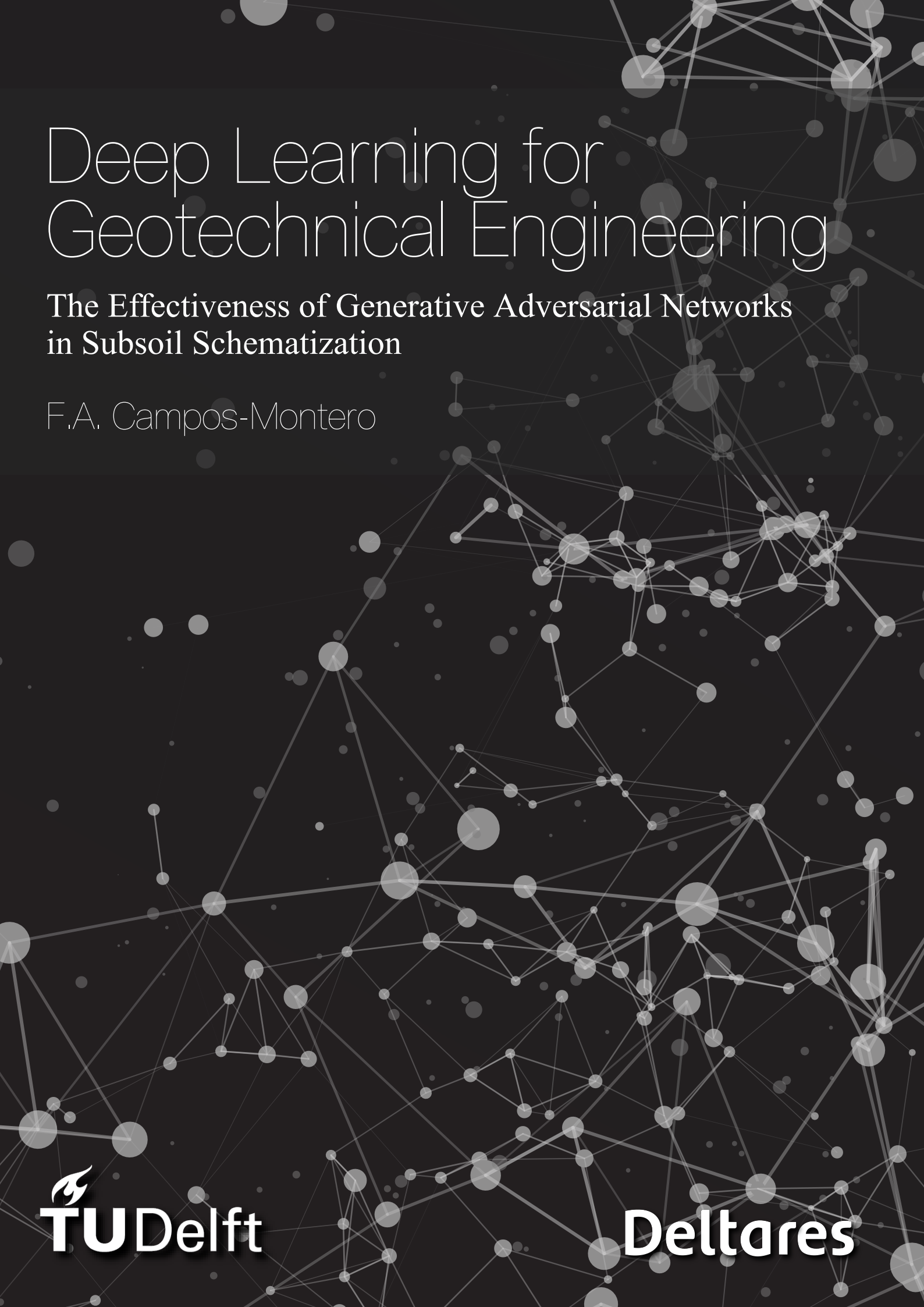
Deep Learning for Geotechnical Engineering

The Effectiveness of Generative Adversarial Networks in Subsoil Schematization

F.A. Campos-Montero



TU Delft

Deltares

Deep Learning for Geotechnical Engineering

The Effectiveness of Generative Adversarial Networks in Subsoil Schematization

by

F.A. Campos-Montero

to obtain the degree of Master of Science for the Geo-Engineering track
in the Applied Earth Science programme at the Delft University of Technology,
to be defended publicly on Friday, August 25, 2023.

Student number:	4877047	
Project duration:	December 2022 – August 2023	
Thesis committee:	Dr. P.J. Vardon	TU Delft, chair
	Dr. R. Taormina	TU Delft
	Dr. B.E. Zuada Coelho	Deltares

This report is confidential and cannot be made public until after August 2023.

Cover: Interpretation of an Artificial Neural Network by the OpenAI system DALL·E 2, with the prompt: *"Neural network abstract background without text"* available at https://openai.com/dall-e-2/.

Abstract

This thesis introduces a novel Generative Adversarial Network application called SchemaGAN, which has been adapted from the Pix2Pix architecture to take Cone Penetration Test (CPT) data as a conditional input and generate subsoil schematizations. For training, validation and testing, a database of 24,000 synthetic schematizations of size 32x512 pixels was created, representing a broad spectrum of stratigraphical complexity in the layered models. Each synthetic cross-section was additionally transformed into a CPT-like image with less than 1% of the original data remaining at random locations along the model. After training for 200 epochs, the best-performing SchemaGAN Generator was chosen from the validation, and the effectiveness of SchemaGAN in generating subsoil schematizations was tested against traditional interpolation methods (Nearest Neighbour, Inverse Distance Weight, Kriging, Natural Neighbour) and newer methods such as Inpainting. The evaluation metrics obtained reveal that SchemaGAN outperforms all other methods, with results characterized by clearer layer boundaries and accurate anisotropy within the layers. In contrast, Nearest Neighbour and Kriging are characterized by a lack of continuity and blurry layer boundaries respectively. Inverse Distance, Natural Neighbour and Inpainting fail to come close to the performance of the other methods. The superior performance of SchemaGAN is confirmed through a blind survey, in which SchemaGAN ranked as the top-performing method in 78% of cases according to experts in the field. Results also suggest that SchemaGAN is the least affected method by the location of CPT data along the cross-section. In a real case study, SchemaGAN demonstrates better predictive accuracy for known CPT data than both Nearest Neighbour and Kriging interpolation methods. The future potential lies in refining its performance by considering enhancements such as training with real CPT data, incorporating additional conditional inputs, and exploring larger inputs or specialized databases. All the code related to the project has been made publicly accessible.

Keywords: Deep learning, machine learning, generative adversarial network, schematization, cone penetration test, interpolation, stratigraphy.

Acknowledgements

I would like to express my sincere gratitude to the following individuals for their contributions and support throughout the completion of this thesis:

To the team at Deltares for generously opening their doors and providing me with the opportunity to work within their respected organization.

To my committee members. Bruno Zuada Coelho, I want to express my appreciation for your kindness and genuine concern. Your guidance and supervision significantly influenced the direction of this project, and your selfless assistance has been immensely valuable. Phil Vardon, your dedication to upholding geotechnical principles is greatly appreciated. Your thought-provoking challenges consistently encouraged me to delve beyond the surface findings. Riccardo Taormina, your contagious enthusiasm and flow of innovative ideas were really inspiring. Your expertise in Machine Learning was vital in maintaining the accuracy and rigour of this project.

My gratitude also extends to Remon Pot for offering me the chance to explore new horizons in Geotechnical Engineering through this thesis. Your open embrace of innovation has been truly motivating. Heartfelt thanks are also due to Eleni Smyrniou for her generous assistance, coding tips and guidance, all of which have significantly contributed to the project's success.

I would also like to thank all my Deltares colleagues with whom I've shared countless lunch breaks and coffees. Your welcoming attitude and enthusiasm have fostered a supportive environment that has been a pleasure to be a part of.

Lastly, I want to extend my thanks to my family and friends, both new and old. I appreciate each of you for being part of this journey and for your constant encouragement.

F.A. Campos-Montero
Delft, August 2023

Contents

Nomenclature

List of abbreviations

1D	One dimensional
2D	Two dimensional
3D	Three dimensional
AI	Artificial Intelligence
ANFIS	Adaptive Neuro-Fuzzy Interference System
ANN	Artificial Neural Network
BCE	Binary Cross-Entropy
cGAN	Conditional Generative Adversarial Network
CNN	Convolutional Neural Network
Conv2D	Convolutional layer
CPT	Cone Penetration Test
DCGAN	Deep Convolutional Generative Adversarial Network
DL	Deep Learning
FIS	Fuzzy Interference System
FNN	Feedforward Neural Network
GAN	Generative Adversarial Network
Ic	Soil Behaviour Type Index
ID	Irregular decoder block
IDW	Inverse Distance Weighting
IE	Irregular encoder block
LSTM	Long Short-term Memory
ML	Machine Learning
MSE	Mean Squared Error
NAP	Amsterdam Ordnance Datum
NatNe	Natural Neighbor Interpolation
NeaNe	Nearest Neighbor Interpolation
NN	Neural Network
P-GAN	Progressive Generative Adversarial Network
px	Pixels
RD	Regular decoder block
RE	Regular encoder block
ResNet	Residual Neural Network
RF	Random Field
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SPT	Standard Penetration Test
SVM	Support Vector Machine
T-Conv2D	Transpose convolutional layer
VD	Voronoi Diagram
WGAN	Wasserstein Generative Adversarial Network

List of symbols

Φ	Phase shift
f_r	Friction ratio
q_t	Cone tip resistance
A	Amplitude
D	Vertical shift
T	Period

1

Introduction and Background

1.1. Background

Geotechnical Engineering is the study of soils and rocks, known as geo-materials, and plays a crucial role in many engineering fields (Baghbani et al., 2022). Every building or structure must be built on the ground, and in order to make sure that they perform satisfactorily, it is necessary to ensure that there are no excessive deformations or failures in the underlying materials. This requires a thorough understanding of the subsoil behaviour and its interaction with the structure (Das and Sivakugan, 2015). Unlike other engineering materials, soils and rocks are anisotropic and heterogeneous, with a highly non-linear behaviour due to differences in their origin and formation process (Zhang et al., 2021; Baghbani et al., 2022). Geo-materials are difficult to predict and can vary greatly both in stratification and in properties (Leunge, 2019; Beiyang, 2022).

One of the first and most important tasks in any geotechnical project is the characterization of the subsoil, which involves mapping the subsoil distribution and variability of the layers, assessing its behaviour, and creating the necessary geotechnical models. This is a crucial input for solving most geotechnical problems, such as bearing capacity, permeability and seepage, consolidation and settlements, mass movements and slope stability, liquefaction, and others. These calculations all rely on the schematization of the subsoil.

The first step in creating a subsoil characterization is accounting for the heterogeneity of the subsoil. To achieve this, geotechnical exploration is carried out at the beginning of any project, resulting in a geotechnical schematization of the layers geometry as illustrated in Figure 1.1. It typically involves in-situ investigation and laboratory testing, which are often descriptive, costly and time-consuming (Baghbani et al., 2022). In-situ tests normally rely on boreholes, penetration tests, and geophysical surveys. Most of these tests only gather data in one dimension at a single point in space, which constrains the validity of the results when the subsoil variability is accounted for. Additionally, exploration of the subsoil is often disregarded and underfunded (Jaksa, 2000; Oluwatuyi et al., 2023).

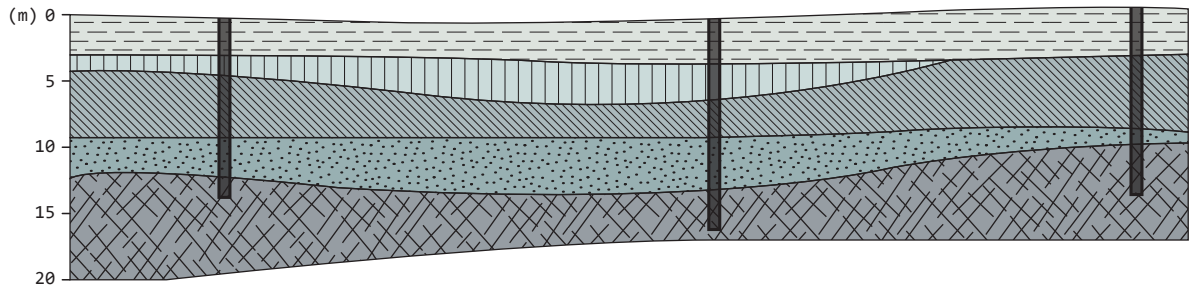


Figure 1.1: Subsoil schematization from in-situ exploration (adapted from Budhu, 2015).

This characterization process results in gaps in which no subsoil information is available, yet an assumption needs to be made to complete the subsoil models. Geotechnical engineers have three main ways of dealing with

this interpretation and extrapolation of the in-situ data: expert criteria, interpolation methods, and more recently Artificial Intelligence (AI).

Traditionally, the interpretation of the information gaps for the schematization is made solely by expert criteria, relying on prior knowledge, the results from the geotechnical campaign and the underlying stratigraphic laws (Harris, 1989). Geostatistical spatial correlation and interpolation tools have also been extensively used in subsoil schematization (McBratney et al., 2003; Grunwald, 2009; Heuvelink and Webster, 2022). They include linear models, regression methods, classification/discrimination methods and geostatistics among others.

More recently, with improvements in computation efficiency, AI has become more widely available (Nguyen et al., 2019), leading to a surge of interest in Machine Learning (ML) research across a wide range of fields (Figure 1.2). Geotechnical Engineering is no exception, as many problems are characterized by great uncertainty and involve factors that can not be directly determined. This has made ML a very attractive tool as reflected in over three decades of research into AI applications in this field (Zhang et al., 2021; Baghbani et al., 2022). This includes the schematization problem, where ML is starting to replace the geostatistical methods, mainly using random forests (RF), gradient boosting and artificial neural networks (ANN) (Heuvelink and Webster, 2022).

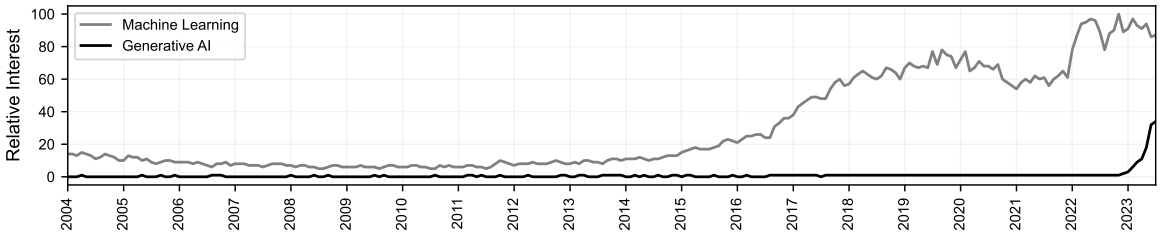


Figure 1.2: Timeline of Google Trends for Machine Learning and Generative AI web searches (adapted from Google, 2023).

To understand how AI could help in solving subsoil schematization, it's useful to briefly discuss how computers solve problems. Goodfellow et al. (2016) explains that computers are capable of solving intellectually challenging problems that are difficult for human beings to tackle, as long as they can be described by a list of formal, mathematical rules. The true challenge to AI comes in the form of solving tasks that are easy for human beings to perform but hard to describe. The kind of problems we solve intuitively and feel automatic (e.g. identifying faces).

This hurdle was solved by a significant breakthrough in AI: computers were able to learn from experience and understand the world in terms of a hierarchy of concepts, with each concept defined through its relationship to simpler concepts. This creates a deep network with many layers, a process known as Deep Learning (DL). By following this approach, AI can avoid the need for human operators to formally specify knowledge, gathering its own experience and then applying it to solve the required tasks (Goodfellow et al., 2016; Shrestha and Mahmood, 2019).

One such application for DL algorithms is the use of Generative Adversarial Networks (GANs). At its core, GANs are a class of ML framework designed to model and generate new data distributions that resemble the input data (Goodfellow et al., 2014). Two Neural Networks contest with each other, with one network striving to generate more realistic data while the other network tries to differentiate between the real and generated data (Goodfellow et al., 2014; Goodfellow, 2017; Tomczak, 2022). The interplay between these two networks leads to rapid learning, resulting in the generation of more and more realistic data. This makes GANs a promising tool for geotechnical schematization, using known information to infer unknown information (Zhang et al., 2023), as illustrated in Figure 1.3. Here the aim is for the model to learn how to interpret in-situ data and produce realistic subsoil schematizations.

This research thesis aims to investigate the potential of AI to solve the challenge of schematization. Specifically, the study will explore the feasibility of GANs to acquire the level of knowledge and expertise necessary to recreate the intuitive decision-making processes of experienced engineers for subsoil schematization.

Figure 1.3: Generative Adversarial Network Results. From left to right shows the original image, input image and results (modified from Yu et al., 2018).

1.2. Problem statement

The process of subsoil schematization is an essential part of every project, as it forms the basis for the subsequent analysis of most geotechnical problems. Traditionally, the schematization is performed by experienced engineers using in-situ test results and their own expert knowledge, and occasionally aided by interpolation and geostatistical tools, to interpret and model the underlying geology and soil conditions. However, this process is labour-intensive, time-consuming, and relies heavily on the expert criteria of the engineer. This means that there is no unique solution, as the results are subject to the individual interpretation of every engineer. As projects become larger and more complex, the challenges of maintaining consistency and accuracy in subsoil schematization can also become quite significant. Additionally, a significant amount of knowledge and information about the subsoil from previous projects is not being actively utilized in the current schematization process, leading to a waste of valuable resources. Only the experience gained by the engineers is carried on, rather than the collective knowledge and data from previous projects. As a result, there is a need for a more efficient, consistent, and accurate approach to subsoil schematization in Geotechnical Engineering.

1.3. Opportunity

AI's ability to learn and gain experience independently positions it as a promising tool for solving the subsoil schematization problem, closely aligning with how engineers currently interpret cross-sections based on their expert criteria. A successful AI tool in subsoil schematization would bring significant benefits to all geotechnical engineering projects, including:

- The schematization process which traditionally is labour-intensive and time-consuming can become quick and efficient with the use of AI.
- The inconsistencies that generally arise from the personal biases in the interpretation made by each engineer could be eliminated with an AI, making the soil models more accurate and consistent.
- Most of the information available from previous projects could be used as knowledge for solving subsoil schematizations, resulting in more accurate and realistic geotechnical models.
- AI algorithms can be trained to recognize patterns and trends in the data that may be difficult for humans to spot, which could lead to schematization of the subsoil that better represents the reality in the field.

1.4. Research objective

From the problem statement presented in Section 1.2 and the opportunities presented in Section 1.3, the following research objective has been defined:

> *Evaluate the effectiveness of using Generative Adversarial Networks for subsoil schematization by comparing the performance against the source models and traditional interpolation techniques.*

1.4.1. Research questions

From the main research objective of section 1.4 the following research questions can be formulated:

1. *How effective are Generative Adversarial Networks at solving the subsoil schematization problem, and*

Table 1.1: Thesis report outline.

Report section	Chapter	Description
Introduction	Chapter 1	Situation, problem statement, opportunity and objectives
Supporting theoretical information	Chapter 2	Background information about subsoil schematization
		Background information about traditional interpolation techniques
		Background information about AI and GANs
Materials and methods	Chapter 3	Overview of the data and methods followed during the investigation
Application and implementation	Chapter 4	Construction of the synthetic schematization databases
	Chapter 5	Construction of Generative Adversarial Network for schematization
		Training, validation and testing of the model
Comparison of methods	Chapter 6	Comparison with traditional interpolation methods
		Expert criteria survey
		Case study: Application of the model to real CPT data
Closure and conclusions	Chapter 7	Discussion and future directions
	Chapter 8	Conclusion and recommendations

what are the limitations of this approach?

2. *How does the use of Generative Adversarial Networks for subsoil schematization compare to the current practice in terms of accuracy, efficiency, and any other relevant metrics?*

1.5. Report outline

This thesis report is presented following a linear narrative. A logical order on how the GAN model was constructed, trained, validated and tested, is used. The introduction, relevant theoretical information and material and methods are presented in Parts I and II. The construction of the synthetic schematization databases and the core of the research with the GAN construction, implementation and comparison against the traditional interpolation methods are presented in Part III. Discussion on the results and conclusions are drawn in Part IV. A breakdown of the report structure is given in Table 1.1.

1.6. Code Availability

The complete implementation of the SchemaGAN tool, along with associated scripts and resources, is available for public access on GitHub. Interested readers and researchers can access and utilize the code repository to better understand the methodology, reproduce the results, and potentially build upon the work presented in this thesis. The GitHub repository can be found at the following URL: https://github.com/fabcamo/schemaGAN.

2

Literature Review

2.1. Introduction

This chapter provides a comprehensive overview of crucial topics central to the research objective. It begins with an examination of subsoil schematization and the foundational principles that govern this practice. The second part presents an overview of traditional interpolation methods. The third section examines the growing role of AI in geotechnics, highlighting the increasing overlap of these two disciplines. Concluding the chapter is a detailed exploration of GANs, a notable advancement in ML, and its potential to significantly impact subsoil schematization techniques.

2.2. Subsoil Schematization

Subsoil schematization is the delineation of subsurface units and involves the determination of geometry and number of soil layers, as well as the demarcation of the stratigraphic boundaries among different layers (Bossi et al., 2016). It is the process of creating a simplified model or representation of the subsoil characteristics (Juang et al., 2019; Zuada-Coelho and Karaoulis, 2022) and should work as a tool to understand the heterogeneous and complex nature of the subsoil, including variables such as soil type, stratum thickness and depth, and various mechanical and physical properties (Wood, 2004). In literature, they are also referred to as soil stratigraphy, subsurface stratigraphy, geotechnical cross-section and geotechnical profile, among others.

These schematizations serve as the basis for all geotechnical investigations and are a paramount activity for the execution of infrastructure earthworks (Li et al., 2019; Zuada-Coelho and Karaoulis, 2022). As an initial step in infrastructure projects, it influences critical aspects like design, construction, costs, risk management and environmental damage (Cosenza et al., 2006; de Rienzo et al., 2008; Das, 2010; Reiffsteck et al., 2018; Dawoody, 2021). From very early in the geotechnical practice, Terzaghi (1929) indicated that “minor geological details” could have a major impact on the performance and the stability of structures and slopes.

By integrating both quantitative and qualitative data, subsoil schematization translates complex subsurface conditions into an understandable diagram that captures the complex heterogeneities of the underground environment. This diagram plays a key role in informing decision-making processes, identifying geotechnical risks and facilitating the optimization of construction activities (Cosenza et al., 2006; Ijaz et al., 2021; Zuada-Coelho and Karaoulis, 2022).

2.2.1. Theoretical Background

Two key concepts govern the formation of subsoil layers: the stratigraphic principles and the weathering processes. Stratigraphy provides insights into the creation and layering of geological materials (Nichols, 2023), while weathering involves the transformation of rocks into soil particles (Willgoose, 2018). Over time, these factors interplay and, influenced by gravity and the deposition processes, result in the subsoil layers in which all geotechnical works take place.

The process of rock weathering involves the physical and chemical breakdown of rocks and minerals by atmospheric and biological factors. Physical weathering includes mechanisms like thermal expansion, frost wedging,

and fractures without altering the chemical composition. Chemical weathering involves transformations into new compounds through processes like hydrolysis, oxidation, and carbonation (Bland and Rolls, 1998). This weathering process is crucial for soil formation and determining subsoil characteristics

The principles of stratigraphy illustrated in Figure 2.1 and established by Steno and Hutton, serve as essential guidelines for understanding the structure of subsoil layers. They include:

1. Principle of Superposition: In an undisturbed sequence of soils or rocks, the youngest layer is at the top and the oldest at the bottom. Each layer is presumed to be younger than the layer beneath it (Steno, 1968).
2. Principle of Original Horizontality: Layers of sediment, when first deposited, are fairly horizontal because sediments settle out of fluid onto surfaces of low gradient (nearly level). Thus, if we find them at an angle, they have been disturbed since their formation (Steno, 1968).
3. Principle of Lateral Continuity: Layers of sediment initially extend laterally in all directions. That is, they are continuous over a large area. Consequently, similar layers of rock over a broad region were likely laid down at roughly the same time (Steno, 1968).
4. Principle of Cross-Cutting Relationships: A geological feature must exist before it can be cut by any other feature. For instance, if a fault line cuts across the rock layers, it must have formed after the layers were deposited (Hutton, 2005).

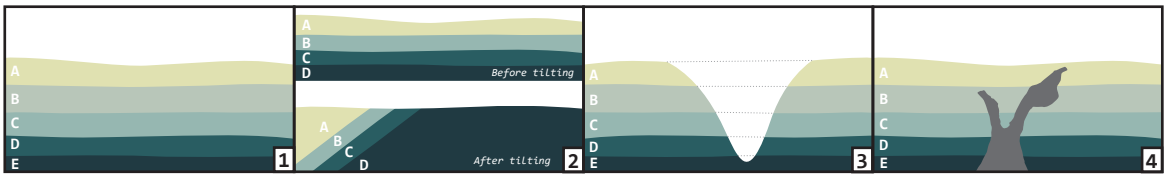


Figure 2.1: Diagram of the stratigraphic principles.

2.2.2. Practical Application

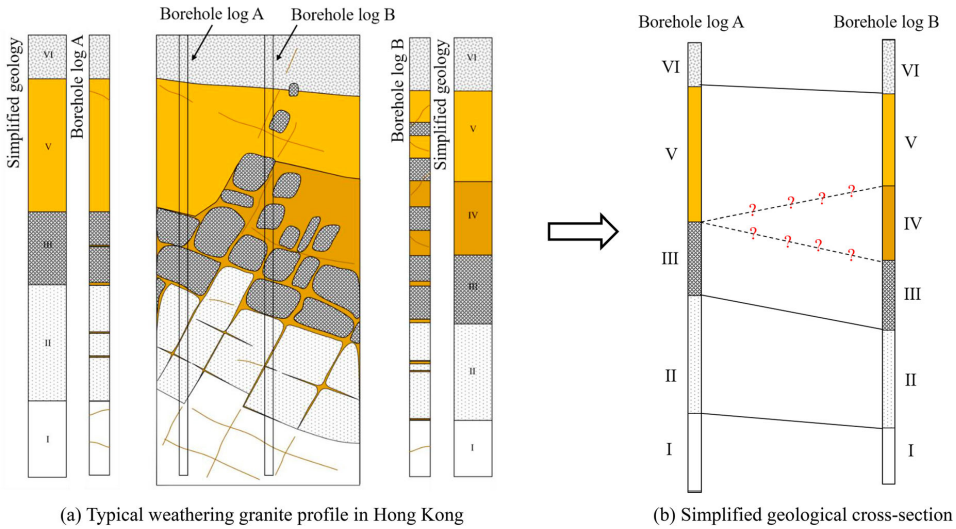
In geotechnical practice, investigation of subsurface conditions commonly employs measurements from scattered sources (Li et al., 2019; Phoon et al., 2022) such as boreholes, cone penetrometers (CPT), and standard penetrometers (SPT), complemented by geophysics. These measurements provide specific profiles at the test location, yet their sparsity often limits the detail of subsurface cross-sections (Bossi et al., 2016; Phoon et al., 2022). The estimation of soil layer depth and thickness at untested locations poses a significant challenge due to the inherent variability in natural soils (Bossi et al., 2016; Li et al., 2019).

Theoretically, understanding the number of units, their thickness, and spatial distribution is vital for aligning real-world phenomena with their mathematical representations (Phoon and Kulhawy, 1999). However, in practice, achieving an exact match is both unattainable and impractical. Consequently, geotechnical practice primarily revolves around optimization, based largely on inferences drawn from data (Bossi et al., 2016).

The challenge of this process is illustrated by Wang et al. (2022) in Figure 2.2 with the interpretation of a typical Hong Kong granitic site. For two boreholes logs A and B, six materials were identified. Stratigraphy in log B is quite complex in comparison with the layering in log A. This makes it difficult to determine the thickness and dipping of the different units.

It's a standard geotechnical practice to simplify the stratigraphy, reducing detail for a broader overview. This is often guided by local experience or expert knowledge (Bogus and Godlewski, 2019; Wang et al., 2022; Phoon et al., 2022). Such simplifications are evident in the creation of geotechnical models, where the modeller decides what details to retain or discard (Phoon and Kulhawy, 1999). These factors may lead to significant stratigraphic uncertainty (Juang et al., 2019; Wang et al., 2022).

Traditionally, the interpretation of the information gaps for the schematization is made solely by expert criteria, relying on prior knowledge, the results from the geotechnical campaign and the underlying stratigraphic laws (Harris, 1989). Geostatistical spatial correlation and interpolation tools have also been extensively used in subsoil schematization (McBratney et al., 2003; Grunwald, 2009; Heuvelink and Webster, 2022). An overview of the most popular methods is presented in Section 2.3.



(a) Typical weathering granite profile in Hong Kong (b) Simplified geological cross-section

Figure 2.2: Illustration of subsoil schematization development in geotechnical practice for (a) typical weathering granite profile in Hong Kong and the (b) simplified cross-section (Wang et al., 2022).

Adding to the complexity of the process, the subsoil model can be tailored to the complexity required by the specific engineering problem, considering various factors such as the type of material, physical properties, and mechanical properties. For instance, a focus on bearing capacity and settlement characteristics with simplified layers would be essential in foundation design, whereas an understanding of shear strength and groundwater conditions with detailed layering is necessary for slope stability analysis (Bowles, 1996; Das, 2010).

2.3. Review of Interpolation Techniques for Subsoil Data

Geotechnical engineers often work with limited data from finite point measurements in order to create a generalized model of the subsoil structure and characteristics (Mlynarek et al., 2005; Heuvelink and Webster, 2022). Here, they are often supported by various geostatistical interpolation techniques which help fill spatial data gaps and improve the accuracy of the subsoil schematization (Mlynarek et al., 2005).

This section highlights four specific methods: Nearest Neighbor (NeaNe), Inverse Distance Weighting (IDW), Natural Neighbor (NatNe), and Kriging; as well as a newer technique called inpainting. Each method has its own strengths and limitations. The aim of this section is to deliver a straightforward understanding of how these methods are used, their benefits, and any potential drawbacks in subsoil schematization. For further details on the different interpolation methods, Ledoux et al. (2022) offers a good introduction, while Rahman et al. (2021) shows their application to CPT data.

2.3.1. What is Spatial Interpolation?

Given a set S of data points, denoted as p_i, within a two-dimensional real plane, each with an attached attribute a_i, spatial interpolation refers to the process of estimating the attribute value at an unsampled location x. The objective of this process is to identify a function $f(x, y)$ that fits the data as optimally as possible (Ledoux et al., 2022). The basis for interpolation lies in spatial autocorrelation, the principle suggesting that attributes of two points in close spatial proximity are more likely to exhibit similarity compared to those of two points situated farther apart (Haining, 2001).

In very simple terms, spatial interpolation is the process of estimating values at unsampled locations within a given area based on known values at sampled locations (McKillup and Dyar, 2010; Chiles and Delfiner, 2012).

2.3.2. Weighted-average Methods

These methods use a subset of the sample points, to which a weight is assigned, to estimate the value of the dependent variable (Rahman et al., 2021; Ledoux et al., 2022). The interpolation function f of such methods, with which we obtain an estimation $\hat{a}$ of the dependent variable a, has the form:

$$f(x) = \hat{a} = \frac{\sum_{i=1}^{n} w_i(x) a_i}{\sum_{i=1}^{n} w_i(x)} \tag{2.1}$$

where a_i is the attributes of each data point p_i, w_i is the weight of each data point and x is the interpolation location.

Nearest Neighbor Interpolation (NeaNe)
This is one of the simpler methods, where the value of an attribute at the location is simply assumed to be equal to the attribute of the nearest data point, as illustrated in Figure 2.3. This data point gets a weight of exactly 1.0, which means distance from the known value to the interpolation point does not play a role (Ledoux et al., 2022).

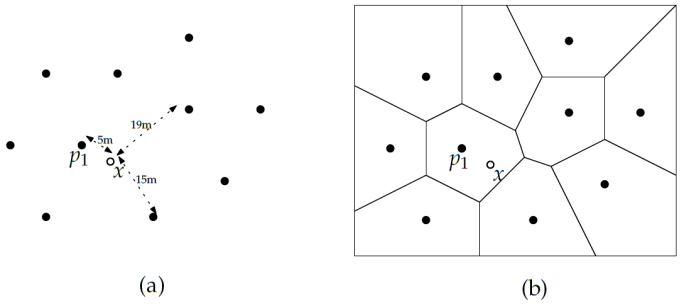


Figure 2.3: Nearest neighbour interpolation (a) the estimated value at x is that of the closest data point, and could also be obtained from (b) a Voronoi diagram (adapted from Ledoux et al., 2022).

The Nearest Neighbor Interpolation (NeaNe) method has notable advantages, including its simplicity, computational speed, and ability to handle anisotropic data distributions. Moreover, it is exact and localized. However, its suitability for creating realistic representations of continuous fields is limited. Specifically, it lacks the ability to maintain smoothness and continuity. A noteworthy limitation is the discontinuity at borders that the method introduces in the interpolation process (Ledoux et al., 2022).

Inverse Distance Weighting (IDW)
Variations of IDW exist depending on the method to choose the data points p_i used in the interpolation at location x (Ledoux et al., 2022). Using as an example the data points and the interpolation location of Figure 2.4(a), the IDW variations are:

- k-nearest neighbours: a given number of neighbours (i.e. k=4) can be used irrespective of how far they are. This ensures that IDW yields a continuous surface (Figure 2.4(b)).
- search circle or ellipse: The data points are selected based on a circle defined by a radius or an oriented ellipse with r_1 and r_2 and an orientation θ (Figure 2.4(c)).
- k-per-quadrant: This method ensures that neighbours used in the interpolation are not all in one direction, by taking the k-nearest per quadrant (Figure 2.4(d)).

