The state of the earth: estimating physical parameters from noisy and incomplete earth observation data

Arp, L.R.

THE STATE OF THE EARTH:
ESTIMATING PHYSICAL PARAMETERS FROM NOISY AND INCOMPLETE EARTH OBSERVATION DATA

Proefschrift

ter verkrijging van
de graad van doctor aan de Universiteit Leiden,
op gezag van rector magnificus prof.dr. S. de Rijcke,
volgens besluit van het college voor promoties
te verdedigen op dinsdag 23 juni 2026
klokke 11:30 uur

door

Laurens Ruben Arp

geboren te Haarlem
in 1995

Promotores:
Dr. M. Baratchi
Prof.dr. H.H. Hoos
Prof.dr. P.M. van Bodegom

Promotiecommissie:
Prof.dr. K.J. Batenburg
Prof.dr. M.M. Bonsangue
Dr. C. de Vries
Prof.dr. D. Borth — University of St.Gallen, Switzerland
Prof.dr. S. Dzeroski — Jožef Stefan Institute, Slovenia

This work is funded by the Dutch Research Council (NWO) under the research programme Open Competition ENW with project number OCENW.KLEIN.425, and by the European Space Agency (ESA) under the Open Space Innovation Platform (OSIP) research project "Physics-aware Automated Machine Learning (PA-AutoML) for Earth Observations".

ISBN 978-94-93539-31-0

An electronic version of this dissertation is available at
https://scholarlypublications.universiteitleiden.nl/.

SUMMARY

Estimating physical parameters from observations is a core theme in AI for science, with broad relevance across diverse scientific fields and real-world applications. These parameters, which define the state of physically governed systems, are often not directly observable: we can only infer them from their observed outcomes. In this thesis we focus on developing AI solutions for reliably estimating Earth system parameters from satellite data, which are critical for, e.g., environmental protection, disaster response, and agriculture. Two core challenges complicate this estimation: 1) spatio-temporal gaps in the available data hamper model training and consistency, and 2) multiple physical states of the parameters can produce the same observations, leading to multiple possible solutions for an observation. In this dissertation, we address four research questions, two of which are related to the first challenge and two others to the second.

The first research question addresses spatio-temporal gaps in ground truth datasets and how to fill them in effectively. In Chapter 3, we propose a novel spatial interpolation method to mitigate this issue. Ground truth data, representing the 'true' parameter values that we need to estimate, often covers only sparse measurement points, while we require a complete map of predictions at every point in an area. Inspired by Markov reward processes, our method iteratively propagates the available information through a system of unknown values, producing an interpolated grid of ground truth values. We demonstrate the strength of our proposed method on simulated data and real-world GDP and COVID-19 datasets.

The second research question addresses spatio-temporal gaps in satellite data, such as those caused by cloud cover. In Chapter 4, we propose a novel method that adapts our spatial interpolation approach from Chapter 3 for image reconstruction. Our method uses a fully observable, but outdated, reference image to guide the reconstruction process. We demonstrate its effectiveness on a popular cloud removal dataset and a diverse dataset that we collected, which we make publicly available for further scholarly work.

The third research question explores multiple solutions in parameter estimation. While using a common radiative transfer model (RTM) inversion approach, we unexpectedly found only a single solution, contradicting prior expectations. In Chapter 5, we present an empirical study on RTM-simulated data to investigate how problems with a single solution might still exhibit multiple solutions in practice. Our analysis shows that the problem meets all the criteria of well-posedness

(i.e., a unique solution exists), and experiments on real-world satellite data confirm that this property is retained in non-simulated settings. Further experiments revealed that observation noise is likely the primary cause of multiple apparent solutions. Even if a unique solution exists for a given observation, errors in the observation itself can lead to incorrect solutions.

The final research question addresses computing the set of solutions for parameter estimation problems. In Chapter 6, we formalise and theoretically support a framework to make this computation tractable, and we propose a method based on constrained black-box optimisation to automatically approximate the set of possible solutions. This solution set enables a wide array of novel analyses and applications. We validate both the theoretical framework and our practical approximation method through empirical experiments across physical, statistical, and machine learning models.

In conclusion, in this dissertation we address two core challenges in Earth system parameter estimation: spatio-temporal data gaps and multiple solutions for the same observations. Through four distinct contributions contained in Chapters 3—6, we advance the understanding of these challenges and improve the reliability of estimation methods. While the impact of these challenges likely cannot be eliminated altogether within the scope of a single dissertation, our work provides tangible advances and actionable pointers for future research. We encourage other scholarly work to build on these findings and contribute toward interdisciplinary efforts aimed at estimating the state of the Earth.

SAMENVATTING

Het inschatten van wetenschappelijke parameters op basis van observaties is een centraal thema in wetenschappelijke AI, met brede relevantie voor diverse wetenschappelijke vakgebieden en applicaties. Deze parameters beschrijven de status van een fysiek systeem, en zijn vaak niet direct te observeren: we kunnen ze alleen inschatten via geobserveerde uitkomsten. In dit proefschrift ligt de focus op het ontwikkelen van betrouwbare AI-oplossingen om aardwetenschappelijke parameters in te schatten met behulp van satellietdata. Deze parameters zijn essentiëel voor, onder andere, milieubescherming, rampenbestrijding en landbouw. Er zijn echter twee centrale uitdagingen die deze inschatting bemoeilijken: 1) in onze data ontbreken grote hoeveelheden spatiëel-temporele informatie, waardoor het trainen van modellen wordt bemoeilijkt en de consistentie van voorspellingen wordt verminderd, en 2) meerdere verschillende fysieke omstandigheden kunnen dezelfde observaties genereren, waardoor er meerdere goed passende oplossingen zijn voor een observatie. In dit proefschrift beantwoorden we vier onderzoeksvragen, waarbij er twee gewijd zijn aan de eerste uitdaging, en twee aan de tweede uitdaging.

De eerste onderzoeksvraag betreft ontbrekende informatie in spatiëel-temporele ground truth datasets, en hoe deze op een effectieve manier ingevuld kunnen worden. In Hoofdstuk 3 stellen we een nieuwe spatiële interpolatiemethode voor, om zo de invloed van dit probleem te verminderen. Ground truth data, die de 'echte' waardes bevat voor de parameters die we moeten inschatten, beslaat vaak slechts een groep specifieke meetpunten verspreid over een onderzoeksgebied, terwijl we meestal geïnteresseerd zijn in een volledige kaart met voorspellingen op alle punten in het gebied. Onze methode, geïnspireerd door Markov reward processes, propageert iteratief de beschikbare informatie door een systeem van onbekende waardes, met als eindresultaat een geïnterpoleerd raster met ground truth waardes. We tonen de effectiviteit van onze methode aan op zowel gesimuleerde data als echte BBP en COVID-19 datasets.

De tweede onderzoeksvraag betreft ontbrekende spatiëel-temporele informatie in satellietdata, bijvoorbeeld veroorzaakt door wolkendekking. In Hoofdstuk 4 stellen we een nieuwe methode voor die onze interpolatiemethode van Hoofdstuk 3 aanpast voor het reconstrueren van afbeeldingen. Onze nieuwe methode stuurt het interpolatiealgoritme via een volledig geobserveerde maar verouderde referentie-afbeelding, en vult hiermee de missende data in. We tonen de effectivi-

teit van deze methode aan op zowel een populaire dataset voor het verwijderen van wolkendekking, als op een dataset met hoge diversiteit die wij zelf hebben gecreëerd. Wij maken deze dataset publiek toegankelijk voor verder wetenschappelijk onderzoek.

De derde onderzoeksvraag betreft de meerdere passende oplossingen bij het inschatten van parameters. Bij het gebruik van een veel gebruikte inversietechniek voor radiative transfer models (RTMs) kwamen we onverwacht tot de ontdekking dat er slechts een enkele oplossing mogelijk was, in tegenstelling tot onze verwachtingen. In Hoofdstuk 5 voeren we een empirisch onderzoek uit op data gesimuleerd door een RTM, om te begrijpen hoe problemen met een enkele oplossing zich in de praktijk toch als meerdere passende oplossingen kunnen presenteren. Uit onze analyse komt naar buiten dat het probleem alle criteria van een well-posed probleem (namelijk dat er een unieke oplossing bestaat), en experimenten op echte satellietdata bevestigen dat deze eigenschap behouden blijft buiten een puur gesimuleerde context. Uit verdere experimenten blijkt dat ruis op de observaties waarschijnlijk de voornaamste oorzaak is van meerdere passende oplossingen. Zelfs als er een unieke oplossing bestaat voor een observatie, leiden fouten in de observatie zelf tot fouten in de oplossingen.

De laatste onderzoeksvraag betreft het berekenen van de set van passende oplossingen voor het inschatten van parameters. In Hoofdstuk 6 formaliseren we dit probleem, en geven we een theoretische onderbouwing voor een kader om deze berekening computationeel haalbaar te maken. We stellen ook een methode voor, gebaseerd op constrained black-box optimalisatie, om automatisch de set van mogelijke oplossingen te benaderen. We valideren zowel het theoretische kader als onze praktische benaderingsmethode via empirische experimenten met natuurkundige-, statistische- en machine learning-modellen.

Samenvattend: in dit proefschrift leveren wij een bijdrage aan twee centrale uitdagingen voor de inschatting van aardwetenschappelijke parameters: ontbrekende spatiëel-temporele informatie in de data, en meerdere passende oplossingen voor dezelfde observaties. Via vier specifieke contributies, beschreven in Hoofdstukken 3—6, hebben wij de kennis over deze uitdagingen verbreed, en de betrouwbaarheid van inschattingsmethodes verbeterd. Hoewel de invloed van deze uitdagingen niet geheel verwijderd kan worden binnen het kader van een enkel proefschrift, vertegenwoordigt ons onderzoekswerk zowel concrete vooruitgang als uitvoerbare aanbevelingen voor verder onderzoek. Wij moedigen andere wetenschappelijk onderzoekers aan om verder te gaan in onze onderzoeksrichting, en om bij te dragen aan verder multidisciplinair onderzoek, met als doel om verbeteringen te realiseren in het inschatten van de status van de Aarde.

TABLE OF CONTENTS

1

INTRODUCTION

Earth system parameters are all around us. These physical parameters are scientific variables, whose values describe the Earth at a certain time and place. For example, the Leiden office in which this thesis is being written on a spring afternoon could be described by the following parameters: a temperature of 22.4 degrees Celsius, an air humidity of about 50%, a leaf area index of 0 (sadly; this parameter describes the concentration of the leaves of vegetation) and a $PM10$ fine dust concentration of about 17 $\mu g/m^3$.

Of course, this set of parameters P can differ per use case. Whereas atmospheric scientists may be interested in parameters such as CO_2, NO_2, $PM10$ and $PM2.5$ concentrations in the atmosphere [1, 2], oceanographers may focus on sea surface salinity, pH and ocean wind speed [3] and environmental biologists may care more about chlorophyll $a+b$ content, soil moisture and leaf area index (LAI) [4]. The unifying factor differentiating these physical parameters from conventional variables is that they are measurable quantities representing the state of a physical system governed by physical laws. In our case, this physical system is the Earth system: a highly complex physical system consisting of multiple sub-systems including meteorological processes, ocean dynamics and ecosystems, with possible interactions between these sub-systems.

Although the relevance of these parameters to scientific applications is clear, one pertinent question would be how we can know the correct values of these parameters for our situation in the first place, like the parameter *configuration* $\boldsymbol{\theta} = [22.4, 0.5, 0.0, 17]$ described above for an office at Leiden University. In fact, many of the parameters we may be interested in cannot be directly measured. Others may be directly measurable, but doing so may be costly in terms of human

labour, financial investment, or damage to the very system we are trying to measure (for example, cutting down trees to measure their properties in a laboratory). As an alternative to this *in-situ* approach to data collection, we could instead opt to *estimate* or *infer* these parameters indirectly from data sources that are both measurable and plentiful. When working with Earth system parameters, arguably the most appealing of such data sources is spaceborne Earth observation data: information on the planet obtained by satellites orbiting the Earth, such as the Sentinel satellites from the European Space Agency (ESA) and the Landsat satellites by the American National Aeronautics and Space Administration (NASA).

Earth observation (EO) data can be obtained in various forms; we provide a brief overview hereof in Section 2.1. However, the main focus of our methods is on spaceborne EO data in the form of satellite data. EO satellites continuously orbit the Earth, and transmit their measurements back for further analysis. However, the quantities being measured are generally electromagnetic waves, measured in the form of optical light spectra denoting the intensity of light at the wavelengths of b spectral bands. Many of the physical parameters describing the Earth impact how sunlight gets reflected back into space. This allows us to use light spectra, measured by EO satellites, as feature data to estimate the values of physical parameters on Earth, such as the concentration of gases in the atmosphere or chlorophyll in plant matter. In some fields using EO data, this task is referred to as *parameter retrieval*; throughout this thesis, we will use the standard machine learning terminology of the *estimation* of the physical parameters as target variables, unless stated otherwise.

Definition 1.1 (physical parameter estimation)**.** The estimation of the values $\boldsymbol{\theta}$ of a set of physical parameters (target variables) P that describe the state of the Earth, based on an observed feature vector $\mathbf{x}$ of Earth observation data. The estimated values are denoted as $\hat{\boldsymbol{\theta}}$.

Parameter estimation, therefore, is an inference problem where the configuration of true values for P, $\boldsymbol{\theta}$, must be inferred from the features $\mathbf{x}$: $\hat{\boldsymbol{\theta}} = f(\mathbf{x})$. The mapping function $f : \mathcal{X} \rightarrow D_P$, where $\mathbf{x} \in \mathcal{X}$ (observation domain; in our case, these are light spectra with b bands in $\mathbb{R}^b$) and $\boldsymbol{\theta} \in D_P$ (domain for the parameters P), is generally unknown in advance. For example, in astronomy, we know how large celestial bodies might bend beams of light, because they are derived from well-studied fundamental causal properties of gravity. However, when merely observing a beam of light, it can be challenging to infer whether there was a celestial body that changed its trajectory, let alone which specific object, and the nature of its impact. As will become apparent in Section 1.1, there are challenges associated with parameter estimation using EO data that are specific to this problem setting.

Although we will maintain generality in most of our methods, we will have a particular focus on the estimation of *biophysical parameters* (ecological variables describing the state of an ecosystem) throughout this work as the main use case, specifically those related to vegetation. These parameters, such as leaf area index, chlorophyll content and water content, play a vital role in the health of vegetation and its role in the ecosystem, which, in turn, can strongly affect other Earth systems (e.g., the impact of vegetation and algae on climate through photosynthesis).

1.1. MOTIVATION

In addition to the typical challenges associated with inference problems, the specifics of the problem setting for parameter estimation result in the following two core technical challenges:

1. Many sources of data are inconsistent, resulting in spatial and temporal gaps in the dataset (missing data) and potential biases. For example, sensor networks can only measure a target variable at a few specific locations, and cloud cover can introduce substantial gaps in satellite imagery.

2. The inference problem could be ill-posed due to insufficiently informative or noisy input data, meaning that multiple solutions could be equally valid for a given input. For example, if two parameters affect light, and therefore the satellite imagery, in the same way (e.g., they both increase near-infrared reflectance), it is unclear which parameter the observed patterns should be attributed to.

These challenges, while differentiating parameter estimation using EO data from conventional inference- and machine learning tasks, such as image classification or house price prediction, are likely to be present in a class of similar inference problems, whose shared properties with EO-based parameter estimation result in the same challenges. Applications in such related problems may benefit from the contributions contained in this work.

Inference problems are likely to contain similar challenges to those presented in this work if they share (a subset of) the following properties: i) the input $\mathbf{x}$ consists of noisy, real-world data, ii) the ground truth data $\boldsymbol{\theta}$ is difficult to measure precisely, iii) there are (non-uniformly distributed) data gaps in the input $\mathbf{x}$ and/or the ground truth data $\boldsymbol{\theta}$, such as missing groups of pixels in images, and iv) the data generating mechanism in the real world consists of causal, physical relationships from the parameters $\boldsymbol{\theta}$ to the outcomes $\mathbf{x}$, but in the inference problem, $\mathbf{x}$ forms the observations while $\boldsymbol{\theta}$ are the target parameters (inverse modeling). Intuitively, fields such as astronomy [5, 6] and medical imaging [7, 8, 9] are likely to

share many of these characteristics with the Earth science settings considered in this work.

1.1.1. CHALLENGE 1: DATA INCONSISTENCY

Data inconsistency can affect both the ground truth parameter data $\boldsymbol{\theta}$ and the EO feature data $\mathbf{x}$.

The target variables P, whose values $\boldsymbol{\theta}$ we are aiming to estimate, are inherently spatio-temporal. A parameter describing the Earth does so for a specific location at a certain time. However, the datasets we have available to train or evaluate predictive models may not have measurements for the full region of interest. For example, a sensor network (see Section 2.1.1) takes measurements at various locations in a region, but the number of locations is necessarily limited. Similarly, sensor defects, transmission errors, inhospitable terrain and other unpredictable factors can prevent measurements from being taken for every possible location in a study area. This results in both systematic and random spatial gaps in the ground truth data $\boldsymbol{\theta}$. Similarly, not all measurements in a dataset may have been taken at the same time, or the temporal resolution of measurements could be low (e.g., due to the cost of measurement), resulting in temporal gaps in the data. These spatio-temporal data gaps can make it challenging to train and evaluate predictive models that, for example, return a full image where every pixel represents a prediction for the corresponding geographical location at the same time.

To improve data consistency and enable the training and evaluation of prediction models, we need a method to interpolate the spatio-temporal gaps in the data. The interpolated data should reflect local spatial structure, such as similar neighbours or abrupt changes, as well as global spatial structure, such that spatial interactions can take place over any distance. This led to research question 1 (RQ1), which aims at developing a method that satisfies these requirements:

RQ1 *How can we effectively interpolate spatial data such that both local and global spatial properties are retained?*

In contrast to the ground truth data described above, the EO data $\mathbf{x}$ is usually already a full satellite image covering a geographical area, while the regular orbit ensures consistent revisits (although certain orbits may prioritise frequent revisits over particular geographical areas). Nonetheless, this type of data is still strongly affected by spatio-temporal data gaps. Unlike the sensor networks described above, which often have a consistent pattern to the data gaps, missing EO data is generally caused by unpredictable factors. These factors include invalid pixels (e.g., due to solar glint), sensor faults (e.g., Landsat ETM+ SLC-off data [10]), transmission errors and, most prominently, cloud cover (for details, see Section

2.1.3). The spatio-temporal data gaps caused by these factors are often not uniformly distributed. For example, cloud cover can be affected by the season (more clouds in winter than in summer) and geographical location (more clouds in tropical regions than in deserts). Therefore, this data inconsistency forms a major obstacle for the training of predictive models, and can introduce biases in the trained models. Furthermore, without the feature data $\mathbf{x}$ available at a specific time and place of interest, direct parameter estimation for this spatio-temporal point will not be possible with a typical prediction model mapping features $\mathbf{x}$ to parameters $\boldsymbol{\theta}$.

To address these limitations, we need a method to interpolate missing data in our input EO data $\mathbf{x}$, thereby greatly improving the consistency of our estimations. We took a particular interest in cloud cover, as one of the most prominent causes of missing EO data, but the method should generalise to other types of missing data. Unlike general spatial interpolation problem settings, most satellite-based EO data comes in the form of images. This results in an image processing gap filling task, where the spatial relationships between pixels can have a far greater intensity and variability than in general spatial interpolation settings, while requiring greater precision. Moreover, this spatial structure may change over time, to which the method should be robust. It should be easy to apply the method to any type of EO image data, without the need to re-train a model for every possible type of satellite, and the resulting reconstructed images should be of high quality, such that they can be used in downstream parameter estimation tasks. This leads to the following research question:

RQ2 *How can we effectively and easily interpolate unpredictable, spatially clustered missing data in Earth observation imagery?*

1.1.2. CHALLENGE 2: NOISE AND ILL-POSEDNESS

When performing parameter estimation as $\hat{\boldsymbol{\theta}} = f(\mathbf{x})$, regardless of the method used to approximate f, we are implicitly assuming that the function f is a one-to-one mapping from $\mathcal{X}$ to D_P. Any uncertainty on $\hat{\boldsymbol{\theta}}$ would then be caused solely by flaws in the approximation of f (e.g., the errors of a machine learning model). However, it may be the case that there is not enough information contained in $\mathbf{x}$ to uniquely estimate the correct $\boldsymbol{\theta}$. As a result, f could be a one-to-many mapping, where many different $\hat{\boldsymbol{\theta}}$ could be the appropriate solution for the same $\mathbf{x}$. For example, if the observations were generated as $\mathbf{x} = \boldsymbol{\theta}^2$, and we observed $\mathbf{x} = 100$, there would be two possible answers: $\boldsymbol{\theta} = 10$ or $\boldsymbol{\theta} = -10$. This is one of the ways in which a parameter estimation problem can be *ill-posed* (see Section 2.2.2 for details).

Alternatively, noise on the observations $\mathbf{x}$ can result in ill-posedness. As explained in Section 2.1.3, the EO input data is necessarily noisy (for example, due

to atmospheric interference in the optical data). Consequently, the parameter estimation result $\hat{\boldsymbol{\theta}}$ we obtain as a best fit for this noisy data may not be the true $\boldsymbol{\theta}$ representing the state of the Earth. Instead, any configuration that could have been the parameter estimation result, if the random noise on the EO observations had been different, could possibly be the true solution. Without knowing the noise on the observations, it will not be possible to judge with certainty which is the true solution, making a commitment to any particular solution inappropriate and resulting in ill-posedness.

These examples illustrate how multiple factors could result in ill-posedness for parameter estimation. These factors may reduce the reliability of parameter estimation results, but their impact is not well understood, complicating the development of strategies to alleviate them. Therefore, a thorough analysis of the impact of these factors on the solution reliability is necessary. From this analysis, it should be clear what the factors are, and how they can be alleviated. This is the scope of the following research question:

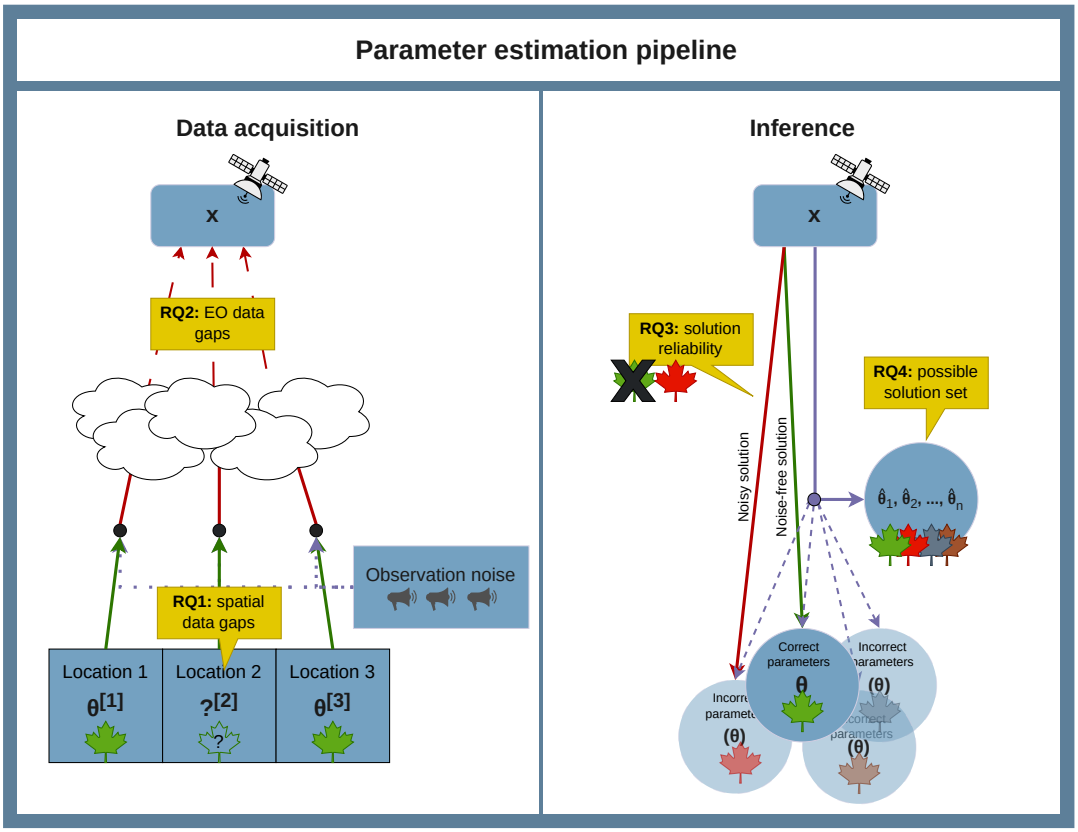
RQ3 *What makes parameter estimation an ill-posed problem, and which factors affect the reliability of parameter estimation results?*

When a problem is ill-posed, finding appropriate solutions can be extremely challenging, but this is not necessarily the case. Even problems known to be ill-posed could be solved at a satisfactory level of accuracy, depending on the problem setting. For example, if there are only a few possible solutions $\hat{\boldsymbol{\theta}}$, all at a negligible distance from each other in the parameter space D_P, the ill-posedness is unlikely to cause major issues for the parameter estimation task. However, if there are many solutions spread around D_P, each of which could be an appropriate solution for the observed EO data $\mathbf{x}$, performing a reliable estimation may be infeasible based on the information available.

Therefore, we need a method to automatically extract all the possible solutions to noisy inference problems. The set of solutions should be accurately approximated, and should include solutions that could have been the parameter estimation result if the noise had been different. This would then allow us to, for example, establish how severely ill-posed a parameter estimation problem instance is. Our final research question, therefore, is as follows:

RQ4 *How can we automatically extract the set of possible solutions to a noisy inference problem?*

An overview of the research questions, and the challenges they address, is shown in Figure 1.1.



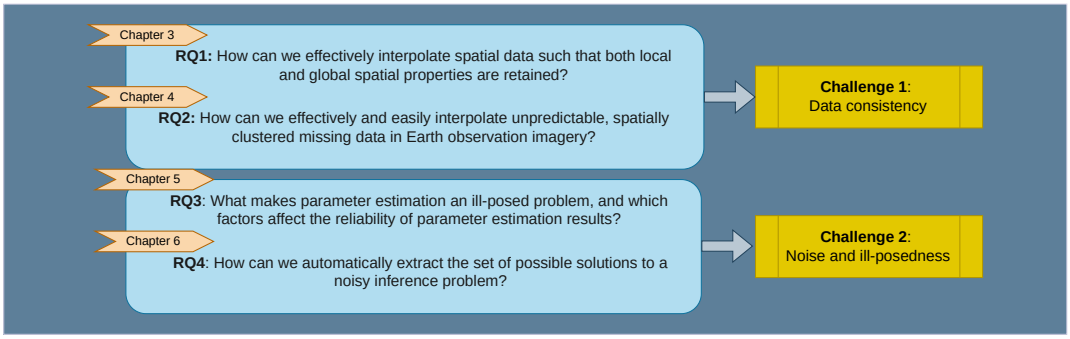


Figure 1.1: Overview of the research questions, the chapters of this work addressing them, which part of the parameter estimation pipeline they address, and the challenges they correspond to. In RQ1, the parameter values $\boldsymbol{\theta}$ are known for some locations (1 and 3), but are unknown for another (2), necessitating a spatial interpolation method for the target values. In RQ2, the satellite measurements of the light reflected by the parameters are blocked by factors such as cloud cover, necessitating a missing data interpolation method for satellite data. In RQ3, random noise causes the spectrum to change shape, resulting in the wrong parameter configuration being returned as a solution, and necessitating an investigation into noise and ill-posedness. In RQ4, there are many possible spectra that random noise could create, resulting in multiple potential solutions and necessitating a method that finds all these potential solutions.

1

1.2. CONTRIBUTIONS

The work covered in this thesis will present the following key contributions to the artificial intelligence and remote sensing communities:

- To address RQ1, we created a spatial interpolation method, called VPint, that provides a novel perspective on spatial interpolation problems and exploits covariates with minimal assumptions. The key idea of our proposed method is its system-oriented perspective, enabling global spatial interactions through a chain of local spatial relationships. We performed extensive empirical experiments on synthetic data, as well as real-world datasets on GDP levels and COVID-19 incidence rates, and compared the performance of VPint to that of representative baseline methods including Kriging methods, spatial autoregressive models, and convolutional neural networks. We found VPint to perform better than competing methods on average in terms of mean absolute error (MAE), root mean squared error (RMSE), peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). This contribution is contained in Chapter 3, and has given rise to the following journal article:

  > Laurens Arp, Mitra Baratchi, and Holger H. Hoos. (2022). *VPint: value propagation-based spatial interpolation.* Data Mining and Knowledge Discovery, 36:1647–1678. Springer.

- To address RQ2, we adapted the VPint algorithm to VPint2, which applies the concepts of VPint to the image processing-like problem of cloud removal in Earth Observation data. We also created a cloud removal benchmark dataset, SEN2-MSI-T, enabling a validation of our proposed method and supporting the development of future cloud removal methods. VPint2 can be easily applied without prior training and requires only a single reference image of the same sensor. The modifications and extensions we made allowed the algorithm to be successfully applied to cloud removal and EO data, which introduce particular complications distinct from general spatial interpolation problems, due to variable spatial autocorrelation structures that could be dynamic over time. We compared our proposed method to baseline methods including mosaicking, automated machine learning, neighbourhood similar pixel interpolation and previously published cloud removal deep learning models, and found it to perform significantly better in terms of mean absolute error (MAE), mean absolute percentage error (MAPE), structural similarity index (SSIM) and downstream task performance in 17 out of 20 conditions. This contribution is contained in Chapter 4, and has given rise to the following journal article:

> Laurens Arp, Holger H. Hoos, Peter van Bodegom, Alistair Francis, James Wheeler, Dean van Laar, and Mitra Baratchi. (2024). *Training-free thick cloud removal for Sentinel-2 imagery using value propagation interpolation.* ISPRS Journal of Photogrammetry and Remote Sensing, 216:168–184. Elsevier.

- To address RQ3, we carried out a thorough empirical study aimed at characterising the ill-posedness of vegetation parameter estimation through radiative transfer model (RTM) inversion (for details on RTM inversion, see Section 2.2.1). We systematically tested whether the formal properties of ill-posedness (see Section 2.2.2) are met when performing RTM inversion. Next, we performed experiments aimed at gaining insight into the relationship between noise on the observed EO data and ill-posedness. Finally, we tested the mechanisms through which common strategies to reduce ill-posedness are effective. Our empirical experiments found that RTM inversion met all the requirements of a well-posed problem, but that noise on the EO data resulted in ill-posedness for the parameter estimation problem. Based on this knowledge, we recommend future work to focus on data-centric contributions, such as improving the quality of the EO data. This contribution is contained in Chapter 5, and has given rise to the following journal article:

  > Laurens Arp, Peter van Bodegom, Holger H. Hoos, and Mitra Baratchi. (2026). *Characterising the Ill-posedness of PROSAIL Inversion for Biophysical Parameter Retrieval.* European Journal of Remote Sensing, 59(1). Taylor and Francis.

- To address RQ4, we formalise the concept of ϵ-manifolds, and propose eMMI, a method to efficiently approximate the ϵ-manifold. Based on our findings on ill-posedness, we found that noise on the observations of an inference problem can cause the inference problem to become ill-posed. An ϵ-manifold is a set of all possible solutions to an inference problem, whose suitability is characterised by a loss function value threshold ϵ. Therefore, approximating the ϵ-manifold enables us to automatically extract the set of all possible solutions to a noisy inference problem. Our proposed approximation method, eMMI, efficiently performs a search around the point prediction based on diversity optimisation, and trains a classifier on the sampled points. We validated our proposed method on simulation models including RTMs, simulation-based inference simulators and machine learning tasks, and compared against statistical methods such as Gaussian processes, Bayesian neural networks and approximate Bayesian computation. Our empirical experiments found that ϵ-manifolds are significantly better than statistical uncertainty quan-

tification at containing the true solution, and eMMI approximated the ϵ-manifold significantly better than the baseline methods. This contribution is contained in Chapter 6, and has given rise to the following journal article:

> Laurens Arp, Peter van Bodegom, Nguyen Dang, Holger H. Hoos, Alistair Francis, and Mitra Baratchi. (2025). *Inference from Noisy Observations through Model Inversion: Constructing ϵ-Manifolds of Potentially Valid Solutions.* Under review.

All contributions in this thesis are accompanied by the code required to use the methods developed, as well as the code required to reproduce our results. Other resources, such as datasets created for the projects, can either be downloaded directly, or can be reproduced using the provided scripts.

1.3. ORGANISATION OF THIS DISSERTATION

The rest of the dissertation is structured as follows. In Chapter 2, we provide the necessary background information to allow readers to put our work in context. The following four chapters (Chapters 3 – 6) cover our key contributions, following the structure of the research questions (RQs) and contributions described in Sections 1.1 and 1.2. We conclude the dissertation in Chapter 7 with a discussion on the work contained therein, and recommendations for future research directions. Since the interdisciplinary nature of this work virtually guarantees that readers will be unfamiliar with parts of the subject matter, we provide a glossary of key terms at the end of this dissertation.

2

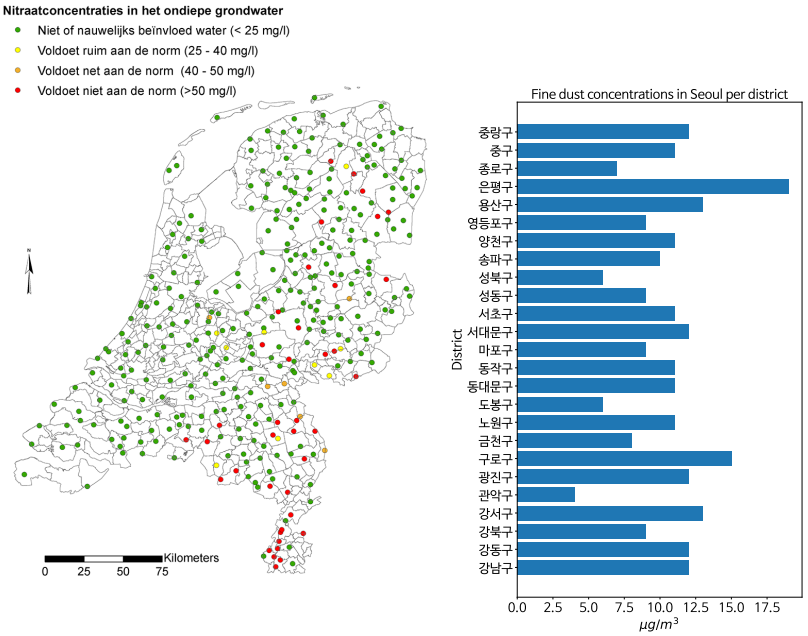
BACKGROUND

The work contained in this dissertation is highly inter-disciplinary, combining concepts from remote sensing, Earth science, physics, environmental biology, and artificial intelligence (AI). The intersection of these fields entails specific challenges that may be unfamiliar to an audience of experts specialised in one of these fields. The information contained in this chapter is intended to improve readability for experts from these different fields, enabling them to more easily follow the content from other fields in the main chapters of the dissertation.

First, Section 2.1 contains information about Earth observation data, which is the main type of input data for parameter estimation tasks. Second, Section 2.2 elaborates on radiative transfer models (RTMs) and the motivation for using them for parameter estimation. Finally, Section 2.3 covers the basics of the main computational methods that can be applied to parameter estimation from Earth observation data.

2.1. EARTH OBSERVATION DATA

In this section, we will cover the background of Earth Observation (EO) data. Section 2.1.1 will cover the main types of EO data available and motivate our goal of leveraging satellite data for physical parameter estimation. Next, we provide some general information on optical EO data, which we will be focusing on throughout this dissertation, in Section 2.1.2, and highlight some of the challenges arising in this context.



(a) Netherlands ground water

(b) Seoul fine dust

Figure 2.1: Examples of sensor network-based in-situ datasets. Figure 2.1a shows the ground water quality in the Netherlands as assessed by the Dutch National Institute for Public Health and the Environment [15] (image credits RIVM). Figure 2.1b shows the concentrations of fine dust (PM10 particles) in the atmosphere at the measuring stations of various districts in Seoul, South Korea, measured by AIRKOREA [12].

2.1.1. TYPES OF EO DATA

Earth observation data is a collective term for data that has been obtained from sensors measuring some properties of the Earth. For example, the Dutch RIVM, a government organisation responsible for national health and environment, operates a sensor network of 350 measuring stations, spread across the country, to monitor the quality of ground water in the Netherlands [11]. Other examples of this type of EO data includes the atmospheric fine particulate matter (fine dust) measuring stations in South Korea [12], the seismic activity and earthquake monitoring network in Japan [13] and various ecological measurements undertaken by NEON in the United States [14].

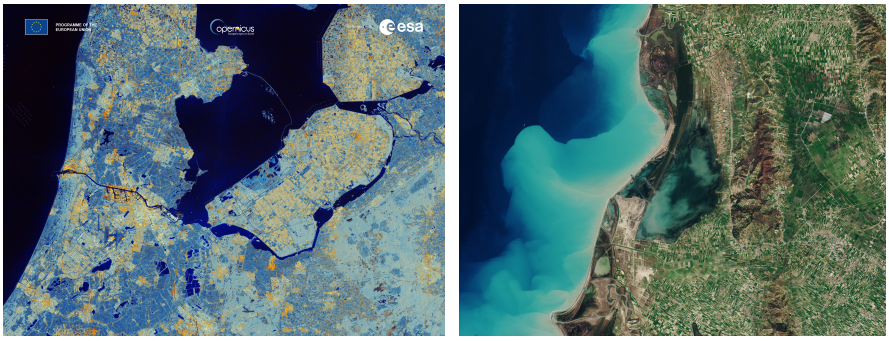
These sensor networks are examples of *in-situ* EO data: they enable direct measurements of the quantities we are interested in. We show two examples of this

type of data, namely the aforementioned ground water and fine dust examples, in Figure 2.1. The data from these sensor networks is typically represented as spatio-temporal point data, where every point is a station where sensors directly measure some quantity. In the ground water example, every measuring station in the sensor network measures water quality at a specific time and location. As a result, this type of data generally needs to be interpolated to enable users to derive estimations of a variable (such as ground water quality) at an arbitrary location; our work in Chapter 3 considers this type of data in more detail and will present a method to effectively perform this interpolation task.

In contrast, the largest part of EO data comes in the form of remote sensing data. In remote sensing, sensors are deployed remotely, which then observe the object of study from a distance. The main sources of remotely sensed EO data are aerial imagery, obtained by airborne sensors mounted on aircraft, and satellite imagery, obtained by spaceborne sensors mounted on satellites orbiting the Earth. The advantage of remotely sensed EO data over in-situ data is that it can be collected at a larger scale and obtain measurements for the full spatial area it is observing (a line determined by an orbit or flight path, with a width that is referred to as its swath). The resulting geo-referenced image can be directly used without requiring interpolation, and remotely sensed data is usually less expensive to set up and maintain at scale than in-situ data.

On the other hand, this type of data (especially aerial data) can still be expensive, is dependent on a sufficient spatial- and temporal resolution to produce meaningful results, and can typically only measure proxy variables such as the intensity of reflected light at different wavelengths (see Section 2.1.2) instead of the actual quantities of interest. This necessitates further processing to be converted into usable *data products*, thereby potentially introducing additional inaccuracies into the process. Most remotely sensed EO data is partially processed prior to data distribution, converting the raw data streams obtained by the sensors into a geo-referenced image with a predetermined map projection (usually *WGS84*), informative metadata and quality flags per pixel. Such images usually contain multiple data bands, where every band is an additional dimension containing an image for a different variable. For example, in optical imagery, the data often contains multiple bands measuring the intensity of reflected light at different spectral wavelengths, and synthetic aperture radar data contains multiple bands for different scene polarisations.

Once in orbit, spaceborne EO sensors can obtain continuous observations without major further interventions from the ground level, providing a large volume of raw data to its operators. For example, the Sentinel satellite platforms alone, operated by the European Space Agency (ESA) for the European Commission's Copernicus programme, transmitted 45 pebibytes (about $5.1 \cdot 10^{16}$ bytes) worth of EO data



(a) Sentinel-1 (b) Sentinel-2

Figure 2.2: Examples of satellite imagery from the Sentinel EO satellites by the European Space Agency (ESA). Figure 2.2a shows an example of Sentinel-1 synthetic aperture radar (SAR) data over the Netherlands (image credits ESA). Figure 2.2b shows an example of Sentinel-2 optical data over the Karavasta Lagoon in Albania (image credits ESA).

over the year 2023, with over 40 000 data products published every day [16]. This large scale makes satellite data an attractive type of EO data to build applications on: if reliable algorithms are created to estimate physical parameters from satellite data, this will provide us with a constant stream of information on the state of the Earth at any given time. Our focus will, accordingly, be on the development of algorithms applicable to spaceborne EO data.

On the other hand, satellite data also comes with significant drawbacks. Compared to aerial data, the increased distance between the sensor and the observation target results in lower spatial resolutions and an increased susceptibility to interference from, e.g., atmospheric conditions. Moreover, a substantial part of the collected data is unusable due to cloud cover (see Section 2.1.3, as well as Chapter 4 for our proposed method to remove clouds from satellite imagery). The validation of satellite-derived estimations can also be challenging, due to their large scale and low resolution, which usually cannot be directly compared to in-situ data. Instead, satellite-derived data products are often validated using derivations from airborne data, which are, in turn, validated using either in-situ data or further intermediate levels of abstraction, such as ground-based spectrometer data.

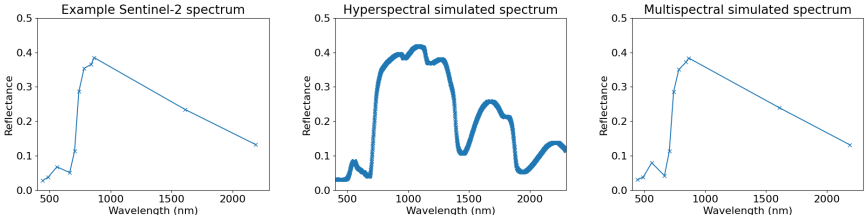
Although this hierarchical validation approach has resulted in the best evaluation of data products currently possible, it entails that the inaccuracies and error rates of lower-level EO data (such as in-situ data) inevitably trickle up to higher-level data (such as satellite data), with an additional information loss at every conversion between two levels of EO data. This can pose challenges for the application

of conventional machine learning and deep learning methods to EO data, because some of the central concepts, such as ground-truth data and reliable metrics to optimise for, are not necessarily available for all types of applications. On the other hand, when building applications that are not reliant on large amounts of accurate ground truth data, certain deep learning techniques can be highly effective, such as transfer learning [17], representation learning [18, 19], self-supervised and semi-supervised methods [20, 21] and foundation models [22, 23, 24]. Furthermore, the models can be trained and evaluated more reliably in problem settings where labels can be reliably generated by human annotators (e.g., land cover classification [25, 26] and segmentation tasks [27, 28, 29]).

2.1.2. OPTICAL EO DATA

The large majority of remotely sensed EO data comes in the form of optical data (shown in Figure 2.2b); a discussion of other forms of remote sensing data, such as radar, lidar and synthetic aperture radar (SAR) (shown in Figure 2.2a), is beyond the scope of this work. The sensors used for optical data are referred to as spectrometers, which measure the intensity of light at certain wavelengths of the electromagnetic spectrum. In addition to the red ($\sim 665nm$), green ($\sim 560nm$) and blue ($\sim 490nm$) wavelengths that are visible to the human eye, spectrometers can measure light intensity for ultraviolet ($\leq 400nm$) and (near-)infrared ($\geq 780nm$) light. This light energy in invisible wavelengths can be highly informative to various applications; for example, red-edge and near-infrared (relatively low wavelength infrared) light is known to be heavily affected by vegetation and photosynthesis [30, 31], while ultraviolet light has applications in, e.g., aerosol detection [32]. A typical light spectrum would not contain the same intensity at every part of the spectrum; for example, light intensity at visible wavelengths tends to be much lower than near-infrared wavelengths; this can be observed in all the spectra visualised in Figure 2.3.

Spaceborne optical sensors measure sunlight reflected by the Earth at a set of pre-determined wavelengths, called spectral bands. Most spaceborne spectrometers produce multispectral data, which contains multiple spectral bands spread around the spectrum. Although there are no strict rules, optical data containing tens of spectral bands is generally referred to as multispectral data. Historically, the NASA Landsat and MODIS satellites have provided multispectral data; over the last decade or so, the ESA Sentinel-2 satellites have gained much traction as a source of optical data. Most of the work in this dissertation will focus on applications based on Sentinel-2 data, since this is the most popular satellite data at the time of writing. An example spectrum measured by Sentinel-2 for a field to the south of Leiden, the Netherlands, on a spring day, can be found in Figure 2.3a.



(a) Sentinel-2 (b) PROSAIL hyperpsectral (c) PROSAIL multispectral

Figure 2.3: Visualisation of example spectra: a) a real-world spectrum measured by the Sentinel-2 multispectral satellite at a field to the south of Leiden, the Netherlands; b) a simulated hyperspectral spectrum simulated by the PROSAIL RTM (see Section 2.2), using the best fitting parameters to the Sentinel-2 observations of Figure 2.3a; c) the same simulation using PROSAIL as in Figure 2.3b, but after converting the hyperspectral output to a multispectral format matching the spectral bands of Sentinel-2 observations.

In addition to multispectral data, there are also sensors producing hyperspectral data. The best-fitting hyperspectral simulation for the Sentinel-2 observations in Figure 2.3a, determined using RTM inversion as described in Section 2.2, can be found in Figure 2.3b, and its corresponding multispectral version can be found in Figure 2.3c. Hyperspectral data is similar to multispectral data, but contains many spectral bands (high tens, hundreds or thousands). The increased spectral resolution may improve parameter estimation approaches by increasing the information content of the data, as illustrated in Figure 2.3 by the extra details in the hyperspectral image in Figure 2.3b compared to the multispectral images in Figures 2.3a and 2.3c. This could potentially reduce ill-posedness in parameter estimation settings; however, the improvements to spectral resolution may come at an expense of lower sensor accuracy, or sacrificing other types of resolution (spatial, temporal). The increased multi-collinearity of the band values may also require additional processing.

Since there are few hyperspectral satellite platforms currently in operation at the time of writing, and the hyperspectral satellite data that is available is often commercial (e.g., the Tanager satellites by Planet Labs or the GHOst satellites by Orbital Sidekick) or otherwise not publicly available for all locations and times (e.g., the PRISMA satellite by the Italian Space Agency), we will focus our methods on multispectral data. Throughout this work, we will specifically focus on Level-2A Sentinel-2 data products, which are optical images that have been atmospherically corrected.

2.1.3. EO DATA CHALLENGES: QUALITY, QUANTITY, AND DIVERSITY

Given the large volumes of EO data available, combined with the scale of EO imagery and the spatial and temporal relationships that are often contained in the data, large deep learning models form an appealing option for many EO tasks, including parameter estimation. However, deep learning models usually require large amounts of labelled data to train. While the EO feature data $\mathbf{x}$ may be plentiful, collecting ground truth data for the physical parameters $\boldsymbol{\theta}$ can be expensive, challenging, or even impossible for parameter estimation in particular, resulting in small dataset sizes for this type of problem. For example, when creating a ground truth dataset for leaf area index (LAI), data collection missions involve sending a team of researchers to an area of interest, who then systematically take measurements at a regular spatial grid that may cover thousands of square metres. Even if such a full grid is sampled, the study area is only a fraction of the total area covered by a single Sentinel-2 data product. Additionally, some parameters may involve extensive lab work (e.g., to test for certain chemicals) which can be destructive to the environment, or may not be directly measurable (for example, even at ground level, LAI is often measured using spectrometers).

The geographical and seasonal diversity of the available data can also be a key factor reducing the effectiveness of parameter estimation models. A low data diversity would not be representative of all possible inputs $\mathbf{x} \in \mathcal{X}$, thereby forcing models to extrapolate beyond training data (a task known to be challenging for deep learning models in particular [33]). For example, in the context of global vegetation monitoring, given the great diversity of species, ecosystems, climates, land cover types, lighting conditions and more, creating a representative dataset of all these conditions would be highly challenging. Additionally, the feasibility and cost of collecting ground truth data may only be acceptable to certain parts of the world (e.g., wealthy countries). This can bias models to only perform well on applications matching the training data, thereby mainly benefiting countries that are already wealthy. The use of physics-aware approaches, such as physics-informed neural networks [33, 34] and simulation-based hybrid models [35, 36] may improve the generalisability of prediction models.

Even if there is a large, representative ground truth dataset available, the reliability of this data must also be considered. Unlike typical machine learning problems with well-defined benchmark datasets containing reliable, human-defined ground truth labels (e.g., MNIST, CIFAR10 [37]), the ground truth data in parameter estimation concerns physical quantities that must be measured, such as the temperature, humidity, LAI and $PM10$ concentrations in the example of Chapter 1. For many parameters, the accuracy of these measurements is limited, and they frequently involve a tradeoff between accuracy and scale (see Section 2.1.1). This type of data is typically not directly derived from true measurements, but rather based

on estimations derived from a measured light spectrum. The measurements (that may be at a microscopic scale) must then be aggregated over a $10 \times 10 m^2$ grid cell (assuming a reasonably high spatial resolution), whose conditions can greatly vary within the spatial coverage of the cell.

Finally, the EO data $\mathbf{x}$ passed to the model will always be noisy, due to limitations of the sensors themselves, atmospheric interference, and spatial aggregation artefacts (e.g., spectral mixing [38]).

CLOUD COVER AND OPTICAL DATA

One of the most important drawbacks of optical data compared to, e.g., synthetic aperture radar (SAR) data, is that the signal measured by the spectrometer can be blocked by cloud cover. When the spectrometer encounters clouds, it becomes impossible to measure light at the ground level, while observing ground-level processes is generally the objective of optical data. At any time, 55% to 72% of the Earth is covered by clouds on average [39], with oceans accounting for the higher end of this range. Moreover, this cloud cover can be affected by spatial- and temporal autocorrelation, exacerbating the issue. For example, tropical regions will experience large amounts of cloud cover at any time of year, while temperate regions may experience long stretches of constant cloudy conditions during winter, followed by long stretches of clear conditions in summer. As a result, cloud cover can be a large obstacle to the use of satellite-based optical data for parameter estimation. In some cases, it can take months before a cloud-free observation can be made, resulting in large temporal gaps in the data. Removing clouds to fill these data gaps is an active area of research (see, e.g., [40, 41, 42]), and existing cloud removal methods are usually limited by a combination of poor scalability, limited reliability of ground truth data and poor transferability between different types of EO data.

Due to these limitations, cloud cover is one of the key challenges for parameter estimation using EO data. Therefore, we propose a novel cloud removal method in Chapter 4, through which we aim to increase the amount of usable optical data, thereby improving parameter estimation. Our approach aims to overcome some of the limitations of existing methods through computational efficiency, and by requiring no model training, thereby avoiding ground truth quality and transferability limitations.

2.2. RADIATIVE TRANSFER MODELS

Given the challenges and limitations of optical EO data described in Section 2.1.3, a conventional application of machine learning models is not always feasible when estimating parameters using Earth observation data: ground truth data may not

be available, may not be accurate enough to train high-quality estimators on, or may not be representative of all conditions in which the model would be deployed. However, the Earth system is studied by numerous scientific disciplines, many of which contain a wealth of scientific domain knowledge on the physical processes affecting this system. This domain knowledge can be used to alleviate some of the drawbacks of purely data-driven approaches, by simulating synthetic data. Since we are interested in parameter estimation from Earth observation data, we would need to use a domain knowledge-based (physical) model to simulate spectral data, corresponding to light spectra as measured by optical Earth observation satellites, for a specified parameter configuration. This type of simulation model is known as a radiative transfer model:

Definition 2.1 (radiative transfer model)**.** A radiative transfer model (RTM) is a simulation model $M : D_P \rightarrow \mathcal{X}$ whose parameterisation $\boldsymbol{\theta} \in D_P$ represents the state of physical parameters on Earth. It simulates a light spectrum $\boldsymbol{x} \in \mathcal{X}$ (where $\mathcal{X}$ is usually a space in $\mathbb{R}^d$ containing d spectral bands) that could be produced by the specified conditions.

RTMs are based on well-studied physical laws and domain knowledge. They model how a beam of light is affected as it is absorbed or reflected by the media it encounters, such as particles and gas concentrations in the atmosphere [1, 2], ocean water and microorganism production [3], or vegetation canopies and leaves [4, 43, 44]. An RTM takes a physical parameter configuration $\boldsymbol{\theta}$, such as the example configuration described in Section 1, as input, and uses this to simulate what a hypothetical beam of light would have looked like under these conditions. These domain knowledge-driven models play a pivotal role in parameter estimation applications where purely data-driven approaches cannot be easily applied.

The RTM we will focus on in Chapters 5 and 6 is PROSAIL, which combines the PROSPECT leaf model [43] and the 4SAIL canopy model [44], and is widely used in state-of-the-art vegetation parameter estimation methods [35, 45, 46]. However, many of our findings, especially those in Chapter 6, are likely to generalise to other RTMs.

2.2.1. RTM INVERSION

An RTM simulates a light spectrum $\mathbf{x} = M(\boldsymbol{\theta})$ based on the input parameters $\boldsymbol{\theta}$ it received. However, the data observed by optical satellites (see Section 2.1.2) already contains the light spectrum $\mathbf{x}$. Instead, the unknown physical parameters are the targets needing estimation, while these form the input parameters $\boldsymbol{\theta}$ of the RTM. Therefore, RTMs must be *inverted* (model inversion) in order to use them for parameter estimation through EO data.

Definition 2.2 (RTM inversion)**.** Given an RTM $M : D_P \rightarrow \mathcal{X}$, where $\mathbf{x} = M(\boldsymbol{\theta})$, RTM inversion refers to computing the inverse function $M^{-1} : \mathcal{X} \rightarrow D_P$ of M such that $\boldsymbol{\theta} = M^{-1}(\mathbf{x})$.

When inverting RTMs for parameter estimation, an analytical inversion is generally not possible to directly formulate M^{-1}. This is because the complex internal structure of the RTM M, often containing highly non-linear relationships and partial- or ordinary differential equations, is ill-suited to the derivation of an inverse function M^{-1} of M. Instead, the RTM inversion problem can be interpreted as a black-box numerical optimisation problem (for details, see Section 2.3.1). Here the task is to find a parameter configuration $\hat{\boldsymbol{\theta}}$ for which, when comparing the associated simulated spectrum $M(\hat{\boldsymbol{\theta}})$ and the observed spectrum $\mathbf{x}$, the difference between these spectra should be minimal:

$$\hat{\boldsymbol{\theta}} \in \underset{\boldsymbol{\theta} \in \mathcal{D}_P}{\operatorname{argmin}} \mathcal{L}\big(M(\boldsymbol{\theta}), \mathbf{x}\big) \tag{2.1}$$

Here $\mathcal{L}$ is a loss function measuring the goodness-of-fit between the observed spectrum $\mathbf{x}$ and the simulated spectrum $M(\boldsymbol{\theta})$ for a given parameterisation $\boldsymbol{\theta}$. This traditional approach to RTM inversion has been applied successfully to parameter estimation using PROSAIL and EO data [47, 48], although some authors note that its primary purpose is to validate the RTMs themselves [49]. Any of the black-box optimisation methods described later, in Section 2.3.1, could be used to find $\hat{\boldsymbol{\theta}}$ by optimising the loss function $\mathcal{L}$. In Figure 2.3, the hyperspectral (Figure 2.3b) and multispectral (Figure 2.3c) spectra both contain a PROSAIL simulation using the configuration $\hat{\boldsymbol{\theta}}$ as identified through Equation 2.1.

In recent years, much of the research focus for RTM inversion has been on so-called *hybrid models* [50, 51, 52, 45, 53, 54, 55, 46]. Hybrid models, sometimes referred to as *inverted simulators* or *inverted emulators*, combine knowledge-driven RTM simulations with data-driven machine learning models performing the inversion [35]. First, the RTM is used to generate a look-up table (LUT) of input parameters $\boldsymbol{\theta}$ and the simulated light spectra $\mathbf{x}$; this LUT can optionally be combined with a tabular dataset containing real-world data [56]. A machine learning model can then be trained on this dataset, taking the spectral data $\mathbf{x}$ as input features to predict the parameters $\boldsymbol{\theta}$ used by the RTM to generate that spectrum.

This hybrid modeling approach amortises the main computational cost of RTM inversion to a machine learning model training procedure. Therefore, parameter estimations can be performed much more efficiently in this manner compared to an approach requiring a new, high-dimensional optimisation procedure for every new problem instance, which may be highly relevant for, e.g., global mapping applications. Recent work on hybrid models often focuses on different sampling

strategies and heuristics through active learning [51, 45], enabling efficient and effective training in parts of the space where, for example, uncertainty is the highest. Although hybrid models can be an effective tool for efficiently performing the inversion of RTMs, they, like other parameter estimation methods, are limited by the ill-posedness of the problem. We explore this further in Chapters 5 and 6.

2.2.2. ILL-POSEDNESS

The inversion of RTMs is generally considered an ill-posed problem [57, 58, 59]. Ill-posed problems are problems that do not meet the requirements of well-posedness; the following must hold for an arbitrary problem to be considered well-posed [60]:

1. **The problem has a valid solution.** In the context of parameter estimation, this means that there exists a configuration of target parameters $\boldsymbol{\theta}$ that explains the observed light spectrum $\mathbf{x}$: $\exists \boldsymbol{\theta} : M(\boldsymbol{\theta}) \approx \mathbf{x}$.

2. **The solution to the problem is unique.** There should be only one configuration $\boldsymbol{\theta}$ that explains the observed light spectrum $\mathbf{x}$: $|\operatorname{argmin}_{\boldsymbol{\theta} \in D_P} \mathcal{L}\big(M(\boldsymbol{\theta}), \mathbf{x}\big)| = 1$.

3. **The solution moves continuously with regard to the inputs.** In parameter estimation, when visualising a point moving through the space $\mathcal{X}$ of possible input spectra $\mathbf{x}$, every movement in this space should correspond to a smooth movement of the solution $\hat{\boldsymbol{\theta}}$ in the parameter space D_P, with no sudden jumps to other parts of the space or other discontinuities. If f is the function mapping $\mathbf{x}$ to $\hat{\boldsymbol{\theta}}$, this function should be continuous: $\forall \mathbf{x}' \in \mathcal{X} : f(\mathbf{x}') \in D_P \wedge \lim_{\mathbf{x}' \to \mathbf{x}} f(\mathbf{x}') \in D_P \wedge \lim_{\mathbf{x}' \to \mathbf{x}} f(\mathbf{x}') = f(\mathbf{x})$.

Although RTM inversion is generally considered an ill-posed problem, due to a violation of requirement 2 (unique solution), this ill-posedness is not yet well understood, and, to our knowledge, there had been no structured, formal analysis of the phenomenon. Therefore, we aimed to fill this knowledge gap in Chapter 5 by systematically evaluating the ill-posedness of PROSAIL inversion, i.e., the inversion of an RTM that is widely used for vegetation parameter estimation applications.

2.3. ESTIMATION METHODOLOGIES

Within the scope of this work, two main strategies are considered for performing parameter estimation: black-box optimisation and machine learning. As a result, many of the chapters in this thesis assume some knowledge of these techniques, which form the backbone of our methodological contributions, but do not provide much background on these methods for readers who may not be familiar with

them. In this section, we will explain the relevant methods in more detail, such that the rest of the chapters can be more easily understood.

2.3.1. BLACK-BOX OPTIMISATION

Optimisation problems are pervasive in many different fields, including logistics [61], operations research [62] and industrial design [63]. Mathematically, suppose we are interested in a set of variables $P_1, P_2, ..., P_d$, whose domain is denoted as D_P (in continuous settings, this would be the d-dimensional space of real numbers $\mathbb{R}^d$). When assigned a specific value, these variables form a *configuration* $\boldsymbol{\theta}$:

Definition 2.3 (configuration)**.** A d-dimensional vector $\boldsymbol{\theta} \in D_P$ containing concrete value assignments $\theta_1, \theta_2, ..., \theta_d$ for the variables $P_1, P_2, ..., P_d$, representing the current state of a physical system in parameter estimation.

We also have access to an *objective function* $g(\boldsymbol{\theta})$:

Definition 2.4 (objective function)**.** A function $g : D_P \to \mathbb{R}$ mapping an input configuration $\boldsymbol{\theta}$ to a scalar $g(\boldsymbol{\theta})$, indicating the quality of the configuration (typically indicated by the distance between $M(\boldsymbol{\theta})$ and $\mathbf{x}$ in parameter estimation).

A practitioner may be interested in finding the *optimum* for g; that is, a configuration $\boldsymbol{\theta}^*$ for which the value of g is either maximised (for example, the best environmental conditions to make crops grow as fast as possible) or minimised (for example, the best water management approaches to ensure the risk of forest fires is as low as possible). In the context of physical parameter estimation, this optimum $\boldsymbol{\theta}^*$ would become the prediction $\hat{\boldsymbol{\theta}}$. Assuming the objective function should be minimised, the goal of optimisation is to find:

$$\boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta} \in D_P}{\operatorname{argmin}}\, g(\boldsymbol{\theta}) \tag{2.2}$$

In ideal cases, the optimum $\boldsymbol{\theta}^*$ can be computed analytically; for example, by solving for $\boldsymbol{\theta}$ after setting the derivative $g'(\boldsymbol{\theta}) = 0$. Unfortunately, in most practical optimisation problems, it is not possible to analytically compute $\boldsymbol{\theta}^*$, because i) the objective function g may be unknown entirely (black-box optimisation), ii) the objective function may be known, but not differentiable (for example, if g involves complex simulations), or iii) there may be an unknown number of *local* optima $\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, ..., \boldsymbol{\theta}_n^*$ (as opposed to a single *global* optimum $\boldsymbol{\theta}^*$) where $g'(\boldsymbol{\theta}_1^*) = g'(\boldsymbol{\theta}_2^*) = ... = g'(\boldsymbol{\theta}_n^*) = 0$.

Black-box optimisation refers to general-purpose optimisation methods where the objective function g is unknown. In this case, the optimisation task requires a *search* over the *search space* D_P, where only the output of the objective function

g, but not the function itself, can be used to guide the search. Examples of such methods include stochastic local search methods [64], evolutionary algorithms [65], metaheuristic algorithms such as particle swarm optimisation [66] and ant colony optimisation [67], as well as surrogate model-based Bayesian optimisation [68]. Black-box optimisation often involves a tradeoff between *exploration* (covering as much of the search space as possible) and *exploitation* (quickly reaching a local optimum for promising regions in the search space).

A typical black-box optimisation setting is limited by its reliance on the objective function g. Although there are methods that are more robust to noisy objective function evaluations [69], such methods cannot make a difference in cases where the signal from the objective function itself is unreliable. For example, the choice of objective function could be inappropriate for the problem instance, or may be a loosely correlated proxy function to an unknown true objective function, which may converge to a solution that is incorrect for the problem. Therefore, even if the global optimum $\boldsymbol{\theta}^*$ can be found reliably, this may not always mean that the identified optimum is also the true solution. If the objective function g is not fully reliable, there may be other points in the search space that consistently evaluate to a worse objective function value, but are actually the true solution. We explore and address this problem in detail in Chapter 5, where black-box optimisation is used extensively in the experiments to characterise the loss landscape of RTM inversion, and Chapter 6, where black-box optimisation is an important component in our proposed method to for approximating the set of potential solutions to inference problems (including RTM inversion).

In addition, Chapters 3, 4 and 6 all rely on black-box optimisation techniques for *automated algorithm configuration*; in this problem setting, the hyperparameters of the methods used (such as machine learning models, whose automated configuration is also known as AutoML) are automatically tuned using optimisation approaches based on their performance on validation data.

2.3.2. SUPERVISED MACHINE LEARNING

Much of machine learning (ML) consists of supervised ML, which refers to a wide range of predictive models whose model parameters can be tuned (trained) via data. Popular examples of traditional supervised machine learning models include linear regression, support vector machines (SVM) and Gaussian processes [70]. The training of supervised machine learning models is an optimisation problem, like those described in Section 2.3.1, where the objective function g to be minimised consists of a *loss function* $\mathcal{L}$, such as the mean squared error (MSE), mean absolute error (MAE), or accuracy. The loss function measures the predictive performance of the machine learning model by comparing the predictions made by

the model (under some parameterisation) to the ground truth values they should approximate (thus 'supervising' the model). Unlike black-box optimisation, the optimisation procedure for training a machine learning model can usually be performed efficiently via gradient-based methods.

ML models can be used to estimate parameters from the EO data $\mathbf{x}$, referred to as *features*, to predict $\boldsymbol{\theta}$. At the same time, the training of ML models is itself an inference problem, where the parameterisation $\boldsymbol{\theta}$ of a machine learning model must be inferred from the observed training data.

In the context of EO, deep learning models (a type of ML model using large neural networks containing many artificial neurons) are often preferred, due to their ability to take advantage of the large volumes of (largely unlabelled) training data, as well as the scale of their predictions (e.g., full images). Given their suitability for image data and modelling local spatio-temporal patterns, convolutional neural networks (CNNs) are particularly popular in EO settings, and have been applied successfully to various problems, such as land cover classification [71, 72], crop classification [73], semantic segmentation [74] and super-resolution [75]. More recently, many proposed models incorporate some form of attention mechanism [76, 77, 78], and transformer architectures have seen a rise in popularity, particularly in zero-shot settings [79, 80], where a model, trained on a problem with a certain set of classes, needs to make accurate predictions for a problem with a different set of classes, without any additional training for this new task. Deep neural networks are a natural choice for *data fusion* approaches, where information from multiple sensors (e.g., optical data from Sentinel-2 and SAR data from Sentinel-1) can be combined automatically in a latent representation [40, 41, 81].

Unlike the examples above, parameter estimation is a more difficult inference problem to perform using conventional machine learning and deep learning approaches (see Section 2.1.3 for details). Although there are deep learning-based data products [82], and prediction models [83, 84] available to perform the estimation for, e.g., LAI, it is difficult to train and evaluate such models for global applications when the ground truth data used for this may be insufficient in terms of quantity and/or quality (see Section 2.1.3). Because the training and validation data would have similar drawbacks, inaccuracies and biases in the model would be difficult to diagnose with the available data. This may, in part, explain why RTM inversion-based hybrid approaches (see Section 2.2.1) remain popular, often using traditional machine learning methods such as Gaussian processes.

In summary, deep learning-based approaches can be highly effective at a large number of typical Earth observation problems, can exploit the large volumes of input data available, and, once trained, scale well to rapidly process large image datasets and produce predictions for, e.g., all the pixels in such images. On the other hand, these approaches run into similar problems as traditional methods,

such as ill-posedness (see Section 2.2.2), while diagnosing those problems may be more difficult for deep learning models compared to traditional approaches. Our work in Chapter 4 may help improve the input data consistency for datasets to be used by deep learning methods, while our work in Chapter 6 can shed light on the nature of the parameter estimation problem, to help diagnose ill-posedness that deep learning models would also be affected by.

3

VPint: value propagation-based spatial interpolation

In this chapter[1], we start addressing the research questions from Chapter 1.1. Specifically, this chapter will cover RQ1: *How can we effectively interpolate spatial data such that both local and global spatial properties are retained?* Answering this research question allows us to address part of Challenge 1: much Earth observation data, especially from in-situ data sources, is collected at a limited number of measuring stations, while a full grid of parameter values is desired. The method proposed in this chapter addresses this problem by introducing a novel spatial interpolation algorithm called VPint.

3.1. Introduction

Under perfect lab conditions, a data scientist can train models, infer variables of interest and discover new knowledge from neatly organised, consistent and complete datasets. However, in real-world scenarios, one is rarely so lucky. Whether it is random measurement noise, inconsistent annotation, missing data or another problem, real-world data can be messy, and tricky to process in such a way that downstream models and processes can use it effectively.

In this chapter, we aim to address the problem of missing data in the specific case of spatial gridded data by proposing a computational method for spatial in-

[1] The contents of this chapter are based on the journal article: Laurens Arp, Mitra Baratchi, and Holger H. Hoos. (2022). *VPint: value propagation-based spatial interpolation.* Data Mining and Knowledge Discovery, 36:1647–1678. Springer. https://doi.org/10.1007/s10618-022-00843-2

terpolation. Prominent examples of such missing data in real-world scenarios, especially in Earth observation settings, can be found in Chapter 2.1, and include satellite imagery [85], the mapping of ecological field measurements and samples collected at a limited set of locations [86], and local precipitation forecasting from meteorological measuring stations covering a limited set of locations [87]. As such, spatial interpolation is a problem highly relevant to many fields, and a large body of literature is dedicated to it in statistical domains [88, 89, 90]. Data in spatial settings is particularly susceptible to missing values, due to, among other reasons, (i) limited and/or variable spatial and temporal resolutions, (ii) limited availability of measuring locations, (iii) measurements being acquired at different times and different locations, and (iv) the characteristics of the locations in question (e.g., cloud cover or inaccessible areas). As a simplified example, consider the task of mapping the temperature at a certain time throughout the Himalayas. Since resources are limited and parts of the terrain are inaccessible, it is infeasible to collect measurements at every $100m^2$. This gives rise to the problem of filling in the entire grid based on measurements from a limited number of locations. In this case, we could also use additional information on the elevation of the terrain to help inform our decisions – a location with a higher elevation than a reference value will likely have a lower temperature, and vice versa.

