Physics-based deep representation learning of vegetation using optical satellite image time series

Yoël Zérah

RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

Université de Toulouse

Doctorat de l'Université de Toulouse

préparé à l'Université Toulouse III - Paul Sabatier

Apprentissage profond de représentations physiques de la végétation à partir de séries temporelles d'images satellite optiques.

Thèse présentée et soutenue, le 20 juin 2024 par

Yoël ZÉRAH

École doctorale
SDU2E - Sciences de l'Univers, de l'Environnement et de l'Espace

Spécialité
Surfaces et interfaces continentales, Hydrologie

Unité de recherche
CESBIO - Centre d'Etudes Spatiales de la BIOsphère

Thèse dirigée par
Jordi INGLADA et Silvia VALERO

Composition du jury
M. Thomas OBERLIN, Président, ISAE-SUPEARO
M. Devis TUIA, Rapporteur, Ecole Polytechnique Fédérale de Lausanne
M. Gustau CAMPS-VALLS, Rapporteur, Universitat de València
M. Jan Dirk WEGNER, Examinateur, Universität Zürich
M. Jordi INGLADA, Directeur de thèse, CNES
Mme Silvia VALERO, Co-directrice de thèse, Université Toulouse III - Paul Sabatier

Membres invités
Mme Marie WEISS, INRAE Provence-Alpes-Côte d'Azur

Abstract

Human-driven climate change is triggering unprecedented and dire transformations of ecosystems and habitats worldwide. Remote sensing offers precious tools for monitoring the state of the Earth, and for understanding how the biosphere functions and is affected by human activities. Satellite remote sensing capabilities and data processing techniques have rapidly improved over the last decades, and have considerably advanced the study of life processes on land masses. The advent of modern machine learning and exponential development of computational power are crucial for the exploitation of the vast amount of data produced by remote sensors. In particular, the Sentinel-2 (S2) mission has been providing high spatial and temporal resolution multi-spectral images at a global scale, for nearly a decade. S2 products are released with an open-data policy that supports research efforts and various applications, such as the enhancement of agricultural practices, land management and disaster response. Remote sensing data is a measurement of incoming radiation and its properties are related to the nature of elements and processes on the surface of the Earth. Extracting useful representations that contain relevant information is fundamental for applications of remote sensing.

The objective of this thesis is to find useful representations from remote sensing data for use in downstream applications. There are several challenges in the retrieval of such representations. First, in order to be useful to different tasks, the representations need to be general and interpretable. This can be achieved with bio-physical variables that characterize the target system, for instance the water and mineral content in the soil, the pigment concentrations, the canopy structure and the temporal evolution of vegetation. Also, remote sensing data has an intrinsic uncertainty, and representations of this data should be associated with a measure of uncertainty. Another challenge lies in the scarcity of reference data in remote sensing. Although remote sensing measurements are big data, it is difficult to obtain the corresponding ground truth data. For instance databases of vegetation bio-physical parameters that can be related to remote sensing measurements are rare. Methods that attempt to retrieve such parameters therefore commonly resort to physical modeling and inversion. This Ph.D. is divided into three main parts, which are associated with its four main contributions. Its first contribution is the identification of a key issue of supervised regression models that perform model inversion. Their performance is shown to be very dependent on the choice of the sampling distribution for simulating their training data-sets. The second contribution of this Ph.D. is the development of a self-supervised approach for retrieving physical representations of remote sensing data. This approach is based on the framework of Variational Autoencoders, and relies on the incorporation of a physical model and physical knowledge in a deep learning framework. Instead of attempting to optimize the physical variable retrieval from an unavailable ground truth or a biased simulated reference, this method uses input data reconstruction as a proxy task. Finally, in a third part, this thesis reports the results of the application of the proposed approach on the retrieval of physical variables in two settings. In a first experiment, it is used with the PROSAIL radiative transfer model for retrieving leaf characteristics and canopy structure variables. The resulting PROSAIL-VAE model is trained directly using S2 multi-spectral images. Validation with in-situ data have corroborated the performance of the approach. In a second application, the proposed approach is used to retrieve phenological variables that characterize the temporal behavior of vegetation. The so-called Pheno-VAE is trained on annual NDVI time series

extracted from S2 data.

Keywords Earth Observation, Artificial Intelligence, Vegetation Monitoring, Representation Learning, Self-Supervised Learning, Model Inversion, Physical Modeling, Stochastic Modeling, Variational Autoencoders.

Résumé

Le changement climatique initié par les activités humaines provoque des transformations drastiques et sans précédent des écosystèmes et des zones habitées dans le monde entier. La télédétection s'impose comme un outil essentiel pour observer la Terre, pour comprendre le fonctionnement de la biosphère ainsi que son altération par les pressions anthropiques. Les capacités d'observation par télédétection spatiale ainsi que les techniques de traitement du signal ont rapidement évolué lors des dernières décennies. L'émergence des techniques d'apprentissage statistique modernes et l'augmentation exponentielle de la puissance de calcul disponible sont cruciaux dans l'exploitation de l'immense volume de données produit par les capteurs de télédétection. En particulier, la mission S2 produit des images multi-spectrales à haute résolution spatiale et temporelle depuis une dizaine d'années à une échelle globale, diffusées gratuitement avec une politique d'accès libre. Les produits S2 ont permis le développement de diverses applications, telles que l'amélioration des techniques agricoles, la gestion du territoire et la réponse aux catastrophes naturelles. Les données de télédétection sont des mesures de radiations électromagnétiques dont les caractéristiques sont reliées à la nature des éléments et aux processus à la surface de la Terre. L'extraction de représentations contenant des informations pertinentes sur ces éléments est fondamentale pour les applications de télédétection.

L'objectif de cette thèse est de proposer une méthode d'inférence de telles représentations à partir de données de télédétection. Plusieurs défis se présentent pour estimer ces représentations. D'abord, elles doivent êtres générales et interprétables, afin d'être utilisables par plusieurs applications. Cela peut être réalisé avec des variables bio-physiques qui caractérisent les systèmes observés, par exemple le contenu minéral et en eau des sols ou la concentration en pigments et la structure de la canopée pour la végétation, ainsi que son évolution temporelle. Par ailleurs, les représentations doivent êtres associées à une incertitudes d'estimation. Le manque de données de référence pose aussi un défi. Contrairement aux acquisitions de télédétection, il est difficile d'obtenir des vérités terrain. Les bases de données qui associent des variables bio-physiques de la végétation et des données de télédétection sont rares. Les approches qui estiment ces variables utilisent donc la modélisation physique et l'inversion. Cette thèse est divisée en trois parties principales qui détaillent ses quatre contributions. La première contribution est la démonstration de la dépendance des modèles de régression supervisée au choix de la distribution d'échantillonnage pour leur jeu de données d'entraînement. La seconde contribution est le développement d'une méthodologie d'estimation de variables physiques non supervisée à partir de données de télédétection, basée sur les Autoencodeurs Variationnels (VAE). Cela consiste en l'incorporation de modèles et de connaissances physiques à priori dans un modèle d'apprentissage profond. Cette approche utilise la reconstruction comme tâche intermédiaire pour estimer une variable physique, plutôt que la comparaison avec une vérité terrain indisponible ou une référence simulée. Dans une troisième partie, ce manuscrit détaille les deux autres contributions de cette thèse : l'application de la méthodologie proposée à l'estimation de variables physiques dans deux applications. Dans la première le modèle de transfert radiatif PROSAIL est utilisé dans le modèle PROSAIL-VAE afin d'estimer les caractéristiques de feuilles et de la canopée à partir d'images S2. La validation avec des données in-situ a permis de confirmer les performances de cette approche. Dans la seconde application, des variables phénologiques caractérisant

le comportement temporel de la végétation sont estimées à partir de séries temporelles de NDVI, avec le modèle Phéno-VAE.

Mots-clés Observation de la Terre, Intelligence Artificielle, Suivi de la Végétation, Apprentissage de Représentations, Apprentissage Auto Supervisé, Inversion de Modèle, Modèles physiques, Modélisation Stochastique, Autoencodeurs Variationels.

Remerciements

Une thèse ne se résume pas à un apprenti-chercheur en tête-à-tête avec un sujet pendant trois ans et plus si affinités. Pour se former, s'impliquer et contribuer à la communauté scientifique, le doctorant que j'ai été a bénéficié de l'appui et de l'encouragement d'un grand nombre de personnes. Je veux ainsi témoigner ici de ma gratitude et de ma reconnaissance envers celles et ceux qui ont rendu la réussite de mon doctorat possible.

Je réserve mes premiers remerciements à mes encadrants de thèse, qui ont suivi de près mon travail et ma progression: programmation, rédactions, présentations, articles, conférences et manuscrit. Je les remercie pour leur patience, leur indispensable soutien moral durant ces trois années, pour m'avoir aidé à prendre confiance en la valeur de la méthode et les résultats obtenus, et m'avoir rendu fier du travail accompli. Merci à Silvia Valero, co-directrice de cette thèse, qui m'a décidé à tenter l'aventure, et qui m'a enseigné les ficelles de la recherche, avec son expertise dans l'art de la « destruction » — l'écriture itérative d'un article lors de laquelle les passages nécessairement médiocres rédigés par le scientifique néophyte seront dynamités à grand coup de surligneur, puis recomposés par son encadrant rompu à l'exercice. Merci à Jordi Inglada, co-directeur de cette thèse, pour les heures passées ensemble devant des codes python, et le tableau blanc à dessiner des diagrammes à boite et se demander « mais au fond, c'est quoi un VAE ? ». J'admire sa considérable culture scientifique et informatique. D'ailleurs, il m'a inspiré à réessayer emacs un jour. Peut-être.

Je remercie également tous les membres de mon jury de thèse pour avoir évalué mon travail. Je remercie en particulier les deux rapporteurs de ce manuscript, Gustau Camp-Valls et Devis Tuia, pour leurs commentaires et suggestions pertinentes qui m'ont donnés de nouvelles perspectives sur mon travail. Je remercie également Marie Weiss qui a permis d'améliorer significativement la qualité de ce travail, grâce à nos discussions au court de ce doctorat.

Je remercie mes camarades doctorantes Valentine Bellet et Iris Dumeur pour m'avoir souvent montré la voie, et pour avoir partagé séminaires, rires et nœuds GPUs sur le HPC du CNES.