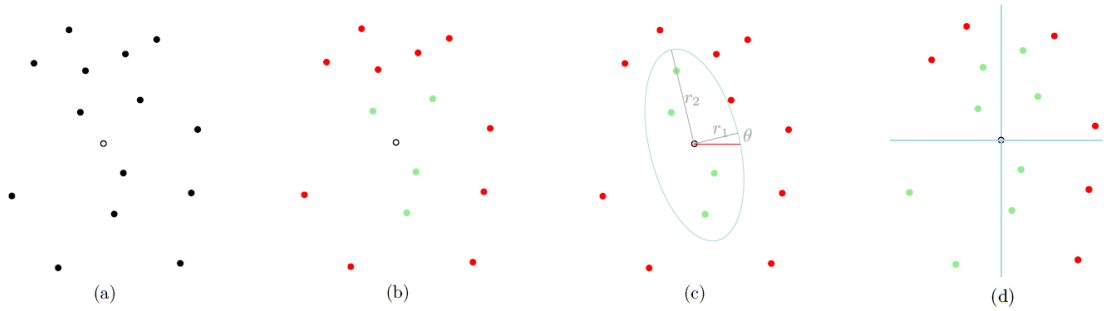


Figure 2.4: IDW variations for (a) a dataset of points and an interpolation location with the (b) k-nearest neighbours method for k=4, (c) the search ellipse and (d) the k-per-quadrant method with k=2 (adapted from Ledoux et al., 2022).

The weight $w_i(x)$ assigned to each data point p_i for a location x is:

$$w_i = \frac{1}{d(x, p_i)^h} \tag{2.2}$$

where h is the power to be used and $d(x, p_i)$ the distance between two points. The power parameter h controls the rate of weight decline with distance. Higher h values prioritize nearby points, leading to a more localized and variable surface. Lower h values give more importance to distant points, resulting in a smoother and globally influenced interpolation (Ledoux et al., 2022).

The IDW interpolation method is accurate, focuses on a specific area, and is computationally efficient. However, it has a significant limitation: it uses a one-dimensional weighting criterion. This means that it only considers the distance between the interpolation point and nearby data points, without taking into account how these points are arranged in space. As a result, this method can sometimes create interpolations that lack smoothness and continuity. This drawback is particularly problematic when using IDW for subsoil schematization (Ledoux, 2022).

Natural Neighbor Interpolation (NatNe)

The natural neighbour interpolation (NatNe) is based on the Voronoi diagram (VD) for both selecting the data points p_i involved in the process (Figure 2.5(a)), and assigning them a weight w_i (Figure 2.5(b)) (Ledoux et al., 2022). Thus it is important to briefly define what is a VD (also called Thiesen polygons). In simple terms, it is a way to divide up a space based on a set of points (Figure 2.5). Each point in a VD has its own region (cell), and every location inside that region is closer to that point than to any other (Lucas, 2021).

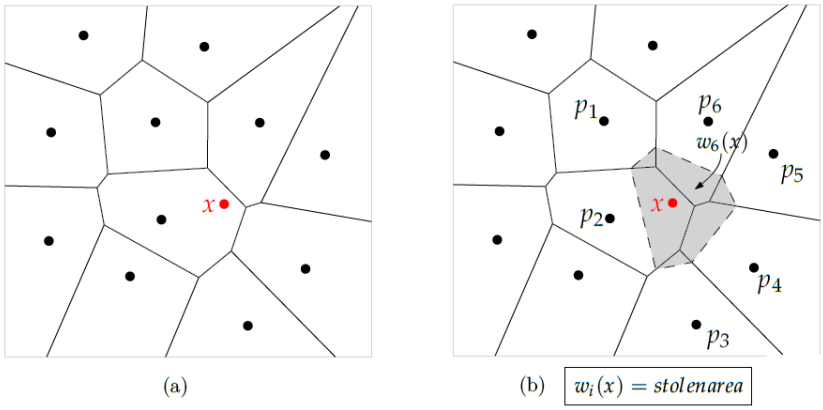


Figure 2.5: (a) The VD of a set of points with an interpolation location x and (b) the NaNI coordinates in 2D for x where the shaded polygon is V_x^+ (adapted from Ledoux et al., 2022).

The neighbours and weights are defined from the two VDs: The first VD establishes the set of data points p_i, and the other one creates a new cell on the location x for the interpolation, virtually stealing some parts from the original cells (Ledoux et al., 2022). Once the neighbours are identified, the next step is to identify the weights (Lucas et al., 2018). The basic equation for the NatNe and the weights is:

$$f(x) = \sum_{i=1}^{k} w_i(x) p_i \tag{2.3}$$

where $f(x)$ is the estimated attribute at location x, w_i are the weights and p_i are the known data points. The weights are calculated as the stolen surrounding areas:

$$w_i(x) = \frac{A(x_i)}{A(x)} \quad and \quad \sum_{i=1}^{k} w_i(x) = 1 \tag{2.4}$$

where w_i are the weights, $A(x)$ is the area of the new cell at location x and $A(x_i)$ is the intersection area between the new cell and the old cell (Lucas et al., 2018). Note that the sum of all the areas stolen from each of the neighbours is equal to 1.0 (Ledoux et al., 2022).

This interpolation method is exact and guarantees smoothness and continuity. It operates locally, using neighbouring samples for estimations which helps in mitigating the spread of errors from any inaccurate samples. Adaptability is one of its strong suits, handling anisotropic data distributions and datasets with variable density proficiently. On the negative side, it lacks the ability to extrapolate, which is commonly done in subsoil schematization, and the implementation is rather complex due to the required manipulation of the VD for better results (Ledoux et al., 2022).

2.3.3. Kriging Interpolation

The Kriging interpolation method (Krige, 1951), is a geostatistical technique for spatial interpolation that in contrast with other methods, involves creating a custom statistical model for every dataset. The basis of Kriging lies in understanding and modelling the spatial trend within the dataset and the spatially correlated randomness; meaning that closer points tend to have more similar values (Ledoux et al., 2022).

As with IDW interpolation, Kriging is calculated as a weighted sum of the data:

$$f(x) = \sum_{i=1}^{N} w_i(x) p_i \tag{2.5}$$

where $f(x)$ is the estimated attribute at location x, w_i are the unknown weights at the ith location and p_i are the known data points for N number of measured values.

In order to determine the weights w_i for this method, not only the distance between the measured points and the prediction location is important, but also the overall spatial arrangement of the measured points. For this, a variogram model needs to be fitted to the measured points p_i. The variogram $\gamma(h)$ is a function that expresses the average dissimilarity of the attribute of a random variable Z between sample points at different distances. In practice, is more useful to compute the experimental variogram (Figure 2.6(a)), which averages the dissimilarity of the pairs of sample points based on distance intervals (Ledoux et al., 2022):

$$\gamma(h) = \frac{1}{2n} \sum (Z(x+h) - Z(x))^2 \tag{2.6}$$

where x is the location of a sample point, h is a vector from x to another sample point, $Z(x)$ is the value of a random variable at x and x is the number of sample point pairs in all the vector for a length interval.

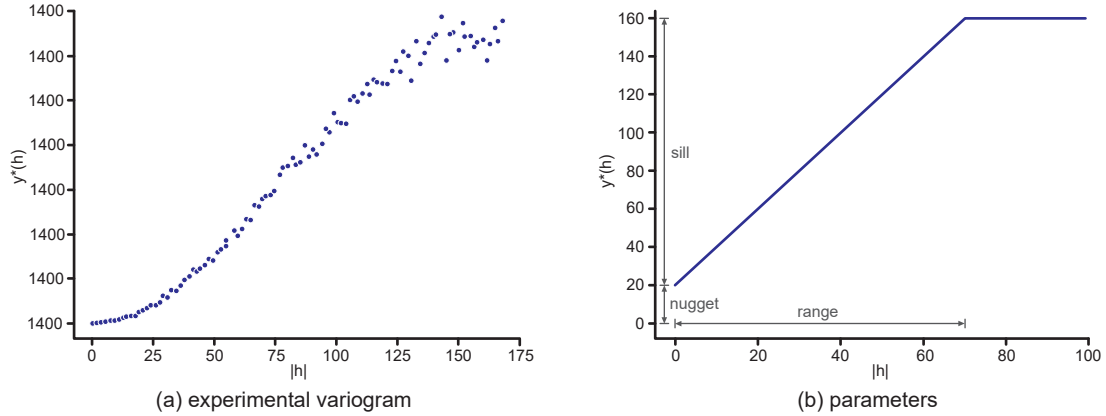


Figure 2.6: (a) The experimental variogram and the (b) parameters used to describe it (Ledoux et al., 2022).

The experimental variogram is based on the sill, range, and nugget parameters (Figure 2.6(b)). They contribute to constructing the theoretical variogram model, which is the foundation for calculating the weights assigned to each data point during Kriging interpolation. The nugget effect is particularly noteworthy as it influences the

weight distribution. A larger nugget typically results in nearby observations receiving more weight. The range sets the limit beyond which observations do not contribute to the prediction due to a lack of correlation. Finally, the sill parameter often helps normalize the semivariances (Ledoux et al., 2022).

Once the variogram model is defined, the Kriging method uses it to weigh the surrounding measured points to derive a prediction for an unmeasured location. The weights are assigned based on the distance to the prediction location, the trend (if any), and the spatial arrangement of data points (Ledoux et al., 2022).

Types of Kriging include simple Kriging, ordinary Kriging, and universal Kriging, each suitable for different kinds of data and situations. Simple Kriging is used when the spatial trend is known, ordinary Kriging when the spatial trend is unknown, and universal Kriging when the spatial trend can be estimated using a deterministic function (Ledoux et al., 2022).

As for benefits, Kriging interpolation is exact (it passes through the given data points) and generates continuous surface, making it a preferable choice for datasets where transitions between points are gradual. The method is highly adaptable as it takes into account the spatial correlation of the data, allowing it to accommodate various forms of spatial continuity. Additionally, it accounts for uncertainty associated with that estimated values, enhancing its utility in decision-making and analysis. However, it also has some limitations. While it excels in local predictions due to its weight assignment based on proximity, this might lead to oversmoothing which will depreciate the quality of subsoil schematizations. Also, defining an appropriate variogram model can be a complex task (Oliver and Webster, 2015; Ledoux et al., 2022).

2.3.4. Inpainting as an Interpolation Technique

Inpainting was initially developed within the realm of image processing to reconstruct lost or deteriorated parts of images and videos as illustrated in Figure 2.7 (Émile-Mâle, 1979; Bertalmio et al., 2000; Jam et al., 2021), and is currently vastly used to perform image corrections. The central premise is to replenish missing data in a manner that is indistinguishable and compatible with the surrounding intact data. Unlike more conventional interpolation techniques which heavily depend on spatial or statistical measures, inpainting gives more weightage to the inherent structures and patterns present within the data (Bertalmio et al., 2000).

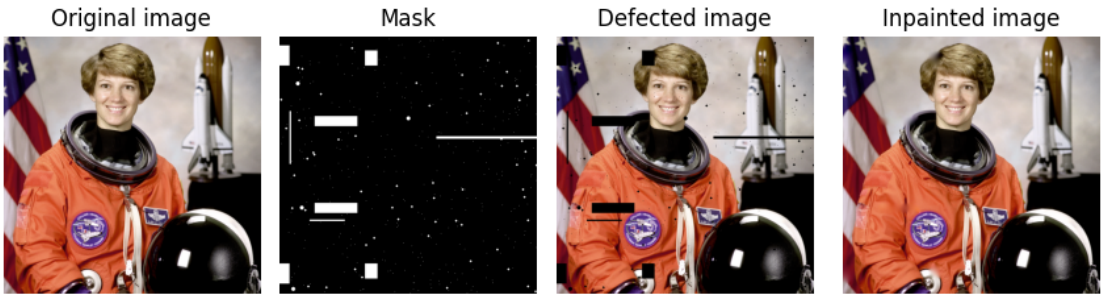


Figure 2.7: Inpainting process of reconstructing damaged images (adapted from “Inpainting”, 2023).

At its most elementary level, inpainting involves using the information from neighbouring pixels to approximate the value of a missing or corrupted pixel (Jam et al., 2021). More advanced implementations of inpainting engage methods involving partial differential equations, most notably the biharmonic equation, to discern the data intrinsic structures and accurately replicate the missing portions (“Inpainting”, 2023).

In short, the biharmonic equation is used in inpainting by treating the missing regions in an image as a diffusion problem, where the idea is to spread colour (value) information from the surrounding intact area into the gaps, similar to how heat diffuses over time. It creates a biharmonic function over the image, satisfying the Laplace equation twice, and uses the known image information as boundary conditions to find a smooth colour transition that fills in the missing or corrupted parts of the image through numerical methods. (Damelin and Hoang, 2018).

2.4. Artificial Intelligence in Geotechnical Engineering

A concise introduction to AI, ML and DL to familiarize the reader with the key concepts relevant to this research is given hereafter. Additionally, an overview of the current utilization of ML techniques in Geotechnical Engineering is presented, emphasizing the commonly employed methods and their application domains. By doing

so, this overview sets the stage for exploring the untapped potential of GANs, which serve as the focal point of this research. Since AI, ML, and DL are extensive subjects on their own, this section aims to provide a foundational understanding rather than an exhaustive exploration. To delve deeper into each aspect, references will be provided for interested readers.

2.4.1. What is Artificial Intelligence?

AI is a science that studies the development of computer systems that can creatively solve problems by imitating human intelligence (Goodfellow et al., 2016; Nguyen et al., 2019 Zhang et al., 2021). It involves the simulation of intelligent behaviour by machines, enabling them to perceive, reason, learn, and make decisions (Campbell et al., 2021).

ML is a subset of AI that focuses on algorithms and statistical models that allow computers to automatically learn and improve from experience without being explicitly programmed (Nguyen et al., 2019; Zhang et al., 2021). ML algorithms enable computers to identify patterns, extract meaningful insights from data, and make predictions or decisions based on learned patterns (Campbell et al., 2021).

DL is a specific subfield of ML that is inspired by the structure and function of the human brain. It utilizes ANN with multiple layers to process and learn from complex data representations (Nguyen et al., 2019), without manually extracting features (Goodfellow et al., 2016; Zhang et al., 2021). DL algorithms excel at automatically extracting hierarchical patterns and features from data, enabling them to do image and speech recognition, natural language processing, and other complex learning tasks (Goodfellow et al., 2016).

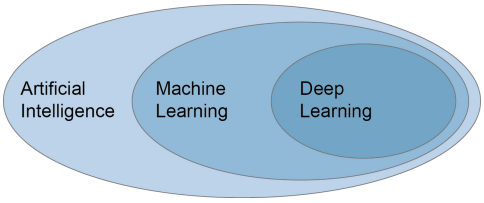


Figure 2.8: Venn diagram showing the relationship between AI, ML and DL (adapted from Goodfellow et al., 2016).

2.4.2. State-of-the-art review of ML in Geotechnical Engineering

Geomaterials exhibit one of the most intricate sets of physical, mechanical, and chemical behaviours compared to other engineering materials. Unlike construction and manufacturing materials, geomaterials exhibit a highly nonlinear response. Additionally, soil and rock possess inherent anisotropic and heterogeneous properties, due to their formation nature. This substantial level of diversity poses challenges in studying and predicting their behaviour, limiting the development of analytical and numerical solutions for certain problems (Baghbani et al., 2022). As a result, ML methods have gained popularity in the geotechnical field, as they have the capacity to capture correlations among information without relying on prior assumptions (Zhang et al., 2021; Phoon and Zhang, 2022).

Baghbani et al. (2022) presents four arguments on why AI modelling methods have captured the attention of geotechnical engineers as a promising alternative: (i) AI methods can model complex and nonlinear processes without requiring initial assumptions about the relationships between input and output variables, (ii) AI techniques have diverse applications and (iii) can accurately predict outcomes, even when the physical relationships between parameters are unknown, and (iv) they are capable of analyzing large amounts of data, identifying patterns, and even generating incomplete data in certain cases.

Baghbani et al. (2022) and Phoon and Zhang (2022) offer a very complete overview of the current state-of-the-art in AI for Geotechnical Engineering. As AI methods, they consider Artificial Neural Networks (ANN), Fuzzy Interference Systems (FIS), Adaptive Neuro-Fuzzy Interference Systems (ANFIS), Support Vector Machines (SVM), Long Short-term Memory (LSTM), Convolutional Neural Networks (CNN), Residual Neural Networks (ResNet) and Generative Adversarial Networks (GAN).

Figure 2.9 shows the number of studies using AI with different algorithms in different geotechnical fields. The highest number of studies belong to site characterization, specifically in the fields of rock mechanics, landslide and liquefaction and tunnelling and TBM, while the least number of studies are in the areas of unsaturated soils, frozen soils and thermal properties, and subgrade soil and pavement. Furthermore, the most used ML algorithm is the ANN, followed by SVM (Baghbani et al., 2022; Phoon and Zhang, 2022).

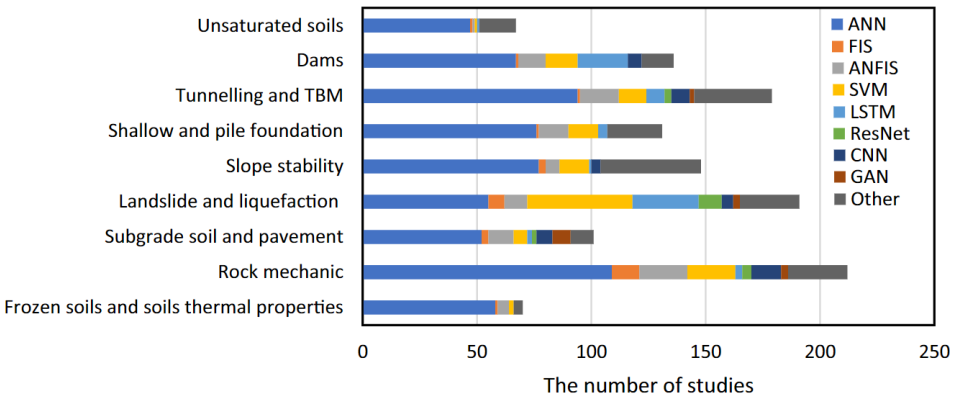


Figure 2.9: Distribution of the use of AI in Geotechnical Engineering (adapted from Baghbani et al., 2022).

A further breakdown of the application of DL methods in Geotechnical Engineering is given in Zhang et al. (2021). They identify four DL methods as the most used in the field of geotechnics: Feedforward Neural Networks (FNN), Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN) and Generative Adversarial Networks (GAN). Figure 2.10 gives a breakdown of the different architectures applied in different geotechnical subjects. Among all methods, GAN is the least used, although as an unsupervised learning algorithm, it offers great potential due to its generating ability (Zhang et al., 2021).

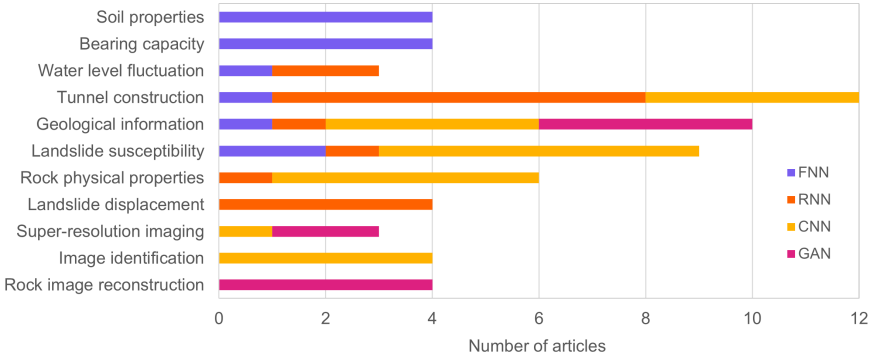


Figure 2.10: Main DL methods applied to specific topics of Geotechnical Engineering (adapted from Zhang et al., 2021).

2.4.3. GAN applications in Geotechnical Engineering

Despite being one of the less frequently employed DL methods in the field of Geotechnical Engineering, GANs have gained traction since their introduction by Goodfellow et al. (2014) just under a decade ago. During this relatively short span of time, a few geotechnical applications have been developed utilizing GANs.

Mosser et al. (2017) applied a GAN to the reconstruction of the solid-void structure of porous media, matching key characteristic statistical and physical parameters. Following, Valsecchi et al. (2020) established a GAN for two-dimensional to three-dimensional structure of porous media, which allows reconstructing 3D models from sets of 2D images, resulting in a huge advantage in terms of applicability with respect to the cost of microcomputed tomography scans. Azevedo et al. (2020) showed that GAN is capable of generating geological models of discrete and continuous properties for stochastic subsurface model reconstruction.

In the task of image processing, Chen et al. (2020) implemented a CycleGAN real-world rock microcomputed tomography image, which can model the mapping of these tomography images at different resolutions. Using conditional GANs, Janssens et al. (2020) proposed a method to solve the fluid flow characteristics assessment with more visually appealing results. Kang and Choe (2020) used GANs to explore reservoir properties in combination with existing geological information, and Oliveira et al. (2019) evaluated the performance of GANs as an interpolation tool for improving seismic data resolution; showing that it outperforms traditional algorithms.

2.5. Generative Adversarial Networks

GANs have emerged as a significant development in the field of DL since their introduction by Goodfellow et al. (2014), with numerous advancements and variations over the years (Tomczak, 2022). These incluse Conditional GANs (cGAN) by Mirza and Osindero (2014), Deep Convolutional GANs (DCGAN) by Radford et al. (2015), Wasserstein GANs (WGAN) by Arjovsky et al. (2017), Progressive GANs (P-GAN) by Mahapatra and Bozorgtabar (2019), CycleGAN by Zhu et al. (2017), StarGAN by Choi et al. (2017), StyleGAN by Karras et al. (2018), and Pix2Pix by Isola et al. (2017), among others. These variants have expanded the possibilities of GAN applications and improved their performance in specific domains (Foster, 2019). In the context of this thesis, particular focus is given to the image-to-image translation architecture known as Pix2Pix (Isola et al., 2017), which leverages cGANs to learn mappings between input and output images.

This section aims to provide a concise overview of the fundamental concepts of GANs in order to facilitate the reader's understanding of the proposed model, the training process, and its implementation in the following chapters. For a more comprehensive understanding, the original publications by Goodfellow et al. (2014) and Goodfellow (2017) serve as the recommended starting point.

2.5.1. How do GANs work?

The basic idea behind GANs is to set up a game-like scenario involving two players: the Generator and the Discriminator. This framework, proposed by Goodfellow et al. (2014), enables the generation of fake samples that closely resemble real data by training the Generator and Discriminator models in an adversarial manner.

The Generator's primary role is to create fake samples intended to come from the same distribution as the training data. It takes random noise as the initial seed and employs a Deep Neural Network architecture to transform this input noise into meaningful output data. During training. the Generator learns to generate samples that are increasingly realistic and closely resemble the training data distribution (Goodfellow, 2017).

On the other hand, the Discriminator plays the role of a binary classifier, tasked with differentiating between real and fake samples. It receives input data, which can be either real or generated by the Generator, and predicts the probability of the input being real. Similar to the Generator, the Discriminator is implemented using a Deep Neural Network architecture. Through training, the Discriminator learns to accurately classify the input as either real or fake (Goodfellow, 2017).

To better understand this concept, we can use the analogy of a counterfeiter and the police as presented in Figure 2.11. The Generator, akin to a counterfeiter, aims to produce fake money (samples) that is indistinguishable from real money and aims to continuously improve its ability to create realistic counterfeited money that the Discriminator cannot easily differentiate from the real one. Conversely, the Discriminator plays the role of the police, which by examining the input samples, attempts to correctly classify them as either real or fake. Through training, the Discriminator becomes more skilled at accurately classifying the samples, thus posing a challenge to the Generator (Goodfellow et al., 2014).

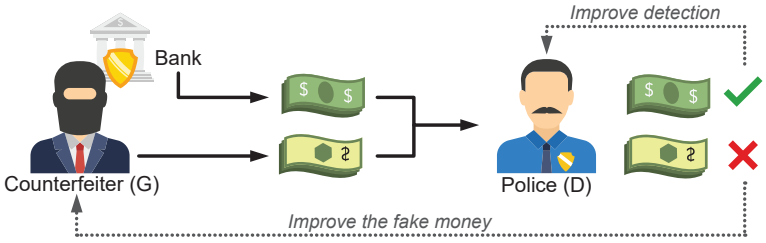


Figure 2.11: Counterfeiter and police analogy for the Generator and Discriminator in the GAN architecture.

Through an iterative process, the Generator and Discriminator engage in a competition. The Generator strives to generate samples that the Discriminator cannot distinguish from real data, while the Discriminator aims to improve its ability to correctly classify the samples. This adversarial interplay illustrated in Figure 2.12 drives the training process, leading to the refinement of both models.

2.5.2. The General Adversarial Training Algorithm

The goal of the training process of GANs is to minimize the Discriminator's ability to distinguish between real and fake samples while maximizing the Generator's capability to produce realistic samples. The objective function

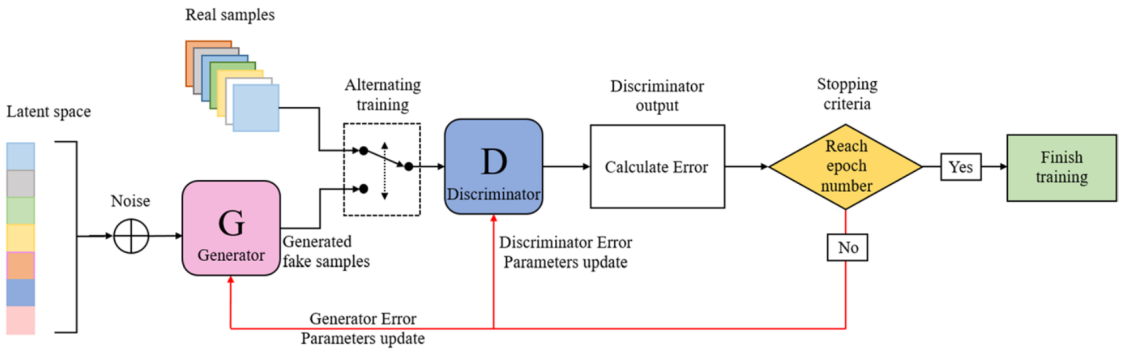


Figure 2.12: Schematic diagram of the GAN training (adapted from Azevedo et al., 2020).

$\mathcal{L}_{GAN}(D, G)$ represents the two-player minimax game played by the Discriminator (D) and the Generator (G). It is formulated as follows:

$$\min_G \max_D \mathcal{L}_{GAN}(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))] \tag{2.7}$$

In this formula, D represents the Discriminator, G represents the Generator, x denotes real data samples, z denotes random noise or latent input to the Generator, $p_{data}(x)$ represents the real data distribution, and $p_z(z)$ represents the prior distribution over the latent space.

During the training process, the Discriminator and Generator are updated iteratively. First, the Discriminator is trained on a batch of real and fake samples, adjusting its parameters to improve its ability to classify them correctly. The goal of the Discriminator is to output the probability that its input is real rather than fake. In scenarios where the input is a real sample, the Discriminator aims for $D(x)$ to be near 1, indicating a high probability that the input is real (Goodfellow et al., 2014; Goodfellow, 2017).

Next, the Generator is trained using the updated Discriminator. It generates fake samples by randomly sampling inputs z from the prior distribution over the latent variables. The Discriminator then receives the generated sample $G(z)$, a fake sample created by the Generator. In this scenario, both players participate. The Discriminator strives to make $D(G(z))$ approach 0, indicating a high probability that the generated sample is fake, while the Generator strives to make $D(G(z))$ approach 1, indicating a high probability that the generated sample is real (Goodfellow et al., 2014; Goodfellow, 2017).

This iterative back-and-forth process continues, with the Discriminator and Generator gradually improving their performance and converging towards an equilibrium. As the training progresses, the Generator learns to produce fake samples that closely resemble the real data distribution, while the Discriminator becomes more adept at distinguishing between real and fake samples. The iterative competition between the Generator and Discriminator drives the training process, updating their weights based on their performance in the minimax game, resulting in the refinement of both models and the generation of high-quality fake samples (Goodfellow et al., 2014; Goodfellow, 2017).

2.5.3. Binary Cross-entropy Loss (BCE)

The adversarial loss function employed in the traditional GAN model follows a two-player minimax game dynamic, with the loss typically computed using Binary Cross-Entropy (BCE). BCE, also referred to as ”log loss”, is frequently employed in binary classification tasks, where each sample is assigned to one of two mutually exclusive classes. It quantifies the divergence between the true class labels and the predicted probabilities associated with each class. In essence, it assesses the quality of the prediction probabilities by measuring how well they correspond to the true classes.

Mathematically, for a single data point, the BCE loss is defined as:

$$L(y, p) = -y \cdot \log p - (1 - y) \cdot \log(1 - p) \tag{2.8}$$

where y is the true class label, which can take the values 0 or 1, representing the two classes and p represents the model’s predicted probability that the class label is 1.

This function behaviour is depicted in Figure 2.13. If the true class label is real ($y = 1$) the second term cancels out and the loss is equivalent to $-\log(p)$, penalizing predictions close to 0 with a higher loss. Conversely, if the true class label is fake ($y = 0$) the first term cancells out and the loss is equivalent to $-\log(1-p)$, which assigns a high loss value when the prediction p approaches 1. The BCE function is designed to impart a higher penalty when a prediction is highly confident but incorrect, while less confident, incorrect predictions are met with a smaller penalty. This aspect of the BCE function motivates the Discriminator to generate accurate predictions with high confidence. Consequently, the Generator is driven to generate more realistic images, in an attempt to fool the Discriminator, thus enhancing the overall efficacy of the GAN model.

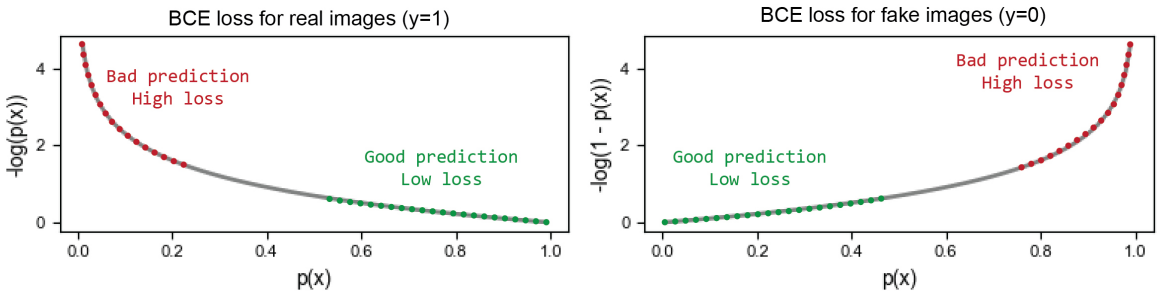


Figure 2.13: Binary Cross-entropy loss for binary classification.

2.5.4. Common Failures in GANs

During the training of GANs, several common failures and challenges can arise, impacting the performance and reliability of the models (Goodfellow, 2017). It is essential to understand and address these issues to ensure the effectiveness of the method. The most common failures encountered in GAN training are overfitting, mode collapse, vanishing gradients, and training instability (Goodfellow et al., 2016; Goodfellow, 2017).

Overfitting occurs when the GAN model becomes overly specialized to the training data, resulting in poor generalization to new or unseen data. To mitigate overfitting, regularization techniques such as dropout and weight decay can be employed, along with early stopping and data augmentation strategies (Goodfellow et al., 2016).

Mode collapse occurs when the Generator produces limited variations of the desired output, ignoring the full complexity of the data distribution. It can manifest as the generation of similar outputs with minimal diversity (Bau et al., 2019). To address mode collapse, architectural modifications, diversity-promoting techniques, and alternative loss functions can be explored (Li et al., 2017; Bau et al., 2019).

Vanishing gradients can impede GAN training by hindering the flow of gradients through the network, making it difficult for the model to learn and converge effectively (Arjovsky et al., 2017). This issue is particularly relevant in deep architectures where the gradients can diminish exponentially as they propagate through multiple layers. Techniques such as gradient clipping, alternative activation functions (LeakyReLU), and appropriate weight initialization strategies can help mitigate the problem (Li et al., 2017).

Training instability refers to the erratic behaviour and lack of convergence observed during GAN training, where the generator and discriminator fail to reach a stable equilibrium (Mescheder et al., 2018). This instability can arise due to the sensitivity of GANs to hyperparameter settings, the imbalance between the Generator and Discriminator capacities, and the complexity of the task. To address training instability, strategies such as learning rate scheduling, batch normalization, and the use of adaptive optimization algorithms (e.g. Adam optimizer) can be employed to stabilize the training process and promote convergence (Gulrajani et al., 2017; Goodfellow, 2017; Mescheder et al., 2018).

2.6. Pix2Pix: Image-to-Image Translation

The Pix2Pix architecture has been selected as the optimal approach for subsoil schematization, aligning with the research objective of this thesis. Pix2Pix is an advanced image-to-image translation framework specifically designed for tasks involving the translation of input images into corresponding output images (Isola et al., 2017). This offers the unique capability of using CPT data as conditional input and aligning it with the desired output of a complete cross-section interpretation.