Spatial interpolation methods, such as Kriging (also known as Gaussian processes) [89, 91], tend to be founded on an assumption of *autocorrelation*, meaning that values are more strongly correlated with one another the closer their spatial proximity is. Our method is no exception in this regard. However, existing methods can be categorised into *local* methods and *distance-based* methods. Local methods, such as spatial autoregressive models [92, 93] or convolutional neural networks [94, 95], rely on adding the information of a strictly defined local neighbourhood around a target cell to enhance their predictions. The downside of these methods is that potentially valuable information outside the predefined neighbourhood is disregarded. Moreover, if local information is not available, local methods may require imputation methods to perform their estimations. Distance-based methods, on the other hand, most notably including various Gaussian process-based approaches [89, 91], can use any measurement available, but rely on a distance-based weighting to use this information for their predictions. The downside of these methods is that, in most spatial settings, paths cannot be assumed to be homogeneous, and thus distance alone may not be sufficient to reliably predict values. For example, in the case of temperature measurements in the Himalayas, the difference in elevation between pairs of locations may vary despite the distance being the same. This problem is further exacerbated by the two-dimensionality of spatial problems, allowing for the existence of multiple paths between any two locations, some of which may be more important than others for the propagation

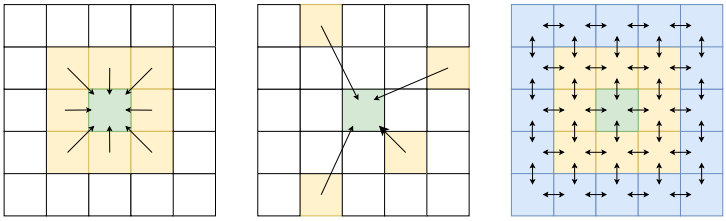


Figure 3.1: Local (left), distance-based (middle) and system-oriented (right) perspectives. In local and distance-based perspectives, the predicted value of the green cell is determined by the yellow cells (equal weights if local, unequal weights if distance-based). In the system-oriented perspective (used by our proposed method), the green cell is predicted using the yellow neighbours, which were in turn affected by their own neighbours (blue, yellow and green cells).

of values (for example, a longer path around a mountain as opposed to a shorter path over it).

In this chapter, we propose a method that incorporates a *system-oriented* perspective, illustrated in Figure 3.1. In this perspective, we use a local neighbourhood to perform estimations, but we rely on recursion to propagate known values through direct neighbours over a network of (mostly indirectly) mutually interacting cells, iteratively updated until an equilibrium is reached. At every recursive call, a weight is applied to the values being propagated to represent spatial autocorrelation. This weight can furthermore be assigned dynamically in a data-driven manner, based on the features of the underlying spatial configuration. This allows for higher autocorrelation weights between, for example, two neighbouring blocks of a city, and lower weights between an industrious port and the open sea. The update rules for every cell were based on the Bellman equation for *Markov reward processes* [96], canonically used to estimate the value of a particular state (cell). With this perspective, we can address both the limitations of local methods and distance-based methods.

Our main contributions in this chapter are as follows:

- We propose a novel method, VPint, for spatial interpolation, incorporating a system-oriented perspective aimed at overcoming the limitations of existing local- or distance-based methods.

- We introduce two variants of our value propagation interpolation algorithm, both of which incorporate elements of Markov reward processes: SD-MRP, using a static discount throughout the grid and requiring no additional data, and WP-MRP, exploiting spatial features to predict neighbour-specific spatial weights.

- We provide a vectorised implementation of our methods allowing for high degrees of parallel processing to speed up the algorithm running time, which we make publicly available[2].

- We empirically evaluate our methods on synthetic data and two real-world datasets and compare their performance against that of popular baselines from the Kriging, machine learning and deep learning fields in terms of mean absolute error, root mean squared error, peak signal-to-noise ratio and structural similarity. We also conducted experiments testing convergence, scalability, and whether the proposed method generalises to spatio-temporal data.

3.2. RELATED WORK

To date, various spatial interpolation methods, both local and distance-based, have been proposed. We will discuss a selection of popular methods in this section.

Gaussian processes. Given its widespread use, the first set of methods of note are Gaussian processes (GP), also known as Kriging [97]. GPs [91, 98] are a set of interpolation techniques based on learning the covariance of target values over distance using variogram (kernel) functions fitted to the data. Popular variants of GPs are discussed in [89] and include ordinary Kriging (OK), universal Kriging (UK) and regression Kriging (Kriging after detrending). Contemporary contributions to GP methods include a scalable gradient-based surrogate function method [99] and a neural network-based method to overcome GPs' limitation of disregarding the characteristics of intermediate locations in paths between pairs of locations [100]. Although the assumptions made differ per variant, all GP-based methods are limited by their reliance on pair-wise distance-based covariance models. Moreover, traditional GP methods tend to scale poorly to larger datasets ($O((nm)^4)$). An overview of modern GP methods aimed at increasing the viability of GPs for large-scale datasets is given in [88], including local approximate GPs [101], stochastic partial differential equation approaches [102] and multi-resolution approximations [103].

Gapfill. Gapfill [104] is a local method utilising no explanatory variables that, unlike GPs, does not build an explicit statistical model. Instead, as a local method, it relies on using subsets of the available data for its predictions. Although its local perspective and cell-specific independent predictions allow gapfill to be highly parallelised, its performance in terms of accuracy tends to fall short of GPs [88], and its dependency on the presence of sufficient amounts of non-missing values within its neighbourhood renders it infeasible for cases where missing values are

[2] https://github.com/ADA-research/VPint

clustered together.

Belief propagation. This family of methods, particularly loopy belief propagation [105], has been used successfully for image denoising [106], image restoration [107] and image completion tasks [108]. It generally considers graphical models [109], such as Markov random fields, and gridded datasets can also be converted to this representation. The key idea is to compute the marginal distributions of nodes in a network, based on the beliefs (estimations) of the values of the child nodes connected to them. This is done through a process of *message passing*, which iteratively propagates beliefs over the network. Conceptually, this type of method is similar to our proposed method, although it does not leverage the Bellman equation or data-driven spatial weights, and unlike belief propagation, our method computes predicted values rather than distributions thereof. While belief propagation can be considered to take the system-oriented perspective, its exact form operating on junction trees scales poorly to larger graphs ($\mathcal{O}(M \cdot N^3)$, where M denotes the number of nodes and N is the number of discrete states per node) [110] and high precision continuous variables [111], rendering it computationally infeasible for most practical applications of grid-based interpolation tasks. Similarly, the standard approximate loopy belief propagation algorithm may have high errors compared to other methods [112], or even oscillate rather than converge [113].

Spatial regression. Spatial autoregressive models [92] (SAR) have remained relatively consistent, but have been expanded in some recent work [114] [115]. Moving average (MA) models are often used in the context of time-series modelling [116], but can also be used for spatial regression problems using the "MA by AR" approach [93]. Highly related to SAR and MA models, autoregressive moving average (ARMA) models have seen recent work of particular relevance to the COVID-19 pandemic, modelling a transmission network of influenza [117]. Apart from SAR, MA, and ARMA models, which include additional features for the spatial lag and/or residuals, there are also approaches using an explicit spatial, temporal or spatio-temporal data representation, such as the tensor decomposition-based work by Corizzo et al [118]. This latter work seems particularly relevant for spatial interpolation problems where collinearity exists within the explanatory variables. Given that the explanatory variables are being leveraged for their shared spatial structure with the target variable, collinearity in the explanatory variables is a likely scenario. However, unlike our method, these types of method are not interpolation methods, aiming instead at predicting target values directly from the spatio-temporal features or a latent representation thereof. Other recent work using spatial regression approaches include house price estimation using geographically weighted regression [119], varying coefficient spatio-temporal regression [120], ambient black-carbon prediction [121] and an analysis of the spatial patterns of COVID-19 [122].

The spatial autoregressive regression models suffer from the limitations of a local perspective: their use of a pre-defined local neighbourhood dismisses information outside of the neighbourhood radius.

Neural networks and deep learning. Deep learning techniques, and convolutional neural network (CNN) in particular, have been used to great effect in many computer vision applications [94, 95]. These computer vision-based interpolation CNNs could also be applied to general spatial interpolation. Moreover, in their 2020 publication, Hashimoto and Suto formulated a CNN architecture for the specific purpose of spatial interpolation [123]. Apart from CNNs, graph neural networks (GNNs) have also been applied recently to spatio-temporal interpolation by Wu et al [124], utilising fully connected networks with distance-based weights determined using a random subgraph sampling strategy. Like autoregressive models, CNNs have a local perspective and therefore, dismiss potentially meaningful information outside their predefined neighbourhood. Conversely, similar to GPs, GNNs suffer from the reliance on distance-based weights, dismissing potential non-homogeneity of intermediate locations on paths between locations.

By adopting a system-oriented perspective, the method we propose in this chapter aims to be situated between these two main categories of existing work (local and distance-based). Moreover, like Gapfill, it offers a computational alternative to existing methods with an emphasis on explicit statistical spatial modelling.

3.3. PROBLEM STATEMENT

Let us define a spatial grid $\mathbf{G}$ as an $(n \times m)$ matrix, where n corresponds to the number of rows and m to the number of columns. We consider a target variable y and its value y_c at a specific location (grid cell) c in $\mathbf{G}$, where $c = \mathbf{G}_{i,j}$ and i and j correspond to the row and column indices in $\mathbf{G}$, respectively. In parameter estimation settings, y usually corresponds to in-situ ground truth measurements of a parameter $p \in P$, where at every location c, there is a configuration of parameters $\boldsymbol{\theta}$ such that $y_c \in \boldsymbol{\theta}$. At every location c, there exists a true value y_c^* that may be either known or unknown. If y_c^* is known, we set the cell value $y_c = y_c^*$. If it is not known, we mark this location as unknown: $y_c = \varnothing$. The $(n \times m)$ matrix $\mathbf{Y}$ contains y_c for all c in $\mathbf{G}$.

We further define a feature grid $\mathbf{Z}$ as an $(n \times m \times f)$ tensor, where f denotes the number of features per cell. Thus $\mathbf{z}_c$ in $\mathbf{Z}$ is a feature vector corresponding to location c in $\mathbf{G}$. In parameter estimation, these features are generally not the target parameters P nor the satellite observations $\mathbf{x}$, but rather covariates used for the spatial interpolation task outside of the parameter estimation problem. We can now define a prediction model $\mathcal{M}_{int}(\mathbf{Y}, \mathbf{Z})$ that takes as input the available data in

Y, along with the corresponding feature vectors per location in **Z**, and returns a prediction matrix $\hat{\mathbf{Y}}$. The objective of spatial interpolation is to find a model $\mathcal{M}^*_{int}$ that minimises the mean absolute error (MAE) for all locations c in **G**, given the predictions in $\hat{\mathbf{Y}}$. Concretely:

$$\mathcal{M}^*_{int} \in \underset{\mathcal{M}_{int}}{\operatorname{argmin}} \sum_{c \in \mathbf{G}} |\hat{y}_c - y^*_c| \tag{3.1}$$

3.4. METHODS

In this section, we will describe our proposed interpolation method, VPint, in four steps. The general procedure and main philosophy will first be illustrated, after which we introduce some background for our update rules, and propose the two concrete variants of our method that we implemented. Finally, we will discuss our approach for ensuring efficient computation allowed by parallel matrix operations.

3.4.1. GENERAL INTERPOLATION PROCEDURE

The core of our proposed method relies on iterative element-wise updates to an estimation grid. We first instantiate $\hat{\mathbf{Y}}$, with missing values given by **Y** being set to arbitrary real values as initial predictions (the mean of known values in our experiments). Next, for every cell $c \in \mathbf{G}$, if $\mathbf{Y}_c$ is known, we use it as a static prediction. If it is not known, we update its value using the *estimated* value of its neighbours $\{c' : c' \in N_S(c)\}$, where $N_S(c)$ denotes the set of spatial neighbours to cell c. Thus, by iterating this procedure, our algorithm recursively propagates known values throughout chains of estimated values in $\hat{\mathbf{Y}}$, through all possible paths in the system, anchored by known values.

3.4.2. BACKGROUND: UPDATE RULE

Our update rule is based on Markov reward processes (MRPs). MRPs [96] are models of the form $M = \{S, T, R\}$, where S is a set of states $\{s_1, s_2, ..., s_{|S|}\}$, **T** is an $|S| \times |S|$ matrix of transition probabilities $T_{(s,s')}$ between all pairs of states s and s', and R is a set of rewards $\{r_{s_1}, r_{s_2}, ..., r_{s_{|S|}}\}$ associated with being in a state s. MRPs extend Markov chains, which do not incorporate *rewards* R, and have been successfully used to model the behaviour of a single variable over time [125, 126]. In these temporal models, a state s represents a set of attribute values at a particular time t in a sample trajectory over time. At every t a state s can probabilistically transition from s to any of a set of successor states (given the current state) $S'|s = \{s'|s_1, s'|s_2, ..., s'|s_{|S|}\}$ based on transition probabilities given by $T_{(s,s')}$, until an *absorbing state* is reached from which no further transitions are possible: $|S'|s| = 0$. Since MRPs are Markovian, the transition probability to go from s to s' are contin-

gent solely on s, and are unaffected by the history of previous states in the trajectory. If a reward r_s is associated with the state s, this gives information about the desirability of state s. However, aside from this immediate reward r_s, intuitively the expected future rewards $\mathbf{E}(s')$ from all $s' \in S'|s$ should also be considered, as states leading to successor states with high future rewards would be more desirable. This leads to a notion of *state values*, where the rewards of all possible successor states s' are used to recursively compute state values $v(s)$ for all $s \in S$. This is typically done by iterating the Bellman equation [96], where the immediate reward $r_{(s,s')}$ is added to the discounted (using the discount parameter γ) average expected values of the successor states:

$$s' : v(s) = \frac{1}{|S'|s|} \cdot \sum_{s' \in S'|s} r_{(s)} + \gamma \cdot \mathbf{E}(s') \tag{3.2}$$

We opted to use this equation as our interpolation update rule. In the case of interpolation, a location c (at a certain time) can be seen as a state s, with the set of spatial neighbours $N_S(c)$ being analogous to the set of successor states $S'|s$ in MRPs. The state values $v(s)$, then, would be the target variable $\hat{y}_c$ to be estimated, with immediate rewards given by known values and the discount γ representing spatial autocorrelation. Using the Bellman equation as an update rule, we can define the set of spatial neighbours $N_S(c)$ as the cells $\{c' : c' \in \mathbf{G}\}$ that share a border with c, such that our spatial interpolation algorithm takes the form of:

$$\hat{y}_c = \begin{cases} y_c & \text{if } y_c \text{ known,} \\ A_S(c) & \text{otherwise} \end{cases} \tag{3.3}$$

Here, $A_S(c)$ denotes an aggregation function over the spatial neighbourhood of c. While in principle, it is possible to add any user-defined aggregation function, we opted to stay close to the canonical Bellman equation, by taking the mean (spatial lag) of $N_S(c)$:

$$A_S(c) = \frac{1}{|N_S(c)|} \cdot \sum_{c' \in N_S(c)} \gamma \cdot \hat{y}_{c'} \tag{3.4}$$

Using Equation 3.4 also allows us to provide an efficient vectorised implementation of our method. While the method runs for a set amount of iterations in principle, it can also incorporate an early stopping criterion by introducing a variable δ, representing the change of a configuration over iterations, and using $\hat{y}_c^{(-1)}$ to denote the predictions from the previous iteration:

$$\delta = \frac{1}{m \cdot n} \cdot \sum_{c \in \mathbf{Y}} |\hat{y}_c - \hat{y}_c^{(-1)}| \tag{3.5}$$

This then allows for the early stopping of the algorithm if δ drops below a user-specified threshold.

One could also consider generalising this approach to spatio-temporal interpolation problems. In that case, the algorithm cannot solely rely on Equation 3.3. Whereas two spatial dimensions share the same scale, and can thus both use the same weight γ as a spatial discount, a temporal dimension may behave very differently. As a result, to generalise to a spatio-temporal domain, we need to introduce an additional parameter τ for discounts representing temporal autocorrelation. This also leads to the set of temporal neighbours $N_T(c)$, which represent the same location at different time steps. Thus, the spatio-temporal update rule becomes:

$$\hat{\mathbf{Y}}_c = \begin{cases} \mathbf{Y}_c & \text{if } \mathbf{Y}_c \text{ known} \\ A_S(c) + A_T(c) & \text{otherwise,} \end{cases} \tag{3.6}$$

where $A_T(c)$ will generally use the temporal lag aggregation function:

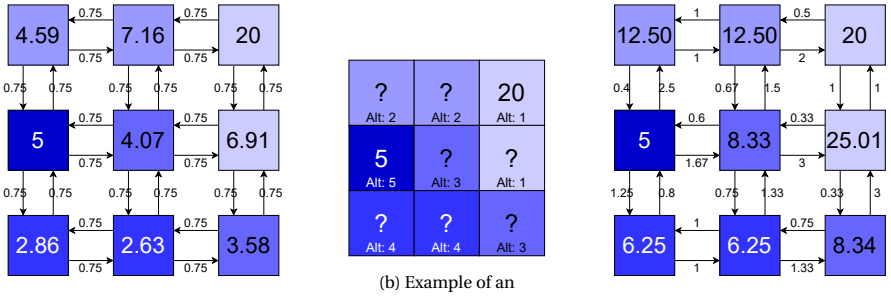
$$A_T(c) = \frac{1}{|N_T(c)|} \cdot \sum_{c'_t \in N_T(c)} \tau \cdot \hat{\mathbf{Y}}_{c'_t} \tag{3.7}$$

3.4.3. VARIANTS

We propose two variants of our value propagation interpolation method. The first, SD-MRP (static discount-MRP), uses a single spatial weight parameter γ for the entire dataset, which can be tuned using random search on subsampled data from known values. The second variant, WP-MRP (weight prediction-MRP) exploits spatial data as explanatory variables to inform its prediction of neighbour-specific weights. Unlike SD-MRP, WP-MRP would therefore not assume isotropy (the same spatial effects in all directions), although it would necessitate the use of Equation 3.4 as an aggregation function. The two variants applied to the example of Figure 3.2b, visualising the interpolation problem of temperature measurements in the Himalayas, are shown in Figure 3.2.

BASIC STATIC DISCOUNTS: SD-MRP

The most basic variant of our proposed method stays closest to the canonical form of the Bellman equation in Equation 3.3. It uses a single discount parameter γ, ranging between 0 and 1, to represent spatial autocorrelation. This means that, for SD-MRP, values can only *decrease* over subsequent recursive calls, making known values reminiscent of a light source in the fog, radiating values around itself and merging with other light sources, but decaying over distance. In the example of spatial interpolation of temperatures, it would propagate the known temperature



(a) SD-MRP

(b) Example of an interpolation problem.

(c) WP-MRP

Figure 3.2: Comparison of the interpolation procedures of SD-MRP (3.2a) and WP-MRP (3.2c) for the example in Figure 3.2b. The values in each cell represent temperature measurements, and the colour of a cell indicates the elevation of a location, where darker colours represent higher elevation. We wish to interpolate these values, such that all '?' are filled with estimated values instead, based on the known values 20 and 5. In SD-MRP, a static discount of $\gamma = 0.75$ was used, meaning values only decrease over distance, but do merge with one another. Meanwhile, for WP-MRP, the information on elevation was used to inform the interpolation, where in this case, the weight was chosen to be inversely proportional to the difference in features (higher elevation led to weights lower than 1 and vice versa).

values over the grid, at an intensity decreasing with every recursive call, like a heat source dissipating over distance. This can be seen in Figure 3.2a. The advantage of this method is that it does not require additional features to be applied to a dataset, nor does it require a prediction model to be explicitly trained. It will also regress to the initialisation value (such as 0, or the mean value) over distance, which can be a desirable property as uncertainty increases, but can also be considered a downside as it does not provide much additional information. Its main hyperparameter γ also requires tuning, which can be done automatically by subsampling known values and performing interpolation using randomly searched γ settings. Furthermore, the spatial characteristics of the grid are not taken into account, and isotropy is assumed. SD-MRP has a time complexity of $\mathcal{O}(4 \cdot |\mathbf{Y}| \cdot k)$, if k is the number of times Equation 3.3 is iterated (every $c \in \mathbf{Y}$ can have at most 4 neighbours).

WP-MRP

In an ideal case, rather than using a single static weight γ, we would use a method allowing us to use location-specific weights $\gamma_{c',c}$. For example, when spatially interpolating temperatures in the Himalayas, knowing the difference in elevation between two neighbouring locations would enable us to know whether one value is likely to be the same, lower, or higher than the other, as illustrated in Figure 3.2c. To this end, we created the weight prediction variant WP-MRP, in which we

use the spatial feature vectors $\mathbf{z}_c \in \mathbf{Z}$ and $\mathbf{z}_{c'} \in \mathbf{Z}$ as inputs to a weight prediction model $\mathcal{M}_w$. This model predicts an individual weight of the location pair (c, c') as $\gamma_{c',c} = \mathcal{M}_w(\mathbf{z}_{c'}, \mathbf{z}_c)$ from spatial data describing the locations (such as houses, shops and land use). $\mathcal{M}_w$ could consist of any machine learning model, ensemble or pipeline, but could also leverage functions directly operating on the feature space such as distance measures and inverse similarity metrics. Mapping features to a dense data manifold of lower dimensionality could also be considered.

In the case of machine learning models and pipelines, in order to train the model, we use the available cells with known true values in $\mathbf{Y}$ to supervise the training. For all pairs of neighbours $\{(c, c') : c, c' \in \mathbf{Y} \wedge y_c \neq \varnothing \wedge y_{c'} \neq \varnothing\}$, we would compute the true weight using the fraction $\gamma^*_{c',c} = \frac{y^*_c}{y^*_{c'}}$, resulting in a ground truth vector Γ^* that can be used as the targets for the training of a regression model. The method for matching the elements of Γ^* to predictive features is a design choice: the location features $\mathbf{z}_{c'}$ and $\mathbf{z}_c$ of every location pair (c', c) would need to be combined, and this could be done in any manner the situation calls for, such as adding the vectors or computing a distance metric. In our experiments we opted to simply concatenate $\mathbf{z}_{c'}$ and $\mathbf{z}_c$. Thus, with Γ^* and $\mathbf{z}_{c'}$, $\mathbf{z}_c$ for all (c', c) pairs, we can train a regression model $\mathcal{M}^*_w(\mathbf{z}', \mathbf{z})$, such that, if $\mathbf{Y}_N := \{(c', c) : (c', c \in \mathbf{Y}) \wedge (c' \in N(c))\}$:

$$\mathcal{M}^*_w \in \underset{\mathcal{M}_w}{\operatorname{argmin}} \frac{1}{|\Gamma^*|} \cdot \sum_{(c',c)\in\mathbf{Y}_N} |\mathcal{M}_w(\mathbf{z}_{c'}, \mathbf{z}_c) - \gamma^*_{c',c}| \tag{3.8}$$

Here we propose to train $\mathcal{M}_w$ on Γ^* using any regression (machine learning) algorithm. The full pipeline of WP-MRP using machine learning weight prediction is outlined in Algorithm 1 (which assumes available functions for model fitting). Lines 1-7 generate the elements of the true weight vector Γ^*, and line 8 fits a weight prediction model to the weights found in line 4. Lines 9-22 show the iterative updates of cells in $\mathbf{Y}$, and lines 23-27 create and return the predictions in the form of a grid $\hat{\mathbf{Y}}$. The time complexity to run WP-MRP is the same as that of SD-MRP, but with the added cost of the model used for $\mathcal{M}_w$ (which can be chosen freely): $\mathcal{O}(4|\mathbf{Y}| \cdot k) + \mathcal{O}_{\mathcal{M}_w}$, if $\mathcal{O}_{\mathcal{M}_w}$ is the time complexity of making predictions with $\mathcal{M}_w$.

Algorithm 1: VPint (WP-MRP)

Input: Target matrix $\mathcal{Y}$, feature matrix $\mathbf{Z}$, maximum MRP iterations max_iter
Result: Interpolated matrix $\hat{\mathbf{Y}}$

1: **for all** $c \in \mathbf{Y}^{train}$ **do**
2: **for all** $c' \in N_S(c)$ **do**
3: **if** $y_c \neq \varnothing$ and $y_{c'} \neq \varnothing$ **then**
4: $\gamma^*_{c',c} = \frac{y^*_c}{y^*_{c'}}$ ▷ $N_S(c)$ denotes neighbours of c
5: $\mathcal{M}_w := fit\_model(\mathbf{Z}, \Gamma^*)$;
6: $iter := 0$;
7: **while** $iter < max\_iter$ **do**
8: **for all** $c \in \mathbf{Y}$ **do**
9: **if** $y_c = \varnothing$ **then**
10: $\hat{y}_c := 0$;
11: **for all** $c' \in N_S(c)$ **do**
12: $\hat{y}_c := \hat{y}_c + \mathcal{M}_w(\mathbf{z}_{c'}, \mathbf{z}_c) \cdot \hat{y}_{c'}$; ▷ $N_S(c)$ denotes neighbours of c
13: **else**
14: $\hat{y}_c := y_c$;
15: $iter := iter + 1$;
16: $\hat{\mathbf{Y}} := \mathbf{Y}$;
17: **for all** $c \in \mathbf{Y}$ **do**
18: $\hat{\mathbf{Y}}_c := \hat{y}_c$;
19: **return** $\hat{\mathbf{Y}}$;

3.4.4. VECTOR-BASED UPDATE RULE FOR PARALLEL COMPUTATION

For the efficient processing of the main iterative loop of lines of our algorithms as in lines 7-15 Algorithm 1, we reformulated our update function as a series of matrix operations, allowing updates to be carried out in a highly parallelised manner through vectorisation. This approach does, however, necessitate the use of weighted averaged (spatial lag) as an aggregation function. We will illustrate the procedure on the simpler case (spatial MRP), but the approach can be generalised to spatio-temporal MRP as well. The main idea of this approach is to shuffle neighbouring values around in matrices and tensors $\mathbf{T}^{y_i}$, where the subscript y indicates this tensor contains values, and i indicates the stage of operations the data is currently in, with the accompanying neighbour weights in $\mathbf{T}^{\gamma_i}$, where γ indicates this tensor contains weights. These operations are performed in order to compute weighted sums of neighbouring values for all cells in the grid as a matrix dot product in Equation 3.11.

Concretely, let $\hat{\mathbf{Y}}$ denote a matrix of size $(n \times m)$ containing predicted values $\hat{y}_c$ at every cell where the true value is not known, and y_c otherwise. We first turn this

matrix into a three-dimensional tensor $\mathbf{T}^{y_0}$ of size $(n \times m \times d)$, where d denotes the maximum number of neighbours $max(|N_S(c)|)$ for any cell $c \in \mathbf{G}$ (in practice, this will generally be 4 as a cell can share at most 4 edges in a grid). For all c, the entries along the d-axis of $\mathbf{T}^{y_0}$ will contain the values of the neighbours of c. Concretely:

$$\mathbf{T}^{y_0}_{c,d_j} = \hat{y}_{c'} : c' \in N_S(c) \tag{3.9}$$

If $|N_S(c)| < d$, the remaining values of the third dimension of $\mathbf{T}^{y_0}_c$ are set to 0. We similarly construct a tensor $\mathbf{T}^{\gamma_0}$ of size $(h \times w \times d)$, of which the entries match those of $\mathbf{T}^{y_0}$. However, the values of this tensor contain weights $\gamma_{c',c}$ from neighbour c' to cell c, rather than the values:

$$\mathbf{T}^{\gamma_0}_{c,d_j} = \Gamma_{c',c} : c' \in N_S(c) \tag{3.10}$$

Next, we systematically stack all columns of $\mathbf{T}^{y_0}$ and $\mathbf{T}^{\gamma_0}$ as additional rows, resulting in the new matrices $\mathbf{T}^{y_1}$ and $\mathbf{T}^{\gamma_1}$ of size $(h \cdot w \times d)$. Now every row represents a single location c in a single dimension, although the information on the original columns of $\mathbf{Y}$ is kept through the order of the rows. The columns of $\mathbf{T}^{y_1}$ now show the values of the neighbouring values for a row's location's neighbours $N_S(c)$, and the columns of $\mathbf{T}^{\gamma_1}$ contain the corresponding weights. We now perform an MRP update by computing the dot product of $\mathbf{T}^{y_1}$ and the transpose of $(\mathbf{T}^{\gamma_1})$, and placing its diagonal values into a new vector $\mathbf{T}^{y_2}$ of size $(h \cdot w)$:

$$\mathbf{T}^{y_2} = diag(\mathbf{T}^{y_1} \cdot (\mathbf{T}^{\gamma_1})^\top) \tag{3.11}$$

Since this vector has the same order as the rows of $\mathbf{T}^{y_1}$, we can reshape this vector into a matrix $\mathbf{T}^{y_3}$ of size $(h \times w)$, corresponding to the shape of $\hat{\mathbf{Y}}$. We now create another $(h \times w)$ matrix $\mathbf{T}^n$, where $\mathbf{T}^n_c = |N_S(c)|$, allowing us to divide $\mathbf{T}^{y_3}/\mathbf{T}^n$ element-wise, resulting in an updated prediction matrix $\hat{\mathbf{Y}}$:

$$\hat{\mathbf{Y}} = \frac{\mathbf{T}^{y_3}}{\mathbf{T}^n} \tag{3.12}$$

Finally, since this operation needlessly updated known values, we substitute original known values in $\hat{\mathbf{Y}}$: $\hat{\mathbf{Y}}_c = \mathbf{Y}_c$ for all $c : \mathbf{Y}_c \neq \varnothing$.

Using this vectorised approach, we found that the complexity of our algorithms in terms of wall-clock time improved by a factor between 10 and 100. In order to adapt this approach to spatio-temporal MRP interpolation, which adds an extra dimension for time, $\mathbf{Y}$ is of size $(h \times w \times t)$, $\mathbf{T}^{y_0}$ and $\mathbf{T}^{\gamma_0}$ are of size $(h \times w \times t \times d)$, and d becomes equal to 6, as any cell can now have up to 6 neighbours. Since all neighbours are already included in the fourth dimension, there is no reason to

keep the spatial and temporal dimensions separate. Thus, we can still generate the 2D matrices $\mathbf{T}^{y_1}$ and $\mathbf{T}^{\gamma_1}$, as we simply add another dimension to the stacking operation (resulting in $h \cdot w \cdot t$ rows instead of $h \cdot w$). As a result, with these exceptions, the pipeline can remain the same as it was for the spatial case.

3.5. EXPERIMENTS

In this section, we will share the details of our experiments. We will first introduce the research questions we were interested in, after which we will list the baselines we compared our method to and the datasets used in our experiments.

3.5.1. RESEARCH QUESTIONS

We were interested in answering the following chapter research questions with our experiments:

- CRQ1: How does VPint compare to baseline methods in terms of mean absolute error, root mean squared error, peak signal-to-noise ratio and structural similarity?
- CRQ2: Does VPint converge to stable prediction values?
- CRQ3: Can VPint be generalised to spatio-temporal problems?
- CRQ4: Can WP-MRP leverage spatial features to perform better than SD-MRP, given sufficiently informative features?
- CRQ5: How do VPint and baseline methods scale as the size of the dataset increases?

In addition to these main research questions, we were also interested in whether different patterns of missing data would give different results.

3.5.2. BASELINES

Our selection of baselines was aimed at including competitive interpolation and regression methods used for spatial and geo-spatial modelling in practice. The selection we made consists of:

- **Ordinary Kriging** (OK), using an implementation by the Python library PyKrige [127]. Like our proposed methods, ordinary Kriging predicts values using weighted sums:

$$\hat{y}_c = \sum_{c' \in N_S(c)} \gamma_{c',c} \cdot y_{c'} \tag{3.13}$$

Here $\gamma_{c',c}$ is the distance-based weight between known cell c' and unknown cell c. However, OK uses $y_{c'}$ instead of $\hat{y}_{c'}$, $N_S(c)$ will contain more cells than only direct neighbours, and weights are determined using a distance-based variogram model.

- **Universal Kriging** (UK), also using PyKrige's implementation. UK is highly similar to OK, but it compensates for the possible existence of a trend in the data. For both OK and UK, while more advanced methods exist, such as local approximate Gaussian processes [101], as these are aimed at improving the scalability of Kriging rather than its accuracy, we consider OK and UK to be suitable representative methods for this class of algorithm.

- **Loopy belief propagation**, using a Python implementation for denoising images[3], which can be applied to interpolation problems by treating missing values as noise (generated from a uniform distribution centred around the mean of the known values, with a range based on their standard deviation). In a basic form, belief propagation is centred around the equation:

$$L(\hat{y}_c) = \Pi_{c' \in N_S(c)} \lambda(\hat{y}_{c'}) \tag{3.14}$$

Here, $L(\hat{y}_c)$ refers to the likelihood of y_c^* being equal to $\hat{y}_c$, and $\lambda(\hat{y}_{c'})$ is the likelihood of the neighbouring values (children) $c' \in N_S(c)$. To ultimately produce a single predicted value, the most likely value can be used:

$$\hat{y}_c \in \operatorname{argmax} L(\hat{y}_c) \tag{3.15}$$

- **Non-spatial regression**, using auto-sklearn [128] to select the best performing regression model (or ensemble) and hyperparameter settings out of a large collection of algorithms, including linear regression, support vector regression, gradient boosted methods and others.[4] We denote this model, which will typically be an ensemble of multiple powerful machine learning models, as $\mathcal{F}$. The resulting general form of the predictions from non-spatial regression is:

$$\hat{y}_c = \mathcal{F}(\mathbf{z}_c) \tag{3.16}$$

We allowed auto-sklearn 150 seconds per run to find the best performing ensemble.

[3] Source code used: https://github.com/sanjeevg15/loopy-bp-denoise

[4] auto-sklearn is an automated machine learning (AutoML) package that allows automatic algorithm selection, hyperparameter optimisation and feature preprocessing ensuring that a high-performing pipeline is selected on given dataset

- **Spatial autoregressive** (SAR), **moving average** (MA) and **autoregressive moving average** (ARMA) models, using auto-sklearn to find the best performing regression model. Canonically, these models are ordinary least squares (OLS)-based linear regression methods, with extra spatial (SAR) or error (MA) terms (both in the case of ARMA). However, since we use auto-sklearn, though OLS is also a possible model, the final model will generally have a different formula, such as the potentially non-linear support vector regression models. For SAR, the spatial term is based on a spatial weight matrix **WM** and a vector **y** containing all the known values of the grid, corresponding to the rows of **WM**. The general form of SAR is:

$$\hat{y}_c = \mathcal{F}(\mathbf{z}_c, \mathbf{WM}, \mathbf{y}) \tag{3.17}$$

  For MA models, we used the "MA by AR" approach [93]. Its formula, using the prediction error vector ϵ instead of SAR's **y**, is:

$$\hat{y}_c = \mathcal{F}(\mathbf{z}_c, \mathbf{WM}, \epsilon) \tag{3.18}$$

  Following the "MA by AR" approach, before we can use Equation 3.18, we first needed to determine ϵ using:

$$\epsilon_c = y_c - \mathcal{F}_s(\mathbf{z}_c) \tag{3.19}$$

  Here, $\mathcal{F}_s$ represents a separate non-spatial regression model, as in Equation 3.16, to compute prediction errors on all known values. These errors can then be used by Equation 3.18 by putting the values of ϵ_c for all c into a single vector ϵ. For ARMA, we again use the "MA by AR" approach for the MA component. As ARMA is a combination of SAR and MA, its formula is:

$$\hat{y}_c = \mathcal{F}(\mathbf{z}_c, \mathbf{WM}, \mathbf{y}, \epsilon), \tag{3.20}$$

  where ϵ is obtained using Equation 3.19.

- **Convolutional neural networks** (CNN), optimised using automated neural architecture search (NAS). The CNN regression predicted $\hat{y}_c$ from $\mathbf{z}_c$ and $\mathbf{z}_{c'}$ for all $c' \in N_S(c)$, where $N_S(c)$ is determined by the convolutional filters of the network, similar to CNN approaches used in computer vision [94, 95]. We used NAS implemented by auto-keras [129] for all training sets (50 trials, 1000 epochs). Although the model architectures for CNNs can be quite complex, on an abstract level these networks are still regression models of the same form as Equation 3.16.

3.5.3. DATASETS

Our main experiments involved a synthetic spatial dataset as well as two real-world datasets (GDP and COVID-19 trajectories), with an additional synthetic spatio-temporal dataset used to address CRQ3. The implementation of our data generation algorithms used to create the experimental synthetic datasets is included in our public code repository; likewise, the real-world datasets are available for public use at their respective sources, allowing others to reproduce our results.

SYNTHETIC DATA

Spatial targets. For this synthetic dataset, based on a parameterised mean μ and standard deviation σ, the interpolation grid $\mathbf{Y}$ of user-specified size $(n \times m)$ (set to $n = 50$ and $m = 50$ in our experiments) was generated, where each cell c was assigned a base value y_c^b by sampling from the normal distribution $\mathcal{N}(\mu, \sigma)$. Next, to assign true values y^* affected by spatial interaction, we updated every cell c as a weighted average (based on a *spatial autocorrelation* parameter a^s) of its own value and the mean of its neighbouring values:

$$y_c^* = (1 - a^s) \cdot y_c^b + a^s \cdot \frac{1}{|N_S(c)|} \cdot \sum_{c \in N_S(c)} y_c^b \tag{3.21}$$

Spatio-temporal targets. To address CRQ3, we also generated synthetic spatio-temporal data. For this type of data we introduced additional parameters for the number of timesteps d and the temporal autocorrelation coefficient a^t. We then built a three-dimensional tensor $\mathbf{Y}$ of size $(n \times m \times d)$ by using Equation 3.21 at every time step. Since, at this point, the temporal layers of $\mathbf{Y}$ are still fully independent, we use the temporal neighbourhood function $N_T(c)$ to perform a final update on the cells of $\mathbf{Y}$ ensuring temporal interaction:

$$y_c^* = (1 - a^t) \cdot y_c^b + a^t \cdot \frac{1}{|N_T(c)|} \cdot \sum_{c \in N_T(c)} y_c^b \tag{3.22}$$

Synthetic features. For our synthetic data, we created a feature vector $\mathbf{z} = (z_{c_1}^b, z_{c_2}^b, ..., z_{c_{|\mathbf{z}^b|}})$ for every location $c \in \mathbf{Y}$. Every base feature $z_{c_i}^b \in \mathbf{z}_c^b$ was generated using a uniform distribution $\mathcal{U}(min, max)$ with user-specified min and max values. These features were then updated in a similar manner to the cell values y, using a parameter called the *feature correlation coefficient* f:

$$x_{c_k} = (1 - f) \cdot x_{c_k}^b + f \cdot y_c^* \tag{3.23}$$

A feature correlation coefficient f of 0 would result in fully random features, whereas a coefficient of 1 would result in features identical to the targets.

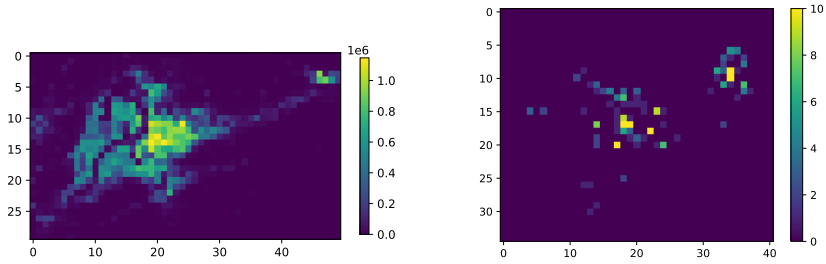
REAL-WORLD DATA

In the case of real-world data, the variables being measured, such as GDP or COVID-19 incidence, are generally not gridded in nature. As a result, to generate these datasets, data needs to be aggregated, e.g., by taking the mean (estimated) GDP per capita for residents in the area covered by a grid cell, or the sum of COVID-19 incidence at that location. The granularity of these datasets thus introduces a trade-off: a high granularity increases the computational cost and may result in relatively sparse datasets (as was the case in our COVID-19 dataset), but does provide a high level of detail. Meanwhile, a low granularity may result in data too low-grained to draw meaningful conclusions from, or cells that simply all regress to a global mean due to the erasure of local spatial patterns, but will be faster to compute and likely results in a higher density dataset. There is no minimal or maximum granularity cutoff point at which an interpolation method becomes infeasible. However, when gauging how applicable an interpolation method is to a users' gridded dataset, this trade-off merits consideration.

Gross domestic product (GDP) targets. For GDP data, we used a gridded spatial dataset containing worldwide GDP estimates sourced from World Bank [130] at a resolution of 1km × 1km. We specifically looked at the city of Taipei in Taiwan and its surroundings, including both heavily populated urban areas expected to have high GDP values, and surrounding sparsely populated mountainous areas with low GDP values. The resulting grid had a size of 51×51 pixels.

Aggregated COVID-19 trajectory targets. This dataset consisted of trajectories of confirmed COVID-19 patients prior to their diagnosis in South Korea [131]. Although this data was spatio-temporal in principle, we opted to aggregate over time both due to the relative sparsity of the data (as it was gathered at the start of the COVID-19 pandemic), and to alleviate potential privacy-related concerns in this relatively sensitive dataset. Thus, every $c \in G$ had a value corresponding to the total number of visits by people infected with COVID-19 over the entire time period. The city of interest in this dataset was Daegu, which was the main hotspot of the epidemic in South Korea at the time the data was collected. A visualisation of this data can be found in Figure 3.3b. Since the target data did not come in gridded form, we set the resolution of this dataset to 35×51 pixels, putting it at a similar scale to the GDP dataset used for Taipei.

Map-based features. To generate features for GDP and COVID-19 trajectories in South Korea and Taiwan, we aggregated a selection of vector and point map data sourced from OpenStreetMap [132]. For all $c \in \mathbf{G}$, every element in $\mathbf{z}_c$ represented the count of all objects in the map data corresponding to a certain *type*, such as apartments, houses and shops. There are various design choices available for preprocessing this type of data, such as dealing with objects without an annotated type (drop or replace), feature selection (none, manually created high-level



(a) GDP Taipei (b) COVID-19 Daegu

Figure 3.3: Visualisation of the GDP data in Taipei (a) and the COVID-19 dataset in Daegu (b). Due to the heavily localised infection clusters, we limited the data from (b) to a range of [0,10] (all values > 10 were set to 10) for greater visibility (the experiments used the raw values instead).

taxonomy, or keeping the most frequent types) and feature normalisation (none, unit length scaling, mean normalisation or Z-score normalisation). In accordance with the design philosophy of programming by optimisation (PbO) [133], we did not commit to any of these choices, and instead used a commonly used Bayesian optimisation-based automated algorithm configurator, version 0.12.0 of SMAC3 [134], to select the best possible feature construction pipeline per method (time budget 24 hours per algorithm per dataset).

3.5.4. EXPERIMENTAL SETUP

The following section will explain the procedures and experimental conditions necessary to carry out our experiments.

MISSING DATA PROCEDURES.

In order to evaluate our methods, we required data that was fully available to compute error metrics, while also having access to grids with missing data. To this end, we introduced two methods for 'hiding' known values, resulting in different patterns of missing data.

Random missing values. This missing value approach was straightforward. Given a proportion of known values p, for all other cells there is a probability of being randomly obscured if a number $z = \mathcal{U}(0,1)$ sampled from a uniform distribution between 0 and 1 is smaller than p. That is, for every cell $c \in G$:

$$y_c = \begin{cases} y_c^* & \text{if } z < p, \\ \varnothing & \text{otherwise} \end{cases} \tag{3.24}$$

In our experiments, p was set to 0.8.

Algorithm 2: Spatially clustered missing values

Input: Location grid **G**, true grid $\mathbf{Y}^* = (n, m)$, number of points k, number of walks w, number of steps per walk r

Result: Interpolation grid with missing values **Y**

```
Y := zeros(n, m)
num_points := 0
while num_points < k do
    i := U(0, n)
    j := U(0, m)
    c := G_{i,j}
    num_walks := 0
    while num_walks < w do
        num_steps := 0
        while num_steps < r do
            Y_c := ∅
            c := random_selection(N(c))
            num_steps := num_steps + 1
        num_walks := num_walks + 1
    num_points := num_points + 1
```


Spatially clustered hidden values. Much like the spatial data itself, the missing data points in a grid may not be independent, and instead subject to spatial auto-correlation themselves. For example, some locations may have missing data due to natural barriers making measurements difficult, or due to local phenomena such as clouds obscuring parts of the measurements. In this missing value approach, we were inspired by optical satellite data, where clouds are the biggest source of missing data in the field. This approach is also why algorithms like Gapfill [104] could not be considered for our experiments, as it requires a part of the data in a neighbourhood to be available. Our method for creating clusters of missing data was based on random walks. Given a number of points k, a number of walks w and the number of steps per walk r, the algorithm creating artificial clusters is outlined in Algorithm 2. When applied to spatio-temporal data, the spatially missing data was applied independently to every time step.

EXPERIMENTAL SETUP.

The general form of our experiments was to run 10 algorithms (SD-MRP, WP-MRP and 8 baselines) 30 times for two types of missing data (random and spatially clustered) on every dataset (3 in total, with 1 additional dataset for spatio-temporal data), including both synthetic data and real-world datasets and addressing CRQ1 and CRQ3. The performance of the methods was compared according to their

ranks based on the Wilcoxon signed-rank test [135], which is similar to a t-test but does not assume normality.

For spatial synthetic data we set the size of the grid to $n = 100$ and $m = 100$, and for the spatio-temporal synthetic data used to address CRQ3, we set the size to $n = 50$, $m = 50$ and the number of timesteps $d = 5$. Tracking δ allowed us to visualise the convergence of our method (CRQ2), using different settings for f in Equation 3.23 allowed us to gauge the effectiveness of WP-MRP relative to SD-MRP as a function of the correlation between the features and true values of locations (CRQ4), and varying n and m allowed us to see how well all methods scaled to larger datasets (CRQ5). Thus, in addition to the general performance results, we used the synthetic data to run additional experiments to address research questions 2 through 5. Conversely, we used the real-world datasets to gauge how well the performance on synthetic data, and the analysis thereof, would generalise to real-world cases. For the scalability analysis we set $n = m$, with n ranging from 20 to 200 in steps of 20. The spatially clustered missing data was generated using $k = 5$ centre points, $r = \frac{n+m}{2}$ steps per walk, and $w = \frac{r}{2}$ walks.

For every combination of a dataset with a type of missing data, all algorithms were run 30 times, and used automated algorithm configuration (for the feature preprocessing pipeline explained in Section 3.5.3.2), automated machine learning (methods using auto-sklearn, automating the selection of machine learning algorithms and their hyperparameters, as explained in Section 3.5.2), NAS (in the case of CNN, automating the neural network architecture explained in Section 3.5.2) and random search (in the case of SD-MRP's γ, explained in Section 3.4.3.1).

All experiments were run on a computing cluster consisting of 26 homogeneous nodes containing 94 GBs of memory and using Intel Xeon E5-2683 v4 CPUs running at 2.10GHz.

3.6. RESULTS

In this section, we will report on the results of our experiments. We first explain the performance metrics used, after which we will cover detailed results for all individual datasets (synthetic spatial data, GDP and COVID-19). These results were computed as the mean of 30 runs per algorithm and dataset for every performance metric. After covering the dataset-specific performance metrics, we investigate other properties of VPint: qualitative visual plausibility, the convergence of Equation 3.3, the degree to which it can be generalised to spatio-temporal problems, the required feature correlation for WP-MRP to perform better than SD-MRP, and the scaling of different methods to larger datasets. Finally, we will provide a high-level summary our findings.

3.6.1. PERFORMANCE METRICS

Since multiple properties can be desirable in an interpolation method, we evaluated our method based on 4 performance metrics. The first of these was the mean absolute error (MAE):

$$MAE(\hat{\mathbf{Y}}, \mathbf{Y}^*) = \frac{1}{|\mathbf{Y}|} \cdot \sum_{c \in \mathbf{Y}} |\hat{y}_c - y_c^*| \tag{3.25}$$

MAE is the main error metric reflecting the accuracy of the predictions obtain from the methods we studied, with all errors weighted equally. We also added root mean squared error (RMSE), which penalises extreme errors relatively more severely:

$$RMSE(\hat{\mathbf{Y}}, \mathbf{Y}^*) = \sqrt{\frac{1}{|\mathbf{Y}|} \cdot \sum_{c \in \mathbf{Y}} (\hat{y}_c - y_c^*)^2} \tag{3.26}$$

In addition to MAE and RMSE as basic error metrics, we also included two metrics common in the computer vision and image processing fields. The first of these is peak signal-to-noise ratio (PSNR):

$$PSNR(\hat{\mathbf{Y}}, \mathbf{Y}^*) = 20 \cdot \log_{10} \frac{max(\mathbf{Y}^*)}{RMSE} \tag{3.27}$$

PSNR is highly related to the root mean squared error (RMSE) metric, and in fact contains it as a component. It computes the logarithm of the RMSE scaled by the maximal (true) value $max(\mathbf{Y}^*)$; as such, it is expected to show similar patterns to RMSE. The main motivation for using PSNR is that it scales values by the maximal (true) value $max(\mathbf{Y}^*)$; therefore, PSNR results will have a similar range for the GDP dataset (where errors of over 30 000 were common) and the COVID-19 dataset (where errors were typically under 10). Finally, we looked at the structural similarity index (SSIM):

$$SSIM(\hat{\mathbf{Y}}, \mathbf{Y}^*) = \frac{(2 \cdot \mu_{\hat{\mathbf{Y}}} \cdot \mu_{\mathbf{Y}^*}) \cdot (2 \cdot \sigma_{\hat{\mathbf{Y}}\mathbf{Y}^*} + c_2)}{(\mu_{\hat{\mathbf{Y}}}^2 + \mu_{\mathbf{Y}^*}^2 + c_1) \cdot (\sigma_{\hat{\mathbf{Y}}}^2 + \sigma_{\mathbf{Y}^*}^2 + c_2)} \tag{3.28}$$

SSIM aims to quantify the similarity of two images in a manner consistent with human perception, emphasising spatial structure over absolute errors.

3.6.2. EMPIRICAL PERFORMANCE (CRQ1)

The results presented in this subsection are dedicated to answering CRQ1. We ran detailed experiments on the synthetic spatial dataset as well as the real-world GDP per capita and COVID-19 datasets.

Synthetic spatial data. The results for synthetic spatial data are shown in

Algorithm	MAE		RMSE		PSNR		SSIM	
	random	clustered	random	clustered	random	clustered	random	clustered
Ordinary Kriging	2.392	2.59	9.004	10.446	0.386	0.379	0.06	-0.009
Universal Kriging	2.381	2.476	9.053	9.609	0.386	0.383	0.062	0.008
Belief propagation	19.981	20.028	408.996	410.901	0.220	0.220	0.000	0.000
Non-spatial regression	2.459	2.418	9.507	9.228	0.384	0.385	0.051	0.088
SAR	2.006	2.333	6.61	8.784	0.399	0.387	**0.397**	**0.156**
MA	2.462	2.459	9.549	9.523	0.383	0.383	0.084	0.036
ARMA	1.969	**2.297**	6.357	**8.423**	0.401	**0.389**	**0.398**	**0.16**
CNN	341.277	47.391	1.263×10^{3}	35.520	0.02	0.288	0.002	-0.0
SD-MRP	3.28	4.71	19.363	42.023	0.37	0.339	0.341	**0.156**
WP-MRP	**1.949**	2.248×10^{10}	**6.244**	1.503×10^{29}	**0.402**	-0.272	**0.402**	0.064

Table 3.1: Results for all algorithms on synthetic spatial datain terms of the average MAE, RMSE, PSNR and SSIM over 30 runs, for randomly hidden and spatially clustered hidden values. All methods were ranked based on the number of other methods they significantly outperformed, established using a Wilcoxon signed-rank test ($\alpha = 0.05$). The method significantly outperforming the most other methods (ties allowed) has been marked **bold** in every column.

Algorithm	MAE		RMSE		PSNR		SSIM	
	random	clustered	random	clustered	random	clustered	random	clustered
Ordinary Kriging	3.863	9.375	**8.995**	31.48	-0.514	-0.558	0.122	0.0
Universal Kriging	3.944	9.822	**9.395**	37.977	-0.516	-0.568	0.121	0.0
Belief propagation	4.009	**4.048**	19.026	**21.308**	**-0.445**	**-0.437**	0.045	0.068
Non-spatial regression	8.259	9.022	27.973	32.89	-0.563	-0.563	0.007	0.011
SAR	6.688	7.39	20.76	28.583	-0.55	-0.551	0.055	0.051
MA	8.376	8.491	27.213	32.943	-0.562	-0.559	0.008	0.019
ARMA	6.836	6.664	21.208	24.934	-0.55	-0.538	0.049	0.048
CNN	5.604	8.131	30.985	45.152	-0.568	-0.569	0.002	-0.001
SD-MRP	3.833	5.496	11.924	**21.121**	-0.524	-0.53	0.155	0.122
WP-MRP	**3.48**	1.435×10^{47}	**9.13**	2.311×10^{95}	-0.514	-0.945	**0.189**	**0.14**

Table 3.2: Results for all algorithms on GDP data in terms of the average MAE, RMSE, PSNR and SSIM over 30 runs, for randomly hidden and spatially clustered hidden values. All methods were ranked based on the number of other methods they significantly outperformed, established using a Wilcoxon signed-rank test ($\alpha = 0.05$). The method significantly outperforming the most other methods (ties allowed) has been marked **bold** in every column.

Algorithm	MAE		RMSE		PSNR		SSIM	
	random	clustered	random	clustered	random	clustered	random	clustered
Ordinary Kriging	0.067	0.072	**1.176**	**1.368**	**0.481**	**0.555**	0.343	0.292
Universal Kriging	0.115	2.36	3.03	2936.144	0.466	**0.51**	0.452	0.273
Belief propagation	0.470	0.209	13.271	**1.335**	0.479	0.533	0.320	0.574
Non-spatial regression	0.058	0.055	**1.211**	**0.569**	**0.478**	**0.579**	0.558	0.498
SAR	0.067	0.07	**1.058**	**1.06**	**0.49**	**0.545**	0.403	0.382
MA	0.069	0.064	**1.169**	**0.936**	**0.48**	**0.571**	0.388	0.346
ARMA	0.07	0.066	1.178	**1.316**	0.481	**0.544**	0.401	0.449
CNN	0.235	883.369	7.479	6.192×10^3	0.407	0.182	0.796	0.005
SD-MRP	**0.036**	**0.038**	**1.175**	**1.258**	**0.481**	**0.552**	**0.941**	**0.939**
WP-MRP	0.244	0.24	8.327	7.956	0.4	0.448	0.785	0.776

Table 3.3: Results for all algorithms on COVID-19 trajectory data in terms of the average MAE, RMSE, PSNR and SSIM over 30 runs, for randomly hidden and spatially clustered hidden values. All methods were ranked based on the number of other methods they significantly outperformed, established using a Wilcoxon signed-rank test ($\alpha = 0.05$). The method significantly outperforming the most other methods (ties allowed) has been marked **bold** in every column.

Table 3.1. On this data, a fairly consistent pattern can be observed for all performance metrics: on randomly missing data WP-MRP performs best, whereas ARMA performs best on spatially clustered hidden data. It is not surprising that ARMA, as well as other regression-based methods, suffer less from missing data being clustered together since they are based on predicting values directly from features. It is more surprising that WP-MRP shows very extreme values for this type of missing data. Since SD-MRP does not suffer from the same problem, it seems that the problem lies in the weight prediction model $\mathcal{M}_w$, rather than being inherent to VPint. One possible cause for the behaviour on spatially clustered missing data may be that a mispredicted (high) weight will get disproportionately amplified with subsequent recursive calls where the target value is supposed to go up. Although these types of runs only seemed to happen on the synthetic and GDP datasets, it is a downside of WP-MRP, and one could consider constraining weights, or applying normalisation techniques, to alleviate the issue. Apart from these cases, there was no big difference between the results of randomly missing and spatially clustered missing data.

GDP per capita. The results for GDP data are shown in Table 3.2. In terms of MAE and SSIM, WP-MRP was the best performing method among all methods for randomly missing data. For spatially clustered missing data, while belief propagation performed better than SD-MRP and WP-MRP suffered from extreme values hampering its performance, both VPint variants still performed well in terms of SSIM. In terms of RMSE and PSNR, belief propagation performed best in most cases, though SD-MRP performed best together with belief propagation for spatially clustered missing data, and WP-MRP, OK and UK performed better in terms of RMSE on randomly hidden data. Also, worth noting is that the performance of

all methods was rather poor, with all methods achieving high error rates and low similarity scores. This may imply that it is hard to predict GDP based on spatial patterns alone (OK, UK, SD-MRP), while the map-based features were also not informative enough to make any worthwhile predictions (all other methods).

COVID-19 trajectories. The results for COVID-19 trajectories are shown in Table 3.3. On this dataset, SD-MRP is performing best out of all methods in terms of MAE and SSIM, though none of the methods was clearly better in terms of RMSE and PSNR than the others in terms of statistical significance. It is, however, unfortunate to see WP-MRP as one of the two only methods performing significantly worse than all others on this dataset in these metrics, despite a high SSIM compared to baseline methods. Since other methods using feature data (apart from CNN) perform better than Kriging, it seems unlikely that the map-derived features are the cause of WP-MRP not performing well on this dataset. Instead, it appears that they are more effective for directly predicting the COVID-19 incidence at a particular location, rather than the relationship between neighbouring locations. This may be caused by the COVID-19 grid being relatively sparse; propagating values from 0 is difficult to do with a spatial weight alone. Thus, for sparse grids with mostly 0 values, SD-MRP with its decay over distance may be more appropriate, whereas WP-MRP, which can increase or decrease values based on the weights that follow from feature data, may be more appropriate in cases where all cells contain values in a non-zero range.

3.6.3. OTHER PROPERTIES (CRQ2-CRQ5)

We now present the results of the experiments addressing the remaining research questions, CRQ2–CRQ5, exploring various properties of our proposed method.

Visual plausibility. An example of hidden synthetic data ($n = m = 50$) is shown in Figure 3.4, with a visual comparison between its reconstruction by the different methods. The reconstructed images caution against relying too much on mean absolute error, as all methods (apart from belief propagation and CNN) were able to reach a similar mean absolute error as WP-MRP or better (around 2.2 for random, 0.9 for clustered). However, our methods (and WP-MRP in particular) appear to capture the spatial characteristics of the original image, captured by the structural similarity index ($SSIM = 0.34$ for random, $SSIM = 0.69$ for spatially clustered), better than OK (which seemingly simply predicts the average value, $SSIM = 0.04$ for random, $SSIM = 0.65$ for spatially clustered). Compared to non-spatial regression, our method delivers less noisy interpolations, while also blurring less than ARMA. The results for belief propagation and CNN catch particular attention, as in these cases belief propagation vastly underestimated the values, whereas CNN overestimated them orders of magnitude higher than other methods. The latter

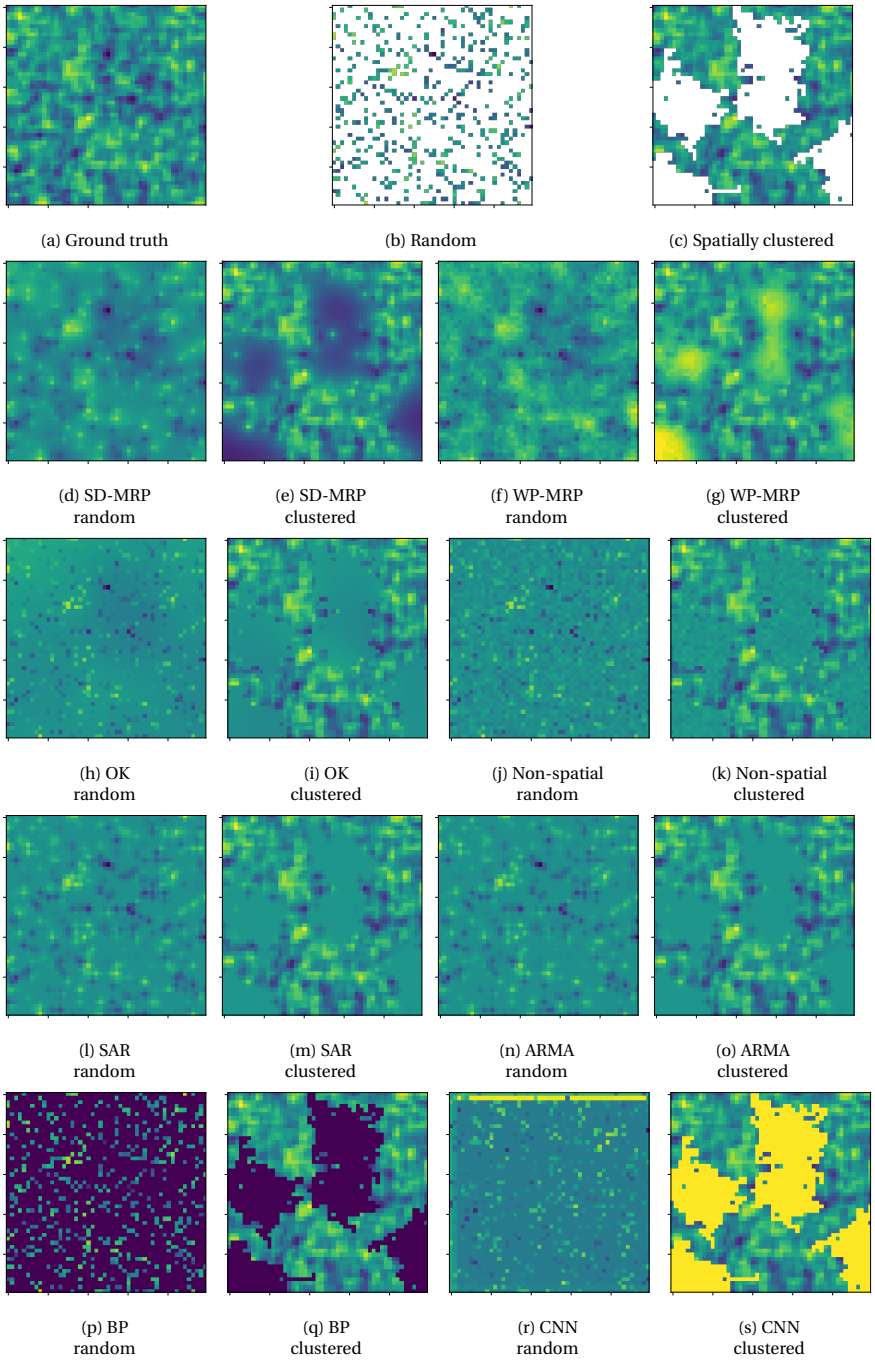


Figure 3.4: Example of synthetically generated spatial data (a), with random (b) and spatially clustered (c) missing data, where white pixels represent missing values in the data. Reconstructed images by SD-MRP (d,e), WP-MRP (f,g), ordinary Kriging (h,i), non-spatial regression (j,k), SAR (l,m), ARMA (n,o), belief propagation (p,q) and CNN (r,s) are shown in the lower rows of the figure. The results for universal Kriging and MA were highly similar to those of ordinary Kriging and ARMA, respectively, and are not shown here.

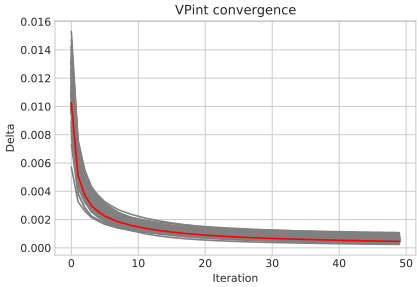


Figure 3.5: Convergence of WP-MRP over 100 runs on randomly hidden synthetic spatial data. The y-axis shows δ, or the amount of change from the configuration from one iteration to the next as a proportion of the mean value of the prediction grid. All individual runs were plotted in grey, with the mean δ values plotted in red. The algorithm showed stable convergence with low variability over 50 iterations.

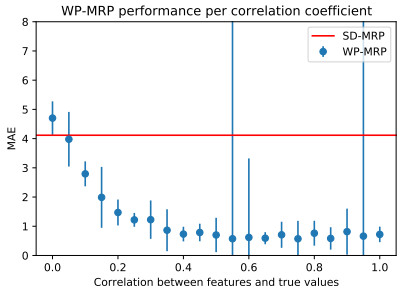


Figure 3.6: SD-MRP and WP-MRP performance on synthetic data for various settings of f. All datapoints were computed using the median and standard deviation of 30 runs per setting (SD-MRP is unaffected by features, and therefore constant). The extreme error bars at $f = 0.55$ and $f = 0.95$ also show the effect of WP-MRP producing extreme values.

implies that there may be a large risk of overfitting for the neural networks, due to the models being too complex for the limited amount of training data available.

Algorithm convergence (CRQ2). An example of the convergence of WP-MRP over iterations can be seen in Figure Figure 3.5. As the figure shows, as the algorithm iterates Equation 3.3, it converges to a stable configuration which we use for our predictions. Moreover, the variability of this convergence was fairly low, indicating that the running time of the algorithm will be relatively stable regardless of the situation. This example considered the convergence of WP-MRP on randomly hidden synthetic spatial data, but similar behaviour could be observed for SD-MRP, and on different datasets with spatially clustered hidden data. This includes the convergence of WP-MRP on spatially clustered hidden data for synthetic spatial data, where Table 3.1 earlier indicated that WP-MRP did not perform well.

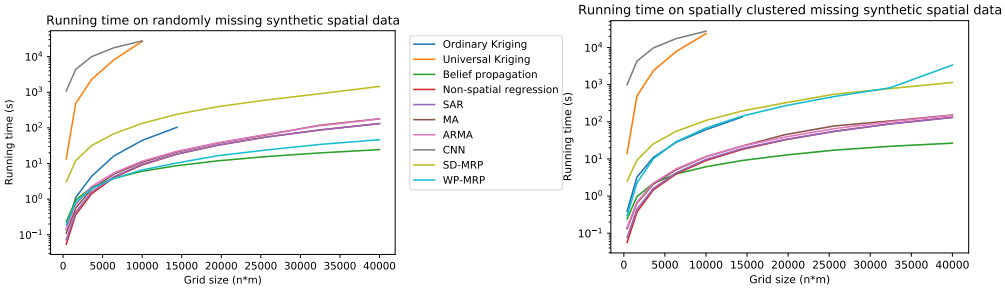
Generalising to spatio-temporal problems (CRQ3). Addressing CRQ3, to gauge whether our method could also be applied to 3-dimensional spatio-temporal problems, we ran an additional set of experiments on synthetic spatio-temporal data. The results of this experiment are shown in Table 3.4. Unfortunately, it appears

Algorithm	MAE		RMSE		PSNR		SSIM	
	random	clustered	random	clustered	random	clustered	random	clustered
Ordinary Kriging	2.741	3.12	12.025	15.714	0.373	0.362	0.001	0.011
Universal Kriging	2.67	3.118	11.465	15.824	0.375	0.361	0.008	0.002
Belief propagation	3.994	4.012	408.672	412.397	0.220	0.220	0.000	0.000
Non-spatial regression	2.516	2.538	9.871	10.039	0.382	0.381	0.001	0.001
SAR	2.635	2.451	11.369	9.588	0.376	0.383	0.046	0.069
MA	2.466	2.524	9.663	10.025	0.383	0.381	0.001	0.001
ARMA	**2.021**	**2.257**	**6.668**	**8.292**	**0.399**	**0.389**	**0.412**	**0.193**
CNN	3.245	4.293	16.646	204.488	0.364	0.344	0.0	0.0
SD-MRP	15.931	14.648	269.688	241.627	0.239	0.249	0.042	0.038
WP-MRP	7.332	8.296	81.403	117.111	0.29	0.275	0.046	0.047

Table 3.4: Results for all algorithms on synthetic spatio-temporal data in terms of the average MAE, RMSE, PSNR and SSIM over 30 runs, for randomly hidden and spatially clustered hidden values. All methods were ranked based on the number of other methods they significantly outperformed, established using a Wilcoxon signed-rank test ($\alpha = 0.05$). The method significantly outperforming the most other methods (ties allowed) has been marked **bold** in every column.

that our proposed method does not (yet) generalise well to 3-dimensional problems, as both VPint variants were the worst performing out of all methods. However, other modifications adapting VPint to the spatio-temporal domain may be more successful. Interestingly, on this synthetic dataset, ARMA performed best across the board – one might have expected a more inherently spatio-temporal method, like Kriging, would have performed better. Given this, it may be the case that the spatio-temporal version of VPint performs badly not due to an inherent problem with the method, but rather a lack of exploitable patterns in the temporal dimension of the data. Whereas Kriging methods will tend to give lower weights to non-informative variables, in its current form, our method weights all dimensions equally, meaning that a non-informative dimension would harm the performance of our method rather than helping it. In the synthetic data we used in our experiments, independent spatial data was generated for all time steps individually, after which temporal autocorrelation was simulated using Equation 3.22. As every time step had two essentially independent neighbours, the autocorrelation between the two may have cancelled out one another on some cells, leading to a diminished performance. As a result, the model may perform better on real-world spatio-temporal data under favourable conditions. However, as this is outside the scope of this work focused on spatial interpolation, further research would be necessary to establish exactly what those favourable conditions would be.

