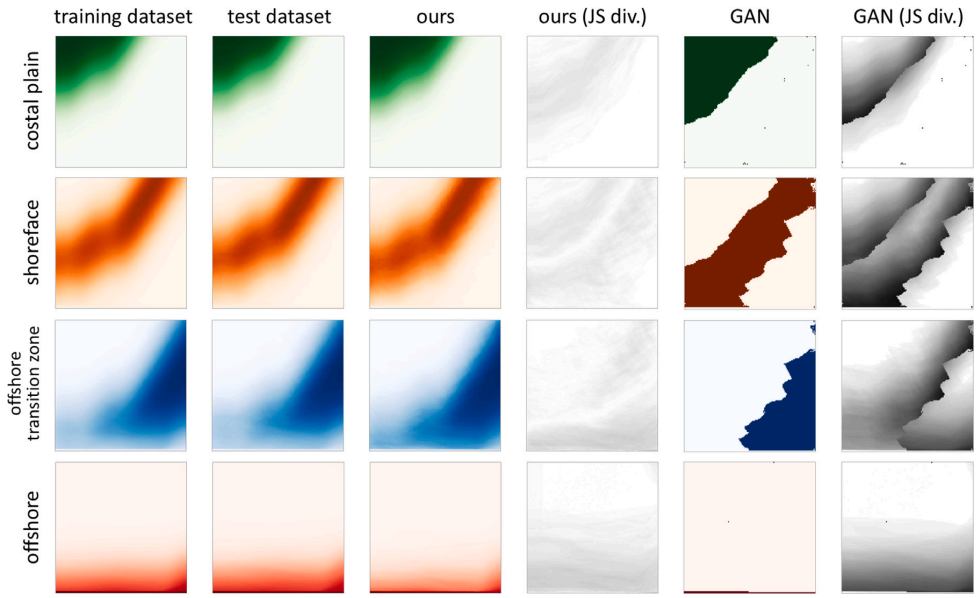


Fig. 12. Comparative visualization of the prior distributions of the training and test datasets and the prior distribution predicted by our diffusion model and the U-Net GAN. The binary-colored figure depicts the discrepancy, measured by JS divergence, between the prior distribution of the training dataset and the predicted prior distribution. At every pixel, there exists a prior distribution encompassing various facies types, where these facies types are visually represented by distinct colors. The presence of intermediate colors signifies a prior distribution with greater diversity. In this context, it is important to highlight that the mode collapse observed in the U-Net GAN results in a reduction of sample diversity, analogous to the absence of those intermediate colors in the visualization.

some structure near the facies transition zones. This indicates some underestimation in the implicit posterior sampling variability.

4.3. Ablation study

We conduct an ablation study to investigate the effects of the use of the proposed components such as L_{preserv} and the multi-level mapping of $\boldsymbol{z}_c$. The evaluation is performed on the test dataset. Table 2 outlines the specific Case (a)–(c) considered in the ablation study, and Table 3 reports $p_{\text{uncond}} L_{\text{LDM}} + (1 - p_{\text{uncond}}) L_{\text{LDM,c}}$ from Eqs. (7)–(8), L_{preserv}, and a preservation error rate on the test set.

When analyzing these results, several key findings emerge. Firstly, in Case (b), both L_{preserv} and the preservation error rate exhibit a significant increase compared to Case (a). This increase can be attributed to the fact that the denoising model in (b) was not explicitly trained to preserve the conditional data, resulting in a notable degradation in preservation quality. Case (c) sheds light on the effectiveness of the multi-level mapping, comparing L_{preserv} and the preservation error rate from the baseline Case (a). It emphasizes the positive impact of the multi-level mapping approach on preservation. Overall, the ablation study reveals that each component in our methodology plays a vital role in enhancing the overall preservation capacity of the conditional sampling.

Table 2
Ablation study cases with respect to the novel and essential components in stage 2. The signs of o and x indicate the use of the item described in the corresponding column name, where o and x denote using and not using, respectively. In the case of (c), instead of employing multi-level mapping for $\boldsymbol{z}_c$, it takes a straightforward route to integrate $\boldsymbol{x}_c$ into the denoising U-Net. This integration is achieved through a simple concatenation of $\boldsymbol{z}_c$ and $\boldsymbol{z}_t$, forming the input denoted as $[\boldsymbol{z}_t, \boldsymbol{z}_c]$ for the denoising U-Net, where [.] represents the concatenation operation.

	L_{preserv}	Multi-level mapping of $\boldsymbol{z}_c$
(a) Base	o	o
(b) − L_{preserv}	x	o
(c) − Multi-level mapping of $\boldsymbol{z}_c$	o	x

5. Conclusion

We have introduced a novel approach for conditional reservoir facies modeling employing LDM. Experimental results show exceptional abilities to preserve conditional data within generated samples while producing high-fidelity samples. Our novelties lie in the proposals to enhance the preservation capabilities of LDM. Throughout our experiments, we have demonstrated the robustness and superiority of our diffusion-based method when compared to a GAN-based approach, across multiple aspects including fidelity, sample diversity, and conditional data preservation. Furthermore, we have presented the critical limitations of the GAN approach, which result in compromised fidelity, limited sample diversity, and sub-optimal preservation performance.

Overall, our work opens up a new avenue for conditional facies modeling through the utilization of a diffusion model. The results indicate some underestimation in the posterior samples, which can possibly improved by more nuanced training or refined loss functions. As future work, we aim to study the statistical properties of the LDM in detail on various geostatistical models, study conditioning to other data types, and extend our method for 3D facies modeling.

Conditioning to seismic data can be done in various ways. For instance by introducing a seismic loss function (possibly including convolution effects in the seismic forward model) to the training loss and the reconstruction loss. Expanding our method for 3D facies modeling may appear straightforward by merely substituting 2D convolutional layers with their 3D counterparts. However, dealing with 3D spatial data presents inherent complexities stemming from its high-dimensional nature. This can manifest in various challenges, including

Table 3
Effects of the use of L_{preserv} and multi-level mapping of $\boldsymbol{z}_c$ for the ablation study cases.

	(a) Base	(b) – L_{preserv}	(c) – Multi-level mapping of $\boldsymbol{z}_c$
$p_{\text{uncond}} L_{\text{LDM}} + (1 - p_{\text{uncond}}) L_{\text{LDM,c}}$	0.02444	0.02306	0.02618
L_{preserv}	**0.00029**	0.00643	0.00082
Preservation error rate	**0.00044**	0.00442	0.00146

high computational demands and the difficult learning of prior and posterior distributions. Therefore, it may need to employ techniques like hierarchical modeling. This can involve employing a compact latent dimension size for sampling, followed by an upscaling mechanism similar to super-resolution, in order to enhance the feasibility and effectiveness of the 3D modeling. Moreover, incorporating additional conditional data, such as seismic information, into the sampling process can be achieved through the use of cross-attention mechanisms, as introduced in the original LDM.

CRediT authorship contribution statement

Daesoo Lee: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Oscar Ovanger:** Writing – review & editing, Writing – original draft, Validation, Resources, Methodology, Formal analysis, Conceptualization. **Jo Eidsvik:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Resources, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Erlend Aune:** Writing – review & editing, Writing – original draft, Validation, Supervision, Resources, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Jacob Skauvold:** Writing – review & editing, Resources, Methodology, Conceptualization. **Ragnar Hauge:** Writing – review & editing, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to thank the Norwegian Research Council for funding the Machine Learning for Irregular Time Series (ML4ITS) project (312062), the GEOPARD project (319951) and the SFI Centre for Geophysical Forecasting (309960).

Code availability section

The source code is available on github.com/ML4ITS/Latent-Diffusion-Model-for-Conditional-Reservoir-Facies-Generation

Contact: daesoo.lee@ntnu.no

Hardware requirements: a sufficient GPU device for training a deep learning model with the PyTorch library. We used a single NVIDIA GeForce RTX 3060.

Program language: Python

Appendix A. Dataset

Our test images are vertical 2D slices through a shoreface deposit in a wave-dominated shallow-marine depositional environment. The cross section is taken along the dip direction, with the proximal or landward side to the left in the image, and the distal or seaward side to the right. The shoreface deposit consists of sediment packages referred to as bedsets. On the landward side of the bedsets is the coastal plain, which is coaly and of poor reservoir quality. The proximal part of each bedset consists of shoreface sand of good reservoir quality, while the distal part has sand interbedded with shale. The offshore region beyond the distal edge of the bedsets has only shale.

We create realizations using the rule-based object model GEOPARD, which was described by Scotti et al. (2022). This model sequentially stacks bedsets in a way that mimics the depositional process. Bedset-scale description is appropriate for reservoir facies modeling (Isla et al., 2018). The trajectory of the shoreline is a function of bedset progradation and aggradation, in other words how much the bedsets build out and build up. These, in turn, are controlled by such environmental factors as the sea level and sediment supply. See also the article by Ovanger et al. (2024), where a conceptually similar shoreface deposition model is considered. GEOPARD first generates base and top surfaces for a sequence of bedsets, and then uses these surfaces to create a 3D grid of facies values. The data in this study consists of 2D arrays extracted from these 3D grids. We take one slice from each 3D grid. That is, we do not take multiple slices from the same realization. Facies is treated as a categorical variable and one-hot encoded, as described in Section 3.1.

The conditional data $\boldsymbol{x}_c$ are generated by taking a subset of $\boldsymbol{x}$ with a random number of straight lines and random angles within certain ranges. The line patterns resemble groups of deviated wells with a common template, in other words originating from a common point somewhere above the image. The number of lines is sampled from a Poisson distribution with expectation four and then shifted up by one so that the expected number of lines is five, and there is always at least one line. Our dataset comprises 5000 facies realizations, split into training (80%) and test datasets (20%). The full dataset is available at https://figshare.com/articles/dataset/Dataset_used_in_Latent_Diffusion_Model_for_Conditional_Reservoir_Facies_Generation/26892868?file=48931588.

Appendix B. Implementation details of our proposed method

B.1. Encoders and decoders: $\mathcal{E}$, $\mathcal{E}_c$, $\mathcal{D}$, and $\mathcal{D}_c$

The same encoder and decoder architectures from the VQ-VAE paper are used and their implementations are from https://github.com/nadavbh12/VQ-VAE. The encoder is a stack of a downsampling convolution block (Conv2d – BatchNorm2d – GELU – Dropout) and a subsequent residual block (GELU – Conv2d – BatchNorm2d – GELU – Dropout – Conv2d). The short notations are taken from the PyTorch implementations. The architecture of the decoder is the inverse of the encoder's. Its upsampling convolutional layer is implemented with (Upsample(mode = 'nearest') – Conv2d). In the encoding process, the spatial size halves and the hidden dimension doubles after every downsampling block, and the bottleneck dimension (*i.e.,* dimension of $\boldsymbol{z}$ and $\boldsymbol{z}_c$) is set to 4, following Rombach et al. (2022).

The stack size determines a downsampling rate. For instance, a single stack corresponds to a downsampling rate of 2. In our experiments, we use a single stack because we observed that a higher downsampling rate leads to higher loss of input information, resulting in an inadequate reconstruction of $\boldsymbol{x}_c$. This inadequacy suggests that $\boldsymbol{z}_c$ fails to fully capture the information contained in $\boldsymbol{x}_c$, ultimately leading to a deficiency in preserving $\boldsymbol{x}_c$ within a generated sample. In addition, Rombach et al. (2022) demonstrated that a low compression rate is sufficient for LDM to generate high-fidelity samples.

In the naive form of $\mathcal{E}$ and $\mathcal{E}_c$, the value ranges of $\boldsymbol{z}$ and $\boldsymbol{z}_c$ are not constrained. However, the diffusion model g_θ is typically designed to

receive a value ranging between −1 and 1. For instance, image data is scaled to range between −1 and 1 to be used as the input. To make $\boldsymbol{z}$ and $\boldsymbol{z}_c$ compatible with the diffusion model, we normalize them as $\boldsymbol{z}/\max(|\boldsymbol{z}|)$ and $\boldsymbol{z}_c/\max(|\boldsymbol{z}_c|)$, respectively.

B.2. U-Net

Two U-Nets are used in our proposed method — one for g_θ and the other for processing $\boldsymbol{z}_c$. The two U-Nets have the same architecture to have the same spatial dimensions for the multi-level mapping. We use the implementation of U-Net from here.[1] Its default parameter settings are used in our experiments except for the input channel size and hidden dimension size. To be more precise, we use `in_channels` (input channel size) of 4 because it is the dimension sizes of $\boldsymbol{z}$ and $\boldsymbol{z}_c$ and `dim` (hidden dimension size) of 64.

B.3. Latent diffusion model

LDM is basically a combination of the encoders, decoders, and DDPM, where DDPM is present in the latent space. We use the implementation of DDPM from here.[2] Its default parameter settings are used in our experiments except for the input size and denoising objective for which we use the prediction of $\boldsymbol{z}_0$ instead of ϵ, as described in Section 2.4.

B.4. Optimizer

We employ the Adam optimizer (Kingma and Ba, 2015). We configure batch sizes of 64 and 16 for stage 1 and stage 2, respectively. The training periods are 100 epochs for stage 1 and 20 000 steps for stage 2.

B.5. Unconditional sampling

The conditional sampling is straightforward as illustrated in Fig. 6. For the unconditional sampling, we replace $\boldsymbol{z}_c$ with mask tokens, typically denoted as `[MASK]` or `[M]` (Lee et al., 2023, 2024). The role of the mask token is to indicate that the sampling process is unconditional. The mask token is a learnable vector trained in stage 2 by minimizing $L_{\text{LDM}}(\boldsymbol{z}, g_\theta)$ in L_{stage2}.

Appendix C. Implementation details of U-Net GAN

We implement U-Net GAN, following the approach outlined in its original paper (Zhang et al., 2021). Two key hyperparameters govern the weighting of loss terms in this implementation: one for preserving conditional data (content loss) and the other for ensuring sample diversity (diverse loss). We maintain the same weights as specified in the paper, with a value of 0.05 for the diverse loss and 100 for the content loss. For optimization, we employ the Adam optimizer with a batch size of 32, a maximum of 750 epochs, and a learning rate set to 0.0002. The implementation is included in our GitHub repository.

To enhance the training of GANs, several techniques like feature matching, historical averaging, and one-sided label smoothing have been proposed. These methods, detailed in Salimans et al. (2016), are primarily aimed at stabilizing the training process. However, in our approach, we have chosen not to implement these techniques. Instead, we focus on maintaining the fundamental structure of the GAN model to assess its performance in a basic form.

[1] https://github.com/lucidrains/denoising-diffusion-pytorch.

[2] See footnote 1.

Appendix D. Pseudocode

To increase the reproducibility of our work and understanding of our codes in our GitHub repository, we present a pseudocode of the training process of our method in Algorithm 1 for stage 1 and Algorithm 2 for stage 2. In the pseudocodes, we provide a more detailed specification of D.

Algorithm 1 Pseudocode of the training process of the proposed diffusion model (stage 1)

while a maximum epoch is not reached **do**
  sample $\boldsymbol{x}$ from $\boldsymbol{X}$ ▷ $\boldsymbol{X}$ denotes a training dataset. In practice, a batch of $\boldsymbol{x}$ is sampled.
  $\boldsymbol{x}_c \leftarrow$ stochastically extracting conditional well data from $\boldsymbol{x}$

  $\boldsymbol{z}, \boldsymbol{z}_c \leftarrow \mathcal{E}(\boldsymbol{x}), \mathcal{E}_c(\boldsymbol{x}_c)$
  $\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_c \leftarrow \text{softmax}(D(\boldsymbol{z})), \text{softmax}(D_c(\boldsymbol{z}_c))$

  $L_{\text{stage1}} \leftarrow CE(\boldsymbol{x}, \tilde{\boldsymbol{x}}) + CE(\boldsymbol{x}_c, \tilde{\boldsymbol{x}}_c)$

  update $\mathcal{E}$, $\mathcal{E}_c$, D, and D_c by minimizing L_{stage1}
end while

Algorithm 2 Pseudocode of the training process of the proposed diffusion model (stage 2)

load the pretrained $\mathcal{E}$, $\mathcal{E}_c$, D, and D_c and freeze them.
randomly initialize g_θ and $g_{\boldsymbol{\phi}}$
while a maximum epoch is not reached **do**
  sample $\boldsymbol{x}$ from $\boldsymbol{X}$
  $\boldsymbol{x}_c \leftarrow$ stochastically extracting conditional well data from $\boldsymbol{x}$

  $\boldsymbol{z}, \boldsymbol{z}_c \leftarrow \mathcal{E}(\boldsymbol{x}), \mathcal{E}_c(\boldsymbol{x}_c)$ ▷ $\boldsymbol{z} = \boldsymbol{z}_0$
  $\boldsymbol{z}_t \leftarrow$ forward diffusion process applied to $\boldsymbol{z}_0$ ▷ adding noise to $\boldsymbol{z}_0$
  if $r \le p_{\text{uncond}}$ **then** ▷ $r \sim U(0,1)$ where U denotes a uniform distribution
    $\hat{\boldsymbol{z}}_0 \leftarrow g_\theta(\boldsymbol{z}_t, t)$ ▷ unconditional generation
    $L_{\text{LDM}} \leftarrow \|\boldsymbol{z}_0 - \hat{\boldsymbol{z}}_0\|_2^2$
    $\ell_{\text{LDM}} \leftarrow L_{\text{LDM}}$
  else
    $\hat{\boldsymbol{z}}_0 \leftarrow g_\theta(\boldsymbol{z}_t, t, g_{\boldsymbol{\phi}}(\boldsymbol{z}_c))$ ▷ conditional generation
    $L_{\text{LDM},c} \leftarrow \|\boldsymbol{z}_0 - \hat{\boldsymbol{z}}_0\|_2^2$
    $\ell_{\text{LDM}} \leftarrow L_{\text{LDM},c}$
  end if

  $\hat{\boldsymbol{x}} = \text{softmax}(D(\hat{\boldsymbol{z}}))$ ▷ $\hat{\boldsymbol{z}} = \hat{\boldsymbol{z}}_0$
  $\hat{\boldsymbol{x}}_c \leftarrow$ retrieving the valid pixel locations in $\boldsymbol{x}_c$ from $\hat{\boldsymbol{x}}$
  $L_{\text{preserv}} \leftarrow CE(\boldsymbol{x}_c, \hat{\boldsymbol{x}}_c)$

  $L_{\text{stage2}} \leftarrow \ell_{\text{LDM}} + L_{\text{preserv}}$

  update g_θ and $g_{\boldsymbol{\phi}}$ by minimizing L_{stage2}
end while

Appendix E. Additional experimental results

In continuation of Fig. 7, additional examples of the transitions in the conditional denoising process are presented in Fig. 13.

Data availability

The dataset is publicly available on https://figshare.com/articles/dataset/Dataset_used_in_Latent_Diffusion_Model_for_Conditional_Reservoir_Facies_Generation/26892868?file=48931588.

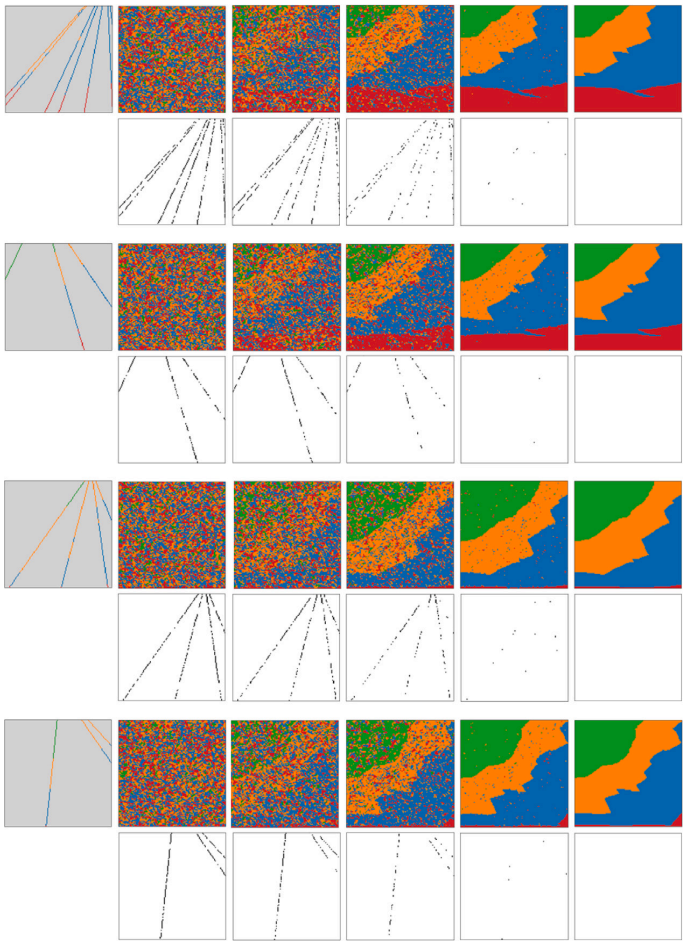


Fig. 13. (Continuation of Fig. 7) Additional examples of the transitions in the conditional denoising process.

References

Azevedo, L., Paneiro, G., Santos, A., Soares, A., 2020. Generative adversarial network as a stochastic subsurface model reconstruction. Comput. Geosci. 24 (4), 1673–1692.

Baranchuk, D., Voynov, A., Rubachev, I., Khrulkov, V., Babenko, A., 2022. Label-efficient semantic segmentation with diffusion models. In: International Conference on Learning Representations.

Cai, S., Wu, Y., Chen, G., 2022. A novel elastomeric UNet for medical image segmentation. Front. Aging Neurosci. 14, 841297.

Chan, S., Elsheikh, A.H., 2019. Parametric generation of conditional geological realizations using generative neural networks. Comput. Geosci. 23, 925–952.

Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T., 2022. Maskgit: Masked generative image transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11315–11325.

Cressie, N., 2015. Statistics for Spatial Data. John Wiley & Sons.

Dhariwal, P., Nichol, A., 2021. Diffusion models beat gans on image synthesis. Adv. Neural Inf. Process. Syst. 34, 8780–8794.

Dupont, E., Zhang, T., Tilke, P., Liang, L., Bailey, W., 2018. Generating realistic geology conditioned on physical measurements with generative adversarial networks. arXiv preprint arXiv:1802.03065.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2020. Generative adversarial networks. Commun. ACM 63 (11), 139–144.

Ho, J., Jain, A., Abbeel, P., 2020. Denoising diffusion probabilistic models. Adv. Neural Inf. Process. Syst. 33, 6840–6851.

Ho, J., Saharia, C., Chan, W., Fleet, D.J., Norouzi, M., Salimans, T., 2022. Cascaded diffusion models for high fidelity image generation. J. Mach. Learn. Res. 23 (1), 2249–2281.

Ho, J., Salimans, T., 2021. Classifier-free diffusion guidance. In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications.

Hu, F., Wu, C., Shang, J., Yan, Y., Wang, L., Zhang, H., 2023. Multi-condition controlled sedimentary facies modeling based on generative adversarial network. Comput. Geosci. 171, 105290.

Isla, M.F., Schwarz, E., Veiga, G.D., 2018. Bedset characterization within a wave-dominated shallow-marine succession: An evolutionary model related to sediment imbalances. Sediment. Geol. (ISSN: 0037-0738) 374, 36–52. http://dx.doi.org/10.1016/j.sedgeo.2018.07.003.

Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1125–1134.

Kim, G., Kwon, T., Ye, J.C., 2022. Diffusionclip: Text-guided diffusion models for robust image manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2426–2435.

Kingma, D., Ba, J., 2015. Adam: A method for stochastic optimization. In: International Conference on Learning Representations. ICLR, San Diega, CA, USA.

Lee, D., Malacarne, S., Aune, E., 2023. Vector quantized time series generation with a bidirectional prior model. In: International Conference on Artificial Intelligence and Statistics. PMLR, pp. 7665–7693.

Lee, D., Malacarne, S., Aune, E., 2024. Explainable time series anomaly detection using masked latent generative modeling. Pattern Recognit. 110826.

Mirza, M., Osindero, S., 2014. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

Ovanger, O., Eidsvik, J., Skauvold, J., Hauge, R., Aarnes, I., 2024. Addressing configuration uncertainty in well conditioning for a rule-based model. Math. Geosci. 1–26.

Pan, W., Torres-Verdín, C., Pyrcz, M.J., 2021. Stochastic Pix2pix: a new machine learning method for geophysical and well conditioning of rule-based channel reservoir models. Natl. Resour. Res. 30, 1319–1345.

Pyrcz, M.J., Deutsch, C.V., 2014. Geostatistical Reservoir Modeling. Oxford University Press, USA.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M., 2022. Hierarchical text-conditional image generation with clip latents. p. 3, arXiv preprint arXiv:2204.06125, 1.

Razak, S.M., Jafarpour, B., 2022. Conditioning generative adversarial networks on nonlinear data for subsurface flow model calibration and uncertainty quantification. Comput. Geosci. 26 (1), 29–52.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B., 2022. High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10684–10695.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer, pp. 234–241.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., 2016. Improved techniques for training gans. Adv. Neural Inf. Process. Syst. 29.

Scotti, A.A., Eide, C.H., Aarnes, I., Skauvold, J., Hauge, R., 2022. Defining the basic rules that describe long-term shoreface dynamics: A process-mimicking approach for reservoir modelling. In: EGU General Assembly, Vienna, Austria, 23–27 May 2022, EGU22-9355. European Geosciences Union, http://dx.doi.org/10.5194/egusphere-egu22-9355.

Song, S., Mukerji, T., Hou, J., 2021. GANSim: Conditional facies simulation using an improved progressive growing of generative adversarial networks (GANs). Math. Geosci. 1–32.

Van Den Oord, A., Vinyals, O., et al., 2017. Neural discrete representation learning. Adv. Neural Inf. Process. Syst. 30.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Yang, Z., Chen, Q., Cui, Z., Liu, G., Dong, S., Tian, Y., 2022. Automatic reconstruction method of 3D geological models based on deep convolutional generative adversarial networks. Comput. Geosci. 26 (5), 1135–1150.

Zhang, C., Song, X., Azevedo, L., 2021. U-net generative adversarial network for subsurface facies modeling. Comput. Geosci. 25, 553–573.

Zhang, T.-F., Tilke, P., Dupont, E., Zhu, L.-C., Liang, L., Bailey, W., 2019. Generating geologically realistic 3D reservoir facies models using deep learning of sedimentary architecture with generative adversarial networks. Pet. Sci. 16, 541–549.

A Statistical Study of Latent Diffusion Models for Geological Facies Modeling

Oscar Ovanger, Daesoo Lee, Jo Eidsvik, Ragnar Hauge, Jacob Skauvold and Erlend Aune

Mathematical Geosciences Special Issue (2025)

Mathematical Geosciences
https://doi.org/10.1007/s11004-025-10178-5

SPECIAL ISSUE

A Statistical Study of Latent Diffusion Models for Geological Facies Modeling

Oscar Ovanger[1] · Daesoo Lee[1] · Jo Eidsvik[1] · Ragnar Hauge[2] · Jacob Skauvold[2] · Erlend Aune[1,3]

Received: 20 February 2024 / Accepted: 25 January 2025

Abstract
There has been much interest recently in implicit artificial intelligence (AI)-based approaches for geostatistical facies modeling. New generative machine learning constructions such as latent diffusion models (LDMs) appear to be competitive with traditional geostatistical approaches for facies characterization. Going beyond visual inspection of predictions, this study examines properties of the statistical distribution of samples generated by an LDM trained to generate facies models. The study uses a traditional truncated Gaussian random field (TGRF) model as a reference data-generating process and as the ground truth for benchmarking the LDM results. The distributions of realizations drawn from the LDM and TGRF models are compared using metrics including bias, variance, higher-order statistics, transiograms and Jensen–Shannon divergence for both marginal and joint (volume) distributions. Comparisons are made with and without conditioning on facies observations in wells for both stationary and nonstationary TGRF models with different covariance functions. The observed distributional differences are modest, and LDMs are regarded as a very promising approach here. Even so, some systematic artifacts are observed, such as underrepresentation of variability by the LDM. Moreover, the performance of the LDM is found to be sensitive to the training data.