2.6.1. The Conditional GAN training algorithm

The objective of the cGAN is to generate samples that are not only realistic but also conditioned on the provided input (Mirza and Osindero, 2014). In the cGAN framework, the Generator network (G) generates outputs based on the provided conditioning input, while the Discriminator network (D) distinguishes between real and generated samples. The algorithm is formulated to find a balance where the Generator can generate realistic and conditioned samples, while the Discriminator can accurately classify between real and generated samples (Isola et al., 2017). This method combines the general training algorithm of GANs (Goodfellow et al., 2014) with the inclusion of conditioning input (Isola et al., 2017):

$$\min_G \max_D \mathcal{L}_{cGAN}(D,G) = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z)))] \tag{2.9}$$

In this formulation, x represents the input data which at the same time serves as the condition or context to generate output data, y represents the corresponding ground truth (target data) that the Generator aims to generate, and z is a random noise vector. The Generator (G) takes both the random noise z and the conditioning input x as inputs to produce the generated samples $G(x,z)$.

The Discriminator (D) estimates two probabilities: First, it estimates the probability that the image pair (x,y) is real. Second, it estimates the probability that the generated output $G(x,z)$ is fake. The Discriminator aims to assign a high probability to real samples and a low probability to generated samples (Mirza and Osindero, 2014; Isola et al., 2017; Goodfellow et al., 2016). When it encounters a real sample, it should output a value close to 1, indicating high confidence in its authenticity. On the other hand, when the Discriminator encounters a generated sample, it should output a value close to 0, indicating high confidence that it is fake.

During the training process, the Generator and Discriminator iteratively update their parameters based on their respective objectives. The Generator aims to minimize the log probability of the Discriminator correctly classifying the generated samples as fake ($\log(1 - D(x, G(x,z)))$). This encourages the Generator to generate samples that have a high probability of being classified as real. Conversely, the Discriminator aims to maximize the log probability of correctly classifying real samples ($\log(D(x,y))$) and the log probability of correctly classifying generated samples as fake ($\log(1 - D(x, G(x,z)))$) (Mirza and Osindero, 2014; Isola et al., 2017; Goodfellow et al., 2016). This competition between the Generator and Discriminator leads to a dynamic equilibrium where the Generator produces increasingly realistic samples and the Discriminator becomes more effective at distinguishing between real and generated samples.

2.6.2. The Network Architectures

Pix2Pix utilizes specific network architectures for its Generator and Discriminator models: the U-Net for the Generator and the PatchGAN for the Discriminator. An illustration of their general behaviour is presented in Figure 2.14.

The U-Net (Ronneberger et al., 2015) is widely used for various image-to-image translation tasks. It consists of an encoder-decoder structure with skip connections. The encoder gradually downsamples the input image, capturing high-level features, while the decoder upsamples the learned features, generating the output image. The skip connections tie the corresponding encoder and decoder layers, allowing the transfer of detailed information. This enables the Generator to maintain both global structure and fine-grained details in the generated output (Ronneberger et al., 2015; Isola et al., 2017).

On the other hand, the Discriminator utilizes a PatchGAN architecture (Demir and Unal, 2018) designed to assess the local image patches instead of the entire image. Instead of producing a single scalar output, the PatchGAN Discriminator outputs a matrix of probabilities, where each value represents the likelihood of a small patch being real or fake. By analyzing image patches, the Discriminator can provide more detailed feedback to the Generator, enabling it to generate visually convincing and contextually relevant outputs (Isola et al., 2017; Demir and Unal, 2018).

2.7. Insights from the Literature

Subsoil schematizations play a crucial role in geotechnical engineering, providing a comprehensive understanding of subsurface conditions that influence various aspects of construction, design, cost estimation, risk management, and environmental considerations (Cosenza et al., 2006; de Rienzo et al., 2008; Das, 2010; Reiffsteck et al.,

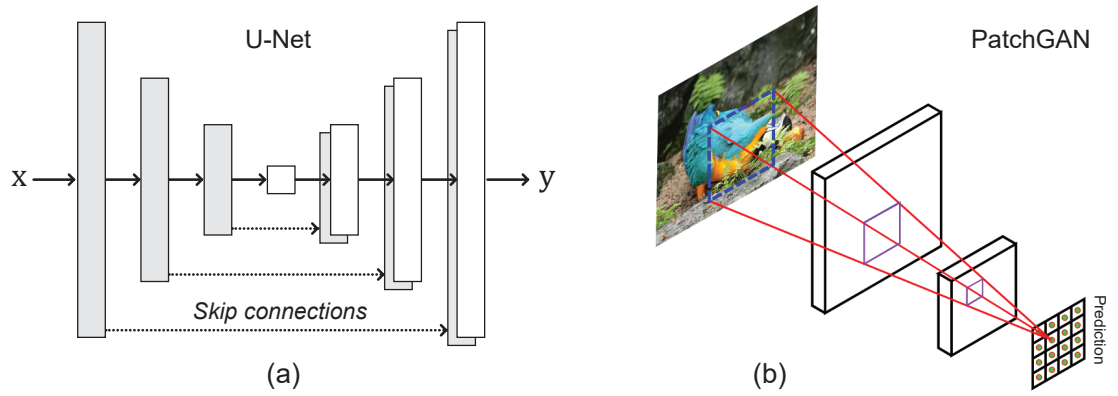


Figure 2.14: Architecture of the (a) Generator with a "U-Net" as an encoder-decoder with skip connections between mirrored layers (adapted from Isola et al., 2017), and the (b) Discriminator as a PatchGAN binary classifier (adapted from Demir and Unal, 2018).

2018; Dawoody, 2021). Despite the utilization of geostatistical tools, translating sparse CPT data into complete cross-sections remains challenging and often requires expertise akin to an art form.

Existing geostatistical interpolation techniques present advantages and limitations, particularly related to data continuity and smoothing, when applied to subsoil characterization (Ledoux et al., 2022).

In recent years, the application of Artificial Intelligence (AI) in geotechnical engineering has gained momentum, primarily focusing on rock mechanics, slope stability, and site characterization, with Artificial Neural Networks being the most commonly employed technique (Baghbani et al., 2022; Phoon and Zhang, 2022). However, GANs have received relatively less attention in geotechnics (Zhang et al., 2021), and there is a lack of previous research on AI-based approaches for geotechnical subsoil schematization. While GANs have found applications in rock mechanics and imaginary simulations related to porosity and flow mechanics, their potential in subsoil schematization remains untapped (Mosser et al., 2017; Azevedo et al., 2020; Chen et al., 2020; Janssens et al., 2020; Kang and Choe, 2020; Valsecchi et al., 2020). Some efforts have also been made to enhance seismic data resolution (Oliveira et al., 2019).

GANs excel at comprehending complex datasets and learning the intricate relationships inside them. This is useful when considering, layering and spatial arrangements within subsoil schematizations. Furthermore, frameworks like the Pix2Pix enable GANs to integrate contextual information, ensuring that generated schematizations align with specified attributes. Comparatively, traditional ML techniques like ANN, SVM, and CNN are adept at classification and pattern recognition tasks. However, GANs outshine them in capturing complex spatial relationships intrinsic to subsoil schematization, offering a distinct advantage.

The Image-to-Image Translation framework has been identified as a promising solution for subsoil schematization. This approach allows for the conditioning of generated outputs based on the arrangement of CPT data, enabling the accurate interpretation of CPT data into cross-sections (Isola et al., 2017). By training the Pix2Pix architecture under this premise, it holds the potential to generate precise and reliable interpretations of CPT data for subsoil schematization.

3

Materials and Methods

3.1. Introduction

This chapter outlines the most relevant tasks followed in order to answer the research questions as illustrated in Figure 3.1; starting with the initial Literature Review study to set the foundation knowledge into ML and subsoil schematization. The method for generating accurate schematization databases is outlined, as well as the pre-processing procedure and the construction of the Generator, Discriminator and training architectures. The validation and testing of the models is described finally arriving at the final SchemaGAN model. Finally, a description of the methods for the comparative analysis with traditional interpolation methods, the expert criteria survey process and the logic for applying the model to a real case scenario are given.

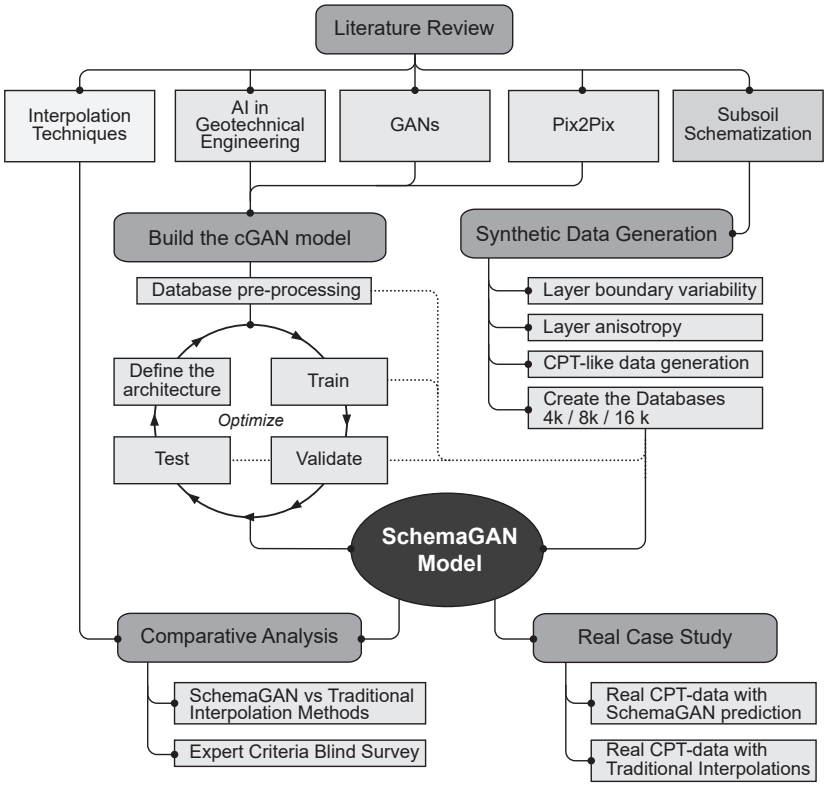


Figure 3.1: General overview of the methodology to build SchemaGAN and test, compare and apply the model.

3.2. Literature Foundations

This section provides an overview of the fundamental literature that establishes the basis for the core tasks undertaken in this thesis. Five key areas of research (Figure 3.1) were examined prior to initiating the primary study objectives. The initial focus centered on subsoil schematization and the principles that govern this practice. Following this, a concise survey of traditional interpolation techniques was conducted. Subsequently, a systematic exploration of ML was conducted, with a specific emphasis on the integration of Artificial Intelligence in the domain of Geotechnical Engineering. This included a comprehensive analysis of GANs, followed by an in-depth assessment of the Pix2Pix framework.

3.3. Subsoil Synthetic Database Generation

The training databases played a crucial role in training the selected DL algorithm. While the Netherlands has a wealth of available CPT records exceeding 180,000 (Zuada-Coelho and Karaoulis, 2022), these records provide only 1D information, and the ground truth between them is unknown. Therefore, the generation of synthetic data emerged as a viable alternative for training purposes.

As discussed in Chapter 2 of the Literature Review, subsoil geometry complexity adhered to underlying rules that needed to be accurately captured in the synthetic data. To achieve this, a Python tool was developed to generate random cross-sections of a specified size. In general, to approximate a realistic geotechnical cross-section, a ratio of 1:16 length over depth was proposed, resulting in images of size 32x512 px which strike a balance between small enough to be computationally efficient and large enough for visual fidelity.

The tool incorporated the ability to create a variable number of layers with geologically realistic behaviour. Additionally, the synthetic models reflected the inherent variability of the material within each layer.

Once the tool was finalized, three databases of increasing size were generated. The process was performed with a known seed to ensure reproducibility. Furthermore, the database was divided into distinct subsets for training, validation and testing, with the user having the flexibility to determine the percentage allocation of the generated data to each subset.

3.4. The Conditional Generative Adversarial Network Model

The DL employed in this study was based on the Pix2Pix framework (Isola et al., 2017). The architecture consisted of the Generator and Discriminator networks, as well as the training algorithm, which were crucial components for the successful implementation of the model (see Section 2.6).

The implementation of a GAN typically involved four main stages, as presented in Figure 3.2. First, the database was pre-processed. Second, the GAN architecture was constructed. Third, the GAN was trained using the pre-processed data and the constructed model, with iterative optimization. Finally, the trained GAN model was evaluated.

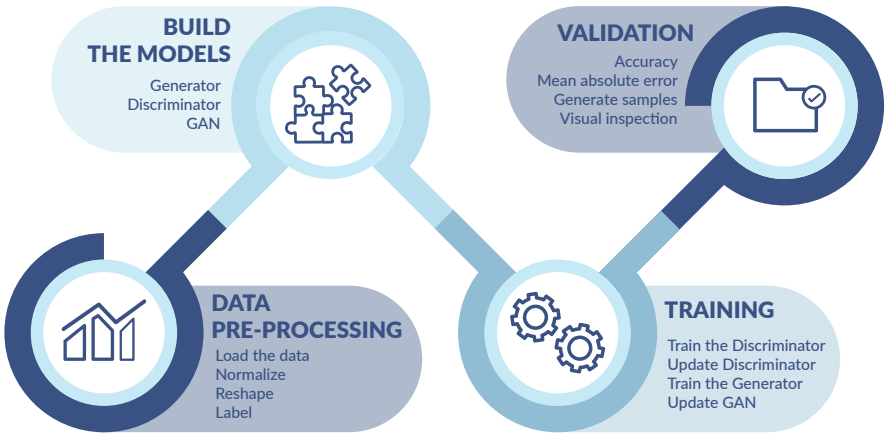


Figure 3.2: Main stages in the implementation of a GAN.

3.4.1. Database Pre-processing

The first step to ensure the data had the correct format before the synthetic databases could be used to train the GAN model, was to split the synthetic database into training, validation and testing subsets. This was done using a ratio of 4:1:1 for the generated samples, with the largest database consisting of 16,000 samples for training, and 4,000 samples for validation and training each.

Next, each synthetic cross-section underwent further processing to generate the associated conditional input. For the purpose of the research, the conditional input was created to resemble CPT-like data. This involved deleting a defined percentage of columns from the cross-section to simulate CPT measurements. Additionally, a random number of rows were deleted from the remaining data to account for variations in depth. These actions collectively generated a CPT-like survey image, which served as the conditional input for training the GAN model, as illustrated in Figure 3.3.

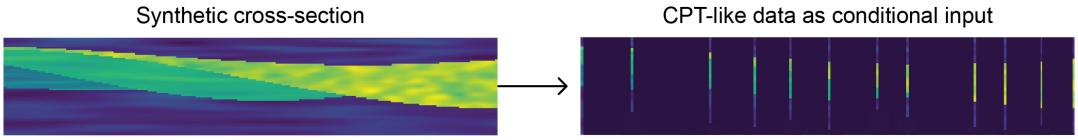


Figure 3.3: Creation of the conditional input from the original synthetic cross-sections.

The geotechnical values represented in the generated synthetic cross-sections were then normalized, and the pre-processed data was reshaped and organized into pairs, with each synthetic schematization paired with its corresponding conditional input (CPT-like data). This pairing was crucial for the training process, as it established the conditional relationship between the input and output data.

3.4.2. The Generator and Discriminator Architectures

To facilitate the generation of accurate subsoil schematizations conditioned on the input data, the Generator and Discriminator architectures were modified for the task at hand. The original Pix2Pix Generator and Discriminator, designed for images of size 256x256 px, served as the foundation for the new model architectures to generate and discriminate images of 32x512 px size.

The networks underwent reconfigurations, including changes to the number of filters and layers, as well as modifications to the stride size and kernel size. These modifications were crucial for creating new U-Net and PatchGAN architectures capable of effectively handling the larger schematization images.

The combination of the modified U-Net Generator and the PatchGAN Discriminator formed the core of the SchemaGAN model for subsoil schematization. This architecture was specifically designed to learn the association between the CPT-like data and the corresponding cross-section interpretations, enabling the generation of realistic and conditioned subsoil schematizations.

3.4.3. The Training Algorithm

After finalizing the construction of the SchemaGAN architecture, the next step was to develop the training algorithm. This involved setting the batch size, determining the number of training epochs, and initializing the evaluation metrics to monitor the progress and performance of the training process.

During training, the losses and accuracy were tracked both for each batch of data processed and for each epoch completed. These metrics served as indicators of the SchemaGAN learning progress and its ability to generate accurate and realistic subsoil schematizations. Snapshots of the generated images as the training progressed were also qualitatively examined. After each training epoch, the SchemaGAN Generator model was saved for later validation and testing.

The training algorithm employed the Adam optimizer, which adaptively adjusted the learning rate for each parameter during training to optimize the learning process and improve the convergence of the model (Zhang, 2018). Additionally, to enhance the performance and stability of the model, batch normalization and dropout techniques were utilized. Batch normalization normalized the output of each layer within a mini-batch, reducing the internal covariate shift and improving training speed and stability (Santurkar et al., 2018). Dropout randomly set a fraction of the input units to zero during each training step, preventing overfitting and enhancing the model's generalization ability (Srivastava et al., 2014).

3.4.4. Validation and Testing process of SchemaGAN

Once the training is concluded, the saved Generators from the SchemaGAN model were validated using a dedicated database of unseen samples. Each Generator model is used to generate unique cross-sections from the validation database, from which a quantitative and qualitative evaluation is carried out.

Both the evaluation metrics and the visual inspection of the generated schematizations helped in detecting and correcting the common GAN failures. From the results of the validation process, the best performing SchemaGAN Generator is chosen to be tested and optimized until the final model is ready to be used in the comparison with the traditional interpolation methods and the real case study. For this final step, a unique testing database of unseen samples was used.

3.5. Comparative Analysis

In order to compare the performance of SchemaGAN against traditional interpolations methods a unique testing database of 4,000 unseen samples was used to generate schematizations with SchemaGAN and all the interpolations methods to be evaluated both quantitatively and qualitatively. Furthermore, an expert criteria survey was done to rank the methods from worst to best, and the influence of the location of the CPT data was studied Finally, a real case study was performed using SchemaGAN and the best interpolation methods.

3.5.1. Data Selection and Preparation

In order to prepare the synthetic schematizations for the traditional interpolations, first, the columns containing CPT-like information were extracted. This captured both the position and value of the columns, providing the input for the interpolation methods. Second, a 2D grid was generated, matching the 32x512 px image size used by SchemaGAN. The coordinates from this grid were used as the locations for subsequent interpolation.

3.5.2. Implementation of the Traditional Interpolation Methods

A comprehensive overview of the interpolation methods used in this project has been provided in the Literature Review in Section 2.3. The chosen techniques of NeaNe, IDW, Kriging, NatNe and Inpainting were applied by a Python tool developed using packages from SciPy (Virtanen et al., 2020) and Sci-kit learn (Pedregosa et al., 2011).

3.5.3. Quantitative and Qualitative Analysis

For the quantitative evaluation, the Mean Absolute Error (MAE) and the Mean Squared Error (MSE) were used. The MAE is easy to interpret and provides robustness against outliers, while the MSE amplifies larger discrepancies which are useful to identify areas with high error. Additionally, computational efficiency was also considered, taking into account their practicality and efficiency in real-world geotechnical engineering projects.

A qualitative comparison focusing on visual descriptions and comparisons was also performed to provide a more intuitive understanding of the differences between methods. Visualizations allowed for a comprehensive analysis of the quality, coherence, and spatial patterns captured by each method.

3.5.4. Expert Criteria Survey

Inspired by Ravuri et al. (2021), a blind survey was carried out to get the expert criteria on which method performed best. For this, 10 synthetic schematizations were chosen ranging from simple to complex geometry (see Appendix A). For each cross-section, random CPT-like data was generated and the interpretation was generated using SchemaGAN and the interpolation methods. These schematizations were ranked by experts blindly, according to which methods produced the most realistic results and added the most geotechnical value.

3.5.5. Effect of the CPT location in the conditional input

In order to study the effect of the location of the CPT-like data in the schematization generation for both SchemaGAN and the traditional interpolation methods, two distinct cross-sections were chosen: a simple geometry and a complex geometry. Each one was generated 1,000 times with the location of the CPT data chosen at random each time. The variability and average MAE results are then computed for each model.

3.5.6. The Real Case Study

A real case study was conducted to assess the applicability of SchemaGAN in interpreting real CPT data. The Delft-Schiedam rail embankment was chosen due to the availability of CPT data and the linearity of the project in order to create a 2D cross-section.

Pre-processing of the real CPT data was performed to prepare it for input into SchemaGAN. First, reshaping and resampling the data was necessary for it to fit the 32x512 px format. Additionally, to facilitate data handling, the top portion of the surveys was trimmed, taking into account the different elevations of the CPT data according to the NAP.

Two cross-sections were constructed with the available CPT data. For each schematization two groups of CPTs were made: data as input for the interpolations and data saved for comparison against the generated results. A script was developed to choose at random 6 CPTs for each cross-section while complying with realistic geotechnical spacings. The remaining CPTs were saved for comparison. The script was designed to take into account all possible combinations of the two data groups.

Schematizations were generated with SchemaGAN and the best-performing traditional interpolation methods. To assess the performance MAE and MSE were used in a 1D analysis at the exact points in which the comparison CPTs were located.

3.6. Final Thoughts on Methodology

This chapter provided a comprehensive overview of the methodology employed to achieve the research objective of this investigation. The process began with an explanation of the logic for the generation of synthetic data with geologically realistic behaviour. The subsequent pre-processing steps to ensure the appropriate format and conditioning of the synthetic databases, including the generation of conditional inputs resembling CPT-like data were also addressed.

The consideration made to create the SchemaGAN model, specifically the reconfiguration of the U-Net Generator and PatchGAN Discriminator were explained. This was crucial for adapting the model to handle the larger schematization images. The steps to set up the training algorithm, incorporate the evaluation metrics and include the optimization techniques were also outlined.

The comparison approach with traditional interpolation methods was presented, including a quantitative and qualitative analysis from a testing database, and additional analysis like the expert criteria survey, the CPT location analysis and the real case study.

4

Synthetic Data Generation

4.1. Introduction

This chapter looks into the process of synthetic data generation within the context of subsoil schematizations for geotechnical investigations. The first part of the chapter explores initial considerations for data schematization and provides an overview of the Soil Behaviour Type Index (Ic). Further, the assumptions integral to synthetic subsoil schematization are examined. The subsequent sections detail the simulation of various aspects of the subsoil, including layering variability, heterogeneity, and the emulation of CPT geotechnical campaigns. The chapter concludes with an examination of the databases generated.

4.2. Initial Considerations About the Schematization

The chosen approach for the schematizations in this research project is to classify the layers based on their soil type. The decision was made considering that soil type classification is a fundamental application of CPT data (Robertson, 2010; Robertson and Cabal, 2022). Furthermore, regardless of the geotechnical problem at hand, the classification of soil type is an essential preliminary step in geotechnical investigations, providing a basis for further analysis and more complex modelling.

Given that CPTs are the most commonly used in-situ tests for subsoil characterization in the Netherlands, it was logical to leverage a parameter that can translate CPT measurements into soil types. The Ic value precisely serves this purpose, assigning a numerical value to represent the expected soil type at a specific test location. Table 4.1 illustrates how Ic values associate with different soil types.

It is important to note that while the subsequent procedures and outcomes of applying GANs to subsoil schematization are based on Ic values, the technology itself can be extended to any parameter that can be represented as a single measurement. By translating such parameters into pixel representations and employing normalization techniques, GANs can effectively learn and generate synthetic data accordingly.

4.2.1. What is the Soil Behaviour Type Index, Ic?

The Ic value is a measure used in geotechnical engineering to characterize and classify soil materials based on their geotechnical properties. It was developed by Robertson (1990) and is derived from geotechnical parameters obtained from CPT data, specifically the cone tip resistance (q_c) and the friction ratio (f_r). It can be defined as:

$$I_c = ((3.47 - \log Q_t)^2 + (\log F_r + 1.22)^2)^{0.5} \tag{4.1}$$

where Q_t is the normalized cone penetration resistance taking into account the in-situ vertical stresses:

$$Q_t = \frac{q_t - \sigma_{vo}}{\sigma'_{vo}} \tag{4.2}$$

q_t is the corrected cone resistance by the cone tip resistance (q_c), the water pressure (u_2) and the net area ratio (a), as:

$$q_t = q_c + u_2(1 - a) \tag{4.3}$$

and F_r is the normalized friction ratio, calculated from the measured sleeve resistance (f_s), the corrected cone resistance (q_t) and the vertical stresses, as:

$$F_r = \frac{f_s}{q_t - \sigma_{vo}} \cdot 100\% \tag{4.4}$$

The Ic value allows for a systematic classification of soil materials into different classes, as presented in Table 4.1. Typically, the cone resistance is high in sands and low in clays, while the friction ratio is low in sands and high in clays (Robertson and Cabal, 2022). Higher Ic values correspond to denser and stiffer soils, while lower Ic values represent softer and looser soils (Robertson, 2010).

Table 4.1: Soil Behaviour Type Index (adapted from Robertson and Cabal, 2022).

Zone	Soil Behaviour Type	I_c
1	Sensitive, fine grained	N/A
2	Organic soils - clay	>3.6
3	Clays - silty clay to clay	2.95 - 3.6
4	Silt mixtures - clayey silt to silty clay	2.60 - 2.95
5	Sand mixtures - silty sand to sandy silt	2.05 - 2.6
6	Sands - clean sand to silty sand	1.31 - 2.05
7	Gravelly sand to dense sand	<1.31
8	Very stiff sand to clayey sand	N/A
9	Very stiff, fine grained	N/A

By assigning appropriate Ic values, different layers within the subsoil profile can be identified and classified. Furthermore, as each material is characterized by a range of Ic values, the geotechnical variability within each layer can also be captured by this parameter. How this heterogeneity is captured in the synthetic data, will be presented in the next sections.

4.2.2. Geotechnical Assumptions in Synthetic Subsoil Schematization

The synthetic subsoil schematization process incorporates several geotechnical assumptions to capture the variability and behaviour of the subsurface. These assumptions ensure a realistic representation of the subsoil by considering the following factors and limitations:

1. Layered System: The subsoil is assumed to consist of multiple layers with distinct properties and compositions. This layered system allows for the representation of different soil types and their corresponding behaviours. Each layer can have unique characteristics, such as different grain sizes, soil structures, or levels of consolidation.
2. Irregular Layers and Indentations: The synthetic models allow for the generation of irregular layers, capturing variations in layer thickness, shape, and orientation. This accounts for the natural irregularities and complexities observed in the subsurface, including indentations or irregular boundaries between layers.
3. Presence of Lenses and Old Channels: The presence of lenses and old channels within the subsurface is possible. These features represent localized variations in soil properties or remnants of past geological processes.
4. Limited Number of Layers: The maximum number of layers possible is five. This constraint ensures a manageable level of complexity while still capturing the essential layering patterns and variations in the subsoil. It strikes a balance between representing the subsurface heterogeneity and maintaining practicality in the analysis.
5. Flat Surface on Top: The topmost layer is assumed to be a flat surface. This assumption provides a reference point and allows for consistent interpretations of the subsoil layers throughout the schematization process.
6. Heterogeneity and Anisotropy: The models account for the inherent heterogeneity and anisotropic behaviour of the subsurface. This includes variations in soil properties and behaviour in both the horizontal and vertical directions. It acknowledges that soil characteristics and responses can vary spatially within a layer as well as with depth.

7. Incomplete Data Penetration: Similar to real-world surveys, the synthetic models acknowledge that data collection may not all reach the full depth. This assumption recognizes the limited depth of investigation typically encountered in geotechnical investigations, where the available data does not provide information about the complete subsurface profile.
8. Non-Adjacent CPTs and Minimal Data Gaps: It is not possible to place multiple CPTs directly adjacent to each other. Additionally, it aims to minimize significant data gaps between adjacent CPT locations. These assumptions contribute to a more realistic portrayal of what a geotechnical survey looks like in reality.
9. Exclusion of Faults: The synthetic subsoil models do not account for the presence of faults. This assumption simplifies the schematization process and focuses on the representation of layering and material variations.

4.3. Generating Synthetic Schematizations

Building upon the geotechnical assumptions outlined in Section 4.2.2, the generation of synthetic data for subsoil schematization considers three key aspects. First, the conditions pertaining to the geometry of the layers themselves, as stated in assumptions 1 to 5. These factors encompass the layered nature of the subsoil, the possibility of irregular layers, indentations, and the occurrence of lenses and old channels. Second, the variability in material properties within each layer, as described in assumption 6, is taken into account to capture the heterogeneity and anisotropy of the subsoil. Third, assumptions 7 and 8 address the generation of synthetic CPT-like data, considering the limited depth reach of the data, and the avoidance of placing multiple CPTs directly adjacent to each other.

4.3.1. Simulating Subsoil Layering Variability

A Python tool was created to randomly generate subsoil geometries comprising between 1 to 5 layers. Each layer boundary is modelled as a periodic function, which could either be a sine or cosine, with randomly determined parameters. These parameters include amplitude (A), period (T), vertical shift (D), and phase shift (Φ). The aim was to produce lines that vary widely in their characteristics, ranging from largely horizontal to extremely sinuous and all the variations in between. These generated lines are demonstrated in Figure 4.1.

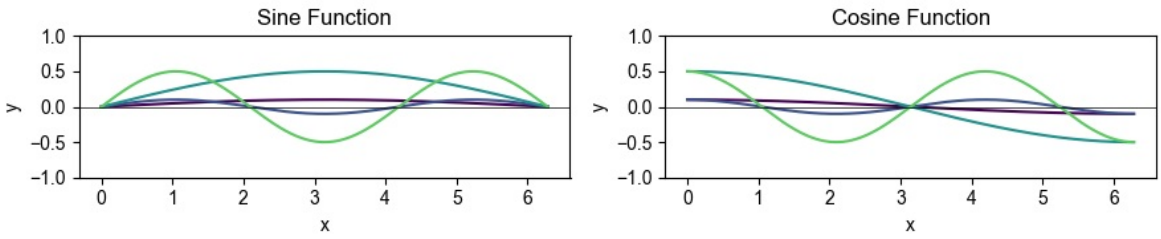


Figure 4.1: Example of periodic trigonometric functions logic as layer boundaries in the generation of synthetic models.

The degree of horizontality or sinuosity in a layer was largely determined by the amplitude (A) and period (T) parameters, which are randomly chosen for each boundary. Specifically, the amplitude (A) sets the maximum and minimum values of the peak and valleys in the layer, while the period (T) impacts the spacing between successive peaks. A Pert distribution was employed for this purpose, given its ability to dictate the most probable value within a range and facilitate smooth transitions to extreme values. This is illustrated in Figure 4.2(a) and Figure 4.2(b). The values utilized are documented in Table 4.2. It's noteworthy that the most probable values are 5 for amplitude (A) and 1000 for period (T), favouring the generation of models with largely horizontal layering while still accommodating models where the boundaries are significantly inclined and variable.

Table 4.2: Sine and cosine function parameter distribution for the generation of random boundaries.

Parameter	Distribution type	Low	Peak	High
Amplitude	Pert	2	5	32
Period	Pert	512	1000	10000
Phase shift	Uniform	0	-	512
Vertical shift	Uniform	0	-	32

The phase shift (Φ) and the vertical shift (D) are also randomly determined, thus introducing horizontal and vertical adjustments to the boundaries. These values are derived from a uniform distribution, thereby ensuring all values within the range have an equal likelihood of being chosen. This distribution is illustrated in Figure 4.2(c) and Figure 4.2(d).

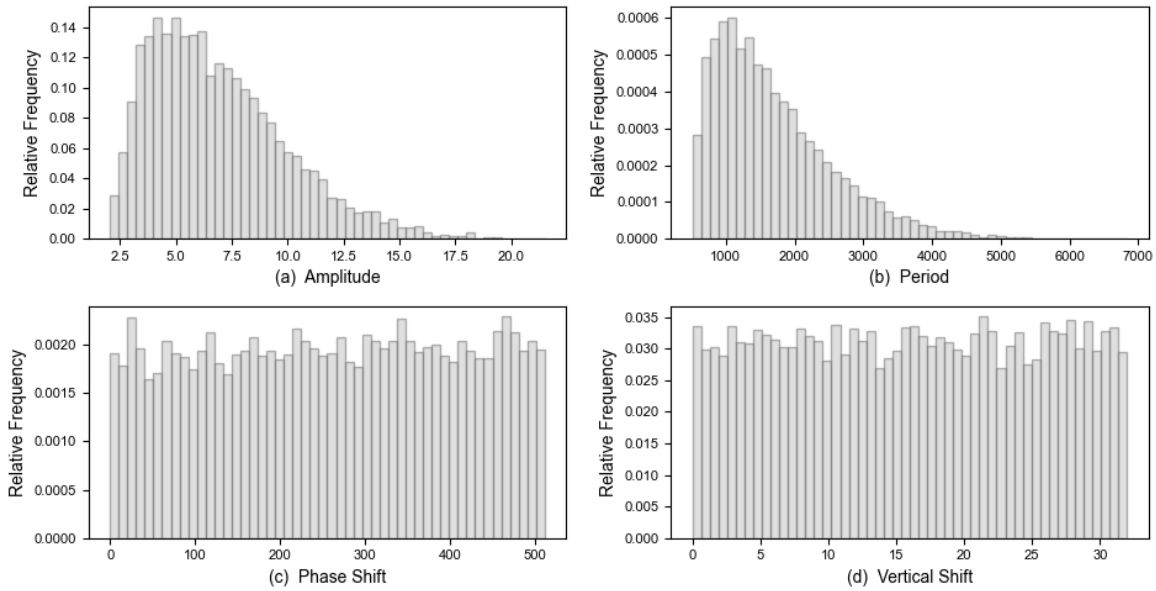


Figure 4.2: Histograms for the selection of the sine and cosine parameters in the random layer generation. Pert distribution in the (a) amplitude and (b) period of the sine and cosine functions, and uniform distributions for the (c) phase shift and (d) vertical shift.