Je remercie les membres de l'équipe IA du CESBIO, Mathieu Fauvel, Julien Michel, Juan Vinasco, Katya Kalinicheva, pour leur aide — relecture d'article, préparation de données, conseils — et pour nos réunions du vendredi après-midi lors desquelles nous partagions bien plus que nos travaux hebdomadaires: réflexions, astuces et état de convergence de modèle en direct.

Je remercie Olivier Hagolle pour m'avoir expliqué longuement la physique de l'acquisition avec Sentinel-2.

Je remercie tous mes camarades et collègues au CESBIO, qui ont fait du laboratoire un environnement où j'ai pu m'épanouir, et dont les déjeuners et les sorties partagées ont ponctué cette thèse de nombreuses notes de joie.

Par ailleurs, je remercie Nicolas Dobigeon, directeur de la chaire ANITI dans laquelle ma recherche s'est inscrite pour sa confiance, et pour m'avoir également permis de découvrir l'enseignement à l'INP-ENSEEIHT.

Je remercie Paul Templier, camarade doctorant de l'autre côté de l'avenue, pour nos débats d'optimisation et réseaux de neurones autour de verres occasionnels. Ceci dit, non, je ne rajouterai pas d'algorithme évolutionnaire dans ma méthode.

J'exprime finalement toute ma gratitude envers ma famille. D'abord, envers mes parents qui m'ont toujours soutenu dans mes projets, et pour qui, je suis définitivement resté un étudiant, et leur perpétuel « alors tu la passes quand cette thèse ? » ayant motivé a enfin y mettre un point final.

Merci à Praline et Bagheera, mes deux chats pour leur participation à chaque rédaction, la première m'aidant à taper chaque texTkuykkkkkkkkkkkkyugr&è_ç, et la seconde, déléguée aux pauses ba-balle.

Enfin, je remercie et embrasse fort Alba, ma chérie, mon pilier de cette réussite pour son soutien sans failles et son affection, et me conforte dans mon avenir scientifique.

Contents

List of Figures

List of Tables

General introduction

Context

Earth observation (EO) capabilities have vastly improved over the last decades. New remote sensing satellites have been deployed with improved spatial resolution and with increased revisit frequency, and have been monitoring the Earth with unprecedented scale and precision. In parallel, the fast development of both computational power and processing techniques have supported the exploitation of the vast remote sensing data and new applications. In particular, optical remote sensing data has enabled to monitor land vegetation globally, by benefiting from multi-spectral measurements with high revisit frequencies.

In many applications that employ remote sensing data, the information is commonly used with end-to-end processing pipelines, that directly take the measurements produced by the remote sensors as input. However, the useful information required for those applications are usually not remote sensing measurements, but transformation of these measurements that represent the properties of the ground. The development of such down-stream tasks could be improved and facilitated if access to useful representations were provided. Deep learning approaches can provide tools for extracting representations from remote sensing data. Furthermore, general and interpretable representations could be used by different applications, and lessen the computation required on the user side.

Physical properties of ground surfaces, such as the state of the soil, and of vegetation are interesting representations for down-stream tasks. They are interpretable, generalizable, and can be considered a product by themselves. Finding physical representations from remote measurement data is an estimation problem. However too little reference data is usually available for building algorithms that retrieve the desired ground properties at the required scale. Physical models that relate satellite observations to ground properties can be used to mitigate this issue, effectively transforming the physical variable estimation problem into an inversion problem.

Physical models are not perfect and remote sensing data has an intrinsic error, since it is a physical measurement. As such it is necessary to quantify the uncertainties associated with the estimated physical variables. The Bayesian theoretical framework provides tools to incorporate uncertainties into the estimation process.

This Ph.D. proposes to learn interpretable and probabilistic representations of vegetation from optical remote sensing data, by introducing physical models and knowledge into the optimization of a deep learning model.

Contributions

This manuscript presents the four main contributions of this Ph.D. thesis:

- The first contribution is the identification of a crucial limitation in supervised deep learning regression approaches when training on a simulated data-set. Supervised deep learning methods that learn to retrieve physical variables from remote sensing data perform model inversion, because reference data is too scarce. These approaches are trained on data-sets simulated by the physical model to invert. It was shown in this

Ph.D. that the choice of the sampling distribution for generating these data-sets is crucial to the estimation performances. This is a problem because such distributions are usually unknown and must be postulated.

- The second contribution of this Ph.D. is methodological. A framework based on variational autoencoders (VAE) for retrieving physical variables as latent representations has been developed. It is based on the incorporation of a physical model and prior knowledge in a deep-learning architecture. The inversion of the model is performed through a representation learning approach. Crucially, this deep learning approach is self-supervised: it doesn't require simulated data-sets and can be trained directly on remote sensing data.
- In a third contribution, the proposed methodology is applied to the retrieval of vegetation bio-physical parameters, with the inversion of the PROSAIL radiative transfer model. The subsequent PROSAIL-VAE model is trained directly on S2 multi-spectral images. Validation with in-situ vegetation variable measurements corroborates the performance of the developed approach compared to classical regression methods. Additionally, the identification of meaningful correlations between biophysical variables is investigated.
- In the fourth contribution, the proposed methodology is analyzed with another application: the retrieval of vegetation phenology from spectral index time series by inverting a double-logistic phenological model. Order relationships between variables were incorporated into the method to enforce physical constraints. The subsequent Pheno-VAE is trained directly on S2 normalized difference vegetation index (NDVI) time series, and shows good prediction performance.

Outline of the thesis

This Ph.D. thesis is organized as follows:

- Part I: introduces the notion of representation of remote sensing data. Chapter 1 discusses the notion of representation and how satellite data represent physical realities of observed ground scenes. Chapter 2 presents the S2 optical imagery used throughout this manuscript, and the in-situ measurement data of vegetation properties, used as a reference for validating the proposed approach.
- Part II: reviews the classical inversion approaches used in remote sensing. Chapter 3 presents the notion of model inversion, along with classical methods with an emphasis on deep learning. Chapter 4 explains the widely used PROSAIL model and details the differentiable implementation developed in this Ph.D. The first contribution of this Ph.D. is introduced in Chapter 5.
- Part III: corresponds to the second contribution. The stochastic modeling and variational inference framework are introduced in Chapter 6. In Chapter 7 is proposed a methodology for incorporating physical models into a VAE for self-supervised probabilistic inference.
- Part IV: presents the two last contributions, that is the results of the application of the proposed methodology in two different settings. Chapter 8 presents the application of the proposed method to the inversion of the PROSAIL model and its evaluation with in-situ data. In Chapter 9, the proposed methodology is used for inverting a phenological model.
- Part V: concludes this thesis. A general conclusion and perspectives are provided in Chapter 10.

- Part VI: contains the appendices. To ensure reproducibility of the research, the data, repositories and computing environment used for the experiments in this Ph.D are provided in Appendix A. Appendix B, Appendix C and Appendix D provide mathematical background. Appendix E presents additional results for Chapter 5, Chapter 8 and Chapter 9. Finally, Appendix F, Appendix G and Appendix H respectively provide a glossary, the list of acronyms used throughout this Ph.D., and the list of notations.

Support

This Ph.D. was supervised by Silvia Valero and Jordi inglada. It was carried out in Toulouse at the Centre d'études spatiales de la biosphère (CESBIO) laboratory, which is a joint research unit of the the Centre national d'études spatiales (CNES), the Centre national de la recherche scientifique (CNRS), Institut de recherche pour le développement (IRD), the Université Toulouse III (UT3) and the Institut national de recherche pour l'agriculture, l'alimentation et l'environnement (INRAe). This Ph.D. was co-funded by the CNES and by Région Occitanie. This work was supported by the Artificial and Natural Intelligence Toulouse Institute (ANITI) from the Université Fédérale Toulouse Midi-Pyrénées under Grant Agreement ANR-19-P3IA-0004. This Ph.D. is part of the ANITI Chair "Fusion-based inference from heterogeneous data" held by Nicolas Dobigeon. Data and computational resources such as the high performance computing (HPC) infrastructure were provided by the CNES.

Scientific productions

Peer-reviewed publications in international journals

- Y. Zérah, S. Valero, and J. Inglada. Physics-constrained deep learning for biophysical parameter retrieval from sentinel-2 images: Inversion of the prosail model. *Remote Sensing of Environment*, 312:114309, 2024. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2024.114309. URL https://www.sciencedirect.com/science/article/pii/S0034425724003274

- Y. Zérah, S. Valero, and J. Inglada. Physics-driven probabilistic deep learning for the inversion of physical models with application to phenological parameter retrieval from satellite times series. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–23, 2023b. doi: 10.1109/TGRS.2023.3284992

Conference communication

- Y. Zérah, S. Valero, and J. Inglada. Méthodes probabilistes d'apprentissage profond avec a priori physiques de représentations interprétables. In *GRETSI 2022: XXVIIIème Colloque Francophone de Traitement du Signal et des Images*, Nancy, France, Sept. 2022a. URL https://hal.science/hal-04186427

Oral presentations

- Y. Zérah, S. Valero, and J. Inglada. Méthodes probabilistes d'apprentissage profond avec a priori physiques de représentations interprétables, Sept. 2022b

- Y. Zérah, S. Valero, and J. Inglada. Interpretable representation learning for high resolution satellite image time series, May 2022c

Dissemination of the Ph.D. work

- Seminar "Met&Eau" at Météo France, December 2023 (abstract and oral presentation): "Inversion Bayésienne de modèles physiques par apprentissage profond, application à l'estimation de variables biophysiques de la végétation à partir d'images optiques Sentinel-2".
- Ph.D. student day of CESBIO laboratory (abstract and oral presentation, January 2021, January 2022 and January 2023).
- Ph.D. student day of doctoral school Sciences de l'Univers, de l'Environnement et de l'Espace (SDU2E) in June 2022 (Oral presentation).
- Ph.D. student day at CNES in October 2022 (Oral presentation and poster).
- Seminar of ANITI's evaluation in September 2022 (poster: "Hybrid Bayesian models for the analysis of big Earth observation data").
- ANITI afterwork in June 2022 (Oral presentation).
- Ph.D. student day of artificial intelligence (AI) students of Observatoire Midi Pyrénées (OMP) laboratories in May 2022 (Oral presentation).

Science popularization

- Oral presentation for "La nuit européenne des chercheur.es", with the session "Les Chercheur.es passent le Grand Oral" on September 29th 2022 at the Toulouse Museum. Presentation of the challenges of remote sensing with high school students in the manner of the "Grand Oral", an oral examination for the Baccalaureate.
- Animation of the "Climate fresk" with bachelor university students at UT3 in September 2021. It is a collaborative game designed for raising awareness about the causes, mechanisms and consequences of climate change.