Keywords Latent diffusion models · Facies modeling · Truncated Gaussian random fields · Statistical evaluation of generative model

1 Introduction

Crucial decisions in the oil and gas industry often rely on multiple realizations of reservoir models. The toolkit for generating such realizations is still growing. It

Oscar Ovanger
oscar.ovanger@ntnu.no

1 Norwegian University of Science and Technology, Trondheim, Norway

2 Norwegian Computing Center, Oslo, Norway

3 BI, Norwegian Business School, Trondheim, Norway

Published online: 24 February 2025

encompasses traditional geostatistical modeling approaches such as truncated Gaussian random fields (TGRF) (Matheron et al. 1987; Mannseth 2014), and object-based models (Haldorsen and Damsleth 1990; Manzocchi and Walsh 2023), training image-based models such as multiple-point statistics (Strebelle 2002), process-based models (Lopez et al. 2009), and rule-based models (Pyrcz et al. 2015). Useful overviews can be found in the books by Caers (2005) and Pyrcz and Deutsch (2014, chapter 4). As different as these approaches to reservoir modeling are, they all balance geological realism with the ability to condition realizations to well data.

As machine learning research progresses and computing resources become increasingly available, new research prospects are emerging for geostatistical simulation techniques. The application of generative adversarial networks (GANs) (Goodfellow et al. 2020) to geomodeling problems has produced much research interest (Chan and Elsheikh 2019; Zhang et al. 2019), but GANs can be difficult to train due to issues such as mode collapse and instability in training (Salimans et al. 2016; Lucic et al. 2018). In recent years, denoising diffusion probabilistic models (DDPMs) (Ho et al. 2020) have excelled in various applications, including image generation (Lugmayr et al. 2022), time series generation and forecasting (Kollovieh et al. 2023), and audio generation (Liu et al. 2023).

This work conducts a thorough study of the statistical properties of the recent work of Lee et al. (2024) on the generation of facies realizations in a reservoir modeling context using a latent diffusion model (LDM) (Rombach et al. 2022). An LDM is a version of a DDPM where the denoising process happens in a learned latent space that efficiently compresses the data. The results obtained by Lee et al. (2024) were encouraging in that (i) the generated LDM realizations looked indistinguishable from training data, (ii) conditional realizations could be made to match hard data of facies in wells almost perfectly, without training the LDM on specific observation locations, and (iii) the quality of the facies realizations produced by the LDM outperformed an existing conditional GAN model (of comparable computational requirements) by a significant margin.

The main contribution of this article is in examining the distributional properties of realizations obtained by LDMs. The idea is to compare statistics of LDM realizations with the distributional properties of the traditional geostatistical TGRF model. It is straightforward to draw ample training data using TGRF. Moreover, one can sample exact conditional realizations with the TGRF model, and it is hence possible to compare the implicit results of LDM with those of the true explicit TGRF solution.

In this paper, the LDM and TGRF samples are compared for two separate cases: The first case is inspired by shoreface geometry, and the TGRF has a trend that imposes large-scale structures. The second case resembles a laterally heterogeneous structure with a stationary TGRF. In each case, a set of realizations is drawn from the TGRF and used to train the LDM. A different set of realizations is then drawn from the trained LDM and compared to a fresh set drawn from the TGRF. The LDM-TGRF comparisons are made both with and without conditioning to facies observations in wells.

Several metrics are used to compare sets of TGRF and LDM realizations. For reliable detection of distributional differences, metrics based on first-, second-, and higher-order statistics are used. These comparative metrics are used to gain insight into the characteristics, benefits and limitations of black-box generative AI models (LDM

in this case) by evaluating them against established models with known sampling behavior (TGRF in this case). If a model's properties cannot be defended in the context of a simple synthetic case, it would be unwise to trust it in real-world scenarios.

The remained of this article is organized as follows. Section 2 presents the LDM, Sect. 3 outlines the TGRF reference models, and Sect. 4 describes the metrics used to compare sets of realizations. Section 5 outlines the first case study, inspired by a shallow marine shoreface environment where large-scale structure is present, and the TGRF model is nonstationary. Section 6 follows with the second case study set in a laterally heterogeneous environment with a stationary TGRF model. Section 7 concludes the article by recapitulating the main findings.

2 Latent Diffusion Model

LDMs represent a significant advance in generative models. Like DDPMs, LDMs are capable of producing high-quality output of various types such as images, audio, and text. LDMs differ from DDPMs in that rather than working directly with the input data, they work in a latent space. This means that LDMs compress data into a more compact representation before initiating the diffusion process, effectively reducing computational complexity, meaning that output can be generated more efficiently.

This section provides a brief description of the LDM studied in this article. Readers should consult Rombach et al. (2022) for details about LDMs in general and Lee et al. (2024) for specifics about the facies-generating LDM studied here.

2.1 Training and Using the LDM

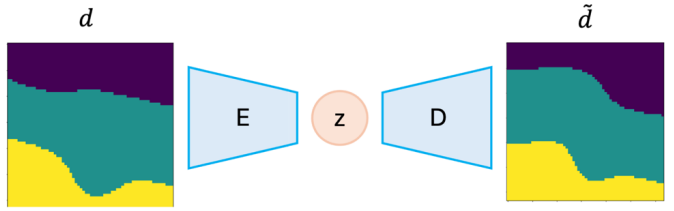
The LDM considered in this article generates facies realizations which are represented as images. LDMs are trained and used in three distinct stages that are summarized in what follows and in Fig. 1.

2.1.1 Training the Autoencoder (Stage 1 Training)

The autoencoder comprises an encoder network, E, and a decoder network, D. The network parameters in the decoder and encoder are denoted by θ. The encoder compresses a high-resolution image, d, into a lower-dimensional latent representation, $z = E(d)$, retaining salient features. The decoder attempts to reconstruct the input image d from the latent representation z, resulting in the reconstruction $\tilde{d} = D(z)$, found by minimizing a reconstruction loss function that has been specified in advance. The mean square difference between $\tilde{d}$ and d is a reasonable choice of reconstruction loss function. This training stage establishes the latent space where the denoising diffusion process will take place.

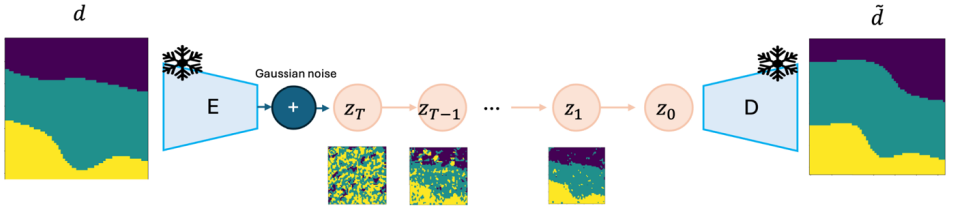
2.1.2 Training the Denoising Model (Stage 2 Training)

Freezing the encoder and decoder from stage 1, the denoising diffusion part of the LDM is based on a sequential diffusion process whereby the latent representation, z,



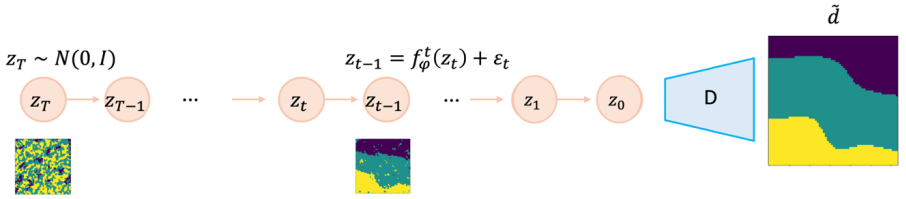
$$L_\theta(d) = \|d - E_\theta(D_\theta(d))\|_2^2$$

(a) Stage 1 training of Latent Diffusion Model. The loss function $L_\theta(d) = ||d - D_\theta(E_\theta(d))||_2^2$, where $||\cdot||_2^2$ stands for the squared norm in $\mathbb{R}^2$, is minimized during training, optimizing the parameters θ of the learnable functions $E(\cdot)$ and $D(\cdot)$.



$$L_\varphi(d) = \left\| d - D\left(f_\varphi^1 \circ f_\varphi^2 \cdots f_\varphi^{T-1} \circ f_\varphi^T(E(d))\right)\right\|_2^2$$

(b) Stage 2 training of Latent Diffusion Model. The loss function $L_\varphi(d) = ||d - D(f_\varphi^1 \circ f_\varphi^2 \cdots f_\varphi^{T-1} \circ f_\varphi^T(E(d))||_2^2$, where $||\cdot||_2^2$ stands for the squared norm in $\mathbb{R}^2$, is minimized during training, optimizing the parameters φ of the learnable functions $f^i(\cdot) \forall i \in [1, T]$.



(c) Sampling from Latent Diffusion Model. Sampling at each stage contains a noise term and the learned diffusion function in training.

Fig. 1 Illustration of training and sampling from the latent diffusion model

is progressively degraded by the repeated addition of noise ϵ in a sequence of diffusion steps. The number of steps, T, between the original representation and the fully noised one is typically on the order of 1,000. In this training stage, the denoising model, ϵ_φ, learns to reverse the noise addition process by estimating the noise added in each diffusion step. To train the denoising model, the latent representation is noised at diffusion steps t to obtain the noised latent representation. Then the learnable parameters (network weights) φ are updated so that the estimated noise is close to the actual noise added,

$$\epsilon_\varphi(z_t, t) \approx \epsilon_t. \tag{1}$$

Due to its dependence on t, the denoising model needs to be exposed to examples at all steps in the diffusion process.

2.1.3 Sampling by Denoising

Once the LDM has been trained, it can generate samples in three steps: (i) Initialization of the process with a latent-space noise vector, z_T, sampled from a Gaussian distribution; (ii) repeated application of ϵ_φ from $t = T$ to $t = 0$ to progressively denoise z_T, running the diffusion process in reverse to obtain the fully denoised latent representation, z_0; and (iii) back-transformation of z_0 from the latent space to the original data space using the decoder D, producing $\tilde{d} = D(z_0)$.

2.2 LDM for Facies Modeling

It is important to recognize the challenges that arise when applying LDMs designed for image generation to facies generation. Whereas image data are usually treated as continuous, facies data are categorical. The LDM proposed by Lee et al. (2024) is specifically designed to work on facies realizations rather than images. This adaptation involves several key modifications to the conventional LDM framework.

First, during the training, a cross-entropy loss is used instead of the mean square error loss, which is ill-suited for categorical data. Second, to improve the generation of conditional facies and ensure the preservation of conditional data, two U-Nets (Ronneberger et al. 2015) are utilized. The first U-Net handles unconditional facies, while the second accounts for facies observations, leading to two distinct reconstruction loss functions for unconditional and conditional facies reconstruction. The first U-Net learns to perform the denoising process. This is enough to perform unconditional sampling. To enable conditional sampling, latent representations of facies observations are extracted using the second U-Net. These representations are then used in the first U-Net to condition the denoising process.

3 Truncated Gaussian Random Fields

This section presents the TGRF models used in this paper. A realization of a TGRF is obtained by first generating a GRF and then thresholding the result at each location (see, e.g., Armstrong et al. (2011) or Lauzon and Marcotte (2022)). In this paper, there are two threshold levels on the real line, giving three different facies classes. With a relatively small amount of conditioning data, the rejection sampler is used for conditional simulation. More advanced sampling methods are required for larger data sizes (see, e.g., Chopin (2011)). The TGRF model is used as a reference data-generating process to create training data for the LDM. Two different cases are studied and outlined next.

3.1 Shoreface Dataset

The shoreface dataset includes a trend that represents geological features such as parasequences or bedsets (Eide et al. 2015; Ovanger et al. 2024). It is related to the dataset used by Lee et al. (2024). The facies observations for this dataset are arranged

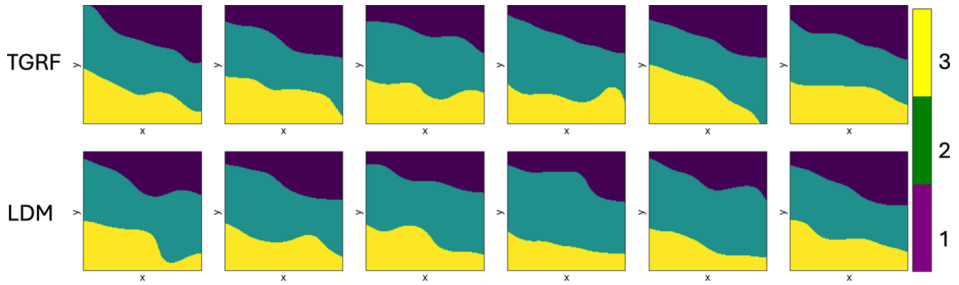


Fig. 2 TGRF and LDM realizations for the shoreface case (unconditional)

in vertical sections, analogous to well data. This dataset is particularly important for assessing the LDM's capacity to handle complex, continuous structures and translate those into the conditional generation process.

The dataset was created based on a GRF with a mean function $\mu(x, y) = -0.0003x + 0.001y$, and a squared exponential covariance function (Müller et al. 2022), with variance parameter $\sigma^2 = 0.005$ and length-scale parameter 25. Realizations are generated on an $n_x \times n_y$ grid defined by $x = y \in \{0, 1, ..., 127\}$, so $n_x = n_y = 128$. The truncation thresholds are set at $[0.15, 0.75]$, which gives the complete specification

$$\begin{aligned} &s(x, y) \sim N\left(\mu(x, y), \Sigma_{\mathrm{SE}}(x, y, x', y')\right), \text{ where } x, y \in \{0, 1, 2, ..., 127\}, \\ &d(x, y) = \begin{cases} 0 \text{ if } s(x, y) < 0.15, \\ 1 \text{ if } s(x, y) \geq 0.15 \text{ and } s(x, y) < 0.75, \\ 2 \text{ if } s(x, y) \geq 0.75, \end{cases} \end{aligned} \tag{2}$$

where $\Sigma_{\mathrm{SE}}(x, y, x', y')$ is the squared exponential kernel function. Figure 2 illustrates examples of these realizations, d. The TGRF dataset employed for training the LDM comprised 5,000 samples.

Conditional data were obtained by first sampling based on Eq. 2, and then extracting a single column, $d(64, y)$, from the middle of the realization. The data are illustrated in Fig. 3. In the rejection sampler (Casella et al. 2004), the match of column values at $x = 64$ determined whether a realization was retained. This process continued until 1,000 suitable realizations were obtained from the conditional TGRF model.

3.2 Laterally Heterogeneous Dataset

The laterally heterogeneous dataset was generated similarly to the shoreface dataset, with dimensions $n_x = n_y = 128$, but with the mean function being zero everywhere, $\mu(x, y) = 0$, and with the truncation thresholds set at $[-0.43, 0.43]$. This second dataset could be seen as a horizontal slice of a laterally heterogeneous reservoir. It was also constructed to increase the degrees of freedom in the TGRF realizations in order to see how the LDM performs in a situation where there is no large-scale pattern in the model. This dataset was constructed using three different covariance functions

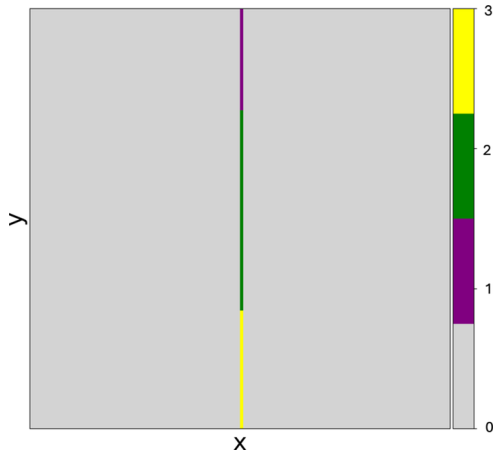


Fig. 3 Shoreface data values

Table 1 Covariance functions used to generate laterally heterogeneous datasets

Covariance function	Variance	Decay parameter	Smoothness parameter
Squared exponential	1.0	24.5	∞
Exponential	1.0	20.0	1/2
Matérn	1.0	21.9	3/2

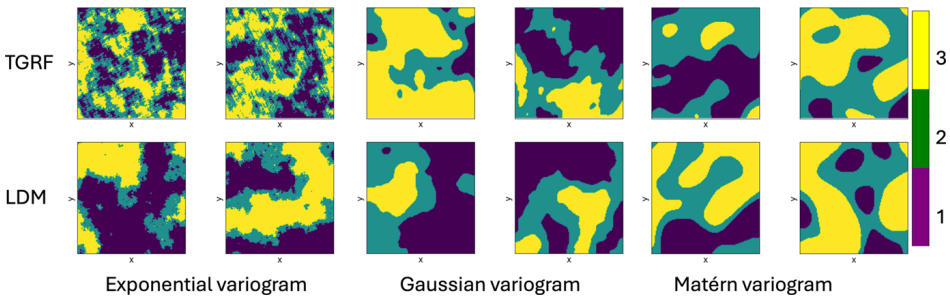


Fig. 4 TGRF and LDM realizations for laterally heterogeneous case (unconditional)

(Table 1). Even though these covariance functions have the same effective correlation length (0.05 correlation at distance 60), they differ in the smoothness they impose on the resulting random field.

Figure 4 illustrates examples of unconditional realizations from both the TGRF model and the LDM model, and Fig. 5 displays the conditioning data. In this dataset, the conditioning data were individual grid cells at random locations. This could represent multiple wells seen in a map view, which is not an uncommon conditioning task in geological settings. Conditional realizations of the TGRF were obtained by rejection sampling, as described for the shoreface case.

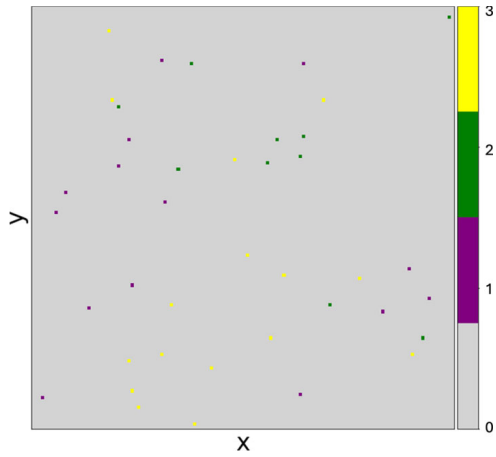


Fig. 5 Data values for the laterally heterogeneous case

4 Comparison Metrics

Multiple metrics are used to compare realizations across the LDM and TGRF models.

4.1 First-Order Statistics

4.1.1 Cell-Wise Probability

Considering a grid with three possible facies in each cell, the occurrences of each facies are counted to determine the marginal probabilities.

4.1.2 Volume Fraction

Volume fraction refers to the proportion of volume in each realization occupied by a particular facies. Specifically,

$$V_{ij} = \frac{\sum_{x=0}^{n_x-1} \sum_{y=0}^{n_y-1} I(d_j(x, y) = i)}{(n_x n_y)}, \tag{3}$$

represents the volume fraction of facies i in realization j. Distributions of volume fractions across both unconditional and conditional realizations offer insight.

4.1.3 Jensen–Shannon Divergence

The Jensen–Shannon divergence (JSD) provides a popular way to compare probability distributions. It is the average of the Kullback-Leibler (KL) divergences computed bidirectionally between two distributions. This symmetrizes the result, avoiding the asymmetry of the KL divergence. This method is applied to compare multinomial cell-wise probabilities between the LDM and TGRF realizations. For each cell, the

trinomial distribution for facies is considered. The computation is performed independently for each cell and yields a comprehensive map of the JSD for both unconditional and conditional datasets. The formulation of the JSD is

$$\begin{aligned} JSD(x, y) &= \frac{1}{2} KLD\left(P_{TGRF}(x, y) \| P_{LDM}(x, y)\right) \\ &\quad + \frac{1}{2} KLD\left(P_{LDM}(x, y) \| P_{TGRF}(x, y)\right), \\ &= \frac{1}{2} \sum_{i=1}^{3} P^i_{LDM}(x, y) \log\left(\frac{P^i_{LDM}(x, y)}{P^i_{TGRF}(x, y)}\right) \\ &\quad + \frac{1}{2} \sum_{i=1}^{3} P^i_{TGRF}(x, y) \log\left(\frac{P^i_{TGRF}(x, y)}{P^i_{LDM}(x, y)}\right), \end{aligned} \tag{4}$$

where $x, y \in \{0, 127\}$ and $P^i_{\cdot}(x, y)$ is the probability of facies type i at location (x, y) for LDM and TGRF. JSD values range from 0 (complete overlap of distributions) to 1 (complete mismatch).

4.2 Second-Order Statistics

The metrics discussed so far all work on a cell-wise basis. However, when dealing with geological objects, one is often interested in pairwise dependencies. This can, for instance, be captured by correlations or variograms. Here, a second-order statistic specifically tailored for discrete realizations is used.

Empirical transiograms (Li 2006; Madani et al. 2019) specify transition probabilities between facies outcomes i and i' as a function of the lag distance $h = (h_x, h_y)$, and can be written as

$$P_{[i,i']}(h) = \frac{\sum_{(x,y)\in S} \mathbf{1}(d(x, y) = i \text{ and } d(x + h_x, y + h_y) = i')}{\sum_{(x,y)\in S} \mathbf{1}(d(x, y) = i)}, \tag{5}$$

where $\mathbf{1}(\cdot)$ is the indicator function, which equals 1 when its argument is true and 0 otherwise, and S is the set of all possible spatial locations.

4.3 Higher-Order Statistics

Although first- and second-order statistics provide essential information about the mean and variance (or correlation) of geological attributes, higher-order statistics capture additional spatial patterns that are often inherent in geological processes.

4.3.1 Sub-Grid Patterns

The study of sub-grid patterns within realizations and their consistency across different datasets is closely related to multiple-point histograms (Lyster et al. 2004). Attention

here is specifically given to 2×2, 3×3, and 4×4 sub-grid patterns, chosen due to their appropriateness for the size of the realizations. Smaller grids might limit the diversity of patterns, whereas larger ones could lead to a sparsely distributed range of patterns. The number of potential sub-grid patterns for an $n \times n$ sub-grid with three facies values is $3^{n \times n}$.

4.3.2 Third-Order Cumulants

The empirical spatial third-order cumulant (Dimitrakopoulos et al. 2010) quantifies asymmetry or directional dependencies in spatial data across three points. For facies $d(x, y)$ at location (x, y), the third-order cumulant $C_3(h_1, h_2)$ is given by averaging the products of deviations from the mean across all triplets separated by spatial lags $h_1 = (h_{x,1}, h_{y,1})$ and $h_2 = (h_{x,2}, h_{y,2})$,

$$C_3(h_1, h_2) = \frac{1}{|N(h_1, h_2)|} \sum \delta_a \delta_b \delta_c, \quad \delta_a = d(x_a, y_a), \tag{6}$$

where the sum goes over all triplet location sets (x_a, y_a), (x_b, y_b) and (x_c, y_c) with lags h_1 and h_2 (a set of cardinality $N(h_1, h_2)$). This cumulant captures higher-order spatial interactions in the facies variable, and it helps detect skewed or asymmetric structures in the spatial data.

5 Shoreface Case

5.1 First-Order Statistics

This subsection presents a comparative analysis of statistical measures for unconditional and conditional realizations using the LDM and TGRF methods with first-order metrics, as outlined in Sect. 4.1. The comparison begins with basic acceptance criteria, including data matching and cell-wise probabilities, and then progresses to more complex marginal metrics.

5.1.1 Data Conditioning

A primary metric in data conditioning is data matching, which evaluates how accurately the generated realizations preserve conditioning data points. In the generation of 1,000 samples, 65 LDM samples failed to maintain all data points in the conditioning, showing a maximum of two mismatching values along a vertical trajectory with 128 data points. Figure 6 displays six samples that did not perfectly match the data, highlighting discrepancies with a red marker. In particular, data mismatches consistently occur during the transition from one facies to another, rather than within homogeneous areas.

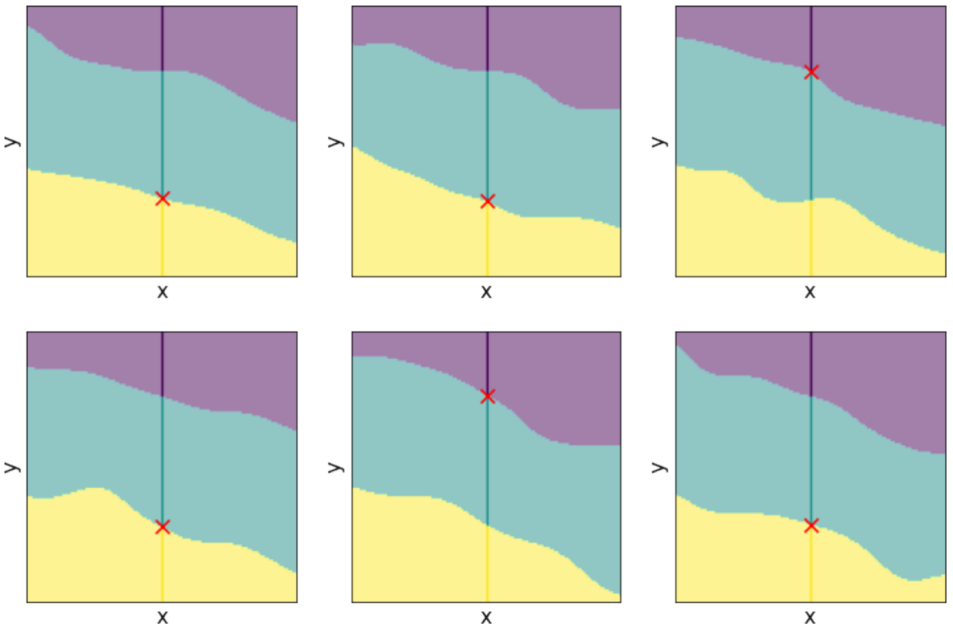
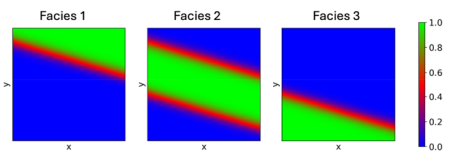
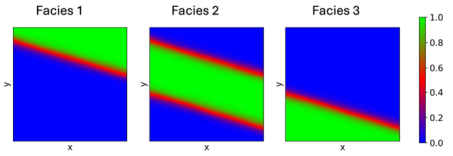


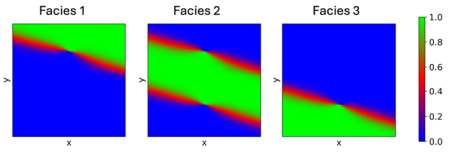
Fig. 6 Conditional realizations from the LDM with data mismatch, highlighted by red markers for the shoreface dataset



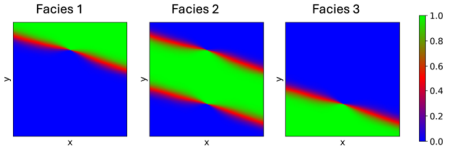
(a) Cell-wise probabilities for unconditional TGRF



(b) Cell-wise probabilities for unconditional LDM



(c) Cell-wise probabilities for conditional TGRF.



(d) Cell-wise probabilities for conditional LDM.