After randomly selecting the wave parameters from the respective distributions, the tool arbitrarily picks either a sine or cosine function. This function was then used to compute the geometry of the boundary, as outlined by the following equations:

$$y(x) = A \cdot sin(2\pi(x - \Phi)/T) + D \tag{4.5}$$

$$y(x) = A \cdot cos(2\pi(x - \Phi)/T) + D \tag{4.6}$$

The aforementioned process was repeated four times for each synthetic generation, in order to create a model with a maximum of five layers. However, the inherent interactions between these periodic wave functions can lead to models with fewer layers. Additionally, such interactions may induce complexity in the geometry of the layers, resulting in phenomena such as indentation, lens formation, and the manifestation of old channels.

After all the boundaries have been delineated, the area corresponding to each layer can be determined. In each synthetic model, these unique areas are assigned to different materials, each possessing its own inherent heterogeneity. Figure 4.3 presents several examples of the 32x512 px synthetic data models, illustrating a spectrum of possible geometries, with an increase in complexity from simple sub-horizontal layering to complex models with indenting layers, lenses and old channels.

4.3.2. Simulating the Subsoil Heterogeneity

Once the area for each layer has been mapped, the material for each layer can be randomly assigned from a list of five soil types, as detailed in Table 4.3, along with their characteristic Ic values. This random allocation process permits repeated occurrences of the same material within a single model, including adjacent regions.

The generation of synthetic data also accounts for the intrinsic anisotropy of the subsoil within individual layers. To achieve this, 2D Gaussian random fields (RF) from the GSTools framework (Müller et al., 2022) were used. The scales of fluctuation were adopted based on Phoon et al. (1995) for sand and clay materials. A triangular distribution was employed to represent horizontal fluctuations, centering around the mean value, while vertical fluctuations were constructed using a uniform distribution. The angle factor was also described with a triangular

Figure 4.3: Synthetic models simulating subsoil layering variability with simple sub-horizontal layering, steep angles, irregular layers, indentations and complex geometries with lenses and old channels.

Table 4.3: Soil type Ic values for the generation of the random synthetic models.

Material	min Ic	max Ic	mean Ic	std dev Ic
Sand	1.31	2.05	1.68	0.12
Sand mixtures	2.05	2.60	2.33	0.09
Silt mixtures	2.60	2,95	2.78	0.06
Clays	2.95	3.60	3.28	0.11
Organic soils	3.60	4.20	3.90	0.10

distribution, prioritizing sub-horizontal structures. From this value, the field angle was calculated as $\pi/factor$. The normalized distributions used for the random assignment of these parameters are demonstrated in Figure 4.4.

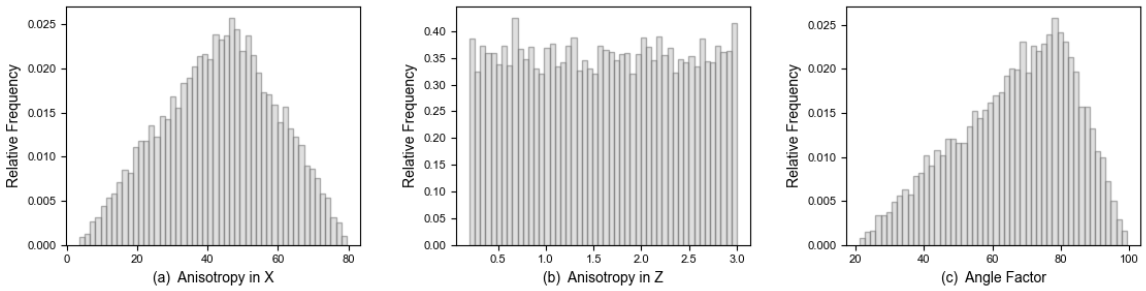


Figure 4.4: Histograms for the selection of the (a) horizontal and (b) vertical scales of fluctuation and (c) the field rotation angle factor, for the generation of the soil type random fields.

Figure 4.5 provides four examples of synthetic models, each featuring the generated RFs corresponding to different soil types in each layer. It is important to note that the dominant structures are sub-horizontal, which is a consequence of the chosen distributions. Furthermore, every RF is the product of randomly selected input parameters. This ensures their uniqueness and allows for the possibility that two similar materials can be adjacent while exhibiting distinct internal characterization.

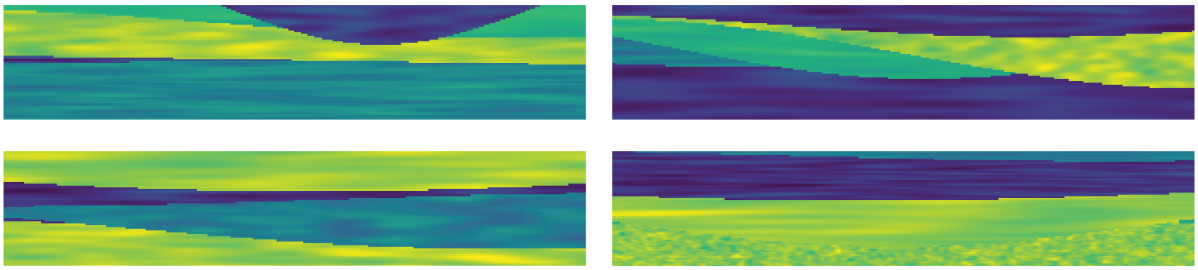


Figure 4.5: Synthetic models simulating the anisotropy within each material layer.

4.3.3. Simulating the CPT Geotechnical Campaign

Additionally to the synthetic cross-section, the CPT data that acts as input for the cGAN needed to be generated and paired with the source schematization. The requirements for the CPT-like data were outlined in Sections 4.2.2 and 4.3.

A complementary tool was generated to receive the complete synthetic cross-section as input and randomly eliminate a user-defined percentage of data, expressed as columns. A constraint was implemented to maintain an appropriate spacing between consecutive CPTs.

To simulate the variability in depths reached during actual geotechnical surveys, the tool removed a random number of rows from the bottom of the remaining columns according to the triangular distribution presented in Figure 4.6. The whole process to transform from the complete synthetic model to the CPT-like image is visually represented in Figure 4.7.

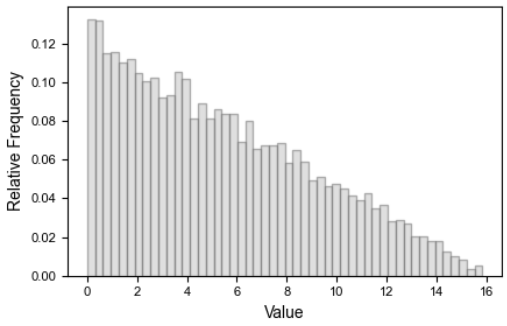


Figure 4.6: Histogram for the selection of the number of rows to remove from the bottom of the CPT-like data, with a peak at 0 and a maximum value of 16 which is half the total depth.

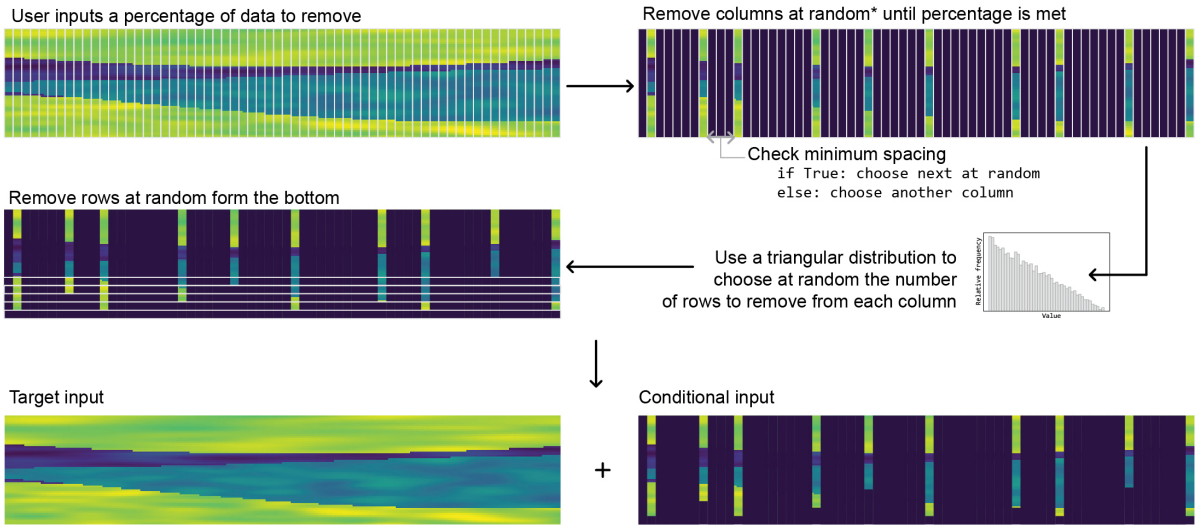


Figure 4.7: Transformation from a synthetic schematization to a CPT-like image as conditional input.

In the majority of the analyses conducted for this project, a missing rate of over 99% has been adopted for generating CPT-like images, closely emulating the realities of real-world geotechnical campaigns where typically less than 1% of the cross-section is surveyed. Unless explicitly stated otherwise, it should be assumed that less than 1% of data is provided as conditional input for the training and validation of the SchemaGAN models.

4.4. The databases

The process detailed previously resulted in the generation of three separate databases, comprising 6,000, 12,000, and 24,000 images respectively. Each was divided into distinct training, validation and testing subsets. Smaller databases were instrumental for rapid architectural iterations, while the largest one was used specifically for the final model training and performance evaluation against traditional interpolation methods and the real-case study.

Table 4.4: Summary of the available databases and their generation seed.

Database name	Samples	Training	Validation	Testing	Seed
4k	6,000	4,000	1,000	1,000	20226086
8k	12,000	8,000	2,000	2,000	20229611
16k	24,000	16,000	4,000	4,000	20223892

4.5. Conclusions on the Synthetic Data Generation

The decision to utilize synthetic data for the training of the cGAN model emerged as the optimal approach following careful consideration. This choice allowed for a series of geotechnical assumptions to be established, which served two key purposes. It outlined a clear framework for the models, while simultaneously ensuring adherence to the realistic nature of geotechnical surveys.

The detailed process to generate 32x512 px synthetic subsoil schematizations based on the Soil Behaviour Type Index (Ic) was meticulously delineated. The methodology encompassed the rationale and decision-making process for system layering, including all geometry alternatives. It also involved the integration of anisotropy within the materials and the transformation of the synthetic schematizations into a CPT-like image, functioning as a conditional input for the training.

Attention was specifically given to the random nature of the generation process. Efforts were made to favour sub-horizontal layer systems and anisotropy structures, as well as CPT-like data that extends almost entirely to the bottom of the models. Simultaneously, the databases incorporated, to a lesser degree, complex models with old channels, steep layering, and lenses, featuring early stopping CPT-like images.

5

SchemaGAN: Conditional Generative Adversarial Network for Subsoil Schematization

5.1. Introduction

This chapter focuses on the construction of a cGAN designed specifically for subsoil schematizations using CPT-like inputs, called SchemaGAN. The architecture for the Generator and Discriminator models, the training configuration, losses and metrics, as well as the validation and testing of the SchemaGAN model are presented and discussed. For more advanced and in-depth information on the background framework, readers are encouraged to refer to the works of Isola et al. (2017), Brownlee (2019), and Bhattiprolu (2021b).

Most of the heavy training of the algorithm was carried out on Delft University of Technology supercomputer, DelftBlue (DHPC, 2022), on a node with an NVIDIA Tesla V100S 32GB of graphic capacity for the fast processing of the images.

5.2. The SchemaGAN Architecture

The SchemaGAN architecture was adapted from the Pix2Pix framework (Isola et al., 2017) in order to perform image-to-image translation tasks specifically for creating synthetic subsoil schematizations. This architecture consists of two main components: the Generator and the Discriminator, forming the complete cGAN.

5.2.1. The Generator

The Generator in the SchemaGAN was built around a modified U-Net architecture which uses encoder and decoder blocks with skip connections in a symmetric structure, allowing to convey both local and global structural information and therefore generation of superior quality images (Ronneberger et al., 2015; Isola et al., 2017). Two encoder blocks and two decoder blocks were used to build the Generator which is illustrated in Figure 5.1.

The regular encoder block (RE) carried out a sequence of operations that shrink the spatial dimensions of the input image and increased its complexity. A convolutional layer (Conv2D) with a 4x4 kernel size and a stride of 2x2 was applied. The irregular encoder block (IE) operated similarly to the regular one but used a stride of 1x2 instead of 2x2 for the convolutional layer in order to account for the 16:1 aspect ratio of the 32x512 px size of the schematization images. Each encoder block also initiates the weight of the layer at random using a normal distribution with a standard deviation of 0.02. Finally, both encoders used the leaky ReLU activation function with an alpha of 0.2 and apply batch normalization.

The regular decoder block (RD) in the SchemaGAN Generator reversed the process carried out by the encoders, increasing the spatial dimensions of the data and reducing its complexity. A transposed convolutional layer (T-Conv2D) with a 4x4 kernel size and a stride of 2x2 was employed. The decoder block also featured

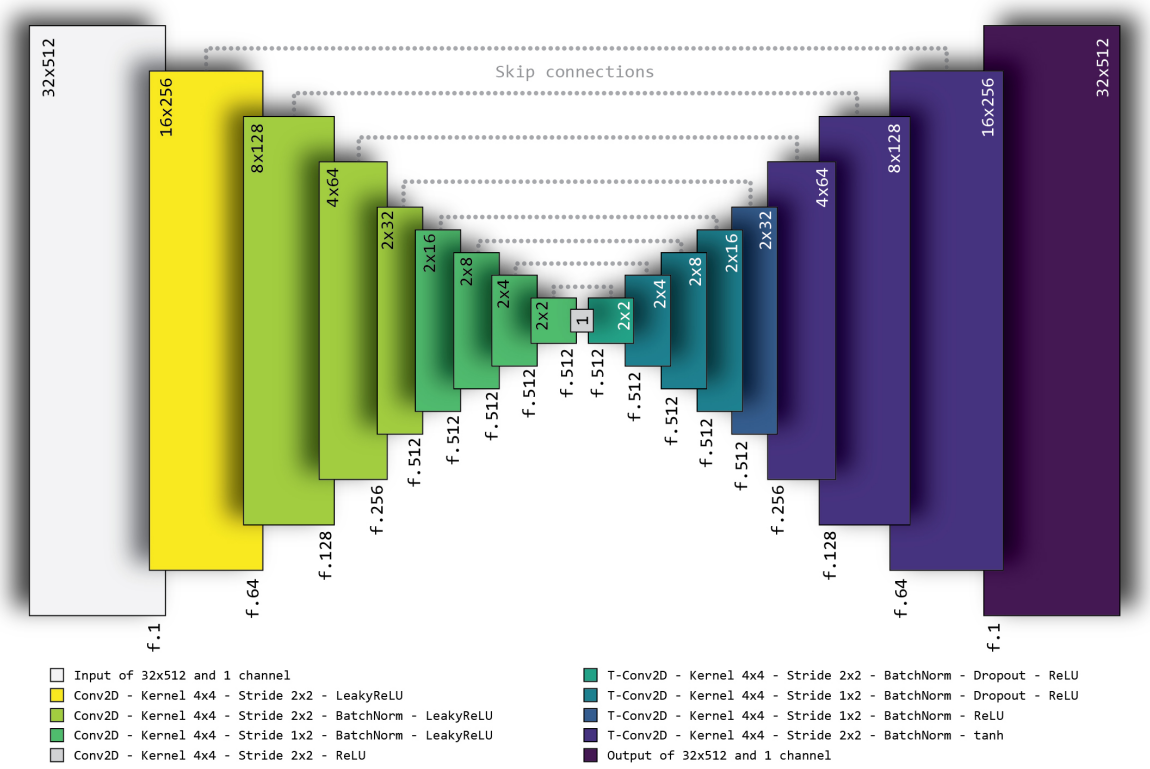


Figure 5.1: SchemaGAN Generator architecture based on a U-net framework.

batch normalization, and potentially, a dropout operation with a rate of 0.5. It also concatenates the result with a skip-connected layer from the encoder part and applied a ReLU activation function.

In order to keep symmetry with the encoders, the irregular decoder block (ID) functioned the same as the regular decoder but used a stride of 1x2 instead of 2x2 for the transposed convolutional layer, providing a different upsampling rate to account for the irregular size of the images.

Using these four encoder-decoder blocks, the modified U-Net architecture for the Generator in the SchemaGAN was built. In summary, it consists of an encoder, a bottleneck, and a decoder.

The encoder progressively downsampled the input through eight sequential blocks. It begins with an image input layer for images of shape (32, 512, 1). The initial four blocks utilize the RE block employing an increasing number of filters (64, 128, 256, 512 respectively) along with standard batch normalization. The subsequent four blocks, from the fifth to the eighth, employ the IE block for a different rate of downsampling, and 512 filters each along with standard batch normalization.

After the last encoder block, a Conv2D layer functions as the bottleneck. With 512 filters, a kernel size of 4x4, and a stride of 2x2, it compressed the input into a lower-dimensional representation. This layer employed a ReLU activation function.

Following, the decoder process is built symmetrically to the encoder. It commenced with an RD block, followed by four ID blocks, and concluded with three additional RD blocks. The number of filters in these layers decreased symmetrically to the encoder (512, 256, 128, 64). Dropout regularization was applied in the first four decoder blocks. Every decoder block was fed with a skip connection from the corresponding encoder block's output, facilitating better localization and detail reconstruction in the output.

The final layer was a T-Conv2D layer using a 4x4 kernel and a stride of 2x2, which upscaled the feature maps back to the original image size of 32x512 px. It utilized a $tanh$ activation function to output values to a range between -1 and 1.

5.2.2. The Discriminator

The Discriminator for the SchemaGAN model is designed based on the PatchGAN Discriminator as a binary classifier and operates on a larger input shape of 32x512 px in order to accommodate the cross-sections generated for the training. The architecture is presented in Figure 5.2.

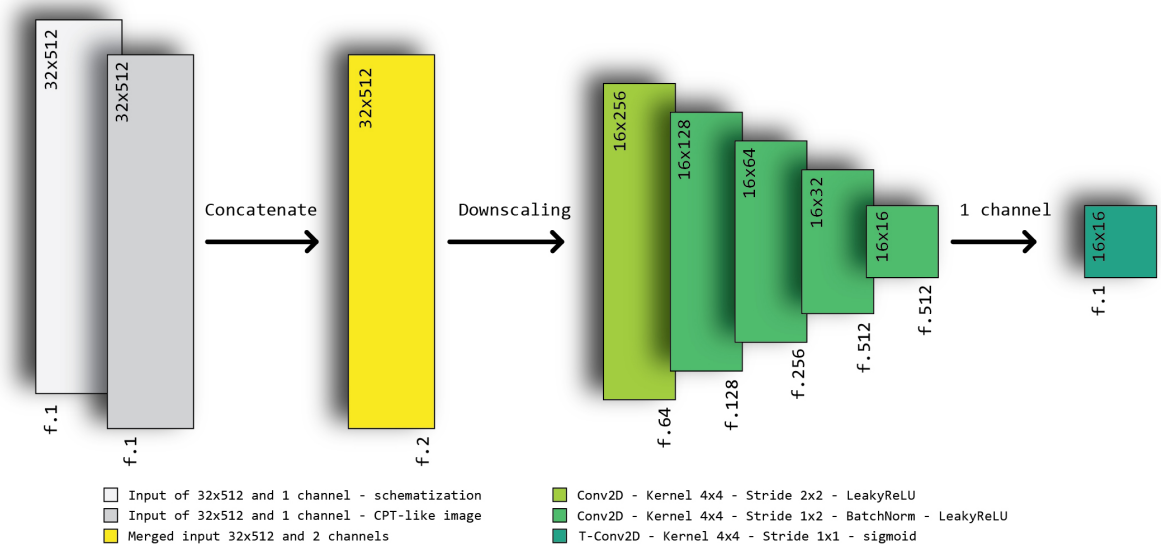


Figure 5.2: SchemaGAN Discriminator architecture based on a PatchGAN framework as a binary classifier.

The model takes two inputs: the CPT-like image and the schematization image (real or generated), both of size 32x512 px and one channel. These two images are then concatenated channel-wise to form a merged input. The first step in the processing sequence was a Conv2D layer with 64 filters, a 4x4 kernel size, and strides of 2x2, which downscales the merged input to 16x256 px. The output from this layer was then passed through a LeakyReLU activation function with an alpha of 0.2.

Following this, the model sequences through four additional Conv2D layers, each with a 4x4 kernel size but an increasing number of filters: 128, 256, 512, and 512. These layers applied a stride of 1x2, progressively reducing the width dimension of the output at each step. All these Conv2D layers, except for the first one, are accompanied by Batch Normalization, followed by another LeakyReLU activation function, promoting a stabilized training process and improved model generalization.

The model then feeds the output through an additional Conv2D layer with 512 filters, maintaining the spatial dimensions at 16x16 with a stride of 1x1. This layer also includes batch normalization and LeakyReLU activation.

The final step in the Discriminator architecture was another Conv2D layer with a single filter and a 4x4 kernel size. This layer produced the patch-based output, scaling the output values to a range between 0 and 1 with a sigmoid activation function. The output represents the probability of each 16x16 patch in the image being real or fake.

Once the Discriminator model was defined with the input layers and the patch-based output, it was compiled using the Adam optimizer with a learning rate of 0.0002 and a beta value of 0.5. The model employed the BCE loss function for training, with each model update being weighted by 50% in the loss. This will be expanded on in the following sections.

5.2.3. The Conditional Generative Adversarial Network

The combined SchemaGAN model integrates the previously described Generator and Discriminator models. In the combined model, the Discriminator weights are frozen to ensure that only the Generator weights are updated during the training. However, an exception was made for Batch Normalization layers as these require updates during training.

For the compilation of the SchemaGAN, the Adam optimizer was used with a learning rate of 0.0002 and a beta value of 0.5. The model utilized two loss functions: BCE for evaluating the output of both the Generator and the Discriminator and a MAE loss that was added only for evaluating the output of the Generator.

The complete SchemaGAN model, encompassing both the Generator and Discriminator, boasts a total of 78,172,226 parameters. Of these, 67,003,137 are trainable parameters, accounting for the capacity of the network to learn and adapt during the training process. The remaining 11,169,089 are non-trainable parameters, contributing to the structure and functionality of the network without being influenced by training.

5.3. The Training Algorithm

The SchemaGAN operates as a type of cGAN, producing images under a given condition. In the context of this study, the CPT-like image was used for conditioning by both the Generator (G) and Discriminator (D). The Generator takes a noise vector and CPT-like data to construct synthetic subsoil schematizations. Meanwhile, the Discriminator differentiates between the real pairs from the training dataset and the synthetic pairs crafted by the Generator. The subsequent sections offer a detailed exploration of this process, with Figure 5.3 visually representing the training framework.

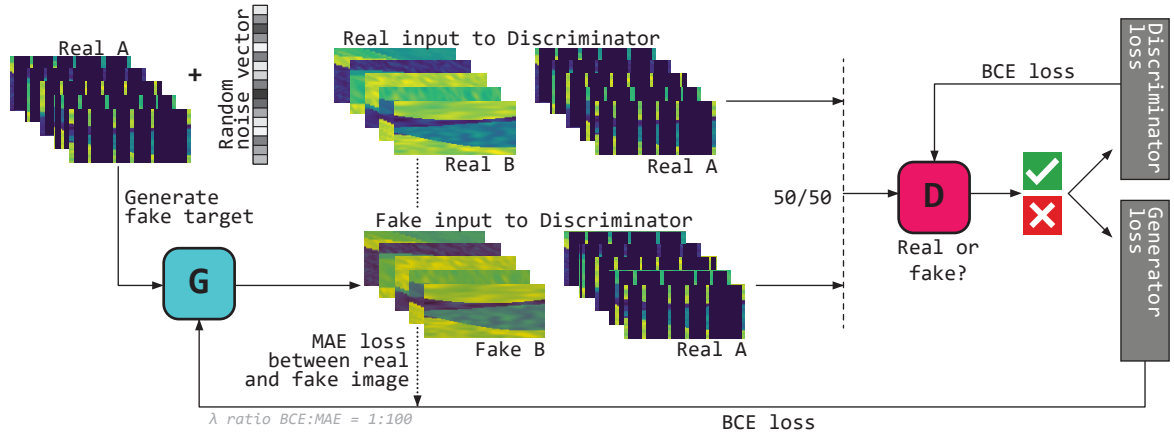


Figure 5.3: Schematization of the adversarial behaviour in the SchemaGAN training for the Generator and Discriminator.

During training, the Generator and Discriminator were updated sequentially. The Discriminator was first updated. Its loss was calculated based on its accuracy in correctly classifying both real and fake data pairs and was minimized by adjusting the Discriminator weights. This process was done independently, without reference to the Generator, relying solely on the actual data from the training dataset and the current synthetic outputs of the Generator.

Following, the Generator was updated with the objective of improving the quality of its subsoil schematizations to such an extent that the Discriminator was fooled into classifying these synthetic outputs as real. The Generator loss was influenced not just by the feedback from the Discriminator but also incorporates an MAE loss term. This term quantifies the resemblance between the generated schematization and the real one, ensuring that the Generator outputs were not only convincing to the Discriminator but also correspond closely to real schematization. The Generator weights were updated using this composite loss.

This iterative, alternating training regimen improves the Generators ability to produce increasingly realistic schematizations and motivates the Discriminator to continually refine its discerning capability. Training continued until the Discriminator was unable to consistently distinguish between real and fake pairs, meaning that the Generator had achieved its goal.

5.3.1. Objective of the SchemaGAN model

Mathematically, an overview of the cGAN training is given in the Literature Review Section 2.6.1. The final objective of the training of the SchemaGAN, based on the works from Isola et al. (2017), is to find the parameters of the Generator network that minimize the objective function, as:

$$G^* = arg \min_G \max_D \mathcal{L}_{cGAN}(D,G) + \lambda \mathcal{L}_{L1}(G) \tag{5.1}$$

where G^* is the optimal Generator network, capable of taking in CPT-like data and generating subsoil schematizations that are as close to the real training one and that performs better than the traditional interpolation methods.

$\mathcal{L}_{cGAN}(D,G)$ is the cGAN loss, which the Generator tries to minimize and the Discriminator tries to maximize along the minimax game, pushing the Generator towards its objective. $\mathcal{L}_{L1}(G)$ is the mean absolute error, added to help the Generator avoid mode collapse and produce realistic images at a pixel-by-pixel level. And λ is a weight factor to control the effect of the MAE loss.

5.3.2. The Adversarial Loss

The GAN loss $\mathcal{L}_{cGAN}(D,G)$ sits at the core of the SchemaGAN and is formulated as a minimax problem. Here, the Discriminator (D) is striving to maximize its capability to correctly classify real and generated images, whereas the Generator (G) is trying to create images that the Discriminator cannot differentiate from real ones. This loss can be expressed in the form of BCE behaviour as:

$$\mathcal{L}_{cGAN}(D,G) = \mathbb{E}(y \log D(y,c) + \mathbb{E}(1-y)\log(1 - D(c, G(z,c))) \tag{5.2}$$

In the above equation, c denotes the conditional CPT-like data, y signifies the real schematization and z is a noise vector. $G(z,c)$ represents the generated schematization, whereas $D(y,c)$ and $D(G(z,c))$ are the Discriminator probability estimates for the real and generated images, respectively.

The GAN loss can be decomposed further into components particular to the Discriminator and the Generator. The Discriminator loss consists of two parts: The first term signifies the expected log-likelihood that the Discriminator correctly classifies a real schematization as real. The second term represents the log-likelihood that a fake (generated) schematization is correctly identified as fake. The Discriminator aims to maximize both terms.

$$\mathcal{L}_{cGAN}(D) = -\mathbb{E}(\log D(y,c)) - \mathbb{E}(\log(1 - D(c, G(z,c))) \tag{5.3}$$

Conversely, the Generator loss is a composite of the Generator loss from the standard GAN and an additional loss term:

$$\mathcal{L}_{cGAN}(G) = \mathbb{L}_c(G) + \lambda \mathbb{L}_{L1}(G) \tag{5.4}$$

The GAN loss for the Generator can be viewed as its attempt to minimize the likelihood that the Discriminator correctly identifies the created schematizations as fake. Essentially, this represents the Generator's goal to fool the Discriminator:

$$\mathcal{L}_c(G) = \mathbb{E}(\log(1 - D(c, G(z,c)))) \tag{5.5}$$

The additional term, $\lambda\mathcal{L}_{L1}(G)$, denotes the MAE between the Generator's output and the actual subsoil schematization:

$$\mathcal{L}_{L1}(G) = \mathbb{E}(|y - (G(z,c))|) \tag{5.6}$$

This term assists in mitigating mode collapse, a prevalent issue in cGANs, and encourages the Generator to create images that not only fool the Discriminator but also closely resemble the ground truth at a pixel level. The λ parameter in the equation is a balancing factor between the cGAN loss and the MAE loss. Manipulating its value facilitates a trade-off between image diversity (encouraged by a smaller λ) and image accuracy (encouraged by a larger λ).

5.3.3. The Training Parameters

The performance and successful training of the SchemaGAN model largely depended on the calibration of various training parameters. The optimization and fine-tuning were conducted via a two-pronged approach: research into established methods and models, and iterative experimentation on the SchemaGAN model. The initial phase involved an in-depth review of existing literature and successful GAN models to gain insights into parameter settings that have proven effective in similar tasks.

Following, the parameters were iteratively adjusted on the SchemaGAN model itself. The smaller databases comprising 4k and 8k samples served as the initial training ground. These allowed for rapid cycles of training and evaluation, providing valuable feedback on the impact of the parameters.

Training epochs ranging from 50 to 5000 were tested. A total of 200 epochs was chosen to strike a balance between model accuracy and computational efficiency. Batch sizes of 1, 2 and 5 were tested, with a batch of 1 and a learning rate of 0.002 chosen for their improved accuracy and following the recommendations of Isola et al. (2017). This has been proven to be very effective in image-generation tasks.

For the Generator loss, individual BCE and MAE losses were tested, as well as a combined loss with λ weights of 10, 50 and 100. A λ value of 100 gave the best results in line with the recommendations of Isola et al. (2017), carefully controlling the trade-off between the BCE loss and the MAE loss.

5.4. Evaluation Metrics on the Model Training

The training process reveals three distinct phases in the accuracy metrics (Figure 5.4). The first phase, ranging from the start to around 30 epochs, displays high variability in accuracy, an expected behaviour as the Discriminator learns to distinguish real from generated schematizations. This phase sees the average accuracy fluctuating around 0.8, which is in line with the expected for GAN training (Bhattiprolu, 2021a).

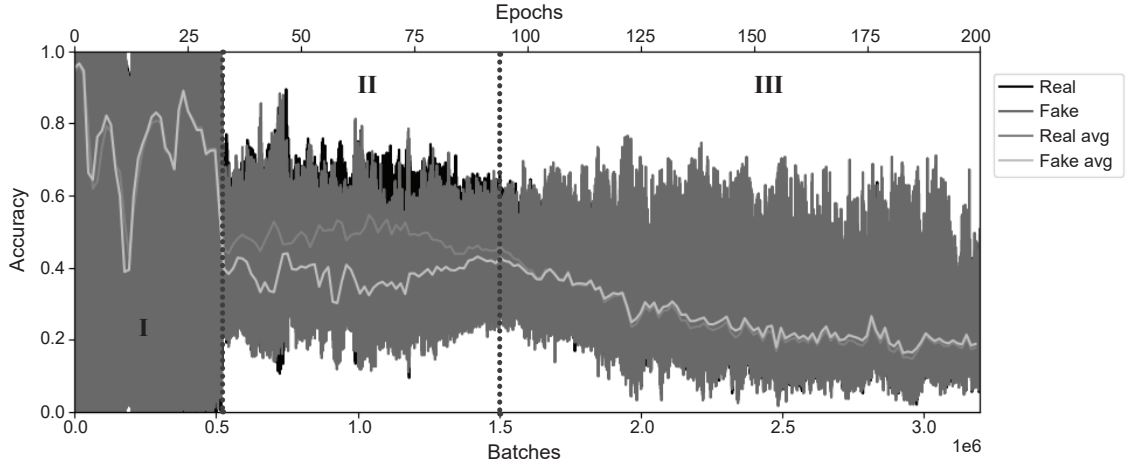


Figure 5.4: Training accuracy in the Discriminator in the classification of real and fake subsoil schematizations.

The second phase, from approximately 30 to 90 epochs, exhibits a general decrease in accuracy and a divergence between real and generated schematizations. During this period, the Discriminator finds it increasingly challenging to correctly identify the schematizations generated by the SchemaGAN, compared to the real schematizations. An average accuracy of around 0.5 during this phase is not unusual in GAN training and often indicates successful training of the model, where the Discriminator stands a 50/50 chance of correctly classifying an input (Goodfellow, 2017).

In the third phase, beginning from around 90 epochs, mode collapse was identified as a drastic drop in accuracy for both real and generated samples, reaching a value of 0.2.