Introduction en français

Contexte

Les capacités d'observation de la Terre ont été grandement améliorées ces dernières décennies. De nouveaux satellites de télédétection avec de meilleures résolution spatiale et fréquences de revisite ont été déployés, et permettent de surveiller la Terre à une échelle et précision sans précédents. En parallèle, le développement exponentiel de la puissance de calcul ainsi que des techniques de traitement du signal ont permis l'exploitation de grandes quantités de données de télédétection et l'émergence de nouvelles applications. En particulier, les données de télédétection optiques permettent le suivi de la végétations terrestre à l'échelle globale, bénéficiant de mesures multi-spectrales.

Dans de nombreuses applications qui utilisent des données de télédétection, le signal est souvent traité de bout en bout dans des chaînes de traitement qui prennent directement les mesures des capteurs de télédétection comme entrée. Cependant, l'information utile pour ces applications est rarement directement la mesure de télédétection, mais plutôt des transformations de ces mesures qui représentent des propriétés au sol. Le développement de ces applications en aval de la mesure pourrait être amélioré et facilité si des représentations utiles de la données étaient accessibles. Les approches par *apprentissage profond*[1] comportent des techniques permettant l'extraction de telles représentations à partir de données de télédétection. De plus, des représentations interprétables et généralisables pourraient être utilisées par plusieurs applications, plutôt que de calculer des représentations spécifiques pour chaque application a partir des données de télédétection. Cela permettrait de diminuer le coût de traitement des données pour un utilisateur et de faciliter l'accès à une information pertinente tout en diminuant l'expertise en télédétection nécessaire à l'utilisation.

Les propriétés physiques des surfaces au sol, telles que l'état du sol et de la végétation sont des représentations utiles pour des applications en aval. Ces représentations sont interprétables, généralisables et peuvent être considérées comme un produit avec une valeur en soi. Déduire des représentations physiques à partir de données de télédétection est un problème d'estimation. Cependant, trop peu de données de référence sont généralement disponibles pour optimiser des algorithmes d'estimation de ces propriétés physiques à grande échelle. Ce problème est généralement contourné en utilisant des modèles physiques qui relient des propriétés physiques au sol avec des observations de télédétection, ce qui transforme le problème d'estimation en un problème d'inversion de modèle.

Les modèles physiques ne sont pas parfaits, et les données de télédétection ont une erreur intrinsèque, car ce sont des mesures. Par conséquent il est nécessaire de quantifier l'incertitude associée aux variables physiques estimées. Le cadre théorique bayésien permet notamment d'incorporer une mesure d'incertitude aux problèmes d'estimation.

Cette thèse propose de développer des méthodes d'apprentissage de représentations interprétables et probabilistes à partir de données de télédétection optique, en intégrant des modèles physiques et des connaissances a priori au sein de modèles d'apprentissage profond.

[1] Deep learning.

Contributions

Quatre contributions principales sont présentées dans cette thèse :

- La première contribution est l'identification d'une limitation cruciale des approches de régression supervisée par apprentissage profond, à partir d'un jeu de données simulé. Les méthodes d'apprentissage profond qui sont entraînées à estimer des variables physiques à partir de données de télédétection réalisent en pratique de l'inversion de modèle car les données de référence ne sont pas disponibles en quantité suffisante. Ces approches sont entraînées sur des jeux de données simulés avec le modèle physique à inverser. Il a été montré dans cette thèse que les performances d'inversion sont très sensibles au choix de la distribution d'échantillonnage permettant de générer ces jeux de données. Cela pose problème car ces distributions sont en général mal connues et doivent être postulées.

- La seconde contribution de cette thèse est méthodologique, avec le développement d'une approche d'estimation de variables physiques basées sur les *Autoencodeurs Variationnels* (VAE)[2]. Cette approche propose d'incorporer un modèle physique ainsi que de l'information a priori au sein d'une architecture d'apprentissage profond. Ainsi, l'inversion du modèle physique est réalisée à travers une approche d'apprentissage de représentations: les variables estimées correspondent à une représentation latente. Cette approche d'apprentissage profond est auto-supervisée et ne nécessite pas de jeux de données simulés. Elle peut être entraînée directement sur des données de télédétection réelles.

- Une troisième contribution est l'application de la méthode proposée pour l'estimation de paramètres bio-physiques de la végétation, avec l'inversion du modèle de transfert radiatif PROSAIL. Le modèle PROSAIL-VAE qui réalise cette inversion est entraîné directement sur des images multi-spectrales Sentinel-2 (S2). La performance de cette approche a été comparée à d'autres méthodes d'inversion par apprentissage profond supervisé classiques grâce à des données de validation terrain. Par ailleurs, les corrélations entre les variables bio-physiques estimées ont été étudiées.

- La quatrième contribution est l'application de la méthode proposée pour l'estimation de variables phénologiques de la végétation par l'inversion d'un modèle phénologique double-logistique. Des relations d'ordre ont été imposées sur les variables latentes afin de garantir des contraintes physiques. Le modèle Pheno-VAE qui en découle est entraîné directement sur des séries temporelles de NDVI[3] de S2, et montre de bonnes performances de prédiction.

Plan de la thèse

Ce manuscrit est organisé comme suit:

- Partie I: Cette partie introduit la notion de représentation de données de télédétection satellite. Dans le Chapitre 1 est étudiée la notion de représentation, ainsi que la manière dont les données satellites représentent des réalités physiques des surfaces au sol observées. Le Chapitre 2 présente l'imagerie optique S2 utilisée tout au long de ce manuscrit, ainsi que les mesures terrain de propriétés de la végétation qui sont utilisées comme référence pour valider l'approche proposée.

- Partie II: Cette partie liste les approches classiques d'inversion de modèle utilisées en télédétection. Le Chapitre 3 présente la notion d'inversion de modèle, ainsi que les

[2] Variational autoencoders (VAE).
[3] Indice de différence normalisée de la végétation, normalized difference vegetation index (NDVI).

méthodes classiques en mettant l'accent sur les méthodes d'apprentissage profond. Le Chapitre 4 détaille le modèle PROSAIL, ainsi que l'implémentation différenciable de ce modèle développée lors de cette thèse. La première contribution de cette thèse est exposée au Chapitre 5.

- Partie III: Cette partie correspond à la seconde contribution de cette thèse. Les modèles stochastiques et l'inférence variationnelle sont introduits au Chapitre 6. Dans le chapitre Chapitre 7 est proposée une méthodologie d'incorporation de modèles physiques en tant que décodeur d'un VAE, permettant une inférence probabiliste auto-supervisée.

- Partie IV: Cette partie présente les deux autres contributions de cette thèse, c'est à dire les résultats d'application de la méthode proposée dans deux cas. Le Chapitre 8 présente les résultats d'application de la méthode à l'inversion du modèle PROSAIL, et son évaluation à l'aide de données terrain. Dans le Chapitre 9, la méthodologie proposée est utilisée pour l'inversion d'un modèle phénologique.

- Partie V: Cette partie conclu la thèse. Une conclusion générale ainsi que des perspectives de recherche sont détaillées au Chapitre 10.

- Partie VI: cette partie contient les annexes. Dans un soucis de recherche reproductible, les liens vers les dépôts de données et de code sont fournis en Annexe A ainsi que l'environnement de calcul utilisé pour les expériences dans cette thèse. Des formulaires et preuves mathématiques sont fournis en Annexe B, Annexe C et Annexe D. Des résultats complémentaires aux Chapitre 5, Chapitre 8 and Chapitre 9 sont fournis en Annexe E. Enfin, l'Annexe F, l'Annexe G et l'Annexe H contiennent respectivement un glossaire, la liste des acronymes utilisés tout au long du manuscrit et la liste des notations.

Financement

Cette thèse a été dirigée par Silvia Valero et Jordi Inglada. Elle a été réalisée à Toulouse au Centre d'études spatiales de la biosphère (CESBIO), qui est une Unité Mixte de Recherche (UMR) du Centre national d'études spatiales (CNES), du Centre national de la recherche scientifique (CNRS), de l'Institut de recherche pour le développement (IRD), de l'Université Toulouse III (UT3) et de l'Institut national de recherche pour l'agriculture, l'alimentation et l'environnement (INRAe). Cette thèse a été cofinancée par le CNES et la région Occitanie. Cette thèse s'inscrit dans le cadre du projet Artificial and Natural Intelligence Toulouse Institute (ANITI) de l'Université Fédérale Toulouse Midi-Pyrénées (ANR-19-P3IA-0004). Ce travail a été soutenu par le projet ANR-JCJC DeepChange (ANR-20-CE23-0003). Cette thèse est rattachée à la chaire ANITI « Inférence basée sur la fusion de données hétérogènes » dirigée par Nicolas Dobigeon. Les données ainsi que les ressources de calcul informatiques telles que l'infrastructure du high performance computing (HPC) ont été fournis par le CNES.

Productions scientifiques

Publications dans des revues internationales à comité de lecture.

- Y. Zérah, S. Valero, and J. Inglada. Physics-constrained deep learning for biophysical parameter retrieval from sentinel-2 images: Inversion of the prosail model. *Remote Sensing of Environment*, 312:114309, 2024. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2024.114309. URL https://www.sciencedirect.com/science/article/pii/S0034425724003274

- Y. Zérah, S. Valero, and J. Inglada. Physics-driven probabilistic deep learning for the inversion of physical models with application to phenological parameter retrieval from satellite times series. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–23, 2023b. doi: 10.1109/TGRS.2023.3284992

Communication de conférence

- Y. Zérah, S. Valero, and J. Inglada. Méthodes probabilistes d'apprentissage profond avec a priori physiques de représentations interprétables. In *GRETSI 2022: XXVIIIème Colloque Francophone de Traitement du Signal et des Images*, Nancy, France, Sept. 2022a. URL https://hal.science/hal-04186427

Présentations orales

- Y. Zérah, S. Valero, and J. Inglada. Méthodes probabilistes d'apprentissage profond avec a priori physiques de représentations interprétables, Sept. 2022b
- Y. Zérah, S. Valero, and J. Inglada. Interpretable representation learning for high resolution satellite image time series, May 2022c

Dissemination des travaux de thèse

- Séminaire « Met&Eau » at Météo France, December 2023 (abstract et présentation orale): « Inversion Bayésienne de modèles physiques par apprentissage profond, application à l'estimation de variables biophysiques de la végétation à partir d'images optiques Sentinel-2 ».
- Journées des doctorant du CESBIO (abstract et présentation orale, January 2021, January 2022 and January 2023).
- Journée des doctorants de l'école doctorale Sciences de l'Univers, de l'Environnement et de l'Espace (SDU2E), Juin 2022 (présentation orale).
- Journée des jeunes chercheurs du CNES in October 2022 (présentation orale et poster).
- Séminaire de l'évaluation d'ANITI, Septembre 2022 (poster: « Hybrid Bayesian models for the analysis of big Earth observation data »).
- ANITI afterwork in June 2022 (présentation orale).
- Journée des doctorants des étudiants en artificial intelligence (AI) de l'Observatoire Midi Pyrénées (OMP), Mai 2022 (présentation orale).