Fig. 7 Cell-wise probability for shoreface dataset

5.1.2 Cell-Wise Probability

The cell-wise probabilities are illustrated in Fig. 7. Distinguishing between the cell-wise probabilities of the TGRF (Fig. 7a and c) and LDM (Fig. 7b and d) samples is challenging. For all three facies, there are distinct areas with probabilities 0 and 1, with transition areas in between (facies transitions). This pattern is attributed to the dominance of the trends and transition areas reflecting the stochasticity of the samples.

In the conditional cell-wise probability (Fig. 7c and d), there is an absence of a transition area at $x = 64$. This is consistent with the observation conditions, as the facies outcomes for that column are known, eliminating variation in the observed section. Furthermore, for both the TGRF and LDM models, increasing thickness of the transition area is observed moving away from the observed data, aligning with the expectation of convergence towards prior probabilities away from observational data.

5.1.3 Volume Fraction

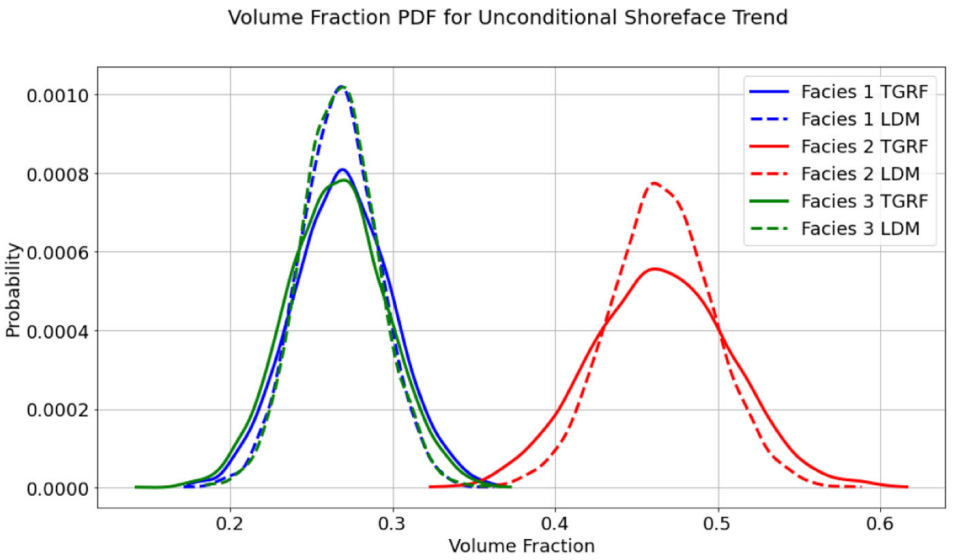
Empirical volume fraction probability density functions are presented in Fig. 8. In both the unconditional and conditional cases, alignment can be observed between the modes of the LDM and TGRF curves. This means that the average volume fractions across all images are quite well preserved. However, a notable distinction is that the distributions for the LDM are much narrower than those for the TGRF. This means that the variances in volume fractions across realizations are significantly underestimated. This is consistent with the general tendency of generative models to slightly underestimate the variance in datasets, especially apparent, for example, in GANs, where mode collapse is a major issue (Thanh-Tung and Tran 2020). In Fig. 8, differences between the unconditional and conditional cases are not very large, and the variances seem to be somewhat more aligned in the conditional case (Fig. 8b). This also might be due to the fact that the overall variance is lower for conditional realizations.

5.1.4 Jensen–Shannon Divergence

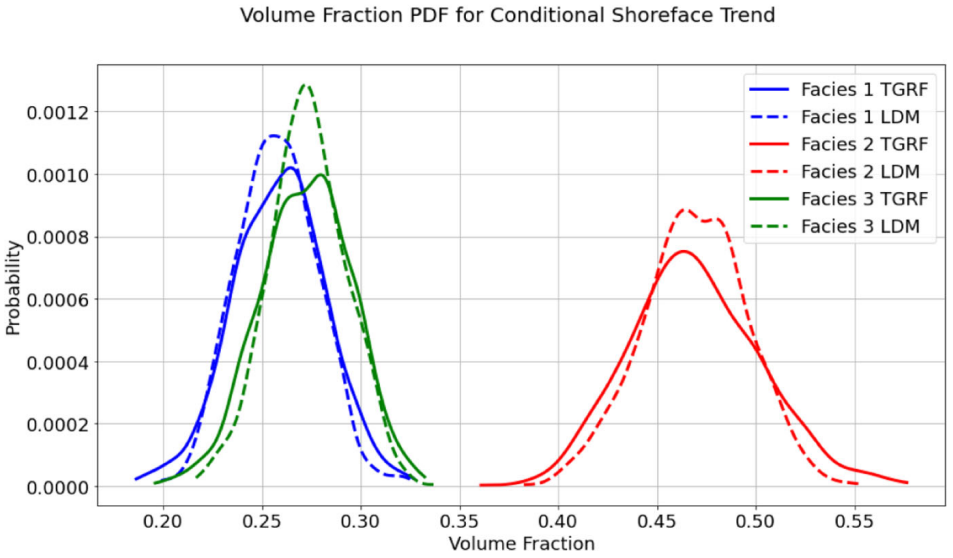
Figure 9 presents the unconditional and conditional JSD between LDM and TGRF realizations. As presented in Sect. 4.1, the JSD compares the occurrence frequencies of facies in each cell of the grid, where 0 indicates a total overlap of frequencies, and 1 indicates a complete mismatch. In the analysis of the unconditional case, a distinct pattern is seen, where all nonzero JSDs occur at a certain region above and below the expected facies transitions. Interestingly, the JSD between the models is exactly zero at transitions in the trend function where there is a probability of 0.5 for both facies present at a transition. This means that the LDM has learned to exactly reproduce the transition. However, the area around the transition is where the largest divergence is observed. This is because LDM realizations have too narrow a transition region between facies. Hence, the variance in this transitional region is underrepresented by the LDM. The same is observed for the conditional case, although of lesser magnitude. It is interesting to note the green line that extends from the conditioning data column. This is because some LDM realizations are not able to preserve the conditioning data, placing the transition one cell above the true transition and propagating this error some cells horizontally away from the conditioning data.

5.2 Second-Order Statistics

The transiograms (Fig. 10) are largely determined by the trend in the shoreface dataset, and the close agreement between the LDM and TGRF models can be attributed primarily to this trend. The LDM results capture the trend well, with mean transiogram



(a) Unconditional distributions



(b) Conditional distributions.

Fig. 8 Volume fraction distributions for unconditional and conditional realizations from LDM (dashed) and TGRF (solid)

values that closely align with those of TGRF. Additionally, the two-standard-deviation envelopes indicate that the variability between datasets is also well matched, showing no underestimation of variance. This alignment underscores the ability of the LDM to accurately capture both correlation structures and variability in the training data. The U-Net architecture, known for capturing long-range dependencies through its fully connected design, further supports this capability by effectively translating patterns

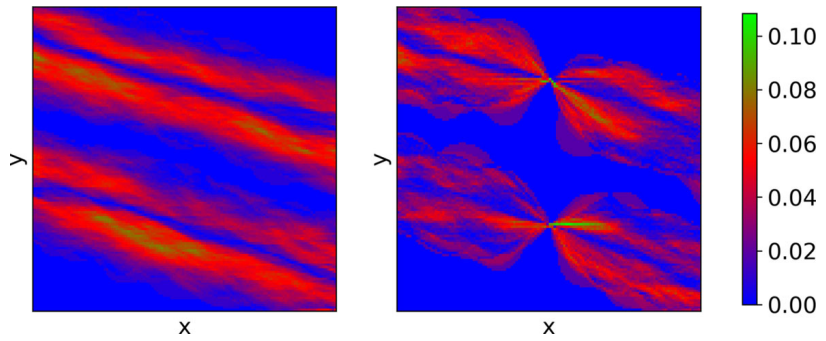


Fig. 9 Jensen–Shannon divergence between the LDM and TGRF datasets. (Left) Unconditional. (Right) Conditional

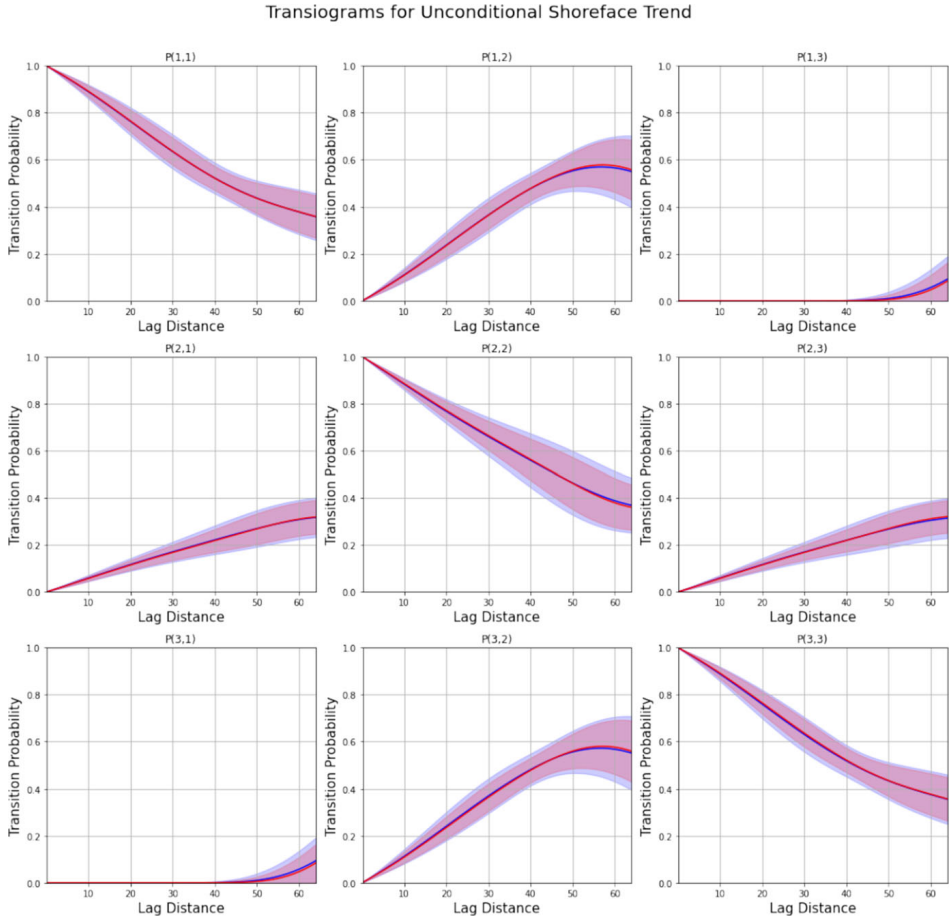


Fig. 10 Transiogram of unconditional realizations from the TGRF model and LDM on the shoreface dataset. The blue color is associated with TGRF and red is associated with LDM

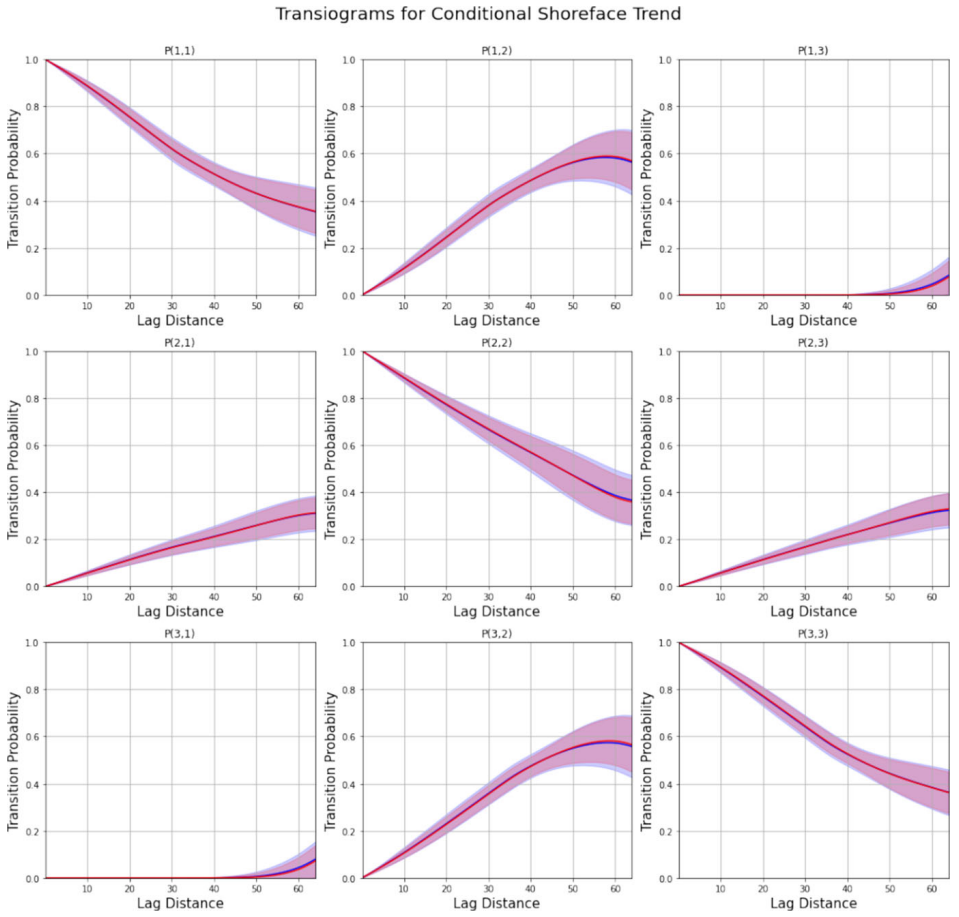


Fig. 11 Transiogram of conditional realizations from the TGRF model and LDM on the shoreface dataset. The blue color is associated with TGRF and red is associated with LDM

across scales (Shelhamer et al. 2014). The same analysis is extended to the conditional case in Fig. 11, where the trend continues to dominate the transiograms, with the stochastic component of the fields playing a minimal role. The high degree of alignment in transition probabilities between LDM and TGRF further demonstrates LDM's reliability in reproducing the model's correlation structure.

5.3 Higher-Order Statistics

For higher-order statistics, local patterns are considered here, starting with 2×2 patterns. Of the 81 possible unique patterns, only 24 are observed. The single facies patterns are highly dominant, and occur at a frequency proportional to the volume fractions. The frequencies of the remaining patterns for the unconditional case are shown in Fig. 12. The histogram shows, on a logarithmic scale, the average number of occurrences of a pattern per realization. Approximately half of the patterns occur with a similar frequency in the LDM and TGRF models, while 10 of the sub-grid patterns

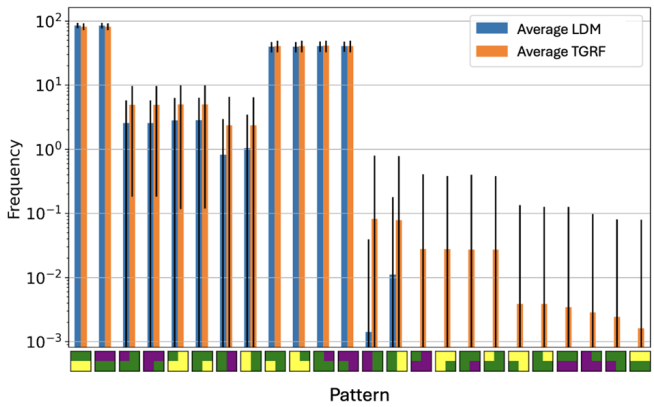


Fig. 12 2 × 2-sub-grid pattern histogram for the unconditional LDM and TGRF model on the shoreface dataset

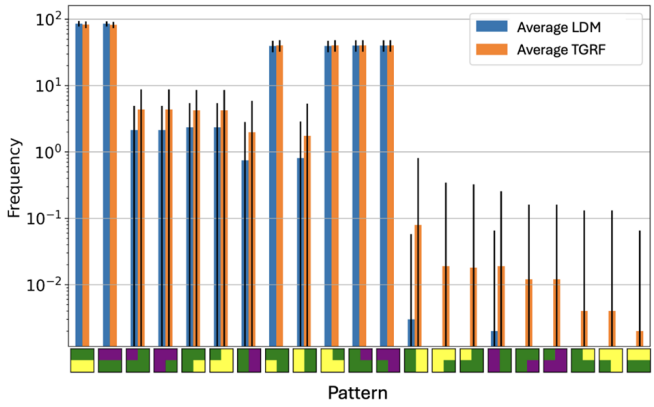


Fig. 13 2 × 2-sub-grid pattern histogram for the conditional LDM and TGRF model on the shoreface dataset

are exclusive to the TGRF model. This trend is consistent at the 3 × 3 and 4 × 4 scales, where approximately half of the patterns are unique to the TGRF dataset. However, only 0.03% of the total number of 2 × 2 sub-grids in the TGRF dataset are missing from the LDM dataset. For 3 × 3 sub-grids, this number is 0.05%, and for 4 × 4 sub-grids it is 0.07%. Thus, while the TGRF dataset exhibits greater pattern variability, the actual number of occurrences where this variability is not mirrored in the LDM dataset is minuscule. The same results are seen in the conditional case, although the number of patterns present there is smaller, as seen in the 2 × 2 case in Fig. 13.

Further investigation into patterns exclusive to the TGRF dataset provides insights into the types of patterns the LDM model does not replicate. The most frequently occurring patterns unique to the TGRF dataset exhibit similarities across different scales. For instance, the second most common pattern at the 2 × 2 scale, the second most common at the 3 × 3 scale, and the most common at the 4 × 4 scale all include a green patch in the bottom right corner within a yellow area. Generally, patterns not present in the LDM dataset are those where the natural order of facies is reversed (e.g., green above purple, yellow above green), contrasting with trends observed in training

Table 2 Summary statistics for shoreface sub-grid patterns

Summary statistic	2×2	3×3	4×4	5x5
Unique unconditional LDM patterns	14	78	265	690
Unique unconditional TGRF patterns	24	153	525	1, 393
Unique conditional LDM patterns	14	73	230	583
Unique conditional TGRF patterns	21	104	338	875

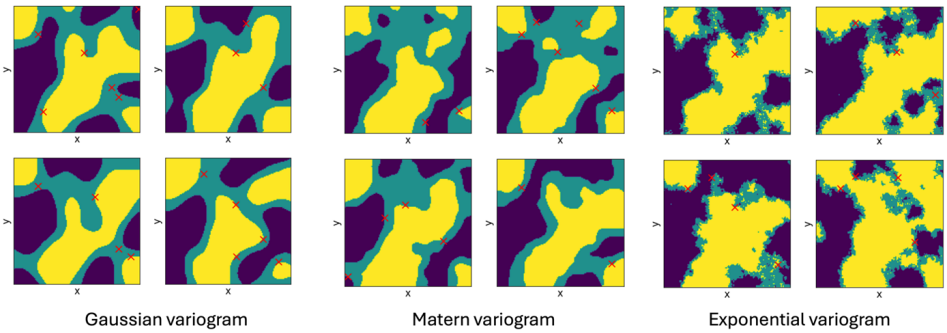


Fig. 14 Conditional realizations from the LDM with data mismatch, highlighted by red markers, for the laterally heterogeneous dataset

realizations. This highlights a limitation in the LDM's ability to capture improbable patterns (Table 2).

6 Laterally Heterogeneous Case

In this experiment, the realizations have larger degrees of freedom because there is no underlying trend creating a fixed pattern. This leads to much larger variability in possible realizations and thus a greater challenge for the LDM model. As expected, the performance of the LDM is worse here.

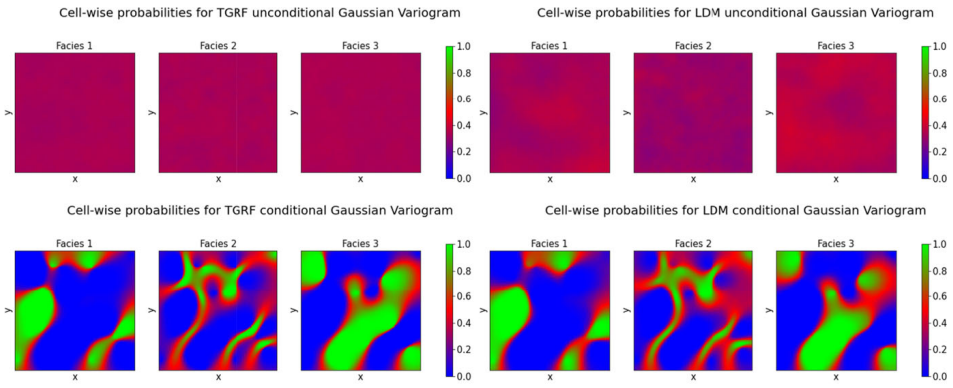
6.1 First-Order Statistics

6.1.1 Data Conditioning

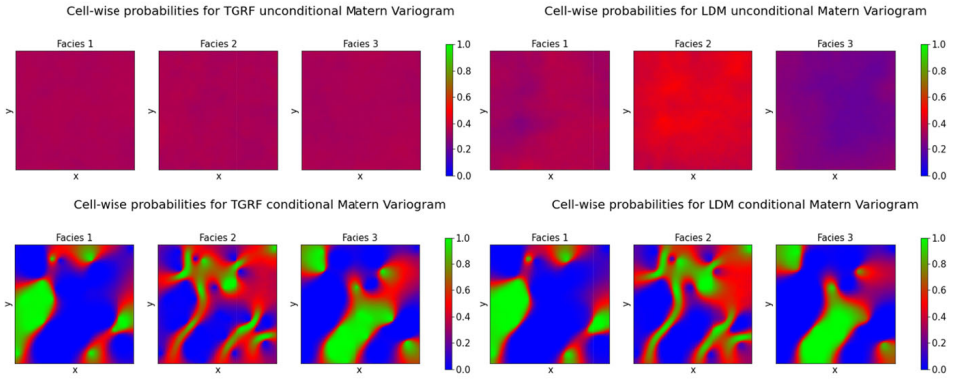
Figure 14 displays conditional LDM realizations. Most of the 40 conditioning points are preserved in the LDM realizations; however, many of the realizations get some of the points wrong. In total, out of 1,000 realizations, 856 get at least one conditioning value wrong. One particularly troublesome conditioning observation in the upper left area of the grid has the wrong facies value in a majority of realizations.

6.1.2 Cell-Wise Probability

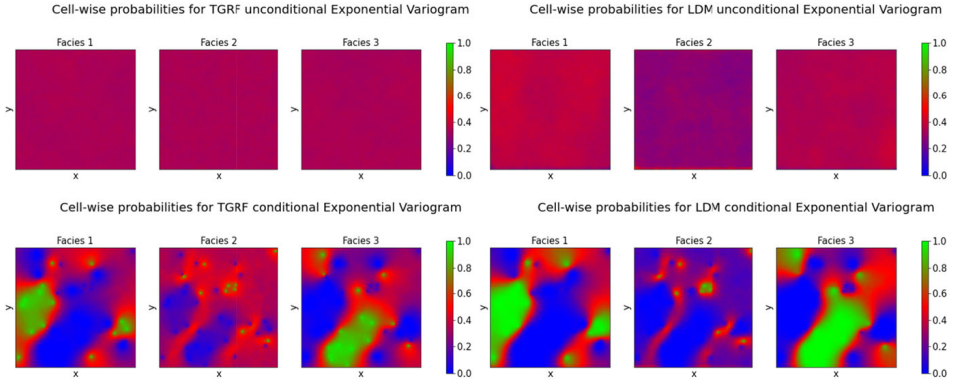
Figure 15 shows the cell-wise probabilities of the laterally heterogeneous dataset for



(a) Gaussian variogram cell-wise probabilities.

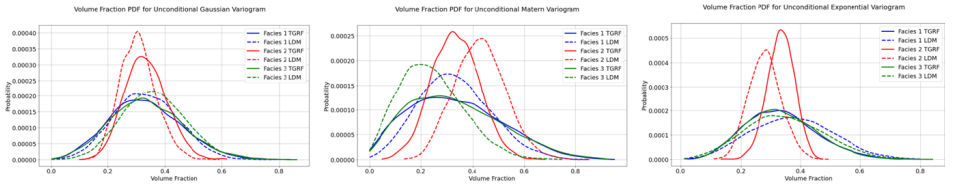


(b) Matérn variogram cell-wise probabilities.

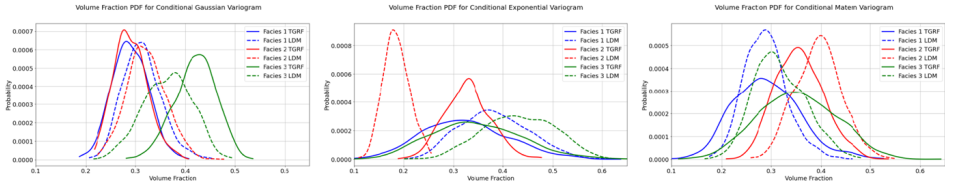


(c) Exponential variogram cell-wise probabilities.

Fig. 15 Cell-wise probabilities of the laterally heterogeneous dataset



(a) Unconditional distributions



(b) Conditional distributions.

Fig. 16 Volume fraction distributions for unconditional and conditional realizations from LDM (dashed) and TGRF (solid)

both the unconditional and conditional states of the LDM and TGRF models. Within the unconditional datasets, a notable artifact is observed, namely that the cell-wise probability for facies 2 is higher in the LDM model than in the TGRF model in the Matérn case, with an average volume fraction of 0.45 versus 0.33 in the TGRF model. This also causes the cell-wise probabilities for facies 1 and 2 to be lower than those of the TGRF model. Both models also show a slight fluctuation in cell-wise probability in the grid due to Monte Carlo noise. For the conditional cell-wise probabilities, a close alignment is observed between the two models. Nevertheless, some differences remain, particularly in the exponential case, where the probability is more widely spread in the TGRF case and more concentrated around the conditioning points in the LDM case.

6.1.3 Volume Fraction

Figure 16a presents the volume fraction for the unconditional case. This figure reinforces the observations made in the previous section, specifically regarding the disproportionate fraction of facies 2 in the Matérn LDM model compared to the TGRF for the unconditional data, along with reduced fractions of facies 1 and 2. The reason behind the LDM model's tendency to overestimate the occurrence of the middle facies remains unclear. The underestimation of variance noted for the previous dataset is also apparent here.

In contrast, the conditional datasets (Fig. 16b) have larger discrepancies between the distributions, especially in the exponential case, where the fraction of facies 2 is vastly underestimated while that of facies 1 and 2 is overestimated. This pattern is not consistent, because LDM overestimates the fraction in the Matérn and Gaussian cases. There is clearly a response to the conditioning points in all cases, but the response can differ between the LDM and TGRF models.

6.1.4 Jensen–Shannon Divergence

The JSD is shown in Fig. 17. The unconditional case shows a low divergence for the Gaussian data, a slightly higher divergence for the exponential case, and a high divergence for the Matérn case, which is likely due to the overrepresentation of facies 2 in the LDM. The conditional cases have regions of low and high divergence consistent across variograms. This is expected from conditioning, typically giving low divergence near the conditioning points and higher away from it. Being less smooth, the exponential case shows higher divergence. The Gaussian case has longer regions of high divergence than the Matérn (despite more information from neighboring cells) close to certain conditioning points where there are high divergences (conditioning errors). This indicates that errors or outliers propagate further in the Gaussian case.

6.2 Correlation Structures

Figure 18a, c, and e show the average transiograms for the unconditional datasets with the two standard deviation bands. Compared with the previous dataset, the correlation structure has not been captured as successfully here. However, this appears to be mainly due to poor reproduction of volume fractions. As the distance increases and correlations vanish, the transiogram converges to $\sum_i p_i(1 - p_i)$, where p_i is the marginal probability of facies i in a cell. When the probability for the most dominant facies increases, this sum decreases. The display shows that the values level off after a distance of 50, which is indicative of independence beyond that point. In the conditional realizations seen in Fig. 18b, d, and f, the transiograms in the exponential case have a larger mismatch, as the volume fraction in facies 2 is underrepresented in the LDM. The same goes for the Gaussian case, where the transiograms converge to different values, especially for facies 3. Conversely, in the Matérn case, the transiograms are more aligned under conditioning.