Loss metrics for both the Generator and the Discriminator, shown in Figure 5.5, followed three similar phases, although slightly less distinct. During the first phase, the Discriminator loss varied between 0 to slightly above 1, while the Generator loss quickly declined to around 5. This trend is indicative of the Generator initial difficulty in generating realistic schematizations, leading to high losses but quick improvements.

After the 30-epoch mark, the Discriminator loss stabilized around 0.5 and hovered around this value for the rest of the training period. The Generator, on the other hand, continued to reduce its loss until the 90-epoch mark, beyond which it appeared to stabilize. This behaviour is in line with the previously mentioned mode collapse, suggesting that the model's performance did not show significant improvement beyond the second phase of training.

5.4.1. Observing the SchemaGAN in the Training Dataset

To visually examine the progression and improvement of the SchemaGAN model throughout the training process, snapshots from epochs 1, 60 and 200 are shown in Figure 5.6. This illustrates an original cross-section and cpt-like input from the training dataset, and the three different results from SchemaGAN within the domains previously defined.

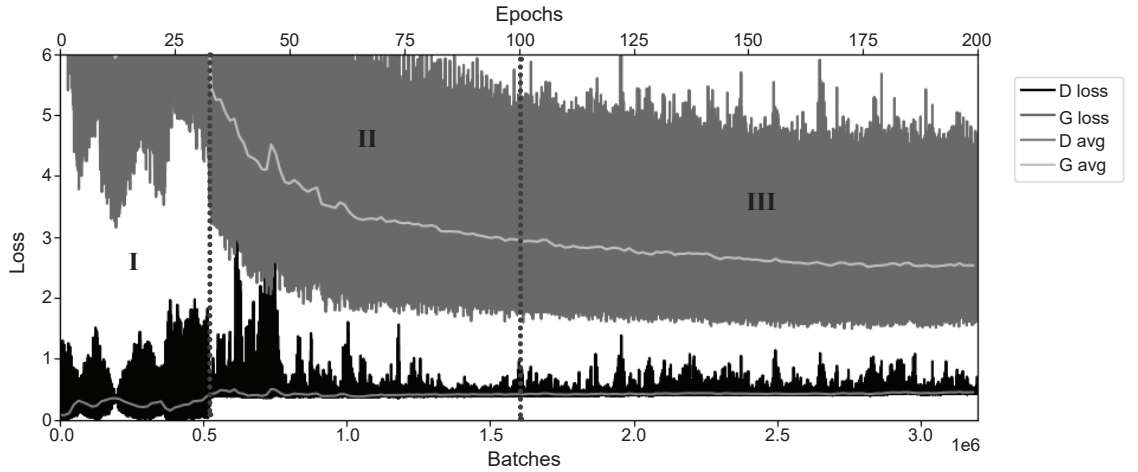


Figure 5.5: Training losses in the Generator and Discriminator over time.

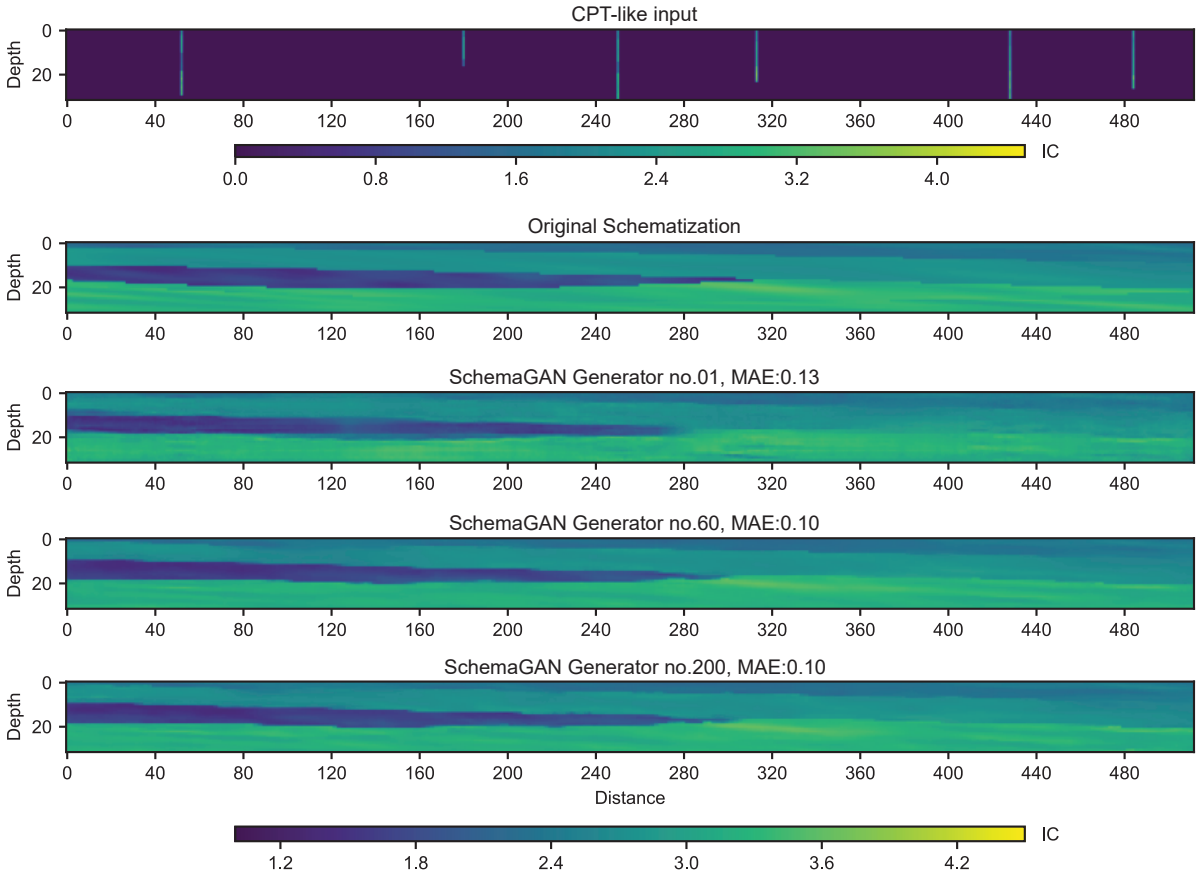


Figure 5.6: Progress of the SchemaGAN training at three distinct epochs: 1, 60 and 200. Including the original schematization, the CPT-like input and the MAE for each generation.

The first SchemaGAN Generator produces results that are blurry and noisy, and although the general structure of the layers is discernible, the geometry lacks definition. This is reflected in a MAE of 0.13. After 60 epochs of training, the results improved significantly. A MAE of 0.10 reflects the strong definition of the layer boundaries and the reduction of noise. Finally, after 200 epochs the visual improvement in the results is identified, which is reflected also in the resulting MAE of 0.10. This falls in line with the training metrics which showed that the model stopped improving during the last domain.

5.5. Validation of the SchemaGAN model

To gain further insight into the performance of SchemaGAN and choose the best-performing Generator model, the validation was carried out with the 4,000 samples validation database. Each of the trained models from the 200 epochs was evaluated through the validation database and their evaluation metrics were computed.

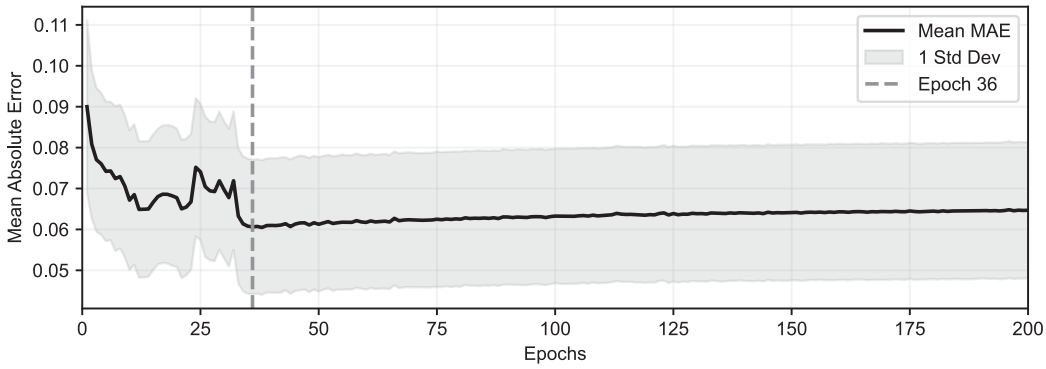


Figure 5.7: Average MAE for each epoch on the validation dataset.

The MAE was computed for each instance in the database and averaged per epoch (Figure 5.7). The validation metrics mirrored the initial phase observed in the training metrics. A rapid decline in the MAE occurred up to the 36th epoch, after which the MAE essentially levelled off, with a slight increase observed as the training carried on until completion. This suggests that the best-performing SchemaGAN model is the 36th epoch of training. This was chosen to carry to the testing phase of the research.

5.5.1. Observing the SchemaGAN in the Validation Dataset

To qualitatively observe the validation of the SchemaGAN Generators, a snapshot of the same schematization was captured at epochs 1, 36 and 200. These epochs represent the three distinct stages from the validation metrics: the initial Generator, the best-performing Generator and the last Generator. Figure 5.8 presents the results and their corresponding MAE.

As anticipated the initial performance was relatively poor. Although the structure of the cross-section is somewhat accurately captured with roughly the correct materials in their proper sequence, the bottom right layer is missing and the layer boundaries are blurry and poorly defined. Moreover, the intra-layer variability in the generated schematization is practically nonexistent. This is reflected in the high MAE of 0.24.

The best results are obtained at epoch 36, with a MAE of 0.15 for this given schematization (Figure 5.8) and visually showing all layer boundaries in a more uniform and defined manner. Even more complex elements, such as the indentation on the bottom right are accurately captured. In addition, the variability within the layers is enhanced and more closely resembles the original.

After 200 epochs of training, no substantial improvement is observed in the results. In fact, following the metrics results, there is a slight depreciation of SchemaGAN ability to create accurate schematizations. This is confirmed by Figure 5.8 in which no visual improvement is immediately apparent and the MAE increases to 0.16.

5.6. Testing the SchemaGAN model

The SchemaGAN best performance was observed at epoch 36; thus all testing was chosen to be done with this model. The final assessment took place using a separate test database of 4,000 new samples. Figure 5.9 shows the MAE results obtained by comparing the original test schematization with the SchemaGAN generated results for all samples. For straightforward horizontal geometries, MAE values are as low as 0.06. As model complexity increases, MAE values gradually rise. The most intricate models, including irregular layering, indentations, lenses, and older channels, have a maximum MAE value of 0.32.

Following, the testing results were explored qualitatively. Figure 5.10 illustrates a common and simple geometry—a sub-horizontal layered system. The SchemaGAN model effectively solved this configuration, faith-

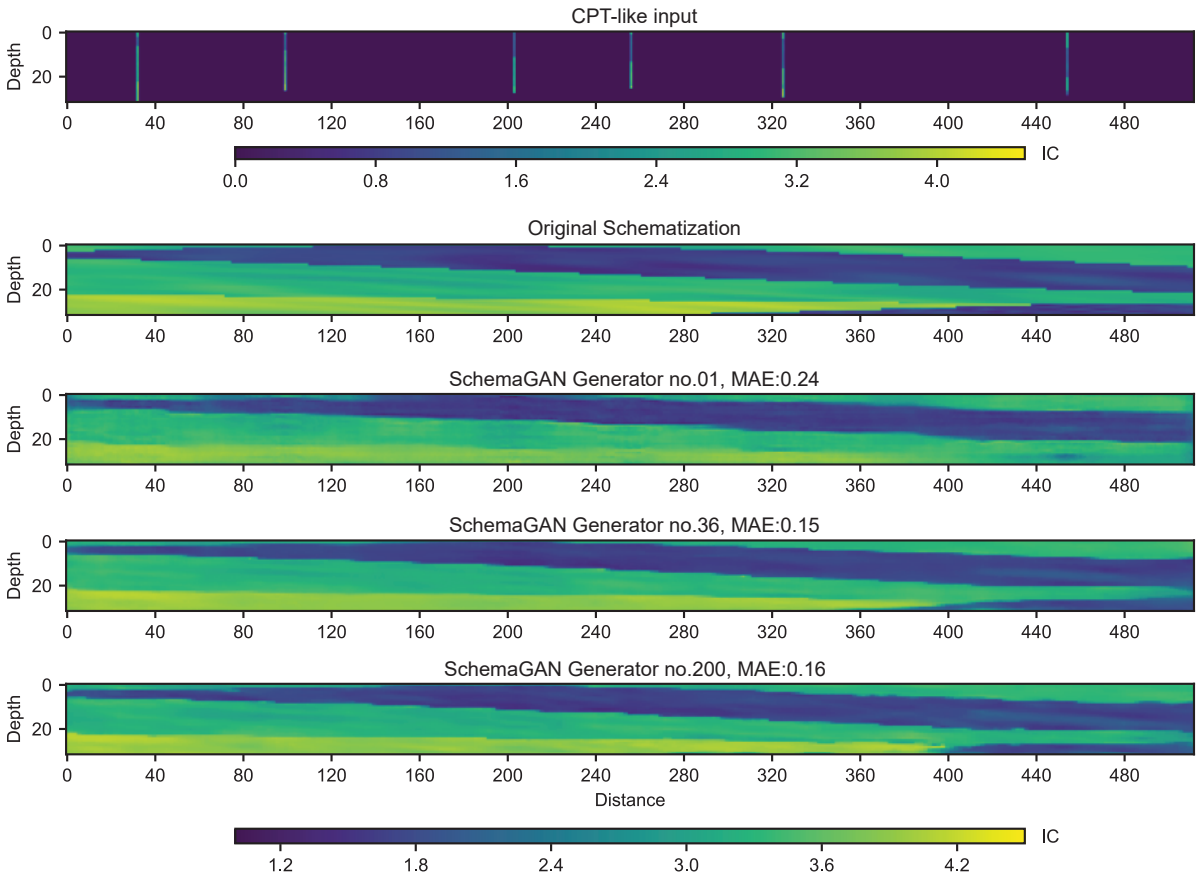


Figure 5.8: Results from the SchemaGAN generation in the validation dataset for epochs 1, 36 and 200, and their corresponding MAE; including the original schematization and the CPT-like input used.

fully reproducing the layers and capturing internal anisotropy accurately. Notably, the most significant discrepancy arises in the lower right corner due to limited depth in the provided CPT-like data. Consequently, the model encounters challenges in precisely defining the lower layer boundaries.

Figure 5.11 presents a moderately complex schematization with two indenting layers. SchemaGAN captures the layer boundaries structure and internal anisotropy with precision. Despite the absence of CPT-like data for the right section where the principal indentation disappears, the SchemaGAN model accurately predicts the layer behaviour, resulting in a MAE of 0.12.

The results presented in Figure 5.12 show an increase in the complexity of the schematizations, with a model that includes indentations and lenses. The SchemaGAN generated interpretation captured the overall structure of the original but struggled to define where the lenses begin and end. This indicates that as the geometry becomes more irregular, SchemaGAN has a harder time following the overall shape of the layer boundaries. Both these conditions are evident in the high MAE values mapping along said areas.

Finally, Figure 5.13 illustrates a highly complex geometry which includes indentations, very thin layers and an old channel. Although quantitatively the MAE value is 0.12 indicating good results, qualitatively the SchemaGAN generation is far from accurately representing the original image. The overall order and distribution of the layers are correct, thus the low MAE, but their definition is lacking. The biggest hurdle in the generated image is the old channel as it has a very sharp and irregular shape which the model does not capture correctly. Other than not capturing the correct shape, the layer boundary is very blurry. The thin original layer and the top right indentation lack definition and continuity.

Examining all the MAE mappings, regardless of the complexity of geometry in the schematization, the errors in the generation are always encountered in the layer boundaries. In simple models the definition is quite accurate, and as the complexity increases, so does the lack of continuity and definition in the boundaries.

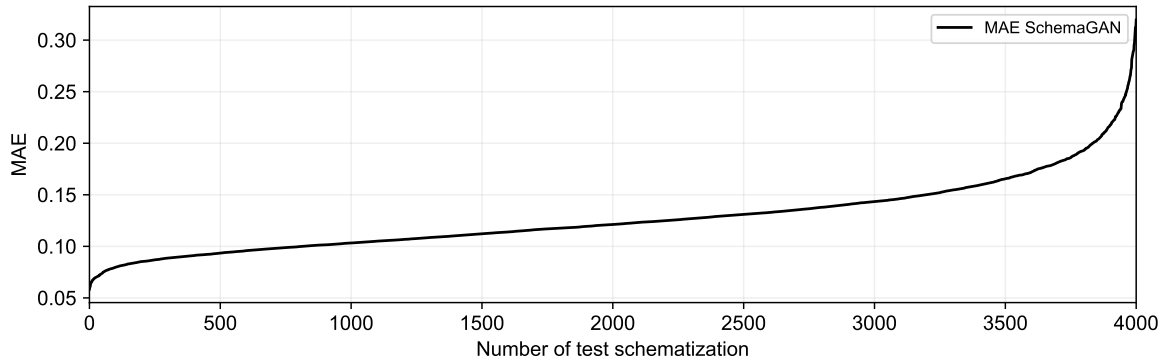


Figure 5.9: Mean absolute error over the entire domain of test samples in the 16k database.

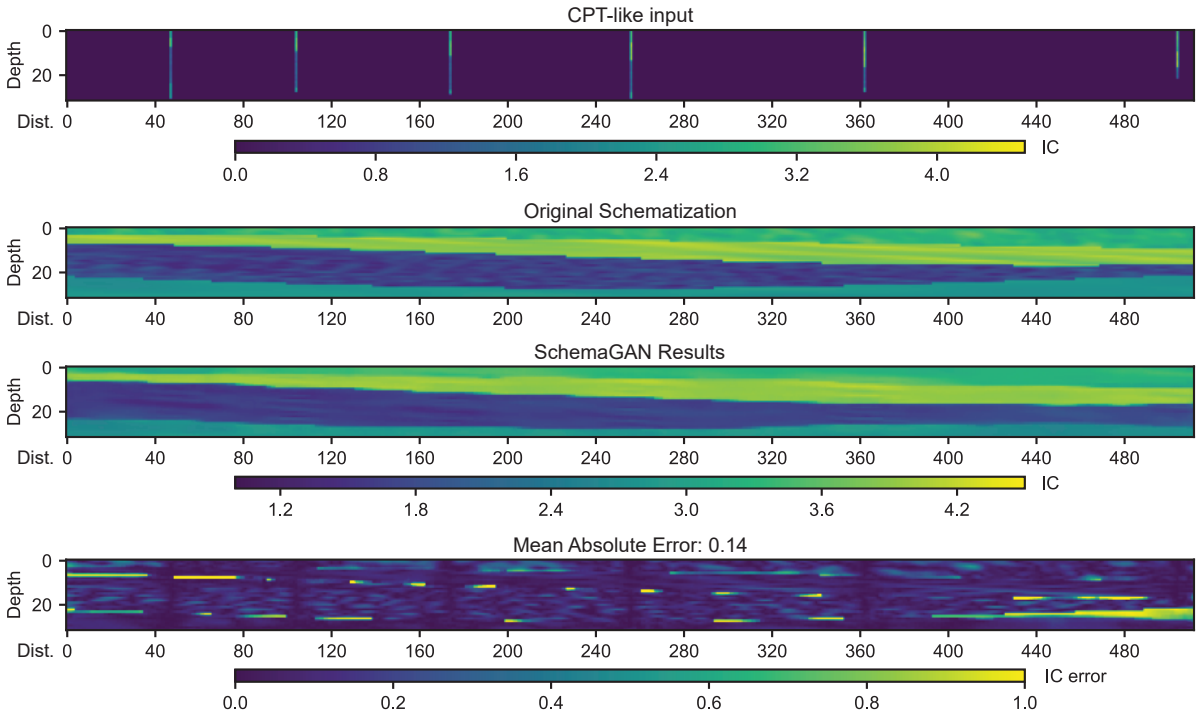


Figure 5.10: Results from a simple sub-horizontal layering schematization generated by the SchemaGAN model from the corresponding CPT-like conditional input.

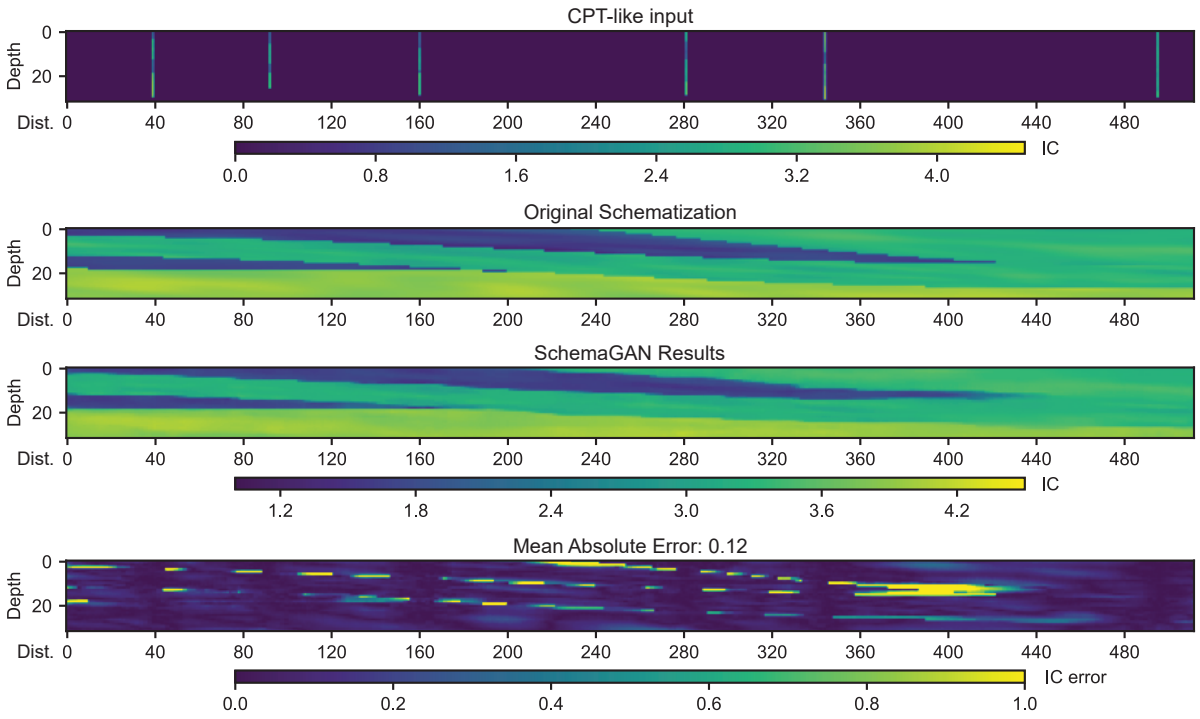


Figure 5.11: Results from a moderately complex schematization with indentations generated by the SchemaGAN model from the corresponding CPT-like conditional input.

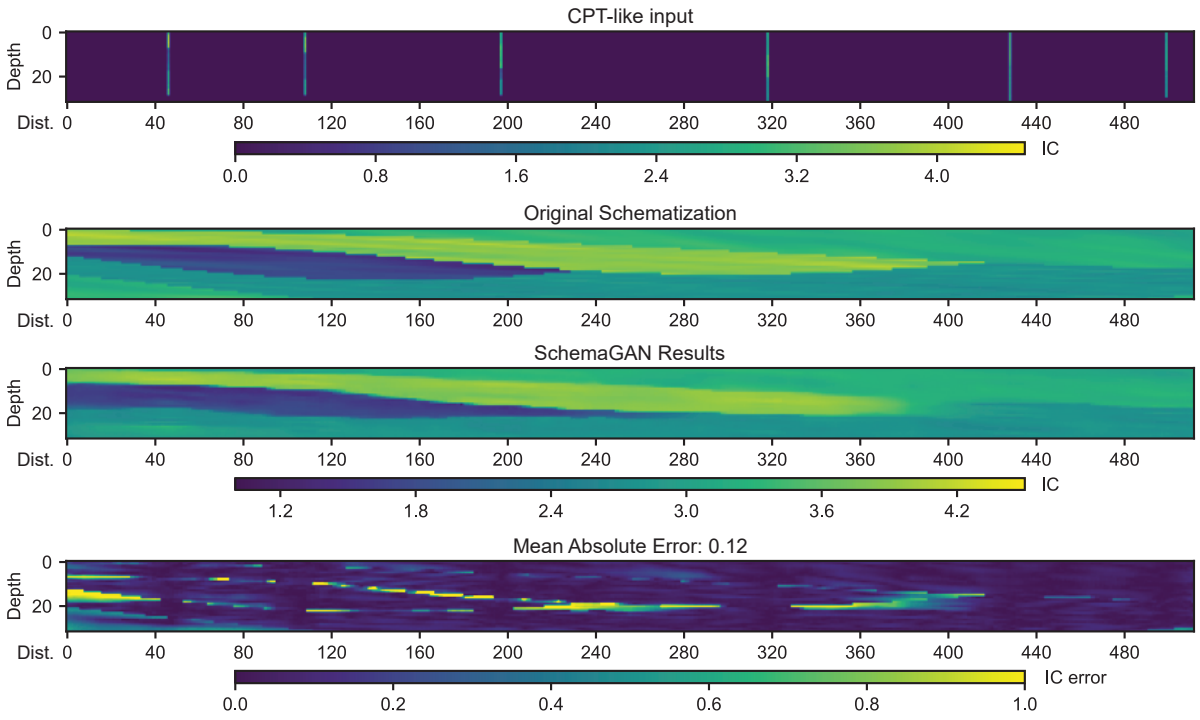


Figure 5.12: Results from a moderately complex schematization with lenses and indentations generated by the SchemaGAN model from the corresponding CPT-like conditional input.

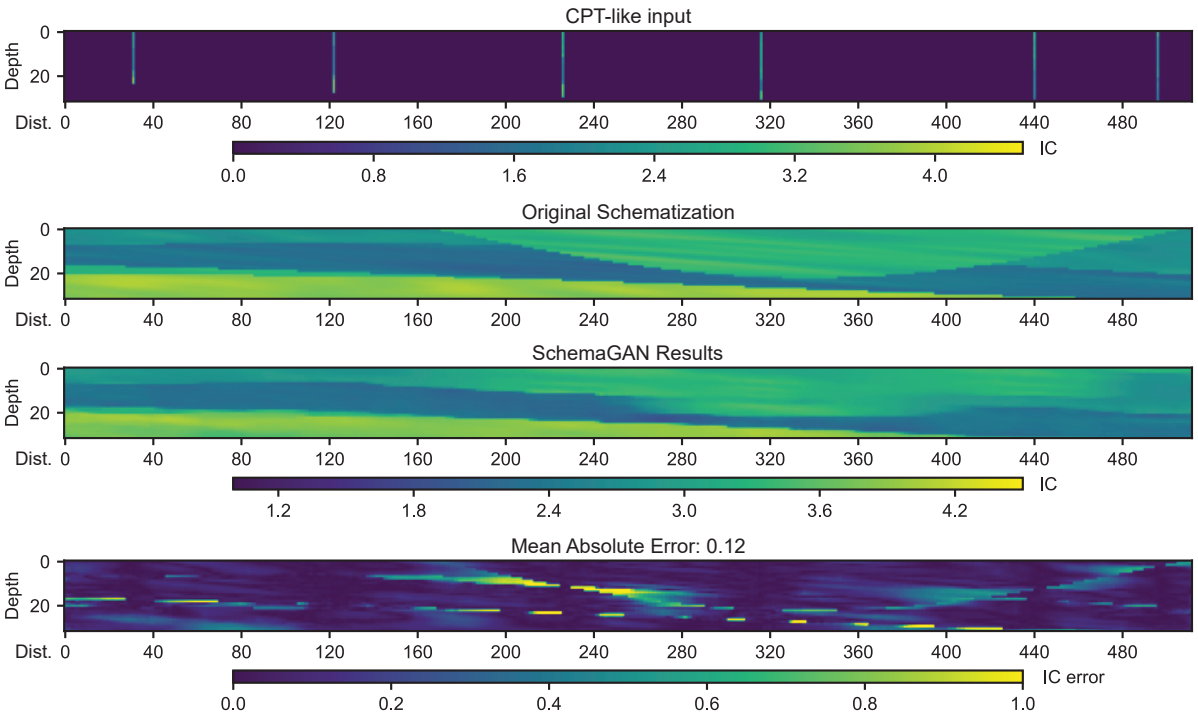


Figure 5.13: Results from a complex schematization with indentations and old channel generated by the SchemaGAN model from the corresponding CPT-like conditional input.

5.7. Key Takeaways from SchemaGAN Implementation

The Pix2Pix framework proved to be an effective foundation on which to build the SchemaGAN model to translate CPT-like data into synthetic subsoil schematizations. The proposed SchemaGANs U-net Generator and PatchGAN Discriminator handle the 32x512 px schematizations correctly, showing a clear improvement over the epochs.

Training of the SchemaGAN model is driven by a combination of BCE loss and MAE loss which work in an adversarial manner. BCE loss steers the Generator and Discriminator competition, while the MAE loss tries to prevent mode collapse, ensuring resemblance between generated and real schematizations. This combination has been has been proven in the past (Isola et al., 2017) and shows good results here in driving the GAN architecture.

Three distinct phases were identified in the accuracy and loss metrics during training. The first phase exhibits high variability and losses as the models start to learn the underlying rules of the schematizations. In the second phase, both the Discriminator and Generator improve rapidly, as the Discriminator struggles to distinguish real from generated schematizations. The third phase indicates failure as mode collapse, stopping the SchemaGAN model performance improvement.

The SchemaGAN model's ability to generate realistic subsoil schematizations significantly improved up to the 36th epoch, as reflected in reduced MAE and enhanced clarity of layer boundaries. Post the 36th epoch, performance plateaued and showed minor degradation, seen in increased MAE and less distinct layer boundaries. Extended training beyond the optimal point may not yield better results, and could even diminish the model's effectiveness.

Despite showcasing impressive performance across various complexity levels, the SchemaGAN model's limitations surface in intricate scenarios. Challenges arise in determining precise layer boundary positioning and in generating features not present in the CPT data. More training with complex geometries is necessary for SchemaGAN to learn to predict this sort of scenario.

6

Comparative Analysis of SchemaGAN and Traditional Interpolation Techniques

6.1. Introduction

This chapter conducts a comparative analysis between SchemaGAN and established interpolation methods, focusing on their effectiveness in generating detailed subsoil schematizations. The primary goal is to evaluate SchemaGAN's ability to capture intricate subsoil features in contrast to conventional techniques like NeaNe, IDW, Kriging, NatNe, and newer approaches such as inpainting.

Traditional methods, though widely accepted in geosciences, often oversimplify the complexities of subsoil formations when applied to schematizations. In contrast, SchemaGAN leverages cGANs to offer a more sophisticated representation of subsoil strata.

Synthetic CPT data from a specialized database of 4,000 samples is employed to generate subsoil schematizations using both SchemaGAN and conventional methods. The assessment encompasses quantitative and qualitative analyses, considering computational efficiency through computation time and enhancing the evaluation with visual comparisons of schematizations. Furthermore, the chapter investigates the influence of CPT data location on resulting schematizations and conducts an expert criteria survey to rank the performance of the methods. Finalizing the comparison, the SchemaGAN model was applied to a case study with real CPT data.

6.2. Quantitative Comparison of Methods

In order to compare the performance of each method with respect to the original schematization, the MAE and MSE were computed for each generation in the testing database. Furthermore, the inference time for each method was also explored.

6.2.1. MAE and MSE Evaluation Metrics

Results from the average evaluation metrics presented in Table 6.1 show that SchemaGAN outperforms all traditional methods, achieving the lowest MAE and MSE value. This indicates that SchemaGAN predictions are, on average, closer to the original schematizations than those generated by the other methods.

Table 6.1: Comparison metrics for different methods from the 4,000 sample test database.

Method	SchemaGAN	NeaNe	IDW	Kriging	NatNe	Inpainting
Mean MAE	0.128	0.168	0.295	0.190	0.249	0.427
Mean MSE	0.074	0.148	0.246	0.096	0.188	0.334

Among the traditional methods, the NeaNe and Kriging methods performed the best, while IDW, NatNe, and Inpainting methods exhibited the worst overall performance, indicating potential limitations in their accuracy. Notably, despite its innovative approach, the Inpainting method was clearly not able to come close to all other methods, underscoring its limitations in scenarios where substantial missing information is to be reconstructed.

To better understand the results from the box plots of Figure 6.1, it's important to recall the varying complexity inherent within the synthetic databases used for this study, as they span a wide spectrum of stratigraphical complexities. At one end, there are simple, sub-horizontal layered systems. Moving along the spectrum, more intricate geometries featuring irregular layering, noticeable indentations, or significantly inclined layers occur. At the extreme end, the models display high complexity, with irregular layering, pronounced indentations, isolated lenses, and even remnants of ancient channels embedded within the stratigraphy at various depths (see Chapter 3).