Médiation scientifique

- Présentation orale pour « La nuit européenne des chercheur.es », avec l'animation « Les Chercheur.es passent le Grand Oral » le 29 Septembre 2022 au Museum de Toulouse. Présentation des défis de la télédétection avec des lycéens à la manière du « Grand Oral », un examen oral du Baccalauréat.
- Animation de la « Fresque du climat » auprès d'étudiants de licence à l'UT3 en Septembre 2021. Il s'agit d'un jeu collaboratif conçu pour sensibiliser aux causes, méchanismes et conséquences du réchauffement climatique.

Part I

Introduction: representations of remote sensing data

Chapter 1

Land surface representation with satellite imagery

Contents

1.1 Introduction

The Earth has never been more monitored than today. The atmosphere, the oceans and the continental surfaces are being constantly scrutinized by the modern spaceborne fleet of Earth observation (EO) satellites, that reached 1192 active units in orbit around the planet by the end of 2023, and is still growing. With instrument technology improving, observations are carried out at increased spatial resolutions. As a result, remote sensing satellites produce a tremendous amount of data that contain essential information. Besides commercial, humanitarian and military uses, satellite data is instrumental to environmental sciences such as ecology, hydrology, geology, and climate science, which has gained more awareness in the last few years, as the climate crisis has been increasingly affecting societies. Satellite data is crucial in understanding and monitoring climate change effects. Guiding decisions and actions with remote sensed information could help mitigation and adaptation efforts.

Remote sensing data are physical measurements, that indirectly relate to physical processes and phenomena that occur on the ground (notwithstanding atmospheric interactions). All EO satellite sensors rely on measures of electromagnetic radiation that emanate from the Earth, from different parts of the spectrum depending on the detector. The Earth doesn't emit much photons on its own, aside from the weak blackbody thermal radiation that is emitted in the long wavelength infra-red (LWIR) domain, radio waves output by human communication systems and visible light pollution from urban areas. Measuring radiation intensity from the Earth in most parts of the spectrum requires an external source for illumination. In such cases, the radiation being measured remotely is one that has been reflected by the Earth. Remote sensing can be divided into two categories, *active* and *passive*. Active sensors include the electromagnetic source onboard the satellite platform, and measures the radiation reflected back to the satellite after being sent toward the target scene and interacting with it[1]. Among active remote sensors are radar, operating with microwaves and radio waves, and LiDaR, that sends laser pulses towards the ground. Passive sensing relies on natural illumination of the Earth from the Sun. Imaging in the visible and infra-red (IR) spectrum is mostly performed with passive sensors.

The remote measurements performed by different sensors each probe different domains of the Earth system. They each operate at a specific spectral, spatial and temporal regimes, both in terms of range and resolutions. Aside from spectral specifications of different sensors briefly discussed above, the orbiting satellite platforms are restricted to observe a limited number of scenes on Earth with a given resolution, and with a given frequency dictated by revisit times. As such, each of these sensors offer their limited point of view of the Earth systems, and they attempt to describe reality from their perspective. Remote sensing data is a *representation* of the Earth surface physical properties. Satellite images are representations of the ground surface.

Representing the Earth surface from remote measurements is imperfect. The electromagnetic signal being sensed interacts with both the atmosphere and the ground surface. This signal itself is thus not a perfect representation of the surface. The detection process itself is affected by noise, since the physical device isn't perfect, and parasite signals can be observed alongside the "true" signal. These representations are also partial, since each sensor has a limited resolution and range. The Earth as a whole is a very complex system that cannot be grasped holistically. The domain of the Earth realm accessed by a given sensor is limited by which surface materials are observed and how. Vegetation is particularly accessible in the visible [Tucker, 1979] and IR spectrum, whereas soil moisture can be better characterized in the micro-wave or radio domains [Prévot et al., 1993]. Some spectral domains of the ground surface are in fact hardly observable, since atmospheric water absorbs all signal within

[1]The amount of reflected radiation that is sensed by a remote sensor is function of the ground surface reflectance.

specific *spectral bands*. Moreover, since remote sensing is intrinsically about the interaction of light with Earth systems, it doesn't directly reflect their nature. Therefore, actual properties of interest, are not directly accessible from this data, they are hidden, *latent*. Satellite data is almost never used as-is. It must be first transformed into other, more useful representations of Earth properties. This brings about key questions. What are "good" representations that enable to grasp ground processes and nature? How can these representations be extracted from remote sensing measurements?

What is a "good" representation is inherently subjective, since it is application-dependent. From the perspective of a downstream task, the best possible representation of the Earth surface is the one that straightforwardly provides all relevant information for the given task. For instance, in a scenario of a response to a natural disaster, such as storms, landslides, earthquakes, etc., an ideal representation of the ground situation would directly identify affected settlements, or even prospective search zones for survivors [Gueguen et al., 2017]. For agriculture, a number of representations could enable better crop management [Bégué et al., 2018]. Early detection of crop sickness or parasites could help minimize yield loss and phytosanitary products spreading. Mapping hydric stress could improve irrigation practices [Arun and Karnieli, 2022; Tolomio and Casa, 2020]. Representations can be common to several applications: hydric stress could also be an input to forestry to prevent and mitigate wildfires [Balzter et al., 2007]. In the case of ecosystem studies, representations may also simply be input for scientific models.

Satellite remote sensing data is faced with its own challenges. It is high dimensional, has high variability, and is non-linearly linked to ground processes. Other constraints depend on the type of sensor. For instance optical remote sensing is affected by clouds and atmospheric state, which leads to missing data both spatially and temporally. Conversely synthetic aperture radar (SAR) imagery has no such data availability outage, but requires more pre-processing before use, such as speckle noise correction. Overall, collecting reference ground data is difficult, since the data gathering process most often cannot be automated on the required scale. It can be designated as a "big data low label" regime.

When extracting representations from remote sensing data, the data from different satellite sources are generally used independently. Data fusion between different sensors, which is the joint use of data from different sources, can be difficult to perform. This is because of heterogeneous data produced by sensors with different specifications (e.g. spectral domain, spatial resolution), or because the observation conditions are different (e.g. different observation times, different viewing angles). Nonetheless, combining measurements from different sensors could provide complementary inputs to produce better representations. For instance, optical imagery characterizes vegetation well, whereas radar is more sensitive to water contents. Furthermore, data processing for a given application is tailored to the associated sensor, which makes it less flexible.

Machine Learning (ML) methods[2] have emerged in the last decade as essential tools to handle remote sensing data. ML designates the set of computational methods to automatically extract information from data, and perform algorithm optimization to make new predictions, instead of being explicitly programmed, i.e. to *learn* (or are *trained*). These methods have been successfully used in a wide variety of applications with great performance, including remote sensing. Deep learning (DL), which are ML methods based on deep artificial neural networks (ANNs), offer a flexible and powerful framework to infer representations from data. DL models are highly parallelizable and scalable, which makes them especially adapted to large scale predictions with remote sensing data. The most straightforward forward optimization of a DL model is through *supervised* training which involves providing pairs of input data with desired output data (e.g. data-sets of image-label pairs for classification). However,

[2] ML is frequently put as part of the broader field of study that is AI, that attempts to enable machines to think, reason and act, in the way humans do — nonetheless, both terms are often employed interchangeably. AI is perhaps a more "fashionable" expression, since current technology cannot yet be truly qualified as "intelligent".

this approach is dependent on the availability of labeled data, which can be a problem for certain EO applications where ground truth data is not available in sufficient quantity. This is why there is potential in methods that lower or eliminate the requirement for labeled input data, such as *unsupervised* learning, *weakly-supervised* learning or *semi-supervised* learning. Among these methods, *self-supervised* learning uses auto-generated reference data to perform training.

ML methods such as DL can generate representations for downstream tasks from input remote sensing data as predictions. However, since each application usually requires its own specific representation, this usually amounts to designing end-to-end computing pipelines, which takes satellite images as input and produce representations as outputs. This is computationally expensive, and requires know-how to handle remote sensing data, which can hamper application development. Furthermore, remote sensing big-data processing consumes a non-negligible amount of energy, along with manufacturing, delivering and operating modern material computing infrastructures additionally is resource-hungry, thus lowering the amount of computation required is crucial to contribute to reducing the global footprint of EO. This is why there is a need to derive *intermediary representations* that make relevant ground information more readily available for downstream tasks than raw images. These representations need to be general enough to encompass different needs. Also, abstract representations (e.g. lossless compression encoding) are not suited for downstream task users. As such, enabling interpretability (explainability) of representations is crucial. Since representations are of physical measurements, they have to be matched with physical concepts. Within ML, this can be achieved by incorporating physical information and constraints as prior knowledge during training. Furthermore, it is fundamentally impossible to truly grasp reality phenomena. Representations are not perfect, prone to noise and reflect a partial, flawed knowledge. Therefore, there is a need to associate a measure of uncertainty to derived representations.

1.2 Mapping with satellite imagery

Space-borne remote sensing is about monitoring the Earth surface properties. The measurements produced by the sensors are transformed into digital signals organized into arrays, i.e. digital *images*. Each *pixel* of these images is associated with a certain area on the ground, the scale of which is quantified by a *spatial resolution*. Optical remote sensors perform measurements of electromagnetic signal in one or several *spectral bands*, so each pixel has one or several channels dedicated to these spectral bands. The number of these bands defines a *spectral resolution*. These measurements are quantized, and thus are characterized with a *radiometric resolution*. Similar concepts can be formulated with non optical measurements.

Maps are the natural visual representation of ground properties retrieved from remote sensing measurements, since the related data, images, are already arranged to reflect spatial phenomena. Producing *cartography* from these representations, is about associating them to a spatial coordinate, to a location on the ground. Depending on the range of available data, maps can have vastly different ground footprints, from a few meters to a global scale. Maps are an important product of remote sensing, and have been realized since the early days of EO satellites. In the following subsection 1.2.1, subsection 1.2.2 and subsection 1.2.3 are discussed some classical representations used for cartography, whereas in subsection 1.2.4 is introduced the modern concept of *digital twin*, that aims at a more general and complete type of representation than cartography.