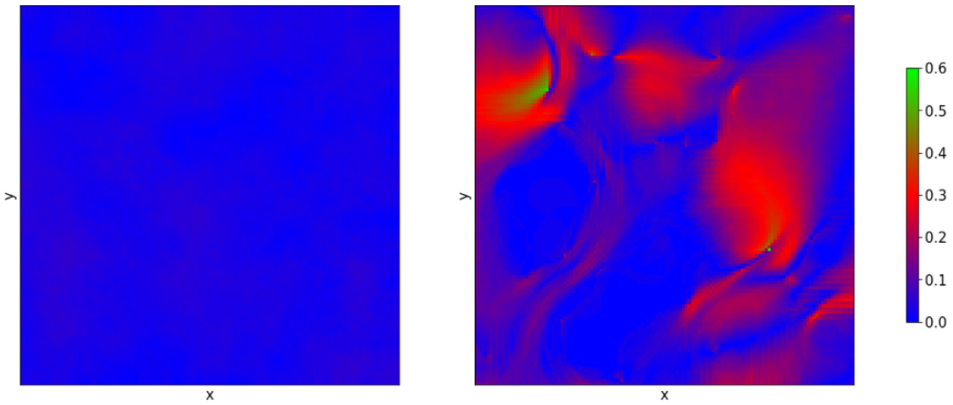
6.3 Third-Order Cumulants

For high-order metrics (Boisvert et al. 2010; De Iaco and Maggio 2011), third-order cumulants are studied here. In Fig. 19, they have been computed for all datasets and for three different combinations of lags and angles. These templates are as follows:

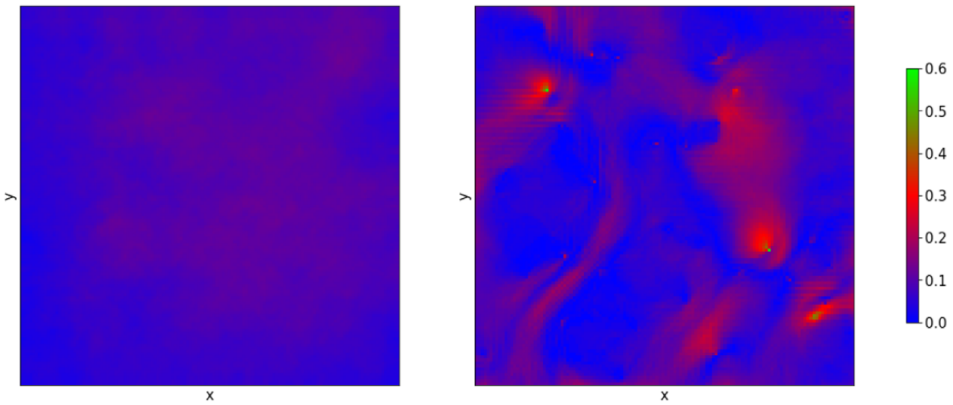
- $h_1 = h_2 = 3$, angle $= 90°$,
- $h_1 = h_2 = 20$, angle $= 90°$,
- $h_1 = h_2 = 3$, angle $= 135°$.

For each of the three templates, cumulants are summarized by scanning through the realizations.

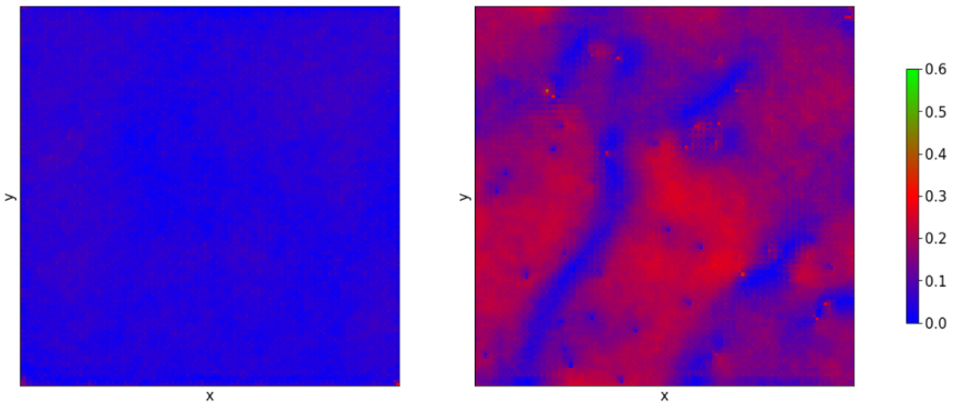
For all cases, the variances in the cumulants are lower in the conditional case. This is especially apparent with the Gaussian variogram. This is a natural effect of conditioning, where the conditioning points reduce the variability of the template pattern. Since the long-range dependencies are less in the Matérn case and even less so in the exponential case, the variance reduction effect by conditioning is less here.



(a) JSD for Gaussian variogram

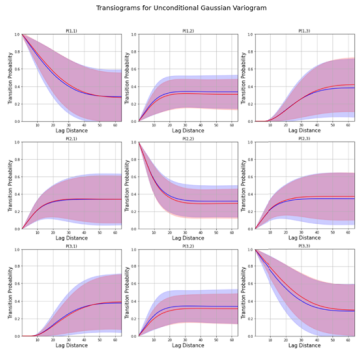


(b) JSD for Matérn variogram

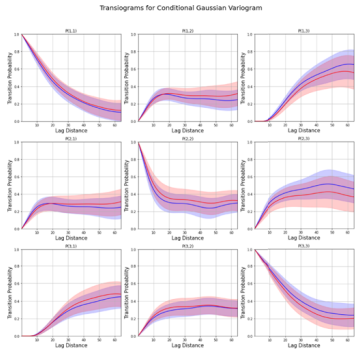


(c) JSD for exponential variogram

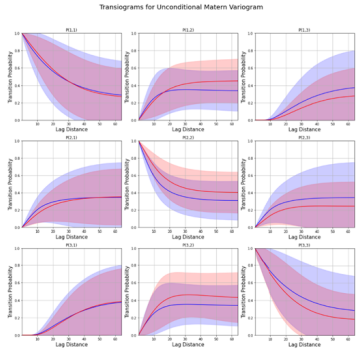
Fig. 17 JSD for zero-trend datasets. Unconditional on the left and conditional on the right



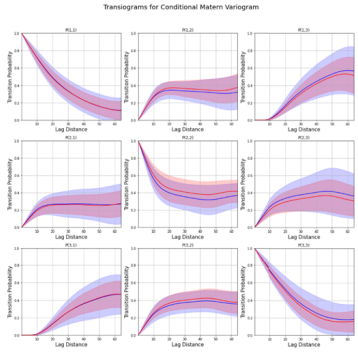
(a) Transiograms of unconditional Gaussian variogram.



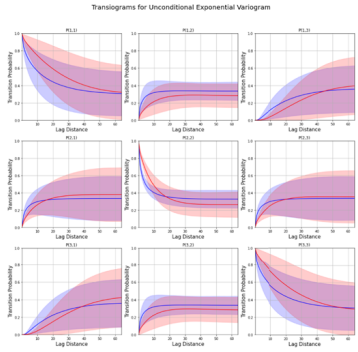
(b) Transiograms of conditional Gaussian variogram.



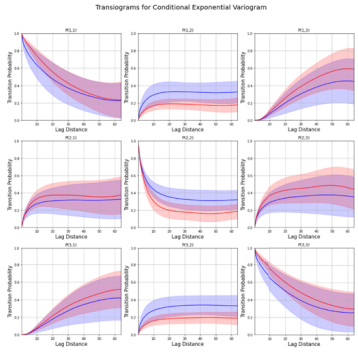
(c) Transiograms of unconditional Matérn variogram.



(d) Transiograms of conditional Matérn variogram.



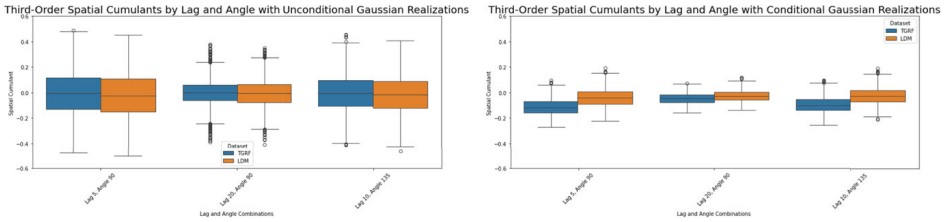
(e) Transiograms of unconditional Exponential variogram.



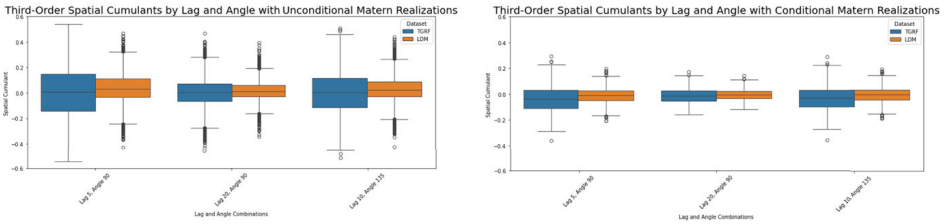
(f) Transiograms of conditional Exponential variogram.

Fig. 18 Transiograms of the lateral heterogeneous case. The blue color is associated with TGRF and red is associated with LDM

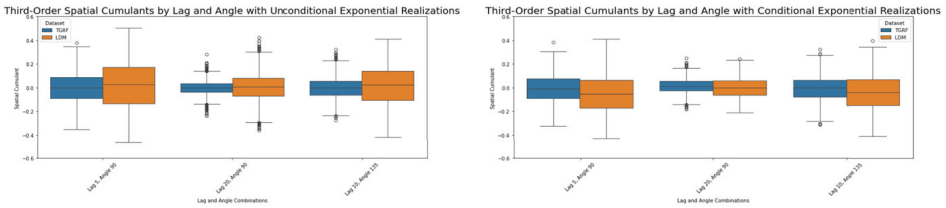
This is captured well by the LDM in all cases. In addition, there is some discrepancy between the LDM and TGRF cumulants in both mean and standard deviation. In the exponential case, the LDM cumulants show much higher variation, revealing greater spatial variability in the three-point pattern than in the TGRF. However, this is the opposite case with the Matérn variogram, revealing no consistent pattern.



(a) Third-order cumulants for the Gaussian variogram.



(b) Third-order cumulants for the Matérn variogram.



(c) Third-order cumulants for the Exponential variogram.

Fig. 19 Third-order cumulants for the lateral heterogeneous datasets. Unconditional (left) and conditional (right)

7 Conclusion

LDMs have considerable potential for high-quality facies model generation. They outperform more traditional machine learning methodologies in reproducing intricate geological features while honoring hard data in the form of facies observations in wells (Lee et al. 2024). This article presented a comparative evaluation of the LDM's output with the TGRF reference. Using multiple metrics, the study gained insight into the statistical properties of the distribution of LDM-generated realizations.

In the shoreface case, by most of the metrics considered, the LDM and TGRF distributions differed only slightly. There were, however, some notable distinctions. First, the LDM favored sharper transitions, leading to underestimation of the marginal variance in transition regions. Second, the LDM underrepresented the diversity of facies patterns at scales ranging from 2×2 to 4×4 grid cells. In the laterally heterogeneous case, greater variability was observed between the realizations, and the differences between the LDM and TGRF output were somewhat larger. First, facies volume fractions were inaccurate in the unconditional case. There was too much of facies 1 in the Matérn case and not enough of the other two. Second, the conditional LDM results got the wrong facies in at least one cell 85% of the time.

High-quality performance on a reference dataset does not mean that an approach will generalize well on a different case. For LDMs, preserving connected data is easier than preserving individual cells, because of the compression imposed by the autoencoder. If the grid size were larger or in three-dimensional data, the compression rate of the autoencoder would have to be higher to keep the model's computational requirements on a comparable level. This would likely exacerbate compression artifacts. Future efforts to apply LDMs to facies modeling should aim to mitigate the issues identified here by emphasizing data preservation and accurate representation of correlation structures.

Acknowledgements Thanks to the Norwegian Research Council for funding the GEOPARD project (319951), the Machine Learning for Irregular Time Series (ML4ITS) project (312062) and the SFI Centre for Geophysical Forecasting (309960).

Funding Open access funding provided by NTNU Norwegian University of Science and Technology (incl St. Olavs Hospital - Trondheim University Hospital) We acknowledge support from the Norwegian Research Council project GEOPARD grant 319951 and the SFI Centre for Geophysical Forecasting grant 309960.

Declarations

Conflict of interest The authors have no conflict of interest. This work has not been submitted elsewhere.

References

Armstrong M, Galli A, Beucher H, Loc'h G, Renard D, Doligez B, Eschard R, Geffroy F (2011) Plurigaussian simulations in geosciences. Springer Science & Business Media, Berlin

Boisvert JB, Pyrcz MJ, Deutsch CV (2010) Multiple point metrics to assess categorical variable models. Nat Resour Res 19:165–175

Caers J (2005) Petroleum geostatistics. Society of Petroleum Engineers Richardson

Casella G, Robert CP, Wells MT (2004) Generalized accept-reject sampling schemes. Lecture Notes-Monogr Ser 45:342–347 (**ISSN 07492170**)

Chan S, Elsheikh AH (2019) Parametric generation of conditional geological realizations using generative neural networks. Comput Geosci 23(5):925–952

Chopin N (2011) Fast simulation of truncated Gaussian distributions. Stat Comput 21:275–288

De Iaco S, Maggio S (2011) Validation techniques for geological patterns simulations based on variogram and multiple-point statistics. Math Geosci 43:483–500

Dimitrakopoulos R, Mustapha H, Gloaguen E (2010) High-order statistics of spatial random fields: exploring spatial cumulants for modeling complex non-Gaussian and non-linear phenomena. Math Geosci 42(1):65–99

Eide C, Howell J, Buckley S (2015) Sedimentology and reservoir properties of tabular and erosive offshore transition deposits in wave-dominated, shallow-marine strata: Book cliffs, usa. Pet Geosci 21:55–73

Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. Commun ACM 63(11):139–144

Haldorsen HH, Damsleth E (1990) Stochastic modeling. J Pet Technol 42, ISSN 0022-3522

Ho J, Jain A, Abbeel P (2020) Denoising diffusion probabilistic models. In: Advances in neural information processing systems 2020-December, ISSN 10495258

Kollovieh M, Ansari AF, Bohlke-Schneider M, Zschiegner J, Wang H, Wang Y (2023) Predict, refine, synthesize: self-guiding diffusion models for probabilistic time series forecasting. arXiv preprint arXiv:2307.11494

Lauzon D, Marcotte D (2022) Statistical comparison of variogram-based inversion methods for conditioning to indirect data. Comput Geosci 160:105032

Lee D, Ovanger O, Eidsvik J, Aune E, Skauvold J, Hauge R (2024) Latent diffusion model for conditional reservoir facies generation. Comput Geosci 105750

Li W (2006) Transiogram: a spatial relationship measure for categorical data. Int J Geogr Inf Sci 20(6):693–699

Liu H, Chen Z, Yuan Y, Mei X, Liu X, Mandic D, Wang W, Plumbley MD (2023) Audioldm: Text-to-audio generation with latent diffusion models. arXiv preprint arXiv:2301.12503

Lopez S, Cojan I, Rivoirard J, Galli A (2009) Process-based stochastic modelling: meandering channelized reservoirs. In: Analogue and numerical modelling of sedimentary systems: from understanding to prediction, Wiley, Oxford, UK, 139–144

Lucic M, Kurach K, Michalski M, Gelly S, Bousquet O (2018) Are gans created equal? A large-scale study. In: Advances in neural information processing systems 31

Lugmayr A, Danelljan M, Romero A, Yu F, Timofte R, Van Gool L (2022) Repaint: Inpainting using denoising diffusion probabilistic models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 11461–11471

Lyster S, Ortiz J, Deutsch C (2004) Scaling multiple point statistics to different histograms. In: Center for Computational Geostatistics Annual Report Papers

Madani N, Maleki M, Emery X (2019) Nonparametric geostatistical simulation of subsurface facies: tools for validating the reproduction of, and uncertainty in, facies geometry. Nat Resour Res 28:1163–1182

Mannseth T (2014) Relation between level set and truncated pluri-Gaussian methodologies for facies representation. Math Geosci 46(6):711–731

Manzocchi T, Walsh DA (2023) Vertical stacking statistics of multi-facies object-based models. Math Geosci 55(4):461–496

Matheron G, Beucher H, de Fouquet C, Galli A, Guérillot D, Ravenne C (1987) Conditional simulation of the geometry of fluvio-deltaic reservoirs. In: SPE annual technical conference and exhibition?, Spe, SPE–16753

Müller S, Schüler L, Zech A, Heße F (2022) `GSTools` v1.3: a toolbox for geostatistical modelling in python. Geosci Model Dev 15(7):3161–3182

Ovanger O, Eidsvik J, Skauvold J, Hauge R, Aarnes I (2024) Addressing configuration uncertainty in well conditioning for a rule-based model. Mathematical Geosciences :1–26

Pyrcz MJ, Deutsch CV (2014) Geostatistical reservoir modeling. Oxford University Press, Oxford

Pyrcz MJ, Sech RP, Covault JA, Willis BJ, Sylvester Z, Sun T (2015) Stratigraphic rule-based reservoir modeling. Bull Can Pet Geol 63(4):287–303

Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B (2022) High-resolution image synthesis with latent diffusion models

Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18, Springer, 234–241

Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X (2016) Improved techniques for training gans. In: Advances in neural information processing systems 29

Shelhamer E, Long J, Darrell T (2014) Fully convolutional networks for semantic segmentation. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 3431–3440

Strebelle S (2002) Conditional simulation of complex geological structures using multiple-point statistics. Math Geol 34:1–21

Thanh-Tung H, Tran T (2020) Catastrophic forgetting and mode collapse in gans. In: 2020 international joint conference on neural networks (ijcnn), IEEE, 1–10

Zhang TF, Tilke P, Dupont E, Zhu LC, Liang L, Bailey W (2019) Generating geologically realistic 3d reservoir facies models using deep learning of sedimentary architecture with generative adversarial networks. Pet Sci 16, ISSN 19958226

Statistical Properties of Binary Image Posterior Vision Transformer Samples

Oscar Ovanger, Jo Eidsvik, Ragnar Hauge and Jacob Skauvold

To be submitted

Statistical Properties of Binary Image Posterior Vision Transformer Samples

Oscar Ovanger[1], Jo Eidsvik[1], Ragnar Hauge[2], and Jacob Skauvold[2]

[1]Department of Mathematical Sciences, NTNU, Norway
[2]Norwegian Computing Center, Norway

Abstract

This paper presents a comprehensive empirical study of Vision Transformer (ViT) models for conditional sample generation of binary images. The performance of ViT-based samplers is evaluated against an exact Variable Elimination Algorithm (VEA) baseline for Markov random field models across a suite of statistical and structural metrics, including marginal distributions, sample log-likelihoods, covariance structures, pattern frequencies, and pointwise structural similarity. The results demonstrate that ViT sampling models successfully capture large-scale spatial structures and respect conditioning constraints, achieving visually plausible samples with significantly improved computational efficiency and flexibility. However, the analysis also reveals statistical limitations: ViT models exhibit systematic biases and misunderstand correlation structure, simultaneously overconditioning around observed pixels (creating excessive long-range correlations) while undersmoothing elsewhere (correlations too short). Temperature adjustment present a problematic trade-off, improving correlation lengths in unconditioned regions while amplifying conditioning biases. While maintaining appropriate distributional variance, ViT samples are systematically shifted toward lower-probability regions. These findings highlight both the promise of ViTs for efficient conditional sampling and the importance of comprehensive statistical evaluation beyond visual quality.

Index Terms

Conditioning problems, Markov random fields, Vision Transformer, generative models, image inpainting, explainable AI.

I. Introduction

Conditioning problems involving spatially distributed *categorical* image data are ubiquitous in science and engineering. A prime example that motivates this work is that of subsurface facies modeling in geoscience, where one aims to infer a discrete geological field (e.g., rock types) from sparse observations such as well data or outcrop measurements [1].

Given observations $\mathbf{d}$, the goal is then to estimate a high-dimensional field $\mathbf{x} = [x_1, \ldots, x_N]^T$ defined on an N-pixel grid, where each x_i is a discrete class label. For characterizing the conditional distribution $p(\mathbf{x} \mid \mathbf{d})$, one combines prior knowledge $p(\mathbf{x})$ with the constraints imposed by the data. Here $p(\mathbf{x})$ encodes spatial continuity and geological patterns (e.g., through training images or Markov random fields (MRF) models). This categorical conditioning problem is inherently challenging: $\mathbf{x}$ consist of combinatorial-size sets of discrete variables which exhibit complex spatial dependencies. Conventional gradient-based methods are not applicable because derivatives with respect to discrete classes are undefined, necessitating either brute-force search in a high-dimensional combinatorial space or stochastic sampling strategies.

Bayesian statistical sampling techniques such as Markov chain Monte Carlo (MCMC) are principled approaches to sample from $p(\mathbf{x}|\mathbf{d})$ assuming a prior that can be evaluated up to a normalizing constant. In theory, MCMC can asymptotically produce samples from the posterior distribution. However, applying MCMC to large categorical image models with realistic prior complexity faces severe difficulties. The state space grows as C^N for C discrete classes, leading to prohibitively slow convergence. Multi-modal or geologically-constrained distributions exacerbate the problem, with slow-mixing Markov chains often trapped in a single posterior mode [2]. Even advanced schemes such as annealing or adaptive importance sampling struggle with this curse of dimensionality [3], [4], [5]. The challenges of Bayesian conditioning for such models motivates alternative more efficient sampling approaches.

Recent years have seen the adoption of generative AI methods for spatial modeling problems. Generative adversarial networks (GANs) [6], variational auto-encoders (VAEs) [7], and diffusion models [8] have demonstrated the ability of generative AI models to learn high-dimensional distributions and produce realistic samples quickly. These models can serve as powerful priors or proposal generators: for instance, a GAN trained on geological examples can generate facies realizations that honor learned patterns [5], while VAEs and diffusion models enable efficient generation via latent space manipulation [9]. The key advantage is computational efficiency – once trained, these models generate diverse samples in a single forward pass (GAN/VAE) or short iterative refinement (diffusion), bypassing MCMC's slow iterations. Moreover, deep generative models can capture intricate spatial distributions beyond traditional parametric priors [10], [11].

Despite their promise, current generative AI methods have notable shortcomings for conditional sampling in a Bayesian context. First, most deep generators are black-box samplers without explicit probability densities. GANs learn to mimic $p(\mathbf{x})$ without providing likelihoods, making it impossible to evaluate $p(\mathbf{x}|\mathbf{d})$ rigorously [6], [12]. VAEs yield approximate likelihoods, but often produce blurry samples and underestimate variability [13], [14]. Diffusion model likelihoods, while theoretically

defined, are often computationally intractable [15], [16]. This lack of tractable likelihoods hinders Bayesian integration where samples must be weighted by their likelihood probability [17]. Second, conditioning these models on arbitrary observation data $\mathbf{d}$ is inflexible. Most require specialized architectures or training procedures for each conditioning scenario [18], [19]. If the spatial arrangement or type of conditioning changes (e.g., different well locations), models typically require retraining [20], [21]. This geometry-dependent conditioning severely limits practical applicability.

Vision Transformers (ViTs) offer a promising alternative that addresses these limitations. When used autoregressively, ViTs factorize $p(\mathbf{x})$ as $p(x_1)\prod_{i=2}^{N} p(x_i \mid x_1, \ldots, x_{i-1})$ and model the expression with a transformer that sequentially predicts tokens [22], [23], [24]. This provides two crucial advantages: (1) exact log-likelihood computation for any image $\mathbf{x}$ by multiplying conditional probabilities, enabling rigorous evaluation of $p(\mathbf{x}|\mathbf{d})$ and proper uncertainty quantification [25]; (2) statistically grounded training via maximum likelihood, ensuring the model can approximate the true distribution as training data increases [26].

Beyond autoregressive decoding, transformers offer conditioning flexibility through masked token modeling. A ViT trained to predict randomly masked tokens (like MaskGIT [22]) develops bidirectional predictive capability – it learns to fill missing parts by attending to surrounding pixels in all directions [27], [28]. This enables a single model to handle arbitrary conditioning patterns without architectural changes. Given observed pixels $\mathbf{x}_O$ at locations O, the transformer naturally generates $p(\mathbf{x}_U \mid \mathbf{x}_O, \mathbf{d})$ for unobserved locations $U = \{1, \ldots, N\} \setminus O$ by iteratively sampling masked tokens. Any geometry of observed vs. unobserved locations can be accommodated by the same model, greatly increasing flexibility for practical problems where data locations vary between cases [29], [30].

In this paper, we present a comprehensive empirical study of ViTs for conditional sampling in the situation with binary images. We focus on a test domain where the ground truth distribution is known analytically, allowing direct evaluation of the ViT sampler's statistical performance. Using a binary facies model with well-defined $p(\mathbf{x})$ and $p(\mathbf{x}|\mathbf{d})$ for which we can draw samples via specialized variable elimination algorithm (VEA) [31], we quantitatively assess how closely the transformer's learned distribution matches the ground truth. We report comparisons of sample statistics, spatial connectivity measures, and posterior log-likelihoods between ViT-generated and true samples, using cross-entropy as a rigorous measure of model accuracy. Our results examine whether ViTs suffer from mode collapse or estimation bias, how well they quantify uncertainty, and how performance scales with training data and model size. These findings suggest that ViTs can achieve fast, flexible conditional sampling without sacrificing statistical consistency, making them a promising direction for high-dimensional spatial modeling problems in geoscience and beyond.

In Section II we introduce the methodology for creating the VEA and ViT samples. In Section III we evaluate the ViT samples against the ground-truth VEA samples. Lastly, in Section IV we conclude the findings of this work and propose future research avenues.

II. Methodology

We propose a ViT framework for conditional sampling of binary spatial images, and compare its performance to a baseline using the VEA on a Markov random field. In this section, we detail the ViT architecture and training, the MRF model and VEA sampling method, and the conditional sampling procedure. Hyperparameters for both methods are summarized in Table I and Table II.

A. Vision Transformer Architecture

Our conditional sampler is a ViT tailored for discrete spatial data. Figure 1 illustrates a single forward pass of the model during both training and inference phases.

The architecture comprises an embedding layer followed by 2 Transformer encoder layers, each with 2 self-attention heads and 64-dimensional token embeddings. The final layer with softmax produces a categorical distribution on the next token; the vocabulary has $5,462$ distinct symbols, which amounts to approximately 8% of the possible 2^{16} patch tokens.

a) Training procedure.: During training, a 64×64 binary image is partitioned into a sequence of 4×4 patches, creating 256 patch tokens. A random subset of patches are masked by setting all pixel values to 0.5 (shown in gray in the figure). Each patch (both masked and unmasked) is linearly embedded into a 64-dimensional continuous vector space, producing a sequence of learnable token embeddings.

The sequence of embeddings is processed through 2 transformer encoder layers, each containing 2 self-attention heads. Each encoder layer consists of multi-headed self-attention with relative positional encoding added to attention scores, followed by add-and-norm, a feed-forward network, and a final add-and-norm layer. The transformer outputs a sequence of contextually-aware 64-dimensional representations where each patch embedding has attended to all other patches in the sequence.