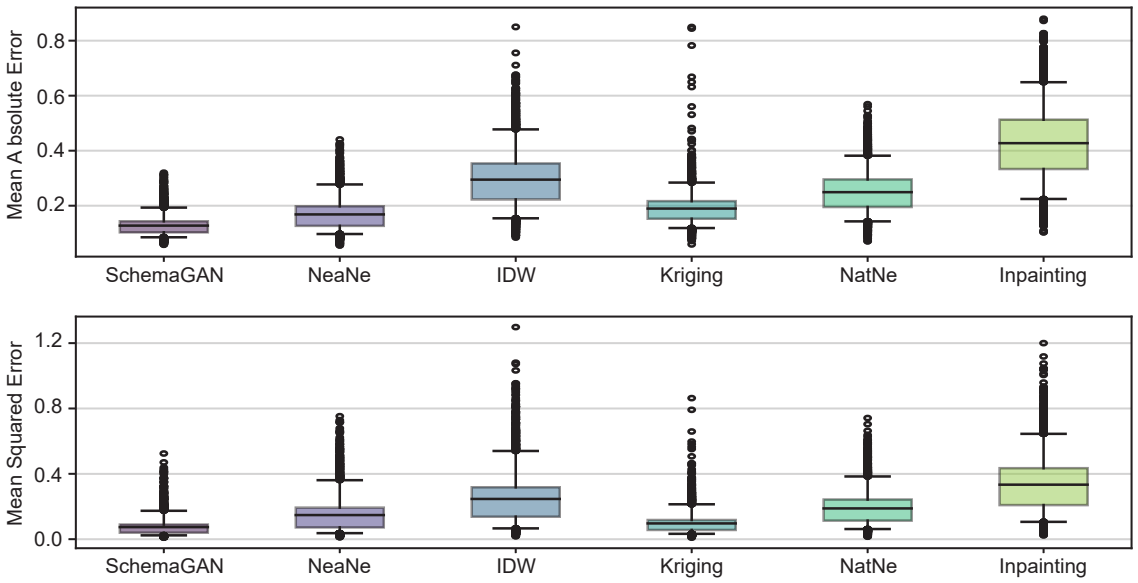


Figure 6.1: MAE and MSE error distribution comparison for the traditional interpolation methods against the SchemaGAN model.

This distribution of complexities provides a context for understanding the resulting box plots. In general, the lower MAE and MSE values are associated with simpler schematizations. Given the reduced complexity, these structures are easier to interpret, resulting in lower error rates. Conversely, higher MAE and MSE values, arise from the interpretation of highly complex geometries. Due to their intricate nature, these schematizations are more challenging to accurately interpolate, thus leading to higher error rates.

Looking at the distribution of errors, SchemaGAN showed a narrow interquartile range for both MAE and MSE, meaning that this method performed well regardless of the complexity of the schematization at hand.

Following, the MAE box plots show NeaNe and Kriging as the second and third-best interpolation methods. NeaNe on average is closer, but its interquartile range is larger than Krigins. This means that NeaNe is more accurate in solving simple geometries while it struggles more with complex ones, while Kriging performs more consistently in the latter ones. This is more evident when looking at MSE results, where kriging is clearly the second-best method with the interquartile range and whiskers much closer to SchemaGAN, confirming that overall is a more consistent method.

The performance of IDW and NatNe methods, when compared to SchemaGAN, falls significantly short in accurately modelling simple subsoil schematizations. The NatNe method slightly outperforms IDW, even when NatNe fails to effectively extrapolate beyond the available CPT data.

Lastly, as elaborated previously, the use of inpainting as an interpolation technique for subsoil schematizations faces a significant challenge due to the vast interpolation space relative to the sparse data provided. Operating with less than 1% of input data is insufficient for this method, which results in the highest MAE and MSE distributions from the comparison.

6.2.2. The Computational Efficiency Comparison

Following the examination of the evaluation metrics of each method, it is equally crucial to consider the computational efficiency of the techniques in practice. The inference time required to generate all 4,000 images in the test database varied widely among the methods and is summarized in Table 6.2.

Table 6.2: Inference time for the interpolation of the 4,000 schematizations in the test database.

Method	SchemaGAN	NeaNe	IDW	Kriging	NatNe	Inpainting
Computation Time (m)	7.0	1.1	14.8	102.2	18.7	11.8

In terms of speed, NeaNe emerged as the fastest method, completing the task in just over a minute. This speed is due to the method's simplicity, without requiring complex calculations. The SchemaGAN method, while more computationally demanding than NeaNe, demonstrated a reasonably efficient total execution time. It took a little over seven minutes to generate all 4,000 images. This method strikes a balance between the sophistication of DL models and computational speed, especially considering the quality of its results.

It's important to note that this time frame doesn't include the substantial time needed to train the SchemaGAN model initially. While this training phase is a one-time requirement, it demands a significant investment of computational resources and time. For SchemaGAN, training took 95 hours on a NVIDIA Tesla V100S 32GB of graphic card on DelftBlue supercomputer.

On the contrary, the remaining interpolation methods, kriging, IDW, NatNe and Inpainting, exhibited considerably longer computation times. Among them, Kriging stood out as the most time-consuming, taking over an hour to generate all the images. It's essential to approach all these results with caution, as none of the traditional interpolation methods are fully optimized. Specifically, Kriging involves an iterative optimization step that significantly increases the computation time.

6.3. Qualitative Evaluation

While quantitative evaluation provides a numerical measure of SchemaGAN performance, qualitative assessment is essential for a more comprehensive understanding of the interpolation methods strengths and weaknesses in comparison. This analysis focuses not only on the statistical accuracy of the models but also on the overall visual quality of the predicted images. This includes aspects such as structural coherence, shape preservation, and visual resemblance to the actual original cross-sections, which are details that might be overlooked in a strictly numerical evaluation.

In order to visually represent the differences in the methods, three cross-sections have been selected, which encompass varying degrees of complexity. The first is a simple model with sub-horizontal layers (Figure 6.2), the second presents moderate complexity with the inclusion of indentations (Figure 6.3), and the third is a highly complex model with irregular layers, indentations, and lenses (Figure 6.4).

When considering simple and moderately complex subsoil schematizations, the SchemaGAN performs exceptionally well, capturing the overall structure and layering of the original cross-section, including indentations and with defined layer boundaries and anisotropy. For more complex geometries, the limitations of SchemaGAN begin to show. This is especially notable in the shape of the old channel in Figure 6.4, which although correctly captures, has poorly defined edges. Furthermore, SchemaGAN is the best method to capture the internal anisotropy in the layers, as illustrated also in Figure 6.4 on the thick top layer.

NeaNe and Kriging generate results that are very similar to SchemaGAN and the original schematization when looking at very simple sub-horizontal geometries (Figure 6.2). The only visual difference is the blurrier layer boundaries identified in the MAE map for the Kriging method. When the complexity of the subsoil models is increased, NeaNe and Kriging quickly fall apart for different reasons.

The biggest problem for NeaNe is the lack of continuity in the generated schematizations. This is an artefact product of the method itself and introduces sharp vertical boundaries that impart a box-like appearance to the interpolation. This appearance is far from the original cross-section and not at all how the natural subsoil layer behaves. Moreover, the NeaNe method struggles where the CPT data do not reach the bottom of the cross-section, like in Figure 6.4 at the distance of 400. Here, the interpolation incorrectly predicts the Ic value, cutting the logical behaviour of subsoil layers.

On the other hand, as complexity increases, the blurriness in the layer boundaries in the Kriging interpolation increases. This effect leads to very soft boundaries which difficult the understanding of the subsoil structure as in Figure 6.4 where the indentation and old channel are very difficult to visualize. Furthermore, the anisotropy inside the layers is completely lost due to the same effect.

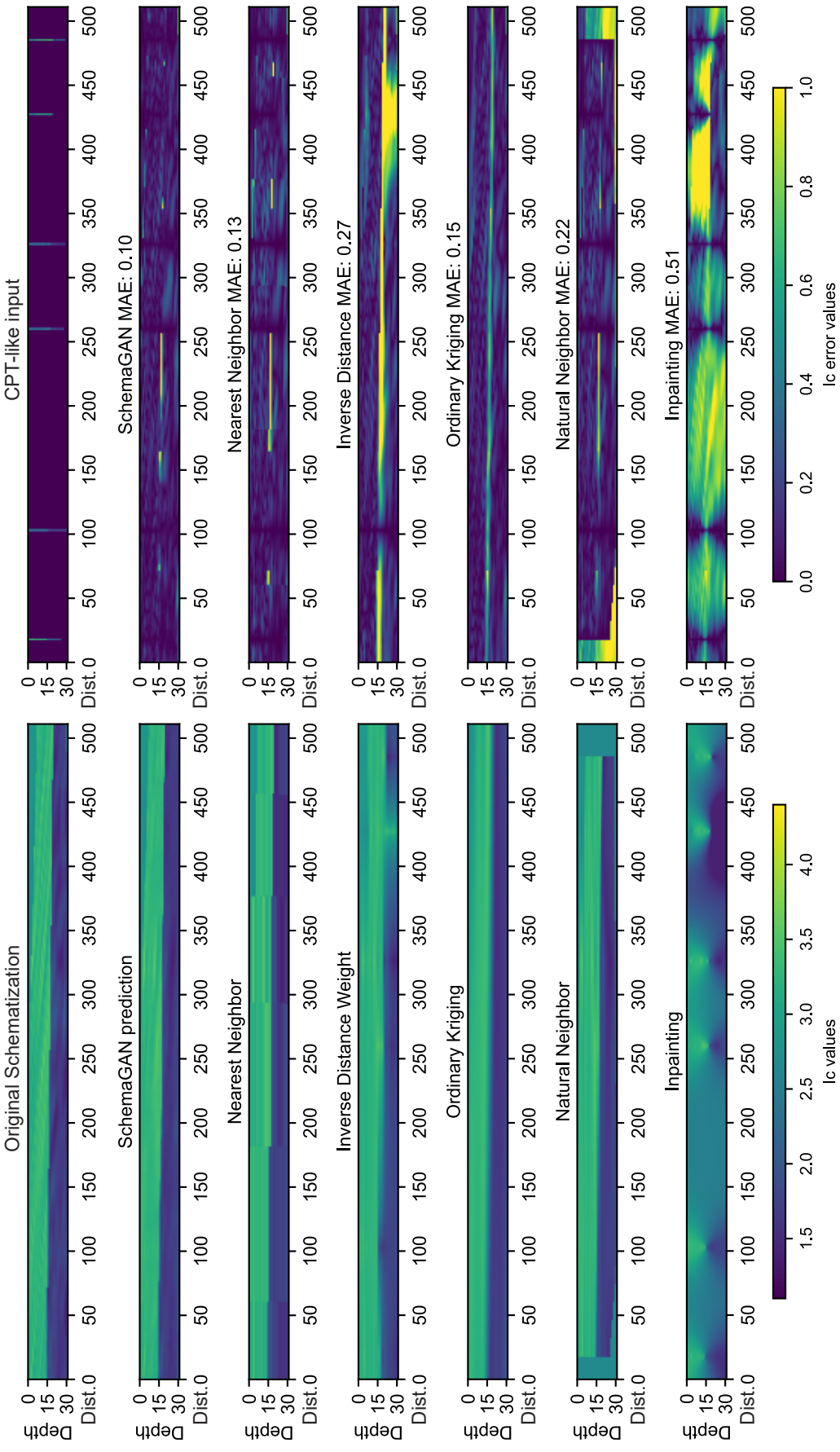


Figure 6.2: Visual comparison of simple subsoil schematization with subhorizontal layering, generated by SchemaGAN and the traditional interpolation methods. The set includes the original cross-section, the CPT-like conditional input used for predictions, and the MAE error map for each method.

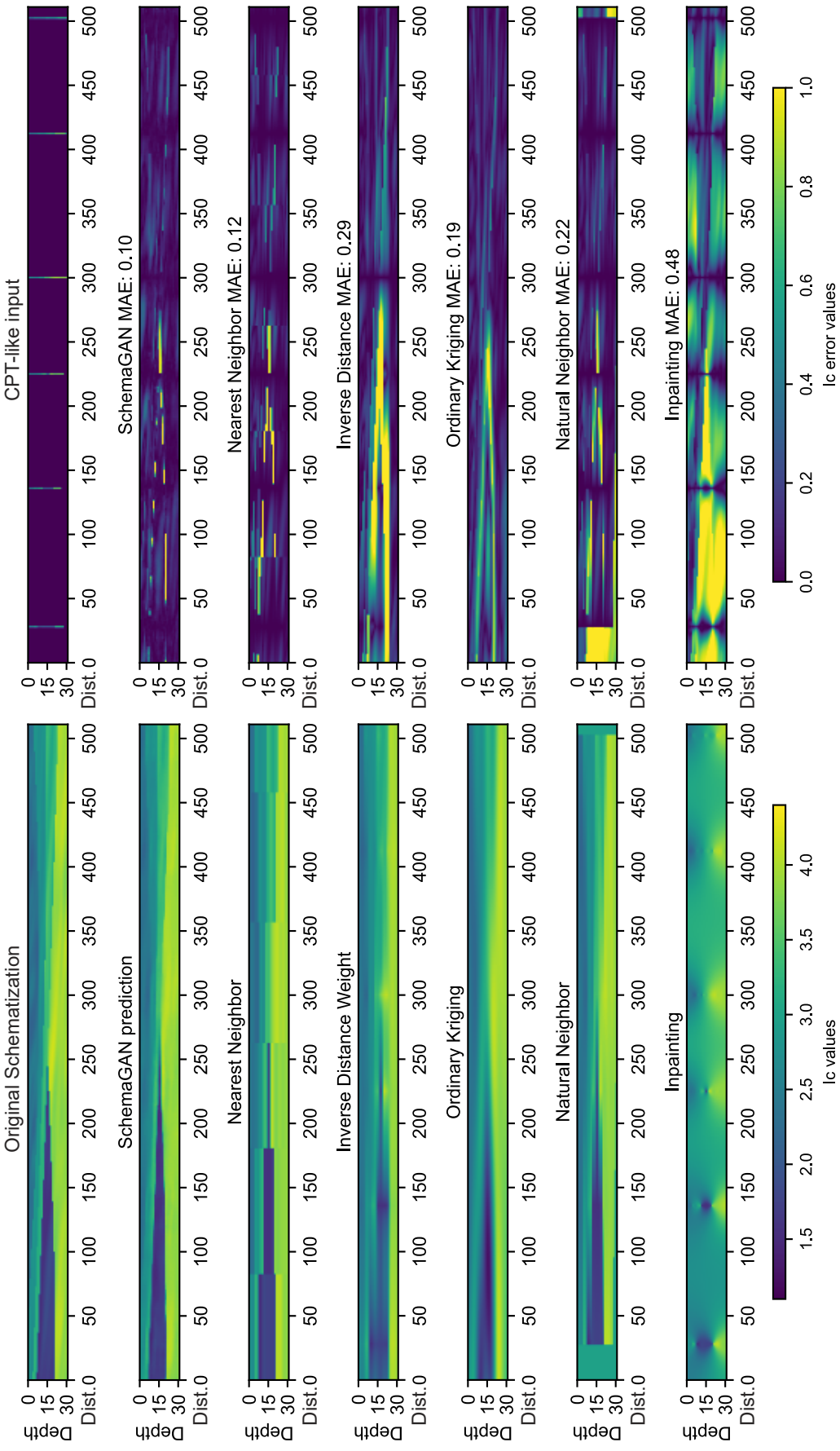


Figure 6.3: Visual comparison of moderately complex subsoil schematization with irregular layers and indentations, generated by SchemaGAN and the traditional interpolation methods. The set includes the original cross-section, the CPT-like conditional input used for predictions, and the MAE error map for each method.

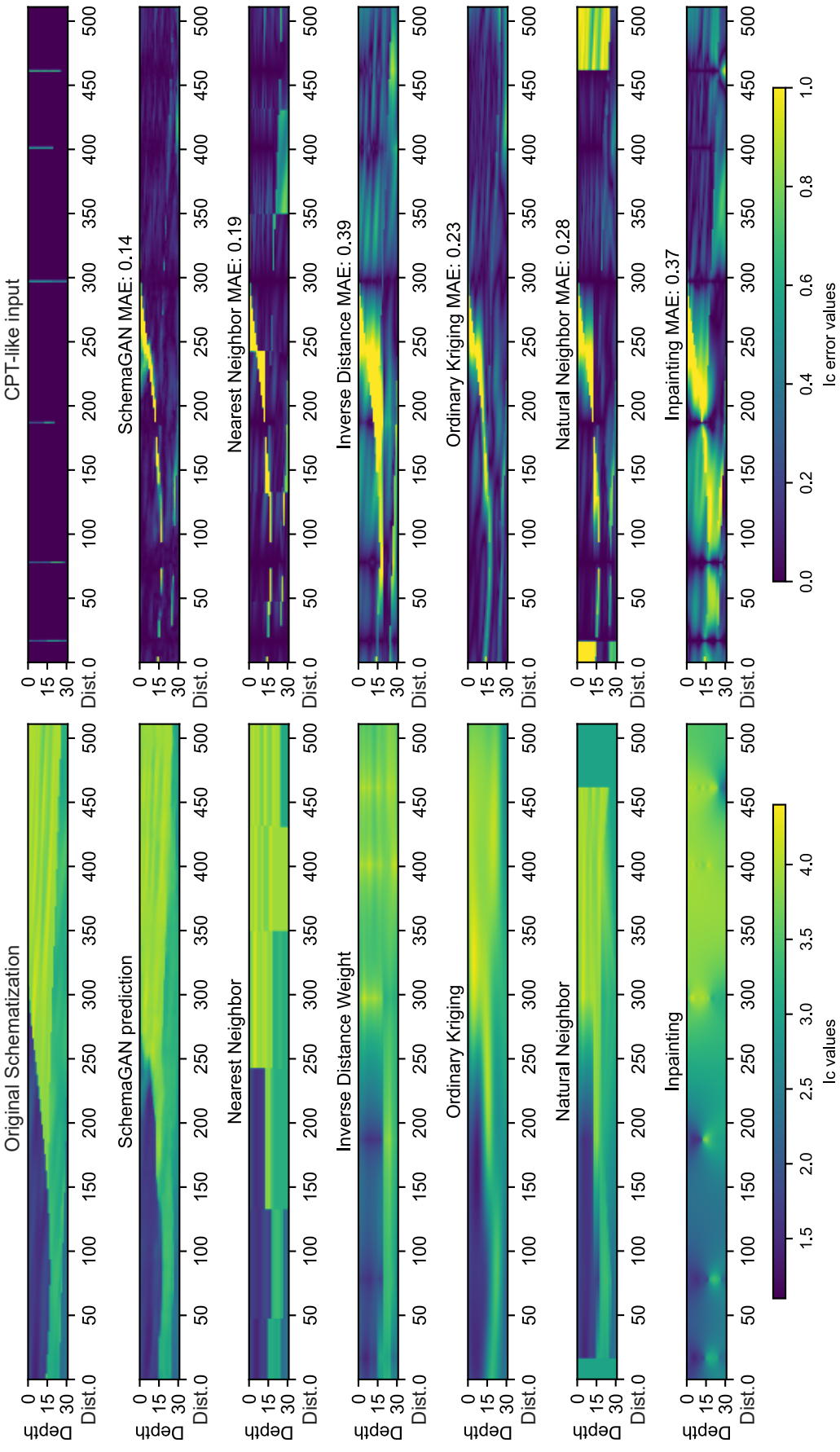


Figure 6.4: Visual comparison of a highly complex subsoil schematization with irregular layers, indentations and lenses, generated by SchemaGAN and the traditional interpolation methods. The set includes the original cross-section, the CPT-like conditional input used for predictions, and the MAE error images for each method.

In the case of IDW and NatNe, worst overall performance is observed. In very simple sub-horizontal models, both methods manage to capture the structure but break down rapidly as complexity increases. IDW suffers from overly blurry layer boundaries to the point where the structure of the system is completely lost. NatNe suffers from the same problem with the addition of the lack of capacity to extrapolate, which makes the edges of the schematization always incomplete.

Finally, inpainting does not work well for subsoil schematization. As it was previously discussed, it does not manage to properly interpolate with the provided amount of input data regardless of the complexity of the schematization.

6.4. Expert Criteria Survey

An extended qualitative evaluation was executed by asking the expert opinion of geotechnical engineers in ranking the schematizations generated from all the working methods; SchemaGAN, NeaNe, IDW, Kriging and NatNe. Inpainting was excluded as it was unable to generate proper results.

The experts were asked to rank 10 series of schematizations from worst to best, based on their expert criteria, considering how accurately the schematization reflects the provided CPT-like data, the quality and the realism in representing a subsoil layered system, and the ease at which each schematization provided an understanding of the subsoil layers. Each one of the 10 series consisted of 6 images: The provided CPT-like data came on top, and under it ordered at random and named from A to E, came the generated schematization from the 5 selected methods. For each series the expert would rank each generated image from 1 to 5, signifying worst to best, without the possibility of repetition. A wide range of complexity in the geometry of the schematizations was chosen as the 10 series samples and can be observed in Appendix A.

Results from the survey clearly indicate that when presented with 5 different schematizations methods, experts prefer 78% of times the solution of SchemaGAN, and in much less proportion NeaNe and Kriging were chosen as the best methods (Figure 6.5). In not a single instance were IDW or NatNe chosen as the best interpolation methods for the task at hand.

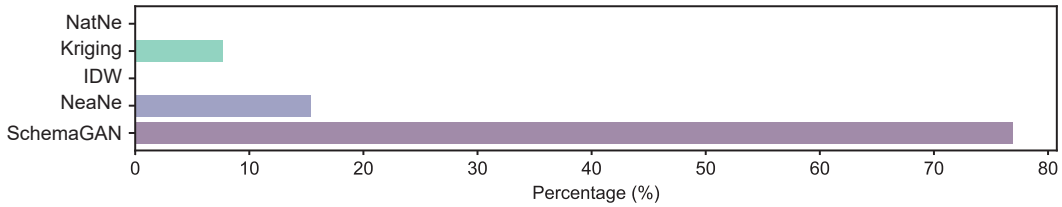


Figure 6.5: Percentage of times that each method was chosen as the best subsoil schematization by expert criteria.

All the expert criteria choices are presented in the violin plot of Figure 6.6, with the width of the violins representing the density of the data at a given score. The preferred method by experts is SchemaGAN, with the widest section at the score of 5 and narrowing towards the lower scores. Following is the Kriging interpolation, with its widest section at value 4 and slowly tapering down to the worst values. The NeaNe interpolation has a constant width throughout all the scores which means that some experts prefer it and some consider it the worst depending on the schematization. On the other extreme are IDW and NatNe, which scored consistently the lowest from the expert criteria, as indicated by the widest sections at the lower scores in the plots.

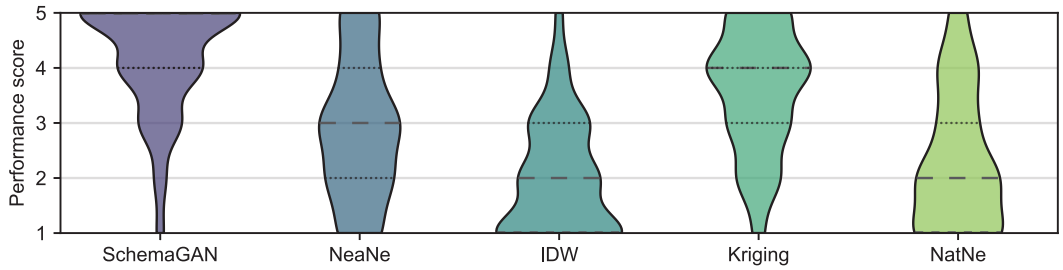


Figure 6.6: Percentage of times that each method was chosen as the best subsoil schematization by expert criteria.

These results fall in line with findings presented by Heuvelink and Webster (2022), which indicated that ML has tended to oust geostatistics for soil mapping in recent years.

6.5. Influence of the CPT Location on the Schematization

A small study into the influence of the location of the CPT-like data in the generation of the subsoil schematization was carried out, in order to get further insight into the advantages and disadvantages of the different interpolation methods. For this two particular synthetic models were chosen: a simple sub-horizontal layered system and a complex irregular model with lenses and indentations (Figure 6.7).

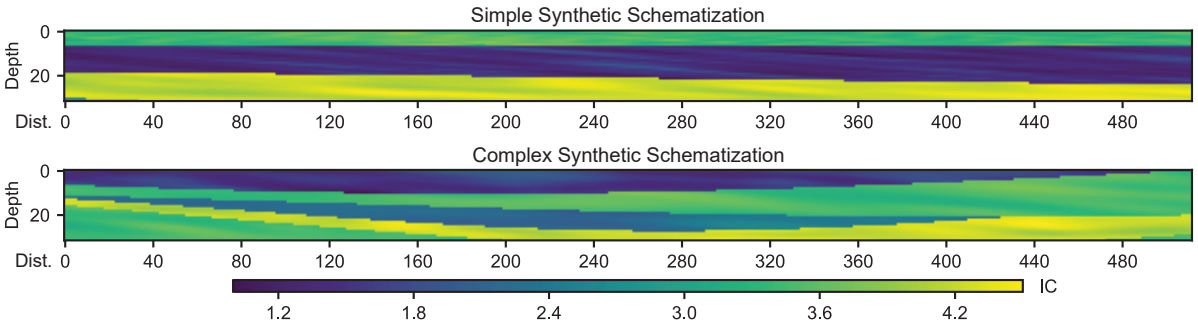


Figure 6.7: Simple and complex geometry synthetic subsoil schematization for investigation of the influence of the CPT location.

The results, presented in Figure 6.8, clearly show that SchemaGAN consistently performed better than the other methods, regardless of where the input CPT-like data is placed. This is evident from the very narrow interquartile range and the lowest MAE values for both simple and complex models. The next best methods are NeaNe and Kriging. In simpler models, the NeaNe interquartile range is wider because it struggles when CPT data misses a change in layers at the bottom. Kriging, although more consistent throughout the increasing complexity of the models, is more affected by the location of the CPT data than the previous two methods.

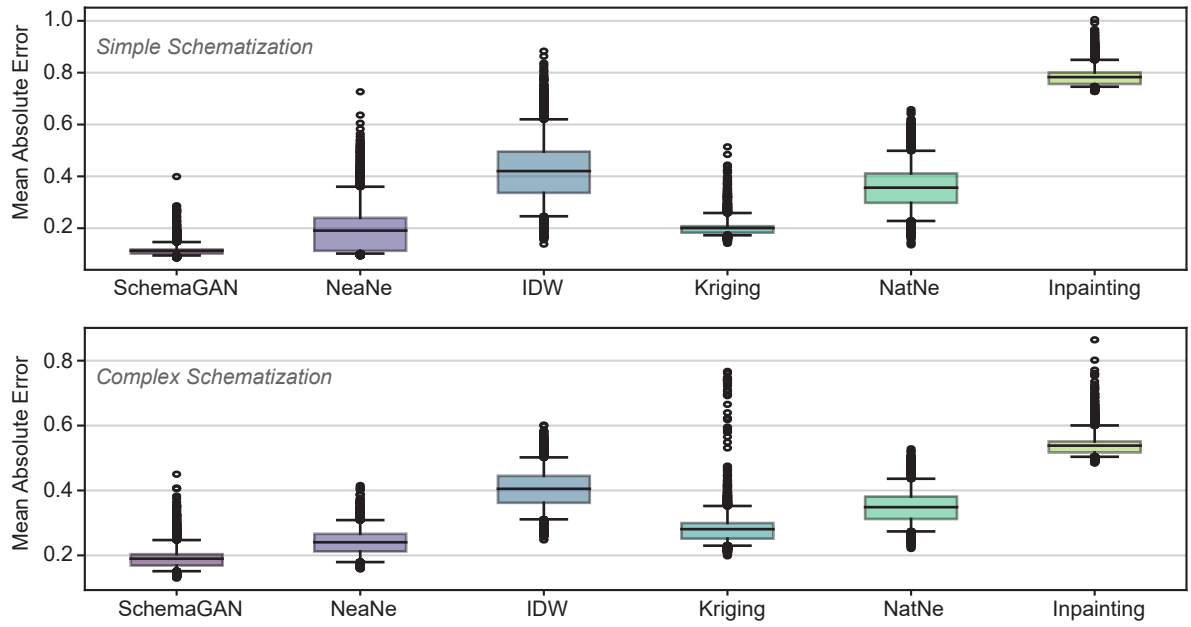


Figure 6.8: Influence of the CPT location for the generation of schematizations with SchemaGAN and the traditional interpolation methods.

IDW and NatNe show the biggest interquartile range of all the methods on both complexity levels, which means that these methods are the most affected by the position of the CPT data used as input for the generation. Inpainting, due to its failure in generating realistic schematizations, is not affected by the position of the CPTs, as it will fail regardless because of the low amount of data provided for the method.

6.6. Case Study: The Delft-Schiedam Railway Embankment

The final evaluation consisted of a case study with real in-situ data, in which SchemaGAN and the best-performing interpolation methods were used to predict the known CPT surveys.

6.6.1. Project Overview

The Rail Embankment between Delft and Schiedam, in the province of South Holland, was chosen for the case study to test SchemaGAN with real CPT data. This site provided 51 CPT surveys arranged linearly, which falls in line with the capacity of SchemaGAN to generate schematizations. The exact location of the area is presented in Figure 6.9. The CPT data available has a semi-ordered spacing of 100 m between most surveys, with surveys as far apart as 600 m, and closer than 1 m in some cases. The maximum depth reached is -36 m NAP.

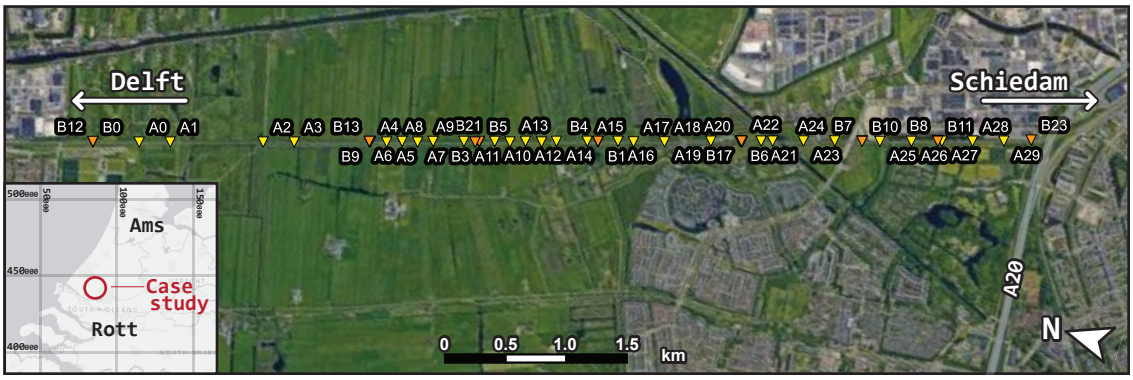


Figure 6.9: Overview of the Delft-Schiedam rail embankment and CPT surveys in the South Holland province.

6.6.2. SchemaGAN and Traditional Interpolations on Real Data

In order to adapt the actual CPT data to the 32x512 px format required by SchemaGAN, resampling of the real data was necessary. For the vertical dimension, the Ic values were calculated as group averages to fit into the 32 px height. In terms of horizontal distance, the 6,100 m length of the rail embankment was divided into two 512 px sections. Furthermore, Additionally, the positioning of the CPTs was aligned with the Amsterdam Ordnance Datum (NAP), and a top cutoff was applied to create a flat surface.

A tool was developed to randomly select 6 CPT surveys within each cross-section. This selection adhered to the same criteria used for generating synthetic CPT-like data, ensuring a balanced distribution of surveys while avoiding extreme proximity or distance between them. This resulted in two sets of CPT data for each cross-section: one for generating the schematizations and the other as a comparison against results produced by different methods.

Schematizations were generated with SchemaGAN, Nearest Neighbour interpolation and Kriging interpolation– identified as the most effective methods through the analysis of synthetic data. After generating each schematization, the precise position of the comparison CPTs was captured for evaluation. MAE and MSE were calculated as performance metrics. This process was iterated 1,000 times for each cross-section, ensuring a comprehensive assessment of all possible combinations of available CPTs for schematization.

Results obtained are shown as boxplots for MAE and MSE in Figure 6.10. The findings indicate that SchemaGAN outperforms other methods in predicting real CPT data. This conclusion is supported by the narrower and lower mean and interquartile range in the results. The NeaNe interpolation achieves a lower minimum MAE value than SchemaGAN, a product of the comparison with CPT data that is less than one meter apart from each other. Otherwise, NeaNe is the worst-performing method from the comparison. In agreement with Rahman et al. (2021), the Kriging interpolation scores closer to SchemaGAN, with the main differentiator being the upper interquartile range of results.

In Figure 6.11, the schematizations generated by SchemaGAN using real CPT data for the Delft-Schiedam rail embankment are shown. It's important to note that since a definitive ground truth for the case study is unavailable, a direct comparison isn't feasible; therefore the metrics were computed for the specific known CPT location. The visual representation illustrates the outcome of significant data averaging, ranging from an extensive dataset of

up to 1,800 Ic measurements for the deepest CPT test, down to 32 Ic values. Consequently, the generated image exhibits a considerable level of noise, leading to an indistinct internal layer structure within the layers.

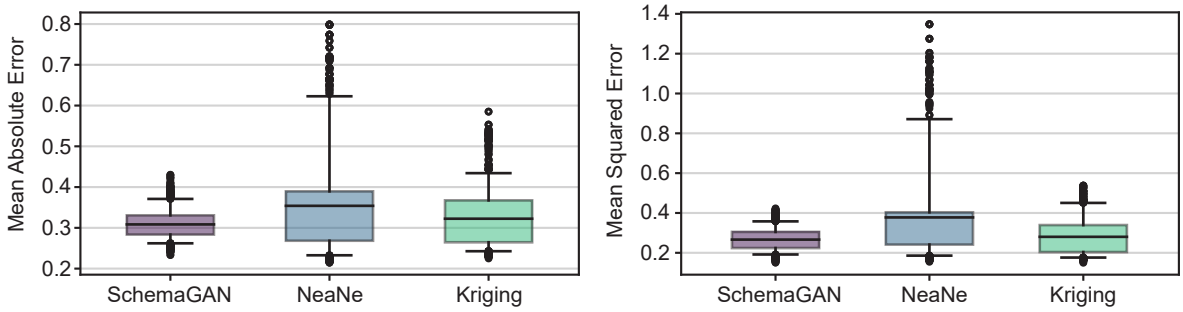


Figure 6.10: Evaluation metrics from the best-performing methods on the real CPT data for the case study.