Performance of WP-MRP compared to SD-MRP as a function of feature correlation (CRQ4). To address CRQ4, we ran an additional experiment on synthetic data with settings of the feature correlation coefficient f in Equation 3.23 ranging between 0.05 and 1.0 in steps of 0.05. Figure 3.6 compares the performance of the two methods based on MAE as a function of feature correlations. The figure shows



(a) Random missing data

(b) Spatially clustered missing data

Figure 3.7: Running time in seconds of the full pipeline of different methods as a function of grid size on synthetic spatial data. The results for universal Kriging and CNN were cut off early due to hitting time-out thresholds. Meanwhile, although the running time of ordinary Kriging was not too different from other methods, its memory usage became prohibitively large and exceeded its allotted resources (13GB).

the error distribution acquired from 30 runs. As expected, Figure 3.6 shows that WP-MRP performs better than SD-MRP for high values of f, and conversely, SD-MRP appears more successful for low feature-target correlations. However, there appear to be diminishing returns for higher f after 0.4, and already at a correlation of 0.1 WP-MRP performed better than SD-MRP on the synthetic data. In conclusion for CRQ4, this experiment shows that WP-MRP leverages spatial features to perform better than SD-MRP in situations where the features are sufficiently informative.

Scaling to larger datasets (CRQ5). Addressing CRQ5, we ran a scalability analysis by running every algorithm once on synthetic data for grid sizes ranging from 20 to 200 (height and width) in steps of 20. The results of these experiments, based on the total running time of methods (including training, if any, but excluding NAS, SMAC and other algorithm configuration as they are optional) can be seen in Figures 3.7a (random) and 3.7b (spatially clustered). The figures show that SD-MRP, while faster than CNNs, does not scale well to larger datasets, and that WP-MRP scales similarly compared to non-spatial regression, SAR, MA and ARMA. This tells us that the iterative MRP-derived update rule likely does not account for a large portion of the running time; instead, it appears that the auto-sklearn training procedure, much like in the case of non-spatial regression and SAR, MA and ARMA is the main bottleneck for WP-MRP. The reason, then, for SD-MRP to scale poorly, would be the random search-based subsampling procedure used to find an optimal static discount γ explained in Section 3.5.4.2.

We can also see in both figures that universal Kriging scales very poorly to larger

datasets; in fact, its runs timed out after grids of the size 100×100. While CNN was slightly less affected than UK by the increasing size, its running times were still exceedingly high, and likewise hit a time-out threshold after 100×100 grids. Similarly, while the running times of ordinary Kriging were similar to those of other methods, its memory usage became prohibitively large by exceeding its allotted 13GB at 120×120 grids. Thus, this experiment showed another weakness of GPs, namely their high memory usage, which is also detrimental to their scalability. Newer GP methods, like local approximation GPs [101], may scale better in terms of running time and memory usage by using local approximations, although this may come at the expense of a decreased ability to capture global information.

In conclusion for CRQ5, our methods scale better than Kriging to larger datasets, on par with non-spatial regression, SAR, MA, and ARMA, though SD-MRP did take longer than these methods on randomly missing data. Generally, our methods use substantially less memory than ordinary Kriging and universal Kriging.

3.6.4. HIGH-LEVEL SUMMARY

Tables 3.1 through 3.3 show the competitive advantage of the VPint variants, in terms of MAE and SSIM. For randomly missing data, the two VPint variants together performed better in terms of MAE than baseline methods on all 3 spatial datasets, although individually both methods only performed better than all baselines on 2 out of 3 datasets. WP-MRP performed better than all other methods on synthetic and GDP data, though SD-MRP also performed better than baseline methods on the GDP data, and SD-MRP performed better than all other methods on the COVID-19 dataset. In terms of SSIM, the two VPint variants together were again the best performing methods on all 3 datasets, where WP-MRP again performed best on the synthetic (though tied with SAR and ARMA, with SD-MRP following one ranking lower) and GDP datasets, and SD-MRP also performed better than the baseline methods on GDP data. SD-MRP performed better on the COVID-19 dataset, where WP-MRP was the third best performing method after CNNs. In terms of RMSE, the results were less consistent, with no method clearly outperforming the others across all datasets, and as expected, the rankings for PSNR and RMSE were almost always the same. This difference implies that, while the VPint variants will often perform better *on average*, when they do fail to produce good results, the errors will be more extreme than those of baseline methods. This was seen especially clearly in runs where WP-MRP obtained error values orders of magnitude higher than all other methods.

On spatially clustered missing data, in terms of MAE, SD-MRP still performed best on the COVID-19 dataset, but was outperformed by belief propagation on GDP data and by ARMA on synthetic data. WP-MRP also failed to significantly

outperform baseline methods on any of the datasets for this type of missing data. However, in terms of SSIM, both VPint variants performed significantly better than all baselines on GDP and COVID-19 data, and SD-MRP tied with SAR and ARMA for synthetic data. Since it seems that SD-MRP preserves the spatial structure of spatially clustered hidden data better than baseline methods on all 3 datasets, and WP-MRP did so on 2 out of 3 datasets, we conclude that VPint would be a better option for this type of missing data if the spatial structure of the interpolations is important. Interestingly, SSIM is higher for all methods for spatially clustered data; this is likely caused by this type of missing data being considered a substantial structural element, thus affecting SSIM less than random missing data.

Regarding our additional experiments, we found that the convergence of VPint tends to progress smoothly and has very little variance between runs (as seen in Figure 3.5). Table 3.4 shows that, in its current form, our method does not yet generalise well from the spatial case to the spatio-temporal case. Figure 3.6 shows that WP-MRP will perform better than SD-MRP starting from a feature correlation coefficient f of around 0.1 for synthetic data, implying that a high correlation between features and targets is not required for the feature data to have added value to the method. Finally, Figure 3.7 showed favourable scalability of our proposed method, particularly compared to Kriging.

3.7. CONCLUSION AND FUTURE WORK

In this chapter, we proposed VPint, a value propagation-based method for spatial interpolation, establishing a system-oriented perspective. To this end, we introduced two variants of our interpolation method (SD-MRP and WP-MRP), the latter of which exploits spatial features describing the characteristics of the grid. In our experiments on gridded GDP and aggregated COVID-19 data, VPint was found to perform significantly better than baseline methods in terms of mean absolute error and structural similarity on randomly missing data in 3 datasets, and 2 out of 3 datasets for spatially clustered missing data.

Overall, whether VPint is the appropriate choice of algorithm appears to depend on the type of data in question, and the goals of the user. In the common case where a low error rate is the objective, particularly in a way that preserves the spatial structure of a grid, VPint (and especially WP-MRP) will generally be the best option for randomly missing values. Despite the advantages offered by VPint, if a practitioner is looking for a method that does not suffer from outliers of particularly bad predictions, and is willing to accept higher average errors as a result, other methods, such as Kriging or ARMA, may be better options.

On spatially clustered missing data, SD-MRP is usually still a better option than other methods, but they are more competitive on this type of missing data.

While still highly effective, the slightly reduced competitive advantage of VPint on spatially clustered missing data suggests a relative sensitivity to biased missing data distributions and low spatial sampling rates. When using SD-MRP, the algorithm slowly converges to a mean value prior as the distance from known values increases. Meanwhile, when using WP-MRP, the errors of the weight prediction model add up over distance to introduce increasing amounts of uncertainty into predictions. This may limit the applicability of VPint, in its current form, for problem instances where data gaps are large (e.g., hundreds of cells), while encouraging its use particularly in use cases with relatively smaller gaps or lower resolutions.

In future work, it would be interesting to focus on exploring the performance of our methods on other real-world datasets, particularly when using other sets of features not derived from map data. Furthermore, we see value in further analysis of the spatio-temporal variant of VPint, focussed on the circumstances under which it will perform well. Alternatively, a different approach to spatio-temporal interpolation could be a temporally layered version of WP-MRP, using a representation similar to the tensor-based approach adopted by Corizzo et al [118]. Such an approach would eliminate the need for explicit feature data, and would instead use known values at different time steps as features to derive spatial weights. This type of approach may well be worth exploring.

Finally, in the next chapter, we will adapt the VPint method to make it applicable to missing data in satellite imagery, particularly due to cloud cover, which can cause the large data gaps described above that VPint is currently less suited for. As explained in Section 2.1.3, the need for cloud removal techniques, given that up to 70% of Earth is covered by clouds at any time, is great; existing techniques are limited and often either difficult to apply for non-experts in deep learning, or fail to produce actionable new information (such as simply predicting a mean, or replacing cloudy pixels). A data-driven, universally and easily applicable method able to fill in cloud cover (or gaps caused by faulty sensors), informed by the highly correlated features of previous imagery at the same place, could increase the availability of data and therefore the efficacy of the many high-impact methods dependent thereon to a large extent. This will be the focus of our work in Chapter 4.

4

TRAINING-FREE CLOUD REMOVAL USING VALUE PROPAGATION INTERPOLATION

In the previous chapter we have addressed RQ1 by introducing VPint, a spatial interpolation method capable of interpolating spatial data, such as in-situ measurements for parameters. In this chapter[1], we will address RQ2: *How can we effectively and easily interpolate unpredictable, spatially clustered missing data in Earth observation imagery?* Towards this end, this chapter will cover the second component of Challenge 1, namely missing data in satellite imagery, with a particular focus on cloud cover. Our proposed method, VPint2, modifies and extends the VPint algorithm to be applicable to this image processing-like remote sensing task. As will become apparent in this chapter, cloud removal (and similar EO imagery interpolation tasks) comes with particular challenges such as high contrast, temporal heterogeneity and exploding values, requiring extensive adaptations of the VPint algorithm.

[1] The contents of this chapter are based on the journal article:
Laurens Arp, Holger H. Hoos, Peter van Bodegom, Alistair Francis, James Wheeler, Dean van Laar, and Mitra Baratchi. (2024). *Training-free thick cloud removal for Sentinel-2 imagery using value propagation interpolation.* ISPRS Journal of Photogrammetry and Remote Sensing, 216:168–184. Elsevier.
https://doi.org/10.1016/j.isprsjprs.2024.07.030

4.1. INTRODUCTION

Remote sensing data, such as the data obtained constantly from Earth observation satellites, is of tremendous importance in monitoring the health of our planet. However, when working with remote sensing data, data processing pipelines that could otherwise produce excellent results are often challenged by clouds obscuring parts of a satellite image, as explained in Section 2.1.3. In some cases, these cloudy images are omitted entirely, even to the point of on-board hardware and software solutions being developed for satellites to avoid sending cloudy data back to Earth [136, 137]. Alternatively, cloud-free images are produced by a combination of cloud masking and mosaicking cloud-free pixels from a previous image onto the cloudy pixels of a target image [138, 139]. Although such an approach allows an application to accept the input image, pixel values from dynamic processes get outdated relatively quickly, and finding recent cloud-free images can be challenging. This can be a problem for tasks such as vegetation monitoring or the mapping of extreme events (e.g., floods or fires). Therefore, accurate and up-to-date estimations for cloudy regions would be much preferred, particularly in such dynamic environments.

Creating these cloud-free estimations for remote sensing images can be a challenging task. Environmental factors like the sun's azimuth and zenith angles, atmospheric conditions, vegetation, and the landscape change over time. A cloud removal algorithm would need to account for these changes, which is difficult, considering the variability of the temporal distance to the last known cloud-free image. Similarly, the environment may have evolved in unpredictable ways, such as by extreme events or human activity. In recent years, deep learning-based cloud removal methods have shown strong performance compared to traditional methods. However, for downstream users, pre-trained (deep) neural networks may require very precise combinations of input conditions such as the sensor, resolution and preprocessing, which may not be feasible for their use case, while developing new models for specific use cases is generally not trivial because new architectures, appropriate training configurations and (potentially large-scale) training datasets would be required.

In this chapter, we present a method to address these challenges by employing a technique that propagates the information in cloud-free pixels of the same image rather than using old pixel values. Specifically, we present a new cloud removal algorithm building upon our previously proposed spatial interpolation algorithm value propagation interpolation (VPint) from Chapter 3. We have extended the algorithm to be suitable for the reconstruction of multispectral imagery. Our method uses the previously sensed imagery from the same time series as a feature dataset to inform the interpolation algorithm on the *spatial structure* of the underlying re-

gion of interest. It uses this structure to interpolate the reflectance values from the *current*, up-to-date image. In doing so, the current environmental conditions will also be propagated, rather than attempting to estimate these *a priori* to correct for them.

Applying VPint to optical remote sensing imagery introduces three main challenges: cloud removal being a remote sensing image processing rather than a general interpolation problem, temporal heterogeneity and exploding values: firstly, the original VPint algorithm leveraged machine learning models to predict the intensity of spatial autocorrelation between two spatially neighbouring points. However, in cloud removal, the intricate textures, transitions and objects in the image must be reproduced exactly, requiring a more precise representation of the spatial structure. Secondly, different sets of pixels may change in different ways between the feature dataset and target image, such as a (dynamic) crop field next to a (static) road. The relationship between these objects will change over time, introducing inaccuracies in the reconstructed target image. We refer to this problem as *temporal heterogeneity*. Thirdly, faulty pixels, solar glint or transmission errors can introduce erroneous values into the feature data, which can result in unrealistically large pixel value predictions that get propagated throughout the reconstructed image. We refer to this problem as *exploding values*. To address these issues, we propose VPint2, which incorporates a new method for computing spatial weights and includes extensions to the original VPint algorithm that alleviate problems caused by these phenomena: identity priority and elastic band resistance. Although VPint2 is aimed at improving VPint's applicability to optical remote sensing data, it is likely that it will similarly enjoy improved performance in other application areas where similar challenges arise, particularly in image processing. Through our experiments, we aimed to gauge the effectiveness of VPint2 as a cloud removal algorithm compared to existing methods, and to investigate under which conditions its performance is particularly strong.

Our contributions presented in this chapter are as follows:

- We propose a novel cloud removal method, leveraging the spatial structure from previously sensed imagery to propagate the non-cloudy values of the up-to-date cloudy image. Our method does not require a training phase and can be easily applied to any type of land surface data, requiring no additional data compared to pixel substitution approaches.

- We extended the spatial interpolation algorithm VPint to create VPint2, which modifies the algorithm to be applicable to remote sensing image processing problems, and features two enhancements we dub *identity priority* and *elastic band resistance*, improving its performance and its applicability to remote

sensing datasets. We include an auto-adaptation mechanism to allow VPint2 to adapt its configuration to specific patches and bands.

- We created a benchmark dataset of 20 matched (target–cloud mask–temporal features) sets of top-of-canopy Sentinel-2 imagery, called SEN2-MSI-T. Unlike existing benchmarks, the true images are available as ground truth, as the clouds are derived from a different image, and the features are available at various specific time intervals. This allows for a better evaluation of methods, and the results from our experiments show that typical evaluation approaches using a recent cloud-free acquisition as ground truth can be problematic.

- We tested our method on SEN2-MSI-T and the existing SEN12MS-CR-TS benchmark dataset against mosaicking (temporal replacement), automated-machine-learning-based regression, similar pixel interpolation and neural-network-based approaches. Our experiments demonstrate that our method performs better than competing methods in all 20 conditions we tested in our main experiments, and in 17 out of 20 conditions in our experiments for Level 1C data.

4.2. PROBLEM STATEMENT

Cloud removal can be formalised as a general spatial interpolation problem. Let image $\mathbf{O}$ denote the matrix representation of the original input image with clouds to be removed. This image consists of pixels $o_{ij} \in \mathbf{O}$, where i the row index, and j is the column index of the pixel, corresponding to the spatial position (i, j). Since all spectral bands in optical images are affected by thick clouds, cloud removal methods will typically need to be applied independently to all bands. In the context of parameter estimation, one pixel would be a vector $\mathbf{x} = [o_{ij}^1, o_{ij}^2, ..., o_{ij}^b]$ containing o_{ij} for all b bands. We use $\mathbf{T}$ to denote the matrix of the corresponding true (ground truth) cloud-free image, consisting of pixels $t_{ij} \in \mathbf{T}$, which would be unknown in practice, and $\mathbf{F}$, consisting of pixels $f_{ij} \in \mathbf{F}$, as the matrix of the cloud-free reference (feature) image obtained at some point in time prior to $\mathbf{O}$. Finally, we use $\mathcal{C}$ to denote the set of cloudy target image pixels o_{ij} for $\mathbf{O}$ (cloud mask). Our aim is to find a model $\mathcal{M}_{cloud}(\mathbf{O}, \mathbf{F}, \mathcal{C})$, taking the target image, feature image and cloud mask as input, and generating a predicted cloud-free image $\hat{\mathbf{T}}$ with pixels $\hat{t}_{ij} \in \hat{\mathbf{T}}$ resembling $\mathbf{T}$ as closely as possible. The problem thus becomes to find:

$$\mathcal{M}^*_{cloud} \in \underset{\mathcal{M}_{cloud}}{\operatorname{argmin}} \mathcal{L}(\mathbf{T}, \mathcal{M}(\mathbf{O}, \mathbf{F}, \mathcal{C})) \tag{4.1}$$

Here $\mathcal{L}$ is the loss function of interest (for example, mean absolute error).

4.3. RELATED WORK

Given its importance to downstream remote sensing tasks, cloud removal in optical satellite data is of significant interest in the research community. Generally speaking, cloud removal must be guided by some type of information complementarity, which may be *spatial*, *multi-modal*, *temporal*, or a mixture of these. In addition to this, there are also cloud removal methods operating on the spectral domain to remove thin clouds (which, due to partial transparency, retain some surface information) [140, 141, 142, 143, 144, 145, 146, 147]. However, since we aim to remove all types of clouds along with their shadows, we do not consider this type of method further in this section.

Spatial methods rely on patterns within the cloud-free regions of an image to reconstruct cloudy regions. Much of the work we will refer to in this section contains a spatial component, for example, through the use of convolutional neural networks (CNNs) or the selection of suitable nearby pixels. However, most of these methods will also exploit other types of information complementarity. In contrast, general-purpose spatial interpolation techniques can also be used for single-image cloud removal by considering cloudy pixels (and cloud shadows) as missing data. This approach has been explored for other types of missing data in remote sensing imagery (sensor faults) by Zhang et al. [148]. However, many interpolation methods suffer from poor scalability, and Shen et al. [42] found that interpolation approaches are primarily effective at filling small gaps, such as the Landsat ETM+SLC-off dataset [149]. In the case of cloud removal, clouds can cover relatively large parts of an image which, combined with the high resolution of the imagery, results in large gaps, for which interpolation methods have not been popular nor especially successful so far.

Multi-modal methods exploit the information complementarity between different sensors, notably synthetic aperture radar (SAR), which penetrates cloud cover, to reconstruct cloud-free images. One of the most prominent examples of this type of cloud removal is DSen2-CR by Meraner et al. [40], which is based on convolutional neural networks (CNNs) and leverages SAR data. Xu et al. [150] proposed global-local fusion approaches to minimise performance degradation due to speckle noise and the domain gap between optical- and SAR data. Han et al. proposed a transformer-based approach for SAR-optical data fusion-based cloud removal [151], and Liu et al. proposed an attention-based network fusing Sentinel-2 and Sentinel-3 data sources [152]. General adversarial network (GAN) methods performing cloud removal using SAR-optical data fusion include the work by Xu et al. [153] and Darbaghshahi et al. [154], whereas Jing et al. proposed a method leveraging denoising diffusion [155]. Fusing optical data with other data sources, particularly SAR data, can be a challenging problem due to temporal shift, the noisy

nature of SAR measurements, incomplete or non-overlapping spatial coverage at desired time steps, and the complex preprocessing pipelines SAR data typically require.

Multi-temporal methods exploit the temporal information complementarity of satellite imagery to gap-fill missing data. Here we differentiate between multi-temporal methods, which may exploit temporal information in a variety of ways, and the specific case of *time-series modeling* methods, which operate on a consistent time-series of images. A commonly used multi-temporal approach is mosaicking. Typically, a user would use cloud detection methods [156], such as s2-cloudless [157] or SEnSeIv2 [29], to automatically detect cloudy pixels, fetch the most suitable non-cloudy pixel from past imagery (based, for example, on temporal distance), and mosaic these non-cloudy pixels onto a target image. We refer to this approach as *temporal replacement*, for which recent examples of practical use of this type of method include work on downstream tasks such as ecological monitoring [158] and (tree extent) mapping [159]. More sophisticated mosaicking approaches may account for changes in atmospheric conditions, solar azimuth and zenith angles, and other potentially confounding processes [138], or incorporate histogram matching [139, 160]. The accuracy of the reconstructed images will be greatly dependent on the availability of suitably recent cloud-free images. Pixel-wise regression models can also be used to directly predict the pixel values of the target image using pixel values in the reference image as features. Due to the multicollinearity likely to be present in the different bands of the feature image, partial least squares (PLS) approaches tend to be preferred over ordinary least squares (OLS) models in remote sensing applications, as they compute inherently independent components [161, 162, 163]. Some approaches, such as the CHAIN and CROSS models proposed by Fischer et al. [164], use graphical and probabilistic models instead of grid-based spatial statistical models. Some methods, such as (M)NSPI [10, 165], WLR [166], STMRF [167], CLMP [168] and STWR [169], use temporal information to predict reconstructed pixel values from a local spatial neighbourhood of matching pixels.

Time-series modeling methods are a specific case of multi-temporal methods in which the full time-series of (possibly cloudy) regular temporal acquisitions by satellites is exploited for information complementarity. Zhu et al. [170] proposed the use of three time-series models, at varying levels of complexity, to predict missing pixel values in time-series of Landsat images, wheras SSTC-CR by Zheng et al. [171] leverages tensor decomposition to model various relationships between the spatial, spectral and temporal domains of the time-series of satellite data. More recently, multiple neural network-based approaches have been proposed for multi-temporal cloud removal, often incorporating multi-modal data as an additional component. UnCRtain-TS by Ebel et al. [41] combines a multi-

temporal approach with SAR-optical data fusion, and supports uncertainty estimates on its predictions. Zhang et al. [172] applied a CNN on fused temporal features, Zhang et al. [173] applied a CNN on decomposed tensors, Zhao et al. [174] leveraged diffusion models for cloud removal on time-series data, and Zou et al. proposed a fast diffusion approach using SAR data [175]. Stucker et al. [176] used convolutional layers to encode and decode spatial information, while performing temporal attention on the resulting latent representations of individual time steps. Multi-temporal approaches are typically less applicable to scenarios where only one or a few target images are relevant to the user, or when there are large or inconsistent temporal gaps in data.

In the following, we address weaknesses in the existing work by proposing a novel cloud removal method called VPint2, which extends the value propagation interpolation (VPint) algorithm of Chapter 3. Interpolation methods, due to the weaknesses of existing methods in gap-filling large clusters of missing data, have not been explored much for the purpose of cloud removal. Our proposed method, overcoming the weaknesses of existing interpolation methods, therefore offers a novel branch of research in cloud removal as a multi-temporal (single reference image) interpolation method.

4.4. METHODS: REVISITING VPINT

The VPint algorithm[2] proposed in Chapter 3 was aimed at general spatial interpolation problems. When applied to cloud removal, given a cloudy target image and a previously sensed non-cloudy image as features, it returns a cloud-free reconstructed target image. The process is applied independently for all spectral bands, thus being robust to the typically diverse spatial patterns in different bands of remote sensing imagery, and offering a considerable potential for parallelisation for imagery with a large number of spectral bands. The general workflow for applying our proposed VPint2 method to cloud removal is shown in Figure 4.1.

The main benefit of using VPint for cloud removal stems from its support for location-specific data-driven weights. In the original VPint algorithm, these weights could be predicted from other, related variables using machine learning models. This allowed the algorithm to make relatively smooth predictions, based mainly on spatial autocorrelation. In optical satellite data, images have sharp edges, textures and other challenging, abrupt changes, calling for a different approach to computing spatial weights. Therefore, we propose a modification to the core of VPint to directly compute exact spatial weights at runtime, without the need of an explicit optimisation or training procedure, making it a training-free method. We

[2] The code for VPint2 can be found on GitHub at https://github.com/ADA-research/VPint2

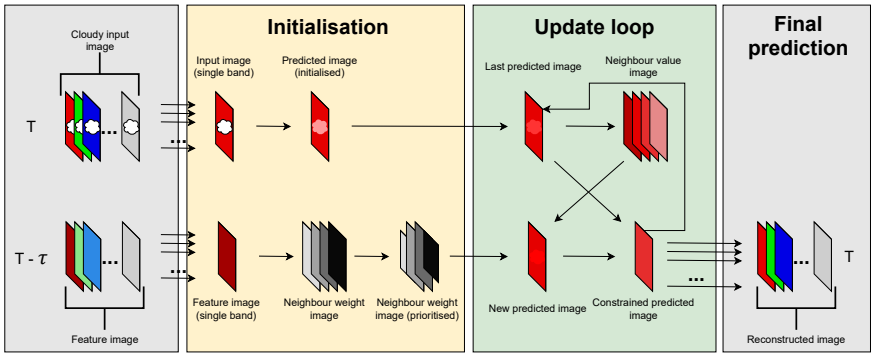


Figure 4.1: General workflow for applying VPint2 to cloud removal problems. First, the input- and feature images are split up into individual bands (for which we show one example in the red band). Next, for every band, the predicted image is initialised to fill cloudy pixels with the mean value of the image (other initialisations are possible), and the spatial weights for the neighbours of every pixel are computed and stored in a 4-channel image, where every channel represents a direction. In this image, for every pixel, the channels contain weights by which a neighbouring value would need to get multiplied to form the pixel's value in the feature image. If identity priority is used, these weights are then refined into a re-weighted neighbour weight image with lower impact for more extreme weights. After this, in the update loop, a new image is computed iteratively by multiplying a neighbour value image (with predicted image values for neighbouring pixels in channels corresponding to those of the weight image) with the weight image. If using elastic band resistance, the growth of values in the resulting new predicted image is constrained by comparing the previous predicted image to that of the current iteration. After auto-termination (or a specified maximum number of iterations has been reached), the most recent (constrained) predicted image becomes the predicted image for this band. All bands are then combined again to form the cloud-removed output of VPint2.

will refer to this updated algorithm as VPint2.

In optical satellite data, if two images are taken of the same area at different times, the spatial structure of the land surface of two co-registered images sensed at different times would remain (mostly) static. For example, a residential suburb may change in hue as time passes and seasons change (dynamic values), but whether it is shiny in the sun or snowed over in winter, the similarity between the pixels within houses, trees and gardens the field will remain relatively high (static spatial structure). This means that the weights within these structures should remain close to 1 (a weight of 1 between a pair of pixels signifies that they are identical). Similarly, the weights on the border can be expected to be further removed from 1, as the neighbouring pixels on the border will be more dissimilar from one another, requiring a transition for the values being propagated.

This intuition gives rise to the notion of *objects* in an image. Pairs of pixels belonging to the same object (*same-object* pairs) will have weights close to 1, and

pairs of pixels not belonging to the same object (*different-object* pairs) will have weights further removed from 1. These objects need not be explicitly defined (i.e., no object detection algorithms are necessary). Instead, they are contained in the spatial weights derived from the reference image.

This consistency of the spatial structure over time is the property we exploit with VPint2, by assuming temporally static spatial relationships for temporally dynamic values, and feeding the reference image as a feature set to the algorithm. The manner of deriving weights from a feature image can vary, and has a high impact on the behaviour of the method. The predictions from a machine learning model, as used in the original VPint algorithm, could not easily model the strong and abrupt changes in remote sensing image processing tasks. However, when using very precise weights with sufficient variability to be applicable to images, mistakes and errors can also have a larger impact on performance. Therefore, one of the challenges in applying VPint to remote sensing (and possibly general image processing) tasks is to use an approach for deriving spatial weights that is both exact and reliable, while mitigating the risks of large errors that exact, non-smoothed weights entail.

To address this problem, we can leverage the property of satellite data automatically revisiting the same area at specific time intervals. Although the latest cloud-free reference image could be months in the past, especially given the temporal autocorrelation of cloudy and rainy weather, we can exploit these reference images in VPint2 by extracting the spatial structure of a location, using the past reference image as a feature image to compute highly accurate spatial weights. A simplified illustration of how this spatial structure is used can be found in Figure 4.2.

Concretely, we instantiate the predicted image $\hat{\mathbf{T}}$ with pixels $\hat{t}_{ij}$ by copying non-cloudy pixels $o_{ij} \notin \mathcal{C}$ from $\mathbf{O}$, and initialise $\hat{t}_{ij}$ for all cloudy pixels $o_{ij} \in \mathcal{C}$ as the mean value of $\mathbf{O}$ (other initialisation approaches are possible). Let $N(i,j)$ denote the set of neighbouring positions (i',j') for a given position (i,j). With τ denoting the current iteration, we iterate the following update rule:

$$\hat{t}_{ij}^{\tau+1} = \begin{cases} o_{ij}, & \text{if } o_{ij} \notin \mathcal{C} \\ \frac{1}{|N(i,j)|} \cdot \sum_{(i',j') \in N(i,j)} \gamma_{ij}^{i'j'} \cdot \hat{t}_{i'j'}^{\tau}, & \text{otherwise} \end{cases} \tag{4.2}$$

The update rule of Equation 4.2 contains a number of important elements. First, all non-cloudy pixels (that is, $o_{ij} \notin \mathcal{C}$) merely take on the static value of the input image. For cloudy pixels, at every iteration, a local prediction is computed, using the weighted average of predicted values of neighbouring unknown pixels (or static known values where available). Here $|N(i,j)|$ denotes the number of neigh-

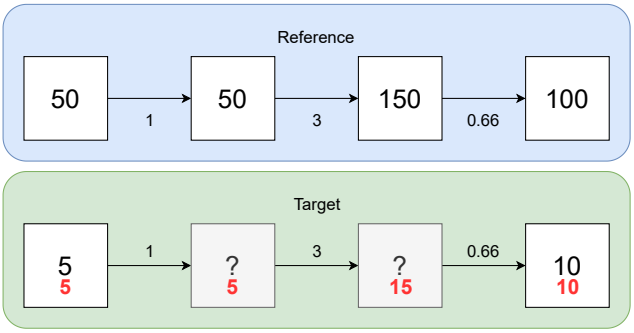


Figure 4.2: A basic, one-dimensional, unidirectional example illustrating the process of VPint2. The middle two cells in the target image are unknown, whereas the reference image is fully known. Although the values themselves lie in a different range (multiplied by 10 in this simplified example), the relationships between neighbouring cells are the same in this example. By computing weights from the reference image, we can interpolate the unknown values in the target image. In reality, interactions would bi-directional, and the problem would have two spatial dimensions.

bours to pixel $\hat{t}_{ij}$ (in practice 4 for all pixels apart from the image edges), and $\gamma_{ij}^{i'j'}$ is the spatial weight between the pixels at positions (i', j') and (i, j). This weight is computed from the corresponding pixels in feature image $\mathbf{F}$ as:

$$\gamma_{ij}^{i'j'} = \frac{f_{ij}}{f_{i'j'}} \tag{4.3}$$

Thus the value of every cloudy pixel in $\hat{\mathbf{T}}$ is determined by the values of its local neighbours, which, if cloudy, are themselves determined by their neighbours. At every iteration and recursive step, the neighbouring values are multiplied by a spatial weight derived from the feature image, allowing the algorithm to incorporate complex structures, textures and variability within subsets of the image.

By iterating Equation 4.2, pixel values are updated repeatedly, anchored by non-cloudy target values that are propagated following the spatial structure given by $\mathbf{F}$ (through γ), until an equilibrium configuration is reached.

4.4.1. VPINT2 PROPERTIES

VPint2 has a number of desirable properties that existing cloud removal methods do not yet offer. First, VPint2 estimates the current state of the measured quantity, as opposed to methods merely copying previous information (which was already known). Second, it offers the advantage of not needing any training, thus avoiding the problems of methods that need to either attempt to train one general model

applicable to all cases, extrapolate from a specialised model that does not generalise well, or train on a prohibitively small training set of non-cloudy pixels from the same image. Third, the results and inner workings of VPint2 can be understood by analysing the reference image in combination with the update rule of Equation 4.2.

On the other hand, as an interpolation method, VPint2 requires at least one non-cloudy pixel in the input data, otherwise it would simply reconstruct the feature image. This might make it less suitable for cloud removal in smaller patches, where the probability of all pixels being cloudy is higher.

In terms of computational cost and efficiency, the running time of VPint2 will depend on its implementation. However, the implementation-independent efficiency can be approximated by drawing a parallel with neural networks, which are often described based on the number of parameters in the network. Since both neural networks and VPint2 are based on matrix multiplication, the computational effort mainly stems from the amount of multiplications and matrix elements that must be multiplied. In the case of VPint2, the time complexity to run on a single band is $\mathcal{O}(e \cdot h \cdot w \cdot 4)$, where e is the number of iterations or epochs for which VPint2 will be run (typically around 20), h is the height of the input image, and w is the width of the image. For example, if we run VPint2 for 100 iterations on a 100×100 image, resulting in 4 000 000 multiplications, the entire pipeline of VPint2 would be the equivalent of running *inference only* on a 4 million parameter neural network, while not requiring a prior training step. In practice, VPint2 will generally perform more frequent, smaller matrix operations compared to a typical neural network with a similar number of parameters, resulting in a slower running time. On the other hand, the multispectral nature of optical satellite data allows for great opportunities in parallelisation over bands, since VPint2 considers these bands independently. Therefore, we have extended the original VPint algorithm with a multiprocessing setup for Earth observation imagery, resulting in a substantial speedup of about 60% to 70% compared to the original algorithm (for empirical results, see Figure 4.7c). In the future, combining the parallelisation with GPU-accelerated matrix computations may speed up the algorithm even further.

4.4.2. FURTHER ENHANCING VPINT2 FOR REMOTE SENSING DATA

Applying VPint2 to remote sensing imagery comes with particular challenges, some of which may also be encountered in general image processing problems. In particular, VPint2 will perform worse when i) objects in the images changed over time between the feature and target images in different ways (e.g., one stays constant while another changes hue), which we will refer to as *temporal heterogeneity*, and ii) sensor faults or other inaccuracies are present in the feature set, resulting in

extremely large weights (and extremely large values that get propagated further), which we refer to as *exploding values*.

To illustrate the problem of temporal heterogeneity, recall the concept of *objects* introduced in Section 4.4. In the case of removing clouds from optical imagery, the *within-object* relationships (weights close to 1) will typically be easy to exploit, whereas *between-object* relationships will be less reliable. For example, a road next to a forest will remain mostly static throughout the seasons, while the forest may be shedding and gaining leaves over time. This means that the spatial weights between the road and the forest computed from the reference image (for example, one from a summer, when the forest was full of green leaves) will no longer apply to the new between-object relationships (for example, one from an autumn, when the leaves may be gone, or yellow and brown). At the same time, the internal homogeneity of both objects will generally be mostly intact. The original VPint2 algorithm, however, relies equally on within-object and between-object weights, and can therefore suffer from artefacts and other inaccuracies caused by temporal heterogeneity. Visually, such artefacts would look like a fading gradient of an incorrect colour that is strong at the borders of objects and gradually fades into the colour hue of the rest of the object.

We consider an image reconstruction to suffer from exploding values when the VPint2 algorithm is diverging from, rather than converging to, a stable solution. This problem can arise in rare cases, because image data, and remote sensing imagery in particular, can suffer from inconsistencies, faulty pixels and other quality issues. If this occurs in the target image, these pixels can be treated as 'missing' and interpolated along with cloudy pixels, as long as the issues are identified in advance. However, some possible causes of quality issues in the data, such as solar glint or transmission errors, are not always easy to detect automatically. Moreover, if the issue exists in the feature image, it cannot simply be interpolated even if detected accurately. Because the weights derived from a faulty pixel can introduce an unrealistically large weight into the system, values multiplied by this weight can then be amplified too much by other weights and propagated along to other pixels as well. Similarly, the location of the border between objects can move over time. If the border between objects lies within the cloudy region in the feature image, but outside the cloudy region in the target image, it would be wrongly applied, despite the transition having already occurred in the target image (the opposite case of not being applied at all is also possible). In both of these cases, a disruption in the balance of the system of weights would cause unreasonably large values to be estimated, passed on to their neighbours, and grow at an even faster rate in the next iteration, resulting in 'exploding values' in that area.

Addressing these challenges, we propose two technical enhancements to the VPint2 method, thereby boosting its general performance and its applicability to

remote sensing imagery.

IDENTITY PRIORITY

Given the problems caused by temporal heterogeneity, our first enhancement of VPint2 aims to exploit reliable within-object relationships, while minimising the impact of less reliable between-object relationships. Since between-object relationships are more likely to suffer from temporal heterogeneity than same-object relationships (though both are possible), prioritising the information from neighbours belonging to the same object can alleviate the impact of this problem. To this end, we extended the VPint2 algorithm by incorporating *identity priority*. Recall Equation 4.2, where $\hat{t}_{ij}^{\tau+1}$ for unknown values was updated to the weighted average of neighbouring values. In effect, this update rule computes four independent predictions $\gamma_{ij}^{i'j'} \cdot \hat{t}_{i'j'}^{\tau}$ for every neighbouring pixel at positions (i', j'), and VPint2 later uses the mean thereof as its prediction. We realise identity priority by using a weighted mean instead of an average prediction. Specifically, we assign a priority weight $\lambda_{ij}^{i'j'}$ (separate from the *spatial* weight $\gamma_{ij}^{i'j'}$) to an individual prediction based on the spatial weight's distance to 1 (which would signify a same-object relationship). By computing $\lambda_{ij}^{i'j'}$ to be equal to $\gamma_{ij}^{i'j'}$ for spatial weights ≤ 1, but dividing 1 by $\gamma_{ij}^{i'j'}$ for spatial weights > 1, this distance is equal in both directions. As a result, we prevent bias towards over- or underestimations, as a spatial weight indicating a halved value (0.5) would have the same priority weight as a spatial weight indicating a doubled value (2). Moreover, in some cases it may be beneficial to increase or decrease the degree to which weights close to 1 are favoured. To control this intensity, we introduce a new parameter β, which determines the strength of the identity priority procedure. Thus, priority weights are computed as:

$$\lambda_{ij}^{i'j'} = \begin{cases} \beta \cdot \gamma_{ij}^{i'j'}, & \text{if } \gamma_{ij}^{i'j'} \leq 1 \\ \beta \cdot \frac{1}{\gamma_{ij}^{i'j'}}, & \text{otherwise,} \end{cases} \tag{4.4}$$

and if we denote the sum of priority weights $\sum_{(i',j') \in N(i,j)} \beta \cdot \lambda_{ij}^{i'j'}$ as Λ_{ij}, the modified version of Equation 4.2 becomes:

$$\hat{t}_{ij}^{\tau+1} = \begin{cases} o_{ij}, & \text{if } o_{ij} \notin \mathcal{C} \\ \frac{1}{\Lambda_{ij}} \cdot \sum_{(i',j') \in N(i,j)} \lambda_{ij}^{i'j'} \cdot \gamma_{ij}^{i'j'} \cdot \hat{t}_{i'j'}^{\tau}, & \text{otherwise} \end{cases} \tag{4.5}$$

The choice for the identity priority intensity parameter β can be highly relevant to the error rates of the algorithm. If it is set too high, error rates tend to increase on images (or regions thereof) where the between-object relationships did not change much in the time between the target and reference images, even if the results still look plausible to the human eye. Conversely, if β is set too low, some images may suffer from higher error rates, and visual artefacts may appear due to the temporal heterogeneity. We opted to automatically adapt β based on the performance of the algorithm on a validation set derived from the available cloud-free pixels, using the procedure described in Section 4.4.2. Regardless of the strength of β, identity priority cannot fully prevent artefacts from occurring at the borders of objects in specific cases, namely when two different objects in the target image were part of the same object in the feature image (thus having weights close to 1).

ELASTIC BAND RESISTANCE

The problems caused by exploding values can be substantial, particularly due to the value propagation-based nature of VPint2 propagating these errors throughout the entire image. To some extent, the impact of such errors can be mitigated by clipping the possible values in an image, but in this case, the exploded values would still greatly hamper performance by filling parts of the image with the user-defined maximal value. Moreover, such a hard threshold would leave little room for unlikely, but physically meaningful, high values. Therefore, we propose to combine a conservatively used hard threshold (that should, ideally, never be met) with a soft additional enhancement to VPint's update rule, which we refer to as *elastic band resistance*. This enhancement is aimed at improving performance on specific pixels, as well as preventing unreasonable predictions from propagating and possibly amplifying further, while still allowing exceptionally large increases in values where exceptional circumstances call for it. As an analogy, we can compare the increase of values at a certain pixel to stretching an elastic band. Up to a certain threshold, in this case the length of the elastic band, it can be moved freely. However, beyond this threshold, the further one stretches the band, the higher the resistance will be, and the more force needs to be applied to achieve even a tiny amount of additional length. This behaviour can be modelled using Hooke's law:

$$F = k \cdot \Delta L \tag{4.6}$$

Here, F is the force required to stretch an elastic band for an additional length ΔL. This is controlled by the resistance k, where high values for k require higher amounts of force for smaller stretching lengths.

We can adapt Equation 4.6 to our update function. If we define μ as the threshold beyond which we wish to apply resistance and $\Delta \hat{t}_{ij}^{\tau+1}$ as the amount of change

between $\hat{t}^{\tau}_{ij}$ and $\hat{t}^{\tau+1}_{ij}$ after running Equation 4.2 or 4.5, we can update an old pixel value $\hat{t}^{\tau}_{ij}$ to its new value $\hat{t}^{\tau+1}_{ij}$ as:

$$\hat{t}^{\tau+1}_{ij} = \begin{cases} \hat{t}^{\tau}_{ij} + \Delta\hat{t}^{\tau+1}_{ij}, & \text{if } \hat{t}^{\tau+1}_{ij} < \mu \\ \hat{t}^{\tau}_{ij} + \Delta\hat{t}^{\tau+1}_{ij} - k \cdot \hat{t}^{\tau}_{ij}, & \text{otherwise} \end{cases} \tag{4.7}$$

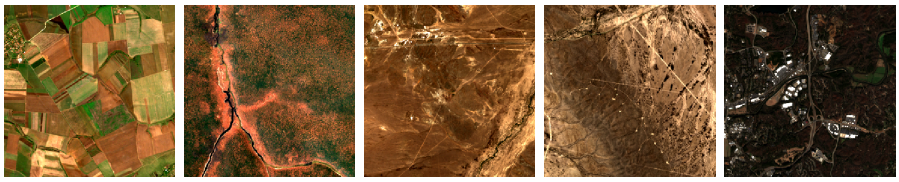
Since we use $\hat{t}^{\tau}_{ij}$ instead of $\Delta\hat{t}^{\tau+1}_{ij}$ as our penalty term, Equation 4.7 deviates somewhat from Hooke's law as stated in Equation 4.6. However, this provides us with the desired property: the larger the absolute difference between the previously predicted value $\hat{t}^{\tau}_{ij}$ and μ, the stronger the resistance applied by $k \cdot \hat{t}^{\tau+1}_{ij}$ will be, even if the force remains constant.

By applying this penalty term to the VPint2 update rule, drastic increases in values caused by exploding values could be dampened to a great extent, while this dampening would be much weaker on lower, more reasonable values. As a result, this functionality can address the problem of exploding values, provided the parameters μ and k are set appropriately. Much like β, overly aggressive settings for these parameters would result in higher error rates, due to values being unable to increase as far as they should. Thus, for these parameters as well, proper configuration is key in the performance of VPint2 for cloud removal, which we achieved through auto-adaptation.

AUTO-ADAPTATION

Identity priority and elastic band resistance introduce new parameters, which we propose to set automatically using auto-adaptation. When using proper configurations for β, μ and k, these extensions successfully alleviate problems VPint2 would encounter when applied to remote sensing imagery. They may also be effective at alleviating similar problems in other applications that suffer from faulty pixels or changing spatial structure, such as video processing or pipelines reliant on noisy measurements. However, as explained in Section 4.4.2 and 4.4.2, inappropriate settings can have deleterious effects on the performance of VPint2. Moreover, the best performing parameter settings tend to vary greatly between scenes, patches within a scene, and even the spectral bands within the same image. As a result, a single configuration for a full image will not perform optimally, since the performance gains in one area may come at the expense of losses in another. On the other hand, manually selecting the appropriate parameter settings for all 12 bands of all 400 patches in a scene would not be feasible. Therefore, the automatic configuration of VPint2 is essential to its successful application.

We added a self-adaptive mechanism for automatically setting β, μ and k to appropriate values. This mechanism leverages the available data by sub-sampling



(a) Cropland (b) Forest (c) Herbaceous (d) Shrubs (e) Urban/built up

Figure 4.3: Example target patches for the five land cover classes of our benchmark dataset. Figure 4.3a shows a patch from cropland in Hungary, Figure 4.3b shows a patch from a forested area in Angola (for which the target is from a dry period, but features for 6 months are green), Figure 4.3c shows a patch with herbaceous vegetation from Kazakhstan, Figure 4.3d shows a patch with shrubland from Mexico, and Figure 4.3e shows an urban patch from the US.

known datapoints into a validation set, where pixels with the greatest mismatch between the target and feature images are prioritised. The adaptation algorithm can then search for appropriate parameter settings by sampling from the parameter space of possible configurations, and running VPint2 on the image with additional datapoints missing. This allows the algorithm to assign a validation loss to every parameter configuration that is sampled. The algorithm supports grid search and random search in its current implementation, but in principle, any black-box optimisation algorithm can be used. To ensure that performance will be *at least* on par with the original VPint algorithm, a configuration with no identity priority or elastic band resistance is always sampled first, although it is possible that the available validation pixels are not representative for some patches.

4.5. EXPERIMENTS

This section explains the experimental setup used to evaluate the performance of VPint2 on cloud removal tasks. We first explain the chapter research questions that motivated our study and then cover the data we use in our experiments, the methods against which we compare, and our experimental setup.

4.5.1. CHAPTER RESEARCH QUESTIONS ADDRESSED IN OUR EXPERIMENTS

Our experiments were aimed at answering the following chapter research questions (which we later refer to as CRQ1 through CRQ5):

1. **Can identity priority and elastic band resistance improve VPint2's applicability to remote sensing imagery?** We lead with this question, as the answer to it will determine how VPint2 is used throughout the rest of our experi-

ments.

2. **Can VPint2 achieve significantly better results than competing methods?** We quantitatively evaluate this across scenes of different land cover classes, taking advantage of the diversity in our dataset to allow different methods to perform well on images from different types of ecosystems and geography.

3. **How do the temporal distance between target and feature images, and the percentage of cloud cover in the input image, affect the relative performance of the methods?** In many cases it can be difficult to procure recent cloud-free images, and interpolation methods are typically better at gap-filling small amounts of missing data, making these meaningful variables to study. Moreover, since VPint2 requires a cloud-free reference image as features, it should still be able to perform well for higher temporal distances. We measure the performance against four temporal distances of 1 week, 1 month, 3 months, and 6 months, and per-patch performance against the percentage of cloudy pixels in the patches.

4. **How well does VPint2 perform in terms of running time compared to competing methods?** Running time can be an important factor in the practical use of a cloud removal method. We compare the average running time for different levels of cloud cover, as this is often a key factor in the computational efficiency of cloud removal methods.

5. **What is the overlap between the errors produced by different methods, and how could complementary strengths be exploited?** We visualise errors between methods, and explore the impact of ensembling strategies for improved performance.

4.5.2. DATA

In our experiments, we focused on multi-spectral Sentinel-2 imagery (level 2A) as a use case for our method. The two Sentinel-2 satellites from the European Space Agency (ESA) measure reflectance values at 13 wavelength bands. The RGB bands (2,3,4) and band 8 have a spatial resolution of 10 m^2, bands 5,6,7,8A,11 and 12 have a spatial resolution of 20 m^2, and bands 1,9 and 10 have a resolution of 60 m^2. Band 10 is dropped in level 2A images, as it is mainly used for atmospheric correction, leaving 12 spectral bands to remove clouds from. We found existing benchmarks datasets to be scarce, and the few available benchmarks, such as SEN12MS by Schmitt et al. [177], its derivative SEN12MS-CR by Meraner et al. [40] and WHUS2-CRv by Li et al. [146], typically use previous cloud-free acquisitions as ground truth values and do not contain past imagery at various time intervals as feature data,

making them inapplicable to our use case. Therefore, we created the SEN2-MSI-T benchmark dataset (referring to Sentinel-2, the optical multi-spectral instrument used by Sentinel-2, and the temporal aspect of the dataset) for our experiments, inspired by the existing datasets mentioned above.

SEN2-MSI-T. This dataset contains co-located optical imagery and a cloud mask for the five most common land cover classes in the Copernicus Global Land Cover [178] dataset: cropland, forest, herbaceous (vegetation), shrubs, and urban/built up area. For every type of land cover, we manually defined multiple search areas predominantly filled with the same land cover, and automatically searched for candidate scene locations meeting our requirements, from which we selected 4 per land cover class, resulting in a total of 20 scenes. Each of these scenes, which we strove to obtain from diverse geographical locations from Europe, Asia, Africa, the Americas and Australia, contains a cloud-free target image sensed at time s. It also contains a matching cloud mask, obtained from a cloudy image from a time as close as possible to s. It furthermore contains four feature images at different (approximate) time intervals: $s-1$ week, $s-1$ month, $s-3$ months, and $s-6$ months. Thus, every scene consists of 6 different co-located large images in total. Each scene was partitioned into non-overlapping patches of 256×256 pixels, examples of which can be seen in Figure 4.3. Candidate solutions were identified, visualised and manually inspected using SentinelHub, and were downloaded as full level 2A data products using SentinelSat. At runtime, patches were loaded using windowed reading and resampled to a $10 \times 10 m^2$ resolution. We provide the code to generate the dataset, along with download locations for the (compressed) raw data, in the code repository accompanying this article.

In many existing cloud removal datasets, such as SEN12MS-CR [40], SEN12MS-CR-TS [179] and WHUS2-CRv [145], model training supervision and performance evaluation is performed by matching cloudy target images with cloud-free acquisitions from one or two satellite revisits before the target (in the case of Sentinel-2, the revisit time is generally 5 days). The advantage of this approach is that real cloudy input data is used, whereas synthetic data experiments may have poor generalisability to real-world data, due to unrealistic cloud profiles, the common types of clouds in real data varying based on geographic properties, and the visual representation of the cloud itself. However, even with small temporal distances, the pixel values of a scene may have changed substantially, potentially reducing the reliability of this type of evaluation approach (our results in Section 4.6.3 will support this intuition empirically). Therefore, to allow for an accurate validation of our cloud removal approach, the clouds used in our experiments were masked from a separate image, which was then applied to a cloud-free image, as this allowed us to compute accurate, up-to-date performance measures. Our evaluation approach, therefore, offers a middle ground between accurate performance metric computa-

tion, and realistic cloud cover suitable for the specific geographical location of the scenes. A similar approach was recently successfully employed by U-TILISE [176], although unlike in our approach, missing (cloudy) pixels were denoted using the maximal pixel value, instead of using explicitly missing data points.

SEN12MS-CR-TS [179]. This dataset contains time-series of multiple ROIs with Sentinel-2 Level 1C imagery. The ROIs are split into 256×256 pixel patches, with 30 potentially cloudy images available for every patch. By performing experiments on this dataset we were able to compare our method against many state-of-the-art methods, such as STGAN [180], U-TAE [181] and UnCRtainTS [41], while also serving as a frame of reference to compare against future methods that are evaluated on this dataset. However, we note that our method was not intended to be used for time-series cloud removal, and could therefore only run on a subset of the test dataset, as determined by the following criteria. To evaluate our proposed method, we required i) a cloudy target image with at least one non-cloudy pixel, ii) a cloud-free reference image, and iii) a cloud-free ground-truth image for evaluation. We could simulate a dataset satisfying these criteria by computing cloud masks for every time step for the patches in SEN12MS-CR-TS, and selecting the patches for which we could identify a combination of temporally close ground truth and target images, with a cloud-free reference image available at some time step prior to the target.

When making use of VPint2 for cloud removal on real-world cloudy input data, such as SEN12MS-CR-TS, users should take care to incorporate a high quality cloud masking algorithm. In our experiments on SEN12MS-CR-TS, we used the SEnSeI-v2 cloud detection model [29] to generate cloud masks, whereas experiments on SEN2-MSI-T used the cloud probability band of the Sentinel-2 Level 2A data products. In general, recall should be prioritised over precision for cloud masks for VPint2. Although high recall may come at the cost of lower precision, needlessly interpolating a few cloud-free pixels will not have a large impact on the performance of the algorithm. On the other hand, wrongly accepting cloudy pixels as true values, and thus propagating cloudy pixel values throughout the image, could have a substantial negative impact. In a similar vein, buffering cloud masks is recommended to ensure full masking around the edges of clouds.

4.5.3. COMPETING AND ALTERNATIVE METHODS

We compared the performance of VPint2 to that of several alternative state-of-the-art methods. We strove to include in our selection representative methods from all the categories listed in Section 4.3 (apart from interpolation methods due to scalability issues of competitive methods). Therefore, we compare the performance of VPint2 to that of temporal replacement, automated machine learning (AutoML)-

based regression ensembling, and a deep neural network specifically designed for cloud removal. The approach most comparable to our proposed method is temporal replacement, since it also requires no training and only relies on a past cloud-free reference image. The AutoML regression and deep neural network methods have more requirements, due to their reliance on training (as well as model selection and hyperparameter tuning), but represent advances in artificial intelligence and deep learning that may offer greater accuracy. Therefore, we consider them to be important competitors as well. Specifically, we selected the following methods for our comparative performance analysis:

- **Temporal pixel replacement [138].** Here we perform mosaicking by copying f_{ij} for all $o_{ij} \in C$:

$$\hat{t}_{ij} = \begin{cases} o_{ij}, & \text{if } o_{ij} \notin \mathcal{C} \\ f_{ij}, & \text{otherwise} \end{cases} \tag{4.8}$$

  Temporal pixel replacement is similar to the frequently used mosaicking setting 'LeastCloudy' in popular Earth observation data frameworks, such as Google Earth Engine [182]. Temporal replacement is a method highly reliant on the availability of recent cloud-free feature images, although in practice, such recent feature data will often not be available. Although we did explore an approach incorporating basic histogram matching [139], we found the original version of the method to perform better numerically on the atmospherically corrected level 2A images used in our experiments. Moreover, other types of mosaicking, such as taking the median of the most recent cloud-free values, would require more data than other methods have access to (a time-series of past data). As a result, we selected the original temporal replacement algorithm from Equation 4.8 as the representative method for this approach.

- **AutoML regression ensembling**. Many machine learning algorithms can be used for regression tasks, and can be combined using ensembling to further boost performance. To ensure that the best ensemble model is configured in our experiments, we remain agnostic about the type of models in question (for example, linear regression, support vector machines or gradient boosting) and their hyperparameters, and instead automate this process using the AutoML system auto-sklearn [128]. In AutoML, the choice of machine learning model and the optimisation of its hyperparameter settings are automated, resulting in more specific, fine-tuned models over general models. In the case of auto-sklearn, multiple machine learning models are optimised using Bayesian optimisation [183], and subsequently combined

into an ensemble. The available regression models include Gaussian processes, adaboost and random forests, as well as neural networks in the form of multi-layer perceptrons (MLPs). If we denote the ensemble found by auto-sklearn as $\mathcal{E}$, we compute $\hat{t}_{ij}$ as:

$$\hat{t}_{ij} = \begin{cases} o_{ij}, & \text{if } o_{ij} \notin \mathcal{C} \\ \mathcal{E}(f_{ij}), & \text{otherwise} \end{cases} \tag{4.9}$$

The Auto-sklearn model was trained on the available data per patch to counteract generalisation problems, as well as to ensure that this method has access to the same amount of data as the other methods.

- **Modified Neighborhood Similar Pixel Interpolator (MNSPI) [10, 165]**. NSPI is an interpolation method originally created for the gap-filling of the relatively small gaps of Landsat 7 ETM+ data [10]. In this method, a variable spatial window is used around a missing pixel, computing the target pixel value as a weighted sum of the values of similar pixels. NSPI combines a spectro-spatial prediction, based on the spectral similarity between pixels in the same image, with a spectro-temporal prediction, based on the spectral difference in a cloud-free reference image. The method was later modified to be applied to thick cloud removal for Landsat imagery. As an interpolation method, MNSPI may perform worse on larger gaps, particularly if the gaps are larger than the maximal spatial window, whereas making the window overly large would render the algorithm computationally infeasible. In our experiments, we used a maximum window size of 17 pixels, as suggested in the original papers [10, 165].

- **DSEN2-CR [40] and UnCRtain-TS [41]** (deep learning). We opted to also explore the effectiveness of deep learning techniques specifically designed for cloud removal tasks, since this type of method is most commonly explored in recent publications, boasting impressive performance. To our knowledge, no cloud removal neural networks currently exist that are specifically aimed at cloud removal using a past cloud-free reference image, and our own preliminary explorations into effectively adapting a network to such data proved to be challenging and out of the scope of this work. Nonetheless, since these methods represent to a large degree the state of the art in cloud removal in recent years, we decided to compare VPint2 to the performance of the popular deep learning-based DSen2-CR [40] model, which leverages SAR-optical data fusion, and UnCRtain-TS, which is a multi-temporal model (also using SAR data), but can be used for a single time step. For both methods, we used the official code repositories made available by the original authors, with

adaptations to the data loading procedure to load the SEN2-MSI-T dataset. This comparison did entail extra acquisitions of SAR data and ran on level 1C input data instead of level 2A, meaning the comparison between these models and VPint2 could only be performed on a separate experiment with level 1C targets, as explained in Section 4.5.4. Since both DSen2-CR and UnCRtain-TS were originally proposed for data fusion- and multi-temporal cloud removal, respectively, this comparison can shed light on whether such models could be successfully applied to this problem setting as well.

4.5.4. EXPERIMENTAL SETUP

The general approach of our experiments was as follows.

First, for every patch in SEN2-MSI-T, we transferred the cloud mask to the target image as missing values, allowing us to simultaneously have access to realistic cloudy images and ground truth values, providing a middle ground between synthetic and real-world dataset evaluation.

Second, we ran all methods on all scenes and their patches with all available feature sets (1 week, 1 month, 3 months and 6 months), and saved the reconstructed images as three-dimensional arrays. Following the standard of existing work [40, 41], input values were clipped to 10000. For our analysis, we also dropped combinations of images where the alignment was incorrect, and patches on the edges of the swath where part of the patch contained no data.

To compare against DSen2-CR and UnCRtain-TS, we ran additional experiments with VPint2 and these neural networks on the level 1C (L1C) version of the target image, along with recent SAR acquisitions (though this was only used by the neural networks). In these experiments, we simulated realistic clouds using Satellite Cloud Generator [184], which the original authors found to be suitable for DSen2-CR, using cloud cover percentages sampled from the real cloud cover percentages in the main dataset. Five scenes did not have recent SAR data available and were therefore not used in this experiment. Similarly, we dropped scenes where the alignment between the L1C targets and level 2A (L2A) feature images was imperfect, and patches that the SAR data product did not cover. As a result, the dataset and sample size we used for this experiment was substantially smaller than what we used for our main experiments. The L1C and SAR data products are included in our dataset specification, and were resampled to a $10m^2$ resolution and collocated using the SNAP tool by the European Space Agency.

All data used by DSen2-CR and UnCRtain-TS were preprocessed as described in the respective publications [40, 41]. The feature dataset used by VPint2 consisted of the reference image from 1 month before the target, still at a L2A processing level, but resampled to $10m^2$ to match the L1C targets. We should note, though,

that this is not the ideal use case for VPint2, as it was designed to be used with the same type of data. Cloud probability masks were obtained from the Satellite Cloud Generator model directly, removing cloud detection quality as a variable, and the derived binary masks were buffered in 5 passes (5 iterations of considering pixels next to currently cloudy pixels in the mask as cloudy).

Additionally, we performed an experiment on SEN12MS-CR-TS, as described in Section 4.5.2. For every patch in the dataset, we loaded all 30 time steps and computed their cloud masks using the SEnSeIv2 cloud detection algorithm [29]. We then checked whether there was any combination of time steps where a cloud-free target image followed a cloudy input image, with a cloud-free feature image available at some point in the past. For all patches where these conditions were met, we ran VPint2 on the input image to create a cloud-free reconstruction, and evaluated using the cloud-free next time step.

For our numerical evaluation, we utilised several performance metrics, some of which overlap with those used in Chapter 3, but whose formulation we reiterate here with our cloud removal-specific notation. Firstly, we used mean absolute error (MAE):

$$MAE(\hat{\mathbf{T}}, \mathbf{T}) = \frac{1}{|\mathbf{T}|} \cdot \sum_{\hat{t}_{ij} \in \hat{\mathbf{T}}} |\hat{t}_{ij} - t_{ij}| \tag{4.10}$$

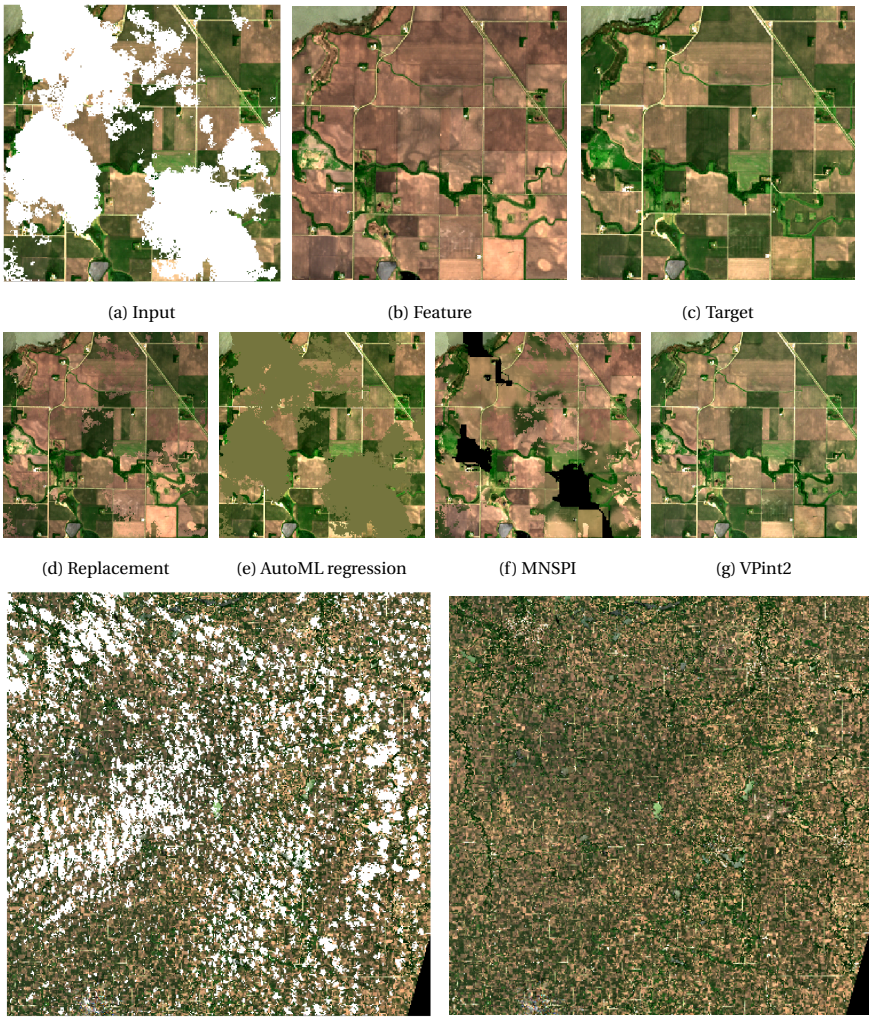
Secondly, we were interested in the utility of the images produced by different cloud removal methods for downstream tasks. We therefore computed the MAE on a normalised difference vegetation index (NDVI) computation task:

$$NDVI(\mathbf{x}_{ij}) = \frac{\mathbf{x}^8_{ij} - \mathbf{x}^4_{ij}}{\mathbf{x}^8_{ij} + \mathbf{x}^4_{ij}} \tag{4.11}$$

Here, $\mathbf{x}_{ij}$ represents a one-dimensional vector containing the band dimension of a single pixel at index ij, and the band superscripts correspond to the near-infrared (band 8) and red (band 4) bands in Sentinel-2 images (different sensors may require different bands). We then computed the MAE on the NDVI as the MAE between an NDVI derived from the reconstructed image $\hat{\mathbf{T}}$ and an NDVI derived from the ground truth image $\mathbf{T}$:

$$MAE^V(\hat{\mathbf{T}}, \mathbf{T}) = \frac{1}{|\mathbf{T}|} \sum_{\hat{t}_{ij} \in \hat{\mathbf{T}}} |NDVI(\hat{t}_{ij}) - NDVI(t_{ij})| \tag{4.12}$$

Thirdly, we included mean absolute percentage error (MAPE), as this gives an indication of errors regardless of the range of the underlying data, which varied

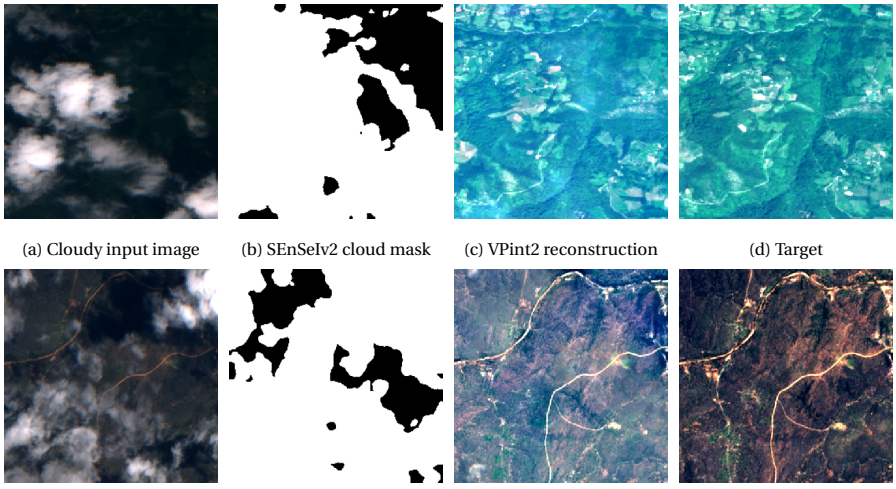


(a) Input (b) Feature (c) Target

(d) Replacement (e) AutoML regression (f) MNSPI (g) VPint2

(h) Input (full scene) (i) VPint2 (full scene)

Figure 4.4: Example reconstruction visualisations for VPint2, temporal replacement, AutoML regression and MNSPI. The visualised patch originated from a scene in Iowa, USA, with a cropland land cover class. The top row shows the feature-, target- and input images, the middle row shows the reconstructions by the different methods, and the bottom row shows the input and reconstruction of a full-sized scene.



(a) Cloudy input image (b) SEnSeIv2 cloud mask (c) VPint2 reconstruction (d) Target

(e) Cloudy input image (f) SEnSeIv2 cloud mask (g) VPint2 reconstruction (h) Target

Figure 4.5: Two example reconstructions with an average performance by VPint2 on the SEN12MS-CR-TS dataset (4.5c, 4.5g), along with the input images (4.5a, 4.5e), cloud mask (4.5b, 4.5f; white pixels denote clouds or cloud shadow) and the temporally close target image (4.5d, 4.5h). The cloud-free regions of the input images are dark in the visualisation due to the relatively high reflectance values of the clouds.

between patches and between the bands within a patch:

$$MAPE(\hat{\mathbf{T}}, \mathbf{T}) = \frac{1}{|\mathbf{T}|} \cdot \sum_{\hat{t}_{ij} \in \hat{\mathbf{T}}} \frac{|\hat{t}_{ij} - t_{ij}|}{t_{ij}} \tag{4.13}$$

Finally, we included the structural similarity index measure (SSIM) [185] as an indication of the quality of the produced images in terms of human perception:

$$SSIM(\hat{\mathbf{T}}, \mathbf{T}) = \frac{(2 \cdot \mu_{\hat{\mathbf{T}}} \cdot \mu_{\mathbf{T}}) \cdot (2 \cdot \sigma_{\hat{\mathbf{T}}\mathbf{T}} + c_2)}{(\mu_{\hat{\mathbf{T}}}^2 + \mu_{\mathbf{T}}^2 + c_1) \cdot (\sigma_{\hat{\mathbf{T}}}^2 + \sigma_{\mathbf{T}}^2 + c_2)} \tag{4.14}$$

Here, μ and σ represent the mean and standard deviation of the pixel values of the given image, respectively, and c_1 and c_2 are constants, for which we used the default values $c_1 = (0.01 \cdot L)^2$, $c_2 = (0.03 \cdot L)^2$ (where $L = \max(\mathbf{T}) - \min(\mathbf{T})$).