1.2.1 Spectral indices

Manual interpretation of optical images is the traditional way to identify and describe elements of observed scenes. This analysis is based on the exploitation of features of the images: the

color, the tone, the texture, the shape, the location, and the context [Green, 2000]. The automated analysis of optical imagery is based on the processing of those features. The advent of high spatial resolution imaging has naturally improved the ability to characterize ground elements. Also, multi-spectral measurements enable to identify vegetation better, such as with a relatively high intensity of measured reflectance in some spectral bands (green in the visible spectrum, and in near infra-red (NIR)). However, it is challenging to characterize vegetation with precision, because remote measurement themselves are not directly interpretable. Specifically, it is difficult to link directly measurements (i.e. reflectances) to particular aspects of ground elements (e.g. the type of vegetation, its density, is health, etc.), the relevant information is hidden within the measured signal: remote measurements are not a good representation for characterizing the land surfaces.

Combining reflectances in different spectral bands together into a single quantity called a *spectral index* enables to derive some meaningful representation. In particular, for vegetation, the NDVI is arguably the most widely used spectral index, and is one the first invented [Rouse et al., 1974]. It is defined as a combination of red (R) and NIR band reflectances ρ:

$$\text{NDVI} = \frac{\rho_{\text{NIR}} - \rho_{\text{R}}}{\rho_{\text{NIR}} + \rho_{\text{R}}}. \tag{1.1}$$

NDVI takes values inside $[-1, 1]$, with positive values highly correlated to the density of photosynthetic vegetation. Negative values generally indicate a water surface, whereas near 0 the NDVI is characteristic of bare soils.

The NDVI enables to distinguish spatially different vegetation better than individual red and NIR bands, and as such can be considered a "better representation" for vegetation (see Figure 1.1).

Since it characterizes the density of vegetation, considering the NDVI from a temporal perspective instead of a spatial perspective also enables useful representations. By collecting the temporal evolution of NDVI for pixels containing vegetation, it enables to obtain information about their *phenology*[3] (see Figure 1.2).

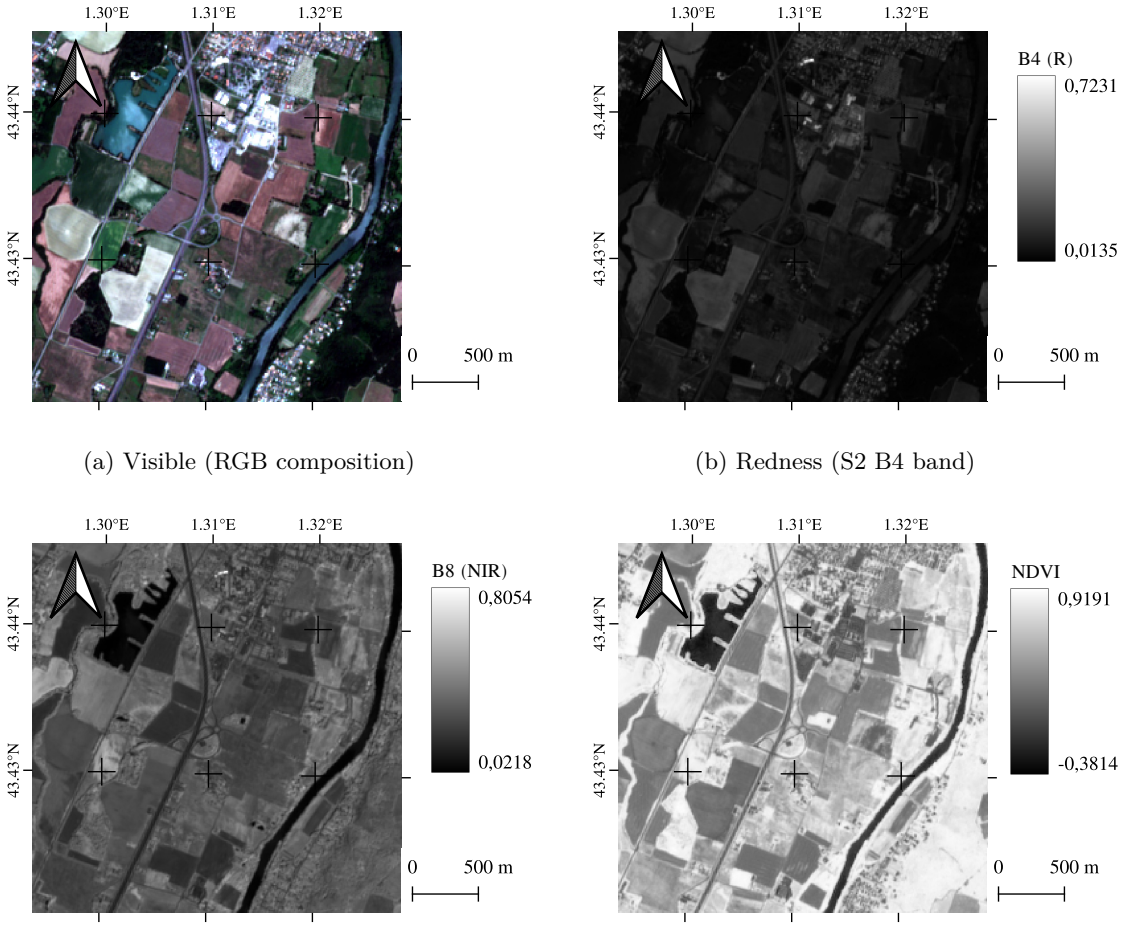
Other spectral indices enable to extract other meaningful representations of vegetation. There are a number of indices that are modifications of the NDVI that compensate for potential atmospheric effects [Gitelson et al., 1996; Huete et al., 2002], or soil effects [Huete, 1988], or focus in the red-edge spectral region [Gitelson and Merzlyak, 1994]. There are actually hundreds of spectral indices [Henrich et al., 2009; Loaiza et al., 2023; Montero et al., 2023] suited for various different sensors and applications. Although vegetation indices make up the majority of the spectral indices, other are suited for representing other types of ground surfaces: bare soil [Nguyen et al., 2021], water [Ma et al., 2019], snow [Hall and Riggs, 2010], burnt areas [Epting et al., 2005], urban areas [Javed et al., 2021].

It can be noted that extracting spectral indices depends on available measured spectral bands for a given optical sensor. As such, the spectral indices may have different values across sensors because of differences in detector specifications, such as the definition of upstream spectral bands. Therefore the exact same representation can usually not be achieved across sensors using spectral indices.

1.2.2 Land cover and land use mapping

Another way to represent the physical reality of Earth surface is to associate each element (e.g. an object or a surface) with a type, a class. For land areas, land use and land cover (LULC) maps are visual representations of the spatial distribution of different classes of land use and land cover in a given area over a limited time frame. Land cover classes mostly refer to the type of physical material, ecosystems that occupy a given area (forest, water, urban area, crops, bare soil), whereas land use classes refer to a socio-economic functional description

[3] Vegetation phenology is the cyclic, seasonal progression of plant status through typical stages: dormancy, active growth and senescence.



(a) Visible (RGB composition)

(b) Redness (S2 B4 band)

(c) NIR (S2 B8 band)

(d) NDVI

Figure 1.1: Comparison between red reflectance, NIR reflectance and NDVI derived from a S2 image of the outskirts of Toulouse, France (2023-06-24). NDVI highlights significantly woody area with high values, and water bodies such as the Garonne river with negative values.

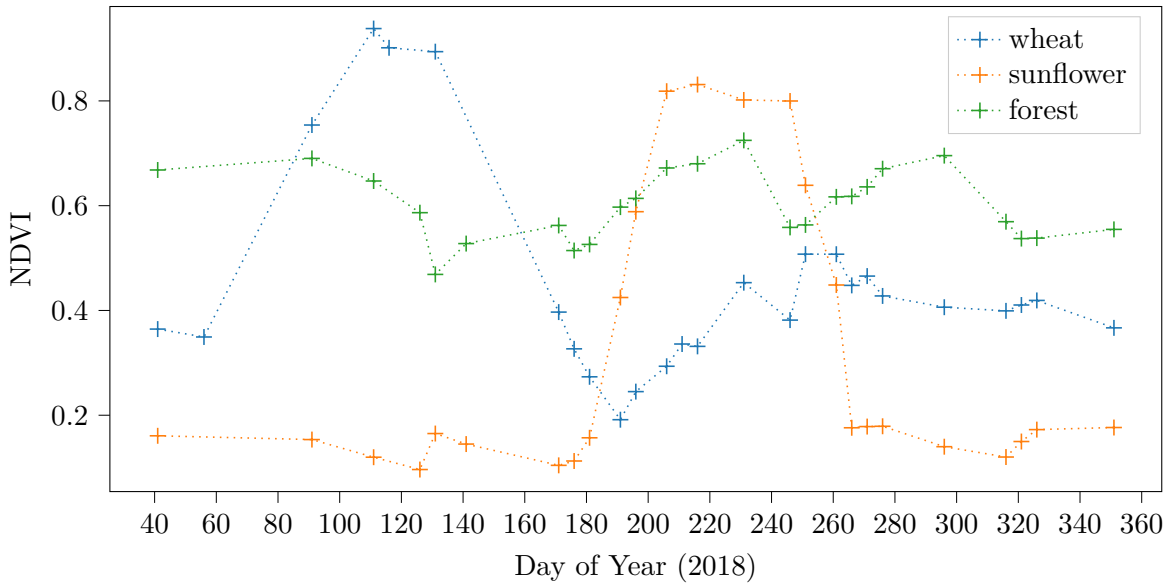


Figure 1.2: NDVI time series derived from S2 pixels containing different vegetation types. Wheat and sunflower crops exhibit a characteristic growing season followed by decay, whereas the forest NDVI doesn't vary much.

(residential, commercial, industrial area, agriculture, etc.). The different classes used within a LULC map are pre-defined within a *nomenclature*, that may be more or less detailed and that can have hierarchical levels. Production of LULC maps is about classifying ground elements into a LULC nomenclature, either unit ground surface area[4] (usually matching the pixel grid of remote sensing images), or objects (in which case it is first necessary to detect said objects and estimate their ground footprint).