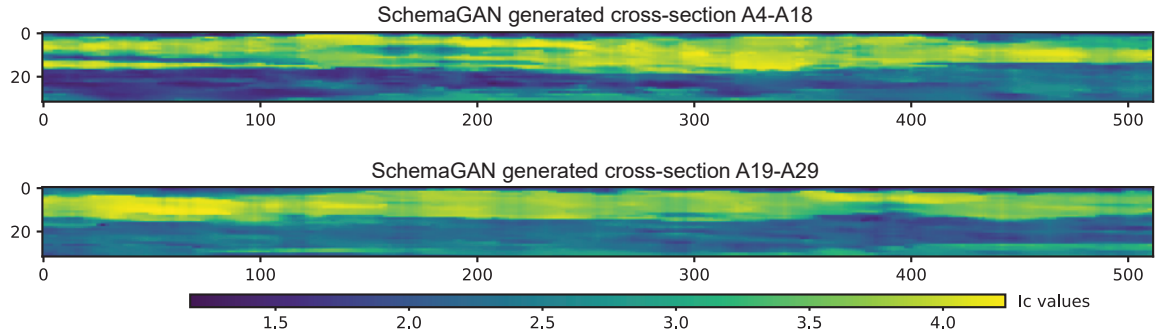


Figure 6.11: SchemaGAN schematization of the case study CPT data.

6.7. Summary of the Comparison Findings

The study effectively applied five distinct interpolation methods: NeaNe, IDW, Kriging, NatNe, and Inpainting, establishing a robust framework to assess the comparative performance of the SchemaGAN model in handling subsoil schematizations.

From the presented results, it is evident that the SchemaGAN model surpasses traditional interpolation methods in producing subsoil schematizations, both in terms of accuracy and computational efficiency. SchemaGAN displayed superior performance in its predictions, reflected by achieving the lowest error metrics. Traditional methods such as NeaNe and Kriging delivered a reasonable performance but fell short of the precision SchemaGAN exhibited. Other techniques, including IDW, NatNe, and Inpainting, registered higher MAE and MSE values, indicating they have limitations in accuracy. Rahman et al. (2021), looking at pure CPT data, also supports that Kriging is one of the best-performing methods. For his case studies, IDW was equally good at interpolation, which is not the case for the subsoil schematizations most likely due to their 2D nature.

In terms of inference time, it is important to consider that the results should be approached with a degree of caution, primarily due to the non-optimized nature of the traditional interpolation methods that have been employed. Additionally, the incorporation of an optimization step within the Kriging interpolation process contributes to an increased time requirement for result generation. Despite these considerations, the analysis offers valuable insights into the computational efficiency of the conventional methods when contrasted with SchemaGAN.

Among the methods evaluated, NeaNe is the fastest, with SchemaGAN following closely behind. Subsequent in line are the Inpainting, IDW, and NatNe. Notably slower is the Kriging interpolation due to the optimization step needed. These are good results for SchemaGAN, and are in agreement with Laloy et al. (2017) which indicates that once the GAN model is trained, inference times are very fast in comparison to geostatistical alternatives.

An essential observation to underline is that despite the efficiency of SchemaGAN in producing schematizations, a substantial computational and temporal investment is required during the training phase. It is imperative, however, to emphasize that this commitment only needs to be undertaken once.

As far as the qualitative evaluation of SchemaGAN and traditional interpolation methods, it is also clear that the SchemaGAN model consistently outperforms traditional methods, demonstrating superior capability in handling all ranges of complex schematizations. It manages to adeptly reproduce the overall structure and details, maintaining a high degree of structural coherence and visual resemblance to the original cross-sections. These results are in agreement with research carried out for geological fluvial facies, in which Sun et al. (2023) also concluded that GANs outperformed geostatistical methods in reproducing complex geometries.

Traditional interpolation methods such as Kriging and Inverse Distance Weighting (IDW) often lose definition at layer boundaries, leading to blurring and a loss of finer details. They also struggle more as the complexity of the subsoil structures increases. Particularly notable is the SchemaGAN superior capability, proving to be robust regardless of the location of the CPT-like data, and even in situations where CPT data are sparse or do not fully penetrate to the bottom of the cross-section.

An important conclusion beyond the numerical accuracy is that SchemaGAN provides outputs that have a more natural appearance, unlike several traditional methods. NeaNe results are unnatural and box-like, lacking continuity in the layering, while Kriging and IDW introduce excessive blurring. This is confirmed by the expert criteria which very clearly supports SchemaGAN as the best-performing and preferred method for subsoil schematizations in 78% of the presented cases. Strikingly similar results were obtained by expert criteria in meteorological generative results, in which experts considered in 73% of the cases the AI-generated result as the best one (Ravuri et al., 2021).

For the application of real CPT data, SchemaGAN continues to outperform the traditional methods, although by a lower margin. This is more impressive if it is considered that SchemaGAN is trained only on synthetic data. NeaNe interpolation scores well in the MAE comparison when exceptionally close CPT data is considered. This scenario is not always realistic in real cases. When normal spacing is considered, the error increases dramatically. Kriging on the other hand performs close to SchemaGAN, confirming once again as the best interpoaltion method after the developed DL tool.

7

Discussion and Future Directions

7.1. Introduction

In this discussion chapter, the focus is on the critical examination and interpretation of the findings of the study. A detailed exploration of the implications of the results within the broader context of geotechnical engineering is conducted. The performance of the DL approach, its limitations, and its comparison to traditional methods are comprehensively addressed. Furthermore, the influence and potential of these results on future research directions in the field of subsoil schematization are deliberated upon. This chapter aims to bring clarity and meaningful interpretation to the results, contributing to the broader knowledge base of Geotechnical Engineering.

7.2. Effectiveness of Deep Learning Algorithms in Subsoil Schematization

Addressing the first research question, the performance of SchemaGAN has been scrutinized in the context of subsoil schematization. The evaluation attests to SchemaGAN's robust performance, particularly when dealing with synthetic data, as it consistently yielded promising results throughout all phases of the study.

On examining a broad spectrum of complexity within the synthetic subsoil schematization databases, it is clear that SchemaGAN capably captures and recreates the general geometry and structure of the original images (Figure 5.11). This holds true even for the inherent variability in Ic values within each layer. Impressively, the generator can recreate the original image effectively, even when furnished with less than 1% of the original image as input.

The Generator unique loss function, which combines BCE loss and MAE loss, played a major role in this success. It helped create schematizations that looked real and had a high level of accuracy. On the other hand, the Discriminator also played a crucial part, contributing significantly to the quality of the schematizations. However, it is crucial to note that the architecture and training regime of SchemaGAN hasn't been optimized fully. While there is potential for further fine-tuning of the network's architecture and hyperparameters, the overall performance of SchemaGAN indicates its effectiveness in producing subsoil schematizations from CPT-like data. This success underlines a positive answer to the first research question of this project.

Although quite effective, the SchemaGAN model is far from perfect. There is a clear and direct difficulty encountered when the geometry and structure of the model become more complex. In this scenario, the model struggles with the definition of the layer boundaries (Figure 5.13). Nevertheless, this does not overshadow the potential of SchemaGAN. While there are areas for improvement and challenges to overcome, the results to date provide a solid foundation for future exploration and refinement. Given the current findings, it is evident that SchemaGAN has considerable potential to bring advancements in the realm of Geotechnical Engineering, offering a novel, effective, and promising approach to subsoil schematization.

7.3. Comparison of SchemaGAN and Traditional Interpolation Methods

The second focus of this discussion revolves around the second research question. The aim is to compare the effectiveness of SchemaGAN with traditional geostatistical interpolation methods. To maintain consistency and

ensure a fair comparison, the same 4,000 samples database used in the SchemaGAN testing phase was employed.

NeaNe proved to be a strong interpolation method, particularly with the subhorizontal layers often found in naturally occurring subsoils. While it achieved the best numerical results among the traditional methods, the schematizations often lacked continuity and were visually unappealing (Figure 6.4). This problem is attributed to the computation principle of the method. This meant that the overall structure and geometry of the cross-sections were not captured accurately. This condition worsened as the models become more and more complex. On the other hand, NeaNe demonstrated the quickest computation time, generating all 4,000 interpolations from the testing dataset in only 1.1 minutes. This efficiency is largely attributed to the simplicity of the method.

Kriging stands out numerically as the second-best traditional interpolation method for subsoil schematization. Qualitatively, this method worked the best in comparison with all others, second only to SchemaGAN. It has a good overall capacity in capturing the structure of the schematization, but falls short in preserving precise layer boundaries, resulting in blurry edges (Figure 6.3). This problem is more critical in complex models where it may result in the loss of thinner structures in the schematization.

The processing time of the Kriging interpolation, considering the optimization process it requires, is considerably longer than for any other tested method. When compared with SchemaGAN, Kriging required nearly 14 times more time to process the same volume of data.

The IDW method demonstrated a marked performance decrease in the results, often failing to effectively replicate the intricate subsoil schematizations. A significant issue with IDW was its tendency to produce overly blurred images, effectively merging layers in the schematizations and thus failing to accurately represent the subsurface's structural diversity (Figure 6.3). This limitation primarily arose in complex models where the precise delineation of distinct layers was critical for an accurate representation.

The blurring effect stems from the method underlying approach. By taking the weighted average of surrounding data points, IDW inevitably smooths out transitions, which in the case of subsoil schematizations, leads to the blurring of distinct layer boundaries. This effect is more prominent in scenarios with abrupt changes, such as sharp layer boundaries, causing the method to generate an output that compromises between the two significantly different data points. As a result, it fails to handle sharp transitions effectively, contributing to the blurred appearance and the merging of layers in the output.

The NatNe interpolation method demonstrated shortcomings in this study, particularly when faced with complex geometric structures. A predominant issue was its inability to extrapolate data. In contrast to other methods, NatNe does not estimate values beyond known data points, leading to an absence of data along the edges of the schematizations (Figure 6.4). This issue critically impacts the accuracy of the method, especially when boundary regions incorporate significant features.

Finally, the application of classical inpainting as an interpolation method for subsoil schematizations posed significant challenges in this study. Inpainting techniques are primarily based on the biharmonic equation, which assumes that the interpolation region is densely surrounded by known data. However, this assumption doesn't hold in a geotechnical engineering context where data is often sparse. In this study, where less than 1% of input data was available, the classical inpainting method was severely disadvantaged, resulting in poor performance.

In both qualitative and quantitative analysis, SchemaGAN outperformed the traditional interpolation methods. This superiority was particularly evident in complex schematizations, where SchemaGAN displayed its strengths in managing intricate geometries and structures. This section's findings answer the second research question, confirming that SchemaGAN, can indeed be more effective than traditional interpolation methods in subsoil schematization.

These results are supported by the criteria of geotechnical experts which after comparing blindly schematizations generated with SchemaGAN and the geostatistical interpolation methods, considered in 78% of the cases that SchemaGAN produced the best results. When confronted with complex geometries, experts quickly chose SchemaGAN as the best method, but when looking at very simple subhorizontal models, experts often struggle to choose between NeaNe, Kriging and SchemaGAN.

Furthermore, when applying SchemaGAN, NeaNe and Kriging to real CPT data from the Delft-Schiedam rail embankment, SchemaGAN performed the best. It managed to generate the closest CPT results to the actual real CPT surveys in the area, followed by Kriging and NeaNe. This is quite a remarkable result when considering that SchemaGAN has been trained only in synthetic data. These results support the fact that the synthetic databases

generated for training managed to capture sufficiently accurately the naturally occurring subsoil layering for the model to learn and recreate real CPT results.

Finally, confirming the superiority of SchemaGAN in subsoil schematization, after testing for the effect of the CPT location along the cross-section, SchemaGAN emerged as the least affected by it. Regardless of the complexity of the geometry and the location of the surveys, SchemaGAN outperformed again all the traditional interpolation methods.

7.4. Limitations of SchemaGAN in Subsoil Schematization

Despite the promising capabilities of SchemaGAN in the realm of Geotechnical Engineering, it does present several limitations and challenges. Practical implementation of the model requires extensive pre-processing of input data, which involves the conversion of geotechnical survey data, typically collected at single points, into image format with specific dimensions and pixel values representing specific geotechnical properties. This data conversion can be a laborious process and often constrains the method's applicability in real-world scenarios.

A noteworthy limitation pertains to the substantial simplification of CPT data. CPTs often generate thousands of measurements in depth, which are significantly reduced in the SchemaGAN models to 32 px in depth. This considerable data reduction leads to a loss of granularity and potential critical subsoil information, impacting the accuracy of the generated schematization.

Furthermore, SchemaGAN is currently limited to processing flat-top images. This characteristic limits its utility in situations with irregular or varied topography.

Specific challenges emerge with SchemaGAN's difficulty in demarcating layer boundaries, especially in complex subsoil structures. The model tends to produce more errors when dealing with irregular layers, sharp indentations, thin layers, lenses, and old channels. When such intricate geometrical features increase in complexity or undergo sudden changes, they are not captured with high precision.

Owing to its GAN architecture, SchemaGAN is unable to provide a measure of uncertainty or confidence in its predictions. The absence of a quantifiable metric of certainty can pose challenges when interpreting the model output, especially in critical applications where there is no ground truth to evaluate the results.

Despite these limitations, SchemaGAN's performance improves significantly with increased data input. However, in practical Geotechnical Engineering scenarios, acquiring such extensive data is often unrealistic, limiting the effectiveness of the model within the domain of sparse data.

Possible solutions to these challenges include the development of an advanced tool for transforming in-situ CPT data to a format compatible with SchemaGAN. Alternatively, an adaptation of SchemaGAN to handle larger file sizes, or and increase in the resolution could reduce the need for significant data simplification, thereby allowing for a more accurate representation of the input data. Special training with more complex geometries or combined with real CPT data could also help improve the model. Finally, an avenue to add additional conditional inputs like geological and geomorphological maps is also available.

7.5. Implications and Practical Applications

In typical Geotechnical Engineering offices, it is not unusual for geotechnical engineers to use the raw CPT results without any schematization. These raw data sets often rely on the experience and underlying understanding of engineers to derive rapid conclusions.

SchemaGAN, however, strives to alter this paradigm. By harnessing the expansive soil investigation data typically stored on servers and seldom carried over to subsequent projects, it serves as a centralized entity for generating consistent subsoil schematizations. This approach not only enhances data utilization but also mitigates the uncertainty stemming from variable schematizations produced by different engineers.

Significantly, SchemaGAN demonstrates substantial potential in real-world Geotechnical Engineering scenarios, particularly in large-scale linear projects. These projects, which often generate vast amounts of sparse data, require swift and precise interpretation. Examples include dike and embankment stability assessments, railway stability and risk management, and highway construction and stability evaluations.

A key aspect of the SchemaGAN approach is the uniform processing and interpretation of project data, which can be a game-changer, eliminating discrepancies arising from different tools, experiences, or biases among

engineers. Moreover, by providing a quick and reliable initial interpretation of CPT data, SchemaGAN can expedite the identification of potentially problematic areas. This efficiency facilitates more focused requests for further surveys and specific stability or risk analysis, thereby optimizing resource allocation and project timelines.

The broader implications of SchemaGAN extend to the geotechnical industry at large. It highlights the potential of DL applications in improving the speed and reliability of geotechnical processes, thereby freeing engineers to concentrate on the critical aspects of their work.

The SchemaGAN model also holds promise beyond the preliminary investigation phase of geotechnical projects. Given its adaptability, the model can be adjusted to analyze parameters represented as pixel values, such as the distribution of shear strength within the subsoil or hydraulic conductivity and permeability. This adaptability opens up a new frontier of possibilities within Geotechnical Engineering.

7.6. Suggestions for Future Research

One of the principal modifications to look for is the integration of real CPT data into the model to enhance its applicability. Another critical advancement is increasing the image size in SchemaGAN to better accommodate real geotechnical data that typically features thousands of measurements in depth.

Furthermore, the SchemaGAN model could be trained to handle irregular surfaces, extending its utility beyond flat topographies. Altering the data masks can also transform the model function, allowing it to generate single 1D CPT interpretations from known data instead of 2D cross-sections. The existing model can also be modified to explore other geotechnical applications, broadening the scope of its parameters.

The current combined BCE and MAE loss function in the Generator could also be refined. It may be beneficial to experiment with different combinations of losses and metrics, particularly in improving the definition of complex boundaries.

The applicability of SchemaGAN could extend to other specific schematization tasks within geotechnical engineering. For example, analyzing the distribution of shear strength within the subsoil or hydraulic parameters could contribute to stability studies and the assessment of piping risks.

Facilitating the integration of this technology into geotechnical engineering practices will necessitate efforts to dispel the notion of machine learning as a “black box”. Simplifying data processing and user experience is crucial to its adoption. Showcasing real-life applications and their positive outcomes could encourage their acceptance within the professional community.

One of the key challenges for the adoption of machine learning models like SchemaGAN is the prevalent resistance to change within the engineering community. Both geotechnical engineers and other stakeholders in infrastructure projects will need to gradually build trust in these new tools. Investing in training and fostering a mindset open to innovation among new geotechnical engineers will be essential for driving this change.

The rapid development of DL and its applications will undoubtedly impact applications like the SchemaGAN model and its future use and development. It is essential to stay abreast of these technological advancements and adapt the model accordingly.

7.7. Main Takeaways and Next Steps

In response to the research questions, the study established the efficacy of DL algorithms in resolving the subsoil schematization problem, specifically through the SchemaGAN model. This model has demonstrated its proficiency in dealing with complex geometries and structures, standing as a favourable alternative to the traditional interpolation methods.

When comparing the use of DL for subsoil schematization to current practices, it’s crucial to note that, while it holds significant promise in terms of accuracy and efficiency, it’s still in its early stages. There is an existing resistance to change within the engineering community, which poses challenges in the adoption of such machine learning models. Therefore, building trust and facilitating the understanding of these new tools among stakeholders will be crucial for their adoption.

The future of DL in Geotechnical Engineering, specifically in subsoil schematization, holds great promise. There is room for improvement and extension of the current SchemaGAN model, including its adaptability to irregular surfaces and other geotechnical applications, and refinement of the combined BCE and MAE loss function

in the Generator. The continuous evolution of these technologies necessitates staying abreast of these advancements and adapting the models accordingly.

Investments in training and fostering a mindset receptive to innovation among new geotechnical engineers will be essential to facilitate this transformation. Furthermore, showcasing real-life applications and their positive outcomes can encourage the acceptance of these models.

8

Conclusions

This research embarked on an exploration of the efficacy of SchemaGAN, a DL model, for subsoil schematization in Geotechnical Engineering. Central to this investigation were questions about the SchemaGAN model performance relative to traditional geostatistical interpolation methods, its inherent limitations, and potential enhancements for future applications.

The SchemaGAN mode was trained using thousands of synthetic subsoil schematizations, with the results showing that the model learned to generate realistic and high-quality schematizations of subsoils even when provided with less than 1% of the original data as an input. Through the training and validation process, three distinct phases became apparent. One of rapid learning for both the Generator and the Discriminator, one of model refinement and one final of mode collapse.

After analyzing the results from the training and the validation on all the available Generators, model number 36 was chosen as the most optimized one to carry on and evaluate its performance.

The architectural choice of SchemaGAN proved to be effective for the task of solving the translation from conditional CPT-like data to a complete subsoil schematization. The research indicated that the combined BCE and MAE loss function in the Generator was a good engine for the task at hand.

The comparison of SchemaGAN with traditional interpolation methods such as NeaNe, IDW, Kriging, NatNe, and Inpainting was a fundamental part of this study. Results indicated that SchemaGAN outperformed these traditional methods, and demonstrated a capability to accurately capture the intricate structure of complex geotechnical schematizations.

Several problems in the traditional interpolation methods for subsoil schematization are worth highlighting. For the NeaNe method, it was found that this technique often led to a lack of continuity. Even when the MAE results are not far from the SchemaGAN results, visually, the produced schematizations are far from correct. Kriging visually captured the structure of the models better than NeaNe, but the blurriness of the layer boundaries made the average MAE slightly lower.

IDW gives results that are overly blurred and in complex stratigraphies, losses the overall structure of the original image. NatNe suffers from not being able to extrapolate on the edges of the schematization, and additionally also outputs blurry layer boundaries. Finally, the classic inpainting method breaks down due to the lack of input information from which to perform the interpolation.

SchemaGAN on the other hand, managed to effectively handle issues that traditional interpolation methods found challenging, such as accuracy and definition of the layer boundaries even in complex models, and low Inference times once the initial model has been trained.

It's clear from the results in the comparison that there is a need for a metric calculation that better captures the visual quality and accuracy of the generations with respect to the original. For the time being, a combination of MAE and MSE with a qualitative analysis proved to be the best approach.

Experts support SchemaGAN as the best method for schematization from the ones studied in this research. Throughout a blind survey, they preferred SchemaGAN results in 78% of the cases. Furthermore, SchemaGAN

also suffers the least from variations in the CPT location along the cross-sections and performs the best when predicting real CPT data. This is a major success for SchemaGAN considering that the model has only been trained on synthetic data, and leaves room for further improvements.

Despite the positive results, the study also uncovered several obstacles in the application of SchemaGAN in Geotechnical Engineering. Extensive data pre-processing, inability to manage irregular or varied topography, challenges in demarcating layer boundaries in complex subsoil structures, and the necessity to simplify CPT data to fit into the model architecture, are among the identified limitations. These aspects contribute to an increased complexity in utilizing SchemaGAN, highlighting areas where the model can be improved.

As for future enhancements, the research pinpointed several critical areas. Among these are an increase in the image size of subsoil schematizations, optimization of data handling to minimize pre-processing, combining the training with real data, refinement of the Generator loss function to improve performance, an adaptation of SchemaGAN to handle irregular topographies, and the use of specialized complex geometry databases. Addressing these identified limitations could unlock the full potential of SchemaGAN in Geotechnical Engineering applications.

Notably, the research recognized a prevailing resistance to change within the engineering community. To facilitate the adoption of SchemaGAN and similar models, a strategic approach is needed. This includes comprehensive training of geotechnical engineers to foster an innovation-friendly mindset, and promote the tangible successes of these models in real-world scenarios.

References

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein gan.

Azevedo, L., Paneiro, G., Santos, A., & Soares, A. (2020). Generative adversarial network as a stochastic subsurface model reconstruction. *Computational Geosciences*, *24*, 1673–1692. https://doi.org/10.1007/s10596-020-09978-x

Baghbani, A., Choudhury, T., Costa, S., & Reiner, J. (2022). Application of artificial intelligence in geotechnical engineering: A state-of-the-art review. *Earth-Science Reviews*, *228*, 103991. https://doi.org/10.1016/j.earscirev.2022.103991

Bau, D., Zhu, J.-Y., Wulff, J., Peebles, W., Strobelt, H., Zhou, B., & Torralba, A. (2019). Seeing what a gan cannot generate.

Beiyang, Y. (2022). *Machine learning for prediction of undrained shear strength from cone penetration test data*. Delft University of Technology. https://repository.tudelft.nl/islandora/object/uuid:f5d58ecc-2b15-46c1-8ecb-1f56e1edf0f2?collection=education

Bertalmio, M., Sapiro, G., Caselles, V., & Ballester, C. (2000). Image inpainting. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, 417–424. https://doi.org/10.1145/344779.344972

Bhattiprolu, S. (2021a). *Tips tricks 15 - understanding binary cross-entropy loss*. https://youtu.be/Md4b67HvmRo

Bhattiprolu, S. (2021b). *Image to image translation using pix2pix gan*. https://github.com/bnsreenu/python_for_microscopists

Bland, W., & Rolls, D. (1998). *Weathering: An introduction to the scientific principles* (1st). Hodder Arnold.

Bogus, W., & Godlewski, T. (2019). Philosophy of geotechnical design in civil engineering – possibilities and risks. *Bulletin of the Polish Academy of Sciences*, *67*.

Bossi, G., Borgatti, L., Gottardi, G., & Marcato, G. (2016). The boolean stochastic generation method - bosg: A tool for the analysis of the error associated with the simplification of the stratigraphy in geotechnical models. *Engineering Geology*, *203*, 99–106. https://doi.org/10.1016/j.enggeo.2015.08.003

Bowles, J. E. (1996). *Foundation analysis and design*. McGraw-Hill.

Brownlee, J. (2019). *How to develop a pix2pix gan for image-to-image translation*. https://machinelearningmastery.com/how-to-develop-a-pix2pix-gan-for-image-to-image-translation/

Budhu, M. (2015). *Soil mechanics fundamentals* (1st). John Wiley & Sons.

Campbell, S. D., Jenkins, R. P., O'Connor, P. J., & Werner, D. (2021). The explosion of artificial intelligence in antennas and propagation: How deep learning is advancing our state of the art. *IEEE Antennas and Propagation Magazine*, *63*, 16–27. https://doi.org/10.1109/MAP.2020.3021433

Chen, H., He, X., Teng, Q., Sheriff, R. E., Feng, J., & Xiong, S. (2020). Super-resolution of real-world rock microcomputed tomography images using cycle-consistent generative adversarial networks. *Physical Review E*, *101*, 023305. https://doi.org/10.1103/PhysRevE.101.023305

Chiles, J.-P., & Delfiner, P. (2012). *Geostatistics : Modeling spatial uncertainty.* John Wiley & Sons.

Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., & Choo, J. (2017). Stargan: Unified generative adversarial networks for multi-domain image-to-image translation.

Cosenza, P., Marmet, E., Rejiba, F., Cui, Y. J., Tabbagh, A., & Charlery, Y. (2006). Correlations between geotechnical and electrical data: A case study at garchy in france. *Journal of Applied Geophysics*, *60*, 165–178. https://doi.org/10.1016/j.jappgeo.2006.02.003

Damelin, S. B., & Hoang, N. S. (2018). On surface completion and image inpainting by biharmonic functions: Numerical aspects. *International Journal of Mathematics and Mathematical Sciences*, *2018*.

Das, B. M., & Sivakugan, N. (2015). *Introduction to geotechnical engineering*. Cengage Learning.

Das, B. M. (2010). *Principles of geotechnical engineering* (7th). Cengage Learning.

Dawoody, H. (2021). *Geotechnical risk allocation on mega design-build infrastructure projects*. California State Polytechnic University.

Demir, U., & Unal, G. (2018). Patch-based image inpainting with generative adversarial networks.

de Rienzo, F., Oreste, P., & Pelizza, S. (2008). Subsurface geological-geotechnical modelling to sustain underground civil planning. *Engineering Geology*, *96*, 187–204. https://doi.org/10.1016/j.enggeo.2007.11.002

DHPC. (2022). *Delftblue supercomputer (phase 1)*. https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1

Émile-Mâle, G. (1979). *The restorer's handbook of easel painting*. Van Nostrand Reinhold.

Foster, D. (2019). *Generative deep learning : Teaching machines to paint, write, compose, and play*. O'Reilly Media Inc.

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

Goodfellow, I. J. (2017). Nips 2016 tutorial: Generative adversarial networks. *CoRR*, *abs/1701.00160*.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial networks.

Google. (2023). *Google trends*. https://trends.google.com/trends/

Grunwald, S. (2009). Multi-criteria characterization of recent digital soil mapping and modeling approaches. *Geoderma*, *152*, 195–207. https://doi.org/10.1016/j.geoderma.2009.06.003

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved training of wasserstein gans.

Haining, R. (2001). *Spatial autocorrelation*. https://doi.org/10.1016/B0-08-043076-7/02511-0

Harris, E. C. (1989). *Principles of archaeological stratigraphy*. Elsevier. https://doi.org/10.1016/C2009-0-21688-6

Heuvelink, G. B., & Webster, R. (2022). Spatial statistics and soil mapping: A blossoming partnership under pressure. *Spatial Statistics*, *50*, 100639. https://doi.org/10.1016/j.spasta.2022.100639

Hutton, J. (2005). *Theory of earth* (Illustrated). Dodo Press.

Ijaz, Z., Zhao, C., Ijaz, N., ur Rehman, Z., & Ijaz, A. (2021). Spatial mapping of geotechnical soil properties at multiple depths in sialkot region, pakistan. *Environmental Earth Sciences*, *80*, 787. https://doi.org/10.1007/s12665-021-10084-z

Inpainting. (2023). https://scikit-image.org/docs/stable/auto_examples/filters/plot_inpaint.html

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5967–5976. https://doi.org/10.1109/CVPR.2017.632

Jaksa, M. (2000). Geotechnical risk and inadequate site investigations: A case study. *Australian Geomechanics*, *35*, 39–46.

Jam, J., Kendrick, C., Walker, K., Drouard, V., Hsu, J. G.-S., & Yap, M. H. (2021). A comprehensive review of past and present image inpainting methods. *Computer Vision and Image Understanding*, *203*, 103147. https://doi.org/10.1016/j.cviu.2020.103147

Janssens, N., Huysmans, M., & Swennen, R. (2020). Computed tomography 3d super-resolution with generative adversarial neural networks: Implications on unsaturated and two-phase fluid flow. *Materials*, *13*, 1397. https://doi.org/10.3390/ma13061397

Juang, C. H., Zhang, J., Shen, M., & Hu, J. (2019). Probabilistic methods for unified treatment of geotechnical and geological uncertainties in a geotechnical analysis. *Engineering Geology*, *249*, 148–161. https://doi.org/10.1016/j.enggeo.2018.12.010

Kang, B., & Choe, J. (2020). Uncertainty quantification of channel reservoirs assisted by cluster analysis and deep convolutional generative adversarial networks. *Journal of Petroleum Science and Engineering*, *187*, 106742. https://doi.org/10.1016/j.petrol.2019.106742

Karras, T., Laine, S., & Aila, T. (2018). A style-based generator architecture for generative adversarial networks.

Krige, D. G. (1951). *A statistical approach to some mine valuations and allied problems at the witwatersrand*. University of Witwatersrand.

Laloy, E., Hérault, R., Jacques, D., & Linde, N. (2017). Training-image based geostatistical inversion using a spatial generative adversarial neural network. https://doi.org/10.1002/2017WR022148

Ledoux, H., Arroyo-Ohori, K., Peters, R., & Pronk, M. (2022). *Computational modelling of terrains*. TU Delft.

Leunge, L. K. (2019). *Machine learning reveling insights into soil stratification. an application for dikes and dams*. Delft University of Technology. http://resolver.tudelft.nl/uuid:1b91c352-0544-4744-9023-4efbcfd8bdd7

Li, J., Madry, A., Peebles, J., & Schmidt, L. (2017). On the limitations of first-order approximation in gan dynamics.

Li, J., Cai, Y., Li, X., & Zhang, L. (2019). Simulating realistic geological stratigraphy using direction-dependent coupled markov chain model. *Computers and Geotechnics*, *115*, 103147. https://doi.org/10.1016/j.compgeo.2019.103147

Lucas, A., Iliadis, M., Molina, R., & Katsaggelos, A. K. (2018). Using deep neural networks for inverse problems in imaging: Beyond analytical methods. *IEEE Signal Processing Magazine*, *35*, 20–36. https://doi.org/10.1109/MSP.2017.2760358

Lucas, G. W. (2021). *An introduction to natural neighbor interpolation*. https://gwlucastrig.github.io/TinfourDocs/NaturalNeighborIntro/index.html

Mahapatra, D., & Bozorgtabar, B. (2019). Progressive generative adversarial networks for medical image super resolution.

McBratney, A., Santos, M. M., & Minasny, B. (2003). On digital soil mapping. *Geoderma*, *117*, 3–52. https://doi.org/10.1016/S0016-7061(03)00223-4

McKillup, S., & Dyar, M. D. (2010). *Geostatistics explained: An introductory guide for earth scientists.* Cambridge University Press.

Mescheder, L., Geiger, A., & Nowozin, S. (2018). Which training methods for gans do actually converge?

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *CoRR*, *abs/1411.1784*.

Mlynarek, Z., Wierzbicki, J., & Wolynski, W. (2005). Use of interpolation methods for geotechnical profiling. *Studia Geotechnica et Mechancia*, *27*, 5–13.

Mosser, L., Dubrule, O., & Blunt, M. J. (2017). Reconstruction of three-dimensional porous media using generative adversarial neural networks. *Physical Review E*, *96*, 043309. https://doi.org/10.1103/PhysRevE.96.043309

Müller, S., Schüler, L., Zech, A., & Heße, F. (2022). Gstools v1.3: A toolbox for geostatistical modelling in python. *Geoscientific Model Development*, *15*, 3161–3182. https://doi.org/10.5194/gmd-15-3161-2022

Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., García, Á. L., Heredia, I., Malík, P., & Hluchý, L. (2019). Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey. *Artificial Intelligence Review*, *52*, 77–124. https://doi.org/10.1007/s10462-018-09679-z

Nichols, G. (2023). *Sedimentology and stratigraphy*. John Wiley & Sons.