Additionally, our experiments on SEN12MS-CR-TS included the peak signal-to-noise ratio (PSNR), root mean squared error (RMSE) and spectral angle mapper (SAM) performance metrics, as these are the metrics methods are compared to in

	VPint2	VPint2 (no IP)	VPint2 (no EB)	Replacement	AutoML regression	MNSPI
			Cropland			
↓ MAE	**357.00±291.72**	367.65±303.88	370.11±310.79	614.23±450.28	460.19±293.43	**363.15±331.40**
↓ MAE^V	**0.0918±0.0787**	**0.0929±0.0799**	**0.0921±0.0809**	0.1563±0.1145	0.1239±0.0765	**0.0995±0.0983**
↓ MAPE	**24.398±58.405**	25.028±58.235	25.511±59.640	27.258±29.552	28.979±61.704	**27.277±63.188**
↑ SSIM	**0.8378±0.1907**	0.8317±0.1936	0.8332±0.1954	0.7028±0.2562	0.7994±0.1889	0.8021±0.2368
			Forest			
↓ MAE	**199.29±227.73**	**206.64±253.55**	**199.19±223.86**	421.39±473.35	402.28±620.58	**214.35±341.47**
↓ MAE^V	**0.0374±0.0274**	**0.0377±0.0282**	**0.0368±0.0256**	0.0758±0.0691	0.0857±0.0971	0.0494±0.0520
↓ MAPE	**6.442±4.508**	**6.620±5.422**	**6.469±4.676**	14.683±14.343	12.183±9.188	**7.003±8.109**
↑ SSIM	**0.9445±0.0705**	**0.9404±0.0855**	**0.9448±0.0712**	0.8756±0.1586	0.8731±0.1599	0.8662±0.2172
			Herbaceous			
↓ MAE	192.91±125.64	**191.18±128.91**	193.45±127.75	549.80±1392.8	264.27±145.17	222.86±201.32
↓ MAE^V	0.0361±0.0451	**0.0359±0.0463**	**0.0354±0.0450**	0.0476±0.0552	0.0394±0.0273	0.0426±0.0477
↓ MAPE	7.624±6.015	**7.543±6.080**	7.621±6.093	22.021±54.523	10.280±6.510	9.271±8.738
↑ SSIM	**0.9400±0.0881**	**0.9397±0.0947**	**0.9404±0.0896**	0.8718±0.2988	0.9321±0.0705	0.8397±0.1873
			Shrubs			
↓ MAE	**162.40±153.25**	**160.95±150.84**	**163.03±155.49**	315.76±202.21	286.42±235.04	209.88±229.97
↓ MAE^V	**0.0229±0.0273**	**0.0233±0.0290**	**0.0226±0.0271**	0.0379±0.0452	0.0293±0.0308	0.0312±0.0477
↓ MAPE	**8.780±81.345**	**9.680±108.487**	**9.595±100.018**	12.647±22.941	10.977±9.593	12.064±117.163
↑ SSIM	**0.9661±0.0599**	**0.9646±0.0632**	**0.9663±0.0602**	0.9466±0.0658	0.9427±0.0819	0.8357±0.1873
			Urban			
↓ MAE	**314.58±183.26**	335.39±201.86	**318.48±199.24**	553.24±494.50	590.15±390.03	388.52±231.95
↓ MAE^V	**0.1057±0.0782**	0.1166±0.0903	**0.1054±0.0784**	0.1465±0.1415	0.1487±0.0546	0.1244±0.1301
↓ MAPE	**18.421±18.116**	21.635±30.842	**19.106±21.157**	46.597±99.988	32.683±36.395	22.264±28.840
↑ SSIM	**0.7857±0.1575**	0.7695±0.1692	0.7755±0.1772	0.7180±0.2386	0.6674±0.1660	0.6563±0.2764

Table 4.1: Numerical results of our experiments. The best performing method per metric, where ↓ indicates a measure to be minimised and ↑ indicates a measure to be maximised, was computed using a one-sided Wilcoxon signed-rank test at a significance level $\alpha = 0.05$, and has been marked **bold** (ties allowed).

the original UnCRtain-TS paper [41]. We computed the metrics using the implementation provided with the code of UnCRtain-TS.

4.6. RESULTS AND DISCUSSION

A visual example of a SEN2-MSI-T patch, with its feature image, cloudy version and example reconstructions by the different methods, can be found in Figure 4.4. The bottom row contains results for VPint2 for a full scene reconstruction that was not split into patches, showing that VPint2 can be applied to larger images as well. In the figure, the reconstruction by temporal replacement contains outdated information for the vegetation in the target image. AutoML regression, due to conflicting relationships between pixels in the old image and the new, where some contain more vegetation in the target image while other pixels remained similar, ended up predicting mainly a mean value somewhere in between. MNSPI created blurry and occasionally outdated (similar to temporal replacement) visually plausible re-

sults within its spatial window, but failed to make a prediction for pixels outside of its window (while increasing this window further would render it computationally infeasible). The reconstruction by VPint2, seems the most visually plausible and seems to contain the most up-to-date information out of these methods on this example patch, with its greatest visual weakness appearing to be the propagation of incorrect colours in small parts of the image.

Two visual examples of SEN12MS-CR-TS patches, with cloudy inputs and their cloud masks, VPint2 reconstructions, and the cloud-free target images of the next time step, can be found in Figure 4.5. In this figure, the reconstructions appeared visually plausible, although the hue of the images were different between the VPint2 reconstruction and the target image. This was likely caused by a difference in atmospheric conditions between the Level 1C feature- and target images, and supports our intuition that a use case on atmospherically corrected Level 2A images would be preferred.

In the following, we report the results for specific chapter research questions in detail.

4.6.1. CRQ1: IDENTITY PRIORITY AND ELASTIC BAND RESISTANCE

To answer CRQ1, we investigated the effect of the extensions we made to the VPint2 algorithm as described in Section 4.4.2. These extensions were identity priority, aimed at reducing the impact of artefacts appearing on the edges of objects due to temporal heterogeneity, and elastic band resistance, aimed at preventing an explosion of extremely high values caused by, for example, quality issues in the data. To gauge the impact of these extensions, we performed an ablation experiment by running the main experiments for VPint2 three times: once with all features enabled, once with identity priority disabled (denoted as 'no ID'), and once with elastic band resistance disabled (denoted as 'no EB'). These results can be found in Table 4.1 and show that the added value of our extensions depends on the land cover type.

Identity priority appears to be particularly effective at improving performance on the urban- and cropland scenes. This is in line with expectations, since urban areas contain many smaller objects for which the between-object spatial relationship may change (such as bright reflections on roofs), resulting in temporal heterogeneity that can be alleviated using identity priority. Similarly, on cropland, the growth and harvest cycles may not have been applied uniformly to all fields, resulting in temporally heterogenous between-object spatial relationships. Elastic band resistance appears to be important on the cropland scenes, but not significantly different from the normal VPint2 results on other land covers. This further underlines that exploding values are rare, but if the phenomenon does occur,

performance can be significantly improved by enabling this enhancement. Therefore, elastic band resistance can be an important tool for cloud removal on certain scenes, but will not be necessary for most other problem settings.

In conclusion, the enhancements of VPint2, appear to improve the performance of VPint2 when enabled, although they are mainly necessary on specific land cover classes. Since VPint2 performed significantly better than the versions without our extensions in these cases, while the results for VPint2 on other land cover classes was generally not significantly worse than those without the extra functionalities, we will report the results for VPint2 with both enhancements enabled in subsequent experiments.

4.6.2. CRQ2: COMPARATIVE ANALYSIS

The numerical results of our empirical performance comparison on SEN2-MSI-T can be found in Table 4.1, and the distribution of the performance of the different methods has been visualised per land cover class in Figure 4.6. As the table and figure show, VPint2 achieved an improvement in performance over temporal replacement, AutoML regression and NSPI in all cases, which was statistically significant in all cases but the comparison with NSPI on cropland and forest land cover classes. The relatively similar performance of NSPI on these two land cover classes may indicate that the use of local spatial information in the input image, which both VPint2 and MNSPI exploit, is especially important for land cover with more vegetation. The spread of the performance by VPint2, as seen in Figure 4.6, tended to be smaller as well.

All methods performed worse on the cropland and urban scenes compared to other land covers, reflecting their challenge as dynamic land cover types (both in terms of values and spatial structure). AutoML regression often ended up predicting close to the target mean value, as can be seen in Figure 4.4e. This was likely caused by conflicting relationships between feature and target pixels in their respective images. For example, in Figures 4.4b and 4.4c, almost all feature pixels are a similar brown, whereas the target pixels for some fields were deep green, and some were light brown. In these cases, the algorithm seemingly converged to models predicting the mean, due to inconsistency of the pixel-wise feature-to-target relationships. We conclude that VPint2 would be a better cloud removal method than temporal replacement, AutoML regression and MNSPI in most cases.

The results for the experiment comparing against DSen2-CR and UnCRtain-TS on L1C targets can be found in Table 4.2. In this experiment, VPint2 performed better than UnCRtain-TS, and better than DSen2-CR on almost all land cover classes, with the exception of MAE^V on cropland and urban land covers, and a non-significant improvement on SSIM for cropland. The performance of VPint2 was stronger in

this experiment for SSIM on urban and cropland scenes, in particular, compared to the main experiments from Table 4.1. This was likely caused by the use of 1 month-old feature images boosting performance, although the relatively low contrast on L1C targets compared to L2A products may have also played a role.

	VPint2	**DSen2-CR**	**UnCRtain-TS (single time step)**
		Cropland	
↓ MAE	**189.88±60.90**	218.95±60.58	734.71±364.02
↓ MAE^V	0.0650±0.0271	**0.0445±0.0275**	0.1260±0.1429
↓ MAPE	**20.254±9.273**	51.780±49.219	28.338±17.161
↑ SSIM	**0.9182±0.0704**	**0.9133±0.0548**	0.3180±0.4199
		Forest	
↓ MAE	**163.67±43.35**	939.84±68.69	1109.16±342.80
↓ MAE^V	**0.0357±0.0126**	0.1424±0.0225	0.2875±0.0634
↓ MAPE	**6.257±1.923**	41.174±2.808	53.265±7.515
↑ SSIM	**0.9760±0.0169**	0.8145±0.0506	0.5667±0.3323
		Herbaceous	
↓ MAE	**182.07±64.27**	757.43±116.02	1147.27±342.80
↓ MAE^V	**0.0142±0.0070**	0.0634±0.0221	0.1241±0.1106
↓ MAPE	**7.106±3.107**	31.502±3.303	47.309±12.217
↑ SSIM	**0.9650±0.0308**	0.8484±0.0500	0.7254±0.1210
		Shrubs	
↓ MAE	**159.29±60.51**	684.01±343.74	1016.05±521.61
↓ MAE^V	**0.0225±0.0194**	0.0332±0.0361	0.1171±0.1086
↓ MAPE	**7.182±3.592**	30.066±11.997	39.309±14.875
↑ SSIM	**0.9704±0.0263**	0.8877±0.0573	0.8026±0.1238
		Urban	
↓ MAE	**182.22±53.50**	327.81±95.03	512.64±318.79
↓ MAE^V	0.0379±0.0211	**0.0282±0.0222**	0.0784±0.0832
↓ MAPE	**12.756±5.370**	36.782±32.532	23.576±13.883
↑ SSIM	**0.9628±0.0234**	0.9272±0.0414	0.8817±0.1030

Table 4.2: Numerical results of our experiments on L1C data for the subset of SEN2-MSI-T for which additional Sentinel-1 SAR data was available. The best performing method per metric, based on a one-sided Wilcoxon signed-rank test at significance level $\alpha = 0.05$, has been marked in **boldface**.

The comparison to DSen2-CR is generally favourable for VPint2, despite being used outside of its intended application of using the exact same type of feature data, and the mean absolute error for DSen2-CR was higher than expected given the performance reported in Meraner et al. [40] for the forest, herbaceous and

Method	↓ RMSE	↑ PSNR	↑ SSIM	↓ SAM
DSen2-CR	0.079	26.04	0.810	12.147
STGAN	0.060	25.42	0.818	12.548
CR-TS Net	0.057	26.68	0.836	10.657
U-TAE	0.051	27.05	0.849	11.649
UnCRtainTS	0.051	27.84	0.866	10.160
VPint2 (suitable subset of data)	0.042	30.38	0.928	6.541

Table 4.3: Comparison of the results of VPint2 on **a subset of** SEN12MS-CR-TS, against the multi-temporal performance of methods on the **full dataset** reported by Ebel et al. [41] Since this comparison is only for reference, and the methods were evaluated on different parts of the dataset, we do not mark the best performance.

shrubs scenes. However, the (value-independent) SSIM was on par, or sometimes better than, what was reported in this paper. As a result, the higher MAE may have been caused by the range of the values themselves in different land cover classes, rather than a truly worse performance, especially considering the relatively similar MAPE for all the land cover classes. UnCRtain-TS, a method intended for multi-temporal cloud removal, had the highest error rates and greatest variation in performance. A possible cause for this behaviour may lie in differences between the dataset it was trained on and our benchmark dataset.

Addressing this type of concern, the results for our experiment on SEN12MS-CR-TS, shown in Table 4.3, demonstrate that VPint2 performs very well on this dataset, in comparison with existing methods on a task they were designed for. We stress that this comparison is mainly included to put the results of our proposed method in perspective compared to a majority of recent state-of-the-art methods; since our method is only suitable for a subset of problem instances in the dataset, these results cannot be used to conclude that one method performs better than another, as competing methods might have also performed better on this subset. However, the results do indicate that, on a subset of suitable instances, there can be numerical advantages to using VPint2 over existing methods.

4.6.3. CRQ3 AND CRQ4: PATCH PROPERTIES AND COMPUTATIONAL EFFICIENCY

To answer CRQ3, we plotted the relationship between the temporal distance of the feature image and the percentage of cloud cover in the target image, with the performance achieved by the different methods, while for CRQ4, we also plotted the average (out of 20 randomly selected patches per level) end-to-end running time in seconds for the different methods for different levels of cloud cover. We

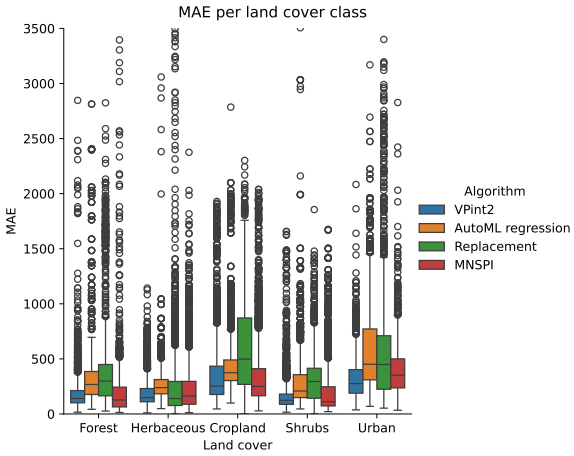
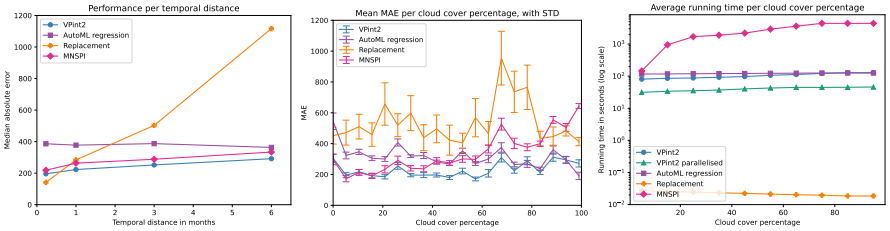


Figure 4.6: Box plots of the distributions of error rates (MAE) of the different methods for the five land cover classes. The visualisation has been limited to error rates of 3000, due to the outliers of the different methods reducing the legibility of the plots.

show these visualisations in Figure 4.7.

In the case of temporal distance, the results are as expected: Temporal replacement performs very well for temporal distances of 1 week, but quickly loses its effectiveness as the temporal distance increases. VPint2 and MNSPI are also affected by the temporal distance, presumably due to cases where the spatial structure of a scene was altered over time, but the effect is fairly mild. AutoML regression appears to not be affected by this variable, though slightly lower errors can be observed for larger temporal distances. However, this effect is small enough (386.27 at 1 week, 363.36 at 6 months), that this was likely caused by chance, rather than a true pattern. The temporal distance was especially important for herbaceous land cover scenes, which were exceptionally static on shorter temporal distances, but also changed exceptionally strongly for longer temporal distances due to seasonal effects (mainly snow cover).

The results in Figure 4.7a carry implications for the evaluation approach of cloud removal methods. In our experiments, we transferred cloud masks from a cloudy image at the same location as the target image, giving us access to realistic cloud cover as well as real ground truth values. On the other hand, many of the popular real-world cloud removal datasets used in high-profile work, such as SEN12MS-CR [40] and SEN12MS-CR-TS [179], rely on evaluating (and training) models by treating a co-located recent cloud-free acquisition as ground truth (with



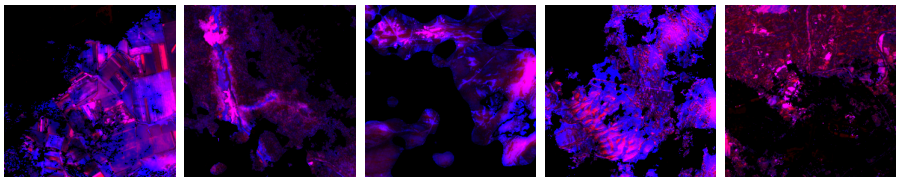
(a) Temporal distance (b) Cloud cover percentage (c) Running time

Figure 4.7: The sensitivity of different methods to the temporal distance of the feature image (4.7a), the cloud cover percentage of the input image (4.7b), based on mean absolute error (MAE), and the average running time in seconds out of 20 random patches per cloud cover percentage (4.7c). For Figure 4.7b, the figure was created by computing the average error for whole percentages and smoothing the resulting curve using splines interpolation. We added error bars for the standard deviation, to maintain an indication of the variability of results. The peak in errors for temporal replacement, which is a pixel-based method and should not be affected by the cloud cover percentage, was likely caused by the exceptionally large errors on particular problem instances (such as snowed-over herbaceous scenes at a distance of 6 months), that happened to contain a relatively large cloud cover percentage.

the closest possible time interval for Sentinel-2 being one 5-day revisit). Similarly, the feature image with a temporal distance of 1 week in our dataset consisted of 1 (preferred) or 2 (if necessary) revisits, which temporal replacement mosaicked into the target image as a cloud removal method. Therefore, the results for temporal replacement at 1 week in Figure 4.7a are an indication of the reliability that could be expected of real-world datasets. Although temporal replacement performed better at this temporal distance than other methods, its MAE at 1 week (140.97) reached levels comparable to the magnitude of the errors of VPint2 for all tested temporal distances (195.64 to 291.82).

These results suggest that, when using a purely real-world evaluation approach, the magnitude of the aleotoric uncertainty of the dataset would be comparable to the magnitude of the performance of cloud removal methods themselves, resulting in noisy and potentially unreliable evaluation. We therefore recommend further research to consider using a cloud mask transfer-based approach, as we have employed in SEN2-MSI-T, to evaluate cloud removal methods more reliably. Although this may not have been possible for neural networks, which do not use explicit cloud masks, and must therefore represent clouds realistically in the input image, recent advances in cloud simulation [184] may allow even neural networks to be trained on data with true ground-truth values available.

In the case of cloud cover percentage, a few observations can be made. First, VPint2 was not as heavily affected by larger percentages as might be expected from



(a) Cropland (b) Forest (c) Herbaceous (d) Shrubs (e) Urban

Figure 4.8: Visualisation of reconstruction errors (normalised and scaled to a 0-1 range) by VPint2 (red) and AutoML regression (blue) for an example patch from every land cover class, with a purple colour indicating overlapping errors. The existence of areas with mostly blue or red colours, as opposed to a constant purple colour, indicates complementary strengths between the two methods.

interpolation methods based on previous work [42]. Although there is an increase in errors (and variability) from 60% to 100%, there is no point where VPint2 clearly performs worse than the alternative methods apart from 100% cloud coverage. Second, temporal replacement was not affected by this variable, with the exception of a spike between 60% and 80%. Since this method is purely pixel-based, it is unlikely that a property of these particular clouds caused the spike. Instead, it is more likely that the performance on a specific challenging condition (for example, herbaceous land cover at 6 months) simply contained more patches with large clouds. Third, VPint2 and AutoML regression contain similar spikes in error rates, implying that similar patches are more challenging for both methods. However, VPint2 had lower error rates than AutoML regression, except for the highest cloud cover percentages, where performance was highly similar. Finally, MNSPI achieved results comparable to VPint2 for low cloud cover percentages, but its MAE increased steeply for higher cloud cover percentages. Overall, it appears that VPint2 is effective at addressing this weakness of interpolation methods, with the caveat that its competitive advantage over competing (non-interpolation) methods does slightly decrease for very high percentages, where it performs on par with the most competitive alternative method.

In terms of running time, Figure 4.7c shows that temporal replacement was by far the fastest method, with a running time on the order of magnitude of 0.01 throughout. VPint2 was the second fastest method, with its parallelised version reducing the average running times from about 100 seconds to between about 30 to 40 seconds. A mild increase in running time can be observed as the cloud cover percentage increases. MNSPI had a running time comparable to the serial version of VPint2 for low cloud cover, but rose to a running time exceeding 1000 seconds after about 20% cloud cover, likely caused by the need for larger window sizes. AutoML regression had a running time slightly above 1000 seconds; however, this

	Cropland	Forest	Herbaceous	Shrubs	Urban
Oracle pixel	**295.19±342.45**	**152.36±246.99**	**180.44±161.64**	**178.14±240.31**	**277.34±210.43**
Oracle pixel (no VPint)	308.36±326.58	169.70±253.35	182.67±151.80	189.76±230.23	312.70±211.84
Oracle patch	344.10±310.40	208.26±268.57	164.42±132.12	182.22±192.00	317.78±208.48
Oracle patch (no VPint)	406.90±298.48	259.37±270.76	183.82±136.31	244.56±183.21	344.47±219.70
Best individual method	357.00±291.27	199.19±223.86	191.18±128.91	160.95±150.84	314.58±183.26

Table 4.4: Numerical results of our ensembling explorations (MAE only). The oracle approaches could not be used in practice, and serve as a lower bound of what a perfect ensemble could achieve with these methods. The best individual method is shown for every land cover class as reference, and corresponds to the strongest method for that land cover class in Table 4.1.

includes training as well as algorithm selection and hyperparameter optimisation time, which was limited by a user-supplied parameter. Therefore, since its inference impact is negligible, its running time depends mainly on how long a user will allow it to search for good configurations (but a lower budget may result in worse numerical performance).

4.6.4. CRQ5: COMPLEMENTARY STRENGTHS AND ENSEMBLING

Although VPint2 achieved strong performances in our experiments, as seen in Table 4.1, a method that performs best on average is not necessarily the strongest on all instances. This is especially the case for Earth observation data, which is inherently diverse in terms of sensors, spectral bands, landscape, atmospheric conditions and more. As a result, when evaluating new algorithms applicable to this type of data, it is potentially problematic to merely consider average performance over a diverse collection of datasets. Instead, we believe that it is preferable to assess the relative strength of new approaches on individual datasets, and in particular, to focus algorithm development on scenarios where currently available methods appear to perform relatively poorly.

From this perspective, when comparing the performance of different methods, it is important to assess the complementarity of the strengths of the methods. We therefore visualised the reconstruction errors for VPint2 *vs* AutoML regression in Figure 4.8. In this figure, VPint2 errors were visualised in the red band, and AutoML regression errors were visualised in the blue band, meaning that only regions with a purple colour would show a strong overlap in performance. Since Figure 4.8 shows many regions with either red or blue colours, it is clear that both methods have strengths that the other does not.

We explored this idea further by probing the potential of ensembling approaches. Although a fully functional ensembling approach would entail addressing non-trivial challenges, such as finding informative features for automated algorithm selection, and is therefore beyond the scope of this work, we wish to show the po-

tential of this type of approach using an "oracle"-based experiment. We show the results in terms of MAE that can be achieved in this way in Table 4.4; we only considered MAE, since the ranking of the methods we studied was consistent across all performance metrics.

The experiments whose results are reported in the table were carried out as follows. First, we ran preliminary experiments using an "oracle" ensemble on a pixel level, selecting the most accurate predicted value out of VPint2, AutoML regression and temporal replacement, for every pixel. This produces a lower bound of the error rates achievable by a perfect ensemble. As shown in Table 4.4, this approach consistently significantly outperforms the best individual method for every scene, which demonstrates that in principle, an ensemble (when accurately selecting methods) could achieve substantially better results than any single method. We also ran this oracle setup without including VPint2 and observed a significantly reduced performance, further underlining the degree to which VPint2 contributes to the state of the art in cloud removal. Finally, we included a setup selecting methods per patch instead of per pixel, which could be used to assess whether the complementary strengths of methods occurred at the patch- or pixel-level. The results for patch-level ensembling were much closer to the best individual method. This suggests that a perfect ensemble nearly always selects the same method for every patch as the overall best method for the land cover class, and that properties of the patch do not contribute strongly toward which method performs best. Moreover, we observe that complementary strengths occur at pixel- rather than at patch-level.

4.7. CONCLUDING REMARKS AND FUTURE DIRECTIONS

In this chapter, we have extended the spatial interpolation algorithm VPint [186] to create VPint2, which is aimed at addressing optical remote sensing cloud removal problems. We made four key technical contributions to the original VPint algorithm, namely the use of exact weights computed directly from co-located past imagery, a running time speedup using parallel computing over bands, identity priority and elastic band resistance, addressing the temporal heterogeneity and exploding values problems in remote sensing data and allowing VPint2 to perform well on optical Earth observation data. Our proposed method does not use any additional data compared to temporal replacement, and requires no training procedure. It also automatically adapts its parameters to the best values based on the available data, which is necessary, because the appropriate settings can vary greatly even within a single image.

We created an evaluation benchmark dataset called SEN2-MSI-T, consisting of 20 geographically diverse scenes for the five most common land cover classes,

enabling us to evaluate cloud removal methods on a diverse set of environmental conditions and spatial patterns that also provides users with true ground truth values. The results from our experiments indicate that this method of evaluation is more reliable than common approaches using fully real-world datasets. Additionally, we performed an experiment on a subset of the popular SEN12MS-CR-TS dataset to better place our work in the context of recent work.

Our empirical results show that VPint2 significantly outperforms alternative methods on all land cover classes on average. We have also found that VPint2 is only mildly affected by the temporal distance of its reference image, which algorithms should be robust to as it may take several months to acquire cloud-free imagery during a rainy season. If no cloud-free data is available at all, this limits the applicability of VPint2 as a cloud removal algorithm, because weights could not be computed in such cases. However, the low sensitivity to temporal distance of the reference image suggest that the pool of past imagery to draw from is substantial. On the other hand, more outdated reference imagery increases the probability of altered spatial structure (e.g., through human activity). Although our empirical experiments have shown the average numerical impact hereof to be limited, in part due to our methodological extensions to mitigate it, the fidelity of specific reconstructed images may be reduced.

We also found a low sensitivity to the percentage of cloud cover in a patch, allowing it to be applied to a wide range of cloud cover conditions, and our experiments showed that VPint2 is more computationally efficient than existing cloud removal interpolation methods such as MNSPI. Moreover, our newly introduced parallelisation further cut the running time for VPint2 by about 60% to 70%. However, VPint2 requires at least one non-cloudy pixel in the input image to propagate values; if there are no non-cloudy pixels at all, the algorithm can only return the feature data as a reconstruction (equivalent to temporal replacement), limiting its advantages under such boundary conditions.

Our findings also encourage the adoption of an approach where new cloud removal methods are evaluated based on their specialist utility in a certain subset of use cases, as general methods tend to perform worse in inherently diverse domains, such as Earth observation data. Our “oracle”-based experimental results show that ensembling approaches using the strengths of multiple methods, especially on a pixel level, hold great potential for further performance improvements. We believe that identifying useful instance features for practically applicable ensembling approaches would be a fruitful endeavour in future work.

Other future work could explore the potential efficacy of VPint2 for time-series cloud removal, deriving weights from the cloud-free regions of the images in the time-series and combining these into one set of weights; we will discuss this further in Section 7.2.2. It may also be worthwhile to explore the impact of adding

feature data of different sensor modalities, such as SAR.

In conclusion, VPint2, as an easy-to-apply and effective cloud removal method, has shown its potential in terms of performance, as well as its complementarity with existing methods.

Having proposed methods addressing both components of Challenge 1 (VPint for spatial interpolation, and VPint2 for interpolating satellite imagery), in the next chapter, we will move on to Challenge 2 (noise and ill-posedness). Even when all EO data is consistently available, and our validation datasets cover the full grids of our study areas, the nature of the parameter estimation inference problem itself can still result in substantial hurdles to reliable estimations. In the next chapter, we will dive into the properties of parameter estimation using EO data, and map the reliability of the estimations we can make in this manner.

5

CHARACTERISING THE ILL-POSEDNESS OF PROSAIL INVERSION FOR PARAMETER ESTIMATION

In the previous two chapters, we focused on addressing Challenge 1 (data inconsistency). Through VPint, we can interpolate our in-situ validation datasets, covering a limited set of locations over a region of interest, into a full gridded dataset, while VPint2 allows us to obtain full satellite imagery without data gaps. We now turn our attention to Challenge 2 (noise and ill-posedness. In this chapter[1], we will address this challenge by answering RQ3: *What makes parameter estimation an ill-posed problem, and which factors affect the reliability of parameter estimation results?* We will map out the properties of PROSAIL inversion as a prominent example of a parameter estimation task, establish how reliable parameter estimation solutions are, and which properties are the likely causes of ill-posedness. This knowledge can then be used to inform further methodological contributions alleviating Challenge 2.

5.1. INTRODUCTION

The estimation of biophysical parameters is an important task for the monitoring of ecosystems, planning interventions where appropriate, and modelling their im-

[1] The contents of this chapter are based on the journal article: Laurens Arp, Peter van Bodegom, Holger H. Hoos, and Mitra Baratchi. (2026). *Characterising the Ill-posedness of PROSAIL Inversion for Biophysical Parameter Retrieval.* European Journal of Remote Sensing, 59(1). Taylor and Francis. https://doi.org/10.1080/22797254.2026.2632518

pacts on other predictive tasks. For some parameters it is possible to perform in-situ measurement campaigns to directly measure the values of these parameters [187, 188]. However, as explained in Sections 2.1.1 and 2.2.1, such campaigns are costly and time-intensive, and can only cover smaller, individual areas at a particular time, leading to scalability and representability issues for regular, global monitoring applications. This necessitates indirect inference methods to estimate these parameters from remote sensing data instead [35]. The parameter estimation performance can then be validated using available in-situ data [35].

Among parameter estimation efforts, those reliant on airborne spectral data are larger in scale than in-situ missions, but are expensive and difficult to scale up further, while the retrieved variables may be less reliable. Due to the cost- and scale constraints, estimations from spaceborne sensors are preferred for regular, global monitoring applications, but performing the parameter estimation required for this task is not a trivial problem.

A common method of estimating these parameters is through the use of radiative transfer model (RTM) inversion [189], which we explain in detail in Section 2.2.1. Possibly the most commonly used RTM for vegetation applications is PROSAIL [190, 191], consisting of the leaf- and canopy vegetation parameters from its constituent models PROSPECT [43] and 4SAIL [44], used to estimate soil and vegetation parameters. Since the PROSAIL model is based on the causal relationship between biophysical parameters and light spectra, while light spectra can be readily observed through remote sensing technologies, these models must be inverted to perform parameter estimation. As explained in Sections 1.1.2 and 2.2.1, this RTM inversion task is widely considered to be an ill-posed problem: multiple solutions may fit the observations equally well [57, 58, 59]. However, the ill-posedness of this problem is not yet well understood.

In this chapter, we aim to address the gap in the knowledge about PROSAIL inversion for vegetation parameter retrieval, through a thorough empirical study of the theoretical properties of the ill-posedness of the problem. We do this through the lens of the inversion *loss landscape*. This landscape quantifies the goodness-of-fit of all possible combinations of parameters, when comparing their simulated output to observed spectra. By empirically studying the properties of this loss landscape, we can verify whether PROSAIL inversion meets the formal definition of an ill-posed problem, and if not, what could be other possible causes of ill-posedness for parameter retrieval.

With this knowledge, we hope to enable practitioners to focus their efforts to alleviate ill-posedness on the factors that contribute strongly to the ill-posedness of biophysical parameter retrieval. Through our experiments, we found that PROSAIL inversion is *not* ill-posed; however, parameter estimation as a whole is. These results encourage future work to focus not on further improving the effectiveness

of finding the optimal solution to the PROSAIL inversion problem, which is actually well-posed, but rather on addressing key limitations of the data of the parameter retrieval problem such as noise or spectral mixing.

Through our analyses, we aimed to answer the following chapter research questions (CRQs):

- **CRQ1:** Does PROSAIL inversion meet the formal requirements of an ill-posed problem?
- **CRQ2:** What are the possible causes of the ill-posedness of biophysical parameter retrieval through PROSAIL inversion?
- **CRQ3:** How does adding priors to the parameter ranges impact the ill-posedness of biophysical parameter retrieval through PROSAIL inversion?

5.2. RELATED WORK

RTM inversion can be performed based on two main approaches. Traditionally, numerical optimisation techniques have been used [47, 49, 48]. More recently, hybrid modelling approaches, where a machine learning model is trained on a look-up table (LUT) of simulations, have gained popularity [50, 51, 45, 53, 54, 55], in part because they can always provide an estimation for the parameters to retrieve, even for ill-posed problems where traditional inversion methods may fail to provide a prediction.

However, there is a danger that hybrid models obscure the ill-posedness. The reason is that, when the underlying problem is ill-posed, there may be more than one valid result to the inversion problem, which may be disjoint from the area around a point prediction, while the metrics for validating the performance of these models only evaluate performance based on (the confidence interval of) one of these many solutions. Similarly, applying conventional machine learning approaches (as opposed to RTM inversion), such as training regression models on spectral observations and in-situ measurements, will be difficult because the training data are necessarily limited to a specific study area (see, e.g., [192, 193, 194]); we discuss this in more detail in Section 2.2.1. Moreover, these regression models may be affected by the same ill-posedness as model inversion methods, if this is an inherent property of the spectral information.

Given the challenges it causes, much work on PROSAIL inversion has incorporated measures to reduce the ill-posedness. Most commonly, the ill-posedness is reduced by adding prior knowledge (priors) to the model, representing domain knowledge on, for example, certain types of vegetation known to be dominant in a study area [195, 190, 196, 197, 198]. Based on this prior domain knowledge, the

ranges of some key parameters can be reduced, which has previously been found to improve estimation performance by addressing ill-posedness [57]. Similarly, in the case of hybrid models, the ill-posedness can be reduced by training specialised models on simulated training data with reduced parameter ranges that are statistically probable for a specific study area [45, 52, 46]. Numerous active learning heuristics have been proposed to improve hybrid models [45], such as approaches sampling points with the highest uncertainty or automatically matching an expected distribution [52, 46].

While such methods may have succeeded in improving performance on ill-posed problems on specific study areas, little work has been done on characterising the ill-posedness underlying the problem in more depth, investigating possible causes and analysing the impact of the mechanisms through which approaches such as constraining parameter ranges could reduce ill-posedness. Without a clear understanding of the critical characteristics of the ill-posed problem and its causes, it will be difficult to establish how to overcome ill-posedness, or even if it is a problem of PROSAIL inversion specifically, or a symptom of the overarching parameter estimation problem. Such knowledge is critical to design scalable biophysical parameter estimation models that are generalisable to a diverse range of environments, and is, therefore, the focus of the work in this chapter.

5.3. METHODS

In the following, we will explain the details of our methods and experimental setup. To this end, we will first introduce the terms and formal definitions that will be used throughout this section, many of which follow the conventions we introduced in Chapters 1 and 2.

Definition 5.1 ((biophysical) parameter)**.** A variable describing a component of a biophysical system (vegetation in the case of PROSAIL), often a target variable to retrieve.

We denote an individual parameter as $p \in P$, where P is the set of all free parameters under study. Following Definition 2.3 in Chapter 2, we denote a configuration as $\boldsymbol{\theta}$, represented by a vector containing concrete value assignments for the parameters $p \in P$, where every jth value $\boldsymbol{\theta}^j$ in $\boldsymbol{\theta}$ corresponds to the jth parameter p^j in P.

Definition 5.2 (parameter space (search space))**.** A $|P|$-dimensional space of all possible configurations (also known as the search space in optimisation contexts), where every point corresponds to a specific configuration.

We denote the parameter space as D_P, and for all configurations $\boldsymbol{\theta}$ it holds that $\boldsymbol{\theta} \in D_P$.

For spectral information, we use $\mathbf{x}$ to denote an observed spectrum, consisting of individual measurements x^b at multiple spectral bands b. Our experiments contained 12 bands b, corresponding to the spectral bands of the popular Sentinel-2 satellite (level 2A data products; cirrus band B10 dropped after atmospheric correction).

Definition 5.3 (instance)**.** One specific problem scenario to solve, consisting of an observed spectrum $\mathbf{x}_i$, and an unknown true configuration $\boldsymbol{\theta}_i^+$ to approximate.

A simulated spectrum can be obtained by running PROSAIL, denoted as M, on a configuration $\boldsymbol{\theta}$, which performs a simulation to generate the simulated spectrum $\hat{\mathbf{x}} = M(\boldsymbol{\theta})$. This allows us to define a spectral loss function for a configuration $\boldsymbol{\theta}$, given the observed spectral information $\mathbf{x}$:

Definition 5.4 ((spectral) loss function)**.** A function measuring the distance between a simulated spectrum and an observed spectrum, that can be used to measure the goodness-of-fit of candidate solutions for optimisation purposes.

We denote the loss function as $\mathcal{L}\big(M(\boldsymbol{\theta}),\mathbf{x}\big) = d(\hat{\mathbf{x}},\mathbf{x})$, where d can be any distance metric between $\hat{\mathbf{x}}$ and $\mathbf{x}$. In our experiments, we used the proportional mean absolute error (PMAE) and the spectral angle mapper (SAM). PMAE is a variant of the mean absolute error (MAE) whose proportionality results in an equal weighting between the lower spectral bands (with lower intensities) and the infrared bands (with higher intensities). SAM focuses on the relationship between bands rather than absolute values, and may therefore measure complementary properties compared to standard error metrics like MAE (for details, see Appendix A.1.1).

If the loss function $\mathcal{L}(M\big(\boldsymbol{\theta}),\mathbf{x}\big)$ were evaluated for every point $\boldsymbol{\theta}$ in D_P, a $|P|$-dimensional manifold in $|P|+1$-dimensional space would emerge, such that all possible coordinates (representing parameter configurations $\boldsymbol{\theta} \in D_P$) are associated with a goodness-of-fit value quantified by the loss function. This manifold is known as a loss landscape:

Definition 5.5 (loss landscape)**.** A $|P|$-dimensional manifold in $|P|+1$-dimensional space, measuring loss function values for every possible configuration $\boldsymbol{\theta} \in D_P$.

Finally, we denote an optimum (or minimum) minimising the loss function $\mathcal{L}$ for an observed spectrum $\mathbf{x}$ as $\hat{\boldsymbol{\theta}}$.

Definition 5.6 (optimum)**.** A configuration $\hat{\boldsymbol{\theta}} \in D_P$ for which the loss function value is minimised compared to its direct neighbourhood (local optimum) or for the entire loss landscape (global optimum).

Parameter	Abbreviation	Distribution	Range	Default	Unit
Leaf area index	LAI	$\mathcal{N}(3.88, 1.98)$	(0.1 – 10.0)	3.88	m^2 leaf / m^2 soil
Chlorophyll a+b	C_{ab}	$\mathcal{N}(32.81, 18.87)$	(0.3 – 106.72)	32.81	$\mu g/cm^2$
Average leaf angle	ALA	$\mathcal{U}(0, 90)$	(0 – 90)	45	$\circ$
Leaf water content	C_w	$\mathcal{N}(0.0129, 0.0073)$	(0.0043 – 0.07)	0.0129	cm

Table 5.1: PROSAIL parameters with their names, ranges, values, and distributions, for the parameters we kept variable in our experiments. A distribution of $\mathcal{N}(\mu, \sigma)$ refers to a normal distribution with mean μ and standard deviation σ, and a distribution of $\mathcal{U}(min, max)$ refers to a uniform distribution within the specified bounds. The ranges were determined through the parameter ranges specified in the documentation of the PROSAIL implementation[2].

The objective of PROSAIL inversion, therefore, is to find this optimum $\hat{\boldsymbol{\theta}}$, using the RTM inversion objective of Equation 2.1. It is possible for more than one configuration $\boldsymbol{\theta}$ to minimise the loss function, if multiple configurations share the same loss function value.

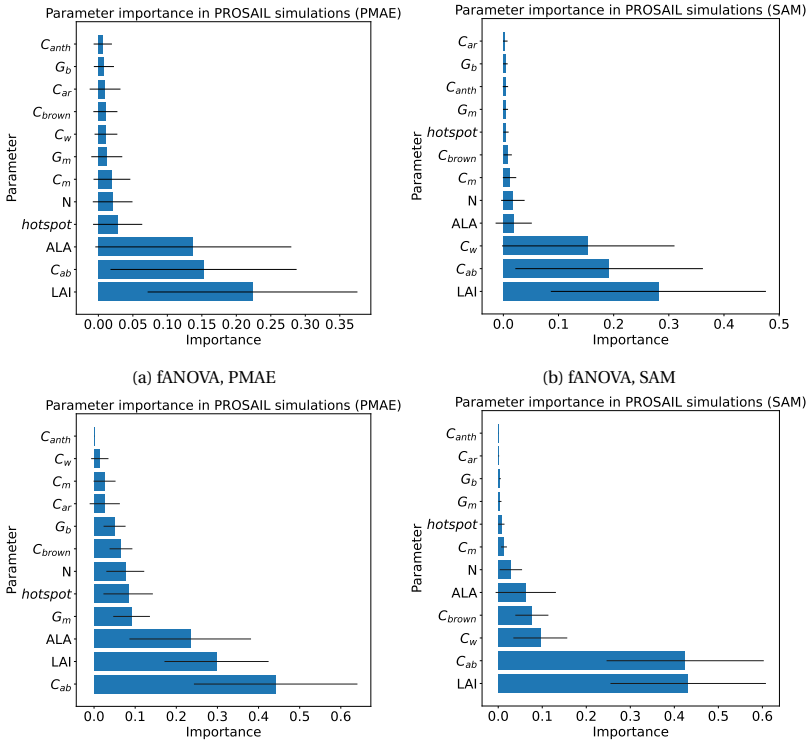
5.3.1. GENERAL EXPERIMENTAL SETUP

Our general study design was as follows. First, we used PROSAIL to generate simulated instances i. The sampling strategy used to select initial generating configurations $\boldsymbol{\theta}^+$ took the prior distributions of individual parameters (see Table 5.1 for details) into account. We generated a simulated look-up table (LUT) D, consisting of 1000 instances i combining the true configurations $\boldsymbol{\theta}_i^+$ and the associated simulated spectra $\mathbf{x}_i$. We mapped the raw PROSAIL outputs, which contain 1300 hyperspectral bands (wavelengths of 400–2700nm in steps of 1nm), to the multispectral format of the popular Sentinel-2 satellite using the spectral responses provided by the European Space Agency (ESA), thereby conforming to a realistic application setup.

5.3.2. PARAMETER IMPORTANCE AND CORRELATION

The PROSAIL model contains 15 numerical parameters that could be explored, although 3 of these concern observer- and solar geometry, which are known in practice, resulting in 12 potential free parameters. Not all of these parameters are equally impactful to the spectral loss, making some more appropriate to retrieve through PROSAIL inversion than others, while exploring all parameters also makes the problem prohibitively computationally expensive. In the interest of efficiency, we selected the most impactful parameters based on their importance in a preliminary experiment, determined by functional ANOVA (fANOVA) [199] and Sobol indices [200].

[1] https://github.com/jgomezdans/prosail



(a) fANOVA, PMAE

(b) fANOVA, SAM

(c) Sobol, PMAE

(d) Sobol, SAM

Figure 5.1: Parameter importance for the spectral loss landscape determined using functional ANOVA [199] (5.1a, 5.1b) and Sobol indicies [200] (5.1c, 5.1d). The PROSAIL parameters shown in these plots are leaf area index (LAI), chlorophyll $a+b$ content (C_{ab}), average leaf angle (ALA), hotspot, structural N, leaf dry matter content (C_m), leaf water content (C_w), leaf brown pigment content (C_{brown}), carotenoid concentration (C_{ar}), anthocyannins (C_{anth}), soil moisture (G_m) and soil brightness (G_b). The relative importance of parameters can be heavily reliant on the spectral loss function used: proportional mean absolute error (PMAE) or spectral angle mapper (SAM).

In fANOVA, the variance of a surrogate model approximating a target function (in our case PROSAIL forward simulations) is partitioned per input parameter. By using a surrogate model, the multiple function evaluations necessary to determine the variance of the model can be computed efficiently for potentially complex, high-dimensional functions. The change in output can be computed as a function of changes in the input, resulting in sensitivity indices that give an indication of parameter importance. Similarly, Sobol indices can be computed by decomposing the total variance of the output into the fraction of the total variance explained by every parameter. These indices can also be computed for higher-order indices; we included both individual contributions and parameter interactions. Parameters with a high sensitivity have a large impact on simulated spectra, while parameters with a low sensitivity have a low impact on the spectra. Therefore, we performed experiments on parameters with a high impact on the spectra, since these are the most suitable parameters to retrieve via PROSAIL inversion.

Concretely, we generated 200 additional instances, keeping all 12 parameters as free parameters. For fANOVA, we sampled 200 possible configurations with their spectral loss for every instance. For Sobol indices, we sampled 2048 configurations with their spectral loss, since this approach likely needed a larger sample size for reliable results. For both methods, this approach allowed us to compute the importance of different parameters for specific instances, which we then aggregated over all 200 additional instances. We performed this preliminary experiment for both the PMAE and SAM loss functions, and the resulting distributions can be found in Figure 5.1. Some parameters that were very important for PMAE (such as average leaf angle – ALA), were far less important for SAM, and vice versa (such as leaf water content – C_w). The results for fANOVA and Sobol indices largely overlapped, although the places of LAI and C_{ab} were switched for PMAE. The results mainly started to differ starting from the fourth-most important parameter (e.g., soil moisture G_m). The most impactful parameters tended to have the highest impact on specific spectral bands; details can be found in Appendix B.1.2.

Based on the information in Figure 5.1, we opted to include all parameters that were in the top 3 most impactful parameters for either of the loss functions and either parameter sensitivity method. This resulted in a parameter selection of leaf area index (LAI), chlorophyll $a+b$ content (C_{ab}), average leaf angle (ALA), and leaf water content (C_w). A description of these parameters, along with and their ranges, prior distributions and default values, can be found in Table 5.1. All other parameters were kept at default values (as in the study by De Sa et al. [50]).

Experiment	Description	Research question
E1	Testing for unimodality VS multimodality.	CRQ1
E2	Testing for continuity of the relationship between input spectra and the identified optimum.	CRQ1
E3	Testing for the average shift in optimum found for various levels of Gaussian noise on the spectral observations.	CRQ2
E4	Testing for the impact of spectral mixing on the retrieval estimations.	CRQ2
E5	Testing for the mechanism through which range constraint priors alleviate ill-posedness in retrieval problems.	CRQ3

Table 5.2: Summary of experiments and their target research questions.

5.3.3. PROSAIL INVERSION APPROACH

Since we are interested in the characteristics of the inversion loss landscape, our experiments were based on numerical optimisation. Numerical optimisation methods, such as black-box optimisation techniques (e.g., stochastic local search procedures [64], evolutionary algorithms [201] and swarm-based metaheuristic algorithms [202]), can provide richer insight into the underlying loss landscape of the inversion problem compared to the point predictions (or distribution parameterisation) of hybrid models, because they sample along the loss landscape.

We used a greedy local search algorithm with a budget of 10 000 function evaluations (simulations with loss function value computations) as an optimisation algorithm. Greedy local search can converge quickly to the global optimum in unimodal settings, though it may get stuck in local optima in multimodal settings. In our experiments, the downside of local optima worked to our advantage, because it allowed us to check for the number of optima in loss landscape by determining whether the optimisation algorithm converged to different local minima. It also enabled us to cover a larger part of the parameter space than the 1000 instances included in our dataset, as part of the 10 000 function evaluation budget used for finding the optimum was spent on exploring the search space, further increasing the probability that, if there are local irregularities in the search space, they would be encountered along the way. For further details on the optimisation algorithm, we refer to Appendix A.1. The budget was sufficient for our experiments; this can be validated through the plots in Appendix B.1.1.

5.4. EXPERIMENTS

In this section, we will describe our experiments aimed at answering the chapter research questions of Section 5.1. A summary of the experiments and the chapter research questions they correspond to can be found in Table 5.2.

5.4.1. ILL-POSEDNESS CHARACTERISTICS (CRQ1)

Ill-posed problems are problems that do not meet the requirements of well-posedness, as defined in Section 2.2.2: a problem is well-posed if and only if i) there is a solution to the problem, ii) this solution is unique, and iii) the appropriate solution changes continuously with changes in the observations (no sudden jumps in the parameter space).

Therefore, the first characteristics to test for are whether the conditions of well-posedness are met in PROSAIL inversion. Our experimental setup for these tests are as follows.

A SOLUTION EXISTS.

The objective of PROSAIL inversion is to find a configuration $\hat{\boldsymbol{\theta}}$ that minimises the spectral loss $\mathcal{L}(\hat{\mathbf{x}}, \mathbf{x})$ (see Equation 2.1).

If there exists any solution $\hat{\boldsymbol{\theta}}$ minimising the loss function $\mathcal{L}$, this property is satisfied. In continuous problems without constraints or undefined operations (e.g., zero divisions), this property is trivially satisfied: if any configuration $\boldsymbol{\theta}$ has a valid output $M(\boldsymbol{\theta})$, there will be at least one configuration minimising $\mathcal{L}(M(\boldsymbol{\theta}), \mathbf{x})$. In PROSAIL inversion, assuming the ranges of the parameters have been set up correctly, this will always be the case.

THE SOLUTION IS UNIQUE.

There is no guarantee that the optimum to a PROSAIL inversion problem is unique. If there are multiple optima (modalities) in the parameter space that would explain the observed spectra equally well, or perfectly flat areas with exactly the same optimal loss, the inversion problem is ill-posed by violating property 2 of well-posed problems (unique solution). Therefore, we tested for multimodality in our first experiment.

Experiment 1 (E1): In this experiment, we iterated over instances in D, and performed iterated greedy local search (see Appendix A.1.2 for details) [64] 5 times with a random initialisation sampled uniformly (overriding the normal distributions described in Table 5.1, as this could bias the experiment), resulting in a new local optimum for every iteration. For every instance, we computed the largest distance between any pair of optima out of the 5 optima in the set. If the landscape is unimodal, the optimisers should all converge to the same point in the parameter space regardless of their initialisation, resulting in a low maximum distance. If the landscape is multimodal, the optimisers can converge to different points in the space, resulting in a large maximum distance.

If more instances contain greater distances than can be explained through minor approximation inaccuracies of the optimisation algorithm (because the optimum for continuous optimisation problems will generally not match the true

configuration exactly: $\hat{\boldsymbol{\theta}} \approx \boldsymbol{\theta}^{+}$), we can conclude that the landscape is likely multimodal, and the PROSAIL inversion problem is ill-posed.

SOLUTION CONTINUOUS WITH OBSERVATIONS.
Unlike the previous two characteristics, which focused on the optimal solutions to the problem, this characteristic describes the underlying loss landscape. Intuitively, well-posed problems should not have sudden 'jumps' in their solution space: if the inputs (in this case: spectra) shift by a certain amount, the shift observed in the outputs (in this case: optimal solution) should be proportional to this shift. Concretely, for any input spectrum $\mathbf{x}$ and its associated optimum $\hat{\boldsymbol{\theta}}$, given another spectrum $\mathbf{x}'$ and its associated optimum $\hat{\boldsymbol{\theta}}'$, the new optimum $\hat{\boldsymbol{\theta}}'$ should converge to the original optimum $\hat{\boldsymbol{\theta}}$ as the new spectrum $\mathbf{x}'$ approaches the original spectrum $\mathbf{x}$. If this property is not satisfied, errors in some parts of the parameter space could become unpredictable, as negligible inaccuracies could still result in a large error due to a 'jump' through the space.

Experiment 2 (E2). We tested for this property by (deterministically) mapping the observed spectrum $\mathbf{x}$ into a perturbed spectrum $\mathbf{x}'$ by perturbation levels β of -10%, -1%, -0.1%, 0%, 0.1% 1% and 10% of the mean band value ($\mathbf{x}'^{b} = \mathbf{x}^{b} + \beta * \overline{\mathbf{x}^{b}}$), and computing the optimum. We compared the change between perturbations and their associated change in the optima. If the distance between the perturbed optima and the original optima converges to 0 as the perturbation intensity approaches 0, PROSAIL inversion likely meets the continuous input-output relationship requirement of well-posedness.

5.4.2. CAUSES OF ILL-POSEDNESS (CRQ2)

When aiming to understand the source of ill-posedness in parameter estimation through PROSAIL inversion, it can be beneficial to disentangle the parameter estimation task from the (PROSAIL) model inversion task.

In parameter estimation, the objective is to obtain an estimate $\hat{\boldsymbol{\theta}}$ of the true configuration $\boldsymbol{\theta}^{+}$ that is as close as possible to the real configuration values, such that a estimation loss function $\mathcal{L}_R(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^{+})$ (e.g., mean squared error) is minimised. Convsersely, PROSAIL inversion is a model inversion problem, and solving this model inversion problem is one of the methods to obtain parameter estimation estimates. In model inversion, the objective is to obtain an optimal configuration $\hat{\boldsymbol{\theta}}$ for which its simulated spectrum $M(\hat{\boldsymbol{\theta}})$ matches the observed spectrum $\mathbf{x}$ as well as possible, minimising the spectral loss function value $\mathcal{L}\big(M(\hat{\boldsymbol{\theta}}), \mathbf{x}\big)$. The implicit assumption here is that the $\hat{\boldsymbol{\theta}}$ found through PROSAIL inversion corresponds to the $\hat{\boldsymbol{\theta}}$ of the parameter retrieval problem.

Ill-posedness on the PROSAIL inversion problem indicates that a unique solu-

tion cannot be reliably found (e.g., due to parameter non-identifiability through compensation effects among parameters or spectral ambiguity, particularly in the infrared bands of Sentinel-2), or that the loss landscape is non-continuous. In contrast, ill-posedness on the parameter retrieval problem can indicate that the retrieval problem may be underdetermined or ill-defined, due to the information contained in the spectral data $\mathbf{x}$ being insufficient to uniquely determine $\hat{\boldsymbol{\theta}}$, or because there is a mismatch between real-world measured data and the input data expected by PROSAIL. If PROSAIL inversion is ill-posed, the parameter retrieval using it is also ill-posed, but PROSAIL inversion is not necessarily ill-posed if parameter estimation is. This is a highly relevant distinction because, if the parameter estimation is ill-posed, but not the PROSAIL inversion, this implies that tweaks to the inversion techniques (e.g., LUT-based hybrid models) or fully data-driven approaches, would still suffer from ill-posedness. In this vein, we hypothesise three possible main causes of the ill-posedness experienced by practitioners when performing parameter estimation through PROSAIL inversion.

First, as model inversion is often assumed to be an ill-posed problem (see, e.g., Darvishzadeh et al. [203] and Verrelst et al. [35, 204]), the PROSAIL inversion problem may indeed be ill-posed, which we already test for in E1 and E2. However, even if the PROSAIL inversion problem itself is not ill-posed, the overarching parameter estimation problem could still be. It is possible that noise and uncertainty in the observed data are causing the uncertainty of solutions for different instances to overlap, resulting in ill-posedness for the inference task. Moreover, it is possible that there are observable spectra for which there are no realistic solutions, for example, due to limitations of the scope of the simulation model, or due to the effects of spectral mixing [38].

Therefore, in addition to PROSAIL inversion itself, we consider two possible alternative causes through which parameter estimation through PROSAIL could be ill-posed: *noise combined with ill-conditionedness*, where small amounts of spectral noise can overpower the signal of matching simulated and observed spectra, and *spectral mixing*, where the observed spectra originate from multiple heterogeneous source spectra, whose combination may not correspond to a meaningful configuration. These characteristics prominently differentiate real-world settings from idealised simulation settings, making them appealing candidates to evaluate. We performed duplicate versions of Experiments 1 and 2 for each of these conditions, to verify that the well-posedness of the PROSAIL inversion still holds, even in these changed conditions. Furthermore, since there is no guarantee that our list of possible cuases of ill-posedness was exhaustive, we included experiments on real-world Sentinel-2 data (details can be found in Appendix A.2). If the patterns hold even for real-world data, this means that PROSAIL inversion is not the source of the ill-posedness.

We will explain our experiments to test these possible causes in the following.

NOISE AND CONDITIONING

Noise on the spectral observations, for example, through interference on or sensor limitations of the spectrometer used, has previously been found to have a strong impact on parameter estimation performance [50]. When there is noise on the observations $\mathbf{x}$, the spectrum is perturbed into a new position $\mathbf{x}'$ in the spectral space. When this happens, the optimal solution $\hat{\boldsymbol{\theta}}$ will shift away from the true configuration $\boldsymbol{\theta}^+$ (we later show examples of this in Figure 5.3), because the loss function $\mathcal{L}$ to minimise is now considering the perturbed spectrum $\mathbf{x}'$ instead of the noise-free version $\mathbf{x}$. Since the noise per instance is unknown *a priori*, two instances i_1 and i_2, with highly similar spectra $\mathbf{x}'_{i_1} \approx \mathbf{x}'_{i_2}$, could have entirely dissimilar true solutions $\boldsymbol{\theta}^+_{i_1} \not\approx \boldsymbol{\theta}^+_{i_2}$, because their original, dissimilar noise-free spectra $\mathbf{x}_{i_1} \neq \mathbf{x}_{i_2}$ were pushed together through unpredictable noise. This problem can be exacerbated depending on the level of *conditioning* of the problem, which can be interpreted as the sensitivity of the loss landscape to perturbations to the spectral observations [205, 206].

Experiment 3 (E3). We tested for this property in Experiment 3 by perturbing dataset D by adding various levels β of randomly sampled Gaussian noise in contrast to the deterministic perturbations of E2) at 1%, 2%, 5%, 10% and 20% of the mean band value $\overline{\mathbf{x}_b}$ in D) to the spectral data per band b: $\mathbf{x}'_b = \mathbf{x}_b + \mathcal{N}(0, \beta * \overline{\mathbf{x}_b})$. We then computed the average shift of the optima between the noise-free version of the data, and the noisy versions thereof. If this shift is high, especially for relatively low amounts of noise (indicating ill-conditionedness), any point in the parameter space that an optimum could have shifted from, can be considered a potential true solution to the problem. In this case, noise on the observed spectra can be considered a source of ill-posedness for the parameter estimation problem.

SPECTRAL MIXING

In real-world applications, spectral mixing is effectively inevitable. PROSAIL can only model a single set of parameters (though some of the structural parameters are already aggregates), thereby assuming a homogeneous vegetation cover for the entire area covered by a pixel (which can be interpreted as, for example, a mean of the vegetation types in the area). Spaceborne observations by the Sentinel-2 satellite cover, at best, an area of $100m^2$, which may contain a combination of several highly diverse vegetation types, or even land cover not related to vegetation (such as buildings or geological features). There is no guarantee that, if the spectra from different vegetation sources are mixed at a certain proportion, their optimal parameter values would consist of, e.g., a weighted average with the same proportional representation. Therefore, the optimum for an observed spectrum, when

mixed, may not correspond to a meaningful parameter configuration.

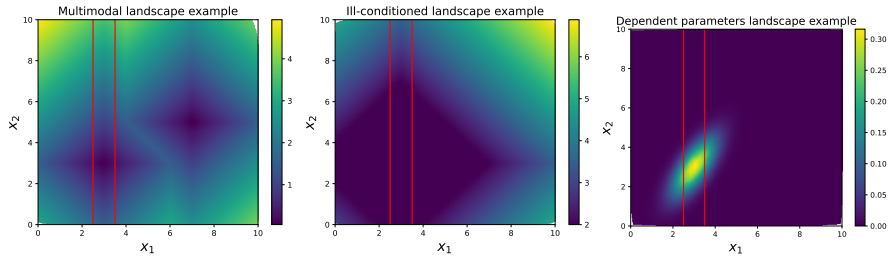
Experiment 4 (E4). In Experiment 4, we tested whether a clear solution can still be found for observations where spectral mixing has occurred. We modified the LUT generation procedure to generate new instances as a weighted combination of three randomly sampled configurations and their simulated spectra, parameterised by the randomly sampled weight parameters α_1, α_2 and α_3 (where $\alpha_1 + \alpha_2 + \alpha_3 = 1$). This results in three distinct, independent spectra $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$, simulated from their true parameter configurations $\boldsymbol{\theta}_1^+$, $\boldsymbol{\theta}_2^+$ and $\boldsymbol{\theta}_3^+$, that are combined into a single mixed spectral observation $\mathbf{x}' = \alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2 + \alpha_3\mathbf{x}_3$. We then ran our optimisation setup for $\mathbf{x}'$ to find the predicted configuration $\hat{\boldsymbol{\theta}}$, and compared this to the individual true configurations $\boldsymbol{\theta}_1^+$, $\boldsymbol{\theta}_2^+$ and $\boldsymbol{\theta}_3^+$, as well as their weighted mean $\alpha_1\boldsymbol{\theta}_1^+ + \alpha_2\boldsymbol{\theta}_2^+ + \alpha_3\boldsymbol{\theta}_3^+$. In parameter estimation settings, the weighted mean of configurations is a likely target quantity to predict for mixed spectra.

If the estimation performance for all these cases (particularly the mean) is worse than the performance in cases without spectral mixing, spectral mixing can be considered a cause of ill-posedness in parameter estimation. The parameter estimation problem would then become ill-defined, since the spectrum no longer corresponds to any single configuration, thereby violating characteristic 1 (a solution exists) of well-posed problems, despite being trivially satisfied for PROSAIL inversion itself. It is also possible that the problem would violate characteristic 2 (the solution is unique) of well-posed problems, because the underlying mechanisms of the spectral mixing can be considered a type of random noise. In these cases, it may be advisible for future work to further explore the impact of spectral unmixing techniques [38, 207, 208] on parameter retrieval through multispectral data.

5.4.3. IMPACT OF RANGE CONSTRAINT PRIORS (CRQ3)

A commonly used method for reducing the ill-posedness of PROSAIL inversion is the addition of priors in the form of range constraints, which was previously found to improve performance [57]. We consider three possible (not mutually exclusive) mechanisms through which the ill-posedness of PROSAIL inversion could be reduced by adding range constraint priors: i) excluding competing optima, ii) reducing the maximal magnitude of errors, and iii) parameter dependency. We show abstract examples of these possible mechanisms in Figure 5.2.

If the loss landscape is multimodal (see Section 5.4.1 for details), the mechanism by which the ill-posedness would be reduced would be intuitive: restricting the ranges of parameters to known feasible parts of the parameter space would rule out optima in different parts of the parameter space, making it more likely to converge to the correct optimum. This mechanism is illustrated in Figure 5.2a.



(a) Multimodality (mechanism i))

(b) Ill-conditioning (mechanism ii))

(c) Dependency (mechanism iii))

Figure 5.2: Examples of landscapes and the possible mechanisms of reduced ill-posedness through range constraint priors on the x_1 variable (red vertical lines). In Figure 5.2a, the constraint on the x_1 variable excludes the second optimum at (7,5), thereby reducing ill-posedness through mechanism i). In Figure 5.2b, there are many points with a value very close to 2 (plotted simply as 2 for illustrative purposes), but the range constraint prevents errors larger than 1 (the width of the range) through mechanism ii); without the constraint, the model may have predicted points like (4,6). Finally, in Figure 5.2c (where, for plotting convenience, we are maximising instead of minimising), due to the dependency between x_1 and x_2, the range constraint prior on x_1 reduces the viable range for x_2 from about [1,5] to about [1.5,4.5] through mechanism iii).

Meanwhile, if the landscape is unimodal, but it is ill-conditioned (see Section 5.4.2 for details) with a wide range of values evaluating to similar losses, adding range constraints could shrink the range of potential points with similar loss values that an optimiser could converge to for noisy spectral observations. In this case, performance would be improved because the degree to which predictions can be wrong would be limited; this mechanism is illustrated in Figure 5.2b. However, in this case it is possible that parameter estimation performance does not significantly differ from random sampling within the specified prior ranges; if this is the case, performing any inference at all would not provide any additional posterior information over the already available prior knowledge, which may reduce the appeal of using priors to reduce ill-posedness for parameter estimation problems.

Finally, if there are dependencies between parameters (for example, more possible C_{ab} combinations for low LAI), constraining the ranges of some parameters may result in reduced ranges for other parameters as well, as illustrated in Figure 5.2c. In the example, the x_2 parameter has a range of about 1 to 5. However, its lowest values can only be reached if the value for x_1 is also low, and its highest values are only reachable if the value for x_1 is high. By adding a range constraint to x_1 as prior knowledge, the viable range for x_2 is also reduced to a range from about 1.5 to 4.5, thereby reducing the ill-posedness.

Experiment 5 (E5). We experimentally gauged in E5 whether the impact of

adding priors comes from mechanism i), ii) or iii). If our results for the multimodality experiment described in Section 5.4.1 show a multimodal landscape, we can assume that range constraint priors reduce ill-posedness through mechanism i) (excluding competing optima). To test for mechanisms ii) and iii), we performed optimisation on the instances in D (with 10% Gaussian noise on spectral observations as described in Section 5.4.2, to ensure the problem is ill-posed), setting priors on LAI with a range interval around the true value at 0%, 10%, 30%, 50% and 100% of the total LAI range. We then repeated this procedure by using random sampling for LAI within its range interval, as a baseline measurement.

If the parameter estimation performance, measured only on LAI, is better for tighter range intervals than for larger intervals, it is likely that the ill-posedness was reduced through mechanism ii). However, if there is no difference with the random sampling-based baseline, though the prior knowledge would improve performance, it is not synergistic with the estimation method, and rather replaces it entirely, as the posterior equals the prior. Finally, if the parameter estimation performance for *other* parameters is significantly better with tighter range constraints for LAI than with a looser or absent constraint, it is likely that reducing the range of LAI also reduced the viable ranges of other parameters, making mechanism iii) more likely.

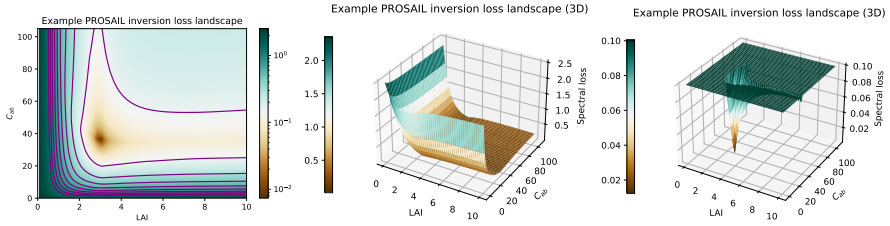
5.5. RESULTS

In the following we will, unless otherwise stated, analyse the results for the PMAE loss function. Due to their patterns being largely the same as the results for PMAE, the results for SAM can be found in Appendix B.1.3.

5.5.1. CRQ1: ILL-POSEDNESS CHARACTERISTICS

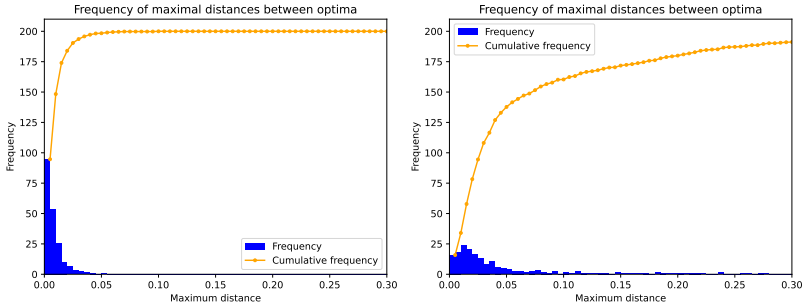
Experiment 1. We visualised an example loss landscape for PROSAIL inversion in Figure 5.3. In this example loss landscape, the landscape appears to be unimodal, as can be seen in Figure 5.3a, with a large plateau of nearly identical spectral loss values surrounding that optimum that can be seen in Figure 5.3b. The optimum itself can only be seen when limiting the range of spectral losses by a substantial margin (in Figure 5.3c: 2.5 to 0.1, a reduction of 96%). Therefore, based on this example, PROSAIL inversion appears to be unimodal, but ill-conditioned: given the small margins of loss function values separating the optimum from the plateau, a small perturbation to the input spectrum would likely have a large impact on the output prediction.

To generalise this observation to a general pattern over all instances, we aggregated the results of Experiment 1 over instances by plotting a histogram of the maximum distances in Figure 5.4. As the figure shows, the optimisation algorithm



(a) Loss landscape contour plot (b) Loss landscape 3D plot (c) 3D plot with clipped losses

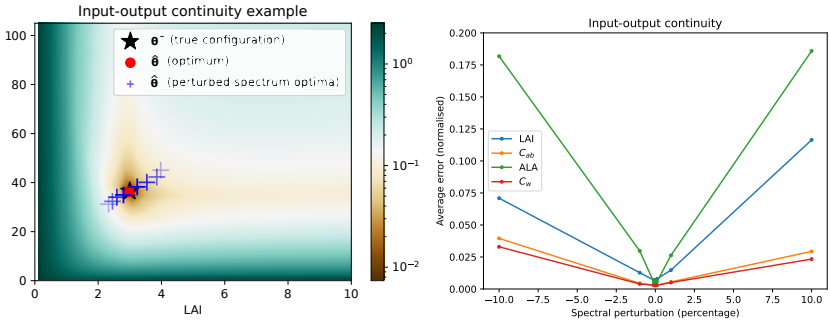
Figure 5.3: Example of a loss landscape for PROSAIL inversion, as a 2D plot with contour lines at a log scale emphasising small differences for low values (5.3a), a 3D plot (5.3b), and a 3D plot with clipped loss values (5.3c). The landscape is unimodal, meaning only a single point in the landscape locally minimises the spectral loss function. However, for most of the loss landscape, the differences in the spectral loss are very small, and only get larger as the parameters near a value of 0 (as indicated by the concentration of contour lines in Figure 5.3a). In Figure 5.3b, there appear to be no differences in loss values within a large plateau of configurations with similar spectral losses. Only when artificially clipping the loss values to set all losses > 0.1 to 0.1 can the global minimum be seen in the 3D loss landscape (shown in Figure 5.3c).