Modern production of LULC maps makes use of high resolution optical imagery Rogan and Chen [2004], such as produced by the S2 mission, Phiri et al. [2020]. Linking a specific semantics nomenclature to reflectances is not as straightforward as computing spectral indices, it involves more complex processing. The recent rise of ML techniques has enabled to tune classifiers using satellite images and reference data to produce LULC maps. Since a single-date satellite image of a location usually isn't enough to accurately determine the LULC classes, using multi-temporal data has been the focus of much research. In particular, S2 optical imagery has a relatively high temporal frequency, enabling to build Satellite image time series (SITS) from observations, as a spectral, spatial and temporal representation. ML classification for LULC maps production has been performed in particular with algorithms such as random forests (RF) [Inglada et al., 2017b; Pelletier et al., 2016a], DL [Ienco et al., 2019; Miller et al., 2024; Stoian et al., 2019], and Gaussian processes (GP) [Bellet et al., 2023, 2024].

The CES[5] OSO[6] is a team made of experts from CESBIO, UMR Ispa, Dynafor, CNRM, UMR Tetis, IGN–Matis, Costel and Sertit. It has developed ML LULC classification algorithms based on RF, and have produced land cover maps of metropolitan France yearly since 2016, based on a 23 classes nomenclature since 2018: the OSO land cover map (see Figure 1.3).

LULC maps are nomenclature-dependent, making it difficult to compare between different maps. It can be noted nonetheless that there have been efforts to unify those representations, by performing *translation* of a given LULC nomenclature into another [Baudoux et al., 2023].

1.2.3 Biophysical variable mapping

Although useful, a LULC map is a rather coarse representation of land properties. Since it is categorical, it doesn't allow quantitative comparison between instances of the same class. In particular, it doesn't enable to characterize nuances on the state of classified elements, e.g. whether a forest is healthy or withering, whether an urban area is damaged, whether crop yield can be expected to be low or high. Conversely, spectral indices, that enable quantitative comparison, are not directly linked to desired interpretable properties on the ground.

As such, there is a need for mapping other quantities, variables that are more directly linked to more precise properties of the land elements. These variables can be broadly referred as *bio/geo-physical variables*. Relevant biophysical variables at a given location depend on the type of object present (e.g. LULC), e.g. variables of vegetation may be unsuitable for describing non-permeable areas of urban sprawl. Identifying these variables also depend on the considered application. For instance, the Global Climate Observing System (GCOS) identified biophysical variables as key indicators that are critical to characterize climate, the essential climate variables (ECVs) [GCOS, 2011; Bojinski et al., 2014]. So far, 55 atmosphere, land and ocean variables have been listed as ECVs, that are related to the

[4] Associating a class to each pixel of an image is referred as *classification* in remote sensing and as *semantic segmentation* in computer vision.

[5] From French: Centre d'expertise scientifique.

[6] From French: Occupation des sols.

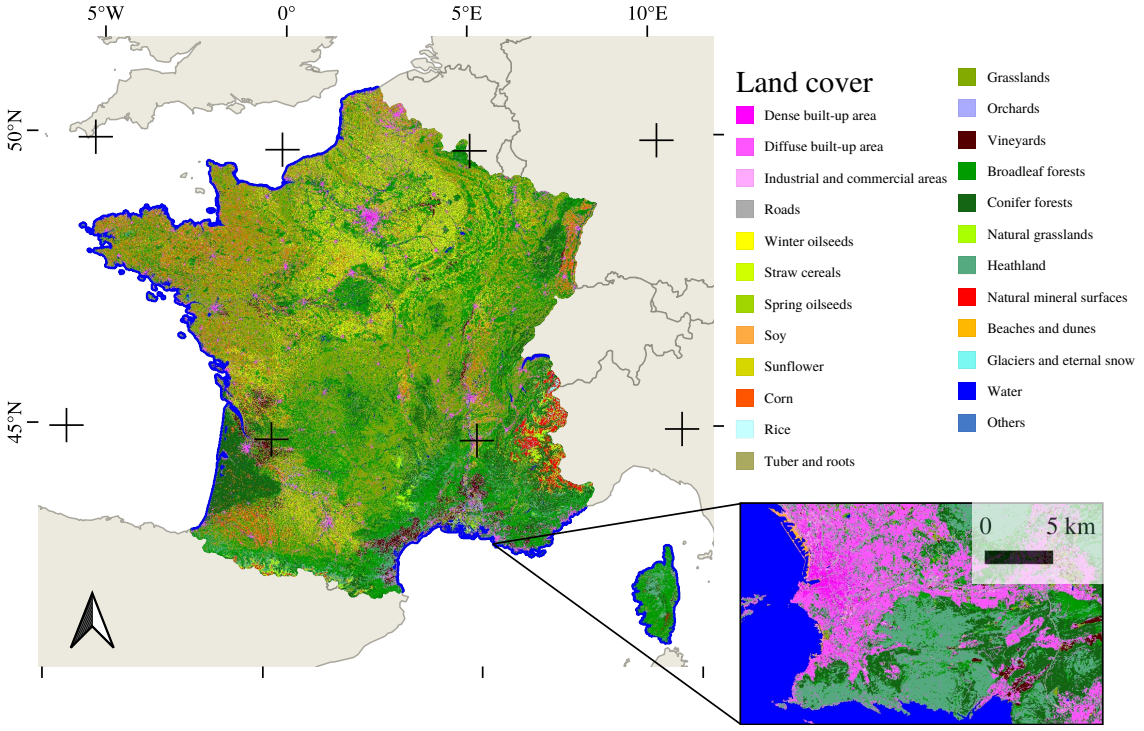


Figure 1.3: OSO land cover map 2022.

state of the hydrosphere[7], the cryosphere[8], the antroposphere[9] and the biosphere[10]. An ECV may encompass several related quantities. For instance the fire ECV contains the burnt area, the active fires, and the fire radiative power. Incidentally, the land cover is listed as a biosphere ECV. ECVs measurements have requirements, in terms of resolution (spatial and temporal), of the data delivery delay (timeliness), instrument error and drift (stability) and uncertainty [Zemp et al., 2022].

Similarly, essential ocean variables (EOVs) have been defined by the Global Ocean Observing System (GOOS), as indicators of the oceans status. A majority of EOVs are also ECVs. The group on Earth observation (GEO) biodiversity observation network (BON) defined essential biodiversity variables (EBVs) that characterize genetic composition, species population and traits, community composition, ecosystems functioning and structure [Pereira et al., 2013]. EBV are complementary to ECV. It is worth noting that the goal EBV representation is that of an "EBV cube", i.e. data with a temporal, spatial dimensions and a biodiversity component dimension, highlighting a need to monitor a spatio-temporal evolution. The concept of "essential variables", as *representations* of the status of the environment, has been used by different scientific and policymakers groups which developed various sets of biophysical variables according to their mandate and objectives. Besides ECV, EOV and EBV, other essential variables have been proposed, often as complements: essential marine ecosystem variables [Hayes et al., 2015], essential variables for invasion monitoring [Latombe et al., 2017], essential sustainable development goals variables [Reyers et al., 2017], essential geodiversity variables essential agriculture variables [Whitcraft et al., 2019] and [Schrodt et al., 2019].

[7]The hydrosphere refers to the whole of water on Earth, namely oceans, freshwater, surface water, groundwater, glacial water, and atmospheric water vapor.

[8]The cryosphere includes the components of the Earth System at and below the land and ocean surface that are frozen, including snow cover, glaciers, ice sheets, ice shelves, icebergs, sea ice, lake ice, river ice, permafrost, and seasonally frozen ground, and solid precipitation

[9]The anthroposphere encompasses the total human presence throughout the Earth system including our culture, technology, built environment, and associated activities.

[10]The biosphere refers to the whole of places on Earth where living beings exist.

1.2.4 Earth digital twin

Most representations reflect a limited set of properties of their target. As such they can be insufficient to grasp the target as a whole, they can fail at representing interconnected systems that constitute the target. With the advent of modern computer-aided simulation techniques, the concept of *digital twin* has emerged, and has been identified as a major trend in the last decade. Digital twins can be defined as computer-based models that simulate, emulate and mirror the state of a physical entity, that can be an object, a process, a human, or a human-related feature [Barricelli et al., 2019]. Digital twins aim at encompassing all relevant elements of their target, sometimes even promoting a near-bijection between them and the modeled object. This concept has been notably first applied in the industrial and manufacturing sectors, in particular in the aerospace field. They are being developed to forecast and monitor the life cycle of complex objects.

Digital twins differ from CAD[11]/CAE[12] software that represent device parts and simulate their behavior in different cases (external forces, vibrations, thermal variations, etc.). Digital twins are characterized by an extensive use of descriptive data about the target object, that is exchanged and updated as frequently and continuously as possible, ideally in real-time. This has been made possible by the development and massive deployment of various sensors and measurement devices in all sectors. Digital twins are meant to be ultra-realistic computerized counterparts of a given object, and rely on measurement input to keep it up to date and make relevant predictions.

The concept of digital twin is currently envisioned for EO, thanks to the large and increasing number of remote sensors. Destination Earth (DestinE) is an initiative launched in 2022 by the European Union (EU) that is promoted by the European Space Agency (ESA) that aims at developing a digital twin for the Earth, or *Digital Twin Earth* by 2030 [Nativi et al., 2021]. NASA's Earth System Digital Twins (ESDT) is the equivalent American initiative.

The Earth being a very complex system made of many interconnected parts, many current digital twin for earth initiatives focus on building a digital twin on a given sub-system of the planet, such as hydrological cycles [Brocca et al., 2024], or forests [Buonocore et al., 2022].

Digital twins of the Earth strive for assimilating as many data sources as possible, remote sensing products, ground measurements, even crowd sourcing [Mazumdar et al., 2017]. They aim at enabling large scale monitoring of the Earth at fine spatial and temporal resolutions, and enhance simulation possibilities of different scenarios and help decision-making.

1.3 Scientific representations

Defining and characterizing representations is an ongoing active philosophical debate. *Theories of representations* are proposed, and their implications discussed ontologically[13] and epistemologically[14]. A general definition of representation is that it is *something* (a source) that can be used on behalf of *something else* (a target) to reason, to think, to make predictions. Images are visual representations of a scene. A map is a visual representation of the spatial repartition of objects. Art is a representation that can convey emotions, ideas. Red green blue (RGB) encoding is a representation of the perceived level of "redness", "greenness" and

[11] Computer-aided design.

[12] Computer-aided engineering.