Oliveira, D. A. B., Ferreira, R. S., Silva, R., & Brazil, E. V. (2019). Improving seismic data resolution with deep generative networks. *IEEE Geoscience and Remote Sensing Letters*, *16*, 1929–1933. https://doi.org/10.1109/LGRS.2019.2913593

Oliver, M. A., & Webster, R. (2015). *Basic steps in geostatistics: The variogram and kriging*. Springer International Publishing. https://doi.org/10.1007/978-3-319-15865-5

Oluwatuyi, O. E., Ng, K., & Wulff, S. S. (2023). Improved resistance prediction and reliability for bridge pile foundation in shales through optimal site investigation plans. *Reliability Engineering & System Safety*, *239*, 109476. https://doi.org/10.1016/j.ress.2023.109476

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Phoon, K., Kulhawy, F. H., & Grigoriu, M. D. (1995). *Reliability-based design of foundations for transmission line structures. final report*. Electric Power Research Inst.

Phoon, K.-K., Ching, J., & Shuku, T. (2022). Challenges in data-driven site characterization. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, *16*, 114–126. https://doi.org/10.1080/17499518.2021.1896005

Phoon, K.-K., & Kulhawy, F. H. (1999). *Characterization of geotechnical variability*.

Phoon, K.-K., & Zhang, W. (2022). Future of machine learning in geotechnics. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, 1–16. https://doi.org/10.1080/17499518.2022.2087884

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks.

Rahman, M. H., Abu-Farsakh, M. Y., & Jafari, N. (2021). Generation and evaluation of synthetic cone penetration test (cpt) data using various spatial interpolation techniques. *Canadian Geotechnical Journal*, *58*, 224–237. https://doi.org/10.1139/cgj-2019-0745

Ravuri, S., Lenc, K., Willson, M., Kangin, D., Lam, R., Mirowski, P., Fitzsimons, M., Athanassiadou, M., Kashem, S., Madge, S., Prudden, R., Mandhane, A., Clark, A., Brock, A., Simonyan, K., Hadsell, R., Robinson, N., Clancy, E., Arribas, A., & Mohamed, S. (2021). Skilful precipitation nowcasting using deep generative models of radar. *Nature*, *597*, 672–677. https://doi.org/10.1038/s41586-021-03854-z

Reiffsteck, P., Benoît, J., Bourdeau, C., & Desanneaux, G. (2018). Enhancing geotechnical investigations using drilling parameters. *Journal of Geotechnical and Geoenvironmental Engineering*, *144*. https://doi.org/10.1061/(ASCE)GT.1943-5606.0001836

Robertson, P. K. (1990). Soil classification using the cone penetration test. *Canadian Geotechnical Journal*, *27*, 151–158. https://doi.org/10.1139/t90-014

Robertson, P. K. (2010). Soil behaviour type from the cpt: An update. *2nd International Symposium on Cone Penetration Testing*, 575–583.

Robertson, P. K., & Cabal, K. (2022). *Guide to cone penetration testing* (7th). Gregg Drilling LLC.

Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-net: Convolutional networks for biomedical image segmentation*. https://doi.org/10.1007/978-3-319-24574-4_28

Santurkar, S., Tsipras, D., Ilyas, A., & Madry, A. (2018). How does batch normalization help optimization?

Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access*, *7*, 53040–53065. https://doi.org/10.1109/ACCESS.2019.2912200

Srivastava, N., Hinton, G., Krizhevsky, A., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Steno, N. (1968). *The prodromus of nicolaus steno's dissertation* (J. G. Winter, Ed.). Hafner Publishing Company. https://doi.org/10.3998/mpub.9690501

Sun, C., Demyanov, V., & Arnold, D. (2023). Geological realism in fluvial facies modelling with gan under variable depositional conditions. *Computational Geosciences*. https://doi.org/10.1007/s10596-023-10190-w

Terzaghi, K. (1929). Effect of minor geologic details on the safety of dams. *American Institute of Mining and Metallurgical Engineers*, 31–44.

Tomczak, J. M. (2022). *Deep generative modeling*. Springer International Publishing. https://doi.org/10.1007/978-3-030-93158-2

Valsecchi, A., Damas, S., Tubilleja, C., & Arechalde, J. (2020). Stochastic reconstruction of 3d porous media from 2d images using generative adversarial networks. *Neurocomputing*, *399*, 227–236. https://doi.org/10.1016/j.neucom.2019.12.040

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*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Wang, Y., Shi, C., & Li, X. (2022). Machine learning of geological details from borehole logs for development of high-resolution subsurface geological cross-section and geotechnical analysis. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, *16*, 2–20. https://doi.org/10.1080/17499518.2021.1971254

Willgoose, G. (2018). *Soils: Physical weathering and soil particle fragmentation*. https://doi.org/10.1017/9781139029339.008

Wood, D. M. (2004). *Geotechnical modelling* (1st). CRC Press. https://doi.org/https://doi.org/10.1201/9781315273556

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2018). Generative image inpainting with contextual attention. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5505–5514. https://doi.org/10.1109/CVPR.2018.00577

Zhang, W., Li, H., Li, Y., Liu, H., Chen, Y., & Ding, X. (2021). Application of deep learning algorithms in geotechnical engineering: A short critical review. *Artificial Intelligence Review*, *54*, 5633–5673. https://doi.org/10.1007/s10462-021-09967-1

Zhang, X., Zhai, D., Li, T., Zhou, Y., & Lin, Y. (2023). Image inpainting based on deep learning: A review. *Information Fusion*, *90*, 74–94. https://doi.org/10.1016/j.inffus.2022.08.033

Zhang, Z. (2018). Improved adam optimizer for deep neural networks. *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, 1–2. https://doi.org/10.1109/IWQoS.2018.8624183

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks.

Zuada-Coelho, B., & Karaoulis, M. (2022). Data fusion of geotechnical and geophysical data for three-dimensional subsoil schematisations. *Advanced Engineering Informatics*, *53*, 101671. https://doi.org/10.1016/j.aei.2022.101671

A

Images used in the expert criteria survey

This appendix presents a collection of ten images used for the blind expert criteria survey. Each image within this section comprises two main components: a primary CPT data image positioned at the top, used as input, and five distinct schematizations labelled with letters A through E. These schematizations correspond to the following methodologies: SchemaGAN, NeaNe, IDW, Kriging, and NatNe. It is important to note that the order of presentation for these schematizations within each image is randomized. To facilitate understanding of the order in which these schematizations appear, refer to Table A.1, which provides the corresponding codes.

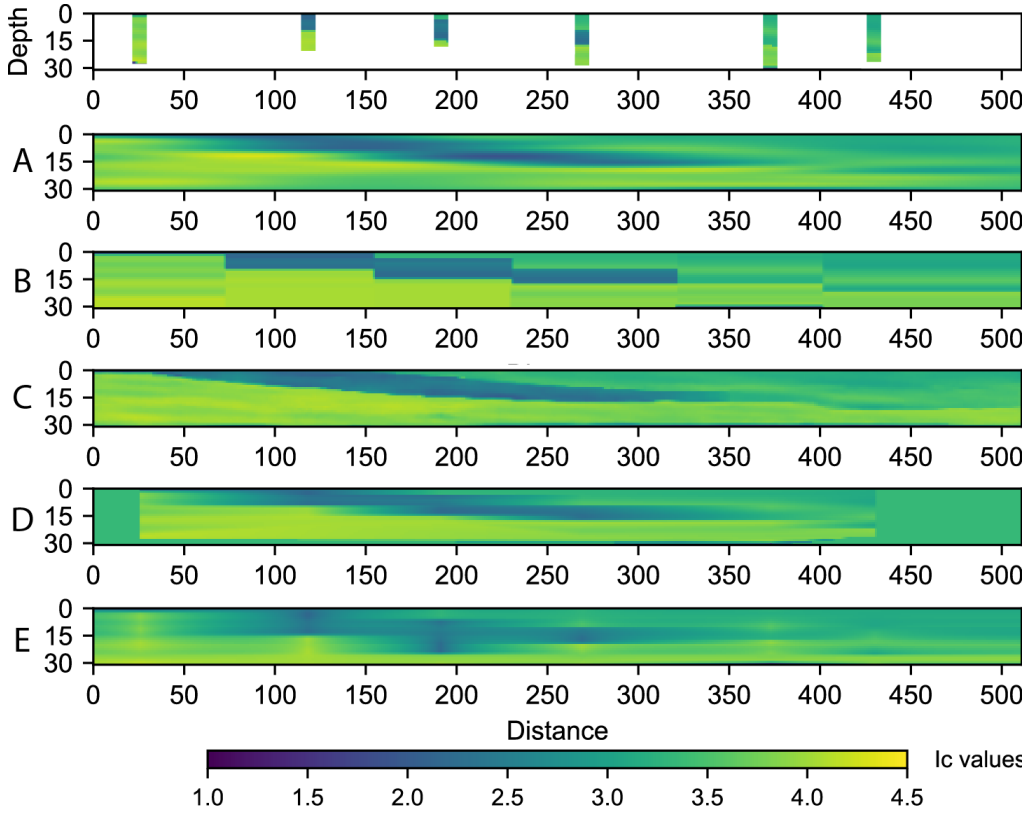


Figure A.1: Survey image number 1 for the expert criteria survey.

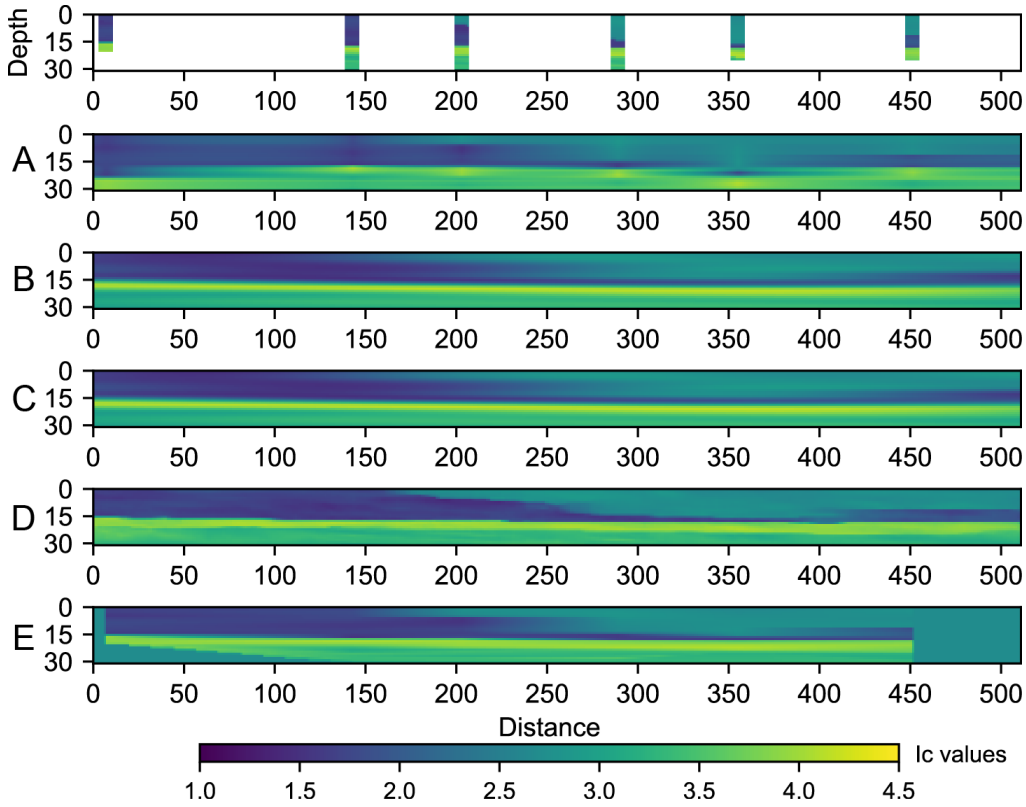


Figure A.2: Survey image number 2 for the expert criteria survey.

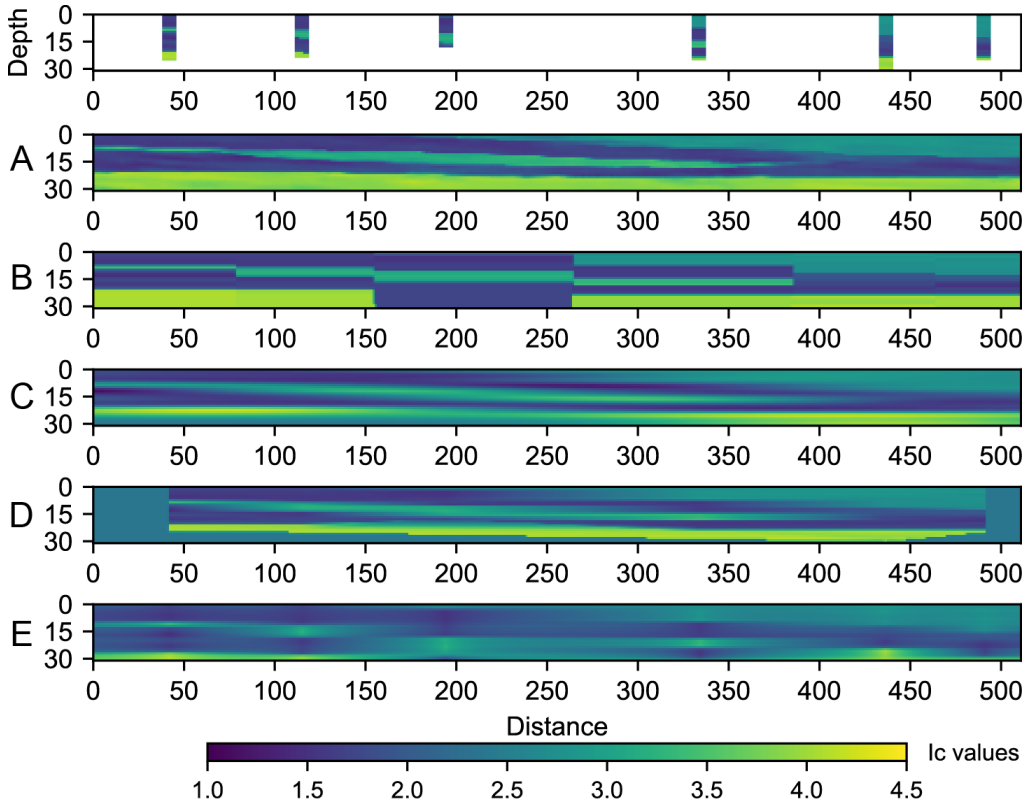


Figure A.3: Survey image number 3 for the expert criteria survey.

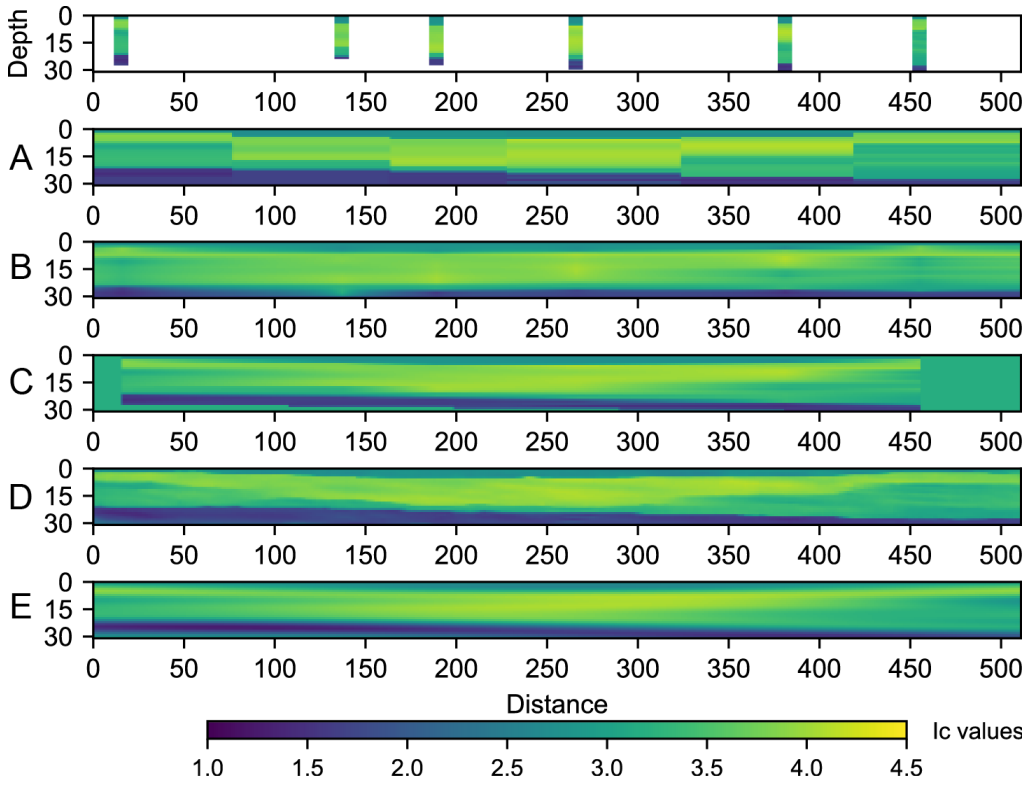


Figure A.4: Survey image number 4 for the expert criteria survey.

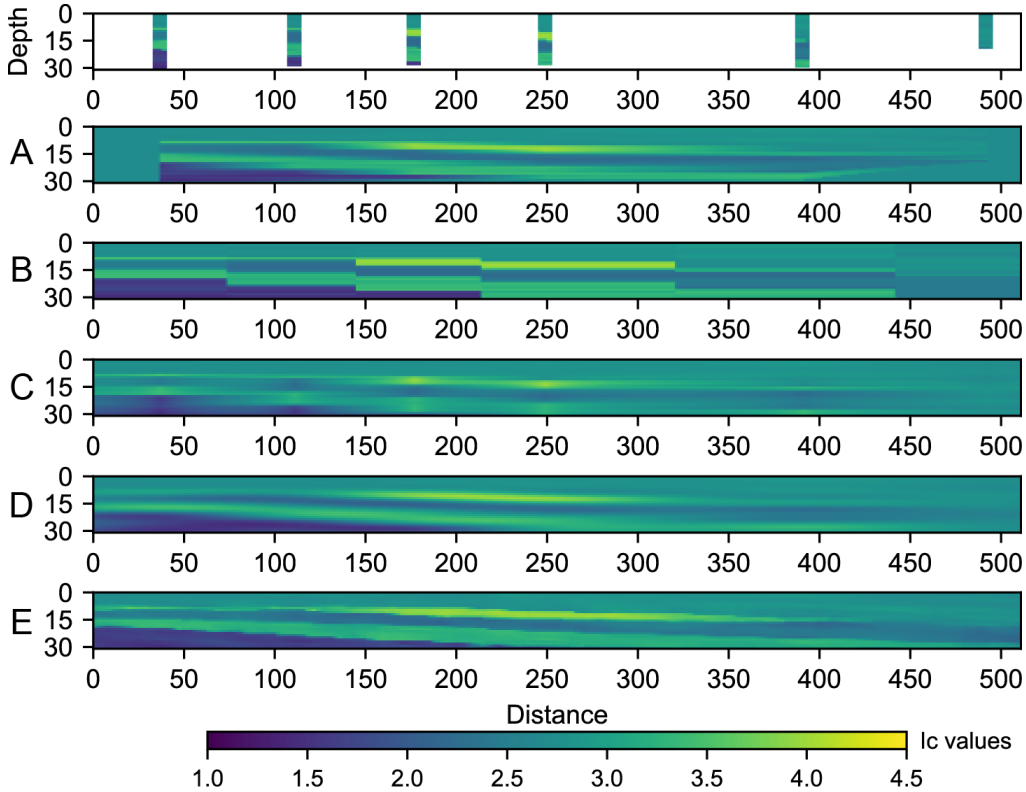


Figure A.5: Survey image number 5 for the expert criteria survey.

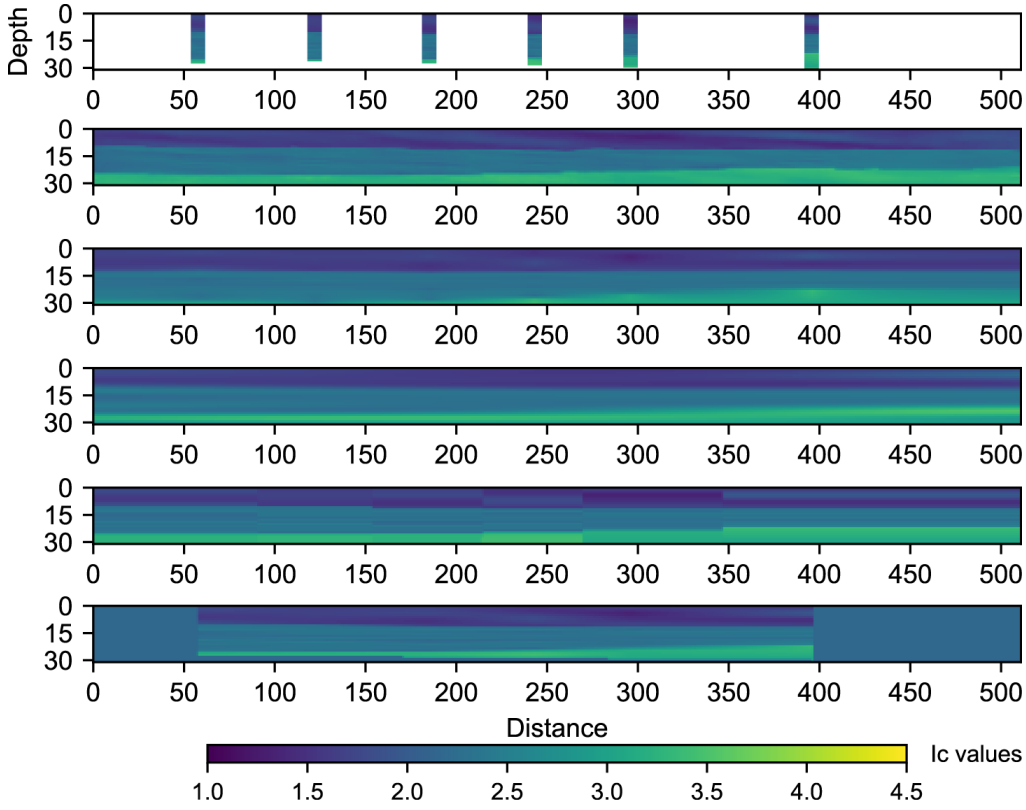


Figure A.6: Survey image number 6 for the expert criteria survey.

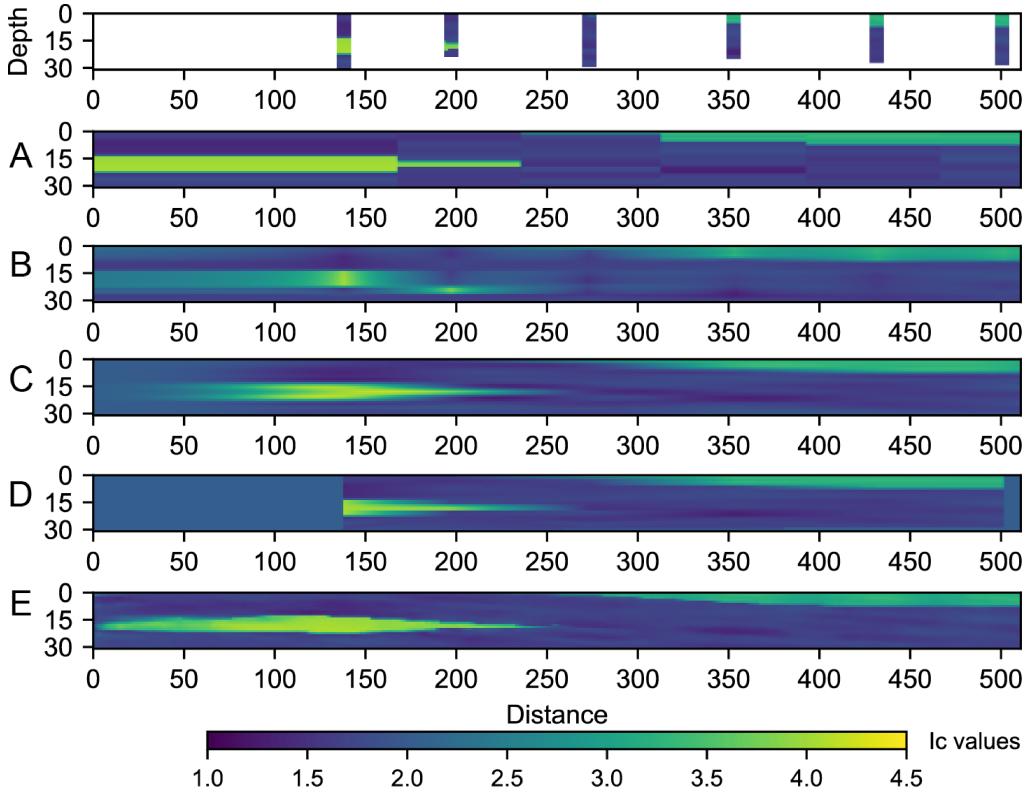


Figure A.7: Survey image number 7 for the expert criteria survey.

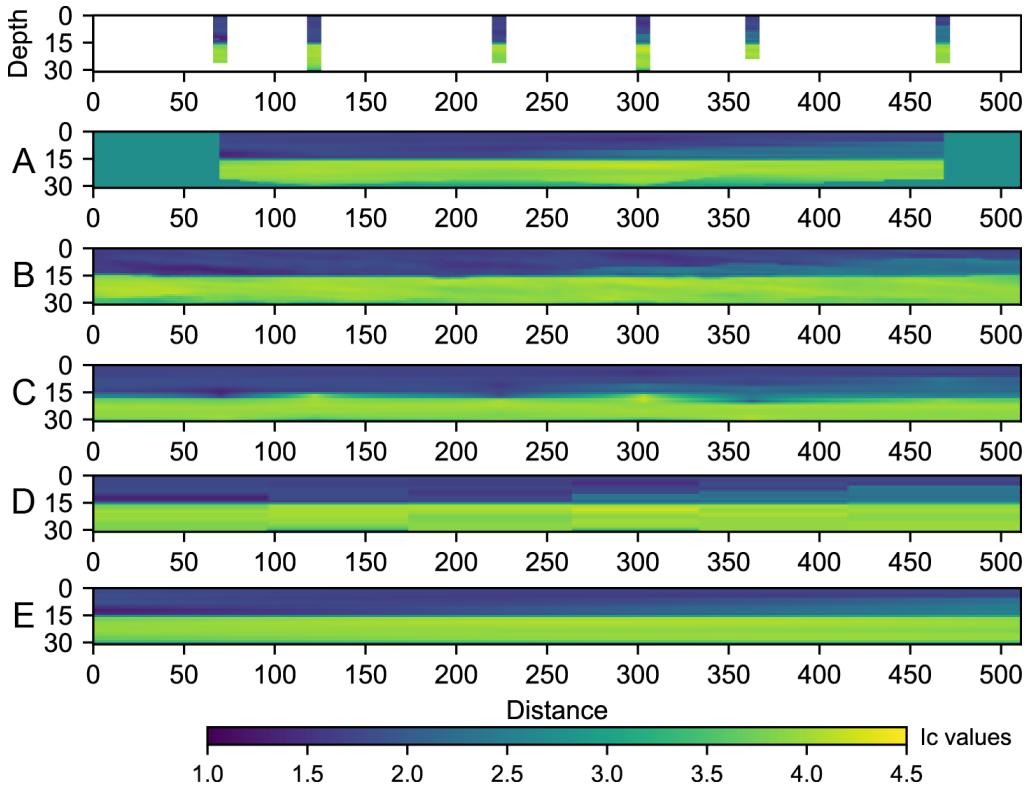


Figure A.8: Survey image number 8 for the expert criteria survey.

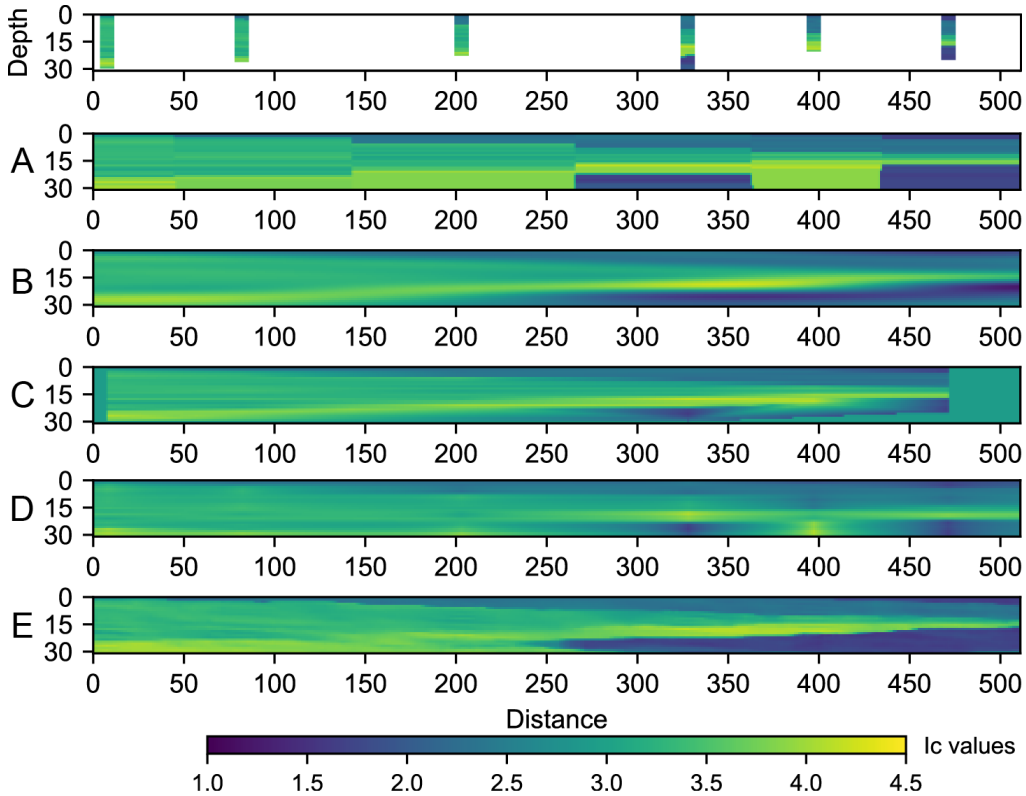


Figure A.9: Survey image number 9 for the expert criteria survey.

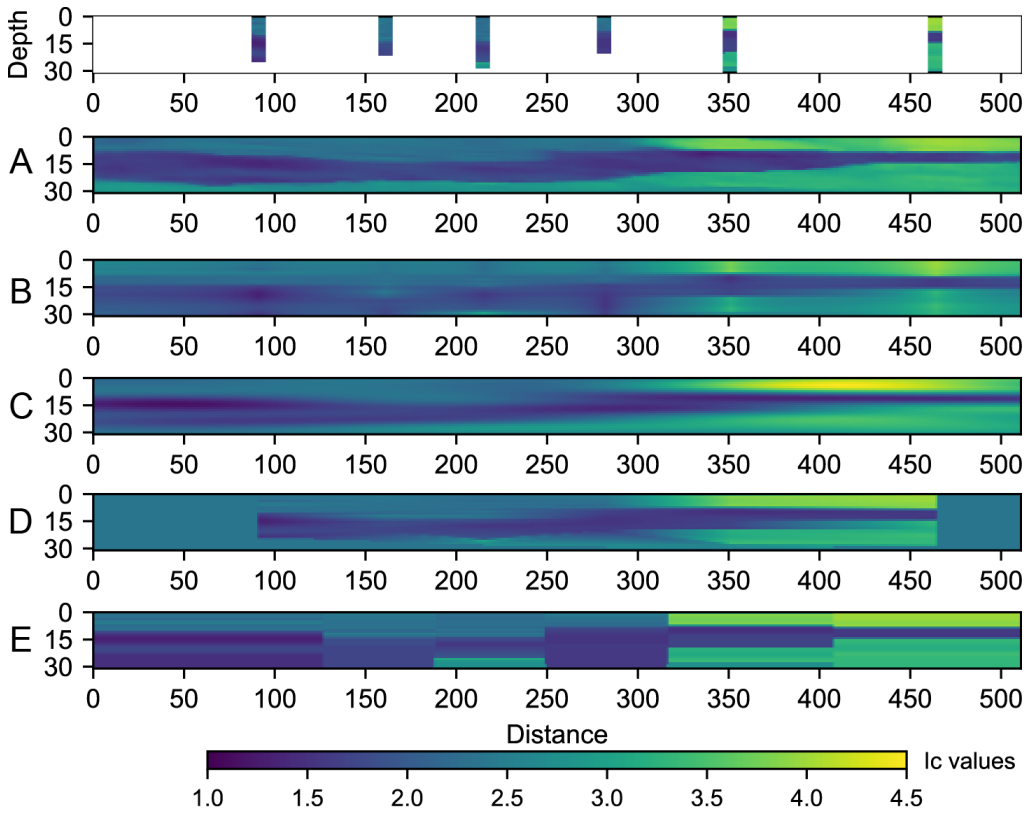


Figure A.10: Survey image number 10 for the expert criteria survey.

Table A.1: Code for the survey images

Survey no.	Figure	SchemaGAN	NeaNe	IDW	Kriging	NatNe
1	Figure A.1	C	B	E	A	D
2	Figure A.2	D	C	A	B	E
3	Figure A.3	A	B	E	C	D
4	Figure A.4	D	A	B	E	C
5	Figure A.5	E	B	C	D	A
6	Figure A.6	A	D	B	C	E
7	Figure A.7	E	A	B	C	D
8	Figure A.8	B	D	C	E	A
9	Figure A.9	E	A	D	B	C
10	Figure A.10	A	E	B	C	D