(a) PMAE (b) SAM

Figure 5.4: Histogram and cumulative frequency of the maximal distances between 5 optima (initialised uniformly randomly) obtained using iterated local search, per instance. The bins on the x-axis (restricted to 0–0.3 instead of the full 0–1 range for better visibility) represent distance in the parameter space as a proportion of the total parameter range . For example, for the LAI parameter with a range of 0 to 10, a proportional distance of 0.1 represents an LAI difference of 1. The results for the PMAE loss function can be found in in Figure 5.4a, and the results for SAM can be found in Figure 5.4b. Since almost all 200 maximum distances are contained within the first bins with the smallest distances, the restarted, randomly initialised optimisation procedure appears to converge to the same local optimum every time, indicating unimodality.

almost always converged to the same optimum after being randomly restarted, indicated by the heavy skew toward the lowest value bins. For this experiment, we



(a) Example visualisation

(b) Shift per perturbation

Figure 5.5: The continuity of the output for PROSAIL inversion (predicted configuration $\hat{\boldsymbol{\theta}}$) with respect to perturbations to the input (spectrum). Figure 5.5a shows an example of a single instance (with only the LAI and C_{ab} parameters free), with colours representing its loss values, forming a loss landscape. The true configuration $\boldsymbol{\theta}^+$ is visualised as a star, which the noise-free optimum $\hat{\boldsymbol{\theta}}$ (the red dot) matches nearly perfectly. The blue +-signs represent the shifted versions of the optimum $\hat{\boldsymbol{\theta}}$, where points with lower opacity represent a larger perturbation to the spectral input. Figure 5.5b aggregates this phenomenon (mean) over all 1000 instances, normalised to a 0-1 range based on the bounds of the parameter range, showing that it is a consistent pattern.

include the results for SAM (Figure 5.4b) because, while they were similar to those of PMAE (Figure 5.4a), the skew was slightly less extreme, though most maximal differences were still within a 5% distance of the parameter ranges. Overall, these results provide empirical evidence that there exists only a single local and global optimum in the loss landscape of PROSAIL inversion, leading to a unimodal problem. Therefore, PROSAIL inversion meets requirement ii) of well-posedness.

Experiment 2. The continuity between inputs (spectra) and outputs (optimal solution) for PROSAIL inversion tested in Experiment 2 can be seen in Figure 5.5. In the example from Figure 5.5a, the blue points marked by a +-sign form a clear line through the parameter space, where the points with a lower perturbation (high opacity) are closer to the optimum for the unperturbed spectrum, while the points with a higher perturbation (low opacity) are further removed from this original optimum. This indicates that, in this example, the location of the optimum (output) shifts smoothly and continuously with changes to the input (observed spectrum), thereby meeting requirement iii) of well-posedness. As Figure 5.5b shows, this pattern continues to hold when aggregated over all 1000 instances: for all parameters, a smooth, convex shape can be observed, indicating that optima converged to the optimum of a specific spectrum as their spectra approached this spectrum. If the property of input-output continuity would not have been met, there would be in-

stances where small perturbations would result in sudden, large jumps across the parameter space, which would not result in the parabola-like shapes in Figure 5.5b that we observe. Therefore, we conclude that PROSAIL inversion meets requirement iii) of well-posedness.

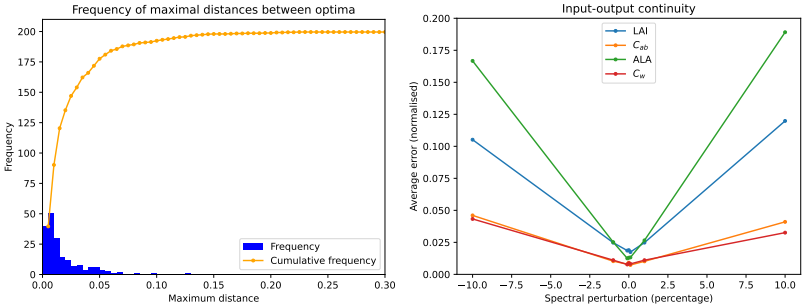
In conclusion for CRQ1, the results from Experiments 1 and 2 imply that PROSAIL inversion is not an ill-posed problem: there is always a solution, this solution is unique, and the output moves continuously with respect to the input.

5.5.2. CRQ2: ILL-POSEDNESS CAUSES

Our results for the additional runs of E1 and E2 for noisy, spectrally mixed and real-world data can be found in Figure 5.6. As the figure shows, the PROSAIL inversion loss landscape continues to show unimodal patterns, with a large skew toward convergence to the same optimum every time. The input-output continuity likewise remains intact, with the exception of real-world data in Figure 5.6f. This shows that some PROSAIL inversion loss landscapes may sometimes violate requirement 3 of well-posedness for real-world spectral data, although there would still only be a single optimum in this case (though it would be harder to find). It is also worth noting is that, while the continuity for the PROSAIL inversion loss landscape remained intact, the normalised error rates for no perturbations (0%) were often larger than 0. This suggests that the best fitting simulated spectrum does not perfectly match the observed real-world spectrum, thereby potentially resulting in a violation of property 1 of well-posedness (a solution exists) for the parameter retrieval problem (but not the PROSAIL inversion problem).

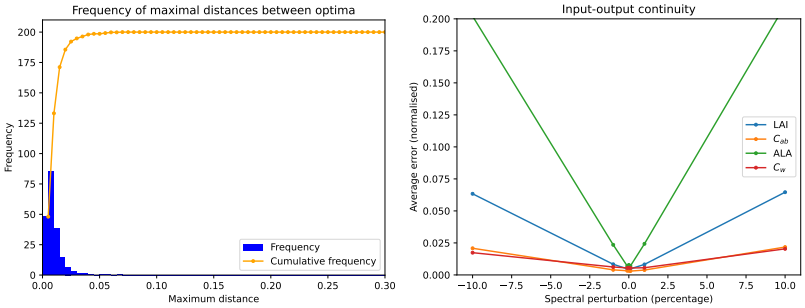
Given our findings that the PROSAIL well-posedness appears to hold for noisy, spectrally mixed and real-world data, we will test for the impact of our hypothesised causes of parameter retrieval ill-posedness in the following.

Experiment 3. For Experiment 3, we visualised the impact of random noise in the spectral inputs on retrieval performance in Figure 5.7, with an example loss landscape visualised in Figure 5.7a. A prediction error on the optimum due to noise on the observations can be interpreted as a 'shift' of the optimum $\hat{\boldsymbol{\theta}}$ in the loss landscape, away from the unknown true configuration $\boldsymbol{\theta}^{+}$, represented by the grey, magenta and blue markers in Figure 5.7a. As illustrated in the figure, each time a Gaussian noise was re-applied to an originally 'clean' spectrum (as described in Section 5.4.2), a new point in the parameter space globally minimised the spectral loss, for which we visualised 10 examples per noise level in the figure. The shifts were fairly small for 3% spectral noise, noticeable for 5%, and highly disruptive with large outliers (e.g., reaching the maximum LAI value of 10) for 10% noise – a realistic setting, as the Sen2Cor atmospheric correction algorithm alone can introduce substantial noise to a spectrum [209, 50, 210]. For example,



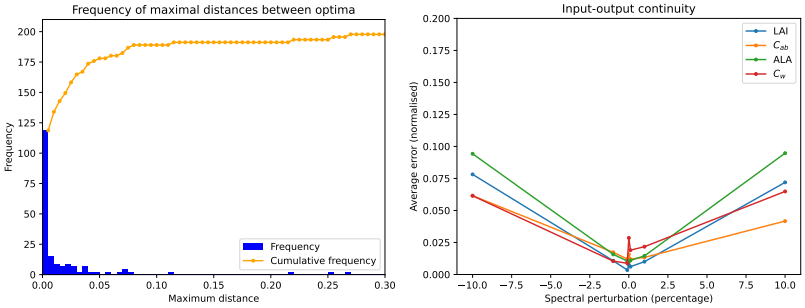
(a) Noisy, E1

(b) Noisy, E2



(c) Spectrally mixed, E1

(d) Spectrally mixed, E2

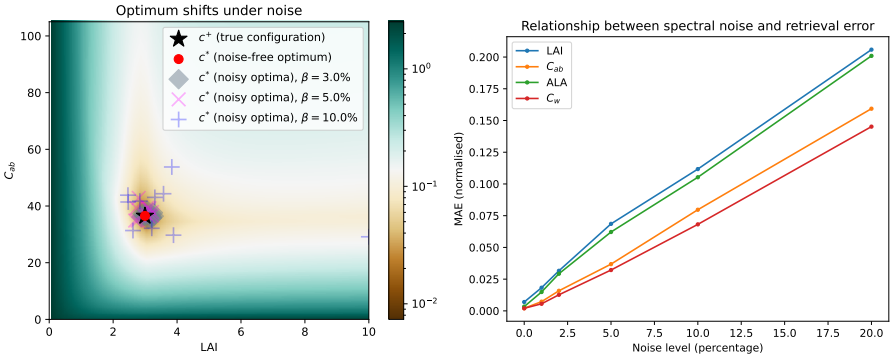


(e) Real-world, E1

(f) Real-world, E2

Figure 5.6: Repeat of Experiments 1 and 2 for noisy (5.6a and 5.6b), spectrally mixed (5.6c and 5.6d), and real-world (5.6e and 5.6f) data.

the RMSE for Sen2Cor per band reported by Sola et al. [209] represents around 14% to 20% of the average band values in our real-world dataset, with one outlier



(a) Example visualisation

(b) Performance per noise level

Figure 5.7: The impact of spectral noise on retrieval performance. Figure 5.7a shows an example of a single instance (with only the LAI and C_{ab} parameters free), with colours representing its loss values, forming a loss landscape. The true configuration $\boldsymbol{\theta}^+$ is visualised as a star, which the noise-free optimum $\hat{\boldsymbol{\theta}}$ (the red dot) matches nearly perfectly. The grey, magenta and blue markers represent the shifted versions of the optimum $\hat{\boldsymbol{\theta}}$, when Gaussian spectral noise at an intensity β of 3%, 5% and 10% is applied 10 times to the same instance (every repeat sampling a new, unpredictable noise term). Figure 5.7b aggregates this 'shifted optimum' phenomenon over all 1000 instances, showing that it is a consistent pattern, and the intensity of the shifts increases as the noise level increases.

5

at nearly 40%. Because, for any given result, we cannot know the degree to which this optimum has been noise-shifted away from $\boldsymbol{\theta}^+$, nor the direction in which it was shifted, any point on the loss landscape that the optimum $\hat{\boldsymbol{\theta}}$ could have shifted from, could be considered a potential solution to the inference problem, thereby violating requirement ii) of well-posedness.

In Figure 5.7b, we aggregated the numerical results and plotted the mean absolute error for parameter estimation (normalised as a proportion of the total possible range of the parameter, e.g., 0-10 for LAI) against the intensity of Gaussian noise added to the spectral observations. As might be expected, the mean absolute error increased with the noise added to the spectra; this is consistent with results reported by De Sa et al. [50]. In the results for PMAE shown in Figure 5.7b, the normalised loss for all parameters shows a linear relationship with the noise level. In the SAM results, which can be found in Appendix B.1.3, the patterns for ALA and LAI parameters were not linear, but instead sharply increased for low levels of noise, while increasing only marginally for higher noise levels. However, in all cases higher levels of noise resulted in higher parameter retrieval error rates.

We conclude that spectral noise seems to be a contributing factor to the ill-posedness of parameter estimation using spectral information, which can be the

Parameter	Normalised MAE target			
	$\alpha_1\boldsymbol{\theta}_1^+ + \alpha_2\boldsymbol{\theta}_2^+ + \alpha_3\boldsymbol{\theta}_3^+$	$\boldsymbol{\theta}_1^+$	$\boldsymbol{\theta}_2^+$	$\boldsymbol{\theta}_3^+$
LAI	0.117 ± 0.089	0.164 ± 0.151	0.158 ± 0.149	0.161 ± 0.151
C_{ab}	0.048 ± 0.048	0.125 ± 0.106	0.12 ± 0.101	0.121 ± 0.104
ALA	0.106 ± 0.134	0.236 ± 0.193	0.237 ± 0.196	0.238 ± 0.187
C_w	0.033 ± 0.045	0.077 ± 0.064	0.079 ± 0.065	0.083 ± 0.067

Table 5.3: Results for E4 on the impact of spectral mixing. Every cell represents the (normalised) MAE between the optima found for the mixed spectrum $\mathbf{x}'$ and the quantities listed in the columns. The first column represents the weighted mean of the true configurations of the constituent spectra $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$, while the other columns represent the MAE compared to these individual constituent configurations. This suggests that the solution for mixed spectra matches the weighted mean of the constituent configurations more closely than the configuration of any individual constituent spectrum.

case even when PROSAIL inversion itself is well-posed. This effect is likely exacerbated by the ill-conditionedness found in Figure 5.3: relatively small perturbations on the spectral observations caused by noise could result in large jumps across the loss landscape, given the large plateau-like region of the loss landscape where noise could relatively easily overpower the signal of the loss function.

Experiment 4. In this experiment, we tested whether spectral mixing could be an additional explanation for the ill-posedness experienced when performing parameter estimation. The results for this experiment can be found in Table 5.3. The cells in this table contain normalised mean absolute error rates for different parameters (rows), when the predictions in $\hat{\boldsymbol{\theta}}$ are compared to 4 different types of 'true' values (columns): the weighted mean of the configurations that correspond to the 3 mixed spectra that formed the observations, and the parameter values of these 3 configurations themselves.

As Table 5.3 shows, there is a notable drop in parameter estimation performance, especially for LAI, compared to the performance expected for noise-free data that was not mixed. For example, the loss values for the weighted mean of the mixed configurations in Table 5.3 appear comparable to the loss values for spectra with 2.5% to 10% Gaussian noise applied to them in Figure 5.7b, while the expected loss for non-mixed noise-free spectra based on this figure is close to 0. This decreased performance suggests that the optimum for a mixed spectrum, while consistently converging to a target quantity most closely related to the weighted mean of the optima of the constituent spectra, and thereby retaining some fidelity, does not perfectly align with such a target. The behaviour of the optimal outcomes $\hat{\boldsymbol{\theta}}$ in response to a linear mixture of input spectra appears to be governed by complex, non-linear and unpredictable mechanisms. While these complex mechanics are unknown, if a user is interested in retrieving, e.g., the weighted mean of the

Parameter	LAI prior range constraint size				
	0%	**10%**	**30%**	**50%**	**100%**
LAI (uniform)	[1]0.0 ± 0.0	[2]0.473 ± 0.289	[3]1.413 ± 0.863	[4]2.214 ± 1.37	[5]2.978 ± 2.124
LAI	[1]0.0 ± 0.0	[2]0.479 ± 0.341	[3]0.808 ± 0.835	[4]0.955 ± 1.133	[4]0.972 ± 1.167
C_{ab}	[1]7.737 ± 11.865	[2]8.143 ± 11.575	[3]8.262 ± 11.559	[2]8.235 ± 11.358	[3]8.23 ± 11.401
ALA	[1]5.978 ± 5.95	[2]8.151 ± 9.201	[3]8.795 ± 9.488	[4]8.943 ± 9.685	[3]8.868 ± 9.373
C_w	[1]0.003 ± 0.006	[2]0.004 ± 0.006	[3]0.004 ± 0.005	[4]0.004 ± 0.005	[3]0.004 ± 0.005

Table 5.4: Mean absolute error rates for parameter retrieval performance for the four different parameters (rows), with columns representing the interval size of a range constraint prior on LAI (with 100% covering the full original parameter range). The 'LAI (uniform)' row represents the performance of estimating LAI through uniform random sampling, while in other columns, performance is acquired through optimisation. In each row, the prior range size in a column marked with a lower number (e.g., [1]) retrieves a parameter significantly better (significance level $\alpha = 0.05$) than one with a higher number (e.g., [2]). Adding range constraint priors on LAI greatly improved LAI retrieval performance, while also improving ALA (but not C_{ab} and C_w) retrieval performance.

true parameter configurations, there are many points around the optimum that could correspond to this desired quantity. Therefore, the ill-posedness caused by spectral mixing appears similar to that of random Gaussian noise in E3, as long as the underlying non-linear mechanics remain unpredictable.

Although the magnitude of this effect is relatively small for an extreme type of spectral mixing (fully independently generated configurations), these results suggest that spectral mixing is a contributing factor to ill-posedness for parameter retrieval by violating at least one of characteristics 1 and 2 of well-posedness.

Out of the types of true values we compare to in he columns of Table 5.3, the parameter estimation performance was best for the mean of the configurations from the 3 generating configurations $\boldsymbol{\theta}^+$ for all parameters, indicating that the optimum $\hat{\boldsymbol{\theta}}$ of a mixed spectrum $\mathbf{x}' = \alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2 + \alpha_3\mathbf{x}_3$ corresponds most closely to the weighted mean of their constituent configurations $\alpha_1\boldsymbol{\theta}_1^+ + \alpha_2\boldsymbol{\theta}_2^+ + \alpha_3\boldsymbol{\theta}_3^+$, albeit with a drop in performance compared to non-mixed spectra.

In conclusion for CRQ2, we consider the impact of spectral noise to be the most likely cause of the ill-posedness experienced in parameter estimation from multi-spectral data, with a possible additional contribution by spectral mixing.

5.5.3. CRQ3: IMPACT OF RANGE CONSTRAINT PRIORS

Experiment 5. In this experiment, we were interested whether prior information in the form of range constraint priors can indeed reduce ill-posedness, and if so, through what mechanisms it is effective. Given our results for Experiment 1 in Figure 5.4, showing PROSAIL inversion to be a unimodal problem, mechanism i) (excluding competing optima) is unlikely to be a big factor. The results for mechanisms ii) (reducing the maximal magnitude of errors) and iii) (parameter depen-

dency) are shown in Table 5.4.

As the table shows, the parameter estimation error for LAI increased as the size of the prior range size increased. For prior range sizes up to 10% the performance was very similar to that of uniform sampling within the range, indicating that the posterior after performing inference was equal to the prior knowledge. The range of ill-posed solutions likely extended beyond this prior range; therefore, introducing a range constraint reduced the ill-posedness. In this case, performing parameter estimation would not add any additional information. However, for larger range constraints, performing parameter estimation resulted in better performance than uniform sampling in the range interval, while the performance also deteriorated with larger intervals. These results indicate that mechanism ii) is a likely factor in the efficacy of range constraint priors for improving ill-posed parameter estimation performance, usually without the prior knowledge replacing the inference method outright.

For mechanism iii), as Table 5.4 shows, the parameter estimation performance of C_{ab} and C_w do not appear to be strongly affected by the LAI prior range interval size, while the parameter estimation performance of ALA appears to be correlated due to its errors increasing with the increased prior range interval, though this was mainly the case for highly precise ($< 10\%$) intervals. Therefore, while it is likely that mechanism iii) plays some part in improving parameter estimation performance, its impact may be limited.

We conclude for CRQ3 that range constraint priors help improve parameter estimation performance primarily through mechanism ii), but mechanism iii) may also contribute in some cases.

5.6. DISCUSSION

In this chapter, we set out to analyse the ill-posedness of PROSAIL inversion (CRQ1), to establish possible causes for the ill-posedness experienced by domain practitioners (CRQ2), and to confirm that a commonly used strategy for reducing ill-posedness, adding range constraint priors, indeed reduces the ill-posedness of the problem (CRQ3).

5.6.1. SUMMARY OF RESULTS

Our results for **Experiment 1** (Figure 5.4) and **Experiment 2** (Figure 5.5) show that PROSAIL inversion itself is unlikely to be ill-posed, since it meets the criteria of a well-posed problem. Our analysis focused on the estimation of LAI, leaf angle, chlorophyll content and water content; however, it is possible that the estimation of other parameters, notably parameters with a limited impact on the spectral output of PROSAIL, could still be ill-posed.

Our results for **Experiment 3** (Figure 5.7) indicate that spectral noise can cause the predicted configuration for a problem instance to shift away from the true configuration. Since this noise is unknown *a priori*, any configuration in the parameter space that a prediction for an observed spectrum could have shifted from is a potential true solution to the problem instance, thereby making the inference problem ill-posed. Spectral mixing also appears to contribute toward ill-posedness (**Experiment 4**; Table 5.3).

Given these results, it appears that the ill-posedness experienced by domain practitioners performing parameter estimation does not stem from the PROSAIL radiative transfer model, but rather from the inherent limitations of the overarching parameter estimation task and the information contained in the spectral data. This would carry strong implications for future work in this field, as multispectral data would not contain sufficient information to reliably retrieve the parameters of interest.

Finally, our results for **Experiment 5** indicated that the use of range constraints improved parameter estimation performance more for smaller range intervals, and less for larger intervals, while performing better than random sampling in the range interval after interval sizes of 10%. This indicates that mechanism ii) (reduced magnitudes of errors) is likely to be a factor.

In contrast, mechanism iii) (inter-parameter dependencies) likely has a modest contribution to the efficacy of range constraint priors to reduce ill-posedness. This is a surprising result, given that biophysical parameters are highly likely to affect one another in nature. A possible explanation for this result may be that models like PROSAIL can perform a simulation for any configuration, regardless of its biological plausibility, and the relationship between parameters (like the example in Figure 5.2c) depends solely on whether they amplify one another's effects on the spectrum. This is not necessarily related to the degree of co-occurrence of certain parameter settings in nature. In fact, such natural relationships may be promising candidates to further constrain the search space.

5.6.2. FUTURE WORK

There are several remaining challenges in understanding the ill-posedness of the parameter estimation problem. First, performing analyses on labelled real-world data could evaluate whether fully data-driven approaches, not reliant on PROSAIL, would indeed be susceptible to the same ill-posedness, as our experimental results suggest. The data collection involved would be a major obstacle to testing this hypothesis in practice, as it would require extremely accurate ground-truth data for all parameters simultaneously, as well as effectively noise-free spectral observations to estimate the impact of noise. Future work in this direction, if the data

requirements were sufficiently met for such a study to become feasible, would be valuable.

Though we were able to test two hypothesised causes of ill-posedness and mechanisms for the impact of range constraint priors, we cannot be certain that our list of hypotheses is exhaustive. It is possible that there are other factors involved that play even larger roles. Therefore, beyond the results of this study itself, we stress in particular the perspective we adopted to test our hypotheses through a careful consideration of the loss landscapes underlying the inversion problem. We invite other scholarly work, should others come up with a set of additional hypotheses, to similarly test these through systematic computational experiments on the loss landscape where appropriate.

Our experimental results suggest that ill-posedness could not be overcome solely with algorithmic contributions that improve the identification of the parameter configuration that best fits the observations (e.g., through better optimisation or by performing more efficient learning). Even if the optimum were predicted perfectly accurately every time, our findings indicate that this optimum itself is not a perfect target. Statistical uncertainty quantification techniques may help capture these uncertainties, although these should be selected with care. Since the ill-posedness appears to originate in the parameter estimation problem itself, rather than any particular property of the PROSAIL model and its inversion, it is likely that purely data-driven methods would be similarly affected by the ill-posedness.

Instead, we would encourage explorations into novel contributions for biophysical parameter estimation from a data-centric perspective, focusing on increasing the information content in the observed data (for example, through incorporating additional data sources, higher resolution data, hyperspectral data, or temporal autocorrelation), regardless of whether the method used for mapping the observations to estimated parameters includes PROSAIL inversion or not. For example, the use of specialist hybrid models, that have been trained on a subset of data that is relevant to a study area (either through manual selection of training data, or through the use of active learning heuristics combined with a study area-specific validation set) has already seen successful applications of PROSAIL inversion [52, 46]. Perhaps an automatic generation or selection of appropriate specialist models might be a fruitful next step to explore. Alternatively, advances in deep learning techniques may be able to contribute toward 'denoising' the observed noisy spectra through the application of, e.g., denoising autoencoders [211]. If noisy spectral data could be reduced to the equivalent of the noise-free simulated data, this would greatly alleviate the ill-posedness caused by noise; we discuss this direction further in Section 7.2.2.

5.7. CONCLUSIONS

Our systematic analysis of PROSAIL inversion and its ill-posedness indicate that pure PROSAIL inversion for the estimation of the often studied parameters leaf area index, chlorophyll content, average leaf angle and water content, meets all the requirements of a well-posed problem: there is a solution, the solution is unique, and the output moves continuously with respect to the input. Given the ill-posedness experienced by domain practitioners, these results suggest that PROSAIL inversion itself is not an ill-posed problem, but rather the associated parameter estimation problem when relying on multispectral data. This seems mainly caused by spectral noise, while spectral mixing also appears to play a role.

Finally, we found that range constraint priors can alleviate the ill-posedness of the parameter estimation problem through a reduction of the magnitude of possible errors, in addition to possible indirect effects through inter-parameter dependencies.

A problem can only be effectively addressed if it is well understood. Currently, much focus in parameter estimation work using PROSAIL inversion has focused on method-centric approaches, for example, by efficiently sampling points using active learning, or on training specialised hybrid models with reduced ranges for specific study sites. We hope that, building on the results reported here, future work can also more efficiently explore novel solutions that might improve the viability of parameter estimation when fewer assumptions (e.g., about the study site) can be made. In particular, we believe that the exploration of data-centric improvements, such as the automatic training or selection of specialist models for a given application area, spectral denoising or data fusion approaches, may be a fruitful endeavour.

Based on the findings in this chapter, it is unlikely that any methodological contribution can eliminate the ill-posedness of noisy inference problems outright, and instead, data-centric approaches aimed at improving the reliability or information content of the input data emerged as the most promising direction to reduce the ill-posedness. However, this type of contribution will be largely out of the control of practitioners making parameter estimations. Although fully eliminating ill-posedness may not be possible, we can still create methods that can be used to, e.g., quantify the severity of the ill-posedness on specific problem instances. For example, even if the loss landscape has two optima (unlike the unimodal PROSAIL inversion loss landscapes of Figure 5.4a), this may not be a big problem, if the distance between these points is negligibly small, while it could invalidate estimations when this distance is large. Therefore, in the next chapter, we will propose a novel method, eMMI, to enable this type of analysis. Using the insights gained in this chapter, the method will be based on the loss landscape of inference prob-

lems, and given the phenomenon of optimum shifts of Figure 5.7a, observation noise will take a central role in determining the possible solution set.

6

eMMI: ϵ-MANIFOLDS OF POTENTIAL SOLUTIONS FOR NOISY INFERENCE

In the previous chapter, we have learned key information about noisy inversion problems through our analysis of biophysical parameter estimation using EO data and PROSAIL inversion: observation noise plays a central role in making inference problems ill-posed, resulting in optimum shifts that appear to follow the loss landscape of the inference problem. In this chapter[1], we aim to leverage this knowledge to propose the concept of ϵ-manifolds, which contain all possible solutions to a model inversion inference problem, and a method, eMMI, to approximate these ϵ-manifolds. In doing so, we address RQ4: *How can we automatically extract the set of possible solutions to a noisy inference problem?*. Although our findings in Chapter 5 suggest that pure methodological contributions are unlikely to eliminate ill-posedness from noisy inference problems altogether, the concepts and method introduced in this chapter will address Challenge 2 by enabling novel types of analyses, enabling users to, for example, make a judgement on whether to trust their parameter estimations, based on the degree of ill-posedness of their specific problem instance.

6.1. INTRODUCTION

Inferring a model parameterisation from observations is a fundamental problem in many scientific domains. Such inference problems are considered *model inver-*

[1] The contents of this chapter are based on the article: Laurens Arp, Peter van Bodegom, Nguyen Dang, Holger H. Hoos, Alistair Francis, and Mitra Baratchi. (2025). *Inference from Noisy Observations through Model Inversion: Constructing ϵ-Manifolds of Potentially Valid Solutions.* Under review.

sion problems when there is a (simulation) model available to simulate observations from a hypothesised parameter configuration, such that model parameters are the target values to infer, and the observations consist of data that can be simulated by the model (e.g., observable variables or class labels). Model inversion-based inference problems are ubiquitous in many scientific fields, including AI. Within physical sciences, model inversion is often used when inferring the unobservable values of a set of physical parameters that resulted in an observed outcome [212, 213, 214, 215], whereas in simulation-based inference (SBI), a probabilistic simulation model is used as a likelihood function in a Bayesian inference setting, where the target parameters are often inferred posterior distributions, given the observed data [216, 217, 218]. Application areas reliant on model inversion include fluid dynamics [212, 213], astronomy and astrophysics [5, 6] and Earth sciences [214, 215]; in these fields, often large amounts of observational data are available, but few labelled examples, making it challenging to apply conventional supervised machine learning approaches to map observations to labels. Instead, physical simulation models are used to estimate these parameters. Within AI, we are often using machine learning to infer a target variable based on the observed features, and the training of machine learning models is itself a model inversion-based inference problem where the model parameterisation must be learned from the observed training data.

A simulation model cannot be used directly to infer the correct parameters from real observations. However, it can be used indirectly to evaluate the quality of a possible parameter configuration by comparing its simulated observations to the real observations through a loss function. Various approximation techniques can be used to infer model parameters. These include numerical optimisation [47, 48], specialised simulation-based inference algorithms [219, 218, 217] and training a machine learning model on simulated data [50, 220, 221].

Model inversion is a non-trivial problem, shown to be NP-hard if addressed using numerical optimisation [222]. It is often *ill-posed*, meaning that the inversion of a single observation can lead into multiple different solutions with equal, or highly similar, quality [57, 223, 224]. It can also be *ill-conditioned*, meaning that small perturbations in the observations cause large shifts in the optimal parameter configuration for a problem instance [205, 206, 225]. Moreover, real-world observations are nearly always noisy, due to limitations of sensing technologies that gather observations, inaccurate scenario specifications [226], or noisy data annotation, leading to ill-posedness of the inference problem. As a result, even if these methods reliably find the global optimum of an inference problem, the resulting configuration may not correspond to the 'true' parameter values that generated the observations. The optimum would thus explain the noisy observations rather than the true state of the system.

Due to these challenges, practitioners may quantify uncertainty by inferring a distribution over parameter values rather than making point predictions [216, 46, 54, 45]. However, statistical distributions usually assume certain properties (e.g., Gaussian parametric form, independence or stationarity). These properties are often not met, or they differ per instance and cannot be known *a priori*. If the inversion problem is ill-posed with multimodal or non-continuous distributions, values will likely not be centred around the mean. For example, for a model $y = \alpha^2$ with observation $y = 100$ and a domain $\alpha \in \mathbb{R}$, the solution for the parameter α could be 10 or -10, but not any values in between. Moreover, uncertainty quantification models a distribution over parameter values, not the degree to which the parameter values fit the observations. Therefore, configurations that could effectively explain the observations, but do not naturally appear in a data set, are unlikely to be included. This confines the applicability of uncertainty quantification methods to indicating statistical confidence on a prediction, while deeper analyses of the inversion problem itself, including solutions that do not occur naturally in the observational distribution, may be desired.

In this article, for the first time, we aim to retrieve the set of all solutions that can reasonably explain the observations in model inversion problem settings, forming a manifold we refer to as the *ϵ-manifold*. The intuition behind ϵ-manifolds is that they contain configurations for which the loss function value is sufficiently close to that of the optimal configuration to be considered a potentially feasible solution, given the level of noise ϵ distorting the signal of the observations. Therefore, using ϵ-manifolds shifts the focus away from distributions over the parameters based on their posterior probability, toward the inversion loss landscape (landscapes consisting of the loss function values for all possible target value configurations). When known, the ϵ-manifold can be used to enable novel types of analysis of model inversion problems, such as ill-posedness quantification, various types of robustness analysis and classification difficulty estimation.

Our main contributions in this chapter are as follows:

- We introduce the concept of ϵ-manifolds for noisy model inversion problems and formalise the problem of capturing the set of potentially valid solutions comprising this ϵ-manifold. To our knowledge, this is the first time the problem has been defined in this manner.

- We propose a novel method, named eMMI (epsilon-manifolds for model inversion), to automatically approximate the ϵ-manifold for a given model inversion problem instance. We provide four variants of our proposed method (U-eMMI, Conv-eMMI, Seq-eMMI, and Dual-eMMI), with different sampling strategies and assumptions on the underlying loss landscape.

- We validate the concept of ϵ-manifolds through an empirical comparison to statistical uncertainty quantification. We also validate our ϵ-manifold approximation method on seven simulation models representing model inversion problem settings from physical models, dynamical systems, simulation-based inference and machine learning. We compare performance against baseline methods from uncertainty estimation methods such as Gaussian processes and Bayesian neural networks, as well as approximate Bayesian computation.

6.2. RELATED WORK

The problem setting for ϵ-manifolds and our proposed method is model inversion with noisy observations. There are three main directions of related work to this setting, which we will discuss in this section. We will start with *robust learning*, a specific type of noisy model inversion-based inference task considering machine learning with noisy labels, that has received considerable attention over the years. Next, we will discuss two methodologically related directions: *simulation-based inference* (SBI) (also referred to as *likelihood-free inference*), and *uncertainty quantification*.

Robust learning. There has long been a research interest in the machine learning community in robust learning: methods that make model training robust to noisy labels (the observations when training a model) [227, 228, 229]. More recently, Northcutt et al. [230] have proposed confident learning, where confidence applies to ground truth labels, rather than model predictions. Uma et al. [231] summarised a body of work on conflicting labels specific to natural language processing and computer vision settings, including majority voting, the source-filter model [232, 233] and the CrowdTruth aggregation approach [234]. Bernhardt et al. [235] proposed to automatically rank training instances based on the label correctness and difficulty as estimated by a prediction model, and found this to improve model performance while reducing the reliance on label-correcting experts. Jiang et al. [236] provided a dataset containing real-world (as opposed to synthetic) noisy labels and proposed the MentorMix method to overcome these noisy labels through curriculum learning and vicinal risk minimisation. Huang et al. [237] investigated the relationship between uncertainty, class imbalance and label noise, and proposed an uncertainty-aware label correction (ULC) framework, which first filters noisy labels based on epistemic uncertainty, after which the remaining corrupted labels are filtered by modelling aleotoric uncertainty as logit corruption with Gaussian noise. Kim et al. [238] proposed to use the relational structure of the data in the embedded feature space to detect noisy labels. Although classification problems tend to receive the most attention, some work has focused on

regression problems as well [239, 240, 241]. In the related predict-then-optimise problem setting [242, 243] (where unobservable parameters are imperfectly predicted using machine learning, which enables the optimisation of a second set of parameters for decision-making), smart predict then optimise (SPO) approaches [226, 244] can be used to train a machine learning model using a loss based on the regret between an optimum found using the predicted unobserved parameters and the true optimum, thereby reducing the impact of noise in the predicted unobserved parameters on the optimisation task.

Unlike the methods above, which are mainly intended to improve the model performance in terms of predictive accuracy when trained using a training data set with noisy labels, the objective of our proposed ϵ-manifolds is to find the set of parameterisations that would all fit the observations (e.g., noisy labels) up to a tolerance level specified by ϵ. These ϵ-manifolds provide deeper insight over performance-focused approaches like robust learning, and have broader applications beyond machine learning model training or robust optimisation, including ill-posedness- and robustness analyses in general model inversion settings.

Simulation-based inference. In many scientific contexts, great effort has been put into formulating models that simulate an observation from a set of input parameters. The goal of simulation-based inference (SBI) is to apply statistical methods to infer the posterior probability of the input parameters $\boldsymbol{\theta}$, which form the inference targets, from a vector of observed outcomes $\mathbf{x}$ [218, 245, 217]. The likelihood of the observations given an input parameter configuration $\boldsymbol{\theta}$ is generally intractable to compute. Following the notation by Cranmer et al. [216], the problem may be formulated (in the Bayesian case) as computing the posterior $p(\boldsymbol{\theta}|\mathbf{x})$, where $\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta},\mathbf{z})$ and $z_i \sim p_i(z_i|\boldsymbol{\theta}, z_{<i})$ ($\mathbf{z}$ represents the latent internal state of the simulator). A point prediction for the inference result can be computed using, e.g., a maximal a posteriori principle $\hat{\boldsymbol{\theta}} \in \text{argmax}_\theta\, p(\boldsymbol{\theta}|\mathbf{x})$. Applications of SBI span a highly diverse set of scientific fields and topics including astrometry [5], Earth sciences [246], gravitational waves [247], astrophysics [6], and genomics [248].

Based on the taxonomy by Cranmer et al. [216], we can broadly split SBI methods into frequentist and Bayesian inference approaches. Frequentist approaches infer the probability of parameters through estimated kernel densities, while Bayesian approaches iteratively approximate the posterior probability of the parameters using observations and prior probabilities. One of the most popular methods for SBI, approximate Bayesian computation (ABC) [249], samples parameter configurations from their prior distributions. It accepts these configurations if the simulated output matches the true observations at a sufficient goodness-of-fit level determined by a threshold ϵ. Similar to our proposed methods, this results in a posterior distribution of possible configurations whose simulated output is ϵ-approximate to the observed data. Unlike our proposed methods, ϵ is a (some-

times dynamic) hyperparameter trading off accuracy for computational efficiency. The posterior distribution depends in part on the prior distribution of the parameters being inferred, making them primarily suitable for conventional inference tasks. ABC methods can often involve Markov chain Monte Carlo (MCMC) [250] and sequential Monte Carlo (SMC) sampling [251]. Some model inversion methods in, e.g., environmental biology, may use numerical optimisation techniques in a manner similar to SBI using ABC [47, 49, 48]. Given the poor scalability of inferring complex posterior probabilities using a Bayesian approach, variational methods and amortised versions thereof can also be used [252, 253, 254].

More recently, advances in machine learning have led to the use of inverse emulation models, often coupled with active learning techniques [50, 51, 52, 45, 53, 54, 255, 55, 46, 54, 45]. These approaches resemble amortised Bayesian inference methods [256, 257, 258], but the machine learning models are trained on the parameters themselves, rather than the posterior distribution parameters identified through Bayesian inference. Other methodological contributions in SBI include reducing the assumptions of models, such as pre-defined priors, targets and dimensionalities [259]. Additionally, scalability has been improved through flow matching for continuous normalising flows [260].

Although SBI research has made many valuable contributions to a wide range of (especially scientific) application areas, it adopts an inherently statistical approach to model inversion, inferring posterior distributions over the parameters instead of capturing characteristics of the loss landscape. Hermans et al. found many SBI-based methods to be overconfident in their inference results [261], indicating that unlikely solutions that could nonetheless explain the observations well would generally not be included. In contrast, our proposed ϵ-manifolds aim to provide insight into the model inversion problem for a given observation based on the model inversion loss landscape, enabling new types of analyses. Unlike methods such as ABC, which return a posterior distribution over parameters based on their goodness-of-fit to the observations and the prior probabilities of the parameters, our proposed method does not aim to make a single prediction (inference) with some degree of uncertainty for a given model inversion instance. Instead, it aims to find the set of configurations that *could* explain the observations, regardless of how likely such configurations are to naturally occur. This can be a desirable trait, for example, when considering interventions or adversarial examples (both of which involve solutions that may not naturally occur in the observational distribution). Therefore, our proposed ϵ-manifolds and our proposed method to approximate them could be considered complementary to SBI, by describing the loss landscape underlying the problem for which SBI is performing inference. Whether SBI or ϵ-manifolds are the most appropriate tool depends on the use case, as both have different types of applications (see also Section 6.4.3).

Uncertainty quantification. The topic of uncertainty quantification (UQ) has received considerable attention in recent years, and UQ is often a part of SBI. Typical approaches for UQ include ensembling [262, 263]), Monte-Carlo dropout [264, 265] and Bayesian neural networks [266, 267]). For a more comprehensive overview, see, for example, the surveys by Adbar et al. [268] or Gawlikowski et al. [269]. Recent examples for scientific machine learning in particular include UQ for physics-informed neural networks [270], physics-constrained surrogate modeling with UQ [271], Monte-Carlo UQ for the Navier-Stokes equations [272], UQ for neural networks trained on physical simulations [273], and the application of a Hamiltonian Monte Carlo algorithm for physical model inversion [195].

The core idea of UQ, similar to our proposed ϵ-manifolds, is that the numerical prediction that seems to explain the observations best may not be the true solution, due to uncertainty (for example, in the form of noisy observations or probabilistic simulations). The statistical perspective of UQ enables a relatively quick computation of uncertainty in terms of, usually, a confidence interval around a (mean) point prediction, but this does not lend itself to the same type of interpretation: a relatively unlikely solution could still explain the observations well, even if it has not been observed in the training data set. Retrieving such solutions could be highly relevant to applications related to, e.g., adversarial robustness or deliberate interventions to achieve a desired outcome. Moreover, the type of relationships that can be expressed by conventional UQ is limited by the parametric form of the assumed distribution; for example, a loss landscape for a chaotic or ill-posed system would be difficult to describe parametrically, especially when the most suitable distribution type cannot be known *a priori*.

In contrast, our proposed ϵ-manifolds describe the loss landscape, identifying a set of solutions that could explain the observations. In doing so, it is possible to use ϵ-manifolds to gain insight into the nature of the inversion problem itself, beyond a statistical perspective of observations and possible outcomes.

6.3. PROBLEM DEFINITION

In this section, we briefly introduce the problems addressed in this work. First, we provide the general problem definition of model inversion, as this is the context within which our work is situated. Next we provide definitions for two problems at the core of our investigation: representing a viable solution set for model inversion (problem 1), and approximating this viable solution set (problem 2).

Context: model inversion. Suppose we are interested in a set of *parameters* P, the values of which are contained in domains $\mathcal{D}_P$. A configuration $\boldsymbol{\theta}$, which we will refer to as a solution in optimisation contexts, is a vector containing the values of the target variables P for a specific point in $\mathcal{D}_P$. The target variables can be ob-

served indirectly via known variables, whose domain is denoted as $\mathcal{X} \subseteq \mathbb{R}^d$. Given a d-dimensional vector of observations $\mathbf{x} \in \mathcal{X}$, we want to infer the (unknown) corresponding true parameter configuration of the system $\boldsymbol{\theta}^+$ via a simulation model M mapping configurations $\boldsymbol{\theta}$ to simulated observations $\mathbf{x}$.

Definition 6.1. A simulation model is a probabilistic function $M : \mathcal{D}_P \rightarrow \mathcal{X}$ that, under a target parameterisation $\boldsymbol{\theta}$, can produce (i.e., simulate) an outcome $\mathbf{x} = M(\boldsymbol{\theta})$, where $M(\boldsymbol{\theta})$ samples $\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})$.

In other words, we want to search for a configuration $\boldsymbol{\theta} \in \mathcal{D}_P$ such that $f_1(M(\boldsymbol{\theta}), \mathbf{x})$ is minimised, thereby approximating the true configuration $\boldsymbol{\theta}^+$ corresponding to the real observations $\mathbf{x}$. This objective can be considered as a more general version of the RTM inversion objective of Section 2.2.1. Here $f_1(.)$ is an objective function quantifying the distance between the simulated outcome $M(\boldsymbol{\theta})$ and the observed outcome $\mathbf{x}$:

$$\hat{\boldsymbol{\theta}} \in \underset{\boldsymbol{\theta} \in \mathcal{D}_P}{\operatorname{argmin}} f_1(M(\boldsymbol{\theta}), \mathbf{x}) \approx \boldsymbol{\theta}^+ \tag{6.1}$$

This optimal configuration $\boldsymbol{\theta}$ is denoted as $\hat{\boldsymbol{\theta}}$ and usually forms the point prediction within the inference results of the model inversion problem. Therefore, unlike the distributions over target values inferred by SBI (whose formulation can be found in Section 6.2), this objective is concerned with the model inversion loss landscape for the inference problem. If the search was successful, $\hat{\boldsymbol{\theta}} \approx \boldsymbol{\theta}^+$.

Problem 1: representing the viable solution set (Section 6.4). We assume an amount of exogenous noise $\mathbf{N}$ on the observations $\mathbf{x}$ or ill-posedness in the inversion loss landscape, thereby potentially invalidating the optimal solution $\hat{\boldsymbol{\theta}}$ from Equation 6.1. Instead, we are interested a subset $\mathcal{V} \subseteq D_P$ of the domain D_P containing all solutions that could possibly be the true solution of the model inversion problem (viable solution set). This set should contain all ill-posed solutions evaluating to the same loss function value as $\hat{\boldsymbol{\theta}}$, but also all solutions that evaluate to a slightly worse loss function value than $\hat{\boldsymbol{\theta}}$, up to a factor of ϵ. The exogenous noise $\mathbf{N}$ and its distribution are often unknown, because noise-free versions of the noisy data are usually impossible to obtain. However, the impact of this noise on the loss function (quantified by ϵ) can be extracted purely from validation data for the target variables $\boldsymbol{\theta}$, without requiring noise-free observations $\mathbf{x}^+ = \mathbf{x} - \mathbf{N}$ to be available. This allows us to define the viable solution set as:

$$\mathcal{V} = \{\boldsymbol{\theta} : f_1\big(M(\boldsymbol{\theta}), \mathbf{x}\big) \leq f_1\big(M(\hat{\boldsymbol{\theta}}), \mathbf{x}\big) + \epsilon\} \tag{6.2}$$

Therefore, for problem 1, we need a framework within which we can represent this set of points $\mathcal{V}$.

Problem 2: approximating the viable solution set (Section 6.5). Computing $\mathcal{V}$ exhaustively is prohibitively expensive for large dimensions of $\boldsymbol{\theta}$, and impossible without interpolation techniques for continuous problem settings. We must therefore find a tractable approximation $\hat{\mathcal{V}}$ of $\mathcal{V}$, by maximising the accuracy on a set of validation points (H_x, H_y):

$$\hat{\mathcal{V}} \in \underset{S}{\operatorname{argmax}} \mathcal{L}(H_y, \hat{H}_y|S) \tag{6.3}$$

Here, H_y denotes the true labels (viable or non-viable solution) corresponding to a sample of points H_x in the target variable space, and $\hat{H}_y|S$ is a vector of predictions of H_y based on a candidate set of viable solutions S. The function $\mathcal{L}$ is a classification loss function of choice (e.g., accuracy), and $\hat{\mathcal{V}}$ is the set of solutions S with the best classification performance on the validation data set (H_x, H_y).

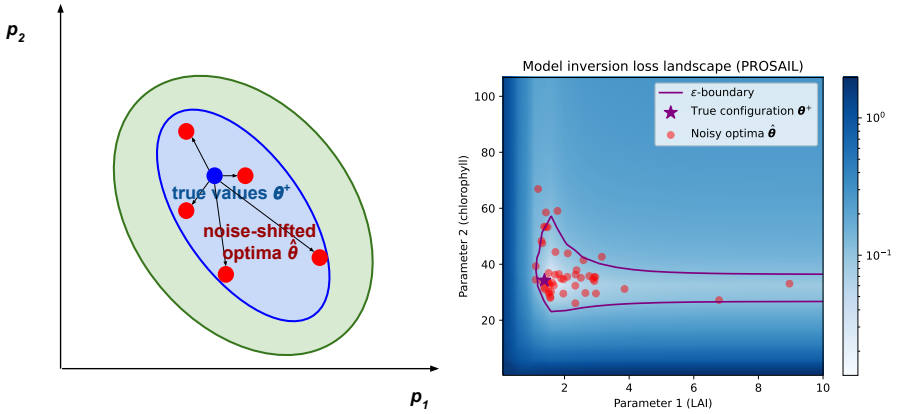
6.4. REPRESENTING THE VIABLE SOLUTION SET

In this section, we will introduce the motivation, concepts and assumptions underlying ϵ-manifolds, which form our representation of the viable solution set for model inversion problems.

6.4.1. INTRODUCING ϵ-MANIFOLDS

Recall the model inversion objective from Equation 6.1. In practice, there are two issues with this naïve search approach. Firstly, multiple configurations $\boldsymbol{\theta} \in \mathcal{D}_P$ can minimise $f_1(M(\boldsymbol{\theta}), \mathbf{x})$ equally (ill-posedness), while only one true configuration $\boldsymbol{\theta}^+$ corresponds to the state of the real-world system. Secondly, if the observations are noisy, the vector of observations $\mathbf{x}$ has been generated from a vector of true observable values, $\mathbf{x}^+$, combined with a vector of exogenous additive noise $\mathbf{N}$: $\mathbf{x} = \mathbf{x}^+ + \mathbf{N}$. Therefore, a configuration $\hat{\boldsymbol{\theta}}$ that precisely minimises $f_1(M(\boldsymbol{\theta}), \mathbf{x})$ is not necessarily the true configuration $\boldsymbol{\theta}^+$. $\hat{\boldsymbol{\theta}}$ would be the appropriate solution for the incorrect observed values $\mathbf{x}$, not the true state of the system (which would have had its own optimal solution $\hat{\boldsymbol{\theta}}^+ \approx \boldsymbol{\theta}^+$ minimising $f_1(M(\boldsymbol{\theta}), \mathbf{x}^+)$. For convenience, similarly to the distinction made by, e.g., Mandi et al. [226], we refer to the optimum $\hat{\boldsymbol{\theta}}$ for the noisy observations $\mathbf{x}$ as the *noisy optimum*, and the optimum $\hat{\boldsymbol{\theta}}^+$ for the original, noise-free observations $\mathbf{x}^+$ as the *noise-free optimum*.

The *optimum shift* from the noise-free optimum $\hat{\boldsymbol{\theta}}^+$ to the noisy optimum $\hat{\boldsymbol{\theta}}$, caused by observation noise, makes the inversion problem effectively ill-posed. There are many potential solutions, each corresponding to the observations with different added noise. Since the noise is unknown, any of these solutions could be the true solution to the model inversion problem. This effect is particularly strong in ill-conditioned problems, where small perturbations on the input (obser-



(a) Optimum shift illustration

(b) Example shift

Figure 6.1: Optimum shifts when noise is introduced to the observations. (a) Illustration of the principle of optimum shifts on an abstract loss landscape for target variables p_1 and p_2. When random noise is applied to the observations, the point in the space where, after simulating with those input parameter settings, the simulation output matches the observations optimally, has shifted from the true parameters (the blue dot, $\boldsymbol{\theta}^+$) to new, shifted optima (the red dots, $\hat{\boldsymbol{\theta}}$). The specific point the optimum shifts to will differ every time the random noise is applied. The blue line represents the ϵ-boundary; beyond this point, there are no possible points the optimum could shift to, at the current level of noise. The blue-shaded area represents the ϵ-manifold, containing the set of all points the optimum could potentially shift to. (b) Example of this phenomenon in practice for one instance from the physical vegetation model PROSAIL, where the application of 15% additive zero-mean Gaussian noise on the observations has caused the optimum $\hat{\boldsymbol{\theta}}$ to shift away from the true values $\boldsymbol{\theta}^+$ (repeated for 50 different samples of random noise added to the noise-free observations). The shifts are more likely for configurations close to the optimum, resulting in a cluster that could possibly be captured by conventional uncertainty quantification methods, but the shifts tend to follow the loss landscape (Assumption 1 in Section 6.4) and stay within the ϵ-boundary.

vations) result in large changes in the output (optimum) [205, 206]. We illustrate this phenomenon with a practical example in Figure 6.1.

As a result of these issues, we need a method describing a set $\mathcal{V}$ of all the potentially viable solutions $\boldsymbol{\theta}$. These solutions could, under some expected noise on the observations, reasonably be the true configuration that generated the observations $\mathbf{x}$. In simple cases matching the underlying assumptions, uncertainty quantification on the optimal configuration $\hat{\boldsymbol{\theta}}$ may be sufficient. For example, if the optimum shifts follow a Gaussian distribution, a confidence interval could be constructed, containing all values within two standard deviations σ of the predicted mean μ. In this case, any point with a probability density higher than some user-specified threshold could be added to the set $\mathcal{V}$ of possible solutions. However, these as-

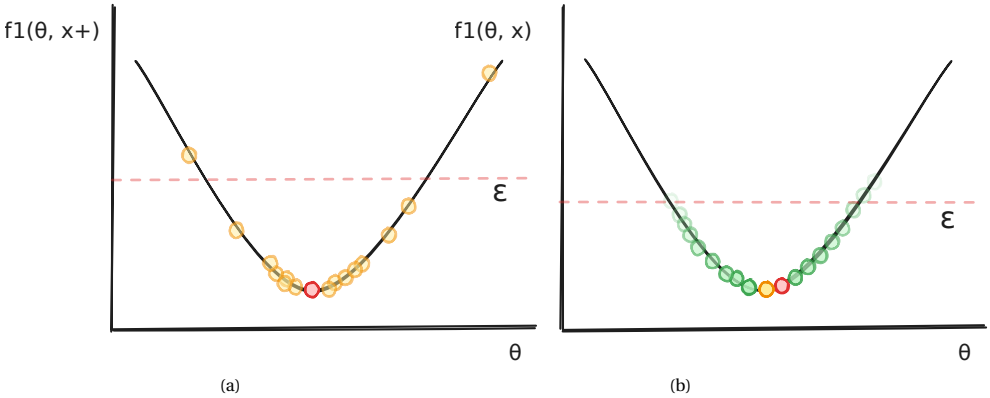


Figure 6.2: Illustration of the two key assumptions made by ϵ-manifolds. (a) In the first assumption, we assume that an optimum for noise-free observations (red dot) will shift to a new location due to observational noise (yellow dots) based on the loss landscape (parabola function) for noise-free observations; a higher loss function value results in a lower likelihood of being shifted to due to noise. (b) In the second assumption, we assume that the probability (green dots, with lower opacity signifying lower probabilities) that an optimum for noisy observations (yellow dot) was originally shifted from a noise-free optimum (red dot) is determined by the loss landscape (parabola function) for the noisy observations.

sumptions are not always met.

The inversion loss landscape for many simulation models, which are often based on complex ordinary differential equations (ODEs) and partial differential equations (PDEs), can be complex, asymmetric, biased and highly non-linear – conditions to which statistical uncertainty quantification would be ill-suited. Instead, by making two key assumptions of *loss-dependent shifts* and *loss-dependent origins* (illustrated in Figure 6.2), this set of points can be represented more accurately using a concept we dubbed ϵ-manifolds.

We formalise the loss-dependent shifts assumption in Assumption 1. According to this assumption (illustrated in Figure 6.2a), the lower the loss function value of a point for a noise-free observation, the higher its likelihood of being the optimum of a noisy version of the observations $\mathbf{x}$ (optimum shift).

Assumption 1. *Let $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ denote two arbitrary points in the target variable space $\mathcal{D}_P$, and let $P(\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}|\mathbf{x})$ denote the probability of a point $\boldsymbol{\theta}$ being the optimum for a noisy observation $\mathbf{x}$ that was generated through exogenous additive noise N being added to the original observations $\mathbf{x}^+$. The **loss-dependent shifts** assumption states that:*

$$f_1\big(M(\boldsymbol{\theta}_1), \mathbf{x}^+\big) < f_1\big(M(\boldsymbol{\theta}_2), \mathbf{x}^+\big) \Rightarrow P(\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_1|\mathbf{x}) > P(\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_2|\mathbf{x}) \tag{6.4}$$

Using this assumption, we could, in principle, set a threshold distance ϵ between the loss of the noise-free optimum $f_1(M(\hat{\boldsymbol{\theta}}^+), \mathbf{x}^+)$ and the loss $f_1(M(\boldsymbol{\theta}), \mathbf{x}^+)$ of a new point $\boldsymbol{\theta}$: if this distance is smaller than the threshold, the point can be considered as a potential location the optimum could shift to. Although there may be applications for this set of points in, e.g., algorithm robustness [274, 275, 276], this set of solutions mainly serves as an intermediate step in our problem setting of model inversion.

Next, we formalise the loss-dependent origins assumption in Assumption 2. This assumption (illustrated in Figure 6.2b) can be considered the inverse of our first assumption. In model inversion, given a noisy observation $\mathbf{x}$, we are interested in a set of solutions in which, at a specified level of confidence, we expect the true solution $\boldsymbol{\theta}^+$ to be included. Under Assumption 2, the lower the loss function value of a point for a noisy observation, the higher its likelihood of having been the original, noise-free optimum $\hat{\boldsymbol{\theta}}^+$, where $\hat{\boldsymbol{\theta}}^+ \approx \boldsymbol{\theta}^+$.

Assumption 2. *Let $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ denote two arbitrary points in the target variable space $\mathcal{D}_P$, and let $P(\hat{\boldsymbol{\theta}}^+ = \boldsymbol{\theta}|\mathbf{x})$ denote the probability of a point $\boldsymbol{\theta}$ being the noise-free optimum $\hat{\boldsymbol{\theta}}^+$. The **loss-dependent origins** assumption states that:*

$$f_1\big(M(\boldsymbol{\theta}_1), \mathbf{x}\big) < f_1\big(M(\boldsymbol{\theta}_2), \mathbf{x}\big) \Rightarrow P(\hat{\boldsymbol{\theta}}^+ = \boldsymbol{\theta}_1|\mathbf{x}) > P(\hat{\boldsymbol{\theta}}^+ = \boldsymbol{\theta}_2|\mathbf{x}) \tag{6.5}$$

As before, we set a threshold ϵ for this distance in loss function value; with this, we can now define the concept of an ϵ-manifold.

Definition 6.2. An ϵ-manifold (eM) is a local connected set of points $\boldsymbol{\theta}$ forming a manifold, where the difference between their loss function values $f_1(M(\boldsymbol{\theta}), \mathbf{x})$ and the loss function value of the noisy optimum $f_1(M(\hat{\boldsymbol{\theta}}), \mathbf{x})$ is equal to or smaller than the threshold parameter ϵ:

$$\begin{aligned} eM = \{\boldsymbol{\theta} : f_1\big(M(\boldsymbol{\theta}), \mathbf{x}\big) \leq f_1\big(M(\hat{\boldsymbol{\theta}}), \mathbf{x}\big) + \epsilon\} \\ \text{subject to} \\ \nexists U, V : U \neq \emptyset, V \neq \emptyset, U \cap V = \emptyset, eM = U \cup V \text{(connectedness)} \end{aligned} \tag{6.6}$$

Here eM is used to denote the ϵ-manifold for a noisy observation $\mathbf{x}$, and the constraint requires the ϵ-manifold to be connected (i.e., eM is not split into two non-empty open subsets U and V such that the union of these sets forms the ϵ-manifold). In traditional SBI algorithms, the solution $\hat{\boldsymbol{\theta}}$ will generally be a point in this ϵ-manifold [216, 249].

The ϵ variable in Equation 6.6 matches the ϵ of Equation 6.2, and its value can be interpreted as the maximum expected change from the objective function value $f_1(M(\hat{\boldsymbol{\theta}}^+), \mathbf{x}^+)$ of the noise-free optimum $\hat{\boldsymbol{\theta}}^+$ to the objective function value of the noisy optimum $f_1(M(\hat{\boldsymbol{\theta}}), \mathbf{x})$. Equivalently, it can be thought of as the maximum

amount of signal ('clean' loss values) that can be distorted or overpowered by noise on the observations. We refer to the boundary between the ϵ-manifold and the rest of the target variable space as the ϵ-boundary, and we use the term ϵ-loss to refer to ϵ added to the loss function value of the optimum $f_1(M(\hat{\boldsymbol{\theta}}),\mathbf{x})$:

$$l^{\epsilon} = f_1\big(M(\hat{\boldsymbol{\theta}}),\mathbf{x}\big) + \epsilon \tag{6.7}$$

6.4.2. PROPERTIES OF ϵ-MANIFOLDS

We will use this section to analyse the theoretical properties of ϵ-manifolds. First, if there is any solution $\boldsymbol{\theta}$ that could be viable solution, this solution is part of an ϵ-manifold:

Lemma 6.1. Let $\boldsymbol{\theta}$ denote an arbitrary point in $\mathcal{D}_P$. Then through Equation 6.6, $f_1\big(M(\boldsymbol{\theta}),\mathbf{x}\big) \leq l^{\epsilon} \Rightarrow \exists eM : \boldsymbol{\theta} \in eM$.

In unimodal landscapes, the total set of potential solutions $\mathcal{V}$(Equation 6.2) exactly matches the ϵ-manifold with the same value of ϵ ($\mathcal{V} = eM$), and the ϵ-manifold for any point $\boldsymbol{\theta}$ where $f_1(M(\boldsymbol{\theta}),\mathbf{x}) \leq l^{\epsilon}$ (Lemma 6.1) will be the same ϵ-manifold eM. However, in multimodal cases (such as the ill-posed example from Section 6.1: $y = \alpha^2$, $\alpha \in \mathbb{R}$), the loss landscape contains multiple local- or approximate global optima, some with their own ϵ-manifold (it is also possible for lower-quality local optima to already be contained in the ϵ-manifold for a higher-quality optimum). In these cases, the viable solution set $\mathcal{V}$ becomes the ϵ-manifold set[2]:

Definition 6.3. An ϵ-manifold set (eMS) is a set of size m containing ϵ-manifolds for a single loss landscape, where every element is a disjoint ϵ-manifold for the ϵ-loss l^{ϵ}, and m is the number of local optima with disjoint ϵ-manifolds:

$$eMS = \{eM_1, eM_2, ..., eM_m\} \tag{6.8}$$

We can use Lemma 6.2 to iteratively construct an ϵ-manifold set:

Lemma 6.2. Let eMS denote a current, potentially incomplete ϵ-manifold set consisting of ϵ-manifolds $eM \in eMS$, and let $\boldsymbol{\theta}$ denote a point in $\mathcal{D}_P$ where $\forall eM \in eMS : \boldsymbol{\theta} \notin eM$. Then $f_1\big(M(\boldsymbol{\theta}),\mathbf{x}\big) \leq l^{\epsilon} \Rightarrow \exists eM' : \boldsymbol{\theta} \in eM'$ (Lemma 6.1) that is disjoint from the existing ϵ-manifolds in the ϵ-manifold set (Definition 6.3) and should be added to the ϵ-manifold set.

[2] Although it is convenient to think of ϵ-manifold sets as sets containing individual manifolds, strictly speaking, the ϵ-manifold set is itself a manifold of which $eM_1, eM_2, ..., eM_m$ are components.

Based on Lemma 6.2, if there exists any point outside the existing ϵ-manifold eM with a loss function value lower than the ϵ-loss, there must exist another ϵ-manifold eM' that, by Definition 6.3, is disjoint from eM. This leads to Theorem 6.1, which allows us to check whether an ϵ-manifold set is complete:

Theorem 6.1. Let $\hat{\boldsymbol{\theta}}^{-eMS}$ denote the optimum $\operatorname{argmin}_{\boldsymbol{\theta}\in[\mathcal{D}_P\setminus eMS]}\left[f_1(M(\boldsymbol{\theta}),\mathbf{x})\right]$ where the current ϵ-manifold set eMS is excluded from the search space $\mathcal{D}_P$. If $f_1\left(M(\hat{\boldsymbol{\theta}}),\mathbf{x}\right)> l^\epsilon$, there exists no further ϵ-manifold that should be added to the ϵ-manifold set eMS, and all possible ϵ-manifolds are contained in the ϵ-manifold set.

Proof.

$$\hat{\boldsymbol{\theta}} \in \underset{\boldsymbol{\theta}\in[\mathcal{D}_P\setminus eMS]}{\operatorname{argmin}} \left[f_1\left(M(\boldsymbol{\theta}),\mathbf{x}\right)\right] \Rightarrow \forall \boldsymbol{\theta}' \in [\mathcal{D}_P \setminus eMS] : f_1\left(M(\boldsymbol{\theta}'),\mathbf{x}\right) \geq f_1\left(M(\hat{\boldsymbol{\theta}}),\mathbf{x}\right)$$

$$f_1\left(M(\hat{\boldsymbol{\theta}}),\mathbf{x}\right)> l^\epsilon \Rightarrow \forall \boldsymbol{\theta}' \in [\mathcal{D}_P \setminus eMS] : f_1\left(M(\boldsymbol{\theta}'),\mathbf{x}\right)> l^\epsilon$$

$$f_1\left(M(\hat{\boldsymbol{\theta}}),\mathbf{x}\right)> l^\epsilon \Rightarrow \nexists \boldsymbol{\theta}' : f_1\left(M(\boldsymbol{\theta}'),\mathbf{x}\right)< l^\epsilon$$

Therefore, by Lemma 6.2, the ϵ-manifold is complete, and no further ϵ-manifolds should be added to it.

□

Whether the ϵ-manifold set for a multimodal landscape contains a single, larger ϵ-manifold, or multiple disjoint smaller ϵ-manifolds, will depend on the loss landscape and the setting of ϵ.

The ϵ-manifolds in the ϵ-manifold set together contain all points in the viable solution set $\mathcal{V}$:

Theorem 6.2. Let eMS denote the complete ϵ-manifold set of a model inversion problem, and $\mathcal{V}$ the set of viable solutions to the model inversion problem (Equation 6.2). Then:

$$\mathcal{V} = \bigcup_{eM\in eMS} eM \tag{6.9}$$

Proof. Let $\boldsymbol{\theta} \in \mathcal{V}$ denote an arbitrary viable solution in $\mathcal{V}$. By Equation 6.2, $f_1\left(M(\boldsymbol{\theta}),\mathbf{x}\right) \leq l^\epsilon$. Therefore, by Lemma 6.1, $\exists eM : \boldsymbol{\theta} \in eM$, and by Equation 6.8 and Theorem 6.1, this $eM \in eMS$. Therefore, $\forall \boldsymbol{\theta} \in \mathcal{V} : \boldsymbol{\theta} \in \bigcup_{eM\in eMS} eM$. Conversely, let $\boldsymbol{\theta} \in eM \in eMS$ denote an arbitrary viable solution in an arbitrary ϵ-manifold in the ϵ-manifold set. By Equation 6.6, $\forall \boldsymbol{\theta} \in eM \in eMS : f_1\left(M(\boldsymbol{\theta}),\mathbf{x}\right) \leq l^\epsilon$, and by Equation 6.2, $\boldsymbol{\theta} \in \mathcal{V}$, so $\forall \boldsymbol{\theta} \in \bigcup_{eM\in eMS} eM : \boldsymbol{\theta} \in \mathcal{V}$. Therefore, $\mathcal{V} = \bigcup_{eM\in eMS} eM$. □

In the illustration of Figure 6.1a we used to show the concept of optimum shifts, the blue-shaded region within which the optima are shifted forms a visual representation of the ϵ-manifold. The appropriate setting of ϵ, similarly to the role of the significance level α in confidence intervals, will depend on the specific problem setting, but can be approximated empirically (see Section 6.6.1).

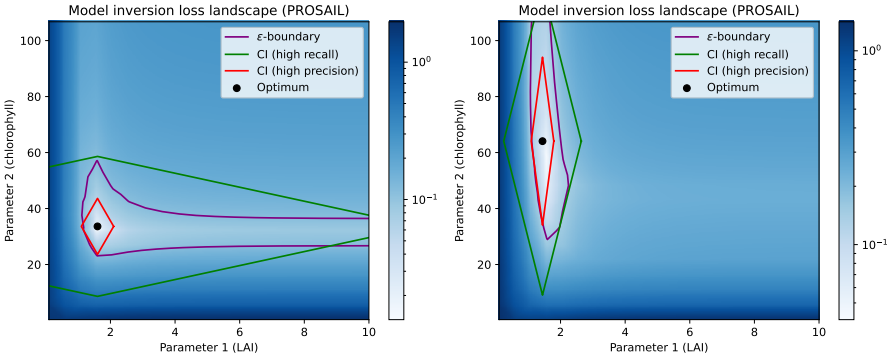
6.4.3. CONTRASTING ϵ-MANIFOLDS AND CONFIDENCE INTERVALS

Our proposed ϵ-manifolds bear some resemblance to the use of confidence intervals in statistical settings, as both concepts represent a type of uncertainty on predictions made for an observation. The key differences lie in their interpretation and application. An ϵ-manifold contains all the solutions that *could* explain the observations, whereas confidence intervals are concerned with the probability of different solutions that they *did* result in the observations.

A confidence interval provides bounds within which the true value is contained at a probability of $1-\alpha$, where α represents the significance level. This statistical quantity can be computed relatively easily. It treats the underlying processes as a black-box generating noisy outcomes, focusing on the spread of possible target values for a given observation, based on the posterior probability of the target variables given the observations.

If there are configurations that are unlikely to appear in data (far removed from the point prediction, with a low prior probability), but could explain the observations equally well (similar loss function value), confidence intervals would be unlikely to include them. This could be, because such cases were not observed in the data, or because including them (improving recall) would come at the expense of an increase in false positives (reducing precision), as illustrated in Figure 6.3. For most distribution types, the probability of target variable values monotonically decreases as distance to the point prediction (the value with the highest likelihood) increases. The range of the confidence intervals would greatly depend on the shape of the assumed distribution (usually Gaussian).