[13] Ontology is the metaphysical study of the nature of being, of reality itself, for which there are two main opposing perspectives: *realism* hypothesizes that reality exists independently from consciousness whereas *idealism* argues that some aspects of reality depends on mental constructs.

[14] Epistemology is the metaphysical study of the nature of knowledge, how we know about it and what is the relation of knowledge with reality. One of its main questions is about the role and existence of *a prior knowledge* (independent of experience and the fruit of pure reason) and *a posteriori knowledge* (that is derived from experience). *Rationalism* emphasizes the importance of the former whereas *empiricism* values the importance of the latter.

"blueness" of pixels in a digital image. One of the key representational questions is: in virtue of what is a given source a representation of a given target ?

A straightforward, perhaps simplistic stance on that question is that of the *stipulative fiat*[15]: Any source can represent any target as long as it is *stipulated*. During family dinners with unavoidable geopolitical debates, a salt shaker may be placed next to some large plate to represent the African east coast and Madagascar. One may choose to represent a animals (e.g. cows) as a spheres with constant density to approximate thermal losses of mammals, while blatantly neglecting actual physical properties. Of course, any source being selectable to represent any target doesn't make it necessarily a "good" representation, especially for scientific purposes. Nonetheless, it enables to highlight two key aspects of representations: they are always associated with a *context* and an *intent*. For instance, a digital image file is an encoded visual representation, however without context nor intent (i.e. suitable software and hardware as context, and the intention of people who designed them) this file just exists in reality as a specific distribution of electric charges within semi-conductors. When context and intent aren't explicit, representations are not well defined and subject to multiple interpretations: what does the Mona Lisa painting represent?

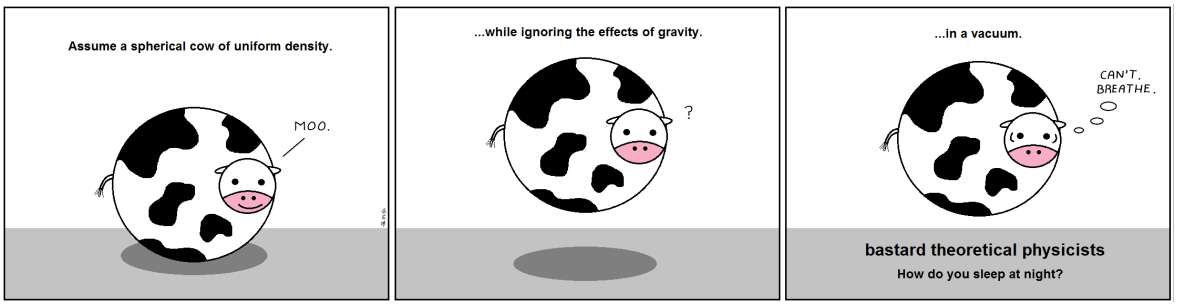


Figure 1.4: Assume a spherical cow - Credit: Abstruse Goose

Representations are also ubiquitous in science, since they enable to grasp and describe the physical reality. *Scientific representations*, or epistemic representations (ERs) are debated as a subject *per se*, they ought to have some additional properties that set them apart from other representations, say art. For instance, they differ by their sources (model, theories, data) and targets (real-world systems, theoretical objects). The debate around those representations is centered on key questions such as the *constitution question* of ER [Callender and Cohen, 2006]: "In virtue of what is there representation between scientific sources and their targets?". Providing a review of the different positions and currents of the active debate on theories of ER, with sufficient details and nuance is an enterprise out-of-the scope of this Ph.D. A general overview of the debate of ER can be found in Frigg and Nguyen [2021]. Nonetheless, the following subsection 1.3.1 shall attempt to introduce some perspective in which the notion of representation used in this manuscript is set. Then subsection 1.3.2 will specify the notion of representation and model within the field of DL.

1.3.1 Scientific models and representations

As Hughes [1997] phrases, "the concept of representation is as slippery as that of a model". This is because models are, in fact, representations themselves. *Scientific models* can be defined as physical (e.g. model ships or planes, model wings in a wind tunnel, sensor structural models on electro-dynamic shakers) and/or mathematical (atmospheric model, radiative transfer model (RTM)) and/or conceptual representations of systems of ideas, events or processes. A key distinction of models as special representations is that models are tools that enable thinking and reasoning, they allow to test scientific hypotheses.

[15]The stipulative fiat is part of General Griceanism that claims that all representations are derived from more fundamental *mental representations*.

Scientific models are notoriously used to make predictions about their target systems, e.g. models of the solar system enable to predict the position of celestial bodies at a given moment and enable to design the trajectory of space probes.

Not all representations in science are models, e.g. diagrams and drawings such as Figure 1.4 are representations that also enable scientific discussion without being models. Measurements are a ubiquitous scientific representations, that are not models. The notion of measurement is debated just like for models and representations. Overall, *measuring* is a procedure that correlates some properties or attributes (characterized by a *unit*) of a target system to a number, the *measurement*. Measurement is not only about the attribution of a number, but also about the procedure that provides this number. The measurement procedure itself integrates models about the sensor. For instance, measuring length with a ruler uses the underlying model of size as a function of the distance between graduations. Surface reflectances, presented as a measurement from optical remote sensors, must be related to the flux of photons that reaches the detector, requiring models of the atmosphere, of the onboard optics, of the detector itself, etc. This last example illustrates that the distinction between measurement and *estimation* can be blurred.

It is finally important to distinguish models and *theories*. Both concepts are often used interchangeably, since models often have a theoretical content and theories are often expressed by models. However, models are representations, whereas theories are conceptual frameworks, sets of ideas, that aim to explain reality. Models can be though of as "*instantiations*[16] of theories, narrower in scope and often more concrete, commonly applied to a particular aspect of a given theory, providing a more local description or understanding of a phenomenon" [Fried, 2020]. Models usually use simplifying assumptions from their parent theory, so that they can be used in practice. The spherical cow model humored in Figure 1.4 is a caricature of oversimplification in scientific models, that can be useful nonetheless: modeling animals as balls of flesh enables analytical facilities to study heat transfer and show that mammals below a certain size cannot exist because compensating heat loss would require unattainable caloric intake.

In science there are two main ways to build models: *first principle modeling* and *data-driven modeling*. First-principle models are the direct instantiation of a theory, they represent the fundamental assumptions of a target system, e.g. they are the result of the application of physical laws to a particular situation. For instance RTM simulate the radiation flux that propagates through a medium, while relying on the theory of optics and electromagnetism (see Chapter 4). On the other hand, data-driven, *statistical or empiric* models, seek to establish relationships between different components of the target system by using measurements. First principle modeling and statistical modeling are not mutually exclusive. The use of the theories underlying the target system can help statistical models achieve better representations, by introducing *biases* (see Chapter 6 and Chapter 7). Conversely, oftentimes an empiric model of a system is first derived from observations, and then the retrieved empirical relationships are formalized into first principle models, and integrated into a broader theory. One famous example of this is the description of elliptical orbits by Kepler's law of planetary motion, which were not first established through a theory of gravitation, but rather from model-fitting of observations of time and angle of celestial bodies.

Regression is an approach for fitting a flexible statistical model to a particular data (see Chapter 3). Regression models can be very simple, for instance linear or affine relationships can be established from a reduced set of data-points: the Ohm law was found by observing that the electrical tension was proportional to the current, the proportionality coefficient being the resistance; similarly, Beer-Lambert's law is a linear relationship between a solution concentration and the proportion of light that propagates through, and the transmittance relating the two. Such models may be valid within restricted regimes of the target systems.

[16] An instance is a case, an application, of a *property* (or a *class* in programming), e.g. 42 is an instance of numbers, the spherical cow model is an instance of the theory of heat transfer theory.

More data-points and more complex models become necessary to describe the target more accurately. In such case, the ML approaches can be relevant to represent the relationships of the target's components. In ML, a statistical model is automatically optimized to fit data. Among these methods, DL is based on a particular class of models: deep neural networks.

1.3.2 Deep learning and representations

In the DL field, the notion of representation and model is actually quite practical and straightforward, although not often explicitly defined. A DL model is an algorithm that transforms input data, more specifically an ANN. An ANN is organized into layers that apply consecutive non-linear transformations to the data. The transformations of data outputted by each individual given layer are can be understood as hidden internal variables of a DL model. These internal transformations and the output of the DL model are seen as representations of the model input data [Rumelhart et al., 1986]. The individual components of the input data, internal transformations and output data of the model are called *features*, and features of a given layer form a representation of the input data [Bengio et al., 2013]. An important aspect about DL representations, is that they are measurable, since features are either numerical or categorical. DL is notoriously interpreted as promoting higher level, more abstract representations of input data within deep layers of the model, enabling in turn to achieve a higher understanding of the input data. Within DL, *representation learning* or *feature learning*, is about designing models and tuning procedures that automatically enforce a set of specified properties in certain features [Bengio et al., 2013; Zhong et al., 2016]. Among them, *semantic features* are features that are interpretable as a given propriety of the data, and that can be arbitrarily abstract, such as a dominant color, the expression of human face [Gudi, 2016].

DL models are typically called *black-box* models, because they optimize thousands, millions of parameters and their inner calculations cannot be explained. The internal representations of DL models are not interpretable, in the sense that they can't be understood by humans [Gilpin et al., 2018]. There are significant efforts to make DL more interpretable. Interpretability is commonly identified *post-hoc*, i.e. meaning is attributed to the internal representations of a given black-box model [Fong and Vedaldi, 2017]. Another paradigm, in which the work of this Ph.D. is inscribed, is to ensure interpretability by design, i.e. by using DL frameworks that guarantee interpretable representations. One way to apply this approach is to promote semantic intermediate representations, i.e. having some designated layers of a DL model matching a specific concept Marcos et al. [2020] (see Chapter 7).

Besides, it can be noted that in DL, the sources of representations are the features of a given model, whereas the target is naturally the input data. Thus, DL representations are representations of input data. However, when using DL with scientific purposes, the input data itself may be a measurement, e.g. remote sensing data within this work. Measurement is itself a representation, often the result of the processing of a sensed signal. As such it is argued in this work that DL representations obtained with measurements as input data are in fact representations of physical processes. Therefore, it is considered more proper to talk about representations of Earth ground phenomena rather than representation of satellite images.

1.4 Scope of this Ph.D.

1.4.1 Representing vegetation properties with satellite images

Vegetation covers a significant area of land masses, especially in temperate climatic regions such as Europe. In metropolitan France for instance, LULC studies estimate that about 92% of the total land is covered with some type of vegetation (58% of croplands and grasslands,

32% of forests and groves, and 2% of moorlands). As such, vegetation characteristics should be a key part of representations of this territory.