A linear transformation layer maps the 64-dimensional embeddings to vocabulary logits: $\mathbb{R}^{256\times 64} \to \mathbb{R}^{256\times v}$, where $v = 5,462$ is the vocabulary size representing all possible 4×4 binary patch patterns. The logits represent unnormalized log-probabilities that can be converted to conditional probabilities via softmax.

b) Inference procedure.: The trained model performs conditional generation by autoregressively sampling patches one at a time from the learned conditional distributions, starting from a set of observed patches and iteratively filling in missing regions by sampling from $P_\theta(x_j \mid x_{\setminus \mathcal{M}})$ until the entire image is completed.

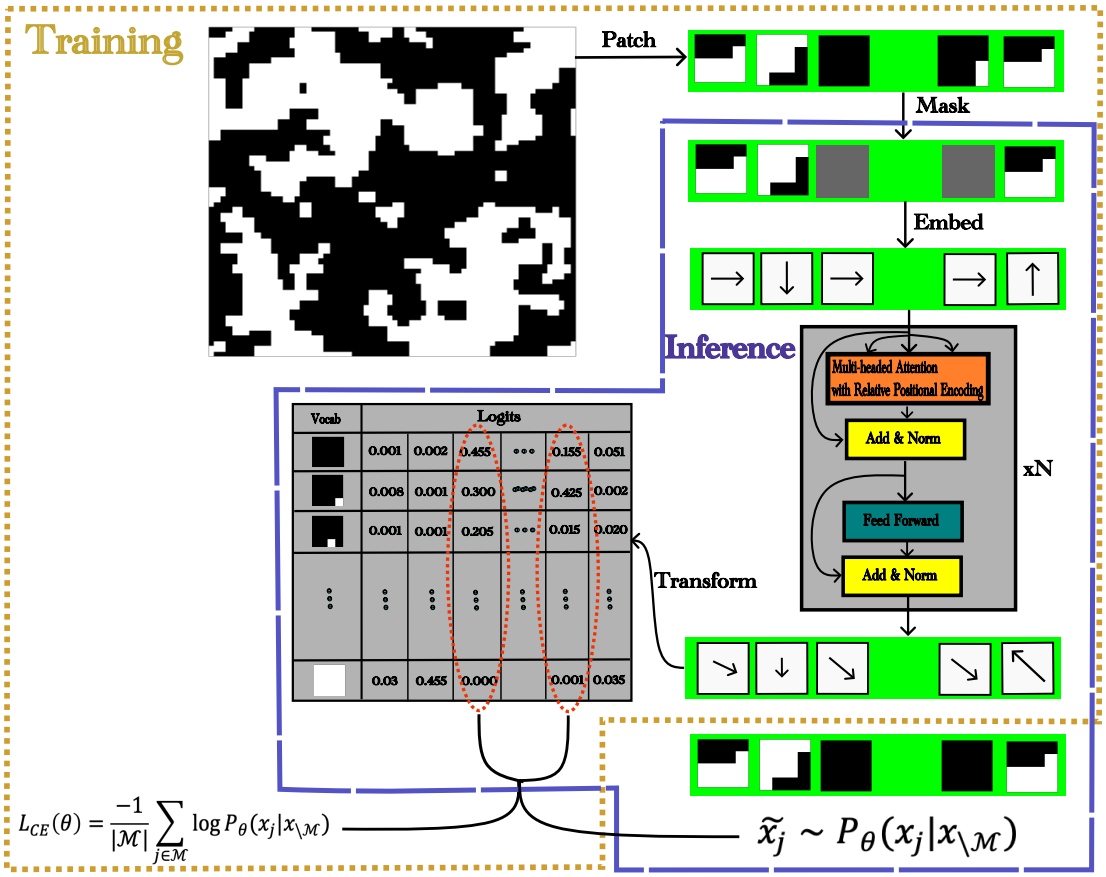


Fig. 1: Vision Transformer architecture for masked patch modeling. The model processes 64×64 binary images partitioned into 4×4 patches through embedding, transformer layers, and vocabulary projection to predict masked patches during training and generate complete images during inference.

c) Relative positional encoding in self-attention.: Instead of adding absolute position embeddings to the token vectors, we use a learned *relative positional bias* matrix $R \in \mathbb{R}^{N \times N}$ that is added inside each self-attention head [32]. With query, key, and value matrices $Q, K, V \in \mathbb{R}^{N \times d}$, the attention map is [33]

$$\text{Attention}(Q, K, V) = \text{softmax}\Big(\frac{QK^\mathsf{T} + R}{\sqrt{p}}\Big) V, \tag{1}$$

where p is the key dimension and R is shared across layers but learned during training. This formulation lets the model encode relative offsets (e.g. Manhattan distances) directly in its pairwise interactions without introducing absolute positional embeddings.

B. Vision Transformer Training

a) Training objective.: A 64×64 binary image is partitioned into a sequence of 4×4 patches, creating 256 patches with associated tokens. For each training image $\mathbf{x}$, we uniformly sample a *mask set* $\mathcal{M} \subseteq \{1, \ldots, 256\}$ and replace those tokens by a special learnable token. The ViT is optimized to reconstruct the true tokens at the masked positions, following the masked language modeling paradigm [34]. With model parameters θ, the loss is

$$L_{\text{CE}}(\theta) = -\frac{1}{|\mathcal{M}|} \sum_{j \in \mathcal{M}} \log P_\theta\big(\mathbf{x}_j \mid \mathbf{x}_{\setminus \mathcal{M}}\big), \tag{2}$$

where

- $\mathbf{x}_j$ is the ground-truth token at patch j;
- $\mathbf{x}_{\setminus\mathcal{M}}$ denotes the *context*, i.e. the unmasked tokens;
- $P_\theta(\cdot \mid \mathbf{x}_{\setminus\mathcal{M}})$ is the categorical distribution produced by the ViT for patch j given that context (obtained after softmax).

Minimizing (2) therefore maximises the conditional likelihood of the true token values given the visible context, training the network to approximate the true conditional distribution over all masking patterns.

b) Cyclic masking curriculum.: Training runs for 1000 epochs with batch size 100 and initial learning-rate 10^{-3} (Adam optimizer). Masking follows a 100-epoch cycle inspired by curriculum learning principles [35]: the masking rate starts at 1% in epoch 1 of a cycle and increases linearly to 99% by epoch 100, after which it resets. Early-cycle steps teach the model to refine nearly complete fields; late-cycle steps force it to predict large missing regions, ensuring robustness across conditioning densities.

After training, the ViT yields an approximation of the ground truth conditional distribution that we exploit for autoregressive sampling (Section II-D).

TABLE I: Vision Transformer architecture and training hyper-parameters.

Parameter	**Value**
Embedding dimension	64
Encoder layers	2
Attention heads / layer	2
Vocabulary size	5 462 tokens
Training epochs	1 000
Batch size	100
Optimizer	Adam
Initial learning rate	10^{-2}
Loss	Cross-entropy (Eq. 2)
Masking schedule	$1\% \rightarrow 99\%$ every 100 epochs (cyclic)
Mask selection	Uniform random per epoch
Relative position bias	Learned R in Eq. 1
Sampling temperatures	$\tau = 1.0,\ 0.9$ (Section II-D)

C. Markov random field Model and variable elimination algorithm

We employ a binary spatial pattern dataset for model training and evaluation. Each sample is a 64×64 0/1 field modeled as a binary MRF. Following Austad & Tjelmeland [31], the joint probability distribution is given by:

$$p(\mathbf{x}) \propto \exp\left(\sum_{\Lambda\in\mathcal{C}} v_\Lambda(\mathbf{x}_\Lambda)\right), \tag{3}$$

where $\mathcal{C}$ denotes the set of all maximal cliques, $\mathbf{x}_\Lambda$ represents the configuration of pixel outcomes in clique Λ, and v_Λ is the potential function for that clique.

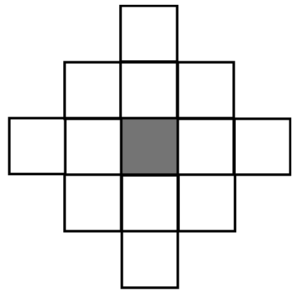


Fig. 2: Third-order neighborhood structure in Markov random field construction.

The MRF has two types of maximal cliques, corresponding to the third-order neighborhood structure shown in Figure 2:

1) the 2×2 *square*; first row Figure 3.
2) the *five-pixel star*; second row Figure 3.

a) Clique potentials.: Table II lists the potential parameters used throughout this study. The entries correspond to the ten equivalence classes of clique configurations shown in Figure 3.

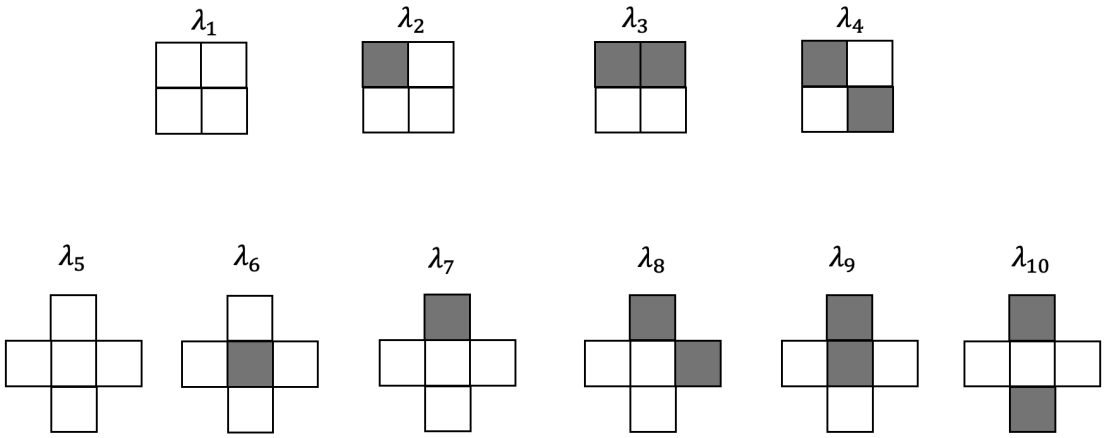


Fig. 3: Clique Potentials in the Markov random field model.

TABLE II: MRF clique-potential parameters used for exact sampling via VEA.

Clique type	**Configuration class**	$\boldsymbol{\lambda}$
2×2 square	Uniform (all 0 or all 1)	$\lambda_1 = 0.9$
	Noise (three of one colour, one of the other)	$\lambda_4 = -3.0$
	Horizontal / vertical border	$\lambda_2 = 0.4$
	Diagonal border	$\lambda_3 = 0.0$
5-pixel star	Uniform	$\lambda_5 = 0.0$
	Lone centre pixel	$\lambda_6 = -5.0$
	Edge pattern	$\lambda_7 = 0.0$
	Corner pattern	$\lambda_8 = 0.0$
	Dead-end (three spokes)	$\lambda_9 = -2.0$
	Straight line (two opposite spokes)	$\lambda_{10} = -2.0$

b) Variable Elimination Algorithm (VEA).: The VEA provides sampling with minuscule approximation error from the MRF distribution [31]. The algorithm operates in two phases:

Forward process: Starting from the joint distribution $p(\mathbf{x})$ in Eq. 3, variables are eliminated sequentially. At each step i, we marginalize out x_i to obtain:

$$p(x_2, \ldots, x_N) = \sum_{x_1} p(x_1, x_2, \ldots, x_N), \tag{4}$$

$$p(x_3, \ldots, x_N) = \sum_{x_2} p(x_2, x_3, \ldots, x_N), \tag{5}$$

$$\vdots$$

$$p(x_N) = \sum_{x_{N-1}} p(x_{N-1}, x_N). \tag{6}$$

This process constructs a sequence of uni-variate distributions, storing intermediate results needed for the backward pass.

Backward process: Sampling proceeds in reverse order. We begin by sampling $x_N \sim p(x_N)$. For each subsequent variable, we use the conditional distribution:

$$x_{i-1} \sim p(x_{i-1}|x_i) = \frac{p(x_{i-1}, x_i)}{p(x_i)}, \tag{7}$$

where x_i and $p(x_i)$ are known from the previous sampling step. This yields a complete realization $\mathbf{x}$ along with its likelihood $p(\mathbf{x})$ evaluation.

For conditional sampling, we organize the field such that observed variables $x_{N-n_O+1}, \ldots, x_N$ appear last in the elimination order. Rather than sampling these, we fix them to their observed values and begin the backward sampling from x_{N-n_O}, effectively sampling from $p(\mathbf{x}_U|\mathbf{x}_O)$.

The VEA with the parameters in Table II produces realizations that we use as ground truth for training (Section II-B) and for evaluation in Section III. Figure 4 shows unconditional realizations from the specified MRF model on the $64/times64$ grid.

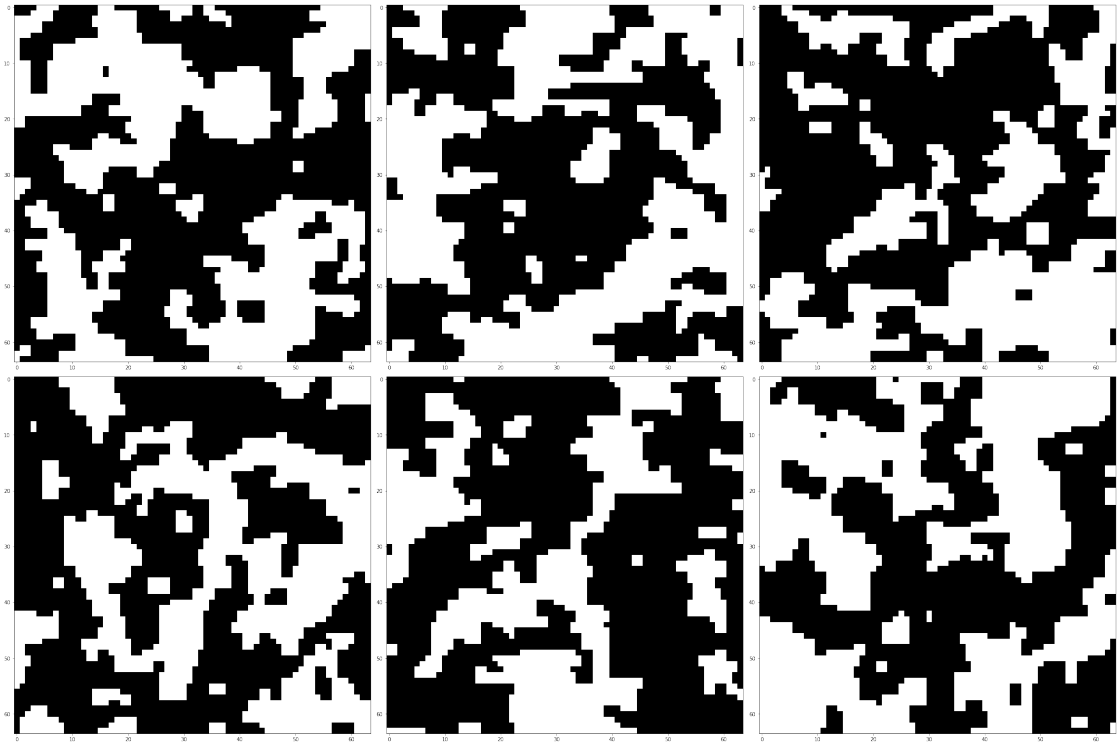
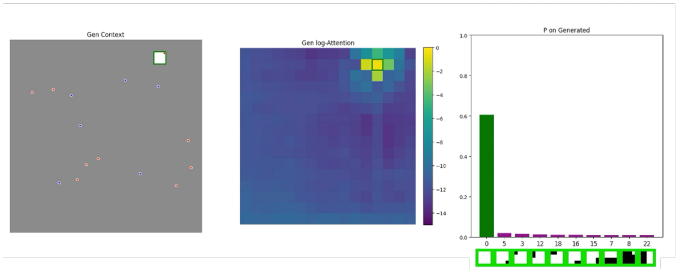
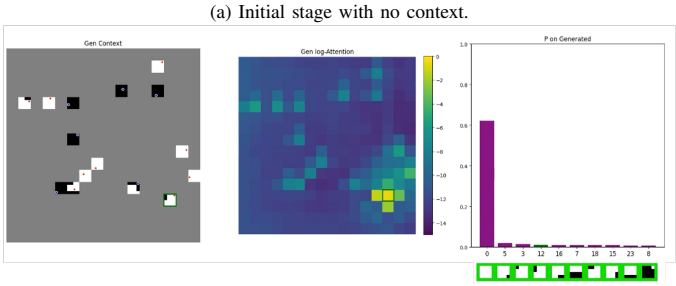


Fig. 4: Unconditional realizations of the MRF model, generated with VEA, used as training data.

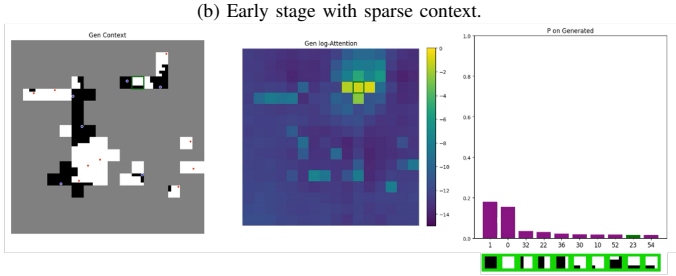
D. Vision Transformer Sampling



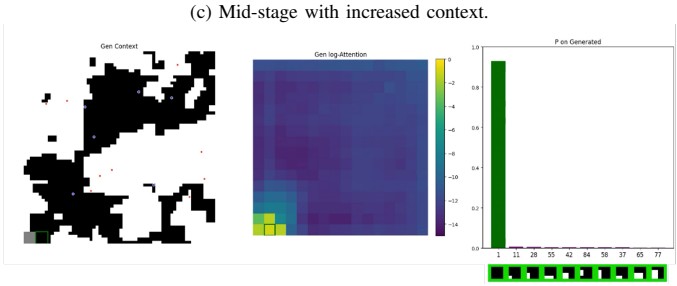
(a) Initial stage with no context.



(b) Early stage with sparse context.



(c) Mid-stage with increased context.



(d) Late stage with nearly complete context.

Fig. 5: Conditional ViT sampling process using inverse Manhattan ordering. Each row shows three panels: **Left:** Current image (conditioning context). Black = 0. White = 1. Gray = unsampled. Red = 1 observed. Blue = 0 observed. Green rectangles = current location. **Middle:** Log-attention scores ($\mathbf{QK}^T + \mathbf{R}$). **Right:** Probability of 10 most probable patches.

The trained ViT operates on image patches of 4×4 pixels (256 tokens for a 64×64 image). In all experiments we *condition on 15* ***pixels***, not on whole patches. Let $\mathcal{O} = \{p_1, \ldots, p_{15}\}$ be the observed pixel indices and $\boldsymbol{x}_\mathcal{O}$ their fixed binary values. Each observed pixel lies inside exactly one patch; denote the set of those *anchor patches* by $\mathcal{P}_\mathcal{O}$.

a) Constraint handling.: Whenever a patch t is predicted, the ViT returns logits $\ell_t \in \mathbb{R}^{|\mathcal{V}|}$ for all vocabulary tokens. If $t \in \mathcal{P}_\mathcal{O}$ we must enforce that the sampled token respects the observed pixel in its spatial slot. We therefore truncate the

distribution:

$$\tilde{\ell}_{t,k} = \begin{cases} -\infty & \text{if token } k \text{ disagrees with } \boldsymbol{x}_{\mathcal{O}}\,, \\ \ell_{t,k} & \text{otherwise,} \end{cases} \tag{8}$$

followed by renormalisation $P(k \mid \text{context}) = \dfrac{\exp(\tilde{\ell}_{t,k})}{\sum_{k'} \exp(\tilde{\ell}_{t,k'})}$. For non-anchor patches $t \notin \mathcal{P}_{\mathcal{O}}$ the logits are unaltered.

b) Sampling order.: Let $d_{\mathrm{M}}(t, \mathcal{P}_{\mathcal{O}}) = \sum_{s \in \mathcal{P}_{\mathcal{O}}} \mathrm{dist}_{\mathrm{M}}(t, s)$ be the *sum* of Manhattan (L_1) distances from an unobserved patch t to all anchor patches. Define also its inverse-score $d_{\mathrm{IM}}(t, \mathcal{P}_{\mathcal{O}}) = \sum_{s \in \mathcal{P}_{\mathcal{O}}} \big[\mathrm{dist}_{\mathrm{M}}(t, s)\big]^{-1}$ (larger when t is near the anchors).

1) **Manhattan-ascending sampling**
   a) Sample all anchor patches $\mathcal{P}_{\mathcal{O}}$ first (with truncation above).
   b) For every remaining patch t, compute $d_{\mathrm{M}}(t, \mathcal{P}_{\mathcal{O}})$.
   c) Visit the unobserved patches in *ascending* order of d_{M} (geometric centre–outward) and sample each exactly once.
2) **Inverse-Manhattan descending sampling**
   a) Sample the anchor patches $\mathcal{P}_{\mathcal{O}}$ first (with truncation).
   b) Compute $d_{\mathrm{IM}}(t, \mathcal{P}_{\mathcal{O}})$ for every other patch.
   c) Visit patches in *descending* order of d_{IM}—hence neighbours of the anchors are filled first, progressing outward.

The four-stage process visualised in Figure 5 mirrors the mathematical description above; panel (a) shows step 1 (anchor-patch inference) and panels (b)–(d) illustrate successive iterations under the inverse-Manhattan schedule.

c) Temperature.: During token sampling we apply a softmax temperature $\tau \in \{1.0, 0.9\}$ to the logits: $P_\tau(k) = \mathrm{softmax}(\ell_k/\tau)$. Lower temperature (0.9) sharpens the distribution, yielding lower-entropy, more deterministic patches; $\tau = 1.0$ preserves the model's raw uncertainty.

d) Complete procedure.: For each of the four *(ordering, temperature)* pairs we generate 1 000 conditional realisations. Anchor patches are always sampled first with their logits truncated to respect the 15 observed pixels, and every remaining patch is sampled exactly once in the sequence determined above—no further refinements are made. This yields fully populated 64×64 binary fields consistent with the conditioning, which are subsequently analysed against the VEA baseline.

E. Experiment Setup

We evaluate the performance of the ViT sampler against the VEA baseline by generating 1,000 conditional samples from each method, all conditioned on the same set of 15 observed pixel locations to ensure a consistent basis for comparison. We consider five different sampling strategies:

- **VEA (Reference)** – the baseline conditional sampler using the variable elimination algorithm [31] (ground-truth distribution);
- **ViT-Manhattan** ($\tau = 1.0$) – Vision Transformer sampler using Manhattan-distance fill ordering and temperature $\tau = 1.0$;
- **ViT-Inverse** ($\tau = 1.0$) – Vision Transformer sampler using inverse-Manhattan ordering, $\tau = 1.0$;
- **ViT-Manhattan** ($\tau = 0.9$) – Vision Transformer with Manhattan ordering, sampling with temperature $\tau = 0.9$;
- **ViT-Inverse** ($\tau = 0.9$) – Vision Transformer with inverse-Manhattan ordering, temperature $\tau = 0.9$.

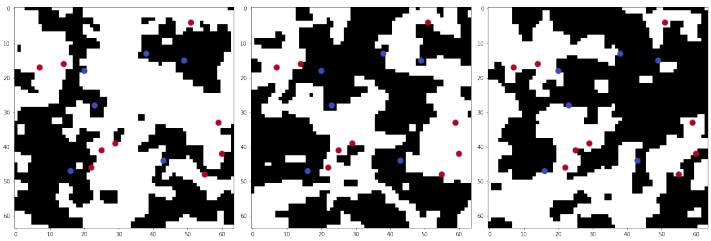
The generated samples are analyzed using a comprehensive suite of evaluation metrics that capture different aspects of spatial distribution accuracy, including pointwise structural similarity (PointSSIM) [36], marginal distributions (JS divergence), sample log-likelihoods, rank alignment, two-point statistics (transiograms), and third-order spatial cumulants. These metrics, detailed in the Results section, provide a thorough assessment of how closely the ViT-generated patterns match the true MRF distribution as represented by the exact VEA baseline.

III. Results

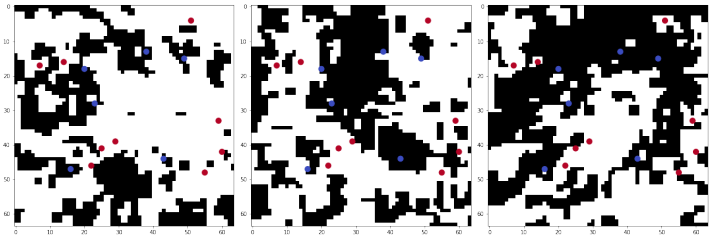
We evaluate the Vision Transformer-based conditional samplers against the exact Variable Elimination Algorithm (VEA) baseline to understand how closely the learned model reproduces the true Markov random field (MRF) distribution. Our ViT models were tested with two sequential pixel-filling orderings – the standard Manhattan-distance order and the inverse Manhattan order – at sampling temperature $\tau = 1.0$ (with additional experiments at $\tau = 0.9$). In the following, we present key findings covering visual sample comparisons, probabilistic fidelity, and structural statistics.

A. Visual Sample Quality

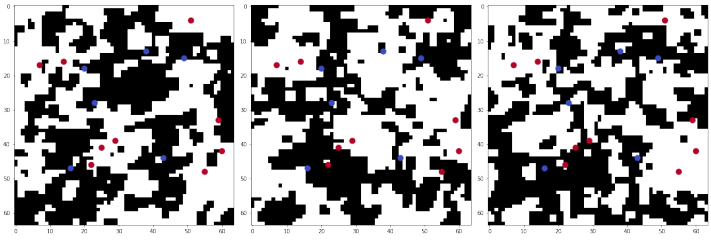
Figure 6 shows five conditional samples from each model (MRF baseline and ViT variants), all conditioned on the same 15 observed pixels (blue dots for 0-values, red dots for 1-values). Visual inspection reveals that ViT samples closely resemble the MRF samples in overall structure and honor all conditioning constraints. Notably, the temperature-adjusted samples ($\tau = 0.9$) appear smoother and more closely match the MRF's visual characteristics, suggesting that temperature tuning helps capture the true distribution's smoothness properties.



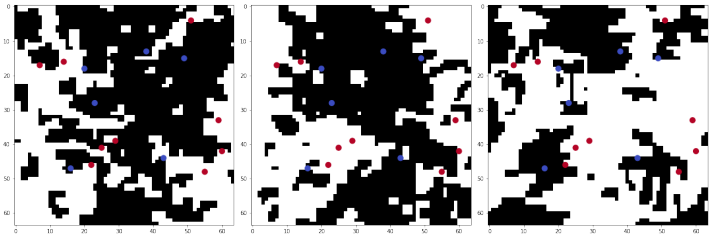
(a) Three conditional realizations from exact MRF sampler (ground truth)



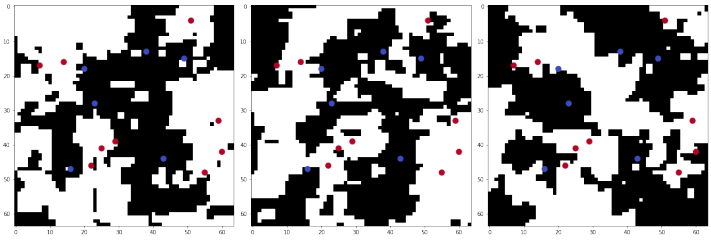
(b) Three conditional realizations from ViT Manhattan ordering



(c) Three conditional realizations from ViT Inverse Manhattan ordering



(d) Three conditional realizations from ViT Manhattan with $\tau = 0.9$



(e) Three conditional realizations from ViT Inverse Manhattan with $\tau = 0.9$

Fig. 6: Conditional realizations from each sampling method. All samples share the same 15 conditioning pixels (blue: 0-values, red: 1-values).