Uncertainty quantification in the form of confidence intervals and other statistical metrics can be a highly effective practical tool for indicating confidence or uncertainty for concrete prediction tasks, and can answer questions such as 'how reliable is my prediction for this specific instance?'. In contrast, an ϵ-manifold provides insight into the relationship between inputs and outputs irrespective of the probability of the inputs, functioning as a tool to analyse and gain an understanding of an inversion problem. It allows us to answer questions related to uncertainty, such as 'what is the set of possible configurations that could have satisfied these observations?', but also 'how ill-posed is this model inversion instance', 'is there a configuration that would satisfy all these observation instances simultane-



(a) Example landscape, instance 1

(b) Example landscape, instance 2

Figure 6.3: Comparing the ϵ-boundary (purple) and confidence intervals (means $\mu\pm$ two times the standard deviation σ, in green for a high recall and red for a high precision) through visualising the loss landscape for the inversion of two different instances with two parameters using a specific physical model for vegetation (PROSAIL). Deeper shades of blue on the loss landscape represent a higher loss function value (plotted at a log scale). (a) An instance where parameter 1 (LAI) is skewed. (b) An instance where parameter 2 (chlorophyll) is skewed. Relying on rigid assumptions on the shape of the distribution, the confidence intervals based on the Gaussian distribution cannot realistically capture the viable solution set for either of the instances, while the ϵ manifolds can flexibly do so in both scenarios.

ously?', 'are there configurations that do not occur naturally that would achieve a desired outcome?', or 'how accurate does my model parameterisation need to be to still achieve similar performance?' Rather than indicating confidence through a summary statistic, ϵ-manifolds enable deeper analyses of problems and problem instances. Unlike confidence intervals, where the focus is on the target variable space and distances within this space, ϵ-manifolds focus on the loss function values of solutions, regardless of how these solutions relate to one another in the target variable space.

Consider the model inversion problems in Figure 6.3. The optimum for Figure 6.3a (which would be the point prediction of a statistical model) for the leaf area index (LAI) variable is around 1.8, but its ϵ-manifold ($\epsilon = 0.1$) covers LAI values between 1.5 and its maximal value, 10. This is a known phenomenon in the domain [277], where the signal from LAI gets 'saturated' after a certain point, after which the observed light spectrum is no longer affected by further increases. As a result, the ϵ-manifold is asymmetric, which a standard Gaussian distribution could not represent, as reducing under- or overinclusivity in one direction would increase it in the other. We show this in Figure 6.3, where 'high recall' refers to a confidence

interval designed to include as many of the points in the ϵ-manifold as possible, and 'high precision' refers to a confidence interval designed to exclude as many of the points outside the ϵ-manifold as possible. Neither objective can be achieved without sacrificing the other, illustrating that confidence intervals often cannot represent the same set of points as ϵ-manifolds. Additionally, phenomena such as the saturation of LAI values would be difficult to identify if only values close to the mean (with high prior probabilities) were included.

Finally, even if all possible viable solutions were represented in the data, and a suitable long-tailed LAI distribution type were found for the problem in Figure 6.3a, the next instance in Figure 6.3b would require a completely different type of distribution to characterise the shape of the loss landscape for the viable solutions. Given the high variability of distribution properties between instances, which cannot be known *a priori*, it is unlikely that similar analyses to those enabled by ϵ-manifolds could be achieved through existing frameworks, such as confidence intervals.

For this reason, we consider the use of ϵ-manifolds to be the appropriate choice when aiming to gain insight into the loss landscape of the inversion problem itself, rather than treating the inversion as a noisy black box that causes uncertainty on the inference results. However, their use may come at the expense of a higher computational cost.

6.5. APPROXIMATING THE VIABLE SOLUTION SET

In this section, we will describe our proposed method, called ϵMMI or, more conveniently, eMMI (ϵ/epsilon-Manifolds for Model Inversion), for approximating the ϵ-manifold in practice[3]. If fully committed to existing statistical frameworks, in some cases it may be possible to approximate a set of points similar to the ϵ-manifold as the non-parametric posterior distribution approximated by an ABC algorithm with a uniform prior over the entire search space, and an acceptance condition based on the ϵ-loss. However, such an approach would quickly become computationally expensive, as the number of simulations required to accurately approximate the posterior for the large search space would quickly become intractable in higher dimensions. If amortised approaches were used to improve computational efficiency, the reliance on summary statistics would reduce the applicability of such approaches to ϵ-manifold approximation (see Section 6.4.3).

In contrast, our proposed method, eMMI, aims to efficiently sample points based on the loss function landscape. In most problem settings, the loss landscape will be intractable to compute fully, particularly in high dimensions where

[3]All code for our proposed method and experiments is publicly available at https://github.com/ADA-research/eMMI

the required number of samples increases exponentially. However, the curse of dimensionality can become a blessing in this problem setting. For example, if 50% of a target variable range is viable per dimension, and dimensionality $d = 10$, only $(0.5)^{10} \times 100\% \approx 0.1\%$ of the space would be viable, enabling the use of efficient local sampling strategies. By Theorem 6.2, an effective approximation of the *local* ϵ-manifold or ϵ-manifold set, through such an efficient sampling approach, approximates the viable solution set $\mathcal{V}$ as $\hat{\mathcal{V}}$ (Equation 6.3). Therefore, eMMI aims to exploit the sparsity of the search space through heuristics and assumptions about the loss landscape. As this will be the first time a method is proposed to explicitly approximate the ϵ-manifold, we strove to keep the design of eMMI modular by splitting its execution into different steps.

6.5.1. eMMI HIGH-LEVEL OVERVIEW

We propose a general three-step approach for approximating the ϵ-manifold in the loss function landscape for a model inversion problem instance. We consider that eMMI is given a finite budget of function evaluations (simulations with loss function value computation) B, which can be split freely between the different steps and whose division is a tunable hyperparameter of the method. The general steps of the method are described below:

1. Searching over the target variable space $\mathcal{D}_P$ for a configuration $\hat{\boldsymbol{\theta}}$ such that $f_1\big(M(\hat{\boldsymbol{\theta}}), \mathbf{x}\big)$ is minimised.

2. Given $\hat{\boldsymbol{\theta}}$ (found in step 1) and ϵ, conducting a search over $\mathcal{D}_P$ to find a diverse set of configurations $\boldsymbol{\Theta}$ around the ϵ-boundary.

3. Approximating the ϵ-manifold using the points sampled in step 2.

We have further provided an overview of eMMI in Algorithm 6.1. We will be providing additional details to the operations contained in Algorithm 6.1 over the rest of this section. In Algorithm 6.1, lines 6 through 9 correspond to step 1 as described above (showing random search as an example for simplicity), lines 11 through 28 correspond to step 2, and lines 29 through 33 correspond to step 3.

Algorithm 6.1 eMMI algorithm overview

Input: observation $\mathbf{x}$; simulator M; maximal shift ϵ; eMMI variant var; maximal ϵ-manifold set size m; optimisation budgets B_1, B_2; #iterations/population size n_{iter}
Output: ϵ-manifold set eMS
$eMS \leftarrow \emptyset$, $l^{\epsilon} \leftarrow \infty$
for $n = 1, ..., m$ **do**
Begin step 1
  $\hat{\boldsymbol{\theta}} \leftarrow$ a random solution sampled from $D_P \setminus eMS$ ▷ initialise; can be warm-started
  for $i = 1, \ldots, B_1$ **do**
    $\boldsymbol{\theta} \leftarrow$ a random solution sampled from $D_P \setminus eMS$ ▷ sample a new solution
    if $f_1(M(\boldsymbol{\theta}), \mathbf{x}) \leq f_1(M(\hat{\boldsymbol{\theta}}), \mathbf{x})$ **then** ▷ f_1 from Eq. 6.10
      $\hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}$ ▷ update optimum
  if $f_1(M(\hat{\boldsymbol{\theta}}), \mathbf{x}) > l^{\epsilon}$ **then break loop** ▷ break loop to return eMS
Begin step 2
  $l^{\epsilon} \leftarrow f_1(M(\hat{\boldsymbol{\theta}}), \mathbf{x}) + \epsilon$ ▷ store the ϵ-loss (Eq. 6.7)
  if var = Seq-eMMI **then**
    $\boldsymbol{\theta}' \leftarrow \hat{\boldsymbol{\theta}}$ ▷ search for a point $\boldsymbol{\theta}'$ on ϵ-boundary
    for $i = 1, \ldots, \lfloor B_2 / n_{iter} \rfloor$ **do**
      $\boldsymbol{\theta} \leftarrow$ a random solution sampled from $D_P \setminus eMS$
      if $|f_1(M(\boldsymbol{\theta}), \mathbf{x}) - l^{\epsilon}| \leq |f_1(M(\boldsymbol{\theta}'), \mathbf{x}) - l^{\epsilon}|$ **then**
        $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}$ ▷ update solution if new point is closer to ϵ-loss
  $\boldsymbol{\Theta} \leftarrow initialise\_population(var, \hat{\boldsymbol{\theta}})$ ▷ initialise population
  $H \leftarrow []$ ▷ list to store function evaluations
  for $i = 1, \ldots, \lfloor B_2 / n_{iter} \rfloor$ **do** ▷ assuming synchronous updates (Seq/Dual-eMMI)
    for $j = 1, \ldots, n_{iter}$ **do** ▷ flip loops for asynchronous updates
      $\boldsymbol{\theta} \leftarrow$ a random solution sampled from $D_P \setminus eMS$
      if $f_2(M(\boldsymbol{\theta}), \mathbf{x}) \leq f_2(M(\boldsymbol{\Theta}^j), \mathbf{x})$ **then** ▷ f_2 from Eq. 6.13, 6.14, Eq. 6.15, 6.16
        $\boldsymbol{\Theta}^j \leftarrow \boldsymbol{\theta}$ ▷ update j-th solution in population
      $y \leftarrow 0$ ▷ original label is false
      if $f_1(M(\boldsymbol{\theta}), \mathbf{x}) \leq l^{\epsilon}$ **then**
        $y \leftarrow 1$ ▷ change label to true if in ϵ-manifold
      append $(\boldsymbol{\theta}, y)$ **to** H
Begin step 3
  if var = Conv-eMMI **then**
    $eM \leftarrow convex\_hull(\boldsymbol{\Theta})$ ▷ ϵ-manifold becomes convex hull of solutions
  if $var \in$ { U-eMMI, Seq-eMMI, Dual-eMMI} **then**
    $eM \leftarrow train\_classifier(H)$ ▷ ϵ-manifold becomes trained classifier
  append eM **to** eMS ▷ Add eM to ϵ-manifold set
return eMS

6.5.2. STEP 1: FINDING AN OPTIMUM

In step 1, the system performs a search with an evaluation budget B_1 to find the solution $\hat{\boldsymbol{\theta}}$ minimising the main objective function f_1. This search can be performed by any black-box optimisation algorithm; our current implementation supports random search, greedy local search, CMA-ES [278, 279] and gradient descent (using finite-difference gradient approximation). Although the objective may vary per domain, we used the proportional absolute difference between the observation $\mathbf{x}$ and the simulated output $M(\boldsymbol{\theta})$, to avoid observed variables with different ranges from dominating the loss function:

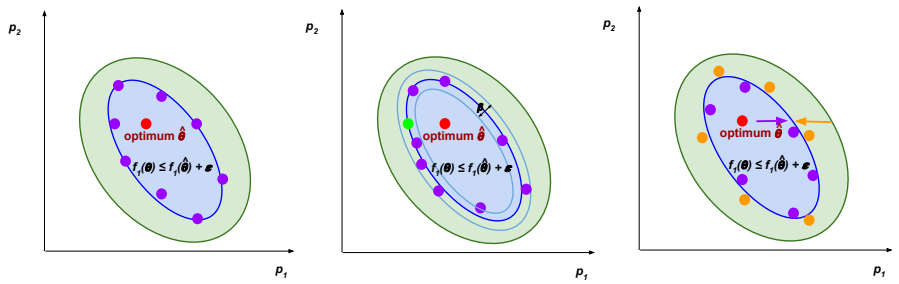
$$f_1\big(M(\boldsymbol{\theta}),\mathbf{x}\big)=\frac{1}{d}\cdot\sum_{j=0}^{d}\frac{|x_j-M(\theta)_j|}{|x_j|} \tag{6.10}$$

Here $M(\boldsymbol{\theta})_j$ is the jth element of the model output (simulated observations) for a sampled configuration $\boldsymbol{\theta}$. The optimum $\hat{\boldsymbol{\theta}}$ can now be searched for as the solution $\boldsymbol{\theta}$ minimising f_1:

$$\hat{\boldsymbol{\theta}}\in\underset{\boldsymbol{\theta}\in D_P}{\operatorname{argmin}}\, f_1\big(M(\boldsymbol{\theta}),\mathbf{x}\big) \tag{6.11}$$

Finding the optimum of a function through black-box optimisation can become a challenging and computationally intensive problem in higher dimensions. To alleviate this problem, we take advantage of the special characteristics of model inversion. Since forward simulation models are available, it is possible to follow a 'hybrid modeling' approach (see, e.g., Verrelst et al. [35], Binh et al. [45] and Ranghetti et al. [55]) to warm-start the optimisation algorithm. In hybrid modeling, a machine learning model is trained on a look-up table (LUT) of simulated data to predict the original inputs from the simulated outputs. Since these models will have their own inaccuracies, we opted to use their output to warm-start the optimisation for step 1 in a part of the search space that is likely closer to the optimum than a random- or mean initialisation would be.

Step 1 will converge to a single (generally global, depending on the landscape and the choice of optimisation algorithm) optimum in the loss landscape of f_1. If the loss landscape is known to be multimodal and globally non-convex, there may be solutions of similar quality to the identified $\hat{\boldsymbol{\theta}}$ in other parts of the target variable space, with their own ϵ-manifold. In this case, it is possible to repeat the three steps for additional optima to approximate the ϵ-manifold set. After obtaining the ϵ-manifold and adding these points to the ϵ-manifold set eMS, we exclude those points in subsequent searches in lines 5, 7, 15 and 22 of Algorithm 6.1. This prevents the algorithm from entering parts of the search space that are already



(a) Conv-eMMI (b) Seq-eMMI (c) Dual-eMMI

Figure 6.4: Illustrations of the sampling strategies of the three optimisation-based variants of eMMI (U-eMMI would simply uniformly sample the space) in an abstracted 2-dimensional loss landscape. The green shaded area represents the parameter space D_P, the red dot represents the optimum $\hat{\boldsymbol{\theta}}$, and the blue shaded area represents the ϵ-manifold with its blue border representing the ϵ-boundary. In Conv-eMMI (Figure 6.4a), the points (purple dots) are optimising for diversity, constrained to not exceed the ϵ-boundary. In Seq-eMMI (Figure 6.4b), the method first finds any point on the ϵ-boundary (the green dot), after which it uses diversity optimisation (purple dots), constrained to not deviate from the ϵ-boundary further than a threshold controlled by a hyperparameter β. In Dual-eMMI (Figure 6.4c), half of the population (purple dots) is maximising its diversity as well as the distance from $\hat{\boldsymbol{\theta}}$, constrained to not move outside the ϵ-boundary, while the other half of the population (orange dots) is maximising diversity and minimising its distance to $\hat{\boldsymbol{\theta}}$, constrained to not move inside the ϵ-boundary.

part of an ϵ-manifold. By Definition 6.3, any point $\boldsymbol{\theta}$ already in an ϵ-manifold $eM_j \in eMS$ could not be contained in any other ϵ-manifold eM_j, enabling the deletion of such points from the search space. By Theorom 6.1, if the objective function value $l^\epsilon = f_1(M(\hat{\boldsymbol{\theta}}_t), \mathbf{x})$ for the optimum $\hat{\boldsymbol{\theta}}_t$ identified at the tth iteration is greater than the ϵ-loss $f_1(M(\hat{\boldsymbol{\theta}}_0), \mathbf{x}) + \epsilon$ for the first optimum $\hat{\boldsymbol{\theta}}_0$, we consider all optima and their ϵ-manifolds that should be within the ϵ-manifold set to have been found.

We note that there may be more efficient solutions for multimodal globally non-convex landscapes possible, especially when there are many viable local optima, by using optimisation algorithms directly converging to multiple local optima in step 1, instead of iterating the entire algorithm. However, in this article, we focus primarily on unimodal and multimodal globally convex landscapes; further extensions to improve the efficiency of eMMI for multimodal globally non-convex landscapes are beyond the scope of this work.

6.5.3. STEP 2: FINDING A DIVERSE SET OF SOLUTIONS AROUND THE ϵ-BOUNDARY

For step 2, we perform diversity optimisation to efficiently obtain a set of samples, usually along the ϵ-boundary. We sample along the ϵ-boundary, because this will allow us in step 3 (see Section 6.5.4) to either directly approximate the ϵ-manifold from the final solution set, or take advantage of the efficient sampling strategy to greatly reduce the number of samples required for training a classification approach.

There are four different variants of eMMI, which differ mainly in their sampling strategy in step 2 (uniform without heuristics, constrained diversity using Equation 6.13, along the ϵ-boundary using Equation 6.14, and mutually opposite objectives and constraints using Equations 6.15 and 6.16). The selection of the appropriate eMMI-variant for a given problem can be automated using hyperparameter optimisation (HPO). The first and simplest variant of our method, U-eMMI, does not leverage any additional optimisation or heuristics, and instead uniformly samples the target variable space (sample size B_2). This variant may have advantages over the other variants in low-dimensional problem settings with many disjoint ϵ-manifolds, as it does not attempt to exploit the locality of viable solutions, but scales poorly to high-dimensional problems, where it would be strongly affected by the curse of dimensionality (requiring an exponentially growing number of samples).

The remaining variants of eMMI are founded on diversity optimisation techniques. The appeal of diversity optimisation is that solutions push each other to the edges of the constrained manifold, where distances are larger, while also maintaining maximum distance to each other to span the entire manifold. We illustrate the ideas of these three variants via the examples in Figure 6.4. If we denote $\boldsymbol{\Theta}$ as a population of points $\boldsymbol{\theta}$ obtained in different ways by the different eMMI variants, every individual $\boldsymbol{\theta}$ in the population $\boldsymbol{\Theta}$ can be thought of as a single point moving through the search space. Based on this population, a new objective function, f_2 (line 23 in Algorithm 6.1), can then be used to optimise for diversity within $\boldsymbol{\Theta}$.

The f_2 function, therefore, requires a metric to quantify diversity within a population. It can be desirable to keep control over how many neighbours to consider when computing the diversity metric value. Therefore, when computing the diversity of a new candidate solution $\boldsymbol{\theta}$, we sort $\boldsymbol{\Theta}$ based on the distance of its elements to $\boldsymbol{\theta}$. If j indexes a target variable as the jth element of the vector $\boldsymbol{\theta}$, the generic form of our diversity term $div(\boldsymbol{\Theta}, \mathbf{x}, \boldsymbol{\theta})$ can be written as:

$$div(\boldsymbol{\Theta}, \mathbf{x}, \boldsymbol{\theta}) = \frac{1}{k} \cdot \sum_{s=0}^{k} \frac{1}{|P|} \cdot \sum_{j=1}^{|P|} |\boldsymbol{\Theta}_j^s - \boldsymbol{\theta}_j| \tag{6.12}$$

Here, we select the k-nearest neighbours to $\boldsymbol{\theta}$ in $\boldsymbol{\Theta}$, indexed by s. For example, $\boldsymbol{\Theta}_3^2$ would refer to the third variable of the second-closest configuration vector $\boldsymbol{\theta}$ in $\boldsymbol{\Theta}$, after being sorted. This results in a diversity scalar term representing the average distance between the values of a new solution $\boldsymbol{\theta}$ and its k-nearest neighbours in $|P|$-dimensional space.

In the following, we will describe the different heuristics enabled by Equation 6.12 to efficiently explore the search space for step 2 of the eMMI algorithm.

CONV-EMMI

In the first variant of our method, we use diversity optimization to push the solutions in the population toward the ϵ-boundary, maximizing diversity to split the population as equally as possible along the ϵ-boundary, forming an outline of the boundary through its final solution set.

In the visual example of Figure 6.4a, the solutions $\boldsymbol{\theta}$ of the population Θ are visualised as purple dots, spread around the ϵ-boundary (though some later iterations may start placing points to the centre of the ϵ-manifold, once the diversity pressure from points on the ϵ-boundary becomes stronger). To achieve this, we perform an iterative constrained diversity optimisation procedure for n_{iter} iterations. The total budget for this step, B_2, is split evenly between the n_{iter} iterations (resulting in an individual budget of $\frac{B_2}{n_{iter}}$). In each iteration of this variant, the algorithm searches for a new solution that maximises a new objective function, f_2, which quantifies the diversity of a candidate solution given the current set of solutions $\boldsymbol{\Theta}$ obtained from previous iterations. Motivated by how diversity is measured in quality diversity (QD) evolutionary algorithms [280], we can define the new objective f_2 (used in line 23 of Algorithm 6.1) through the diversity term from Equation 6.12, and add a constraint to achieve the desired behaviour. In Conv-eMMI, the search is constrained to not exceed the ϵ-boundary, only allowing solutions $\boldsymbol{\theta}$ within a distance of ϵ from the f_1 value of the optimum $\hat{\boldsymbol{\theta}}$. This can be formalised as:

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{maximise}} \quad & f_2(\boldsymbol{\Theta}, \mathbf{x}, \boldsymbol{\theta}, \epsilon) = div(\boldsymbol{\Theta}, \mathbf{x}, \boldsymbol{\theta}) \\ \text{subject to} \quad & f_1(M(\boldsymbol{\theta}), \mathbf{x}) \leq f_1(M(\hat{\boldsymbol{\theta}}), \mathbf{x}) + \epsilon \end{aligned} \tag{6.13}$$

The population $\boldsymbol{\Theta}$ is initialised to the optimum found in step 1, encouraging initial solutions to move to the points in the ϵ-manifold at the furthest distance from the optimum, and each individual in the population updates sequentially. By optimising for diversity while constraining the points not to exceed the ϵ-boundary, the final set of points $\boldsymbol{\Theta}$ will create an outline of the shape of the ϵ-manifold (line 30 in Algorithm 6.1).

Conv-eMMI is conceptually intuitive, can easily integrate with arbitrary optimisation frameworks because every iteration essentially searches for a single new optimum in a new (f_2) loss landscape, and does not rely on an additional approximation step that may introduce inaccuracies to the algorithm. On the other hand, this variant will 'waste' some computation on filling the space between the optimum and the ϵ-boundary, making it less efficient for large ϵ-manifolds. Moreover, when representing the ϵ-manifold in step 3, Conv-eMMI can also only use the convex hull-based approach described in Section 6.5.4, because its optimisation only samples points within the manifold. This can limit its applicability in non-convex use cases, while the computation of the convex hull can also be impossible for some solution sets, or become intractable in higher dimensions.

SEQ-EMMI

The next two variants of our proposed method approximate the ϵ-manifold using classifiers, trained such that their decision boundary corresponds to the ϵ-boundary, as opposed to the convex hull of the solution set. Although the details of this procedure will be discussed for step 3 in Section 6.5.4, the efficient training of such a classifier requires a change in the sampling approach for step 2, with a new focus on sampling points that contribute most toward training such a classifier. The history of function evaluations for the points sampled during the optimisation procedure can later form a training set for a classifier.

The intuition behind the second variant of our proposed method, Seq-eMMI, is that it aims to sample along the ϵ-boundary. To do this, it first identifies any point on the ϵ-boundary, and pushes its solutions along the boundary, at a specified tolerance level.

Visually, in Figure 6.4b, after finding a point on the ϵ-boundary (the green dot), the solutions in the population move along the ϵ-boundary (the thick blue line) within some tolerance level (indicated by the thin blue lines), until the budget is exhausted. To this end, it uses a synchronous population-based update rule for its diversity optimisation, rather than the asynchronous iterated approach employed by Conv-eMMI. In Seq-eMMI, the size of this population n_{pop} is analogous to the number of iterations n_{iter} in Conv-eMMI, and likewise, the individual budget for every individual in the population is $\frac{B_2}{n_{pop}}$. In this sequential (Seq) version of eMMI, the optimisation budget of one invdividual in the population is dedicated to finding any point $\boldsymbol{\theta}^\epsilon \in \text{argmin}_{\boldsymbol{\theta}} \left[|f_1(M(\boldsymbol{\theta}), \mathbf{x}) - l^\epsilon| \right]$ on the ϵ-boundary, which will usually be relatively close to the optimum (lines 13-17 in Algorithm 6.1).

Once $\boldsymbol{\theta}^\epsilon$ has been found, we initialise the rest of the population as copies of $\boldsymbol{\theta}^\epsilon$, and optimise for diversity within the population $\boldsymbol{\Theta}$, while constraining the values to remain close to the ϵ-boundary (parameterised by a tolerance hyperparameter

β). The new objective function f_2 (used in line 23 of Algorithm 6.1) for this variant then becomes:

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{maximise}} \quad & f_2(\boldsymbol{\Theta}, \mathbf{x}, \boldsymbol{\theta}, \epsilon) = div(\boldsymbol{\Theta}, \mathbf{x}, \boldsymbol{\theta}) \\ \text{subject to} \quad & l^{\epsilon} - \beta \leq f_1(M(\boldsymbol{\theta}), \mathbf{x}) \leq l^{\epsilon} + \beta \end{aligned} \tag{6.14}$$

In this variant, the set of solutions $\boldsymbol{\Theta}$ contains all the current positions of the population, and the final set of points, like Conv-eMMI, shows an outline of the ϵ-boundary (but its function evaluations will contain both points barely inside the ϵ-manifold and points barely outside of it). Seq-eMMI will not waste function evaluations on sampling far away from the ϵ-boundary, or be pushed away from the boundary by an overpowering push from the diversity objective. It also supports the use of classifiers as the ϵ-manifold representation approach for step 3 as described in Section 6.5.4. However, it can be an inefficient way of exploring the ϵ-boundary, because its steps must be made in a precise direction that does not violate its constraints, and because only the outer-most points in the population can have a large impact on the diversity objective. For example, when imagining a population spread out over a line, only the two outermost individuals could increase diversity by moving away from the rest of the population, while diversity improvements in one direction by individuals in the centre of the line would come at the cost of a reduction of diversity in the other direction, thereby potentially wasting computation on these function evaluations.

DUAL-EMMI

The intuition behind the last variant of our method is that the population is split into two 'competing' halves, with both sub-populations pushing from opposite sides against, but unable to exceed, the ϵ-boundary, thus achieving a balanced data set of in-samples and out-samples in the process.

In Figure 6.4c, the purple dots are trying to 'push' the ϵ-boundary outward from their origin point of the optimum, while the orange dots are trying to push the ϵ-boundary inward from their origin point outside of the ϵ-manifold. Like Seq-eMMI, Dual-eMMI uses a population-based approach. The population is split into sub-population $\boldsymbol{\Theta}_a$, contained inside the ϵ-manifold (initialised to the optimum), and sub-population $\boldsymbol{\Theta}_b$, situated outside of it (initialised randomly). The individuals in $\boldsymbol{\Theta}_a$ will try to maximise their distance from the optimum $\hat{\boldsymbol{\theta}}$, as well as the diversity within the population $\boldsymbol{\Theta}_a$, constrained to not move outside of the ϵ-boundary. Meanwhile, individuals in $\boldsymbol{\Theta}_b$ will still optimise for diversity within their sub-population $\boldsymbol{\Theta}_b$, but will also aim to minimise the distance to the optimum $\hat{\boldsymbol{\theta}}$, constrained to not move within the ϵ-boundary. The balance between the

diversity and distance objectives can be tuned via the hyperparameter α, following a scalarisation approach to multi-objective optimisation [281, 282]. Formally, individuals in $\mathbf{\Theta}_a$ solve:

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{maximise}} \quad & f_{2_a}(\mathbf{\Theta}_a, \mathbf{x}, \boldsymbol{\theta}, \epsilon) = \alpha \cdot div(\mathbf{\Theta}_a, \mathbf{x}, \boldsymbol{\theta}) + (1-\alpha) \cdot \sum_{j=0}^{|P|} |\boldsymbol{\theta}_j - \hat{\boldsymbol{\theta}}_j| \\ \text{subject to} \quad & f_1(M(\boldsymbol{\theta}), \mathbf{x}) \leq l^{\epsilon} \end{aligned} \tag{6.15}$$

Here l^{ϵ} refers to the ϵ-loss. Meanwhile, individuals in $\mathbf{\Theta}_b$ solve:

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{maximise}} \quad & f_{2_b}(\mathbf{\Theta}_b, \mathbf{x}, \boldsymbol{\theta}, \epsilon) = \alpha \cdot div(\mathbf{\Theta}_b, \mathbf{x}, \boldsymbol{\theta}) - (1-\alpha) \cdot \sum_{j=0}^{|P|} |\boldsymbol{\theta}_j - \hat{\boldsymbol{\theta}}_j| \\ \text{subject to} \quad & f_1(M(\boldsymbol{\theta}), \mathbf{x}) \geq l^{\epsilon} \end{aligned} \tag{6.16}$$

Dual-eMMI is more efficient than Seq-eMMI, obtains more diverse samples away from the ϵ-boundary and ensures that there is a reasonable balance between in-samples and out-samples. On the other hand, its two objectives must be combined, with benefits to one objective potentially coming at the expense of the other. Unlike in typical multi-objective settings, where a Pareto-front of non-dominated solutions is the desired outcome, our reliance on diversity and need for training data set samples made this a non-trivial extension to add; however, future work may look further into the use of other types of multi-objective optimisation approaches.

6.5.4. STEP 3: APPROXIMATING THE ϵ-MANIFOLD

Finally, in step 3, we use the points sampled in step 2 to approximate the ϵ-manifold. We required methods applicable to problems of arbitrary dimensionality, because, unlike the two-dimensional examples used in our visualisations, these boundaries cannot be trivially drawn using, e.g., contour lines. In our current approach, there are two options for extracting a representation of the ϵ-manifold from these points.

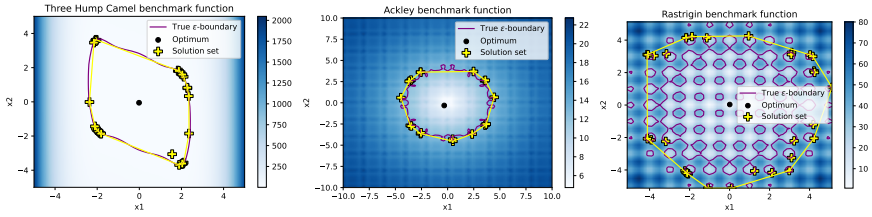
The first approach is to compute the convex hull of the final set of points $\mathbf{\Theta}$. The optimisation procedure would have encouraged points to span the ϵ-boundary, making the convex hull of the resulting point cloud an intuitive representation of the ϵ-manifold that follows directly from the final points without further approximation steps. We used Delaunay triangulation (see, e.g., Lee and Schachter [283]) to check whether a new point is contained in this convex hull. This approach assumes convexity and may fail to compute a convex hull if the assumption is violated, may be costly in higher dimensions, and does not take advantage of the many points sampled during the optimisation procedure.

The second approach is to store the results of the function evaluations for the points sampled during step 2 in a data set H, consisting of the points H_x sampled during step 2 and their labels H_y indicating whether their loss is lower or higher than the ϵ-loss, and train a classifier on these points. Intuitively, the ϵ-boundary can be viewed as an ideal decision boundary for a binary classification problem, separating points inside the ϵ-manifold (positive labels) from points outside of it (negative labels). Therefore, the ϵ-boundary can be approximated by the decision boundary of a binary classifier trained on positive and negative examples in H, that had already been sampled during optimisation. The ϵ-manifold them becomes the space bounded by this decision boundary, or, if finite sets are preferred, the points in H themselves can be used.

The advantages of using a classifier include reduced assumptions on the loss landscape (as different types of classifier can model different types of decision boundaries), improved computational efficiency, and a convenient method of checking whether new points are in the ϵ-manifold. On the other hand, representing ϵ-manifolds as a classifier may be less intuitive than using a convex hull. There can also be technical downsides to such an approach: classifiers usually perform best when presented with balanced data (roughly equal numbers of in- and out-samples), some classifiers may be unable to extrapolate beyond the neighbourhood of the sampled training points (requiring training samples spread throughout $\mathcal{D}_P$), and using classifiers introduces an additional step of imperfect approximation.

To make our method more robust to two specific scenarios it may be weak to, we added two additional options to eMMI. First, we allow users to filter points in low-density regions out of H for probabilistic simulators or loss functions. The rationale for this option is that, in probabilistic cases, parts of the search space with a low sampling density may be misrepresented by one or a few points that returned an unlikely, non-representative result for a single point. By filtering points based on their density, this problem can be alleviated (if computational resources are available for it, it is also possible to sample every point multiple times in these cases).

The second option is to allow users to spend a proportion of the budget of step 2 (B_2) on pure exploration, as in U-eMMI; we refer to this proportion as B_{exp}. Depending on the type of classifier, there may be a benefit of H containing points that sparsely cover the entire search space, to enable extrapolation beyond the area in the direct vicinity of the decision boundary. Both options can be controlled through hyperparameters, which can be tuned automatically through hyperparameter optimisation.



(a) Three hump camel function (b) Ackley function (c) Rastrigin function

Figure 6.5: Examples of Conv-eMMI applied to two-dimensional non-convex optimisation benchmark functions with increasing difficulty. The three hump camel function (Figure 6.5a) is globally non-convex, but locally convex around its three local optima with no sub-maxima within the neighbourhood of the optima. In this case, the optima were close together, so all optima were contained within the ϵ-manifold. The Ackley function (Figure 6.5b) is globally non-convex and locally convex for the level of ϵ, but with multiple sub-optima between the optimum and the ϵ-boundary. In this case, the sub-maxima did not exceed the ϵ-loss, allowing eMMI to effectively extract most of the ϵ-manifold, but with some lower detail at the border. The Rastrigin function (Figure 6.5c) is globally and locally non-convex, with the sub-maxima between the optimum and the ϵ-boundary exceeding the ϵ-loss. In this case, the ϵ-manifold extracted by Conv-eMMI contained many false positives for the parts of the parameter space where sub-maxima exceeded the ϵ-loss, marking the limits of Conv-eMMI in its current form.

6.5.5. LIMITATIONS OF THE CURRENT HEURISTICS

The current heuristics for eMMI (excluding U-eMMI, which is generally applicable but scales poorly) are applicable to loss landscapes where viable solutions are centred around an optimum. This makes our method directly applicable to unimodal, globally convex inversion loss landscapes. Even if a full loss landscape is globally non-convex, our method is still applicable if there is local convexity around the optimum. We describe how the method could be applied to this type of multimodal, locally convex landscape in Section 6.5.2.

Within local convexity, we can further differentiate between monotonic locally convex landscapes and non-monotonic locally convex landscapes. If the landscape is (locally) monotonic, it is also locally convex: if its loss values only strictly increase or stay equal with distance to the optimum, it is impossible for another optimum to exist within this local neighbourhood. If the local landscape is non-monotonic, there may be 'sub'-minima and maxima within the local neighbourhood of the optimum. If these sub-maxima do not exceed the ϵ-loss, eMMI can still be applied to such non-monotonic local landscapes (because even all of its highest values lie between the minimum and the ϵ-loss).

Finally, the current heuristics of eMMI would be less well-suited to landscapes where the sub-maxima of a non-monotonic local landscape exceed the ϵ-loss, in

which case the ϵ-manifold could contain false positives, while the approximated ϵ-boundary is likely to have stopped expanding too early. For this type of problem, we recommend using U-eMMI to avoid the assumption of local convexity, although the lack of efficient sampling heuristics from the other variants may result in poor scalability.

We have visualised example runs of Conv-eMMI (for details, see Section 6.5.3) for three non-convex benchmark functions in Figure 6.5: the three-hump-camel function, which is globally non-convex but monotonically locally convex, the Ackley function [284], which is globally convex, but locally non-monotonically convex, and the Rastrigin function [285], which is globally and locally non-monotonically non-convex. As the Figure shows, Conv-eMMI approximated the ϵ-manifold well for the three hump camel function, performed reasonably well (but showed some inaccuracies along the ϵ-boundary) for the Ackley function, but contained many false positives for the sub-maxima in the Rastrigin function. Therefore, this type of landscape can be considered the limit of the type of ϵ-manifold that can efficiently be approximated by eMMI with the current heuristics (although U-eMMI could still approximate it inefficiently, or other, novel heuristics may perform better).

6.6. EXPERIMENTS

In the following, we explain the details of our computational experiments aimed at answering the following chapter research questions (CRQs):

1. What is the effectiveness of ϵ-manifolds at representing the set of viable solutions $\mathcal{D}_P^{'}$ compared to uncertainty quantification approaches?

2. How does the ϵ-manifold approximation performance of eMMI compare to statistical baseline methods in terms of classification performance on validation points spread around the target variable space?

3. How large does the eMMI budget need to be to converge to its best performance, and how is this impacted by the dimensionality of the problem?

We will first expand on the research questions and the experimental setup we used to answer them in Section 6.6.1, after which we will introduce the simulators used in our experiments in Section 6.6.2, and the baseline methods in Section 6.6.3.

6.6.1. EXPERIMENTAL SETUP

We will explain the motivation behind the research questions, and the experiments we created to answer them, individually per research question.

CRQ1: EFFECTIVENESS OF ϵ-MANIFOLDS

For this research question, we were interested whether a 'perfect' ϵ-manifold would result in a performance increase over generalising existing statistical uncertainty quantification techniques to this purpose. If these ideal ϵ-manifolds significantly outperform uncertainty quantification techniques, it suggests that approximating them is a worthwhile exercise. Conversely, if they do not outperform these methods, it indicates that the assumptions underlying ϵ-manifolds (loss-dependent shifts and loss-dependent origins) may not hold. This would imply that a strong approximation performance by eMMI might not necessarily lead to more robust analyses.

To test for this, we introduce two 'oracle' style methods. These methods enable a direct comparison between ϵ-manifolds and confidence intervals, independent of their approximation efficacy. Given a noisy observation vector $\mathbf{x}$, both methods return a set of potential solutions $\mathcal{V}$ for this model inversion problem. Intuitively, the solution set of a perfect method should always contain the optimum for the (unavailable) noise-free observations $\mathbf{x}^+$, while excluding any solutions that could not have been the true solution if the random noise had been different. In this experiment, we iterated over instances, and performed classification for two points: first, the true values $\boldsymbol{\theta}^+ \approx \hat{\boldsymbol{\theta}}^+$, which should always be contained in the viable solution set, and second, a negative example point that should be excluded.

The first oracle-style method concerns ϵ-manifolds. Classifying the noise-free optimum and any negative example point is straightforward: the pre-computed noisy optimum $\hat{\boldsymbol{\theta}}$ and its loss function value are already known to the oracle method. Therefore, for any ϵ, we can compute the ϵ-loss as $l^\epsilon = f_1(M(\hat{\boldsymbol{\theta}}), \mathbf{x}) + \epsilon$. We classify new points $\boldsymbol{\theta}$ by computing their loss function value $f_1(M(\boldsymbol{\theta}), \mathbf{x})$ and comparing it to the ϵ-loss l^ϵ following Equation 6.6. If the loss-dependent origins assumption holds, we would expect this oracle-based method to classify these points nearly perfectly, bounded only by the suitability of ϵ to the current instance and the accuracy of the optimum $\hat{\boldsymbol{\theta}}$. If the performance is weaker, it could indicate either a large variability of the appropriate setting for ϵ between instances, or that the loss-dependent origins assumption is less strongly satisfied. In other words, a lower score suggests that the loss landscape for a noisy observation corresponds less closely to the probability of being the original, noise-free optimum.

As a baseline, we compared our approach to an oracle-based uncertainty quantification approach using a Gaussian distribution parameterised by mean μ and standard deviation σ. We perform classification using the confidence interval $[\mu - 2\sigma, \mu + 2\sigma]$ as described in Section 6.6.3. The parameters μ and σ were set using oracle knowledge, with the mean $\mu = \boldsymbol{\theta}^+$, and the standard deviation σ derived directly from the evaluation points labelled as being in the ϵ-manifold (see Section 6.6.2 for details). This setup, using unknown true values and computing sample statistics directly from the validation points used to evaluate perfor-

mance, ensured that the Gaussian distribution parameterisation had the strongest possible performance. We expected this baseline method to perform well on the simulation-based inference tasks that are based on distribution parameterisation, but perform worse than ϵ-manifolds for complex loss landscapes, such as those of physical models or dynamical systems.

CRQ2: EMMI PERFORMANCE FOR APPROXIMATING THE ϵ-MANIFOLD

Having shown that ϵ-manifolds can entail substantial performance improvements (CRQ1), the next step is to empirically validate the performance of our proposed approximation method, eMMI.

In all experiments, we set the total function evaluation budget B available to eMMI at 20 000. For the baseline methods (see Section 6.6.3), ABCSMC shared this budget, while the uncertainty quantification baselines do not rely on sampling or optimisation at inference time. Instead, these baseline methods were trained on 20 000 training instances. Prior to running the methods on our main experiments, we performed hyperparameter optimisation for all methods using SMAC3 [286] for 48 hours to ensure that the methods were properly configured. For a single hyperparameter configuration, we evaluated the performance in batches of 10 instances to make the procedure more robust to noisy simulations and f_1 function evaluations.

We measured the performance of the different methods based on the classification performance (notably accuracy) on a balanced set of validation points, as described in Section 6.6.2. A high classification performance indicates that a large proportion of the validation points were correctly classified to be either inside or outside of an oracle ϵ-manifold (as used in CRQ1), thereby indicating effective approximation.

CRQ3: EMMI BUDGET AND SCALABILITY

The eMMI method requires an optimisation procedure to identify the optimum $\hat{\boldsymbol{\theta}}$ (step 1), after which it must spend more function evaluations to sample points around the search space (step 2). As a result, running eMMI will often be more computationally intensive than running the baseline methods at inference time, and the applications for eMMI may differ from those of uncertainty quantification (see Section 6.4.3). For CRQ3, we were interested in how much budget B eMMI needs to converge to a strong classification performance, and how this budget is affected by the dimensionality of the problem.

To test this, we ran eMMI on 50 instances for all versions of the multi-dimensional linear regression simulator (see Section 6.6.2), which we configured for 2, 5, 10, 20 and 50 dimensions. We ran eMMI with a budget of 50 000 function evaluations, and trained a classifier on subsets of the sampled points in steps of 500 additional

evaluations (i.e., $0-500$, $0-1000$, $0-1500$, etc). This simulated different step 2 sampling budgets B_2. We then plotted the average classification performance over the instances as a function of the budget, for all 5 dimensionalities. This plot will show how many function evaluations are needed to converge to a stable performance, as well as showing whether larger-dimensional problems require a larger budget to converge.

GENERAL EXPERIMENTAL WORKFLOW

For all experiments described above, we first needed a suitable value for ϵ for every simulator, based on the expected optimum shift. Although the desired level of confidence is up to the user, we designed the following process to set the value of ϵ to correspond to the 95% confidence intervals of UQ, thereby allowing a comparison between the two approaches. To automatically determine this ϵ value, we loaded the original true target variable configuration $\boldsymbol{\theta}^+$ for the validation instances, along with the pre-computed optimum $\hat{\boldsymbol{\theta}}$ for the noisy observations $\mathbf{x}$ of the same instance, and computed the increase in loss value between the simulated output for the true configuration $M(\boldsymbol{\theta}^+)$ and the simulated output for the pre-computed optimum $M(\hat{\boldsymbol{\theta}})$. Finally, we derived the appropriate value for ϵ as the 95th percentile of the differences in f_1 values between the pre-computed optima and true configurations. Here we note that, despite this statistical approach to setting ϵ, the relevant statistics are still loss function values, and not distances in the target variable space.

Every experiment was run on a compute cluster with an Intel Xeon E5-2683 v4 CPU and 128GB RAM per node, of which we reserved 16GB per experiment.

In the following, we will introduce the data sets (simulators), baselines and performance metrics used in our experiments.

6.6.2. SIMULATION MODELS

Our workflow for the simulators is highly modular, and adding new simulators to our implementation consists only of adding a description of its input parameters (target variables) and a function call for the forward simulation, allowing users to incorporate their own simulators if they wish to. We performed empirical experiments on the following 7 simulator models representing physical models (Earth science), dynamical systems, simulation-based inference and machine learning problem settings. We will introduce the simulation models grouped by the type of model.

Physical models (1 model, 2 versions). As our representative physical model we used PROSAIL, [190, 43, 44] a radiative transfer model (RTM) that is extensively used in real-world applications estimating vegetation properties from remotely

sensed spectral data. We selected the 4 most impactful variables (leaf area index (LAI), chlorophyll $a+b$ content, average leaf angle, and dry matter content) of this model to perform our experiments, while fixing the remaining variables to static values. We also included experiments with only two of the most important variables, to show the impact of the dimensionality of the problem for physical models, and enable the visualisations of Figures 6.1b and 6.3.

Dynamical systems (2 models). We used the implementation of dynamical systems simulation models provided by DAPPER [287], from which we selected the Two Pendulums (TP) and Lorenz63 models. In dynamical systems, the parameters of the simulator represent the state of some system, that is updated over multiple time steps. The observations, in this case, consists of the state of the system after updating for a user-defined number of time-steps. Inverting this type of model is often ill-posed, and the system is often considered chaotic (dynamic systems with a strong sensitivity to the initial conditions) [288]. Finding a set of potential solutions through uncertainty quantification in such ill-posed landscapes would be challenging, because it violates the assumptions of the distributions, while the current versions of eMMI will likewise not be well suited to these landscapes due to the convexity assumptions (though ϵ-manifolds themselves may still be highly effective, if they are approximated well). For both simulators, we limited our inversion to a single time step, as the compound of uncertainty over multiple time steps of inversion may quickly get computationally infeasible.

Simulation-based inference (2 models). We used the Gaussian mixture (GM) [251] and Two Moons (TM) tasks from the simulation-based inference benchmark by Lueckmann et al. [219] to evaluate our proposed method. Both were selected for their tractable execution time, with GM representing a relatively simple problem conforming to the assumptions made by eMMI and the baseline methods, while TM represented a more complex, bimodal and often non-convex landscape.

Machine learning parameterisation (1 model, 5 versions). As explained in Section 6.1, machine learning training is a special kind of model inversion problem, where its parameterisation (usually model weights) must be set in such a way that, when combined with a data set of features, its simulated output (predictions) has a minimal distance to the observations (ground truth data). Therefore, although the predictions and their quality-of-fit are generally the variables of interest in machine learning, the weight parameters are the variables being inferred during the training procedure. Although it is unlikely that ϵ-manifolds can be computed for deep neural networks with numbers of parameters orders of magnitude higher than typical problem cases, we consider the analyses on traditional machine learning algorithms enabled by ϵ-manifolds (for example, certain types of robustness analysis, loss landscape analysis, ill-posedness analysis, and data set difficulty) to have a high potential for impact.

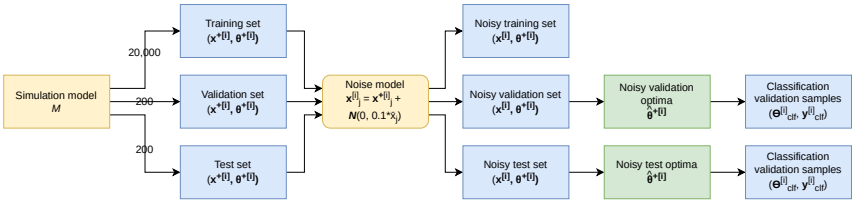


Figure 6.6: Visualisation of our data generation pipeline.

The simulation model we used consisted of an d-dimensional linear regression (LR) model, with $d+1$ trainable parameters (the weights and a bias term). We included problems with a dimensionality d of 2, 5, 10, 20, and 50 weights. For every dimensionality, we pre-generated d independent feature vectors for 100 examples (we refer to the instances of the machine learning task as examples to avoid confusion with the model inversion instances), which we kept constant for all instances. We simulated ground truth data using randomly generated model parameterisations, after which we perturbed the simulated ground truth to emulate noisy training data – a realistic scenario, for which ϵ-manifolds could be used in the same manner as for the other model inversion tasks.

EXPERIMENTAL DATA GENERATION

We visualised the general workflow for data generation in Figure 6.6. We have organised the data generation steps as *phases*. The data required by our experiments consisted of, for every instance of every simulator, i) a true target variable configuration $\boldsymbol{\theta}^+$ and its simulated output (observations) $\mathbf{x}^+ = M(\boldsymbol{\theta}^+)$, ii) a noisy version $\mathbf{x}$ of the observations $\mathbf{x}^+$ and the pre-computed true noisy optimum $\hat{\boldsymbol{\theta}}$, and iii), a balanced set of validation points $\boldsymbol{\theta}_{clf}$ with associated labels y_{clf}, such that $y_{clf} = 1$ if and only if $f_1\big(M(\boldsymbol{\theta}),\mathbf{x}\big) \leq f_1\big(M(\hat{\boldsymbol{\theta}}),\mathbf{x}\big) + \epsilon$, and $y_{clf} = 0$ otherwise.

Phase 1: generating noise-free and noisy data. For every simulator M, we generated a training-, validation- and test set. The training data set was used by eMMI (if warm-starting the optimisation) and the uncertainty quantification baselines (for details, see Section 6.6.3), and consisted of 20000 instances. The validation- and test sets contained 200 instances each. To create these data sets, we generated individual instances i by randomly sampling a true input vector $\boldsymbol{\theta}^{+[i]}$ from the target variable space $\mathcal{D}_P$, for which we created a noise-free simulated observation vector $\mathbf{x}^{+[i]}$ by performing a simulation on $\boldsymbol{\theta}^{[i]}$: $\mathbf{x}^{+[i]} = M(\boldsymbol{\theta}^{+[i]})$. We then saved the pair of noise-free observations and true configurations $(\mathbf{x}^{+[i]}, \boldsymbol{\theta}^{+[i]})$ for all instances i. After this, we created a noisy version $\mathbf{x}^{[i]}$ of the noise-free observations $\mathbf{x}^{+[i]}$ by adding 10% additive Gaussian noise to the elements j of $\mathbf{x}^{+[i]}$:

$\mathbf{x}_j^{[i]} = \mathbf{x}_j^{+[i]} + \mathcal{N}(0, 0.1 \cdot \overline{x}_j)$. Here $\overline{x}_j$ is the mean value of target variable P_j in the sample of noise-free observation instances.

Phase 2: pre-computing the noisy optimum. Every instance in the validation- and test sets required a 'ground truth' global optimum $\hat{\boldsymbol{\theta}}$ to the loss landscape of $f_1\big(M(\boldsymbol{\theta}), \mathbf{x}\big)$, allowing us to compute the true ϵ-loss and evaluate methods later. To this end, we performed 50 000 iterations of black-box optimisation to pre-compute the optimum $\hat{\boldsymbol{\theta}}^{[i]}$ of the instance for the noisy observations $\mathbf{x}^{[i]}$ (using random search with a very high budget to avoid confounding through the choice of optimisation algorithm). Because this budget is much larger than the budget used by eMMI in practice, pre-computing the optima in this manner enabled reliable and efficient evaluation of method performance later, although the high computational cost limited our experiments on simulation models with tractable computational costs. In the interest of conserving computational resources, we did not pre-compute optima for the training set, where the ground truth values $\boldsymbol{\theta}^+$ could be used for training.

Phase 3: generating validation points. In addition to the simulation instances themselves, every instance in the validation- and test sets required a sample of points $\boldsymbol{\Theta}_{clf}^{[i]}$, consisting of individual points $\boldsymbol{\theta}_{clf}^{[i]}$, allowing us to evaluate the classification performance, as explained in Section 6.5.4, for the ϵ-manifold of that instance. These points were associated with the label vector $\mathbf{y}_{clf}^{[i]}$, whose labels denote if the instance is inside or outside of the ϵ-manifold.

To ensure that we obtained balanced validation data sets (especially in higher dimensions), we performed random sampling with a large budget of 100 000 function evaluations, and performed post-hoc rejection sampling to obtain a balanced number of true and false evaluation examples. Computing these evaluation samples, along with the pre-computed optima mentioned above, is highly computationally intensive, and these elements together form the largest computational bottleneck of our experiments. We note that this computational load was used only to ensure a *fair empirical evaluation* of the methods, and it would not be a factor when applying our proposed method or baseline methods to real model inversion problems.

A summary of the data set characteristics can be found in Table 6.1.

6.6.3. BASELINES AND PERFORMANCE METRICS

In our experiments for CRQ1 we compared ϵ-manifolds to an oracle-based uncertainty quantification (UQ) baseline in the form of parameterised Gaussian distributions; we explain this baseline in Sections 6.6.1, since it does not necessarily represent any particular method. In contrast, when evaluating eMMI, we compared the empirical performance of our method to those of concrete baseline methods.

Property	Data split		
	Training set	**Validation set**	**Test set**
# noise-free instances	20 000	200	200
# noisy instances	20 000	200	200
# validation points / instance	0	2 – 100 000	2 – 100 000
contains noisy optimum	✗	✓	✓

Table 6.1: Summary of the data generated for every simulator. Because we used rejection sampling to generate validation points for every instance, the number of points can vary between 2 and 100 000 points.

These methods can be highly effective at their intended purpose of inference and uncertainty quantification; our empirical comparison will, therefore, not evaluate the value of the methods themselves, but rather gauge whether such existing methods can be generalised such that the resulting statistical confidence intervals can be interpreted as an effective approximation of ϵ-manifolds. We compared against the following baselines:

- **Gaussian processes (GP).** We used GPs as an UQ baseline because these are the primary prediction model preferred by domain users of physical models [46, 54], where GPs are preferred in part because of their inherent uncertainty estimation. It is, therefore, useful to test whether the uncertainty estimation of GPs approximates the actual ϵ-manifold of a problem instance.
- **Random forests (RF).** RFs are a frequently used ensemble method, allowing for uncertainty quantification through the standard deviation of the predictions of individual trees following an ensembling approach to uncertainty quantification [262, 263].
- **Bayesian neural networks (BNN).** BNNs represent advances in neural network-based approaches boasting impressive performance. To perform inference with the BNN, we computed the standard deviation of the predictions of the model, which, due to the weights being a distribution, are not deterministic. The model itself was based on the implementation provided by Lee et al. [289], although we automated the selection of the architecture using hyperparameter optimisation.
- **Approximate Bayesian computation – sequential Monte Carlo (ABCSMC) [290].** We used this recent variant of ABC as a representative method for conventional SBI methods. Although SBI methods and ϵ-manifolds are not strictly competitive (see Section 6.2), in principle, the inherent statistical nature of SBI could be interpreted as an equivalent to ϵ-manifolds, making this

a meaningful comparison. Since ABC usually infers a (non-parametric) posterior distribution over the target variables, a 95% confidence interval can be constructed by including points whose weights lie between the 2.5th and 97.5th percentiles.

- **Tabular prior-fitted network (TabPFN)** [291, 292]. TabPFN is a state-of-the-art foundation model for tabular data, able to perform zero-shot inference on tabular data sets, often with a strong performance comparable to specialised models. When performing inference, the model can include percentiles in its predictions, enabling the construction of 95% confidence intervals. To prevent out-of-memory issues, we increased the memory available to this method fourfold to 64GB.

For all machine learning-based baselines we trained (GP, RF, BNN) or fine-tuned (TabPFN) the model on 20 000 simulated training instances (with noise), and we also trained the machine learning model used to warm-start the eMMI optimisation from step 1 (see Section 6.5.2) on this training set. We quantified uncertainty for the UQ methods as the 95% confidence interval, given by two standard deviations σ from the mean unless otherwise specified. This confidence interval matches the confidence interval used to derive ϵ, as described in Section 6.6.1. Given a model inversion instance i, for which the UQ methods predict a mean $\mu^{[i]}$ and a standard deviation $\sigma^{[i]}$, and a sample of pre-computed evaluation points for instance i, the points within the ϵ-manifold should lie within the interval $[\mu^{[i]} - 2\sigma^{[i]}, \mu^{[i]} + 2\sigma^{[i]}]$.

To evaluate the performance of the different methods numerically, we computed the accuracy (suitable because we strictly enforced data set balance using our post-hoc rejection sampling approach) for the different methods on all model inversion problems. We determine the significance of a performance difference using Wilcoxon signed-rank tests at a significance level $\alpha = 0.05$. We note that the empirical results for these experiments are not designed to test the quality of the predictions of these methods themselves, but rather the suitability of the uncertainty quantification components of existing methods for approximating an ϵ-manifold.

6.7. RESULTS

The empirical results presented in the following are organised per chapter research question (CRQ).

Dataset	ϵ-manifold	Uncertainty quantification
PROSAIL	$\mathbf{1.0 \pm 0.0}$	0.84 ± 0.37
PROSAIL 2D	0.68 ± 0.47	0.74 ± 0.44
TP	$\mathbf{0.8 \pm 0.4}$	0.62 ± 0.49
Lorenz63	$\mathbf{0.9 \pm 0.3}$	0.62 ± 0.48
GM	0.86 ± 0.35	0.89 ± 0.31
TM	$\mathbf{0.97 \pm 0.17}$	0.74 ± 0.44
LR	$\mathbf{0.86 \pm 0.34}$	0.76 ± 0.43

Table 6.2: Results for the oracle-based ϵ-manifold validation experiment for RQ2, showing accuracy scores for a classification task where, for every instance, the true values $\boldsymbol{\theta}^+$ had to be predicted along with a negative sample from the validation points. A bold column in the table represents a significantly better result, determined by a Wilcoxon signed-rank test at a significance level $\alpha = 0.05$, where the samples consisted of correct or incorrect classification (thereby enabling the computation of a standard deviation and rank-based comparisons).

6.7.1. CRQ1: EFFECTIVENESS OF ϵ-MANIFOLDS

The results for CRQ1 can be found in Table 6.2. As the table shows, a perfect ϵ-manifold performed significantly better than a perfect Gaussian distribution parameterisation as uncertainty quantification on nearly all tested simulation models, only tying on the Gaussian mixture and PROSAIL 2D simulators (both of which have loss landscapes that adhere relatively well to the assumptions of a Gaussian statistical kernel). This means that an ϵ-manifold for a noisy model inversion instance is more likely to contain the true solution $\boldsymbol{\theta}^+$, without needlessly including infeasible points.

Despite the strong advantage over perfectly parameterised uncertainty quantification, the performance of ϵ-manifolds appears to vary between simulation models, especially between the full version of PROSAIL (perfect scores) and the two-dimensional version of PROSAIL (lowest scores out of the tested simulators). This counterintuitive result implies that higher-dimensional problems are not necessarily more difficult to deal with, in terms of uncertainty quantification, than higher-dimensional manifolds.

In the case of PROSAIL, this behaviour might be explained through domain knowledge by examining the properties of the physical model, in which the leaf area index (LAI) parameter is known to have an impact on the behaviour of other parameters. In lower-dimensional settings, the relative impact of the complexities introduced by the LAI parameter is higher than it is in higher-dimensional settings, where the other model parameters may have a more modest and independent impact on the loss landscape. It is also possible that the appropriate setting of ϵ, which we set constant for all instances, is more variable for low-dimensional

problems than for high-dimensional settings, in which case future work aimed at further improving the heuristic by which ϵ is set might hold significant promise.

The results for the dynamic systems simulators (Two Pendulums and Lorenz63) were quite favourable for ϵ-manifolds compared to UQ. This follows expectations, considering the chaotic nature of these simulators, which a Gaussian distribution may be ill-suited to represent. The results for the probabilistic simulation-based inference simulators (GM and TM) also both favoured ϵ-manifolds over uncertainty quantification, as did those for the linear regression (LR) machine learning model parameterisation task.

We found that the lower accuracy of uncertainty quantification often stemmed from a combination of near-perfect precision with low recall or, more rarely, high recall with poor precision (these results can be found in Appendix B.2). This frequent overconfidence may be explained in part by a low prior probability of some viable solutions, where a solution that is far removed from the point prediction (and therefore has a low probability) would be considered unlikely, regardless of whether there is a large difference in the loss function value. Therefore, these results support the intuition we described in Section 6.4.3 on why statistical uncertainty quantification may not always be an appropriate choice for finding viable solution sets. Alternatively, the loss landscape may simply have not adhered to a Gaussian distribution form, such as in the chaotic landscapes of the dynamical systems simulators (as the particularly low accuracy scores for TP and Lorenz63 also indicate).

In conclusion, regarding CRQ1, perfect ϵ-manifolds appear to be more effective than perfect uncertainty quantification for identifying the set of potentially valid solutions in most noisy model inversion problems. This indicates that UQ methods, while highly valuable when applied to their intended purpose, cannot be directly generalised to approximate ϵ-manifolds.

6.7.2. CRQ2: EMMI PERFORMANCE

The results for our experiments validating eMMI as a method to approximate ϵ-manifolds can be found in Tables 6.3 (comparing eMMI to baseline methods) and 6.4 (comparing eMMI variants).

The results in Table 6.3 show that eMMI performed significantly better than the baseline methods we considered on all tested simulators, with a generally high accuracy that only dropped for Lorenz63 (a chaotic system) and TM (a bimodal problem), both of which give rise to loss landscape types for which the current heuristics were not designed. The baseline methods, as might be expected based on the results from Table 6.2, often achieved an accuracy close to 0.5, indicating overconfidence with a low recall (precision and recall results can be found in Ap-

Method	PROSAIL	PROSAIL 2D	TP	Lorenz63	GM	TM	LR
RF	0.54 ± 0.06	0.55 ± 0.08	0.5 ± 0.0	0.5 ± 0.01	0.88 ± 0.12	0.57 ± 0.19	0.5 ± 0.0
GP	0.74 ± 0.1	0.61 ± 0.13	0.51 ± 0.03	0.5 ± 0.0	0.57 ± 0.16	0.5 ± 0.0	0.5 ± 0.0
BNN	0.53 ± 0.09	0.73 ± 0.16	0.51 ± 0.03	0.5 ± 0.0	0.58 ± 0.13	0.5 ± 0.0	0.5 ± 0.0
ABCSMC	0.52 ± 0.16	0.53 ± 0.2	0.52 ± 0.17	0.49 ± 0.1	0.45 ± 0.18	0.53 ± 0.17	0.5 ± 0.0
TabPFN	0.5 ± 0.0	0.5 ± 0.0	0.5 ± 0.0	0.5 ± 0.0	0.42 ± 0.05	0.5 ± 0.0	0.5 ± 0.0
eMMI	$\mathbf{0.87 \pm 0.1}$	$\mathbf{0.89 \pm 0.09}$	$\mathbf{0.88 \pm 0.12}$	$\mathbf{0.57 \pm 0.1}$	$\mathbf{0.97 \pm 0.07}$	$\mathbf{0.73 \pm 0.14}$	$\mathbf{0.89 \pm 0.12}$

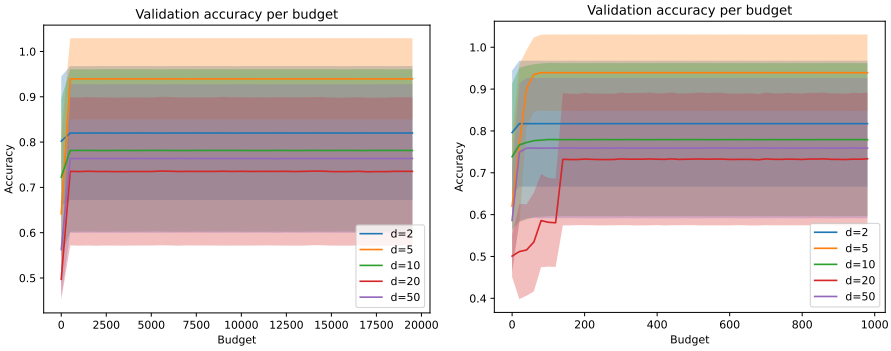
Table 6.3: Accuracy of the different methods approximating the ϵ-manifolds, with all hyperparameters for all methods (including the appropriate eMMI variant) automatically determined through hyperparameter optimisation. For every simulator, the best performance has been marked in **boldface**, with statistical significance determined by a Wilcoxon signed-rank test at significance level $\alpha = 0.05$. These results suggest that existing statistical methods cannot be generalised to effectively approximate ϵ-manifolds, necessitating new, specialised frameworks (i.e., eMMI).

Method	PROSAIL	PROSAIL 2D	TP	Lorenz63	GM	TM	LR
U-eMMI	0.55 ± 0.09	0.72 ± 0.19	0.56 ± 0.07	0.5 ± 0.01	$\mathbf{0.99 \pm 0.03}$	$\mathbf{0.76 \pm 0.17}$	$\mathbf{0.92 \pm 0.15}$
Conv-eMMI	$\mathbf{0.68 \pm 0.15}$	0.82 ± 0.1	0.54 ± 0.03	$\mathbf{0.5 \pm 0.02}$	$\mathbf{0.99 \pm 0.04}$	0.75 ± 0.11	0.51 ± 0.01
Seq-eMMI	$\mathbf{0.69 \pm 0.17}$	0.77 ± 0.15	0.55 ± 0.06	0.5 ± 0.01	0.98 ± 0.06	0.68 ± 0.12	0.89 ± 0.11
Dual-eMMI	$\mathbf{0.72 \pm 0.18}$	$\mathbf{0.88 \pm 0.1}$	$\mathbf{0.58 \pm 0.08}$	0.5 ± 0.01	0.97 ± 0.06	0.73 ± 0.14	0.91 ± 0.11

Table 6.4: Accuracy of approximating the ϵ-manifolds by individual eMMI variants. For every simulator, the best performance has been marked in **boldface**, with statistical significance determined by a Wilcoxon signed-rank test at significance level $\alpha = 0.05$.

pendix B.2). This pattern was likely caused by many solutions that could satisfy observations with low prior probabilities not being represented in the training data set. In some cases, such as PROSAIL 2D, the results for eMMI were better than those for the oracle-based ϵ-manifolds in Table 6.2. This is possible, because the results in Table 6.2 measure the efficacy of ϵ-manifolds themselves at including the true solution, while the results in Table 6.3 measure the efficacy of eMMI in approximating the ϵ-manifold. Therefore, the accuracy would still be expected to be relatively low for PROSAIL 2D, despite eMMI approximating the ϵ-manifold for this simulator well.

The results in Table 6.4 indicate that the different variants of eMMI perform well on different types of simulators, supporting our view that having multiple eMMI variants (whose selection can be automatically tuned using hyperparameter optimisation, as we did) can be beneficial. U-eMMI, Conv-eMMI and Dual-eMMI all performed significantly best on 3 simulators (subject to possible ties in performance); however, out of the four variants we tested, we consider Dual-eMMI to have achieved strong results most reliably. Even in cases where U-eMMI performed significantly better than Dual-eMMI (GM, TM and LR), the differences in performance were very small, while in cases where U-eMMI performed poorly (PROSAIL), Dual-eMMI performed substantially better.



(a) $B_2 = 20\,000$ (b) $B_2 = 1\,000$

Figure 6.7: Average classifier accuracy over 50 instances of the linear regression simulator (dimensionality d of 2, 5, 10, 20, and 50 parameters) when trained on a dataset consisting of points sampled by eMMI, as a function of the eMMI sampling budget B_2, for a maximum budget of 20 000 (Figure 6.7a) and 1 000 (Figure 6.7b). In all cases, the classifier converged to a stable performance very early on (< 200 samples), even for the 50-dimensional linear regression model, although the variability between instances was high.

In conclusion, regarding CRQ2, the current form of eMMI appears to be able to capture ϵ-manifolds better than UQ-based methods, indicating that even sophisticated UQ methods cannot be generalised to ϵ-manifold approximation. However, when faced with chaotic systems (such as the dynamical systems simulators), the performance improvement over UQ can be small. In those cases, none of the methods performed particularly well. Based on these results, eMMI appears effective at approximating ϵ-manifolds compared to UQ-based approximations, in a broad range of applications (physical model inversion, simulation-based inference and machine learning model parameterisation), but is not yet universally applicable to all possible types of loss landscapes.

6.7.3. CRQ3: BUDGET AND SCALABILITY

The results for our experiment aimed at answering CRQ3 on scalability can be found in Figure 6.7. As the figure shows, the classifier tend to converge to a stable performance in fewer than 200 samples, indicating that the budget B_2 for step 2 does not need to be overly large for eMMI to achieve good performance on the simulators we tested. Our results further indicate that the dimensionality of the problem did not affect this pattern much: although performance on problems of higher dimensionality tended to be lower than lower dimensional models, this was not always consistent (e.g., performance was slightly worse for $d = 20$ than $d = 50$).

This result implies that eMMI may be applicable to larger-scale model inversion problems than expected, because the budget necessary to converge for the largest problem we tested (up to 50-dimensional linear regression) was less than 1% of the total budget B we made available in our experiments (20 000). Although part of this budget needs to be spent on finding the optimum $\hat{\boldsymbol{\theta}}$, this cost may be alleviated using, e.g., the hybrid model warm-starting approach we deployed in our experiments.

In conclusion, regarding CRQ3, it appears that eMMI requires a surprisingly low amount of sampling budget to converge to its final performance level, and that its convergence speed is not heavily affected by the dimensionality of the problems under consideration. Efficiently sampling around the optimum $\hat{\boldsymbol{\theta}}$ to train a classifier, thereby approximating the ϵ-manifold, appears to only require a fraction of the function evaluation budget that would, in any case, need to be spent on finding the optimum $\hat{\boldsymbol{\theta}}$ in the first place. Therefore, eMMI would be applicable to many problem settings any SBI or black-box optimisation algorithm is applicable to, since the marginal cost of approximating the ϵ-manifold would be small.