B. Sequential Sampling Dynamics

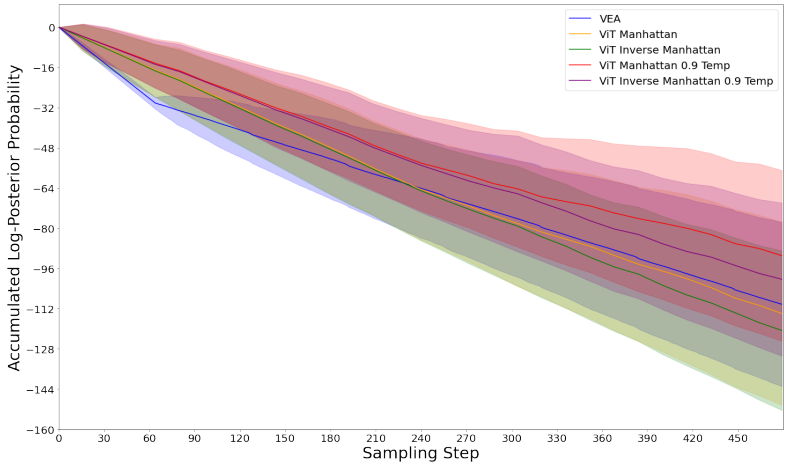


Fig. 7: Accumulated log-posterior probability trajectories for the first 500 sampling steps. Lines show mean trajectories; shaded regions indicate ±2 standard deviations. MRF shows steeper initial descent with lower variance compared to ViT methods.

Figure 7 reveals fundamental differences in how the methods build up probability during sequential sampling. The MRF trajectory exhibits a steep initial descent with minimal variance, reflecting low-probability assignments when sampling with limited context (upper-left corner start). After approximately 90 samples, the curve flattens and variance increases as more observations provide context, though individual low-probability choices can create ripple effects.

In contrast, ViT trajectories appear nearly linear due to their sampling strategy: starting near observed points provides immediate confidence, and the method maintains proximity to newly sampled patches throughout. Temperature-adjusted trajectories ($\tau = 0.9$) consistently remain above $\tau = 1.0$ curves, as expected from their more peaked distributions.

C. Marginal Probability Maps and Calibration

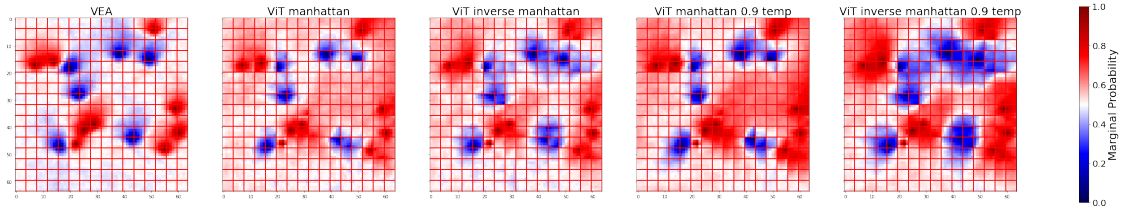


Fig. 8: Cell-wise marginal probabilities for each model. MRF (ground truth) shows well-defined probability structure around conditioning points. ViT models exhibit longer-range dependencies and spatial biases.

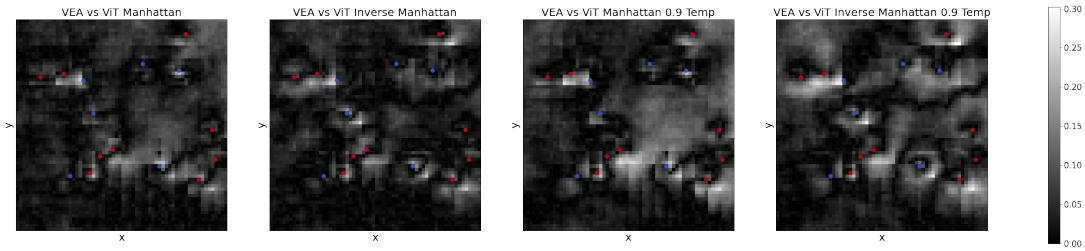


Fig. 9: JS divergence between MRF and ViT probability maps. High divergence (red) indicates regions where models disagree most strongly, particularly in boundary zones.

Figures 8 and 9 reveal fundamental differences in how the models assign pixel-level probabilities. The MRF exhibits well-localized probability fields around conditioning points with sharp transitions between high and low probability regions. In contrast, the ViT models show extended spatial dependencies that propagate far beyond the immediate neighborhood of observed pixels. This manifests as systematic biases where the ViT tends to assign intermediate probabilities (closer to 0.5) in regions where the MRF is highly confident (near 0 or 1).

The Jensen-Shannon (JS) divergence map (Figure 9) quantifies these differences, with the highest divergences occurring precisely at the boundary zones between different probability regimes. This pattern suggests the ViT has learned a smoother, more diffuse representation of the conditional distribution, potentially due to the transformer's tendency to aggregate information across broad spatial contexts. Temperature adjustment exacerbates these biases: while $\tau = 0.9$ produces visually more coherent samples, it amplifies the probability distortions, creating even stronger spatial biases in the marginal distributions.

D. Sample Probability Distributions

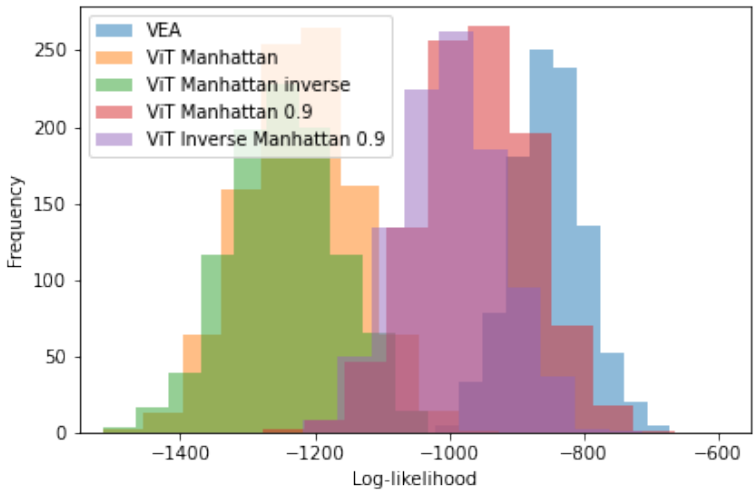


Fig. 10: Log-posterior probability distributions evaluated under the MRF model. MRF samples show highest log-posteriors; temperature-adjusted ViT samples overlap MRF distribution better than $\tau = 1.0$ samples.

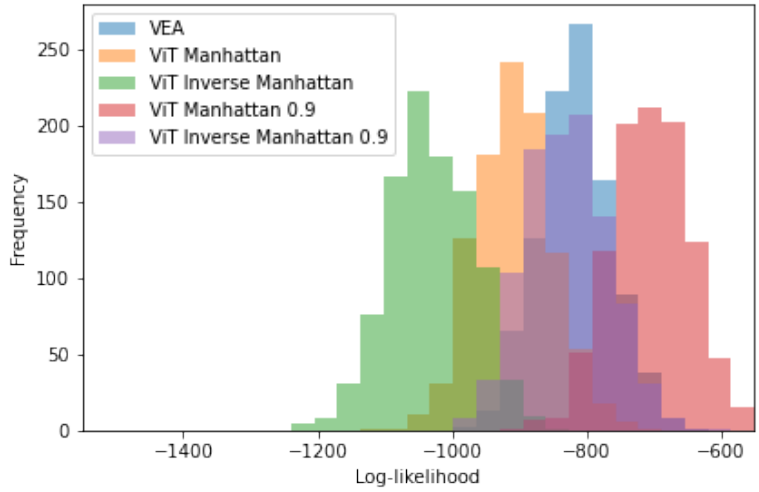


Fig. 11: Log-posterior probability distributions evaluated under ViT Manhattan model. ViT consistently assigns higher probabilities to its own samples compared to MRF samples.

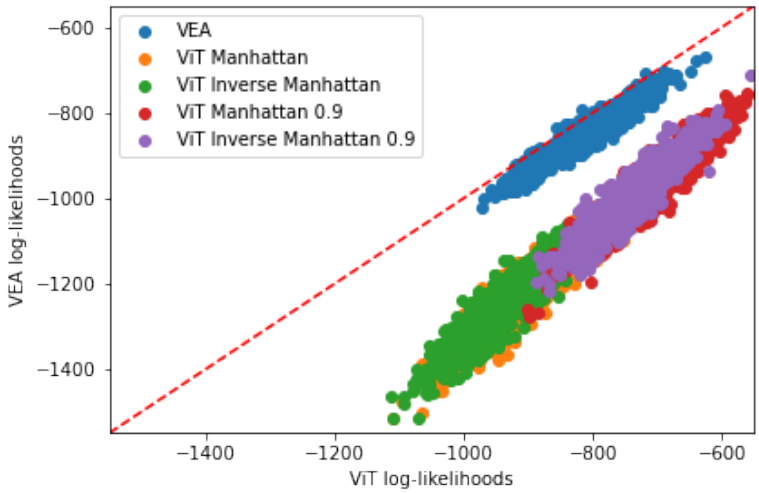


Fig. 12: Scatter plot comparing log-posterior evaluations: ViT model (x-axis) vs MRF model (y-axis). Diagonal indicates perfect agreement. MRF samples (blue) align with diagonal; ViT samples deviate, indicating evaluation disagreement.

The analysis of sample-level log-probabilities provides crucial insights into how well the ViT captures the true distribution's support and probability mass allocation. Figure 10 shows the distribution of log-posterior values when samples from each method are evaluated under the ground-truth MRF model. The MRF's own samples (blue) exhibit a distribution centered around -800, reflecting the natural variability in the true conditional distribution.

The ViT-generated samples reveal a critical limitation: while they maintain similar variance to the MRF distribution, they are systematically shifted toward lower probabilities. The non-temperature ViT samples ($\tau = 1.0$) form distributions centered around -1300 to -1200, showing minimal overlap with the MRF baseline. This indicates that the ViT fails to generate the high-probability configurations that characterize the true distribution. The temperature-adjusted samples ($\tau = 0.9$) partially address this issue, shifting closer to the MRF distribution (centered around -1000 to -900) and achieving better overlap, though still falling short of matching the MRF's high-probability modes.

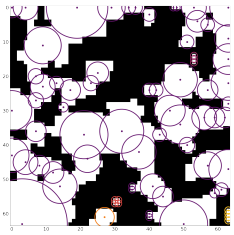
When we reverse the evaluation perspective (Figure 11), using the ViT itself as the evaluator, all distributions converge to a much narrower range around -1100 to -700. Notably, the ViT assigns the highest probabilities to its temperature-adjusted samples, intermediate probabilities to MRF samples, and lowest to its own non-temperature samples. This reveals that while the ViT maintains consistent relative rankings across different evaluation methods, it systematically overestimates the probability of its own samples when compared to ground truth. Importantly, both evaluators assign similar probabilities to the MRF samples, confirming the consistency of the ground-truth distribution.

The scatter plot (Figure 12) provides a sample-by-sample comparison that crystallizes these findings. Perfect agreement between models would place all points on the diagonal line. While MRF samples (blue) cluster close to this diagonal – confirming consistency between evaluation methods – the ViT samples form a dispersed cloud with systematic deviations. The predominance of points below the diagonal confirms that the ViT overvalues its own generations relative to their true probability.

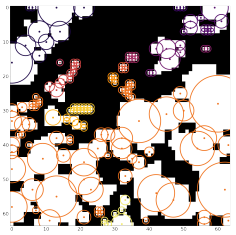
This probability miscalibration has important implications for uncertainty quantification. While the ViT successfully captures the variance of the true distribution, its systematic shift toward lower-probability samples means it misses the most likely configurations of the conditional distribution. The model generates a diverse but suboptimal set of patterns, exploring the correct breadth of possibilities but centered on the wrong region of probability space.

E. Structural Analysis

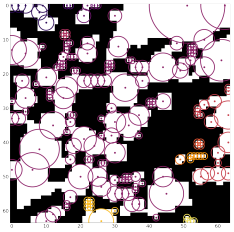
1) PointSSIM: PointSSIM is a resolution-invariant image comparison metric that converts binary images into sparse anchor point representations [36]. Anchor points are extracted as locally adaptive maxima from the minimal distance transform, with each point marked by its radius to the nearest object boundary and object label. From these anchor points, four structural features are computed: anchor count (number of points), area coverage (total area spanned by anchor radii), anchor points per object (structural heterogeneity), and spatial variance irregularity (clustering measure). The PointSSIM metric combines these four features to provide robust, rotation-invariant image comparison that outperforms traditional pixel-based metrics like SSIM, MSE, and MS-SSIM in distinguishing between different structural patterns while maintaining high within-class consistency.



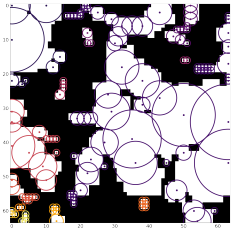
(a) MRF sample with PointSSIM anchor points



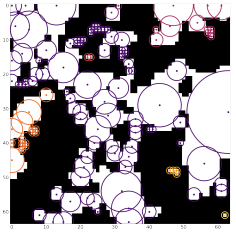
(b) ViT Manhattan sample



(c) ViT Inverse Manhattan sample



(d) ViT Manhattan $\tau = 0.9$ sample



(e) ViT Inverse Manhattan $\tau = 0.9$ sample

Fig. 13: PointSSIM structural analysis showing anchor point locations, radii, and labels. Non-temperature ViT samples show higher anchor point density with smaller radii; temperature samples more closely match MRF structure.

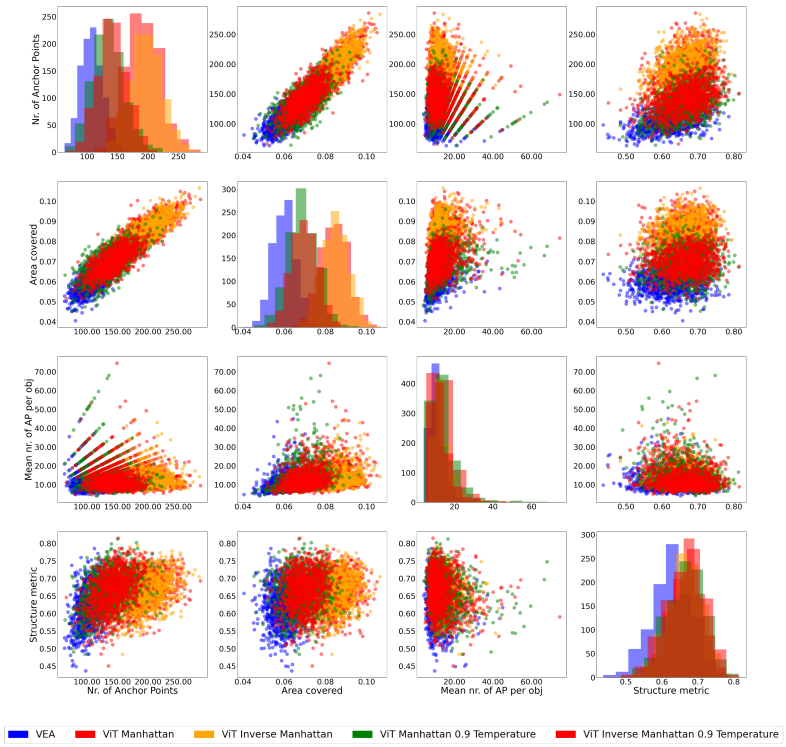


Fig. 14: PointSSIM feature distributions: anchor point count, area coverage, mean anchors per object, and structure metric. ViT samples show higher area coverage (more 1-values) and less smooth objects than MRF.

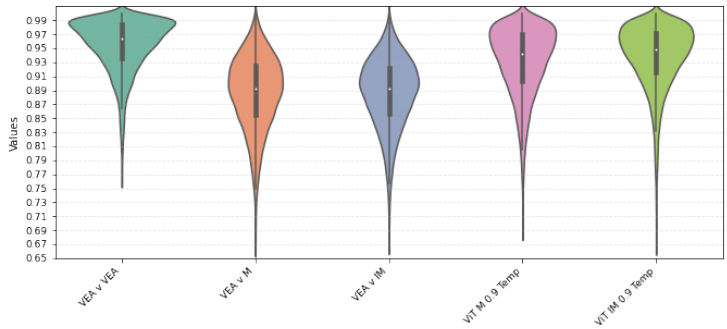


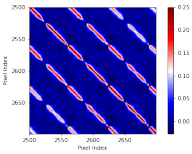
Fig. 15: PointSSIM similarity scores between samples. Temperature-adjusted ViT samples achieve higher structural similarity to MRF samples on average.

The structural analysis through PointSSIM metrics reveals how the probability biases identified earlier manifest as tangible differences in image structure. Figure 13 visualizes these differences through anchor points that capture key structural elements. The MRF sample (panel a) shows a balanced distribution of anchor points with moderate radii, representing well-formed structural elements. In contrast, non-temperature ViT samples (panels b,c) exhibit a proliferation of clustered anchor points with tiny radii, indicating fragmented, irregular structures. Temperature adjustment (panels d,e) significantly improves structural coherence, producing anchor patterns more similar to the MRF baseline.

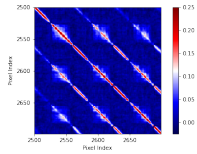
The quantitative analysis in Figure 14 confirms these visual observations across multiple structural dimensions. The anchor point count distribution shows ViT samples, particularly without temperature adjustment, generate significantly more anchor points than MRF samples. This excess indicates over-segmentation – the ViT creates many small, disconnected components rather than the larger, coherent structures of the true distribution. The area coverage metric reveals another manifestation of the white-pixel bias: ViT samples consistently cover more area (higher fraction of 1-values), with temperature adjustment amplifying this effect.

The mean anchor points per object metric provides insight into structural smoothness. MRF samples show the lowest values, indicating smooth, regular objects. Non-temperature ViT samples have the highest values, reflecting rough, irregular boundaries with many protrusions. The cross-plot between total anchor points and mean per object reveals distinct diagonal stripes, suggesting that as objects become more irregular, they require proportionally more anchor points to represent their complexity. Finally, the structure metric shows ViT samples tend toward higher values (more clustered anchor points) compared to the MRF's more random spatial distribution, likely due to the proliferation of small, clustered irregular features.

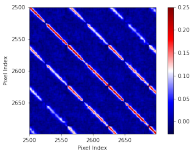
Figure 15 synthesizes these structural differences into overall similarity scores. Temperature-adjusted samples achieve significantly higher similarity to MRF samples, confirming that temperature helps recover appropriate structural properties. However, even the best ViT samples show a tail of low similarity scores, indicating persistent structural anomalies in a subset of generated samples.



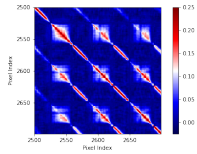
(a) MRF: symmetric, well-structured covariance



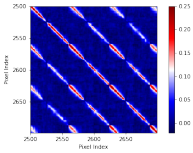
(b) ViT Manhattan: biased with extended correlations



(c) ViT Inverse Manhattan: reduced bias



(d) ViT Manhattan $\tau = 0.9$: amplified bias



(e) ViT Inverse Manhattan $\tau = 0.9$: most similar to MRF

Fig. 16: Empirical covariance matrices (zoomed near conditioning point). MRF shows symmetric structure; ViT variants exhibit varying degrees of bias and correlation length distortion.

The covariance structure analysis (Figure 16) provides deeper insight into how the models capture spatial dependencies. These matrices are zoomed in near a conditioning point to reveal local correlation behavior. The MRF covariance (panel a) exhibits a symmetric, well-structured pattern reflecting the true spatial correlations induced by the clique potentials. The correlation strength decays smoothly with distance, creating concentric patterns around the conditioning point. The ViT variants show markedly different behaviors that reveal a fundamental misunderstanding of correlation structure. Without temperature adjustment, the models exhibit a problematic dual nature: excessively long correlations around conditioning points (creating heavy biases) while maintaining correlations that are too short in the rest of the field. Manhattan ordering (panel b) introduces strong directional biases with extended correlations around the conditioning point that propagate preferentially along certain axes. This reflects the sequential nature of the sampling process and the transformer's tendency to overweight conditioning information. Inverse Manhattan ordering (panel c) reduces but does not eliminate these biases, showing more isotropic but still distorted correlation patterns. Temperature adjustment ($\tau = 0.9$) attempts to address the short-range correlation issue but dramatically amplifies the conditioning bias. The Manhattan model with temperature (panel d) shows severe bias around the conditioning point with correlation strengths that extend far beyond the MRF's natural range, creating artificial long-range dependencies. While this increases correlation lengths throughout the field (addressing the undersmoothing), it exacerbates the overconditioning problem.

Interestingly, the Inverse Manhattan with temperature (panel e) achieves the best balance. The correlation lengths throughout the field appear similar to the ground truth, while the conditioning bias, though still present, is less severe than in the Manhattan variant. This suggests that the combination of inverse ordering and temperature adjustment partially compensates for the transformer's inherent tendency to overcondition on observed data while undersmoothing elsewhere. This finding aligns with the earlier observation that Inverse Manhattan with temperature produces the most structurally faithful samples.

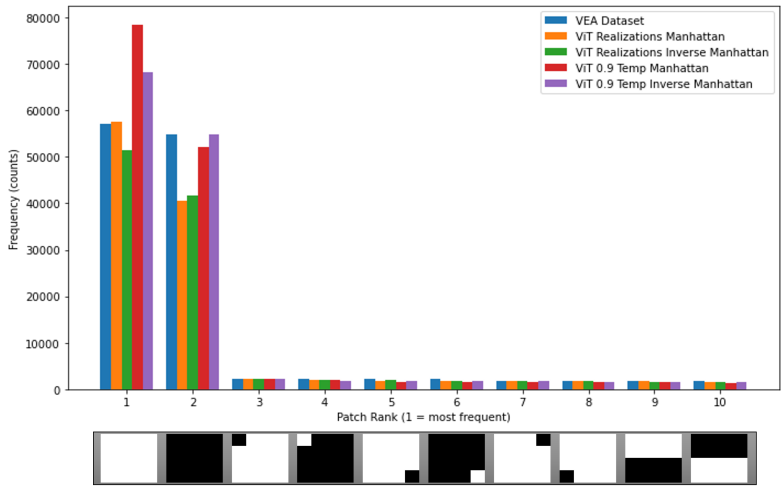


Fig. 17: Frequency of 4×4 patterns (token size). Temperature samples oversample homogeneous white patterns; non-temperature undersample homogeneous patterns. Both exhibit white bias.

Figure 17 provides a critical insight into the mechanism behind the ViT's biases. Since the transformer operates on 4×4 tokens, these pattern frequencies directly reflect the model's token-level probability assignments. The analysis reveals a fundamental trade-off: non-temperature sampling underestimates homogeneous patterns (both all-white and all-black), creating excess texture and fragmentation. Temperature adjustment overcorrects, dramatically oversampling homogeneous white patterns while still underrepresenting black ones.

This asymmetry explains many of the observed phenomena. The white bias manifests directly in token probabilities, propagating through the sequential generation process to create the extended spatial biases seen in marginal probabilities and covariance structures. The inability to correctly balance homogeneous versus textured patterns at the token level fundamentally limits the model's capacity to reproduce the true distribution's statistical properties. Temperature adjustment partially mitigates texture issues but exacerbates the white bias, suggesting that simple temperature scaling cannot fully address the underlying probability miscalibration.

F. Summary of Findings

Our comprehensive evaluation reveals both the potential and limitations of Vision Transformers for conditional image sampling.

Strengths:

- ViT models successfully capture large-scale spatial structures and respect all conditioning constraints
- Temperature adjustment ($\tau = 0.9$) helps address correlation length issues in unconditioned regions
- The Inverse Manhattan ordering with temperature achieves the best balance between correlation structure and conditioning bias
- Computational efficiency and flexibility advantages over exact methods remain significant

Key Limitations:

- *Systematic probability biases*: ViT models exhibit persistent bias toward white pixels (1-values), manifesting in marginal probabilities, covariance structures, and token frequencies
- *Dual correlation failure*: The transformer simultaneously overconditions around observed pixels (creating excessive long-range correlations) while undersmoothing elsewhere (correlations too short in unconditioned regions)
- *Probability shift*: While maintaining appropriate variance, ViT samples are systematically shifted toward lower-probability regions, missing the high-probability modes of the true distribution
- *Token-level miscalibration*: The 4×4 pattern analysis reveals biases originate at the token level, with temperature adjustment creating a trade-off between proper correlation lengths and amplified conditioning bias

Implications: The transformer's learned representation reveals a fundamental misunderstanding of how conditioning information should influence spatial correlations. The model overweights the importance of observed pixels, creating artificial long-range dependencies around conditioning points, while failing to maintain proper correlation structures in unconditioned regions. Temperature adjustment partially addresses the correlation length issue but exacerbates the conditioning bias, and no

parameter setting successfully balances both aspects. For applications requiring accurate spatial statistics or proper uncertainty quantification, these correlation structure failures represent a critical limitation that must be weighed against computational advantages.

IV. Conclusion

This study demonstrates that Vision Transformers provide a remarkably general framework for conditional image sampling, offering explicit posterior probabilities and the flexibility to handle arbitrary conditioning geometries without architectural modifications. The ability to compute exact likelihoods for any configuration enables rigorous statistical analysis—a critical advantage over black-box generative models where such evaluation would be impossible.

Our comprehensive evaluation reveals that ViT models can reproduce the visual characteristics of MRF samples with reasonable fidelity, successfully capturing large-scale structures and respecting conditioning constraints. However, this visual similarity masks fundamental statistical limitations. The models exhibit systematic biases, most notably an overrepresentation of white pixels that originates at the token level and propagates through the generation process. While temperature adjustment can improve structural coherence, it exacerbates these biases, revealing a fundamental trade-off: we can optimize for either structural accuracy or unbiased probabilities, but not both simultaneously.

The analysis uncovered that ViT models learn a contracted, smoothed approximation of the true distribution. This limitation appears intrinsic to the current architecture and training approach, though it remains an open question whether substantially more compute, training data, or model parameters might overcome these biases.

Perhaps the most important finding is the critical role of thorough statistical evaluation. Temperature tuning and other modifications can make generated samples appear visually compelling, potentially masking severe distributional biases. Without comprehensive analysis across multiple statistical dimensions—marginal probabilities, covariance structures, pattern frequencies, and higher-order statistics—these biases would remain hidden, leading to false confidence in the model's fidelity. The Vision Transformer's amenability to such analysis, through its explicit probability computations, makes it particularly valuable for understanding these trade-offs.

Looking forward, these findings suggest several research directions. Addressing token-level biases through modified training objectives or architectural changes could improve distributional fidelity. Hybrid approaches combining the flexibility of transformers with the statistical rigor of classical methods like MRF may offer a path toward models that are both efficient and accurate. Most importantly, this work emphasizes that visual quality alone is insufficient for validating generative models in scientific applications—rigorous statistical analysis must be an integral part of model evaluation and deployment. While Vision Transformers offer compelling advantages in efficiency and flexibility, practitioners must carefully weigh these benefits against the statistical limitations when uncertainty quantification and distributional accuracy are paramount.

Acknowledgments

Thanks to the Norwegian Research Council for funding the GEOPARD project (319951) and the SFI Centre for Geophysical Forecasting (309960).