6.8. CONCLUSIONS AND FUTURE WORK

In this chapter, we addressed the problem of finding viable solution sets to model inversion problems. Firstly, we introduced ϵ-manifolds, i.e. manifold of potentially valid solutions to a model inversion problem instance. We formalised these ϵ-manifolds and their core assumptions, as well as ϵ-manifold sets, and derived several theoretical properties. Secondly, we proposed eMMI, a method to automatically approximate ϵ-manifolds. We introduced four variants of our method to approximate ϵ-manifolds: U-eMMI, Conv-eMMI, Seq-eMMI and Dual-eMMI, each of which relies on different sampling heuristics, mostly based on diversity optimisation techniques, to extract the ϵ-manifold. We performed computational experiments evaluating the advantages of perfect ϵ-manifolds using an oracle-based approach, and compared the practical ϵ-manifold approximation performance of eMMI compared to statistical baseline methods. The results from these experiments demonstrate that ϵ-manifolds are much more effective than an oracle-based uncertainty quantification approach at including the true solution, indicating that existing statistical frameworks may not be sufficient to address the problem, thereby necessitating the use of ϵ-manifolds. The eMMI heuristics were effective at approximating the ϵ-manifold compared to statistical uncertainty quantification baseline methods.

Access to these ϵ-manifolds can improve the interpretability of model inversion and inference results, contribute to a greater understanding of the scientific processes underlying the simulation models, and enables novel types of analyses.

These analyses can extend beyond the traditional domain of simulation-based inference, such as the training of machine learning models and robustness analyses. A large ϵ-manifold may suggest ill-posedness, while the degree to which an ϵ-manifold shape conforms to statistical priors (e.g., Gaussian) enables practitioners to decide whether to trust statistical uncertainty quantification on their inference predictions. In the context of biophysical parameter estimation, we can use ϵ-manifolds to describe the behaviour we observed in Chapter 5 and determine its impact on parameter estimation results. They may also contribute to the discovery of new scientific patterns; for example, if the "LAI saturation" of Section 6.4.3 had not been a known phenomenon, the long-tailed ϵ-manifold for this parameter dimension in Figure 6.1b would have shown it.

Future work building upon our research presented here could focus on fundamental extensions of our proposed framework, as well as exploring novel applications of ϵ-manifolds. Examples of further explorations we would recommend include dynamic settings for ϵ, as opposed to the static simulator-wide settings we have been using, extending the heuristics of eMMI by introducing new objective functions f_2 or constraints, or incorporating surrogate model-based approaches. Amortised approaches, similar to the amortisation of Bayesian inference methods, could also be explored, as could approaches optimising in a latent space with reduced dimensionality (we explore this idea further in Section 7.2.2). As further applications, we believe it would be interesting to explore machine learning data set difficulty analyses through a comparison of viable parameterisations between instances, identifying manifolds of possible counterfactual predictions for interventions, or performing adversarial robustness analyses.

This concludes the technical contributions of this thesis. We have now covered every step of the parameter estimation pipeline shown in Figure 1.1. Chapters 3 and 4 have addressed Challenge 1 by answering RQs 1 and 2, thereby improving data consistency, while Chapters 5 and 6 addressed Challenge 2 by answering RQs 3 and 4. Although eliminating the ill-posedness of Challenge 2 altogether is likely infeasible through methodological contributions alone, our work in Chapter 5 resulted in concrete recommendations to alleviate the ill-posedness in a data-centric manner, while the concepts and method introduced in this chapter enable users to judge whether their specific parameter estimation results are reliable through an inspection of the ϵ-manifold.

In the next chapter, we will reflect further on the findings throughout Chapters 3–6, and answer the research questions from Section 1.1.

7

GENERAL DISCUSSION AND CONCLUSION

We are now ready to look back at the research questions introduced in Chapter 1 and answer them. Over the last four chapters, we have introduced novel methods and analyses to improve parameter estimation from EO data. These contributions all tackled the problem from different angles, motivated by the particular challenges of parameter estimation using EO data (see Section 1.1):

- Challenge 1: Data inconsistency and spatio-temporal gaps (EO and ground truth)
- Challenge 2: Noise and ill-posedness on the inference problem

The answers to the research questions will contribute toward reducing the impact of these challenges.

7.1. ANSWERING RESEARCH QUESTIONS

We will answer the research questions, explained in Chapter 1, one by one.

7.1.1. RQ1: SPATIAL INTERPOLATION

The first research question we will answer is RQ1: *How can we effectively interpolate spatial data such that both local and global spatial properties are retained?* (Chapter 3)

For this research question we primarily focused on the spatio-temporal ground truth data (e.g., sensor network data) that is affected by Challenge 1. Existing methods were subject to many limitations, mainly regarding the tradeoff between modelling either local or global spatial relationships, and assumptions, including stationarity and isotropy. In Chapter 3, we proposed a novel spatial interpolation method, VPint (value propagation interpolation), capable of addressing these limitations and without assuming stationarity or isotropy. VPint incorporates a novel system-oriented perspective, as an alternative to the local- or global perspectives offered by existing methods. In this approach, inspired by Markov reward processes (MRPs), the values of known grid cells are propagated through the values of unknown grid cells, enabling spatial interactions at arbitrary distances and paths, while enabling specialised local interactions between a cell and all its neighbours. By iterating the core update rule, this system will converge to a stable state where all values have been interpolated.

We proposed two variants of VPint in Section 3.4: SD-MRP, which requires no additional data and propagates values at a static weight (discount rate) at every cell, and WP-MRP, which leverages additional datasets of covariates that are known for the full grid, enabling the use of predicted, locally specialised spatial weights between neighbouring cells. We found VPint, especially WP-MRP, to be a method that effectively interpolates missing spatial data, that satisfies the requirement of retaining both local and global spatial properties. In our experiments, VPint performed better than baseline methods representing both powerful interpolation approaches (Gaussian processes) and advanced machine learning-based approaches incorporating spatial statistics and automated machine learning techniques. VPint also converged to a stable state in relatively few iterations (around 20), and WP-MRP needed only a modest correlation (about 0.1) between the features and target variables to reliably perform better than SD-MRP. *In conclusion, for RQ1, we can effectively interpolate spatial data such that both local and global spatial properties are retained through our proposed method, VPint.*

Limitations: We found that VPint did not yet generalise well to spatio-temporal interpolation problems. In spatio-temporal settings the method, in its current form, should be applied independently for every time step. We also found the performance of VPint to be less favourable for spatially clustered missing data than for randomly missing data, which hampers its application to, e.g., cloud removal tasks in EO data. We, therefore, recommend VPint for problems where data gaps are more uniformly distributed over the space, such as sensor network data or data with a low sampling rate (few measurement points over a long spatial distance).

7.1.2. RQ2: EO DATA INTERPOLATION

The second research question we will answer is RQ2: *How can we effectively and easily interpolate unpredictable, spatially clustered missing data in Earth observation imagery?* (Chapter 4)

For this research question we focused on the EO input data that is also affected by Challenge 1. Although our focus was on missing data due to cloud cover, the methods proposed for this problem would generalise to any type of missing data in EO imagery that would share many of the same challenges, such as gaps introduced by non-overlapping spatial coverage or Landsat ETM+ SLC-off data gaps. Existing state-of-the-art cloud removal methods often come in the form of specialised deep neural networks, which show impressive numerical performance on curated datasets, but are difficult to apply in practice and typically need to be re-trained for every type of sensor, band subset, resolution and type of missing data. The resulting limited practical uptake means that, most of the time, cloud removal is performed by mosaicking cloud-free pixels from older observations into the cloudy pixels of new images. Such approaches are convenient, but have substantial drawbacks in terms of numerical performance.

VPint, presented in Chapter 3, was a good candidate as an alternative cloud removal method with the same requirements as mosaicking approaches, while potentially offering far greater numerical performance. The existing VPint algorithm, however, could not be applied effectively to this new problem setting without substantial modifications and extensions.

In our proposed VPint2 algorithm, we kept the core concepts of VPint interpolation, but modified WP-MRP to compute highly precise spatial weights from a reference image, instead of estimating them from correlated covariates. This weight computation approach, which is available only in special cases, such as EO data and similar spatio-temporal settings, where regular measurements of the exact same spatial area are taken, enabled the application of VPint2 to image processing tasks. In addition to this modification, we introduced identity priority and elastic band resistance to the VPint2 update rule. These extensions greatly improved the stability and performance of the method for EO data, as they limit the impact of spatial relationships that are likely to have changed between the reference image and the target cloudy image.

Through our experiments, we found VPint2 to perform significantly better than competing methods, including specialised deep neural networks, in 17 out of 20 conditions spanning diverse geographical locations, land cover classes and temporal distances of the reference image. Unlike the original VPint algorithm, which performed notably worse for larger, spatially clustered missing data, VPint2 was not strongly affected by the size of the clouds being removed. The temporal distance of the reference image had a modest impact on the performance of VPint2

and the competing methods. We further found that many of the cloud removal methods showed a particularly strong performance in different parts of the images, and an oracle experiment demonstrated great potential for improvement in cloud removal performance if an effective ensembling method could be found. *In conclusion, for RQ2, we can effectively and easily interpolate unpredictable, spatially clustered missing data in Earth observation imagery using our proposed method, VPint2.*

Limitations: The current VPint2 method relies on a single, completely cloud-free reference image. In some applications, such an image may be difficult to obtain, and a user may instead have access to a time-series of cloudy images. An adaptation of VPint2 to derive weights from the cloud-free parts of the images in this time-series may be valuable. Additionally, while our oracle-based experiments showed the promise of ensembling-based approaches, our work did not include a practical implementation of such an approach, which may improve performance beyond that of VPint2 in its current form.

7.1.3. RQ3: PARAMETER ESTIMATION ILL-POSEDNESS

The third research question we will answer is RQ3: *What makes parameter estimation an ill-posed problem, and which factors affect the reliability of parameter estimation results?* (Chapter 5)

For this research question, we were interested in how Challenge 2, namely noise and ill-posedness, affected the reliability of the solutions of parameter estimation. In Chapter 5, we focussed on the inversion of the PROSAIL RTM, which we used as a prominent example of a parameter estimation method using EO data. The inversion of PROSAIL is generally known as an ill-posed problem, which is considered a downside of the method; however, we were interested in whether this ill-posedness was a property of the PROSAIL inversion specifically (where a single, best-fitting configuration must be found for an observed spectrum), or whether the parameter estimation problem itself (where the real-world parameter configuration must be estimated from noisy spectral observations) was the source of ill-posedness. We were unable to find any existing systematic analyses of the problem that could provide evidence for the ill-posedness of PROSAIL inversion specifically in cases where the parameter estimation problem itself was well-posed. This raised some doubts on whether the challenges encountered by users attempting PROSAIL inversion to perform parameter estimation, where a unique solution could often not be found, were truly caused by the ill-posedness of RTM (PROSAIL) inversion, or rather a general property of the parameter estimation problem.

In Chapter 5, we performed a systematic analysis, testing for the ill-posedness of RTM inversion (specifically the PROSAIL vegetation model), alternative poten-

tial causes for effects commonly ascribed to RTM inversion ill-posedness, and how adding prior knowledge can alleviate the problems. Our empirical results indicated that, unlike what is often assumed, the RTM inversion met all the requirements of a well-posed problem. This suggests that the unreliable results obtained by users were caused not by limitations of the RTM inversion approach, but rather by inherent properties of the parameter estimation problem. We hypothesised two possible properties that may cause ill-posedness on the parameter estimation problem as a whole: noise on the spectral observations and spectral mixing. We found that both factors resulted in ill-posedness on the parameter estimation problem, despite the RTM inversion procedure correctly identifying the correct solution to the (flawed) input data it received. Finally, we found that, under these conditions, adding prior knowledge to constrain the search space is an effective method for reducing the ill-posedness of the problem, even if the ill-posedness is a part of the parameter estimation problem rather than the RTM inversion procedure.

These findings indicate that the parameter estimation problem will still be ill-posed, even if other prediction methods (e.g., hybrid models or fully data-driven models) are used. However, because such methods will always provide an answer for the input, and because statistical biases in the model may partially compensate for the ill-posedness, this ill-posedness will be more difficult to diagnose compared to RTM inversion based on numerical optimisation. Our findings point to data-centric approaches, aimed at reducing spectral noise or adding additional data sources, as a promising direction for improving parameter estimation performance and reliability. *In conclusion, for RQ3, parameter estimation is likely an ill-posed problem due to flawed input data, with factors such as data noise and spectral mixing affecting the reliability of parameter estimation results.*

Limitations: With the exception of our experiments on the loss landscape properties of PROSAIL inversion, our analyses were carried out primarily on simulated data. This was a deliberate choice, because it enabled a highly accurate evaluation approach that would not have been possible with flawed ground truth data from real-world settings (see Section 2.1.3 for why such data is not suitable for accurate evaluation). In particular, given the subtlety of the differences in the loss function landscape that results in ill-posedness for small amounts of observation noise, even small inaccuracies in the ground truth data would have made these analyses challenging to carry out. However, without a full empirical study on real-world data, we cannot know with certainty that data-driven approaches trained fully on real-world data would be affected by the same factors as RTM inversion on simulated data, despite our results strongly pointing to such an effect.

7.1.4. RQ4: POSSIBLE SOLUTION SET

The final research question we will answer is RQ4: *How can we automatically extract the set of possible solutions to a noisy inference problem?* (Chapter 6)

For this research question we were interested in the severity of the ill-posedness caused by Challenge 2, through the factors identified in RQ3. We generalised the findings from parameter estimation to the general problem class of noisy inference and model inversion problems, of which parameter estimation is an example. Although we have established that noise on the observations (EO data) can cause ill-posedness for an inference (parameter estimation) task, it is important to be able to establish the severity of this ill-posedness for specific problem instances. If the severity is low, the inference results can still be reliable, while high severity greatly reduces the reliability of any solution to an inference problem.

In Chapter 6, we proposed the concept of ϵ-manifolds. These manifolds contain all the potential solutions to an inference problem that could reasonably explain the observations. When the observations are noisy, an incorrect solution will have a lower loss function value than the correct solution. Therefore, any solution that explains the observations, up to a factor ϵ, should be considered a viable solution to the inference problem. Here we made two key assumptions: loss-dependent shifts and loss-dependent origins, where the impact of observation noise is assumed to be governed by the loss function landscape. This focus on the loss function landscape differentiates ϵ-manifolds from statistical distributions, because the (prior) probability is not relevant when we aim to identify all the solutions that *could* explain the observations, not those that *did*. As a result, ϵ-manifolds enable novel types of analyses, including analysing the ill-posedness of parameter estimation problems.

We proposed a novel method, eMMI, to automatically approximate the ϵ-manifold for inference problems. This method was based on constrained diversity optimisation, leveraging different heuristics through its objectives and constraints to efficiently explore the space around the point prediction. Through our empirical experiments, we found that ϵ-manifolds offer substantial advantages over statistical interpretations of the problem, and that eMMI approximated these ϵ-manifolds better than the confidence intervals of baseline methods such as Gaussian processes, Bayesian neural networks and approximate Bayesian computation. *In conclusion, for RQ4, we can automatically extract the set of possible solutions to a noisy inference problem through the approximation of ϵ-manifolds by our proposed method, eMMI.*

Limitations: Although we introduced a theoretical framework for ϵ-manifolds that can be applied to multimodal problems (using ϵ-manifold sets) that may be highly ill-posed, non-convex or chaotic, we tested on unimodal problems, and the heuristics implemented for the current version of eMMI are based on (weak) as-

sumptions of local convexity. Extending the implementation of eMMI to approximate ϵ-manifolds in all possible scenarios, though possible within our proposed framework, was beyond the scope of the work. Instead, we focussed on a thorough analysis for scenarios that were relevant for our objective of physical parameter estimation. We also assumed a static ϵ for all instances of a particular simulator, while in reality, this assumption may not always hold, necessitating an extension incorporating dynamic settings for ϵ.

7.2. OUTLOOK AND FUTURE WORK

Throughout this dissertation we have introduced four concrete contributions, focusing on different parts of the pipeline shown in Figure 1.1, to improve parameter estimation. However, in all these components, there are remaining challenges to overcome, and further opportunities to explore. We start this section by describing two research directions we briefly explored, but ultimately did not pursue. We provide our experience in the hope that other researchers, if they are interested in this topic, will be aware of the hurdles to these approaches, and the conditions necessary for them to become feasible in the future. After this, we will provide our concrete recommendations for the next steps of research that could build on the work contained in this dissertation.

7.2.1. ALTERNATIVE RESEARCH DIRECTIONS

During the research work contained in this dissertation, there were two research directions whose merits we partially explored, but which we ultimately did not pursue. In this section, we will briefly explain our motivation for exploring them, the barriers to a successful execution that we encountered, and the necessary conditions that, in our view, could render them feasible in the future.

AUTOMATED INSTANCE GENERATION FOR SPECIALIST INVERTED SIMULATORS

When examining the most successful examples of model inversion-based parameter estimation, a common theme is that the scope of the study area is narrow and very clearly defined. For example, the model may be used for specific crops, such as maize [55, 52], wheat [46, 54] or rice [53], or specific locations [54, 45]. This enables the use of strict range constraint priors when performing model inversion, as described in Chapter 5, or the biased training of specialised machine learning models whose training data only includes solutions relevant to the study area. Our objective was to perform parameter estimation in general problem settings: the method should be applicable to any location and any species, or indeed, any type of physical simulation model.

We considered large, generalist models to be unlikely to be effective, because

of the ill-posedness of parameter estimation. Therefore, we were interested in borrowing concepts from the AutoML community in benchmark instance generation [293] to automatically generate specialised LUTs for local areas. We intended to match the distribution of the simulated spectra in the LUT to the distribution of the observed spectra for the local study area. The principle behind this approach was that we aimed to avoid a situation where a model would need to predict one value for one context, and predict another value for the same input data in a different context (e.g., a different geographical area or season). The main obstacle to this approach was obtaining ground truth data for the simulated data. Although we could perform post-hoc selection of instances using the simulated spectra, these matching spectra could still have been generated from very different sets of parameters. Therefore, while we maintain that the automated training of specialist models may be an effective method for global parameter estimation with minimal assumptions, we concluded that the implementation of such a method would be infeasible until the ill-posedness of model inversion approaches was addressed. Although we explored this ill-posedness in Chapters 5 and 6 by identifying its sources and enabling analyses, further work is still needed to truly overcome ill-posedness, enabling a resumption of this line of research.

PHYSICAL CONSTRAINTS AND EQUATION DISCOVERY

In the real world, physical parameters are often not independent, as these variables affect and are affected by a large, causal system consisting of many physical parameters and their relationships [294]. In simulation models, such as the PROSAIL model we have covered extensively in this dissertation, such relationships are usually not taken into account. For example, in PROSAIL, the physical relationships modelled by the simulator relate to the behaviour of light when encountering various media described by the physical input parameters of the model, such as water- and chlorophyll content. These input parameters can be independently manipulated by the user, and the model will generally perform a valid simulation using these values, regardless of how physically plausible their combination is. In reality, many of these input parameters could affect one another (e.g., a high value for one parameter is only possible if another parameter also has a high value, or they are mutually exclusive).

This led to the idea that capturing such physical dependencies may be a promising approach for imposing constraints on the search space of the inversion of physical models, such as PROSAIL. Imposing such constraints may have resulted in a similar reduction in ill-posedness as could be seen for range constraints in Chapter 5. Although this approach of constraining the search space using codified domain knowledge is conceptually appealing, to our knowledge, so far the relationships between such parameters have not been studied and formalised to an extent that

would be sufficient to construct, e.g., constraints for an optimisation algorithm.

To overcome this problem, we considered using equation discovery algorithms, such as ProGED [295], to automatically extract such relationships from data. However, we found the available data to be insufficient to make this approach feasible. Although isolated, small datasets exist, such as the ANGERS database [296], the available data is currently too limited in scope to reliably extract general constraints applicable to a global scale. Other, larger datasets may themselves consist of estimates that may introduce inaccuracies into the equation discovery procedure (e.g., NEON data [297]), or may not contain measurements of all the relevant parameters. However, should the availability of data improve, or should studies by domain experts result in formalised relationships between input parameters, a constraint-based approach to alleviate ill-posedness may become more viable.

7.2.2. FUTURE WORK

We believe that there is great promise in extending our work in the directions described in the remainder of this chapter. Our recommendations, like our contributions in Chapters 3–6, will consider the parameter estimation problem from the perspectives of the different components of the pipeline shown in Figure 1.1. Together, these directions can be combined to further improve inference performance and reliability, and learn even more about the parameter estimation problem. We will conclude this chapter, and this dissertation, with a hypothesised pipeline of what such a combined setup could look like in the future.

NEURAL VPINT

In Chapter 3, we introduced VPint as an independent spatial interpolation method that can fill in missing data in a target dataset. In the WP-MRP method, the spatial weights of the dataset were computed by a machine learning model based on a feature dataset of covariates, and the system of grid cell values was iteratively updated using these weights. This approach, while effective, required user input in the form of suitable feature datasets and choices on how the spatial weights were computed, while the use cases were fairly specialised (spatial settings where a gridded representation of point data is desired). This limits the applicability of VPint in parameter estimation settings where relevant covariates cannot be measured – for example, many of the PROSAIL parameters described in Chapter 5 are equally difficult to measure.

Moreover, applying VPint independently prior to any downstream deep learning model risks discarding the advantages (particularly with regard to keeping many parameters trainable) that deep learning has to offer. The VPint algorithm lends itself very well to an implementation as a neural network 'VPint block'. When incorporating VPint as a component of a neural network, many of the choices that

currently need manual input could be automatically learned from data, such as the computation of the spatial weights or the masking of missing data. Iterations could be represented by layers, where each cell is connected to its neighbours in the next layer, and the trainable weights of these connections (the spatial weights in VPint) are shared between all layers, boosting efficiency. The weights could be either predicted from covariates via neural network connections, or trained independently from a training dataset of the same area of interest. The network could also, in previous processing steps, learn to identify and mask missing data automatically.

Integrating VPint into deep learning pipelines in this manner, where users only need to specify a 'VPint block' in their network to remove any possible missing data from their input data, could greatly improve its applicability to general image processing tasks, as well as EO data specifically. Moreover, the performance of deep learning models, which represent the state of the art for many problems, may be significantly improved through the inclusion of a dedicated, efficient block to explicitly remove missing data. The research question for this approach would be: *How can VPint be integrated into existing deep learning frameworks such that missing input data can be automatically identified and interpolated in an effective and efficient manner?*

TIME-SERIES VPINT2

When using VPint2 to fill in missing satellite data in particular, as we did in Chapter 4, our proposed method was designed to interpolate missing data (especially clouds) from a single image, and used a single, fully observable reference image to guide the reconstruction. While this setup can be convenient in setups where only a single target image is of interest, other applications may be presented with a time-series of images with potentially missing data, and require the missing data in the full time series to be filled in. In existing work, time-series data already forms an appealing source of information complementarity, exploited by multiple deep learning-based cloud removal methods [41, 172, 174]. It seems probable that VPint2 could likewise be adapted to this problem setting, particularly when combining this approach for VPint2 with the neural network version of the original VPint described above. This would further reduce the data requirements of the method, and potentially enable bulk processing to generate full datasets without missing data.

In a basic version of the algorithm, the spatial weights used by VPint2 could be computed from the observable parts of the images in the time-series, in a manner similar to that of the current VPint2 algorithm. When multiple images have cloud-free data available for the same pairs of pixels, algorithm stability could be improved by computing, e.g., the median of the weights, or a recency bias could

be added. If none of the images contain cloud-free data for certain parts of the image, the algorithm could default to a weight of 1 (identity). The neural network version of the algorithm could entail substantial advantages by enabling complex, non-linear relationships between the flawed input time-series data and the spatial weights, enabling the computation of spatial weights using abstract features from a rich latent space. This approach would also enable the model to learn cloud masks and other data imperfections, reducing the need for explicitly defined 'missing data'. This adaptation of VPint2 may be another valuable contribution to improve the reliability of EO data. The research question for this method would be: *How can missing pixels in a time-series of satellite data be effectively and efficiently interpolated when a fully observable reference image is not available?*

SPECTRAL DENOISING

Moving beyond missing data, much of our focus in Chapters 5 and 6 has been on the ill-posedness caused by observation noise in parameter estimation and other inference problems. When this noise, such as atmospheric interference or spectral mixing, is present in the data, our results in Chapter 5 showed that parameter estimation problems become ill-posed, while our proposed methods in Chapter 6 enabled estimations of the severity of the ill-posedness caused by this noise. Although it is certainly useful to diagnose the problems, these analyses point to spectral denoising as a potentially valuable approach to further improve estimation performance. If this denoising task could be performed perfectly, such that the denoised data exactly matches noise-free simulated data, performing parameter estimation would become a regular, well-posed problem, as we found the inversion of the PROSAIL RTM to be in Chapter 5.

In recent years, there have been considerable advances in deep learning techniques, such as denoising autoencoders [211, 298], and data fusion approaches [40, 41, 81]. For EO data in particular, the spatial and temporal autocorrelation of images may enable such deep learning methods to automatically reduce the level of noise on the spectral data. It is also possible that neural VPint or VPint2 blocks could be used to perform this denoising, by automatically masking out unreliable pixel values, or interpolating data with random data masks to perform a type of implicit regularisation. Performing denoising on EO data to reduce the ill-posedness of parameter estimation would result in the following research question: *How could we effectively remove noise from EO data, such that the denoised data closely resembles the noise-free data as simulated by RTMs?*

LATENT SPACE EMMI

Despite the promise of denoising and further improvements to the VPint algorithm, some inaccuracies will be unavoidable, and the inversion some physical

models (e.g., other RTMs with different properties from PROSAIL) may still be ill-posed even if the data is noise-free. In this case, we still need ϵ-manifolds and eMMI to approximate viable solution sets. The results in Chapter 6 showed that ϵ-manifolds and eMMI are highly effective at approximating the viable solution set of various inference problems, notably including machine learning. By constructing ϵ-manifolds for specific instances, users could, for example, find sets of adversarial examples for machine learning models, analyse the ill-posedness of inference problems, or compare model parameterisations.

For many machine learning EO applications (especially deep learning, which represents a large part of the state-of-the-art methods), there are concerns about generalisability and extrapolation, because the complex Earth system is highly diverse (we explained this in Section 2.1.3). The robustness analyses enabled by eMMI would clearly be beneficial to diagnose generalisably problems. However, most current state-of-the-art machine learning- and deep learning models are far more complex and high-dimensional than even the 100-dimensional linear regression models we considered in our experiments in Chapter 6, both in terms of the number of trainable parameters (for model parameterisation problems) and input parameters (for adversarial examples). This means that the current version of eMMI is ill-suited to approximating ϵ-manifolds for state-of-the-art models we are interested in for EO data and data-driven parameter estimation models.

In order to overcome this obstacle and make eMMI directly applicable not just to approximating ϵ-manifolds of RTM inversion problems, but also the analysis of deep learning model parameterisations or their robustness to certain types of input, we would need to greatly improve the applicability of eMMI to high-dimensional parameter spaces. To achieve this, it may be possible to approximate ϵ-manifolds by optimising in a latent space [299, 300], compressing the information contained in the high-dimensional problems associated with large models, thereby enabling ϵ-manifold-based analyses in more practical machine learning- and deep learning settings. Since the loss landscapes associated with such latent spaces would likely be much less favourable than those considered in Chapter 6, this approach may require some extensions suggested in the future work of Section 6.8, such as a surrogate model-based version of the method, and better support for multi-modal, highly non-convex landscapes. The research question associated with this direction would be: *How can we approximate ϵ-manifolds in nonlinear latent spaces for high-dimensional inference problems?*

PHYSICS-INFORMED REPRESENTATION LEARNING

Finally, we recommend approaching parameter estimation from EO data from another angle altogether. Although we have mainly focused on addressing the challenges of parameter estimation through AI techniques, there may be great promise

in doing the opposite: addressing AI challenges through parameter estimation, and notably, RTM inversion. In parameter estimation, we often use a form of hybrid models: machine learning-based models whose training was supervised using a training dataset generated by an RTM (usually PROSAIL). However, there are other types of hybrid models, most notably including physics-informed neural networks (PINNs) [33, 36]. These models can take multiple different forms; the form most relevant to us consists of a standard neural network encoder, whose latent representation forms the inputs of a physical model based on differential equations. The loss of the final output prediction by the physical model can be propagated back, because its differential equations lend themselves well to back-propagation.

This type of setup could be replicated by an RTM, where a neural network takes EO data as input, and encodes this to a latent representation that forms the input for an RTM. The RTM can then be used to simulate EO data based on these inputs, which should match the input spectra provided to the neural network. This type of physics-informed neural network has recently been successfully applied to PROSAIL inversion [301], showing the promise of this research direction. The neural network could learn a mapping from EO data to RTM input parameters, without needing any ground truth dataset containing these parameters. If desired, given that this approach would likely still suffer from the same ill-posedness as all other parameter estimation methods, the model could also learn to predict a diverse set of solutions in the ϵ-manifold, obtained by eMMI, and possibly process these solutions further into a latent representation, to summarise the ϵ-manifold. This approach would address several key limitations of conventional, fully data-driven approaches (see Section 2.1.3), by eliminating the need for accurate ground truth measurements and sufficient support (representation in the training dataset) for all possible viable solutions.

In addition to direct parameter estimation or the amortised estimation of the associated ϵ-manifolds, existing representation learning techniques (e.g., for foundation models) could be supplemented by a this latent representation of the ϵ-manifold for PROSAIL inversion, thereby guaranteeing the inclusion of a set of informative features highly relevant to known physical processes related to, for example, vegetation. This would lead to the following research question: *How can we train a model to learn a representation of physically relevant features for EO data, in a fully unsupervised manner?*

IDEALISED FUTURE PIPELINE

Based on the recommendations above, an idealised pipeline (visualised in Figure 7.1) may take the following shape. The user receives a time-series of satellite images as input, with a goal of returning an estimated distribution of parameter val-

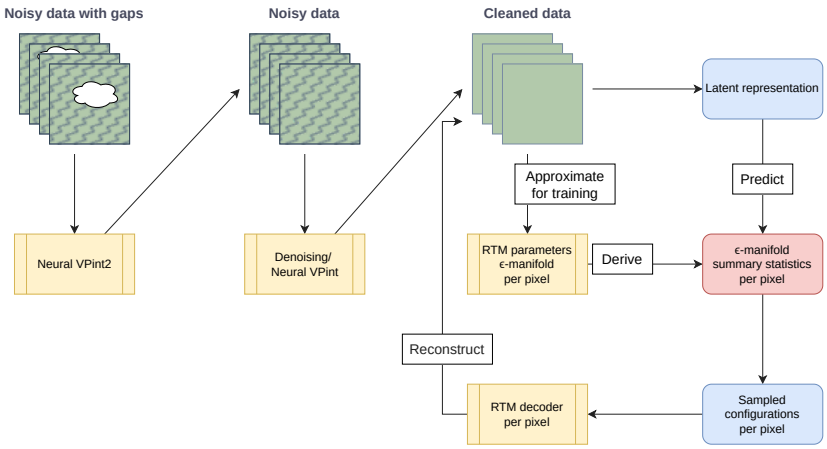


Figure 7.1: Idealised future parameter estimation pipeline.

ues at every pixel. We use a deep neural network to enable efficient large-scale processing and take advantage of their state-of-the-art performance, particularly for problems with large volumes of data available (e.g., EO data), and train this network as follows.

i) All missing data, such as clouds or faulty pixels, are first processed using a time-series neural VPint2 block. ii) The user can choose to use the previous VPint2 block to simultaneously perform denoising, or use a separate neural VPint block for this purpose. iii) The model next maps the time-series data to a latent representation using standard deep learning layers. iv) For all valid pixels in the denoised input, we use eMMI (possibly its latent version) to estimate the ϵ-manifold for every pixel in the data (this will be computationally expensive at training time). v) We estimate summary statistics for the ϵ-manifold from the latent representation of iii) in a manner similar to amortised Bayesian inference (this will be a lossy conversion, because ϵ-manifolds are not themselves a statistical distribution). vi) Based on the estimated statistical distribution, we sample from this distribution to pass inputs to an RTM, which reconstructs pixel values. vii) We train the network on the either a regression loss (e.g., mean squared error) for the pixel values throughout the entire time-series, or a pixel-wise classification (segmentation) loss measuring whether the spectra of reconstructed pixels are within a value of ϵ of the original spectra.

Of course, the above pipeline is highly idealised, and assumes that every step

of this procedure functions as intended. Many of these steps could likely become new research projects on their own. However, if successful, the training pipeline described above would result in a trained model that a) can perform large-scale predictions for parameter estimation, b) automatically fills in data gaps and denoises the data, c) requires no in-situ ground truth data to train, and d) incorporates ill-posedness (including for unobserved configurations, which would be a major limitation of methods not using eMMI, given the generalisation concerns described above) in the uncertainty of its predictions.

Although this idealised pipeline is still many years of intensive research work removed from being realised, we consider the recommendations we described in this chapter to be the most promising concrete next steps to take to bring it closer to fruition. It is our hope that, one day, we could see a parameter estimation approach similar to what we described above be implemented and deployed in practice.

7.3. CONCLUDING REMARKS

Physical parameters describing the current state of the Earth, such as ecological parameters describing vegetation [191], atmospheric parameters describing gases and particles in the air [1, 2] and marine parameters describing our oceans [3], are of tremendous importance to scientific research and informed decision making in, for example, environmental policy and public health campaigns. We would like to estimate these parameters indirectly from the abundant Earth observation data collected by satellite platforms, thereby creating regular, global maps of important environmental and scientific parameters. Unfortunately, performing this estimation is greatly hampered by data gaps, noise and ill-posedness, resulting in unreliable and inaccurate estimations.

In this dissertation, we have proposed multiple methods for increasing data reliability and diagnosing ill-posedness. We have also identified observation noise as the likely primary cause of ill-posedness for parameter estimation problems, leading to concrete recommendations for future work to focus on improving the data quality and information content in the observed data. We believe that these findings are highly relevant to domain practitioners, who depend on accurate parameter estimation results to perform their research, but may misattribute inaccurate results to the methods being deployed, rather than a fundamental characteristic of the problem. While particularly severe in parameter estimation, data gaps, noise and ill-posedness are not unique to this problem. We invite any researchers from different fields, if they are interested in the solutions presented in this work, to apply them to their own problem settings.

Although, befitting scientific research work, we wrap up this dissertation with

more questions than we had before we started, it is our hope that the findings and methods contained in this dissertation contribute to the collective scientific efforts across disciplines to better understand, monitor and maintain the state of the Earth.

BIBLIOGRAPHY

[1] OP Hasekamp and J Landgraf. "A linearized vector radiative transfer model for atmospheric trace gas retrieval". In: *Journal of Quantitative Spectroscopy and Radiative Transfer* 75.2 (2002), pp. 221–238 (cit. on pp. 1, 19, 183).

[2] D Schepers, JMJ Aan de Brugh, Ph Hahne, A Butz, OP Hasekamp, and J Landgraf. "LINTRAN v2. 0: A linearised vector radiative transfer model for efficient simulation of satellite-born nadir-viewing reflection measurements of cloudy atmospheres". In: *Journal of Quantitative Spectroscopy and Radiative Transfer* 149 (2014), pp. 347–359 (cit. on pp. 1, 19, 183).

[3] Lise Kilic, Catherine Prigent, Jacqueline Boutin, Thomas Meissner, Stephen English, and Simon Yueh. "Comparisons of ocean radiative transfer models with SMAP and AMSR2 observations". In: *Journal of Geophysical Research: Oceans* 124.11 (2019), pp. 7683–7699 (cit. on pp. 1, 19, 183).

[4] Stéphane Jacquemoud, Wout Verhoef, Frédéric Baret, Cédric Bacour, Pablo J. Zarco-Tejada, Gregory P. Asner, Christophe François, and Susan L. Ustin. "PROSPECT+SAIL models: A review of use for vegetation characterization". In: *Remote Sensing of Environment* 113 (2009), S56–S66 (cit. on pp. 1, 19).

[5] Siddharth Mishra-Sharma. "Inferring dark matter substructure with astrometric lensing beyond the power spectrum". In: *Machine Learning: Science and Technology* 3.1 (2022), 01LT03 (cit. on pp. 3, 126, 129).

[6] Xiaosheng Zhao, Yi Mao, Cheng Cheng, and Benjamin D Wandelt. "Simulation-based inference of reionization parameters from 3D tomographic 21 cm light-cone images". In: *The Astrophysical Journal* 926.2 (2022), p. 151 (cit. on pp. 3, 126, 129).

[7] Jacques-Donald Tournier, Susumu Mori, and Alexander Leemans. "Diffusion tensor imaging and beyond". In: *Magnetic resonance in medicine* 65.6 (2011), p. 1532 (cit. on p. 3).

[8] Jeff Kershaw, Babak A Ardekani, and Iwao Kanno. "Application of Bayesian inference to fMRI data analysis". In: *IEEE Transactions on Medical Imaging* 18.12 (1999), pp. 1138–1153 (cit. on p. 3).

[9] Jana Lipková, Panagiotis Angelikopoulos, Stephen Wu, Esther Alberts, Benedikt Wiestler, Christian Diehl, Christine Preibisch, Thomas Pyka, Stephanie E Combs, Panagiotis Hadjidoukas, et al. "Personalized radiotherapy design for glioblastoma: integrating mathematical tumor models, multimodal scans, and Bayesian inference". In: *IEEE Transactions on Medical Imaging* 38.8 (2019), pp. 1875–1884 (cit. on p. 3).

[10] Jin Chen, Xiaolin Zhu, James E Vogelmann, Feng Gao, and Suming Jin. "A simple and effective method for filling gaps in Landsat ETM+ SLC-off images". In: *Remote Sensing of Environment* 115.4 (2011), pp. 1053–1064 (cit. on pp. 4, 64, 79).

[11] F L Naus, D van Gils, and T J Brussée. *Status and trend of the shallow and medium-deep groundwater quality in the Netherlands as measured by the National Groundwater Quality Monitoring Network*. 2023 (cit. on p. 12).

[12] Korean Ministry of Environment. *AIRKOREA*, https://www.airkorea.or.kr. 2005 (cit. on p. 12).

[13] National Research Institute for Earth Science and Disaster Resilience (NIED). *Kyoshin Network (K-NET)*. 2008. URL: https://www.kyoshin.bosai.go.jp/ (cit. on p. 12).

[14] National Ecological Observatory Network (NEON). *NEON Data Portal*. 2012. URL: https://data.neonscience.org/ (cit. on p. 12).

[15] B Fraters, A E J Hooijboer, G B J Rijs, N van Duijnhoven, and J C Rozemeijer. *Waterkwaliteit in Nederland; toestand (2012–2015) en trend (1992–2015)*. 2016 (cit. on p. 12).

[16] European Commission. *Copernicus Sentinel Data Access Annual Report 2023*. Tech. rep. European Commission, 2024 (cit. on p. 14).

[17] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. "A comprehensive survey on transfer learning". In: *Proceedings of the IEEE* 109.1 (2020), pp. 43–76 (cit. on p. 15).

[18] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828 (cit. on p. 15).

[19] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. "Contrastive representation learning: A framework and review". In: *IEEE Access* 8 (2020), pp. 193907–193934 (cit. on p. 15).

[20] Jesper E Van Engelen and Holger H Hoos. "A survey on semi-supervised learning". In: *Machine Learning* 109.2 (2020), pp. 373–440 (cit. on p. 15).

[21] Longlong Jing and Yingli Tian. “Self-supervised visual feature learning with deep neural networks: A survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.11 (2020), pp. 4037–4058 (cit. on p. 15).

[22] Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David B. Lobell, and Stefano Ermon. “SatMAE: Pre-training Transformers for Temporal and Multi-Spectral Satellite Imagery”. In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho. 2022. URL: https://openreview.net/forum?id=WBhqzpF6KYH (cit. on p. 15).

[23] Johannes Jakubik, Sujit Roy, C. E. Phillips, Paolo Fraccaro, Denys Godwin, Bianca Zadrozny, Daniela Szwarcman, Carlos Gomes, Gabby Nyirjesy, Blair Edwards, Daiki Kimura, Naomi Simumba, Linsong Chu, S. Karthik Mukkavilli, Devyani Lambhate, Kamal Das, Ranjini Bangalore, Dario Oliveira, Michal Muszynski, Kumar Ankur, Muthukumaran Ramasubramanian, Iksha Gurung, Sam Khallaghi, Hanxi (Steve) Li, Michael Cecil, Maryam Ahmadi, Fatemeh Kordi, Hamed Alemohammad, Manil Maskey, Raghu Ganti, Kommy Weldemariam, and Rahul Ramachandran. “Foundation Models for Generalist Geospatial Artificial Intelligence”. In: *Preprint Available on arxiv:2310.18660* (Oct. 2023) (cit. on p. 15).

[24] Casper Fibaek, Luke Camilleri, Andreas Luyts, Nikolaos Dionelis, and Bertrand Le Saux. *PhilEO Bench: Evaluating Geo-Spatial Foundation Models*. 2024. arXiv: 2401.04464 [cs.CV] (cit. on p. 15).

[25] Gencer Sumbul, Marcela Charfuelan, Begüm Demir, and Volker Markl. “BigEarthNet: A large-scale benchmark archive for remote sensing image understanding”. In: *IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2019, pp. 5901–5904 (cit. on p. 15).

[26] Ava Vali, Sara Comai, and Matteo Matteucci. “Deep learning for land use and land cover classification based on hyperspectral and multispectral earth observation data: A review”. In: *Remote Sensing* 12.15 (2020), p. 2495 (cit. on p. 15).

[27] Thorsten Hoeser and Claudia Kuenzer. “Object detection and image segmentation with deep learning on earth observation data: A review-part i: Evolution and recent trends”. In: *Remote Sensing* 12.10 (2020), p. 1667 (cit. on p. 15).

[28] Alistair Francis, Panagiotis Sidiropoulos, and Jan-Peter Muller. “CloudFCN: Accurate and robust cloud detection for satellite imagery with deep learning”. In: *Remote Sensing* 11.19 (2019), p. 2312 (cit. on p. 15).

[29] Alistair Francis. “Sensor independent cloud and shadow masking with partial labels and multimodal inputs”. In: *IEEE Transactions on Geoscience and Remote Sensing* (2024) (cit. on pp. 15, 64, 77, 81).

[30] Alexander A Gilerson, Anatoly A Gitelson, Jing Zhou, Daniela Gurlin, Wesley Moses, Ioannis Ioannou, and Samir A Ahmed. “Algorithms for remote estimation of chlorophyll-a in coastal and inland waters using red and near infrared bands”. In: *Optics Express* 18.23 (2010), pp. 24109–24125 (cit. on p. 15).

[31] Jianfeng Zhang, Wenting Han, Lvwen Huang, Zhiyong Zhang, Yimian Ma, and Yamin Hu. “Leaf chlorophyll content estimation of winter wheat based on visible and near-infrared sensors”. In: *Sensors* 16.4 (2016), p. 437 (cit. on p. 15).

[32] O Torres, PK Bhartia, JR Herman, Z Ahmad, and J Gleason. “Derivation of aerosol properties from satellite measurements of backscattered ultraviolet radiation: Theoretical basis”. In: *Journal of Geophysical Research: Atmospheres* 103.D14 (1998), pp. 17099–17110 (cit. on p. 15).

[33] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational Physics* 378 (2019), pp. 686–707 (cit. on pp. 17, 181).

[34] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. “Scientific machine learning through physics–informed neural networks: Where we are and what’s next”. In: *Journal of Scientific Computing* 92.3 (2022), p. 88 (cit. on p. 17).

[35] Jochem Verrelst, Zbyněk Malenovskỳ, Christiaan Van der Tol, Gustau Camps-Valls, Jean-Philippe Gastellu-Etchegorry, Philip Lewis, Peter North, and Jose Moreno. “Quantifying vegetation biophysical variables from imaging spectroscopy data: A review on retrieval methods”. In: *Surveys in Geophysics* 40 (2019), pp. 589–629 (cit. on pp. 17, 19–20, 98, 108, 144).

[36] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and F Prabhat. “Deep learning and process understanding for data-driven Earth system science”. In: *Nature* 566.7743 (2019), pp. 195–204 (cit. on pp. 17, 181).

[37] Alex Krizhevsky, Geoffrey Hinton, et al. *Learning multiple layers of features from tiny images*. 2009 (cit. on p. 17).

[38] Rosa Maria Cavalli. “Spatial validation of spectral unmixing results: A systematic review”. In: *Remote Sensing* 15.11 (2023), p. 2822 (cit. on pp. 18, 108, 110).

[39] Michael D King, Steven Platnick, W Paul Menzel, Steven A Ackerman, and Paul A Hubanks. “Spatial and temporal distribution of clouds observed by MODIS onboard the Terra and Aqua satellites”. In: *IEEE Transactions on Geoscience and Remote Sensing* 51.7 (2013), pp. 3826–3852 (cit. on p. 18).

[40] Andrea Meraner, Patrick Ebel, Xiao Xiang Zhu, and Michael Schmitt. “Cloud removal in Sentinel-2 imagery using a deep residual neural network and SAR-optical data fusion”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 166 (2020), pp. 333–346 (cit. on pp. 18, 24, 63, 75–76, 79–80, 87, 89, 179).

[41] Patrick Ebel, Vivien Sainte Fare Garnot, Michael Schmitt, Jan Dirk Wegner, and Xiao Xiang Zhu. “UnCRtainTS: Uncertainty Quantification for Cloud Removal in Optical Satellite Time Series”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2085–2095 (cit. on pp. 18, 24, 64, 77, 79–80, 84, 88, 178–179).

[42] Huanfeng Shen, Xinghua Li, Qing Cheng, Chao Zeng, Gang Yang, Huifang Li, and Liangpei Zhang. “Missing information reconstruction of remote sensing data: A technical review”. In: *IEEE Geoscience and Remote Sensing Magazine* 3.3 (2015), pp. 61–85 (cit. on pp. 18, 63, 91).

[43] Stéphane Jacquemoud and Frédéric Baret. “PROSPECT: A model of leaf optical properties spectra”. In: *Remote Sensing of Environment* 34.2 (1990), pp. 75–91 (cit. on pp. 19, 98, 156).

[44] Wouter Verhoef. “Light scattering by leaf layers with application to canopy reflectance modeling: The SAIL model”. In: *Remote Sensing of Environment* 16.2 (1984), pp. 125–141 (cit. on pp. 19, 98, 156).

[45] Nguyen An Binh, Leon T Hauser, Pham Viet Hoa, Giang Thi Phuong Thao, Nguyen Ngoc An, Huynh Song Nhut, Tran Anh Phuong, and Jochem Verrelst. “Quantifying mangrove leaf area index from Sentinel-2 imagery using hybrid models and active learning”. In: *International Journal of Remote Sensing* 43.15-16 (2022), pp. 5636–5657 (cit. on pp. 19–21, 99–100, 127, 130, 144, 175).

[46] Rabi N Sahoo, Shalini Gakhar, Rajan G Rejith, Jochem Verrelst, Rajeev Ranjan, Tarun Kondraju, Mahesh C Meena, Joydeep Mukherjee, Anchal Daas, Sudhir Kumar, et al. “Optimizing the Retrieval of Wheat Crop Traits from UAV-Borne Hyperspectral Image with Radiative Transfer Modelling Using

Gaussian Process Regression". In: *Remote Sensing* 15.23 (2023), p. 5496 (cit. on pp. 19–20, 100, 122, 127, 130, 160, 175).

[47] S Jacquemoud, S Flasse, J Verdebout, and G Schmuck. "Comparison of several optimization methods to extract canopy biophysical parameters-application to CAESAR data". In: *International Symposium on Physical Measurements and SignaturesSignatures in Remote Sensing*. CNES Paris. 1994, pp. 291–298 (cit. on pp. 20, 99, 126, 130).

[48] K Richter, C Atzberger, F Vuolo, P Weihs, and G d'Urso. "Experimental assessment of the Sentinel-2 band setting for RTM-based LAI retrieval of sugar beet and maize". In: *Canadian Journal of Remote Sensing* 35.3 (2009), pp. 230–247 (cit. on pp. 20, 99, 126, 130).

[49] S Jacquemoud, C Bacour, H Poilve, and J-P Frangi. "Comparison of four radiative transfer models to simulate plant canopies reflectance: Direct and inverse mode". In: *Remote Sensing of Environment* 74.3 (2000), pp. 471–481 (cit. on pp. 20, 99, 130).

[50] Nuno Cesar de Sa, Mitra Baratchi, Leon T Hauser, and Peter van Bodegom. "Exploring the impact of noise on hybrid inversion of PROSAIL RTM on Sentinel-2 data". In: *Remote Sensing* 13.4 (2021), p. 648 (cit. on pp. 20, 99, 104, 109, 115, 117, 126, 130, 217).

[51] Katja Berger, Juan Pablo Rivera Caicedo, Luca Martino, Matthias Wocher, Tobias Hank, and Jochem Verrelst. "A survey of active learning for quantifying vegetation traits from terrestrial Earth observation data". In: *Remote Sensing* 13.2 (2021), p. 287 (cit. on pp. 20–21, 99, 130).

[52] Gabriele Candiani, Giulia Tagliabue, Cinzia Panigada, Jochem Verrelst, Valentina Picchi, Juan Pablo Rivera Caicedo, and Mirco Boschetti. "Evaluation of hybrid models to estimate chlorophyll and nitrogen content of maize crops in the framework of the future CHIME mission". In: *Remote Sensing* 14.8 (2022), p. 1792 (cit. on pp. 20, 100, 122, 130, 175).

[53] Marta Rossi, Gabriele Candiani, Francesco Nutini, Marco Gianinetto, and Mirco Boschetti. "Sentinel-2 estimation of CNC and LAI in rice cropping system through hybrid approach modelling". In: *European Journal of Remote Sensing* (2022), pp. 1–20 (cit. on pp. 20, 99, 130, 175).

[54] Gabriel Caballero, Alejandro Pezzola, Cristina Winschel, Alejandra Casella, Paolo Sanchez Angonova, Juan Pablo Rivera-Caicedo, Katja Berger, Jochem Verrelst, and Jesus Delegido. "Seasonal Mapping of Irrigated Winter Wheat Traits in Argentina with a Hybrid Retrieval Workflow Using Sentinel-2 Imagery". In: *Remote Sensing* 14.18 (2022), p. 4531 (cit. on pp. 20, 99, 127, 130, 160, 175).

[55] Marina Ranghetti, Mirco Boschetti, Luigi Ranghetti, Giulia Tagliabue, Cinzia Panigada, Marco Gianinetto, Jochem Verrelst, and Gabriele Candiani. "Assessment of maize nitrogen uptake from PRISMA hyperspectral data through hybrid modelling". In: *European Journal of Remote Sensing* 56.1 (2023), p. 2117650 (cit. on pp. 20, 99, 130, 144, 175).

[56] Victor Neuteboom. *AutoML for creating hybrid Earth science models*. LIACS, Leiden University, 2022 (cit. on p. 20).

[57] B Combal, Frédéric Baret, M Weiss, Alain Trubuil, D Macé, A Pragnere, R Myneni, Y Knyazikhin, and L Wang. "Retrieval of canopy biophysical variables from bidirectional reflectance: Using prior information to solve the ill-posed inverse problem". In: *Remote Sensing of Environment* 84.1 (2003), pp. 1–15 (cit. on pp. 21, 98, 100, 110, 126).

[58] Jia Sun, Lunche Wang, Shuo Shi, Zhenhai Li, Jian Yang, Wei Gong, Shaoqiang Wang, and Torbern Tagesson. "Leaf pigment retrieval using the PROSAIL model: Influence of uncertainty in prior canopy-structure information". In: *The Crop Journal* 10.5 (2022), pp. 1251–1263 (cit. on pp. 21, 98).

[59] Xingwen Quan, Binbin He, and Xing Li. "A Bayesian network-based method to alleviate the ill-posed inverse problem: A case study on leaf area index and canopy water content retrieval". In: *IEEE Transactions on Geoscience and Remote Sensing* 53.12 (2015), pp. 6507–6517 (cit. on pp. 21, 98).

[60] Sergei Igorevich Kabanikhin. "Definitions and examples of inverse and ill-posed problems". In: *Journal of Inverse and Ill-posed Problems* (2008) (cit. on p. 21).

[61] Hyun Jeung Ko and Gerald W Evans. "A genetic algorithm-based heuristic for the dynamic integrated forward/reverse logistics network for 3PLs". In: *Computers & Operations Research* 34.2 (2007), pp. 346–366 (cit. on p. 22).

[62] David Meignan, Sigrid Knust, Jean-Marc Frayret, Gilles Pesant, and Nicolas Gaud. "A review and taxonomy of interactive optimization methods in operations research". In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5.3 (2015), pp. 1–43 (cit. on p. 22).

[63] Roy de Winter, Jan Furustam, Thomas Bäck, and Thijs Muller. "Optimizing ships using the holistic accelerated concept design methodology". In: *Practical Design of Ships and Other Floating Structures*. Springer, 2019, pp. 38–50 (cit. on p. 22).

[64] Holger H Hoos and Thomas Stützle. "Stochastic local search". In: *Handbook of Approximation Algorithms and Metaheuristics*. Chapman and Hall/CRC, 2018, pp. 297–307 (cit. on pp. 23, 105–106, 218).

[65] Thomas Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996 (cit. on p. 23).

[66] James Kennedy and Russell Eberhart. "Particle swarm optimization". In: *International Conference on Neural Networks*. Vol. 4. ieee. 1995, pp. 1942–1948 (cit. on p. 23).

[67] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. "Ant colony optimization". In: *IEEE Computational Intelligence Magazine* 1.4 (2007), pp. 28–39 (cit. on p. 23).

[68] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. "Taking the human out of the loop: A review of Bayesian optimization". In: *Proceedings of the IEEE* 104.1 (2015), pp. 148–175 (cit. on p. 23).

[69] Jean-Bastien Grill, Michal Valko, and Rémi Munos. "Black-box optimization of noisy functions with unknown smoothness". In: *Advances in Neural Information Processing Systems* 28 (2015) (cit. on p. 23).

[70] Peter Flach. *Machine Learning: the Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press, 2012 (cit. on p. 23).

[71] Grant J Scott, Matthew R England, William A Starms, Richard A Marcum, and Curt H Davis. "Training deep convolutional neural networks for land–cover classification of high-resolution imagery". In: *IEEE Geoscience and Remote Sensing Letters* 14.4 (2017), pp. 549–553 (cit. on p. 24).

[72] Naftaly Wambugu, Yiping Chen, Zhenlong Xiao, Mingqiang Wei, Saifullahi Aminu Bello, José Marcato Junior, and Jonathan Li. "A hybrid deep convolutional neural network for accurate land cover classification". In: *International Journal of Applied Earth Observation and Geoinformation* 103 (2021), p. 102515 (cit. on p. 24).

[73] Shunping Ji, Chi Zhang, Anjian Xu, Yun Shi, and Yulin Duan. "3D convolutional neural networks for crop classification with multi-temporal remote sensing images". In: *Remote Sensing* 10.1 (2018), p. 75 (cit. on p. 24).

[74] Keiller Nogueira, Mauro Dalla Mura, Jocelyn Chanussot, William Robson Schwartz, and Jefersson Alex Dos Santos. "Dynamic multicontext segmentation of remote sensing images based on convolutional networks". In: *IEEE Transactions on Geoscience and Remote Sensing* 57.10 (2019), pp. 7503–7520 (cit. on p. 24).

[75] Julia Wąsala, Suzanne Marselis, Laurens Arp, Holger Hoos, Nicolas Longépé, and Mitra Baratchi. "AutoSR4EO: An AutoML Approach to Super-Resolution for Earth Observation Images". In: *Remote Sensing* 16.3 (2024), p. 443 (cit. on p. 24).

[76] Saman Ghaffarian, João Valente, Mariska Van Der Voort, and Bedir Tekinerdogan. "Effect of attention mechanism in deep learning-based remote sensing image processing: A systematic literature review". In: *Remote Sensing* 13.15 (2021), p. 2965 (cit. on p. 24).

[77] Zhi Li, Siying Cao, Jiakun Deng, Fengyi Wu, Ruilan Wang, Junhai Luo, and Zhenming Peng. "STADE-CDNet: Spatial–temporal attention with difference enhancement-based network for remote sensing image change detection". In: *IEEE Transactions on Geoscience and Remote Sensing* 62 (2024), pp. 1–17 (cit. on p. 24).

[78] Seyd Teymoor Seydi, Meisam Amani, and Arsalan Ghorbanian. "A dual attention convolutional neural network for crop classification using time-series Sentinel-2 imagery". In: *Remote Sensing* 14.3 (2022), p. 498 (cit. on p. 24).

[79] Xin Li, Feng Xu, Linyang Li, Nan Xu, Fan Liu, Chi Yuan, Ziqi Chen, and Xin Lyu. "AAFormer: Attention-attended transformer for semantic segmentation of remote sensing images". In: *IEEE Geoscience and Remote Sensing Letters* (2024) (cit. on p. 24).

[80] Pallavi Jain, Dino Ienco, Roberto Interdonato, Tristan Berchoux, and Diego Marcos. "SenCLIP: Enhancing zero-shot land-use mapping for Sentinel-2 with ground-level prompting". In: *IEEE/CVF Winter Conference on Applications of Computer Vision*. IEEE. 2025, pp. 5656–5665 (cit. on p. 24).

[81] Jiaxin Li, Danfeng Hong, Lianru Gao, Jing Yao, Ke Zheng, Bing Zhang, and Jocelyn Chanussot. "Deep learning in multimodal remote sensing data fusion: A comprehensive review". In: *International Journal of Applied Earth Observation and Geoinformation* 112 (2022), p. 102926 (cit. on pp. 24, 179).

[82] European Union's Copernicus Land Monitoring Service. *Leaf Area Index 2014-present (raster 300 m), global, 10-daily*. https://doi.org/10.2909/219fdc9f-616b-444b-a495-198f527b4722. 2017 (cit. on p. 24).

[83] Pedro Castro-Valdecantos, Orly Enrique Apolo-Apolo, Manuel Pérez-Ruiz, and G Egea. "Leaf area index estimations by deep learning models using RGB images and data fusion in maize". In: *Precision Agriculture* 23.6 (2022), pp. 1949–1966 (cit. on p. 24).

[84] Shuaibing Liu, Xiuliang Jin, Chenwei Nie, Siyu Wang, Xun Yu, Minghan Cheng, Mingchao Shao, Zixu Wang, Nuremanguli Tuohuti, Yi Bai, et al. "Estimating leaf area index using unmanned aerial vehicle data: shallow vs. deep machine learning algorithms". In: *Plant Physiology* 187.3 (2021), pp. 1551–1576 (cit. on p. 24).

[85] Luis Carrasco, Aneurin W O'Neil, R Daniel Morton, and Clare S Rowland. "Evaluating combinations of temporally aggregated Sentinel-1, Sentinel-2 and Landsat 8 for land cover mapping with Google Earth Engine". In: *Remote Sensing* 11.3 (2019), p. 288 (cit. on p. 28).

[86] Hongliang Fang, Frédéric Baret, Stephen Plummer, and Gabriela Schaepman-Strub. "An overview of global leaf area index (LAI): Methods, products, validation, and applications". In: *Reviews of Geophysics* 57.3 (2019), pp. 739–799 (cit. on p. 28).

[87] Guillermo Q Tabios III and Jose D Salas. "A comparative analysis of techniques for spatial interpolation of precipitation 1". In: *JAWRA Journal of the American Water Resources Association* 21.3 (1985), pp. 365–380 (cit. on p. 28).

[88] Matthew J Heaton, Abhirup Datta, Andrew O Finley, Reinhard Furrer, Joseph Guinness, Rajarshi Guhaniyogi, Florian Gerber, Robert B Gramacy, Dorit Hammerling, Matthias Katzfuss, et al. "A case study competition among methods for analyzing large spatial data". In: *Journal of Agricultural, Biological and Environmental Statistics* 24.3 (2019), pp. 398–425 (cit. on pp. 28, 30).

[89] Zhe Jiang. "A survey on spatial prediction methods". In: *IEEE Transactions on Knowledge and Data Engineering* 31.9 (2018), pp. 1645–1664 (cit. on pp. 28, 30).

[90] José-María Montero, Gema Fernández-Avilés, and Jorge Mateu. *Spatial and spatio-temporal geostatistical modeling and Kriging*. John Wiley & Sons, 2015 (cit. on p. 28).

[91] Noel Cressie. *Statistics for spatial data*. John Wiley & Sons, 2015 (cit. on pp. 28, 30).

[92] Luc Anselin. "Spatial econometrics: methods and models (Vol. 4)". In: *Studies in Operational Regional Science. Dordrecht: Springer Netherlands* (1988) (cit. on pp. 28, 31).

[93] RP Haining. "The moving average model for spatial interaction". In: *Transactions of the Institute of British Geographers* (1978), pp. 202–225 (cit. on pp. 28, 31, 42).

[94] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. "Image super-resolution using deep convolutional networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.2 (2015), pp. 295–307 (cit. on pp. 28, 32, 42).

[95] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1874–1883 (cit. on pp. 28, 32, 42).

[96] Richard Bellman. "A Markovian decision process". In: *Journal of Mathematics and Mechanics* (1957), pp. 679–684 (cit. on pp. 29, 33–34).

[97] Daniel G Krige. "A statistical approach to some basic mine valuation problems on the Witwatersrand". In: *Journal of the Southern African Institute of Mining and Metallurgy* 52.6 (1951), pp. 119–139 (cit. on p. 30).

[98] Oliver Schabenberger and Carol A Gotway. *Statistical methods for spatial data analysis*. CRC press, 2017 (cit. on p. 30).

[99] Mohamed A Bouhlel and Joaquim RRA Martins. "Gradient-enhanced Kriging for high-dimensional problems". In: *Engineering with Computers* 35.1 (2019), pp. 157–173 (cit. on p. 30).

[100] Koya Sato, Kei Inage, and Takeo Fujii. "On the performance of neural network residual Kriging in radio environment mapping". In: *IEEE Access* 7 (2019), pp. 94557–94568 (cit. on p. 30).

[101] Robert B Gramacy and Daniel W Apley. "Local Gaussian process approximation for large computer experiments". In: *Journal of Computational and Graphical Statistics* 24.2 (2015), pp. 561–578 (cit. on pp. 30, 41, 56).

[102] Håvard Rue, Andrea Riebler, Sigrunn H Sørbye, Janine B Illian, Daniel P Simpson, and Finn K Lindgren. "Bayesian computing with INLA: a review". In: *Annual Review of Statistics and Its Application* 4 (2017), pp. 395–421 (cit. on p. 30).

[103] Matthias Katzfuss. "A multi-resolution approximation for massive spatial datasets". In: *Journal of the American Statistical Association* 112.517 (2017), pp. 201–214 (cit. on p. 30).

[104] Florian Gerber, Rogier de Jong, Michael E Schaepman, Gabriela Schaepman-Strub, and Reinhard Furrer. "Predicting missing values in spatio-temporal remote sensing data". In: *IEEE Transactions on Geoscience and Remote Sensing* 56.5 (2018), pp. 2841–2853 (cit. on pp. 30, 46).

[105] Judea Pearl. *Reverend Bayes on inference engines: A distributed hierarchical approach*. Cognitive Systems Laboratory, School of Engineering and Applied Science, 1982 (cit. on p. 31).

[106] Le Song, Arthur Gretton, Danny Bickson, Yucheng Low, and Carlos Guestrin. "Kernel Belief Propagation". In: *International Conference on Artificial Intelligence and Statistics*. Ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, Apr. 2011, pp. 707–715. URL: https://proceedings.mlr.press/v15/song11a.html (cit. on p. 31).

[107] Xin Zheng, Xiaobin Lin, and Pengfei Wu. "Outdoor image restoration based on belief propagation algorithm and formalized MTF". In: *Journal of Physics: Conference Series*. Vol. 1651. 1. IOP Publishing. 2020, p. 012168 (cit. on p. 31).

[108] Anat Levin, Assaf Zomet, and Yair Weiss. "Learning How to Inpaint from Global Image Statistics." In: *ICCV*. Vol. 1. 2003, pp. 305–312 (cit. on p. 31).

[109] Steffen L Lauritzen. *Graphical models*. Vol. 17. Clarendon Press, 1996 (cit. on p. 31).

[110] Julian McAuley and Tibério Caetano. "Exploiting within-clique factorizations in junction-tree algorithms". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 525–532 (cit. on p. 31).

[111] Lu Zheng, Ole Mengshoel, and Jike Chong. "Belief propagation by message passing in junction trees: Computing each message faster using GPU parallelization". In: *arXiv preprint arXiv:1202.3777* (2012) (cit. on p. 31).

[112] Victor Garcia Satorras and Max Welling. "Neural enhanced belief propagation on factor graphs". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 685–693 (cit. on p. 31).

[113] Kevin Murphy, Yair Weiss, and Michael I Jordan. "Loopy belief propagation for approximate inference: An empirical study". In: *arXiv preprint arXiv:1301.6725* (2013) (cit. on p. 31).

[114] Kai Yang and Lung-fei Lee. "Identification and QML estimation of multivariate and simultaneous equations spatial autoregressive models". In: *Journal of Econometrics* 196.1 (2017), pp. 196–214 (cit. on p. 31).

[115] Miranda J Fix, Daniel S Cooley, and Emeric Thibaud. "Simultaneous autoregressive models for spatial extremes". In: *Environmetrics* 32.2 (2021), e2656 (cit. on p. 31).

[116] James Durbin. "Efficient estimation of parameters in moving-average models". In: *Biometrika* 46.3/4 (1959), pp. 306–316 (cit. on p. 31).

[117] Jianqing Qiu, Huimin Wang, Lin Hu, Changhong Yang, and Tao Zhang. "Spatial transmission network construction of influenza-like illness using dynamic Bayesian network and vector-autoregressive moving average model". In: *BMC infectious diseases* 21.1 (2021), pp. 1–9 (cit. on p. 31).

[118] Roberto Corizzo, Michelangelo Ceci, Hadi Fanaee-T, and Joao Gama. "Multi-aspect renewable energy forecasting". In: *Information Sciences* 546 (2021), pp. 701–722 (cit. on pp. 31, 58).

[119] Ali Soltani, Christopher James Pettit, Mohammad Heydari, and Fatemeh Aghaei. "Housing price variations using spatio-temporal data mining techniques". In: *Journal of Housing and the Built Environment* (2021), pp. 1–29 (cit. on p. 31).

[120] Junho Lee, Maria E Kamenetsky, Ronald E Gangnon, and Jun Zhu. "Clustered spatio-temporal varying coefficient regression model". In: *Statistics in Medicine* 40.2 (2021), pp. 465–480 (cit. on p. 31).

[121] Yara Abu Awad, Petros Koutrakis, Brent A Coull, and Joel Schwartz. "A spatio-temporal prediction model based on support vector machine regression: Ambient Black Carbon in three New England States". In: *Environmental Research* 159 (2017), pp. 427–434 (cit. on p. 31).

[122] Chao Wu, Mengjie Zhou, Pengyu Liu, and Mengjie Yang. "Analyzing COVID-19 using multisource data: An integrated approach of visualization, spatial regression, and machine learning". In: *GeoHealth* 5.8 (2021), e2021GH000439 (cit. on p. 31).

[123] Riku Hashimoto and Katsuya Suto. "SICNN: Spatial Interpolation with Convolutional Neural Networks for Radio Environment Mapping". In: *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE. 2020, pp. 167–170 (cit. on p. 32).

[124] Yuankai Wu, Dingyi Zhuang, Aurelie Labbe, and Lijun Sun. "Inductive graph neural networks for spatiotemporal Kriging". In: *arXiv preprint arXiv:2006.07527* (2020) (cit. on p. 32).

[125] N Sato and Kishor S Trivedi. "Accurate and efficient stochastic reliability analysis of composite services using their compact Markov reward model representations". In: *IEEE International Conference on Services Computing*. IEEE. 2007, pp. 114–121 (cit. on p. 33).

[126] Francesco Bianchi and Francesco Lo Presti. "A Markov reward model based greedy heuristic for the virtual network embedding problem". In: *International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. IEEE. 2016, pp. 373–378 (cit. on p. 33).

[127] Benjamin S. Murphy. *pykrige*. https://pypi.org/project/PyKrige/. 2020 (cit. on p. 40).

[128] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, and Frank Hutter. "Auto-sklearn: efficient and robust automated machine learning". In: *Automated Machine Learning*. Springer, Cham, 2019, pp. 113–134 (cit. on pp. 41, 78).

[129] Haifeng Jin, Qingquan Song, and Xia Hu. "Auto-keras: An efficient neural architecture search system". In: *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 1946–1956 (cit. on p. 42).

[130] World Bank DECRG. *Gross Domestic Product 2010*. https://datacatalog.worldbank.org/search/dataset/0037850. 2010 (cit. on p. 44).

[131] DACON. *Corona Data Visualization AI Contest*. https://www.dacon.io/competitions/official/235590/data/. 2020 (cit. on p. 44).

[132] OpenStreetMap. *OpenStreetMap*. https://www.openstreetmap.org/. 2019 (cit. on p. 44).

[133] Holger H Hoos. "Programming by optimization". In: *Communications of the ACM* 55.2 (2012), pp. 70–80 (cit. on p. 45).

[134] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. "Sequential model-based optimization for general algorithm configuration". In: *International Conference on Learning and Intelligent Optimization*. Springer. 2011, pp. 507–523 (cit. on p. 45).

[135] Frank Wilcoxon. "Individual comparisons by ranking methods". In: *Breakthroughs in Statistics*. Springer, 1992, pp. 196–202 (cit. on p. 47).