References

[1] C. Sun, V. Demyanov, and D. Arnold, "A conditional gan-based approach to build 3d facies models sequentially upwards," *Computers & Geosciences*, vol. 181, p. 105460, 2023.
[2] C. Jäggli, J. Straubhaar, and P. Renard, "Parallelized adaptive importance sampling for solving inverse problems," *Frontiers in Earth Science*, vol. 6, 2018.
[3] D. S. Oliver and Y. Chen, "Recent progress on reservoir history matching: A review," *Computational Geosciences*, vol. 15, no. 1, pp. 185–221, 2011.
[4] N. Linde, J. A. Vrugt, and A. B. Vrugt, "Challenges and strategies in geophysical inversion of categorical variables," *Near Surface Geophysics*, vol. 13, no. 3, pp. 183–189, 2015.
[5] E. Laloy, R. Hérault, D. Jacques, and Y. Robin, "Training-image based geostatistical inversion using a spatial gan," *Water Resources Research*, vol. 54, no. 1, pp. 381–406, 2018.
[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 27, 2014, pp. 2672–2680.
[7] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations*, 2014.
[8] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6840–6851.
[9] O. Ovanger, D. Lee, J. Eidsvik *et al.*, "A statistical study of latent diffusion models for geological facies modeling," *Mathematical Geosciences*, 2025.
[10] D. Lee, O. Ovanger, J. Eidsvik, E. Aune, J. Skauvold, and R. Hauge, "Latent diffusion model for conditional reservoir facies generation," *Computers & Geosciences*, vol. 194, p. 105750, 2025.
[11] J. Zhao and S. Chen, "Facies conditional simulation based on a vae–gan model and image quilting algorithm," *Journal of Applied Geophysics*, vol. 219, p. 105239, 2023.
[12] S. Mohamed and B. Lakshminarayanan, "Learning in implicit generative models," in *arXiv preprint arXiv:1610.03483*, 2016.
[13] S. Zhao, J. Song, and S. Ermon, "Towards deeper understanding of variational autoencoding models," in *arXiv preprint arXiv:1702.08658*, 2017.
[14] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
[15] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021.

[16] D. P. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 696–21 707, 2021.

[17] K. Cranmer, J. Brehmer, and G. Louppe, "The frontier of simulation-based inference," *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30 055–30 062, 2020.

[18] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[19] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[20] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.

[21] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi, "Palette: Image-to-image diffusion models," in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022, pp. 1–10.

[22] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, "Maskgit: Masked generative image transformer," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 315–11 325.

[23] M. Chen, A. Radford, R. Child *et al.*, "Generative pretraining from pixels," in *International Conference on Machine Learning*, vol. 119, 2020, pp. 1691–1703.

[24] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 873–12 883.

[25] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications," in *International Conference on Learning Representations*, 2017.

[26] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *International Conference on Machine Learning*. PMLR, 2016, pp. 1747–1756.

[27] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

[28] H. Bao, L. Dong, S. Piao, and F. Wei, "Beit: Bert pre-training of image transformers," in *International Conference on Learning Representations*, 2022.

[29] J. Yu, Y. Xu, J. Y. Koh, T. Luong, G. Baid, Z. Wang, V. Vasudevan, A. Ku, Y. Yang, B. K. Ayan *et al.*, "Scaling autoregressive models for content-rich text-to-image generation," *Transactions on Machine Learning Research*, 2023.

[30] T. Li, H. Chang, S. K. Mishra, H. Zhang, D. Katabi, and D. Krishnan, "Mage: Masked generative encoder to unify representation learning and image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2142–2152.

[31] H. M. Austad and H. Tjelmeland, "Approximate computations for binary markov random fields," *Statistics and Computing*, vol. 27, no. 5, pp. 1271–1292, 2017.

[32] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," 2018. [Online]. Available: https://arxiv.org/abs/1803.02155

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[34] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[35] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 41–48. [Online]. Available: https://doi.org/10.1145/1553374.1553380

[36] O. Ovanger, R. Hauge, J. Skauvold, M. J. Pyrcz, and J. Eidsvik, "Pointssim: A novel low dimensional resolution invariant image-to-image comparison metric," 2025. [Online]. Available: https://arxiv.org/abs/2506.23833

PointSSIM: A novel low dimensional resolution invariant image-to-image comparison metric

Oscar Ovanger, Ragnar Hauge, Jacob Skauvold, Michael J. Pyrcz and Jo Eidsvik

Under review (IEEE Transactions on Image Processing); preprint: arXiv:2506.23833

PointSSIM: A novel low dimensional resolution invariant image-to-image comparison metric

Oscar Ovanger, Ragnar Hauge, Jacob Skauvold, Michael J. Pyrcz, Jo Eidsvik

This paper presents PointSSIM, a novel low-dimensional image-to-image comparison metric that is resolution invariant. Drawing inspiration from the structural similarity index measure and mathematical morphology, PointSSIM enables robust comparison across binary images of varying resolutions by transforming them into marked point pattern representations. The key features of the image, referred to as anchor points, are extracted from binary images by identifying locally adaptive maxima from the minimal distance transform. Image comparisons are then performed using a summary vector, capturing intensity, connectivity, complexity, and structural attributes. Results show that this approach provides an efficient and reliable method for image comparison, particularly suited to applications requiring structural analysis across different resolutions.

***Index Terms*—Image comparison metric, Mathematical Morphology, SSIM.**

I. INTRODUCTION

Image comparison is a fundamental task in fields such as computer vision (Lowe, 2004; Szelisk, 2020), medical imaging (Litjens et al., 2017), and geostatistics (Pyrcz & Deutsch, 2014). Accurate and efficient image comparison metrics are essential for applications like image registration, quality assessment, and structural analysis (Wang & Bovik, 2009).

The most straightforward way of comparing images is pixel-to-pixel correspondence, such as Mean Squared Error (MSE) (Maindonald, 2007),

$$MSE(x,y) = \frac{1}{n_{rc}}\sum_{i=1}^{n_{rc}}(x_i - y_i)^2$$

In this case, x and y are images with n_{rc} number of pixels. This is often used as a loss function for training Machine Learning (ML)-models and can be effective at image reconstruction for example in generative ML-models such as Variational Autoencoders (VAEs) (Kingma & Welling, 2019). However, for image-comparison it has limitations. It is particularly sensitive to rotations, and the comparison value can be difficult to interpret (Goodfellow, 2016).

A more robust approach to image comparison is to use summary statistics that are rotation invariant. Univariate statistics summarize and compare pixel frequencies. The most common univariate statistics are the mean and the variance:

$$\mu_x = \frac{1}{n_{rc}}\sum_{i=1}^{n_{rc}} x_i, \;\; \sigma_x^2 = \frac{1}{n_{rc}-1}\sum_{i=1}^{n_{rc}}(x_i - \mu_x)^2.$$

Here, μ_x is the mean and σ_x^2 is the variance of the values in an image. Comparing mean and variance of images can be effective, although it does not take into consideration interactions between pixels, which can be critical in capturing structure in the image.

Second-order statistics are commonly used to capture interactions between pixels. In geostatistics, variograms are used to visualize and study second-order interactions (Pyrcz & Deutsch, 2014). The variogram is directly connected to the covariance and it quantifies spatial correlations at different lags. The formula for the semi-variogram (variogram divided by 2) for an image x at lag distance h is:

$$\gamma(h) = \frac{1}{2|N(h)|}\sum_{(i,j)\in N(h)} \left|x_i - x_j\right|^2,$$

where $N(h)$ is the collection of pixel pairs where the lag distance is h. Variograms of images can be compared at different lags, or one can take the MSE between two variograms at all lag distances. The variogram is limited to characterizing second-order interactions, but higher-order interaction terms may be relevant.

There is an array of metrics to characterize higher-order interactions (Grammer et al., 2020; Leuangthong et al., 2004; Lyster et al., 2004; Zuo et al., 2023). Many of them rely on scanning the image templates. For a single inspection at one location a limited number of immediate neighbors is used. In 2D the template typically comprises of 5 to 9 cells (template center plus immediate neighbors). By scanning the entire image with the template, we can calculate frequencies of specific template events and then compare these frequencies between images. Such methods are called n-point histograms (Boisvert et al., 2010; Deutsch & Pyrcz, 2013; Grammer et al., 2020; Pyrcz, 2016; Tahmasebi, 2018). The histogram part comes from the fact that we often put the specific template events and their associated frequencies in histograms. A common metric of

n-point histograms for image-comparisons is the MSE over the template value frequencies. These n-point histograms can be effective, but they pose computational challenges if the number of pixels in the image or the template size is large, for example, if more than the adjacent neighbors are considered. Furthermore, as the template size increases the number of possible template values increases exponentially, making it difficult to summarize in histograms. This is a challenge, as we are often interested in larger scale structures than the 4-point or 8-point template.

To address this challenge a lot of methods have been proposed. Tan et al., (2014) proposed using multi-dimensional scaling (MDS) to cluster patterns based on a training image. For each cluster, the centroid was extracted, such that for each pattern in an image the closest centroid decides which cluster the pattern belongs to. In this way one can compare histograms of pattern clusters rather than every possible pattern for large templates. Honarkhah & Caers, (2010) proposed an adaptive template selection method based on elbow point detection on the entropy. They defined the entropy to be the information needed to encode a pattern. They additionally used MDS to cluster patterns. Zuo et al., (2023) proposed a pattern classification distribution method (PCDM) inspired by Honarkhah & Caers, (2010) and correlation-driven direct sampling (Zuo et al., 2019) to make adaptive templates. Pattern clusters were found using hierarchical clustering (Vichi et al., 2022). All these methods rely on dimensionality reduction techniques (Nanga et al., 2021). This can be effective at binning templates values into clusters or groups, making it easier to summarize in histograms. However, it becomes difficult to interpret (Tahmasebi, 2018). Lilleborge et. al., (2024) proposed a method based on counting 3D template patterns and looking at the probability of the counts being samples from the same distribution instead of relying on dimensional reduction techniques.

Other approaches for image comparison that has gained a lot of popularity recently are composite metrics. In particular, the Structural Similarity Index Measure (SSIM) proposed by Wang et al. (2004) provides a metric between 0 and 1 for structural similarity. Originally, it was constructed to measure image degradation as perceived changes in structural information, but it was later adopted as an image comparison metric. It works by assessing and comparing three measures of a pair of images. The measures are luminance; represented by the mean of the pixel values, contrast; represented by the variance of the pixel values and structure; represented by the covariance between the image pixels. Combining it all together the metric for two images x and y becomes:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)},$$

$\sigma_{xy} = \frac{\sum_{i=1}^{n_{rc}}(x_i-\mu_x)(y_i-\mu_y)}{n_{rc}-1}$ is the covariance between pixel values of images x and y. Further, $c_1 = (0.01 \cdot L)^2$ and $c_2 = (0.03 \cdot L)^2$ stabilize the division with the weak denominator, with L being the dynamic range of the pixel-values (Wang et al., 2004). The SSIM metric has notable weaknesses (Brunet et al., 2012). First, the metric is not resolution invariant, meaning that x and y must have the same pixel dimensions. Further, the structure measure in SSIM is particularly sensitive to pixel-level information, making it less robust as it computes the covariance at a pixel-to-pixel basis. Improvements such as Complex Wavelet SSIM (CW-SSIM), proposed by Sampat et al. (2009), which first applies the complex wavelet transform and then calculates SSIM on the transformed signals, yield better performance and higher scores for transformed images (Sampat et al., 2009). While effective for continuous images, wavelet transforms are unsuitable for discrete images like binary or segmented ones, as they assume smooth intensity variations. Another popular extension is the Multi-Scale SSIM (MS-SSIM) (Wang et al., 2003) that gives a more robust metric for comparing structures at different scales.

This paper introduces a novel image-to-image comparison metric that addresses these limitations, offering robustness to resolution and rotation. While theoretically invariant to resolution, some smoothing effects at higher resolutions introduce slight sensitivity to scale. Our approach, inspired by SSIM (Wang et al., 2004) and the principles of Mathematical Morphology (MM) developed by Matheron & Serra (2000), applies transformations to convert the image into a point process represented by anchor points, like how MM uses anchors to define points invariant under transformation. This allows us to bypass pixel-to-pixel correspondence. It measures and compares diverse aspects of the images, facilitating image comparison at different resolutions. Our proposed image-to-image comparison metric first transforms both images into a lower dimensional marked point-process representation (Ripley, 2014) where each point in addition to its location has two marks: radius and object label. We then do the comparison based only on this representation. The locations are found by extracting significant landmarks, called anchor points. These anchor points are determined through a novel locally adaptive maxima from the minimal distance transformation MM operator. This transform is resolution invariant, meaning it is not sensitive to pixel dimensions. Dimensionality reduction is accomplished by compressing the image with n_{rc} dimensionality into a set of anchor points with $n_p \times 2$ dimensionality (2 coordinates) where $n_p \ll n_{rc}$ enabling efficient and robust comparison. The added marks of radius and label (described in Section 2) make the procedure output a marked point-process with dimensionality $n_p \times 4$. Once the point process is described, images are compared by evaluating four key measures related to the marked point-process,

- Anchor count: Number of anchor points
- Area coverage: a measure of the overall area spanned by the anchor point radii relative to the image size.
- Anchor points per object: a measure of the average heterogeneity of objects.
- Spatial variance irregularity: a measure of the spread of anchor points.

Notably, the measure comparison is rotation invariant.

Section 2 describes the methodology. Section 3 demonstrates the performance on multiple datasets and benchmarks our proposed PointSSIM metric against MSE, MS-SSIM, and SSIM image-to-image metrics. Section 4 reviews the results and their implications. We conclude with a summary of our findings and suggestions for future research directions in Section 5.

II. Methodology

We introduce the methodology by first showing a schematic representation of the PointSSIM method, explaining each step of the process. We then develop the point-process representation using an example binary image. Figure 1 shows a schematic representation of the PointSSIM method. There are two steps: grid- to marked point-process representation and marked point-process representation to PointSSIM scalar value.

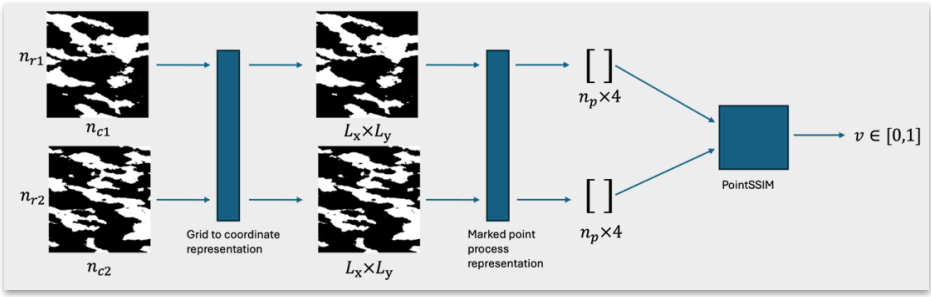


Figure 1: Schematic representation of the PointSSIM method.

A. Grid- to marked-point process representation

The first step of the method is transforming the images from grid coordinates to a base coordinate system. If the two binary images have the same size, i.e. $n_{c1} = n_{c2}$ and $n_{r1} = n_{r2}$, then we can proceed to the next step. If not, we set $(L_x, L_y) = \min((n_{c1}, n_{r1}), (n_{c2}, n_{r2}))$ and $c_{xi} = n_{ci}\Delta x_i, c_{yi} = n_{ri}\Delta y_i, i \in [1,2]$, where (c_{xi}, c_{yi}) are the coordinates in the base coordinate system and $\Delta x_i = \frac{L_x}{n_{ci}}$, $\Delta y_i = \frac{L_y}{n_{ci}}$ are the cell sizes of the base coordinate system.

Once a base coordinate system is established, we can extract the anchor points of the images. We introduce this step of the PointSSIM method by showing the procedure to a marked point-process on a single example binary image $x = \{x_{ij}, i = 1,2,\dots,\frac{L_x}{\Delta x}, j = 1,2,\dots,\frac{L_y}{\Delta y}\}$. Figure 2 shows a display of the binary image produced by multiple point-statistics-based simulation (Grammer et al., 2020).

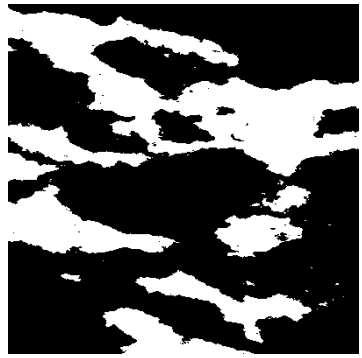


Figure 2: Display of a 300x300 binary image produced using Multiple Point-Statistics.

1) Minimal Distance Transform

The first step involves performing a minimal distance transform on the image (Banerji, 2000). This transform is defined as follows:

Equation 1

$$D(x)_{ij} = \{\min_{(k,l)} d((i,j),(k,l)) : x_{kl} = 0\},$$

where $d((i,j),(k,l))$ is the Euclidean distance between two grid cell positions (i,j) and (k,l). For a grid cell where $x_{ij} = 0$ this distance is 0. For a grid cell where $x_{ij} = 1$, $D(x)_{ij}$ in Equation 1 is larger than 0. This transform calculates the minimal distance from any pixel to the nearest pixel with a value of 0. In Figure 3 we display the minimal distance transform of Figure 2.

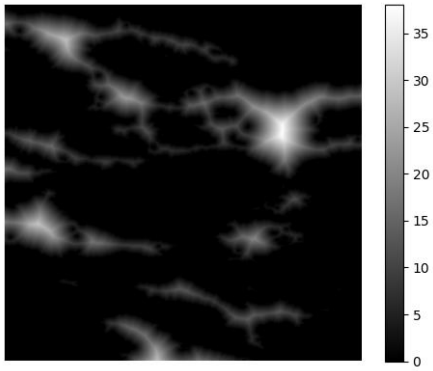


Figure 3: The minimal distance transform of the binary image in Figure 2.

2) Identifying Anchor points

A. The image $D(x)_{ij}, i = 1, \dots, \frac{L_x}{\Delta x}, j = 1, \dots, \frac{L_y}{\Delta y}$ has maximum values along the skeleton of the objects, which lie along ridges. The concept of anchor points involves finding these maximum values of the minimal distance transform, which can be viewed as points invariant under differentiation, analogous to anchor points in MM (Van Droogenbroeck, 2009).

B. For discrete images, local maxima can be identified using a template. Common choices include the 4-point and 8-point templates, depending on whether diagonals are considered neighbors. We apply the 8-point template, treating diagonals as neighbors. We also allow ties, meaning if any point is greater than or equal to all its 8-point neighbors, it is considered a local maximum. For an image y, the local maximum is defined as:

Equation 2

$$L(y)_{ij} = \begin{cases} 1 \text{ if } y_{ij} \geq y_{kl} \text{ for all } y_{kl} \in \mathcal{N}_{ij}, \\ 0 \qquad \text{otherwise,} \end{cases}$$

C. where $\mathcal{N}_{ij} = \{i + a, j + b \mid a, b \in \{-1,0,1\} \wedge (a, b) \neq (0,0)\}$ are the 8 neighbor grid cells of (i, j).

Figure 4 highlights the local maxima within the distance transform $L\big(D(x)\big)$ as red points.

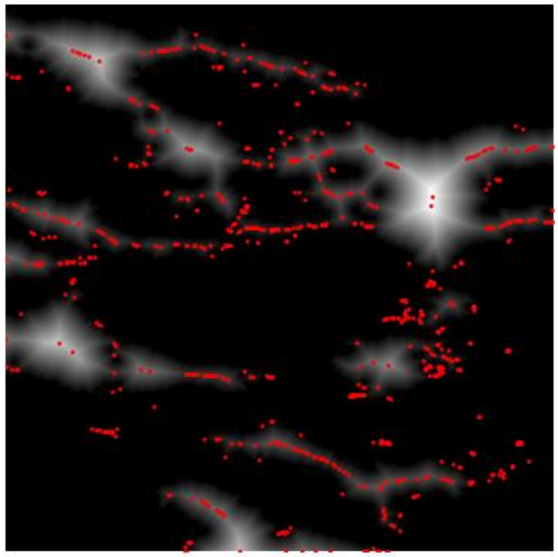


Figure 4: Local maximum points highlighted by red markers of the minimal distance transform of the binary image in Figure 2.

3) Locally Adaptive Anchor points

To avoid superfluous high-density points and to integrate local scale information, we use locally adaptive anchor points, ensuring that no two anchor points are closer to each other than to the edge of the object:

Equation 3

$$L'(y)_{ij} = \begin{cases} 1 \text{ if } D(y)_{ij} \leq \min\limits_{(k,l)} \{d((i,j),(k,l)) : L(y)_{kl} = 1\} \\ 0 \qquad \text{otherwise} \end{cases}.$$

Figure 5 shows the customized anchor points $L'(D(x))$ with both the distance transformed image and the original image as background.

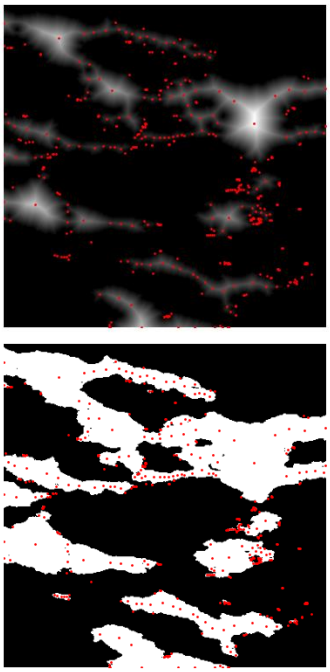


Figure 5: Locally adaptive local maximum points of the minimal distance transform of the binary image in Figure 2 highlighted with red markers, with the minimal distance transform as background to the left and the original image as background to the right.

The locally adaptive method described in Equation 3 ensures that anchor points are distributed based on object size, preserving the relative location and number of points for objects of the same shape, regardless of scale. This approach maintains sufficient spacing between anchor points while preserving enough density to capture meaningful structural details.

B. Marked Point-Process Representation

Once the anchor points are identified, the image is effectively converted into a low-dimensional point-process representation. The point-process representation has a designed feature in that no two anchor points can be closer to each other than to the border of the object. This can be viewed as a marked point-process, where each anchor point has two marks. The first is the effective radius of the anchor point. If we draw a circle around each anchor point to the border of the object, we have the property that each circle only contains a single point, the center. The radius is an important mark as it tells us about the local regularity around the anchor point. Figure 7 shows an illustration of the binary image with the anchor point circles included. The second mark is the object label of the anchor point, telling us which object the anchor point belongs to. Object labels are found by a standard connected component method(Virtanen et al., 2020). This method is using a structuring element to scan the image and labeling based on the labels of its neighbors. We use the 8-neighbors as the structuring element. This is highlighted by the color in Figure 7.

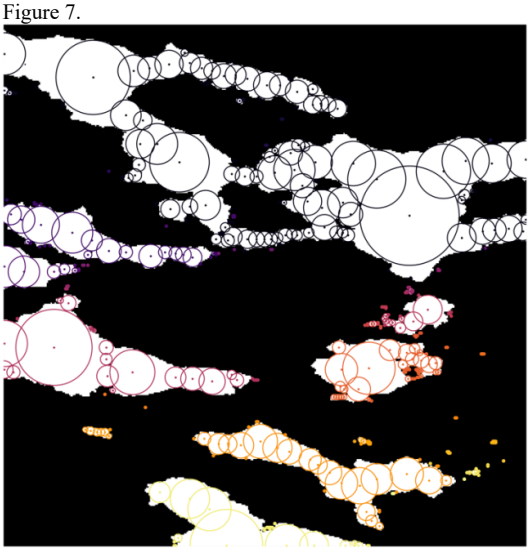


Figure 6: Anchor points of the binary image presented in Figure 2, with the inclusion of position (represented as dots), circles (with the anchor point positions as origo, and the radius representing the distance to the closest border of the object), and label (represented by color scheme).

Drawing inspiration from the SSIM we focus on properties of the marked point-process that are invariant under rotation and resolution scaling transformations. We introduce the following anchor point notation:

Equation 4

$$A^p = \{(i,j): L'(D(x))_{ij} = 1\},$$

Equation 5

$$A^r = \left\{D(x)_{ij}: \ L'\left(D(x)\right)_{ij} = 1\right\},$$

Equation 6

$$A^l = \{l_1, l_2, \dots, l_{|A^p|}\},$$

where A^p represents the grid coordinates of the marked point-process, A^r represents the corresponding radii of the marked point-process, while A^l is a collection of the anchor point labels corresponding to the unique connected areas in the image (color coded in Figure 7). Putting the coordinates and the 2 marks together, we end up with an $n_p \times 4$ vector representation of the binary image: $[A_i^p, A_j^p, A^r, A^l]$.

C. *Marked point-process representation to PointSSIM scalar*

Given the marked point-process vector provided in the previous section we present four measures that capture features of the binary image, which help construct the PointSSIM metric.

Anchor count

The first measure is calculated as the number of unique anchor points in the image,

Equation 7

$$V_1(x) = |A^p|.$$

It summarizes the intensity of points in the image.

Area coverage

The second measure counts the sum of the squared radius marks (proportional to the area of the circle) for the marked point-process, and scales against the area of the image, giving a dimensionless measure. This global measure relates to the foreground proportion in the image (+ some circle overlap),

Equation 8

$$V_2(x) = \frac{\sum_{i=1}^{|A^r|} A_i^{r\,2}}{L_x \cdot L_y}.$$

Anchor points per object

The third measure evaluates the average number of anchor points per object:

Equation 9

$$V_3(x) = \frac{V_1(x)}{\max(A^l)}.$$

Here, the number of objects in the image is $\max(A^l)$, as the objects are ordered from 1 to the number of objects. This measure is related to the heterogeneity of objects in the image. When this number is low it indicates homogeneous objects, a high number indicates heterogeneous objects.

Spatial variance irregularity

The first three measures capture the basic parameterization of the marked point-process. The fourth measure assesses the point structure, focusing on the spatial correlation of the anchor points. This measure provides insights into the clustering, randomness, and structure of the points. Given how the marked point-process is defined in this context, classical measures of spatial point clustering, such as Moran's I (Moran, 1950), are inappropriate. This is because Moran's I is designed for assessing spatial clustering of points, whereas in this context, anchor points are inherently separated due to the repulsive effects of local adaptivity and the separation of objects. Therefore, we need to define clustering differently.

We compare the distribution of points to a Poisson point-process (Daley & Vere-Jones, 1990). In a Poisson point-process, the number of points in an area B follows a Poisson distribution with mean $\lambda|B|$, where λ is the intensity of points. It can be assessed by $\lambda = \frac{n}{|A|}$, where n is the total number of points in the domain, and $|A|$ is the area of the domain.

Given the theoretical variance, we compare it to the empirical variance by splitting the entire domain into several disjoint quadratic subregions B and then counting the number of anchor points within these. In this work, we use 100 subregions, meaning we split our domain into 10x10 subregions, which seems to work well empirically.

We compute the empirical variance for all such counts. To normalize the measure, we take the difference between the empirical variance and the theoretical variance, divided by their sum. This gives a measure that can take values in the range $[-1,1]$. To normalize the measure to the range $[0,1]$ we add 1 and divide by 2. This simplifies to the following measure:

Equation 10

$$V_4(x) = \frac{1}{1 + \frac{\lambda|B|}{s^2}},$$

where:

- s^2 is the empirical variance in the number of anchor points within subregions,
- $\lambda|B| = \frac{n}{100}$ is the theoretical variance,
- $|B|$ is the area of a quadratic subregion, which is $\frac{L_x L_y}{100}$.

We calculate s^2 as,

Equation 10

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(n_i - \bar{n_i})^2,$$

here n_i is the collection of anchor points in subregion i.

Figure 8 is an illustration of the measure for 3 different binary images. When the 1-valued pixels are evenly spaced, the variance between subregion counts is 0 and the measure is 0, if the 1-valued pixels are randomly placed the measured and expected variance is the same, giving a value close to 0.5, and if the 1-valued pixels are clustered the measured variance is much larger than the expected such that the contribution from the expected variance becomes negligible and the measure gets close to 1.

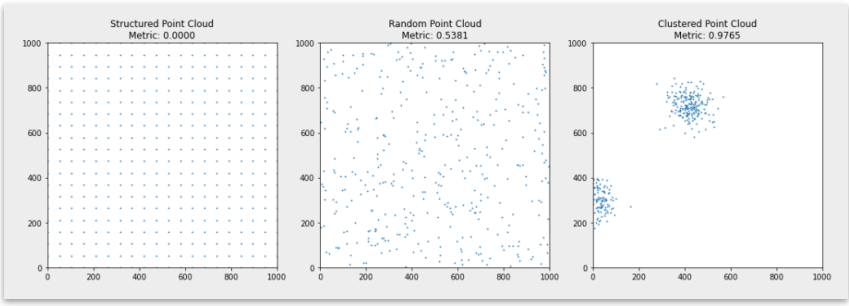


Figure 7: Illustration of the variance irregularity measure for three different binary images. The left display represents a structured arrangement of points, the middle image represents a random arrangement of points and the image to the right represents a clustered arrangement of points. Each of the images have a structure score (ranging from 0 to 1) representing the degree of clustering in the image.

Given the four measures, we represent the binary images as 4-dimensional vectors that can be compared among each other to capture structural similarity with the PointSSIM metric. We employ a reference-free form of comparison, like in SSIM (Wang et al. 2004). The Euclidean distance between each measure is compared and normalized by the maximum value to yield a value between zero and one. The normalized distance of each measure is then averaged to provide a comparison value. For comparing two images x_1 and x_2 we have metric:
Equation 11

$$PointSSIM(x_1, x_2) = 1 - \frac{1}{4}\left(\sum_{i=1}^{3} \frac{\left(V_i(x_1) - V_i(x_2)\right)^2}{\max\left(V_i(x_1), V_i(x_2)\right)^2} + \left(V_4(x_1) - V_4(x_2)\right)^2\right),$$

we avoid double normalization of $V_4(\cdot)$ by putting it outside the sum. This metric gives values in the range $[0,1]$, where 0 represents no structural similarity, and 1 represents full structural similarity in the marked point-process vector space.

III. Results

A. Test Image Scenarios

To assess the performance of the proposed PointSSIM metric, we evaluated it across a range of simulated binary images designed to capture various structural patterns. The primary goal was to see how well PointSSIM captures the similarities and differences between images of different types and structures, particularly in scenarios where traditional methods might struggle. One key consideration is the number of anchor points: PointSSIM relies on enough anchor points being present in the image to generate meaningful statistical comparisons. For example, if large continuous objects cover most of the image, there may be too few anchor points to reliably capture structural differences, leading to increased variance in the metric.

We considered five distinct types of image scenarios, each containing multiple objects. We generated 50 realizations for each dataset to evaluate the performance of PointSSIM. The datasets used for comparison include:

- MPS images: These were generated using Multiple-Point Statistics to emulate geological binary surfaces.

Figure 8: Multiple-Point Statistics binary images.

- TGRF images: Truncated Gaussian Random Field realizations were used, which include nested variograms.

Figure 9: Truncated Gaussian Random Field images.

- Structured ellipses images: These images consist of ellipses arranged in a grid in a highly structured manner.

Figure 10: Structured ellipses images.

- Distorted ellipses images: These images feature ellipses placed randomly in a grid with some

added Gaussian noise to distort their shapes.

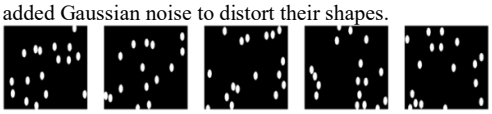


Figure 11: Randomly placed ellipses images.

- Mixture of ellipses and circles: These images include a mix of ellipses and circles with overlapping objects, restricted to certain areas of the image (corners).

Figure 12: Mixture of circles and ellipses images, allowing for overlap and restricted to corners of the image.

These scenarios were chosen to test how well PointSSIM differentiates between images with varying structural complexities, object arrangements, and noise levels. The ability to handle both structured and random configurations is critical for geostatistical applications, where image structures can vary significantly.

B. Results with Test Images

Figure 14 shows histograms and scatterplots of all four key measures for each image scenario, allowing us to observe the separation between datasets. From these plots, PointSSIM effectively distinguishes between the different image scenarios, and the measures themselves are not redundant, meaning no two measures have strong correlations. This independence between measures is important because it ensures that PointSSIM captures a variety of structural aspects of the images, rather than over-emphasizing a single characteristic.

In particular, the structured ellipses dataset shows zero variance in the measures across its 50 realizations. This is as expected because each realization is a direct copy of the original structure. By contrast, both the TGRF and MPS datasets display much higher variance, reflecting the greater diversity in structures between individual realizations. This result highlights how PointSSIM can effectively capture and quantify both structured and irregular spatial patterns.

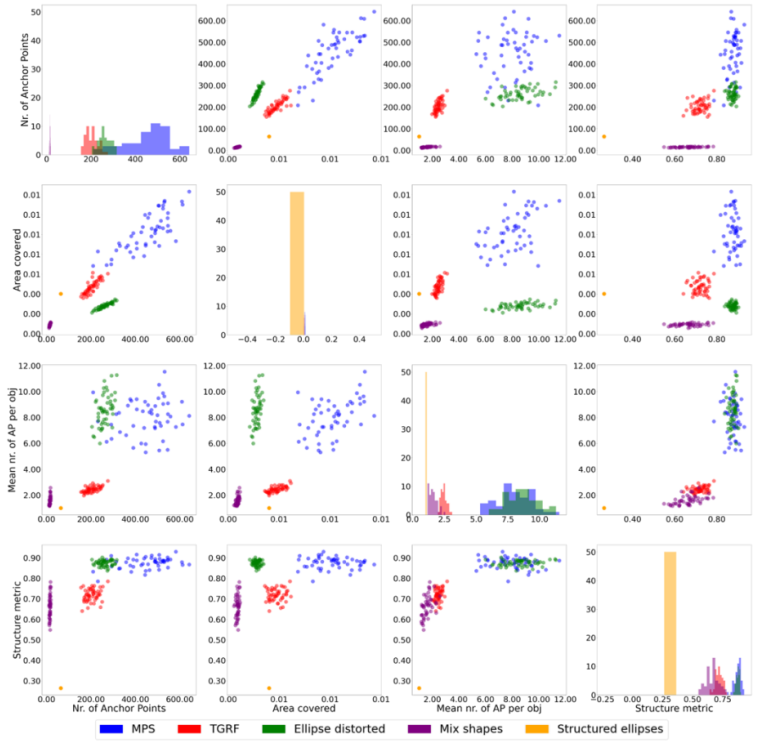


Figure 13: Histograms of all 5 datasets (Figure 9-13) for each of the measures along the diagonals of the Figure. The rows represent the combinations of two measures with the individual data points represented as colored markers.

In Figure 15, we compare PointSSIM against three other popular metrics: SSIM, MSE, and MS-SSIM. The violin plots show the distribution of metric values for pairwise comparisons between the different datasets. Each subplot represents a different dataset combination, with PointSSIM, SSIM, MSE, and Multiscale-SSIM values plotted in blue, orange, green, and red, respectively.

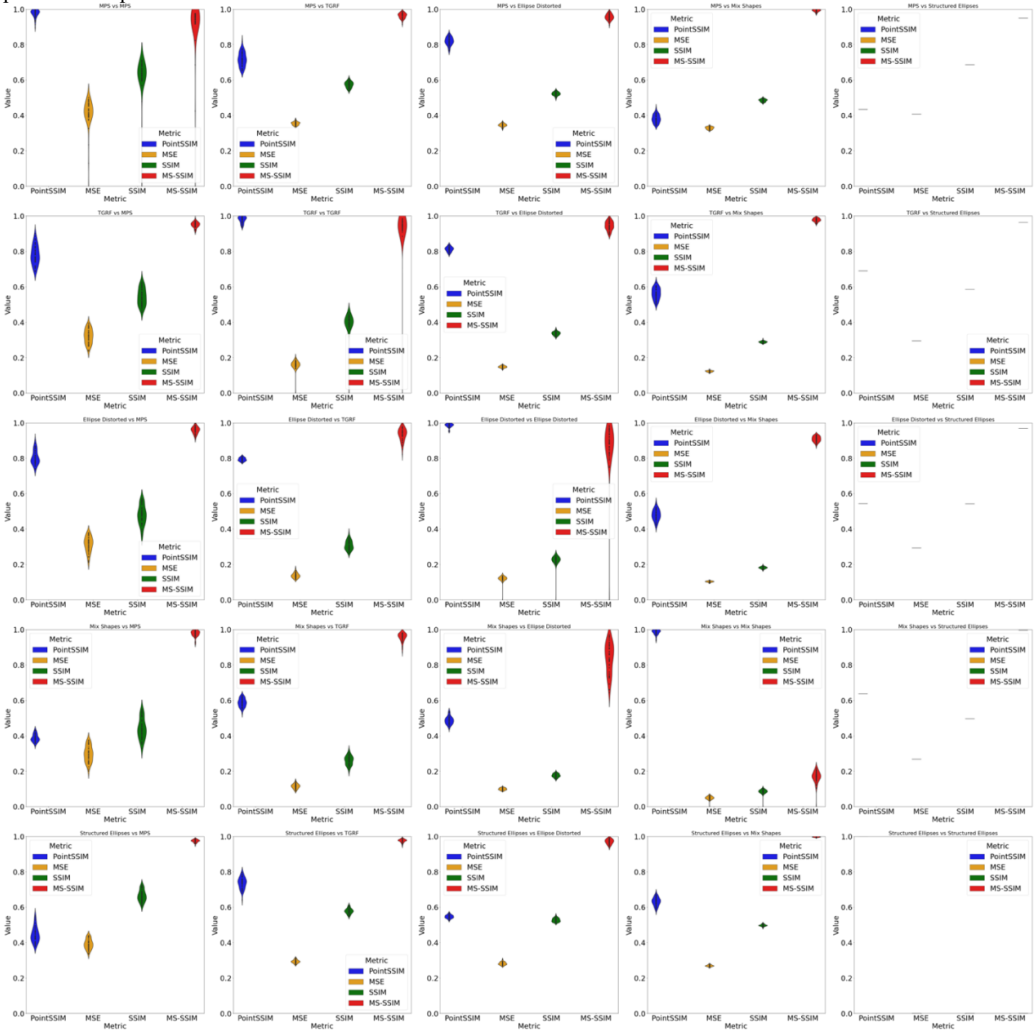


Figure 15: The violin plot of the distribution of the PointSSIM (blue), MSE (yellow), SSIM (green) and MS-SSIM (red) values for each combination of datasets (Figure 9-13). Each combination of datasets are individual subplots, with a unique color for each metric.

Several important observations can be made from this figure:

- Within-class similarity: For each dataset, PointSSIM consistently returns a value close to 1 for within-class comparisons (diagonal elements in the display), with minimal variance. This high within-class similarity indicates that PointSSIM can recognize and quantify structural consistency within these datasets more effectively than the other metrics. By contrast, SSIM, MSE, and MS-SSIM often show lower within-class similarity, with greater variance, making them less precise.
- Between-class differentiation: PointSSIM also excels at distinguishing between different image classes. In many cases, PointSSIM produces lower between-class similarity scores than SSIM, MSE, or MS-SSIM. This superior ability to differentiate between distinct datasets is critical for applications where accurate discrimination between image types is required, such as in geological modeling or pattern recognition.

- Responsiveness to dataset combinations: In the first column of the violin plots, PointSSIM demonstrates a clear response to different combinations of datasets, with varying means and variances. This contrasts with the other metrics, which tend to produce similar responses for different dataset combinations, making it harder to distinguish between them. This indicates that PointSSIM is more sensitive to structural differences, allowing for finer-grained comparisons.

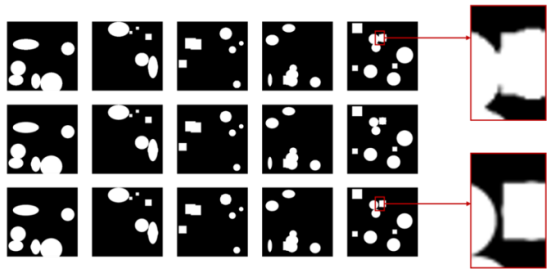


Figure 16: 5 realizations of 256x256, 512x512 and 1024x1024 resolution images.

C. *Additional Resolution Experiment*

One of the main advantages of PointSSIM is its resolution invariance, meaning the metric remains effective even when images are rescaled. This property is especially useful in geostatistics, where images of different resolutions need to be compared. To test this, we conducted an experiment where the mixture of ellipses and circles dataset was generated at three different resolutions: 256x256, 512x512, and 1024x1024. For each resolution, 50 realizations were generated, and the results were compared.

Figure 16 shows five realizations for each resolution, illustrating how the objects are scaled across different resolutions. As the resolution increases, the edges of the objects become smoother, which naturally reduces the number of local maxima detected in the minimal distance transform. This reduction in anchor points could potentially affect the metric, but the PointSSIM method adapts well to these changes.

The histograms in Figure 17 display the measures for the three resolutions, confirming that the values overlap significantly across different resolutions. Figure 18 shows the individual images as scatter points for all measures, with the low and high resolution on the x- and y- axis. Ideally, all the scatter points would lie on the line $y = x$. We observe that there are some fluctuations from this line, especially for the 3rd and 4th measure, where some points lie below the diagonal line. This is an effect of smoothing, where low resolution images have fewer nr. of pixels that can cause objects to merge as in Figure 17. Since measure 3 is the same as measure 1 except that we divide by the number of objects, this causes the low-resolution images to have a lower value when objects are merged. While some small fluctuations are observed due to the smoothing effect between realizations, the measures consistently capture the structural integrity of the images, regardless of resolution. This demonstrates the robustness of the method in scenarios where pixel resolution varies, a significant improvement over pixel-based methods like MSE and SSIM, which are resolution-dependent.

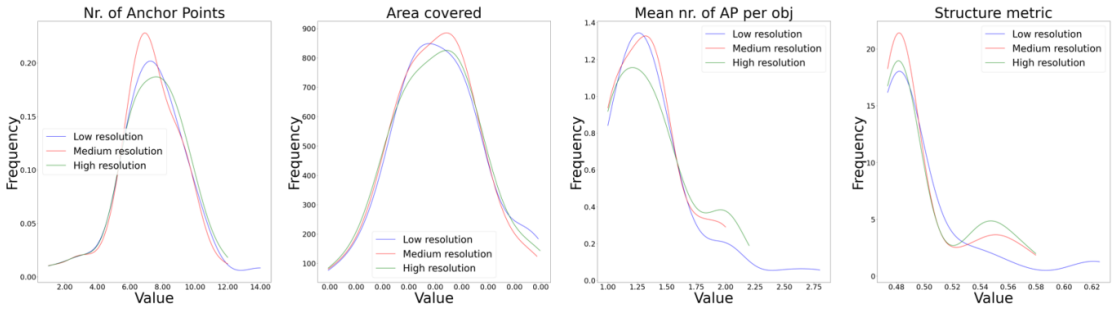


Figure 17: Histograms of all 3 datasets of different resolution (Figure 17) for each of the measures.

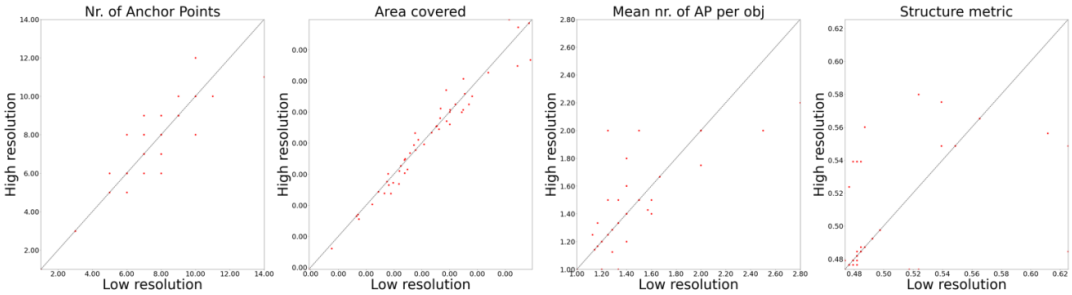


Figure 18: Scatter plot of each measure for low vs high resolution.

IV. Discussion

The results demonstrate that PointSSIM effectively distinguishes image scenarios, outperforming common metrics like SSIM, MSE, and MS-SSIM. By comparing point processes, we can compare across resolutions and create a rotation invariant measure.

Compressing a binary image into four summary measures inevitably leads to some loss of detail. For example, two datasets with different object shapes (e.g., curved vs. straight in Figure 19 and 20) may not be fully distinguished by the current metric, as the curvature is not explicitly captured. This highlights the somewhat arbitrary nature of the chosen measures. While the four measures (anchor count, area coverage, anchor points per object, and spatial variance irregularity) provide a good summary for these datasets, other measures could be more appropriate in different contexts, depending on the specific structural features of interest.


Figure 19: Binary images of curved objects


Figure 20: Binary images of straight objects

Additionally, relying on anchor points requires a sufficient number of these points to generate meaningful statistics. When there are too few anchor points, especially in images with large, continuous objects, the point-based summaries become overly sensitive and less reliable. This can lead to reduced robustness in distinguishing between images with subtle structural differences.

The transformation of binary images into anchor points is efficient, but it involves a trade-off between detail and computational speed. While PointSSIM is effective for binary images, alternative methods like CW-SSIM (Sampat et al., 2009), which maps images to the frequency domain, may be better suited for more complex image types like RGB images, where finer pixel-level details matter.

Despite these limitations, the flexibility of PointSSIM is a strength. The framework allows for adjustments at each stage. The measures derived from the point process can be tailored to look for specific features relevant for the task at hand. Likewise, the extraction of points and marks can be refined, and the choice of marks could also be different from what we have used here. This makes the core idea very adaptable for a wide range of both datasets and applications.

V. Conclusion

In this work, we introduced PointSSIM, a novel, low-dimensional image-to-image comparison metric designed to be invariant to resolution and rotation. The metric compresses complex binary images into a marked point-process representation, allowing us to capture essential structural information efficiently. By utilizing four key measures—anchor point intensity, area coverage, anchor points per object, and anchor autocorrelation—PointSSIM offers a robust and scalable method for comparing binary images across a range of structural scenarios. Our evaluations show that PointSSIM not only outperforms popular comparison metrics like SSIM, MSE, and MS-SSIM but also exhibits the critical advantage of being resolution invariant.

Despite its strengths, PointSSIM has limitations. The compression of binary images into four summary measures inherently leads to some loss of information, meaning that certain image characteristics, such as fine geometric details or subtle shape variations (e.g., curvature differences), may not be fully captured. The method works best when there are enough anchor points in the image, which typically requires a reasonable number of objects or distinct features. In images where the number of anchor points is low, or when large continuous objects dominate the scene, the resulting point-based summaries may become overly sensitive to minor variations, reducing the robustness of the comparisons.

While we have focused on four specific measures for PointSSIM, these are not exhaustive. The choice of these measures reflects a balance between efficiency and structural descriptiveness, but other measures could be explored to capture additional image properties, depending on the specific application. For example, integrating measures that better account for curvature, texture, or more complex spatial interactions could enhance the method's ability to distinguish between images with more intricate differences. However, incorporating additional measures must be done with caution to avoid redundancy and unnecessary complexity, as this could reduce interpretability and make the method harder to apply consistently.

A valuable extension of this work would be to experiment with alternative transformations and measures that go beyond binary images, applying the PointSSIM framework to grayscale or RGB images. In such cases, anchor points could be selected based on other criteria, such as intensity gradients or color homogeneity, and then compared using an expanded set of measures. This would open the method to a broader range of applications, including medical imaging, geological analysis, and remote sensing, where preserving structure across scales is crucial.

Additionally, the PointSSIM framework could serve as a foundational tool in machine learning contexts, particularly for training generative models. By incorporating structural measures alongside pixel-level accuracy, PointSSIM could act as a regularization term, ensuring that the structural integrity of generated images is preserved. This is particularly valuable in tasks such as image synthesis, where maintaining the underlying geometry or patterns of the training data is critical to the quality of the generated outputs.

In summary, PointSSIM offers an efficient, scalable, and flexible approach to image comparison, particularly for binary images requiring structural analysis. Its resolution invariance, combined with its ability to represent images in a low-dimensional space, makes it a valuable tool for a wide range of geostatistical and image analysis applications. Future research should explore more sophisticated transformations and measures to further expand its applicability and utility, particularly in handling more complex datasets.

References

Banerji, A. (2000). An introduction to image analysis using mathematical morphology. In *IEEE Engineering in Medicine and Biology Magazine* (Vol. 19, Issue 4).

Boisvert, J. B., Pyrcz, M. J., & Deutsch, C. V. (2010). Multiple point metrics to assess categorical variable models. *Natural Resources Research*, *19*(3). https://doi.org/10.1007/s11053-010-9120-2

Brunet, D., Vrscay, E. R., & Wang, Z. (2012). On the mathematical properties of the structural similarity index. *IEEE Transactions on Image Processing*, *21*(4). https://doi.org/10.1109/TIP.2011.2173206

Deutsch, C. V, & Pyrcz, M. J. (2013). A Review and Teachers Aide on Multiple Point Statistics. *Center for Computational Geostatistics*, *15*.

Pyrcz and Deutsch (2014): Geostatistical Reservoir Modeling. Oxford University Press, USA.

Grammer, G. M., Harris, P. M. “Mitch,” & Eberli, G. P. (2020). Multiple-point Geostatistics. In *Integration of Outcrop and Modern Analogs in Reservoir Modeling*. https://doi.org/10.1306/m80924c18

Honarkhah, M., & Caers, J. (2010). Stochastic simulation of patterns using distance-based pattern modeling. *Mathematical Geosciences*, *42*(5). https://doi.org/10.1007/s11004-010-9276-7

Kingma, D. P., & Welling, M. (2019). An introduction to variational autoencoders. In *Foundations and Trends in Machine Learning* (Vol. 12, Issue 4). https://doi.org/10.1561/2200000056

Learning, D. (2016). Deep Learning - Goodfellow. *Nature*, *26*(7553).

Leuangthong, O., McLennan, J. A., & Deutsch, C. V. (2004). Minimum acceptance criteria for geostatistical realizations. *Natural Resources Research*, *13*(3). https://doi.org/10.1023/B:NARR.0000046916.91703.bb

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. In *Medical Image Analysis* (Vol. 42). https://doi.org/10.1016/j.media.2017.07.005

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2). https://doi.org/10.1023/B:VISI.0000029664.99615.94

Lyster, S., Ortiz, J. C., & Deutsch, C. V. (2004). Scaling Multiple Point Statistics to Different Histograms. *Center for Computational Geostatistics Annual Report Papers*.

Maindonald, J. (2007). Pattern Recognition and Machine Learning. *Journal of Statistical Software*, *17*(Book Review 5). https://doi.org/10.18637/jss.v017.b05

Matheron, G., & Serra, J. (2000). The Birth of Mathematical Morphology. *Context*, *June*.

Moran, P. A. (1950). Notes on continuous stochastic phenomena. *Biometrika*, *37*(1–2). https://doi.org/10.1093/biomet/37.1-2.17

Nanga, S., Bawah, A. T., Acquaye, B. A., Billa, M.-I., Baeta, F. D., Odai, N. A., Obeng, S. K., & Nsiah, A. D. (2021). Review of Dimension Reduction Methods. *Journal of Data Analysis and Information Processing*, *09*(03). https://doi.org/10.4236/jdaip.2021.93013

Mariethoz and Caers (2014): Multiple-Point Geostatistics. *John Wiley & Sons.*

Ripley, B. D. (2014). Pattern recognition and neural networks. In *Pattern Recognition and Neural Networks*. https://doi.org/10.1017/CBO9780511812651

Sampat, M. P., Wang, Z., Gupta, S., Bovik, A. C., & Markey, M. K. (2009). Complex wavelet structural similarity: A new image similarity index. *IEEE Transactions on Image Processing*, *18*(11). https://doi.org/10.1109/TIP.2009.2025923

Szelisk, R. (2020). Computer Vision: Algorithms and Applications. In *Algorithms and applications* (Vol. 42).

Tahmasebi, P. (2018). Multiple point statistics: A review. In *Handbook of Mathematical Geosciences: Fifty Years of IAMG*. https://doi.org/10.1007/978-3-319-78999-6_30

Tan, X., Tahmasebi, P., & Caers, J. (2014). Comparing training-image based algorithms using an analysis of distance. *Mathematical Geosciences*, *46*(2). https://doi.org/10.1007/s11004-013-9482-1

Van Droogenbroeck, M. (2009). Anchors of Morphological Operators and Algebraic Openings. In *Advances in Imaging and Electron Physics* (Vol. 158). https://doi.org/10.1016/S1076-5670(09)00010-X

Vichi, M., Cavicchia, C., & Groenen, P. J. F. (2022). Hierarchical Means Clustering. *Journal of Classification*, *39*(3). https://doi.org/10.1007/s00357-022-09419-7

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … Vázquez-Baeza, Y. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*(3). https://doi.org/10.1038/s41592-019-0686-2

Wang, Z., & Bovik, A. C. (2009). Mean squared error: Lot it or leave it? A new look at signal fidelity measures. *IEEE Signal Processing Magazine*, *26*(1). https://doi.org/10.1109/MSP.2008.930649

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4). https://doi.org/10.1109/TIP.2003.819861

Wang, Z., Simoncelli, E. P., & Bovik, A. C. (2003). Multi-scale structural similarity for image quality assessment. *Conference Record of the Asilomar Conference on Signals, Systems and Computers*, *2*. https://doi.org/10.1109/acssc.2003.1292216

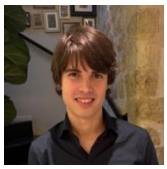
Zuo, C., Li, Z., Dai, Z., Wang, X., & Wang, Y. (2023). A Pattern Classification Distribution Method for Geostatistical Modeling Evaluation and Uncertainty Quantification. *Remote Sensing*, *15*(11). https://doi.org/10.3390/rs15112708

Zuo, C., Pan, Z., Gao, Z., & Gao, J. (2019). Correlation-driven direct sampling method for geostatistical simulation and training image evaluation. *Physical Review E*, *99*(5). https://doi.org/10.1103/PhysRevE.99.053310

Daley, D. J., & Vere-Jones, D. (1990). An Introduction to the Theory of Point Processes. *Journal of the American Statistical Association*, *85*(409). https://doi.org/10.2307/2289568

Lilleborge, M., Hauge, R., Fjellvoll,B. & Abrahamsen,P. (2024). Using Pattern Counts to Quantify the Difference Between a Pair of Three-Dimensional Realizations. *Mathematical Geosciences*. https://doi.org/10.1007/s11004-024-10145-6

AUTHORS



Oscar Ovanger
Norwegian University of Science and Technology
oscar.ovanger@ntnu.no
Alfred Getz' vei 1, 7034 Trondheim
Norway

Dr. Ragnar Hauge
Norwegian Computing Center
hauge@nr.no
Gaustadalléen 23A, 0373 Oslo, Norway

Dr. Jacob Skauvold
Norwegian Computing Center
jas@nr.no
Gaustadalléen 23A, 0373 Oslo, Norway

Prof. Michael Pyrcz
UT Austin
mpyrcz@austin.utexas.edu
Chemical and Petroleum Engineering, 200 E Dean Keeton St, Austin, TX 78712

Prof. Jo Eidsvik
Norwegian University of Science and Techonology
jo.eidsvik@ntnu.no
Alfred Getz' vei 1, 7034 Trondheim
Norway