[136] Esam El-Araby, Tarek El-Ghazawi, Jacqueline Le Moigne, and Richard Irish. "Reconfigurable processing for satellite on-board automatic cloud cover assessment". In: *Journal of Real-Time Image Processing* 4.3 (2009), pp. 245–259 (cit. on p. 60).

[137] Gianluca Giuffrida, Lorenzo Diana, Francesco de Gioia, Gionata Benelli, Gabriele Meoni, Massimiliano Donati, and Luca Fanucci. "Cloudscout: a deep neural network for on-board cloud detection on hyperspectral images". In: *Remote Sensing* 12.14 (2020), p. 2205 (cit. on p. 60).

[138] Min Li, Soo Chin Liew, and Leong Keong Kwoh. "Producing cloud free and cloud-shadow free mosaic from cloudy IKONOS images". In: *IEEE International Geoscience and Remote Sensing Symposium*. Vol. 6. Ieee. 2003, pp. 3946–3948 (cit. on pp. 60, 64, 78).

[139] Eileen H Helmer and Bonnie Ruefenacht. “Cloud-free satellite image mosaics with regression trees and histogram matching”. In: *Photogrammetric Engineering & Remote Sensing* 71.9 (2005), pp. 1079–1089 (cit. on pp. 60, 64, 78).

[140] Huanfeng Shen, Huifang Li, Yan Qian, Liangpei Zhang, and Qiangqiang Yuan. “An effective thin cloud removal procedure for visible remote sensing images”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 96 (2014), pp. 224–235 (cit. on p. 63).

[141] Jun Liu, Xing Wang, Min Chen, Shuguang Liu, Xiran Zhou, Zhenfeng Shao, and Ping Liu. “Thin cloud removal from single satellite images”. In: *Optics Express* 22.1 (2014), pp. 618–632 (cit. on p. 63).

[142] Gensheng Hu, Xiaoyi Li, and Dong Liang. “Thin cloud removal from remote sensing images using multidirectional dual tree complex wavelet transform and transfer least square support vector regression”. In: *Journal of Applied Remote Sensing* 9.1 (2015), p. 095053 (cit. on p. 63).

[143] Meng Xu, Xiuping Jia, Mark Pickering, and Sen Jia. “Thin cloud removal from optical remote sensing images using the noise-adjusted principal components transform”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 149 (2019), pp. 215–225 (cit. on p. 63).

[144] Wenbo Li, Ying Li, Di Chen, and Jonathan Cheung-Wai Chan. “Thin cloud removal with residual symmetrical concatenation network”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 153 (2019), pp. 137–150 (cit. on p. 63).

[145] Jun Li, Zhaocong Wu, Zhongwen Hu, Jiaqi Zhang, Mingliang Li, Lu Mo, and Matthieu Molinier. “Thin cloud removal in optical remote sensing images based on generative adversarial networks and physical model of cloud distortion”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 166 (2020), pp. 373–389 (cit. on pp. 63, 76).

[146] Jun Li, Zhaocong Wu, Zhongwen Hu, Zilong Li, Yisong Wang, and Matthieu Molinier. “Deep learning based thin cloud removal fusing vegetation red edge and short wave infrared spectral information for Sentinel-2A imagery”. In: *Remote Sensing* 13.1 (2021), p. 157 (cit. on pp. 63, 75).

[147] Yujun Guo, Wei He, Yu Xia, and Hongyan Zhang. “Blind single-image-based thin cloud removal using a cloud perception integrated fast Fourier convolutional network”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 206 (2023), pp. 63–86 (cit. on p. 63).

[148] Chuanrong Zhang, Weidong Li, and David Travis. “Gaps-fill of SLC-off Landsat ETM+ satellite image using a geostatistical approach”. In: *International Journal of Remote Sensing* 28.22 (2007), pp. 5103–5122 (cit. on p. 63).

[149] USGS. *Phase 2 gap-fill algorithm: SLC-off gap-filled products gap-fill algorithm methodology*. https://www.usgs.gov/faqs/what-landsat-7-etm-slc-data. 2004 (cit. on p. 63).

[150] Fang Xu, Yilei Shi, Patrick Ebel, Lei Yu, Gui-Song Xia, Wen Yang, and Xiao Xiang Zhu. “GLF-CR: SAR-enhanced cloud removal with global–local fusion”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 192 (2022), pp. 268–278 (cit. on p. 63).

[151] Shuning Han, Jianmei Wang, and Shaoming Zhang. “Former-CR: A transformer-based thick cloud removal method with optical and SAR imagery”. In: *Remote Sensing* 15.5 (2023), p. 1196 (cit. on p. 63).

[152] Hao Liu, Bo Huang, and Jiajun Cai. “Thick cloud removal under land cover changes using multisource satellite imagery and a spatiotemporal attention network”. In: *IEEE Transactions on Geoscience and Remote Sensing* 61 (2023), pp. 1–18 (cit. on p. 63).

[153] Meng Xu, Furong Deng, Sen Jia, Xiuping Jia, and Antonio J Plaza. “Attention mechanism-based generative adversarial networks for cloud removal in Landsat images”. In: *Remote Sensing of Environment* 271 (2022), p. 112902 (cit. on p. 63).

[154] Faramarz Naderi Darbaghshahi, Mohammad Reza Mohammadi, and Mohsen Soryani. “Cloud removal in remote sensing images using generative adversarial networks and SAR-to-optical image translation”. In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), pp. 1–9 (cit. on p. 63).

[155] Ran Jing, Fuzhou Duan, Fengxian Lu, Miao Zhang, and Wenji Zhao. “Denoising diffusion probabilistic feature-based network for cloud removal in Sentinel-2 imagery”. In: *Remote Sensing* 15.9 (2023), p. 2217 (cit. on p. 63).

[156] Alber Hamersson Sanchez, Michelle Cristina A Picoli, Gilberto Camara, Pedro R Andrade, Michel Eustaquio D Chaves, Sarah Lechler, Anderson R Soares, Rennan FB Marujo, Rolf Ezequiel O Simões, Karine R Ferreira, et al. “Comparison of Cloud cover detection algorithms on Sentinel–2 images of the amazon tropical forest”. In: *Remote Sensing* 12.8 (2020), p. 1284 (cit. on p. 64).

[157] SentinelHub. *s2cloudless*. https://github.com/sentinel-hub/sentinel2-cloud-detector. 2018 (cit. on p. 64).

[158] Fengli Zou, Qingwu Hu, Yichuan Liu, Haidong Li, Xujie Zhang, and Yuqi Liu. "Spatiotemporal Changes and Driving Analysis of Ecological Environmental Quality along the Qinghai–Tibet Railway Using Google Earth Engine—A Case Study Covering Xining to Jianghe Stations". In: *Remote Sensing* 16.6 (2024), p. 951 (cit. on p. 64).

[159] John Brandt, Jessica Ertel, Justine Spore, and Fred Stolle. "Wall-to-wall mapping of tree extent in the tropics with Sentinel-1 and Sentinel-2". In: *Remote Sensing of Environment* 292 (2023), p. 113574 (cit. on p. 64).

[160] Pasquale Scaramuzza and Julia Barsi. "Landsat 7 scan line corrector-off gap-filled product development". In: *Proceeding of Pecora*. Vol. 16. 3. 2005, pp. 23–27 (cit. on p. 64).

[161] Hannes Feilhauer, Gregory P Asner, Roberta E Martin, and Sebastian Schmidtlein. "Brightness-normalized partial least squares regression for hyperspectral data". In: *Journal of Quantitative Spectroscopy and Radiative Transfer* 111.12-13 (2010), pp. 1947–1957 (cit. on p. 64).

[162] Xingwang Fan, Yuanbo Liu, Jinmei Tao, and Yongling Weng. "Soil salinity retrieval from advanced multi-spectral sensor with partial least square regression". In: *Remote Sensing* 7.1 (2015), pp. 488–511 (cit. on p. 64).

[163] Katherine Meacham-Hensold, Christopher M Montes, Jin Wu, Kaiyu Guan, Peng Fu, Elizabeth A Ainsworth, Taylor Pederson, Caitlin E Moore, Kenny Lee Brown, Christine Raines, et al. "High-throughput field phenotyping using hyperspectral reflectance and partial least squares regression (PLSR) reveals genetic modifications to photosynthetic capacity". In: *Remote Sensing of Environment* 231 (2019), pp. 111–176 (cit. on p. 64).

[164] Raphael Fischer, Nico Piatkowski, Charlotte Pelletier, Geoffrey I Webb, François Petitjean, and Katharina Morik. "No cloud on the horizon: probabilistic gap filling in satellite image series". In: *International Conference on Data Science and Advanced Analytics*. IEEE. 2020, pp. 546–555 (cit. on p. 64).

[165] Xiaolin Zhu, Feng Gao, Desheng Liu, and Jin Chen. "A modified neighborhood similar pixel interpolator approach for removing thick clouds in Landsat images". In: *IEEE Geoscience and Remote Sensing Letters* 9.3 (2011), pp. 521–525 (cit. on pp. 64, 79).

[166] Chao Zeng, Huanfeng Shen, and Liangpei Zhang. "Recovering missing pixels for Landsat ETM+ SLC-off imagery using multi-temporal regression analysis and a regularization method". In: *Remote Sensing of Environment* 131 (2013), pp. 182–194 (cit. on p. 64).

[167] Qing Cheng, Huanfeng Shen, Liangpei Zhang, Qiangqiang Yuan, and Chao Zeng. “Cloud removal for remotely sensed images by similar pixel replacement guided with a spatio-temporal MRF model”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 92 (2014), pp. 54–68 (cit. on p. 64).

[168] Farid Melgani. “Contextual reconstruction of cloud-contaminated multitemporal multispectral images”. In: *IEEE Transactions on Geoscience and Remote Sensing* 44.2 (2006), pp. 442–455 (cit. on p. 64).

[169] Bin Chen, Bo Huang, Lifan Chen, and Bing Xu. “Spatially and temporally weighted regression: A novel method to produce continuous cloud-free Landsat imagery”. In: *IEEE Transactions on Geoscience and Remote Sensing* 55.1 (2016), pp. 27–37 (cit. on p. 64).

[170] Zhe Zhu, Curtis E Woodcock, Christopher Holden, and Zhiqiang Yang. “Generating synthetic Landsat images based on all available Landsat data: Predicting Landsat surface reflectance at any given time”. In: *Remote Sensing of Environment* 162 (2015), pp. 67–83 (cit. on p. 64).

[171] Wen-Jie Zheng, Xi-Le Zhao, Yu-Bang Zheng, Jie Lin, Lina Zhuang, and Ting-Zhu Huang. “Spatial-spectral-temporal connective tensor network decomposition for thick cloud removal”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 199 (2023), pp. 182–194 (cit. on p. 64).

[172] Qiang Zhang, Qiangqiang Yuan, Jie Li, Zhiwei Li, Huanfeng Shen, and Liangpei Zhang. “Thick cloud and cloud shadow removal in multitemporal imagery using progressively spatio-temporal patch group deep learning”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 162 (2020), pp. 148–160 (cit. on pp. 65, 178).

[173] Qiang Zhang, Qiangqiang Yuan, Zhiwei Li, Fujun Sun, and Liangpei Zhang. “Combined deep prior with low-rank tensor SVD for thick cloud removal in multitemporal images”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 177 (2021), pp. 161–173 (cit. on p. 65).

[174] Xiaohu Zhao and Kebin Jia. “Cloud Removal in Remote Sensing Using Sequential-Based Diffusion Models”. In: *Remote Sensing* 15.11 (2023), p. 2861 (cit. on pp. 65, 178).

[175] Xuechao Zou, Kai Li, Junliang Xing, Yu Zhang, Shiying Wang, Lei Jin, and Pin Tao. “DiffCR: A Fast Conditional Diffusion Framework for Cloud Removal From Optical Satellite Images”. In: *IEEE Transactions on Geoscience and Remote Sensing* 62 (2024), pp. 1–14 (cit. on p. 65).

[176] Corinne Stucker, Vivien Sainte Fare Garnot, and Konrad Schindler. "U-TILISE: A Sequence-to-sequence Model for Cloud Removal in Optical Satellite Time Series". In: *IEEE Transactions on Geoscience and Remote Sensing* 61 (2023) (cit. on pp. 65, 77).

[177] Michael Schmitt, Lloyd Haydn Hughes, Chunping Qiu, and Xiao Xiang Zhu. "SEN12MS–A Curated Dataset of Georeferenced Multi-Spectral Sentinel-1/2 Imagery for Deep Learning and Data Fusion". In: *arXiv preprint arXiv:1906.07789* (2019) (cit. on p. 75).

[178] M. Buchhorn, B. Smets, L. Bertels, M. Lesiv, N.-E. Tsendbazar, D. Masiliunas, L. Linlin, M. Herold, and S. Fritz. *Copernicus Global Land Service: Land Cover 100m: Collection 3: epoch 2019: Globe (Version V3.0.1)*. Zenono. DOI: 10.5281/zenodo.3939050. 2020 (cit. on p. 76).

[179] Patrick Ebel, Yajin Xu, Michael Schmitt, and Xiao Xiang Zhu. "SEN12MS-CR-TS: A Remote Sensing Data Set for Multi-modal Multi-temporal Cloud Removal". In: *IEEE Transactions on Geoscience and Remote Sensing* (2022) (cit. on pp. 76–77, 89).

[180] Vishnu Sarukkai, Anirudh Jain, Burak Uzkent, and Stefano Ermon. "Cloud removal from satellite images using spatiotemporal generator networks". In: *IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 1796–1805 (cit. on p. 77).

[181] Vivien Sainte Fare Garnot and Loic Landrieu. "Panoptic segmentation of satellite image time series with convolutional temporal attention networks". In: *IEEE/CVF International Conference on Computer Vision*. 2021, pp. 4872–4881 (cit. on p. 77).

[182] Google. *Google Earth Engine*. https://earthengine.google.com/. 2022 (cit. on p. 78).

[183] Jonas Mockus. *Bayesian approach to global optimization: theory and applications*. Vol. 37. Springer Science & Business Media, 2012 (cit. on p. 78).

[184] Mikolaj Czerkawski, Robert Atkinson, Craig Michie, and Christos Tachtatzis. "SatelliteCloudGenerator: Controllable Cloud and Shadow Synthesis for Multi-Spectral Optical Satellite Images". In: *Remote Sensing* 15.17 (2023). ISSN: 2072-4292. DOI: 10.3390/rs15174138. URL: https://www.mdpi.com/2072-4292/15/17/4138 (cit. on pp. 80, 90).

[185] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (1989), pp. 600–612 (cit. on p. 83).

[186] Laurens Arp, Mitra Baratchi, and Holger H. Hoos. “VPint: value propagation-based spatial interpolation”. In: *Data Mining and Knowledge Discovery* 36 (2022), pp. 1647–1678 (cit. on p. 93).

[187] Luke A Brown, Courtney Meier, Harry Morris, Julio Pastor-Guzman, Gabriele Bai, Christophe Lerebourg, Nadine Gobron, Christian Lanconelli, Marco Clerici, and Jadunandan Dash. “Evaluation of global leaf area index and fraction of absorbed photosynthetically active radiation products over North America using Copernicus Ground Based Observations for Validation data”. In: *Remote Sensing of Environment* 247 (2020), p. 111935 (cit. on p. 98).

[188] National Ecological Observatory Network (NEON). *NEON)*. URL: https://data.neonscience.org (cit. on p. 98).

[189] C Bacour, F Baret, D Béal, M Weiss, and K Pavageau. “Neural network estimation of LAI, fAPAR, fCover and LAI× Cab, from top of canopy MERIS reflectance data: Principles and validation”. In: *Remote Sensing of Environment* 105.4 (2006), pp. 313–325 (cit. on p. 98).

[190] Stéphane Jacquemoud, Frédéric Baret, Bruno Andrieu, FM Danson, and K Jaggard. “Extraction of vegetation biophysical parameters by inversion of the PROSPECT+ SAIL models on sugar beet canopy reflectance data. Application to TM and AVIRIS sensors”. In: *Remote Sensing of Environment* 52.3 (1995), pp. 163–172 (cit. on pp. 98–99, 156).

[191] Stéphane Jacquemoud, Wout Verhoef, Frédéric Baret, Cédric Bacour, Pablo J Zarco-Tejada, Gregory P Asner, Christophe François, and Susan L Ustin. “PROSPECT+ SAIL models: A review of use for vegetation characterization”. In: *Remote Sensing of Environment* 113 (2009), S56–S66 (cit. on pp. 98, 183).

[192] Gabriele Bai, Jadunandan Dash, Luke Brown, Courtney Meier, C Lerebourg, E Ronco, N Lamquin, V Bruniquel, M Clerici, and N Gobron. “GBOV (ground-based observation for validation): a Copernicus service for validation of vegetation land products”. In: *IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2019, pp. 4592–4594 (cit. on p. 99).

[193] NASA. *SMAPVEX08 In Situ Vegetation Data*. 2008. DOI: 10.5067/US4X5QPYH6DB. URL: https://cmr.earthdata.nasa.gov/search/concepts/C1000001420-NSIDC_ECS.html (cit. on p. 99).

[194] Liangyun, Liu and Bowen, Song. *ValLAI_Crop: Validation dataset for coarse-resolution satellite LAI product over Chinese cropland*. 2021. DOI: 10.5281/zenodo.4080910. URL: https://data.niaid.nih.gov/resources?id=zenodo_4080910 (cit. on p. 99).

[195] Jingwen Wang, Raul Lopez-Lozano, Marie Weiss, Samuel Buis, Wenjuan Li, Shouyang Liu, Frédéric Baret, and Jiahua Zhang. "Crop specific inversion of PROSAIL to retrieve green area index (GAI) from several decametric satellites using a Bayesian framework". In: *Remote Sensing of Environment* 278 (2022), p. 113085 (cit. on pp. 99, 131).

[196] M. Meroni, R. Colombo, and C. Panigada. "Inversion of a radiative transfer model with hyperspectral observations for LAI mapping in poplar plantations". In: *Remote Sensing of Environment* 92.2 (2004), pp. 195–206. ISSN: 0034-4257 (cit. on p. 99).

[197] Cédric Bacour, Stéphane Jacquemoud, Marc Leroy, Olivier Hautecœur, Marie Weiss, Laurent Prévot, Nadine Bruguier, and Habiba Chauki. "Reliability of the estimation of vegetation characteristics by inversion of three canopy reflectance models on airborne POLDER data". In: *Agronomie* 22.6 (2002), pp. 555–565 (cit. on p. 99).

[198] Anting Guo, Wenjiang Huang, Binxiang Qian, Huichun Ye, Quanjun Jiao, Xiangzhe Cheng, and Chao Ruan. "A hybrid model coupling PROSAIL and continuous wavelet transform based on multi-angle hyperspectral data improves maize chlorophyll retrieval". In: *International Journal of Applied Earth Observation and Geoinformation* 132 (2024), p. 104076 (cit. on p. 99).

[199] Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. "An efficient approach for assessing hyperparameter importance". In: *International Conference on Machine Learning*. PMLR. 2014, pp. 754–762 (cit. on pp. 102–103).

[200] Ilya M Sobol. "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates". In: *Mathematics and computers in simulation* 55.1-3 (2001), pp. 271–280 (cit. on pp. 102–103).

[201] Thomas Bäck. *Evolutionary computation 1: Basic algorithms and operators*. CRC press, 2018 (cit. on p. 105).

[202] Amrita Chakraborty and Arpan Kumar Kar. "Swarm intelligence: A review of algorithms". In: *Nature-inspired Computing and Optimization: Theory and Applications* (2017), pp. 475–494 (cit. on p. 105).

[203] Roshanak Darvishzadeh, Andrew Skidmore, Martin Schlerf, and Clement Atzberger. "Inversion of a radiative transfer model for estimating vegetation LAI and chlorophyll in a heterogeneous grassland". In: *Remote Sensing of Environment*. Earth Observations for Terrestrial Biodiversity and Ecosystems Special Issue 112.5 (May 2008), pp. 2592–2604. ISSN: 0034-4257. DOI: 10.1016/j.rse.2007.12.003. URL: https://www.sciencedirect.com/science/article/pii/S0034425707004968 (cit. on p. 108).

[204] Jochem Verrelst, Juan Pablo Rivera, Ganna Leonenko, Luis Alonso, and José Moreno. "Optimizing LUT-Based RTM Inversion for Semiautomatic Mapping of Crop Biophysical Parameters from Sentinel-2 and -3 Data: Role of Cost Functions". In: *IEEE Transactions on Geoscience and Remote Sensing* 52.1 (Jan. 2014), pp. 257–269. ISSN: 1558-0644. DOI: 10.1109/TGRS.2013.2238242 (cit. on p. 108).

[205] M Varah James. "On the numerical solution of ill-conditioned linear systems with applications to ill-posed problems". In: *SIAM Journal on Numerical Analysis* 10.2 (1973), pp. 257–267 (cit. on pp. 109, 126, 134).

[206] David A Belsley and RW Oldford. "The general problem of ill conditioning and its role in statistical analysis". In: *Computational Statistics & Data Analysis* 4.2 (1986), pp. 103–120 (cit. on pp. 109, 126, 134).

[207] Fei Xu and Ben Somers. "Unmixing-based Sentinel-2 downscaling for urban land cover mapping". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 171 (2021), pp. 133–154 (cit. on p. 110).

[208] Katja Kowalski, Akpona Okujeni, and Patrick Hostert. "A generalized framework for drought monitoring across central European grassland gradients with Sentinel-2 time series". In: *Remote Sensing of Environment* 286 (2023), p. 113449 (cit. on p. 110).

[209] Ion Sola, Alberto García-Martín, Leire Sandonís-Pozo, Jesús Álvarez-Mozos, Fernando Pérez-Cabello, María González-Audícana, and Raquel Montorio Llovería. "Assessment of atmospheric correction methods for Sentinel-2 images in Mediterranean landscapes". In: *International journal of applied earth observation and geoinformation* 73 (2018), pp. 63–76 (cit. on pp. 115–116).

[210] Jochem Verrelst, Jorge Vicent, Juan Pablo Rivera-Caicedo, Maria Lumbierres, Pablo Morcillo-Pallarés, and José Moreno. "Global sensitivity analysis of leaf-canopy-atmosphere RTMs: Implications for biophysical variables retrieval from top-of-atmosphere radiance data". In: *Remote sensing* 11.16 (2019), p. 1923 (cit. on p. 115).

[211] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. "Extracting and composing robust features with denoising autoencoders". In: *Proceedings of the 25th International Conference on Machine Learning*. 2008, pp. 1096–1103 (cit. on pp. 122, 179).

[212] Grzegorz Łukaszewicz and Piotr Kalita. "Navier–Stokes equations". In: *Advances in Mechanics and Mathematics* 34 (2016) (cit. on p. 126).

[213] Charles R Doering and John D Gibbon. *Applied analysis of the Navier-Stokes equations*. 12. Cambridge University Press, 1995 (cit. on p. 126).

[214] Magdalena Main-Knorn, Bringfried Pflug, Jerome Louis, Vincent Debaecker, Uwe Müller-Wilm, and Ferran Gascon. "Sen2Cor for Sentinel-2". In: *Image and Signal Processing for Remote Sensing*. Vol. 23. SPIE. 2017, pp. 37–48 (cit. on p. 126).

[215] Vitor Souza Martins, Claudio Clemente Faria Barbosa, Lino Augusto Sander De Carvalho, Daniel Schaffer Ferreira Jorge, Felipe de Lucia Lobo, and Evlyn Márcia Leão de Moraes Novo. "Assessment of atmospheric correction methods for Sentinel-2 MSI images applied to Amazon floodplain lakes". In: *Remote Sensing* 9.4 (2017), p. 322 (cit. on p. 126).

[216] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. "The frontier of simulation-based inference". In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30055–30062 (cit. on pp. 126–127, 129, 136).

[217] Lorenzo Pacchiardi and Ritabrata Dutta. "Score matched neural exponential families for likelihood-free inference". In: *Journal of Machine Learning Research* 23.38 (2022), pp. 1–71 (cit. on pp. 126, 129).

[218] Michael U Gutmann, Jukka Cor, et al. "Bayesian optimization for likelihood-free inference of simulator-based statistical models". In: *Journal of Machine Learning Research* 17.125 (2016), pp. 1–47 (cit. on pp. 126, 129).

[219] Jan-Matthis Lueckmann, Jan Boelts, David Greenberg, Pedro Goncalves, and Jakob Macke. "Benchmarking simulation-based inference". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 343–351 (cit. on pp. 126, 157).

[220] Yuji Kim and Nori Nakata. "Geophysical inversion versus machine learning in inverse problems". In: *The Leading Edge* 37.12 (2018), pp. 894–901 (cit. on p. 126).

[221] Gregory Ongie, Ajil Jalal, Christopher A Metzler, Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett. "Deep learning techniques for inverse problems in imaging". In: *IEEE Journal on Selected Areas in Information Theory* 1.1 (2020), pp. 39–56 (cit. on p. 126).

[222] Anil Aswani, Zuo-Jun Shen, and Auyon Siddiq. "Inverse optimization with noisy data". In: *Operations Research* 66.3 (2018), pp. 870–892 (cit. on p. 126).

[223] Pingheng Li and Quan Wang. "Retrieval of leaf biochemical parameters using PROSPECT inversion: A new approach for alleviating ill-posed problems". In: *IEEE Transactions on Geoscience and Remote Sensing* 49.7 (2011), pp. 2499–2506 (cit. on p. 126).

[224] Joel L Horowitz. “Ill-posed inverse problems in economics”. In: *Annual Review of Economics* 6.1 (2014), pp. 21–51 (cit. on p. 126).

[225] David A Cicci. “Improving gravity field determination in ill-conditioned inverse problems”. In: *Computers & Geosciences* 18.5 (1992), pp. 509–516 (cit. on p. 126).

[226] Jayanta Mandi, Peter J Stuckey, Tias Guns, et al. “Smart predict-and-optimize for hard combinatorial optimization problems”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 02. 2020, pp. 1603–1610 (cit. on pp. 126, 129, 133).

[227] Carla E Brodley and Mark A Friedl. “Identifying mislabeled training data”. In: *Journal of Artificial Intelligence Research* 11 (1999), pp. 131–167 (cit. on p. 128).

[228] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. “Learning with noisy labels”. In: *Advances in neural information processing systems* 26 (2013) (cit. on p. 128).

[229] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. “Learning from noisy labels with deep neural networks: A survey”. In: *IEEE transactions on neural networks and learning systems* 34.11 (2022), pp. 8135–8153 (cit. on p. 128).

[230] Curtis Northcutt, Lu Jiang, and Isaac Chuang. “Confident learning: Estimating uncertainty in dataset labels”. In: *Journal of Artificial Intelligence Research* 70 (2021), pp. 1373–1411 (cit. on p. 128).

[231] Alexandra N Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. “Learning from disagreement: A survey”. In: *Journal of Artificial Intelligence Research* 72 (2021), pp. 1385–1470 (cit. on p. 128).

[232] Alexander Philip Dawid and Allan M Skene. “Maximum likelihood estimation of observer error-rates using the EM algorithm”. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 28.1 (1979), pp. 20–28 (cit. on p. 128).

[233] Silviu Paun, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. “Comparing Bayesian models of annotation”. In: *Transactions of the Association for Computational Linguistics* 6 (2018), pp. 571–585 (cit. on p. 128).

[234] Lora Aroyo and Chris Welty. “Truth is a lie: Crowd truth and the seven myths of human annotation”. In: *AI Magazine* 36.1 (2015), pp. 15–24 (cit. on p. 128).

[235] Mélanie Bernhardt, Daniel C Castro, Ryutaro Tanno, Anton Schwaighofer, Kerem C Tezcan, Miguel Monteiro, Shruthi Bannur, Matthew P Lungren, Aditya Nori, Ben Glocker, et al. "Active label cleaning for improved dataset quality under resource constraints". In: *Nature communications* 13.1 (2022), p. 1161 (cit. on p. 128).

[236] Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. "Beyond synthetic noise: Deep learning on controlled noisy labels". In: *International conference on machine learning*. PMLR. 2020, pp. 4804–4815 (cit. on p. 128).

[237] Yingsong Huang, Bing Bai, Shengwei Zhao, Kun Bai, and Fei Wang. "Uncertainty-aware learning against label noise on imbalanced datasets". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 36. 6. 2022, pp. 6960–6969 (cit. on p. 128).

[238] Jang-Hyun Kim, Sangdoo Yun, and Hyun Oh Song. "Neural relation graph: A unified framework for identifying label noise and outlier data". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 43754–43779 (cit. on p. 128).

[239] Kush Bhatia, Prateek Jain, Parameswaran Kamalaruban, and Purushottam Kar. "Consistent robust regression". In: *Advances in Neural Information Processing Systems* 30 (2017) (cit. on p. 129).

[240] Dong Huang, Ricardo Cabral, and Fernando De la Torre. "Robust regression". In: *IEEE transactions on pattern analysis and machine intelligence* 38.2 (2015), pp. 363–375 (cit. on p. 129).

[241] Ruidi Chen and Ioannis Ch Paschalidis. "A robust learning approach for regression models based on distributionally robust optimization". In: *Journal of Machine Learning Research* 19.13 (2018), pp. 1–48 (cit. on p. 129).

[242] Toon Vanderschueren, Tim Verdonck, Bart Baesens, and Wouter Verbeke. "Predict-then-optimize or predict-and-optimize? An empirical evaluation of cost-sensitive learning strategies". In: *Information sciences* 594 (2022), pp. 400–415 (cit. on p. 129).

[243] Georgiana Ifrim, Barry O'Sullivan, and Helmut Simonis. "Properties of energy-price forecasts for scheduling". In: *International Conference on Principles and Practice of Constraint Programming*. Springer. 2012, pp. 957–972 (cit. on p. 129).

[244] Adam N Elmachtoub and Paul Grigas. "Smart "predict, then optimize"". In: *Management Science* 68.1 (2022), pp. 9–26 (cit. on p. 129).

[245] Jarno Lintusaari, Henri Vuollekoski, Antti Kangasrääsiö, Kusti Skytén, Marko Järvenpää, Pekka Marttinen, Michael U Gutmann, Aki Vehtari, Jukka Corander, and Samuel Kaski. "Elfi: Engine for likelihood-free inference". In: *Journal of Machine Learning Research* 19.16 (2018), pp. 1–7 (cit. on p. 129).

[246] Duncan Watson-Parris, Andrew Williams, Lucia Deaconu, and Philip Stier. "Model calibration using ESEm v1. 1.0–an open, scalable Earth system emulator". In: *Geoscientific Model Development* 14.12 (2021), pp. 7659–7672 (cit. on p. 129).

[247] Maximilian Dax, Stephen R Green, Jonathan Gair, Jakob H Macke, Alessandra Buonanno, and Bernhard Schölkopf. "Real-time gravitational wave science with neural posterior estimation". In: *Physical Review Letters* 127.24 (2021), p. 241103 (cit. on p. 129).

[248] David B Bernstein, Snorre Sulheim, Eivind Almaas, and Daniel Segrè. "Addressing uncertainty in genome-scale metabolic model reconstruction and analysis". In: *Genome Biology* 22 (2021), pp. 1–22 (cit. on p. 129).

[249] Scott A Sisson, Yanan Fan, and Mark Beaumont. *Handbook of approximate Bayesian computation*. CRC press, 2018 (cit. on pp. 129, 136).

[250] Paul Marjoram, John Molitor, Vincent Plagnol, and Simon Tavaré. "Markov chain Monte Carlo without likelihoods". In: *Proceedings of the National Academy of Sciences* 100.26 (2003), pp. 15324–15328 (cit. on p. 130).

[251] Scott A Sisson, Yanan Fan, and Mark M Tanaka. "Sequential Monte Carlo without likelihoods". In: *Proceedings of the National Academy of Sciences* 104.6 (2007), pp. 1760–1765 (cit. on pp. 130, 157).

[252] Manuel Glöckler, Michael Deistler, and Jakob H Macke. "Variational methods for simulation-based inference". In: *International Conference on Learning Representations*. 2022 (cit. on p. 130).

[253] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. "Variational inference: A review for statisticians". In: *Journal of the American statistical Association* 112.518 (2017), pp. 859–877 (cit. on p. 130).

[254] Ankush Ganguly, Sanjana Jain, and Ukrit Watchareeruetai. "Amortized variational inference: A systematic review". In: *Journal of Artificial Intelligence Research* 78 (2023), pp. 167–215 (cit. on p. 130).

[255] Laura von Rueden, Sebastian Mayer, Rafet Sifa, Christian Bauckhage, and Jochen Garcke. "Combining machine learning and simulation to a hybrid modelling approach: Current and future directions". In: *Advances in Intelligent Data Analysis XVIII: 18th International Symposium on Intelligent Data Analysis, IDA 2020, Konstanz, Germany, April 27–29, 2020, Proceedings 18*. Springer. 2020, pp. 548–560 (cit. on p. 130).

[256] Andrew Zammit-Mangion, Matthew Sainsbury-Dale, and Raphaël Huser. "Neural methods for amortized inference". In: *Annual Review of Statistics and Its Application* 12 (2024) (cit. on p. 130).

[257] Tom Rainforth, Adam Golinski, Frank Wood, and Sheheryar Zaidi. "Target–aware Bayesian inference: how to beat optimal conventional estimators". In: *Journal of Machine Learning Research* 21.88 (2020), pp. 1–54 (cit. on p. 130).

[258] Manuel Gloeckler, Michael Deistler, and Jakob H Macke. "Adversarial robustness of amortized Bayesian inference". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 11493–11524 (cit. on p. 130).

[259] Manuel Gloeckler, Michael Deistler, Christian Weilbach, Frank Wood, and Jakob H Macke. "All-in-one simulation-based inference". In: *Proceedings of the 41st International Conference on Machine Learning* (2024) (cit. on p. 130).

[260] Jonas Wildberger, Maximilian Dax, Simon Buchholz, Stephen Green, Jakob H Macke, and Bernhard Schölkopf. "Flow matching for scalable simulation-based inference". In: *Advances in Neural Information Processing Systems* 36 (2024) (cit. on p. 130).

[261] Joeri Hermans, Arnaud Delaunoy, François Rozet, Antoine Wehenkel, and Gilles Louppe. "A crisis in simulation-based inference? Beware, your posterior approximations can be unfaithful". In: *Transactions on Machine Learning Research* (2022) (cit. on p. 130).

[262] Rahul Rahaman et al. "Uncertainty quantification and deep ensembles". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 20063–20075 (cit. on pp. 131, 160).

[263] Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. "Hyperparameter ensembles for robustness and uncertainty quantification". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6514–6527 (cit. on pp. 131, 160).

[264] J Padarian, B Minasny, and Alex B McBratney. “Assessing the uncertainty of deep learning soil spectral models using Monte Carlo dropout”. In: *Geoderma* 425 (2022), p. 116063 (cit. on p. 131).

[265] Daily Milanés-Hermosilla, Rafael Trujillo Codorniú, René López-Baracaldo, Roberto Sagaró-Zamora, Denis Delisle-Rodriguez, John Jairo Villarejo-Mayor, and José Ricardo Núñez-Álvarez. “Monte Carlo dropout for uncertainty estimation and motor imagery classification”. In: *Sensors* 21.21 (2021), p. 7241 (cit. on p. 131).

[266] Yongchan Kwon, Joong-Ho Won, Beom Joon Kim, and Myunghee Cho Paik. “Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation”. In: *Computational Statistics & Data Analysis* 142 (2020), p. 106816 (cit. on p. 131).

[267] Audrey Olivier, Michael D Shields, and Lori Graham-Brady. “Bayesian neural networks for uncertainty quantification in data-driven materials modeling”. In: *Computer Methods in Applied Mechanics and Engineering* 386 (2021), p. 114079 (cit. on p. 131).

[268] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. “A review of uncertainty quantification in deep learning: Techniques, applications and challenges”. In: *Information Fusion* 76 (2021), pp. 243–297 (cit. on p. 131).

[269] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. “A survey of uncertainty in deep neural networks”. In: *Artificial Intelligence Review* 56.Suppl 1 (2023), pp. 1513–1589 (cit. on p. 131).

[270] Apostolos F Psaros, Xuhui Meng, Zongren Zou, Ling Guo, and George Em Karniadakis. “Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons”. In: *Journal of Computational Physics* 477 (2023), p. 111902 (cit. on p. 131).

[271] Yinhao Zhu, Nicholas Zabaras, Phaedon-Stelios Koutsourelakis, and Paris Perdikaris. “Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data”. In: *Journal of Computational Physics* 394 (2019), pp. 56–81 (cit. on p. 131).

[272] Andrea Beck, Jakob Dürrwächter, Thomas Kuhn, Fabian Meyer, Claus-Dieter Munz, and Christian Rohde. “hp-Multilevel Monte Carlo Methods for Uncertainty Quantification of Compressible Navier–Stokes Equations”. In: *SIAM Journal on Scientific Computing* 42.4 (2020), B1067–B1091 (cit. on p. 131).

[273] Laura Martínez-Ferrer, Álvaro Moreno-Martínez, Manuel Campos-Taberner, Francisco Javier García-Haro, Jordi Muñoz-Marí, Steven W Running, John Kimball, Nicholas Clinton, and Gustau Camps-Valls. "Quantifying uncertainty in high resolution biophysical variable retrieval with machine learning". In: *Remote Sensing of Environment* 280 (2022), p. 113199 (cit. on p. 131).

[274] Matthias König, Annelot W Bosman, Holger H Hoos, and Jan N van Rijn. "Critically Assessing the State of the Art in Neural Network Verification". In: *Journal of Machine Learning Research* 25.12 (2024), pp. 1–53 (cit. on p. 136).

[275] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572* (2014) (cit. on p. 136).

[276] Linyi Li, Tao Xie, and Bo Li. "Sok: Certified robustness for deep neural networks". In: *2023 IEEE Symposium on Security and Privacy*. IEEE. 2023, pp. 1289–1310 (cit. on p. 136).

[277] Jing Zhao, JJing Li, Qinhuo Liu, and Le Yang. "A preliminary study on mechanism of LAI inversion saturation". In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 39 (2012), pp. 77–81 (cit. on p. 140).

[278] Nikolaus Hansen and Andreas Ostermeier. "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation". In: *Proceedings of IEEE international conference on evolutionary computation*. IEEE. 1996, pp. 312–317 (cit. on p. 144).

[279] Jacob de Nobel, Diederick Vermetten, Hao Wang, Carola Doerr, and Thomas Bäck. "Tuning as a means of assessing the benefits of new ideas in interplay with existing algorithmic modules". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 2021, pp. 1375–1384 (cit. on p. 144).

[280] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. "Quality diversity: A new frontier for evolutionary computation". In: *Frontiers in Robotics and AI* (2016), p. 40 (cit. on p. 147).

[281] Johannes Jahn. *Scalarization in multi objective optimization*. Springer, 1985 (cit. on p. 150).

[282] Ruihao Zheng and Zhenkun Wang. "A generalized scalarization method for evolutionary multi-objective optimization". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 10. 2023, pp. 12518–12525 (cit. on p. 150).

[283] Der-Tsai Lee and Bruce J Schachter. “Two algorithms for constructing a Delaunay triangulation”. In: *International Journal of Computer & Information Sciences* 9.3 (1980), pp. 219–242 (cit. on p. 150).

[284] David Ackley. *A connectionist machine for genetic hillclimbing*. Vol. 28. Springer Science & Business Media, 2012 (cit. on p. 153).

[285] Frank Hoffmeister and Thomas Bäck. “Genetic algorithms and evolution strategies: Similarities and differences”. In: *International Conference on Parallel Problem Solving from Nature*. Springer. 1990, pp. 455–469 (cit. on p. 153).

[286] Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. “SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization”. In: *Journal of Machine Learning Research* 23.54 (2022), pp. 1–9 (cit. on p. 155).

[287] Patrick Raanes, Yumeng Chen, Colin Grudzien, Maxime Tondeur, and Remy Dubois. *DAPPER*. 2023. URL: https://nansencenter.github.io/DAPPER/ (cit. on p. 157).

[288] William Ditto and Toshinori Munakata. “Principles and applications of chaotic systems”. In: *Communications of the ACM* 38.11 (1995), pp. 96–102 (cit. on p. 157).

[289] Sungyoon Lee, Hoki Kim, and Jaewook Lee. “Graddiv: Adversarial robustness of randomized neural networks via gradient diversity regularization”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022) (cit. on p. 160).

[290] Yannik Schälte, Emmanuel Klinger, Emad Alamoudi, and Jan Hasenauer. “pyABC: Efficient and robust easy-to-use approximate Bayesian computation”. In: *Journal of Open Source Software* 7.74 (2022), p. 4304. DOI: 10.21105/joss.04304. URL: https://doi.org/10.21105/joss.04304 (cit. on p. 160).

[291] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. “TabPFN: A transformer that solves small tabular classification problems in a second”. In: *International Conference on Learning Representations*. 2023 (cit. on p. 161).

[292] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. “Accurate predictions on small data with a tabular foundation model”. In: *Nature* (2025) (cit. on p. 161).

[293] Nguyen Dang, Özgür Akgün, Joan Espasa, Ian Miguel, and Peter Nightingale. “A Framework for Generating Informative Benchmark Instances”. In: *28th International Conference on Principles and Practice of Constraint Programming*. 2022, p. 1 (cit. on p. 176).

[294] Will Steffen, Katherine Richardson, Johan Rockström, Hans Joachim Schellnhuber, Opha Pauline Dube, Sébastien Dutreuil, Timothy M Lenton, and Jane Lubchenco. “The emergence and evolution of Earth System Science”. In: *Nature Reviews Earth & Environment* 1.1 (2020), pp. 54–63 (cit. on p. 176).

[295] Jure Brence, Ljupčo Todorovski, and Sašo Džeroski. “Probabilistic grammars for equation discovery”. In: *Knowledge-Based Systems* 224 (2021), p. 107077 (cit. on p. 177).

[296] S Jacquemound, L Bidel, C Francois, and G Pavan. *ANGERS Leaf Optical Properties Database*. 2003 (cit. on p. 177).

[297] National Ecological Observatory Network (NEON). *LAI - spectrometer - mosaic (DP3.30012.001)*. en. 2023. DOI: 10.48443/7X5A-MN68. URL: https://data.neonscience.org/data-products/DP3.30012.001/RELEASE-2023 (cit. on p. 177).

[298] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.” In: *Journal of Machine Learning Research* 11.12 (2010) (cit. on p. 179).

[299] Natalie Maus, Haydn Jones, Juston Moore, Matt J Kusner, John Bradshaw, and Jacob Gardner. “Local latent space bayesian optimization over structured inputs”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 34505–34518 (cit. on p. 180).

[300] Austin Tripp, Erik Daxberger, and José Miguel Hernández-Lobato. “Sample-efficient optimization in the latent space of deep generative models via weighted retraining”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 11259–11272 (cit. on p. 180).

[301] Peng Sun, Peter M van Bodegom, Joris Timmermans, Shuwen Liu, Jin Wu, and Marco D Visser. “Hybrid modelling of leaf traits: Integrating neural networks with radiative transfer theory”. In: *Remote Sensing of Environment* 329 (2025), p. 114958 (cit. on p. 181).

[302] Roberta H Yuhas, Alexander FH Goetz, and Joe W Boardman. “Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm”. In: *JPL Airborne Geoscience Workshop. Volume 1: AVIRIS Workshop*. 1992 (cit. on p. 218).

A

SUPPLEMENTARY INFORMATION

A.1. PROSAIL INVERSION IMPLEMENTATION DETAILS

A.1.1. LOSS FUNCTIONS

When performing analyses on a loss landscape, the selection of an appropriate loss function (similarity metric d) can be of great importance. In the case of multispectral data, the simplest approach would be to consider all bands independently, and apply the mean absolute error (MAE) (or L_1 loss) function to compare the simulated spectrum $\hat{\mathbf{x}}$ and the observed spectrum $\mathbf{x}$ as:

$$MAE(\hat{\mathbf{x}}, \mathbf{x}) = \frac{1}{B} \cdot \sum_{b=0}^{B} |\hat{\mathbf{x}}_b - \mathbf{x}_b| \tag{A.1}$$

Here B is the number of bands in the spectrum. However, the values of spectral bands can greatly differ in magnitude, causing the MAE to be biased towards the bands with the greatest expected reflectance values (such as infrared). A relatively simple way to alleviate this problem is to use the proportional mean absolute error (PMAE), which is equivalent to the mean absolute percentage error (MAPE) used in other work [50], but does not convert the representation to percentages:

$$PMAE(\hat{\mathbf{x}}, \mathbf{x}) = \sum_{b=0}^{B} \frac{|\hat{\mathbf{x}}_b - \mathbf{x}_b|}{s_b} \tag{A.2}$$

MAE and PMAE are intuitive loss functions for the reconstruction of spectral data. However, in some applications, the brightness or albedo of a spectrum overall is less important than the ratio of band values compared to other band values,

forming the hue of the spectrum. The spectral angle mapper (SAM) loss function [302] can be used to try to capture this aspect of a reconstruction, and can be computed as:

$$SAM(\hat{\mathbf{x}},\mathbf{x}) = \arccos \frac{\hat{\mathbf{x}}\cdot\mathbf{x}}{(\hat{\mathbf{x}}\cdot\hat{\mathbf{x}}) * (\mathbf{x}\cdot\mathbf{x})} \cdot \frac{180}{\pi} \tag{A.3}$$

A.1.2. OPTIMISATION PROCEDURE

We treat the numerical optimisation procedure as a black-box optimisation problem. Within black-box optimisation algorithms, the tradeoff between *exploration* (identifying promising new parts of the search space) and *exploitation* (improving already known promising solutions until convergence) is often a central concept. In unimodal landscapes, only a single optimum exists, allowing greedy algorithms such as hill climbing or greedy local search (for an overview of stochastic local search methods, see [64]) to quickly converge to a local optimum, committing fully to exploitation. If the landscape is multimodal, there are multiple different local optima, causing greedy algorithms to get stuck in local optima. In this case, algorithms that add more exploration to their optimisation heuristics would be necessary.

Our experiments used greedy local search as an optimisation algorithm. For every instance, we initialised the current solution c to the mean of the prior distributions of the free parameters. At every optimisation step, we generated a candidate new solution $\boldsymbol{\theta}'$ by perturbing the elements of $\boldsymbol{\theta}$: $\boldsymbol{\theta}'_p = c_p + \mathcal{N}(0,\sigma_p)$, where p is a parameter in $\boldsymbol{\theta}$ and $\sigma_p = 0.05 \cdot \frac{max(p)-min(p)}{2}$. Here σ_p represents the intensity of the perturbation for a parameter p; we set it to 5% of the middle-way point of the parameter range (e.g., if a parameter ranges from 0 to 10, its perturbation intensity would be $0.05 \cdot \frac{10-0}{2} = 0.25$), though other mutation strategies could also be considered. If $\mathcal{L}(M(\boldsymbol{\theta}'),\mathbf{x}) < \mathcal{L}(M(\boldsymbol{\theta}),\mathbf{x})$, $\boldsymbol{\theta}'$ becomes the new $\boldsymbol{\theta}$. This procedure is iterated until the function evaluation budget has been exhausted.

Using greedy local search resulted in two main advantages. First, convergence will be fast, reducing the computational load of our experiments. Second, using a greedy algorithm allows us to test for multimodality (since the algorithm could converge to different optima when repeating an optimisation procedure), which is important to Section 5.4.1.

We note that, if the results for our experiments described in Section 5.4.1 indicate that PROSAIL inversion is a multimodal problem, a global optimisation method would need to be used to enable reliable convergence to a global optimum.

A.2. PROSAIL INVERSION: REAL-WORLD SENTINEL-2 DATA

Although our main experiments in Chapter 5 focussed on simulated data, since this allowed us access to both noise-free and noisy data, it is possible that there are other factors, beyond the Gaussian noise and spectral mixing we considered in Section 5.4.2, that result in ill-posedness for parameter retrieval using real-world data. To ensure that the patterns we found (well-posedness of PROSAIL inversion) hold for real-world data as well, we performed additional analyses on real-world data, as shown in Figure 5.6.

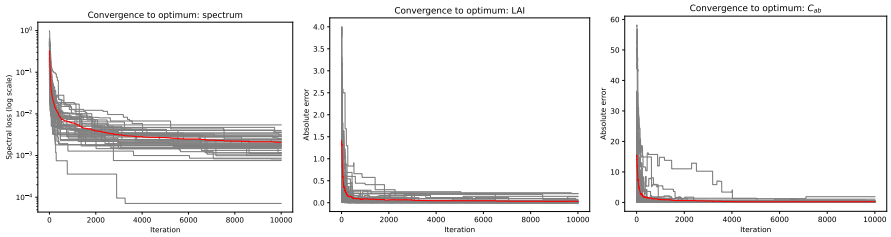
The Sentinel-2 dataset in question that we used was the SEN2-MSI-T cloud removal dataset from Chapter 4. While any Sentinel-2 Level-2A-based dataset would work, this dataset offered a convenient mix of scale and diversity. The dataset consists of 5 land cover classes, each split into a geographically diverse set of 4 scenes, resulting in 20 total locations. Every scene contained a cloud-free observation at 5 time steps within a period of 6 months, resulting in a total of 100 geospatially- and temporally diverse images. From each image, we took the centre pixel as the representative spectrum for the image (using multiple pixels from the same image could have biased the evaluation to the specific images of the dataset). Therefore, this dataset enabled us to evaluate on a curated, diverse set of 100 real-world instances and check if the PROSAIL inversion loss landscapes were still well-posed.

B

ADDITIONAL RESULTS

B.1. PROSAIL INVERSION

B.1.1. OPTIMISATION CONVERGENCE



(a) Spectral loss function (b) LAI retrieval loss (c) C_{ab} retrieval loss

Figure B.1: Visualisation of convergence to a stable solution based on the spectral loss function (B.1a), the retrieval loss for LAI (B.1b), and the retrieval loss for C_{ab} (B.1c). In these plots, every gray line represents a randomly sampled instance from D, and the red line represents the mean of these runs. Out of the free parameters in our experiments, LAI took longer than other parameters to converge, but still did so within the budget limit. On average (the red lines), the parameters converged to stable values within 1000 iterations, with later iterations only slightly improving the spectral loss further.

We performed this additional experiment to verify that the function evaluation budget used in our experiments is sufficient to converge to a stable optimum, both in terms of the spectral loss value and the parameters of the configuration $\boldsymbol{\theta}$. We performed our optimisation approach on 100 random instances, and plotted the loss values for the spectrum (which the optimisation algorithm uses to perform

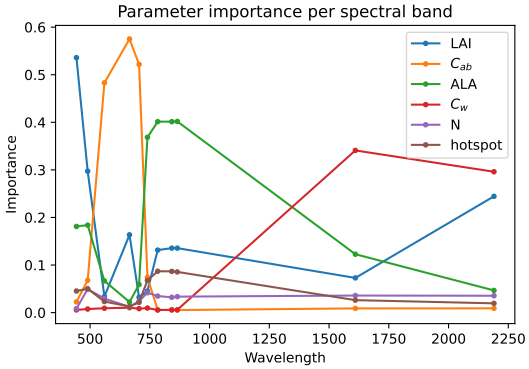


Figure B.2: Relative importance of different parameters for different wavelengths. Every point on a line represents the average importance of the parameter of that line in determining the MAE for a particular spectral band's wavelength.

optimisation) and for the retrieved parameters in the configuration (which the algorithm does not have access to, and is plotted for evaluation purposes). A very low rate of improvement for later iterations would indicate that the budget is sufficient to converge to a stable optimum.

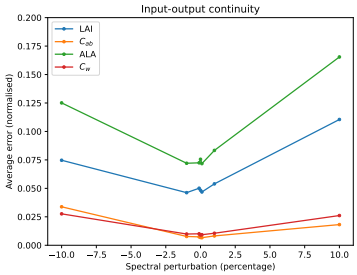
The results of this experiment can be found in Figure B.1. As the figure shows, the function evaluation budget allotted to the optimisation algorithm is sufficient to converge to stable values on average, and a convergence of the spectral loss (Figure B.1a) corresponds to a convergence of the retrieval losses of individual parameters (Figures B.1b and B.1c). As a result, our experimental setup appears to be well suited to answer our research questions.

B.1.2. PARAMETER IMPORTANCE PER BAND

In addition to our main parameter importance experiment, we computed the importance of 6 of the globally most important parameters at every Sentinel-2 band, and plotted this is Figure B.2. As can be seen in the figure, the relative importance of the parameters can vary greatly between spectral bands. Therefore, parameters with a relatively low global importance may still be relatively easily retrievable, due to the sensitivity of the loss landscape to these parameters in local, specialised parts of the parameter space. All parameters from our selection have a part of the spectrum where they have the highest impact (LAI for ultraviolet and blue, chlorophyll for green, red and near-infrared, leaf angle for short-wave infrared, and leaf water content for infrared).

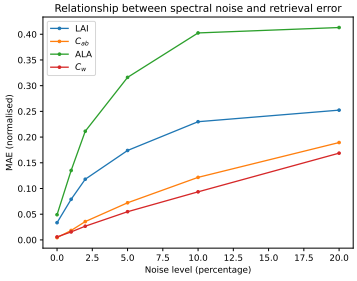
B.1.3. PROSAIL INVERSION SAM RESULTS

These supplementary figures contain the results for our experiments for the SAM loss function, corresponding to the same analyses we provide in Section 5.5 for the PMAE loss function. We have moved these figures to the supplementary material because their patterns largely conformed to those for PMAE. Figure B.3 contains results for Experiment 2, Figure B.4 contains results for Experiment 3, Table B.1 contains results for Experiment 4, and Table B.2 contains results for Experiment 5.



(a) Shift per perturbation

Figure B.3: The continuity of the output for PROSAIL inversion (predicted configuration $\hat{\boldsymbol{\theta}}$) with respect to perturbations to the input (spectrum), aggregated over all 1000 instances and normalised to a 0-1 range based on the bounds of the parameter range. Unlike in the PMAE results, the best solution for no perturbation (0 on the x-axis) did not have a near-zero error rate for LAI and ALA; this suggests that SAM may not be an appropriate choice as an optimisation loss function.



(a) Performance per noise level

Figure B.4: The impact of spectral noise on retrieval performance, aggregating the 'shifted optimum' phenomenon over all 1000 instances, showing that it is a consistent pattern, and the intensity of the shifts increases as the noise level increases.

Parameter	Normalised MAE target			
	$\alpha_1\boldsymbol{\theta}_1^+ + \alpha_2\boldsymbol{\theta}_2^+ + \alpha_3\boldsymbol{\theta}_3^+$	$\boldsymbol{\theta}_1^+$	$\boldsymbol{\theta}_2^+$	$\boldsymbol{\theta}_3^+$
LAI	0.136 ± 0.167	0.199 ± 0.182	0.203 ± 0.187	0.206 ± 0.19
C_{ab}	0.057 ± 0.058	0.128 ± 0.113	0.121 ± 0.109	0.123 ± 0.11
ALA	0.24 ± 0.172	0.296 ± 0.233	0.299 ± 0.241	0.307 ± 0.233
C_w	0.043 ± 0.053	0.082 ± 0.069	0.084 ± 0.072	0.087 ± 0.073

Table B.1: Results for E4 on the impact of spectral mixing. Every cell represents the (normalised) MAE between the optima found for the mixed spectrum $\mathbf{x}'$ and the quantities listed in the columns. The first column represents the weighted mean of the true configurations of the constituent spectra $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$, while the other columns represent the MAE compared to these individual constituent configurations. This suggests that the solution for mixed spectra matches the weighted mean of the constituent configurations more closely than the configuration of any individual constituent spectrum.

Range interval	LAI prior range interval size				
	0%	**10%**	**30%**	**50%**	**100%**
LAI (uniform)	[1]0.0 ± 0.0	[2]0.473 ± 0.289	[3]1.413 ± 0.863	[4]2.214 ± 1.37	[5]2.978 ± 2.124
LAI	[1]0.0 ± 0.0	[2]0.755 ± 0.33	[3]1.606 ± 1.054	[4]1.995 ± 1.572	[5]2.151 ± 1.851
C_{ab}	[2]10.86 ± 15.61	[1]10.35 ± 15.497	[3]11.455 ± 15.626	[4]12.084 ± 15.965	[5]12.23 ± 16.092
ALA	[1]24.908 ± 22.832	[2]30.773 ± 22.476	[3]34.7 ± 23.514	[4]35.712 ± 24.424	[4]35.725 ± 24.357
C_w	[1]0.004 ± 0.007	[1]0.005 ± 0.007	[3]0.005 ± 0.008	[4]0.005 ± 0.008	[5]0.005 ± 0.008

Table B.2: Mean absolute error rates for parameter retrieval performance for the four different parameters (rows), with columns representing the interval size of a range constraint prior on LAI (with 100% covering the full original parameter range). The 'LAI (uniform)' row represents the performance of estimating LAI through uniform random sampling, while in other columns, performance is acquired through optimisation. In each row, the prior range size in a column marked with a lower number (e.g., [1]) retrieves a parameter significantly better (significance level $\alpha = 0.05$) than one with a higher number (e.g., [2]). Adding range constraint priors on LAI greatly improved LAI retrieval performance, while also improving ALA (but not C_{ab} and C_w) retrieval performance.

B.2. eMMI

B.2.1. ϵ-MANIFOLD EFFECTIVENESS WITH PRECISION AND RECALL

Dataset	ϵ-manifold		Uncertainty quantification	
	Precision	Recall	Precision	Recall
PROSAIL	1.0	1.0	1.0	0.68
PROSAIL 2D	0.94	0.38	1.0	0.48
TP	0.99	0.6	1.0	0.24
Lorenz63	0.99	0.8	0.85	0.3
GM	0.81	0.94	0.99	0.79
TM	0.96	0.98	0.87	0.56
LR	0.82	0.94	1.0	0.52

Table B.3: Results for the oracle-based ϵ-manifold validation experiment for RQ2, showing precision and recall scores for a classification task where, for every instance, the true values $\boldsymbol{\theta}^+$ had to be predicted along with a negative sample from the validation points. For these metrics, only a single score could be computed.

B.2.2. eMMI RESULTS WITH PRECISION AND RECALL

Table B.4: Precision of the different methods approximating the ϵ-manifolds, with all hyperparameters for all methods (including the appropriate eMMI variant) automatically determined through hyperparameter optimisation. For every simulator, the best performance has been marked in **boldface**, with statistical significance determined by a Wilcoxon signed-rank test at significance level $\alpha = 0.05$.

Method	PROSAIL	PROSAIL 2D	TP	Lorenz63	GM	TM	LR
RF	0.67 ± 0.46	0.57 ± 0.37	$\mathbf{0.89 \pm 0.12}$	0.51 ± 0.15	0.91 ± 0.17	0.38 ± 0.37	0.05 ± 0.22
GP	0.75 ± 0.09	0.58 ± 0.13	0.51 ± 0.03	0.5 ± 0.0	$\mathbf{1.0 \pm 0.01}$	0.5 ± 0.0	0.5 ± 0.0
BNN	0.51 ± 0.09	0.67 ± 0.15	0.51 ± 0.03	0.5 ± 0.0	0.58 ± 0.13	0.5 ± 0.0	0.5 ± 0.0
ABCSMC	0.34 ± 0.31	0.37 ± 0.32	0.38 ± 0.32	0.37 ± 0.24	0.27 ± 0.38	0.32 ± 0.33	0.42 ± 0.48
TabPFN	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
eMMI	$\mathbf{0.84 \pm 0.1}$	$\mathbf{0.86 \pm 0.12}$	0.9 ± 0.12	$\mathbf{0.59 \pm 0.2}$	1.0 ± 0.0	$\mathbf{0.68 \pm 0.14}$	$\mathbf{0.96 \pm 0.1}$

Method	PROSAIL	PROSAIL 2D	TP	Lorenz63	GM	TM	LR
RF	0.09 ± 0.11	0.15 ± 0.16	0.01 ± 0.01	0.03 ± 0.02	0.84 ± 0.23	0.27 ± 0.37	0.0 ± 0.0
GP	0.67 ± 0.21	0.7 ± 0.22	$\mathbf{0.97 \pm 0.11}$	1.0 ± 0.01	0.26 ± 0.24	$\mathbf{1.0 \pm 0.0}$	1.0 ± 0.0
BNN	0.93 ± 0.2	0.85 ± 0.27	$\mathbf{0.98 \pm 0.08}$	$\mathbf{1.0 \pm 0.0}$	$\mathbf{0.85 \pm 0.29}$	$\mathbf{1.0 \pm 0.0}$	$\mathbf{1.0 \pm 0.0}$
ABCSMC	0.34 ± 0.37	0.44 ± 0.41	0.32 ± 0.33	0.35 ± 0.32	0.16 ± 0.29	0.4 ± 0.44	0.0 ± 0.0
TabPFN	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
eMMI	$\mathbf{0.95 \pm 0.19}$	$\mathbf{0.96 \pm 0.07}$	0.86 ± 0.26	0.43 ± 0.3	$\mathbf{0.95 \pm 0.14}$	0.98 ± 0.12	0.79 ± 0.25

Table B.5: Recall of the different methods approximating the ϵ-manifolds, with all hyperparameters for all methods (including the appropriate eMMI variant) automatically determined through hyperparameter optimisation. For every simulator, the best performance has been marked in **boldface**, with statistical significance determined by a Wilcoxon signed-rank test at significance level $\alpha = 0.05$.

ACKNOWLEDGEMENTS

After all the “academic we” that you, as the reader, have just endured, it seems only fitting that this short section can provide some relief in favour of a more personal touch. After four years of working as a PhD, I certainly understand now why everyone feels the need to add these acknowledgements: so much in how a PhD is experienced depends on the people who support it.

In this vein, the first people to thank are my supervisors. Mitra, you are clearly doing something right, because even after working together since 2019 I still enjoy the process, and constantly learn from you. Thank you for all the opportunities you have given me that I have gladly taken, and for being the kind of supervisor who makes academia both a safe and pleasant experience for PhD students, and a stimulating environment to learn and grow. Peter, thank you for guiding me in the scary foreign place that is “the domain”, and for the unfailingly kind and patient welcome I received in it. At times I nearly forgot I had work to do as I learned about concepts in biology and ecology, and to this day, I continue to read about these topics out of general interest. Holger, your views on AI, science and Europe are inspiring and closely match my own ideals, therefore finding very fertile soil in my brain. I suspect that my newfound hobby of following your social media posts, which bring some much-needed reality checks to the business influencer-dominated public discourse on AI, will remain a highly enjoyable passtime for the foreseeable future.

I would like to express my gratitude to everyone in the ADA research group, especially my fellow PhDs. I have learned much from you, from the excellent examples I could look up to when I joined as a Master student to the new students newly introduced to the group. In a vaguely chronological order: Jan, Koen, Marie, Anna, Can, Jeroen, Matthias, Bram, Mike, Annelot, Maedeh, Samira, Hadar, Thijs, Andreas, Khashayar, Inês, Nansheline and Sietse. A special mention should go to Julia, who was the first Master student I ever supervised (feeling woefully underqualified for the task as a fresh PhD), and not soon after, became the one colleague-PhD with whom I could jointly commiserate over the frustrations of working with EO data as an AI researcher. Also thanks to the part of ADA at RWTH Aachen, whom I always feel like I should find ways to spend more time with, and all the Master and Bachelor students who have come and gone (I’m sorry, there are just so many of you!).

Thank you to Alistair for being my guide during my research visit to ϕ-lab at

ESA, both in terms of work and in terms of cool slightly hidden vegan restaurants in the middle of Rome. Thanks also to James and Athanasia for their involvement when they were my contacts at ESA. Thanks to Juhuhn, my friend from back in our Master student days, with whom my research trajectory always seems strangely, but enjoyably, synced.

I tend to be more reserved when it comes to my private life, but the brevity of this paragraph should not be misinterpreted as it being the least important category. My work brings me satisfaction, which I appreciate a great deal. However, my happiness stems from my family and my partner Areum, which deserves more thanks than can be expressed through this public medium.

CURRICULUM VITÆ

Laurens Arp was born on 23 January 1995 in Haarlem, The Netherlands. He pursued his Bachelor studies at Vrije Universiteit Amsterdam, where he studied Lifestyle Informatics, followed by his Master studies in the Artificial Intelligence specialisation of Computer Science at Leiden University. He carried out his Master thesis with the ADA research group at Leiden Institute of Advanced Computer Science (LIACS), supervised by Dr. Mitra Baratchi, and graduated in 2020. After this, he kept working with the ADA group as a PhD candidate supervised by Dr. Mitra Baratchi and Prof.dr. Holger Hoos from LIACS, and Prof.dr. Peter van Bodegom from the Institute of Environmental Sciences (CML). During this time he also collaborated closely with research fellows from the European Space Agency (ESA), and spent time as a visiting researcher at the ϕ-lab at the European Space Research Institute (ESRIN) location of ESA in Frascati, Italy.

Laurens' interests lie in interdisciplinary work where typical machine learning assumptions are violated, particularly for relevant spatio-temporal problems such as environmental challenges, urban planning and socio-economic conditions, which often involves Earth observation data. He is currently working as a postdoctoral researcher at Leiden University, researching causal machine learning for spatio-temporal data. During his PhD studies, he took various AI courses at the European Summer School on AI (ESSAI) and Advanced Course on Artificial Intelligence (ACAI) in Ljubljana, in addition to courses on transferable skills such as project management, oral presentation skills and scientific conduct, among others.

List of Publications

Under Review

1. **Laurens Arp**, Peter van Bodegom, Nguyen Dang, Alistair Francis, Holger H. Hoos, and Mitra Baratchi. Inference from Noisy Observations through Model Inversion: Constructing ϵ-Manifolds of Potentially Valid Solutions. Under review.

2026

1. **Laurens Arp**, Peter van Bodegom, Holger H. Hoos, and Mitra Baratchi. Characterising the Ill-posedness of PROSAIL Inversion for Biophysical Paramter Retrieval. In *European Journal of Remote Sensing*, 59(1).

2024

2. **Laurens Arp**, Holger H. Hoos, Peter van Bodegom, Alistair Francis, James Wheeler, Dean van Laar, and Mitra Baratchi. 2024. Training-free thick cloud removal for Sentinel-2 imagery using value propagation interpolation. In *ISPRS Journal of Photogrammetry and Remote Sensing*, 216.

1. Julia Wąsala, Suzanne Marselis, **Laurens Arp**, Holger Hoos, Nicolas Longépé and Mitra Baratchi. 2024. AutoSR4EO: An AutoML Approach to Super-Resolution for Earth Observation Images. In *Remote Sensing*, 443.

2022

1. **Laurens Arp**, Mitra Baratchi, and Holger H. Hoos. 2022. VPint: value propagation-based spatial interpolation. In *Data Mining and Knowledge Discovery*, 36.

Included in this Thesis.

GLOSSARY

automated algorithm configuration an optimisation problem where the hyperparameters of algorithms are automatically tuned based on an objective function 23

AutoML automated machine learning: automated model selection and algorithm configuration for machine learning problems 23

band a data dimension for an EO image containing measurements for a specific variable 13

black-box optimisation a type of optimisation problem/algorithm in which only the inputs and outputs of an objective function g can be observed, with no knowledge about g itself 22

configuration a vector $\boldsymbol{\theta}$ containing concrete value assignments for the physical parameters P 1, 22

convolutional neural network a type of neural network where local spatial patterns are extracted using convolutional kernels 32

data product a dataset containing estimations for quantities of interest through the processing of EO data 13

Earth Observation data obtained through sensors observing the Earth 11

hybrid model a machine learning model trained on the inverse of simulated data produced by an RTM 20

hyperspectral spectral (optical) data containing many spectral bands 16

ill-posed a problem that does not meet the requirements of well-posedness 21

in-situ direct measurements of relevant physical parameters 12

inference the estimation of the properties that generated an observed outcome 2

Landsat satellites operated by NASA, many of which contain a spectrometer as a sensor 15

look-up table a tabular dataset generated by an RTM, containing generating parameters and simulated outcomes 20

loss landscape a surface in $d+1$-dimensional space describing the loss function value of every point in the parameter space 101

model inversion the inversion of a (simulation) model capturing the data generation process for an inference problem 19

MODIS optical satellites operated by NASA with a high temporal resolution and a low spatial resolution 15

multispectral spectral (optical) data containing multiple spectral bands, often including wavelengths outside the visible ranges 15

optical data data containing measurements of a light spectrum 15

parameter estimation an inference setting where the value of a physical parameter p must be inferred from some observed feature data $\mathbf{x}$ 2

parameter space (search) space of the parameter domain D_P containing all possible combinations of parameter values 100

physical parameter a variable p in a set of scientific variables P describing the state of a physical system 1

PROSAIL a radiative transfer model for vegetation, combining the PROSPECT and 4SAIL models 19

radiative transfer model a type of physical model that simulates a light spectrum based on physical input parameters 19

remote sensing data obtained remotely by sensors not directly interacting with the object being observed 13

search space space containing all possible combinations of parameter values, through which we must perform a search to find an optimum 22

sensor network a spatially distributed network of sensors, measuring a target variable at specific points 12

Sentinel-2 multispectral optical satellites operated by ESA 15

spatial interpolation filling in a spatial grid of missing values in between of a number of known measurements 27

spectral band a measurement of light intensity for a single wavelength on a light spectrum 15

spectrometer an optical sensor measuring light intensity at certain wavelengths 15

swath width of a remote sensing sensor passing over a study area 13

VPint our proposed spatial interpolation algorithm founded on a system-oriented perspective 33

VPint2 our proposed spatial interpolation algorithm suitable for filling in gaps in optical satellite imagery 66