S2 data is especially suited to provide EO measurements to characterize vegetation. Its high spatial resolution of 10 m enables to characterize precisely vegetation elements, and to identify landscape details, such as parcel boundaries. Some of the measured spectral bands that operate in the visible and NIR spectrum are especially suited to vegetation, with notably three spectral bands in the red-edge region. Finally, its frequent revisit of observed scenes enables to monitor the evolution of vegetated surfaces, and infer representations that reflect seasonal changes (see section 2.1).

This Ph.D focuses on providing methods to retrieve representations of vegetation at pixel-level using S2 data. S2 images are measurements which are related to the physical properties of the observed land surfaces. These properties should intervene in representations retrieved from S2 data. In particular, for vegetation, the spectral dimension of S2 data is related to the concentration of various molecules, and to the structure of the canopy (see Chapter 4), whereas its temporal dimension informs about phenology (see section 9.1). However, purely data-driven approaches might only find representations of observed surfaces for which the physical information remains hidden. *Disentanglement* is a data-driven approach which incorporates additional statistical biases into DL models, that aim at finding *factors of variation* as representations within the data (see section 6.5). While such representations may be interesting for discovering new relationships within the target systems, they may not match known physical phenomena. Methods which are solely data-driven retrieve representations of the data and not of the underlying system, for which additional assumptions and knowledge are available.

This is why in this thesis, methods for learning semantic representations that contain physical knowledge will be studied. Such representations aim at being interpretable, and assimilated to physical quantities. In the case of vegetation, an ideal semantic representation derived from remote optical measurements can be bio-physical variables. Unfortunately, there is lack of reference data on vegetation bio-physical variables, which hampers regression approaches (see Chapter 3). *Unsupervised* approaches do not necessitate reference data for optimization, and only need remote sensing data. As discussed above, pure data-driven models are not enough, even if they were unsupervised. Therefore, this Ph.D. proposes to investigate the integration of physical knowledge into unsupervised representation learning models, for both ensuring the physical consistency of of representations and mitigating the need for reference data.

A common approach to retrieve identified physical variables using physical knowledge is to perform *model inversion*: when there are *forward models* that produce synthetic observations from input physical parameters, model inversion attempts to retrieve these parameters from observations (see Chapter 3). As a consequence, the quality of representations inferred by the method developed in this Ph.D can be assessed in the framework of model inversion.

Finally, quantifying uncertainty over predicted representations is required. This suggests using Bayesian methods that integrate uncertainty by explicitly modeling predictions and data as random variables and inferring full posterior distributions for the variables given the observed data..

1.4.2 Challenges

Retrieving interpretable representations from S2 data at large scale is a complex problem because of three data-related aspects:

1. the complexity, variability and irregularity of the data;
2. the scale of the data;
3. the lack of reference data.

S2 data is not available on a regular temporal and spatial grid. Adjacent satellite orbits have intersecting swaths leading to a difference in revisit frequency between certain covered area. Furthermore the presence of clouds is a transitory phenomenon both temporally and spatially, leading to missing clear data in measured images. Mitigating the temporal and spatial irregularity of this data is not the focus of the present work. Nonetheless the possibility of missing data will have to be taken into account. This will notably be done by using the cloud and pixel validity masks that are distributed with the reflectance products. The geographical and seasonal variability of the landscape itself is transferred to S2 data. As such, this variability must be taken into account in training data-sets for ML approaches, otherwise they may not be generalizable, and the results are at risk of not being accurate outside of a limited spatial and temporal range.

The amount of S2 data is very large. This constrains processing methods to be very efficient at large scale for feasibility. Some classical Bayesian techniques that could satisfy the requirement for uncertainty quantification such as Markov Chain Monte Carlo (MCMC) are especially computer-intensive, and would not be applicable at such a scale. For ML methods that require training, the variability of the data means that their training data-set must be of a certain size as well. This limits ML possibilities to approaches that can handle large data-sets. DL models can fulfill these constraints and has strong advantages which makes it fit for EO applications [Persello et al., 2022]:

- They can take into account vast amounts of data for training.
- DL can be adapted to a Bayesian framework by using distributions as intermediate representations and with specific optimization objectives.
- DL models are composable and can integrate differentiable components as modules (e.g. such as differentiable physical models).
- A wide variety of architectures for DL models are available for taking different aspects of the data into account (e.g. convolutional neural network (CNN) for S2 images, see section 3.3).
- The cost of training is outweighed by the relative efficiency of inference, which is only a forward pass through a model.
- DL models are "embarrassingly parallel", i.e. neural network computations can be easily divided into smaller independent computations, and parallelized. This allows to take advantage of particular hardware such as graphical processing units (GPUs) which perform parallel tasks very fast.

Finally, contrary to the S2 data that is massive, there is generally very little reference data about the properties of the ground surface. Measuring vegetation properties, such as leaf area index (LAI) or phenology requires costly and fastidious field survey effort, and is difficult to obtain with sufficient spatial and temporal frequency. This limits the feasibility of *supervised ML*, which requires *labeled*[17] data-sets for training. It can be noted that training models to predict a land cover representations such as the OSO land cover map using supervised classification is made possible by the (rare) existence of reference data in France, such as the "registre parcellaire graphique" (RPG) that collects the agricultural parcel crop names and boundaries. In the general case, for arbitrary representations of land surfaces, such data-bases do not exist, or not in sufficient quantity. To mitigate the lack of labeled data, some classical approaches simulate labeled data-sets using a simulation (physical) model, and predict representations from inverting this model. However, as will be discussed in Chapter 5, these approaches are plagued with different challenges, namely, the difficulty of choosing a

[17]The term of label refers to the reference data that the model learns to predict. It can be categorical (classification), or numerical (regression).

distribution for the simulated samples. There is a need for approaches that do not rely on reference data for training/tuning, and this is why this Ph.D. focuses on a self-supervised approach, that can be optimized using only S2 data.

1.4.3 Contributions

This Ph.D. brings 4 main contributions.

When attempting to estimate a ground physical parameter from remote sensing data as an interpretable representation of the Earth surface, reference data are usually too scarce. This limits the applicability of data-driven approaches, such as supervised ML. When there exist scientific models for these quantities, model inversion allows to retrieve them (i.e., *physics-driven* approaches). However these methods are usually *simulation-driven*, i.e. they generate synthetic samples using the scientific model, effectively replacing unavailable reference data with simulations. As a first main contribution, this Ph.D. shows in a specific application case that a mismatch between the distributions of simulations and of EO application samples impacts the performance negatively. Specifically, the bio-physical variable retrieval accuracy is hampered when performing the inversion of the PROSAIL RTM using simulated data as a training data-set for a supervised regression ANN.

To mitigate the shortcomings of supervised methods to retrieve physical variables as interpretable representations (i.e. the lack of reference data for data-driven approaches and the sensitivity of simulation-based inversion methods to the distributions of simulations), this Ph.D. develops an unsupervised methodology as a second contribution. The proposed approach exploits the representation learning framework of variational autoencoder to perform probabilistic model inversion. This methodology incorporates physical priors into training by integrating a user-defined physical model within a DL architecture, effectively bridging the gap between data-driven and physics-driven approaches. Although this approach also relies on simulations, it crucially doesn't require to choose a sampling distribution for pre-simulations, since it can learn directly from real EO data.

Finally, the two remaining contributions are the exploitation of the proposed approach in two applications. The inversion of the canopy reflectance model PROSAIL is performed with the subsequent PROSAIL-VAE, showing performances on par or superior fine-tuned production inversion model of the Biophysical Processor (BP) of Sentinel Application Platform (SNAP). Contrary to classical approaches, PROSAIL-VAE estimates all PROSAIL variables. The method is then applied to perform the inversion of temporal phenological model on crop NDVI time series with Pheno-VAE, also showing interesting results. In particular, Pheno-VAE integrates additional order constraints between temporal variables.

1.5 Conclusion

This chapter has discussed the notion of representation and how it relates to models, data and measurements, in the context of remote sensing. Before delving into approaches and models that learn representations from data in Part II and Part III, the measurements, that is, the proxy for observing and understanding reality, will be described in Chapter 2.

Chapter 2

Physical measurements

Contents

Measurement is the process of empirical, objective assignment of numbers to the attributes of objects and events of the real world, in such a way as to describe them.

Finkelstein and Grattan [1994]

2.1 Sentinel-2 imagery

Sentinel-2 is an Earth Observation *mission*, that is part of the Copernicus *programme*[1] of the EU. The objective of this mission is to acquire high resolution optical images of land masses and coastal waters. The data produced is freely accessible and is used for land monitoring, agriculture, forestry, natural disaster assessment and assistance. As of 2024, the Sentinel-2 mission uses a *constellation* of two identical satellites, Sentinel-2A and Sentinel-2B, respectively launched on 2015-06-28 and 2017-03-07 with Arianespace's Vega launcher. These satellites were designed with a 7-year nominal mission length, extendable up to 12 years. As such, Sentinel-2A operational use was planned until 2022, but since consumables (fuel for orbital manoeuvres) are not yet exhausted, it is still in operation as of early 2024. The scheduled end-of-life of the Sentinel-2 mission is 2038-09-30. Since Sentinel-2A and 2B will be both retired well before this date, replacement satellites are already planned to pursue the mission: Sentinel-2C and Sentinel-2D that will respectively substitute 2A and 2B. In fact, Sentinel-2C is ready and planned for launch in 2024, depending on 2A end-of life, and 2D is integrated and undergoing test campaigns.

2.1.1 Multi-spectral Instrument

The payload of the Sentinel-2 (S2) satellites is the multi spectral instrument (MSI), which measures the Earth reflected *radiance*. The spectral radiance L is the derivative of the spectral flux[2] ϕ that reaches the instrument w.r.t. the detector surface s w.r.t. the solid angle Ω_s that covers the light source:

$$L(\lambda) = \frac{\mathrm{d}^2\phi(\lambda)}{\mathrm{d}s\,\mathrm{d}\Omega_s \cos\theta_s} \tag{2.1}$$

with θ_s the angle between the normal vector of the source and the normal vector of the detector.

The multi spectral instrument (MSI) doesn't actually capture a full radiance spectrum. Several detectors made of photo-sensitive material are mounted on the instrument. These detectors measures the radiance transmitted through spectral filters placed in front of them. These spectral filters have a certain transmittance band $[\lambda_1, \lambda_2]$, called a *spectral band*. For each spectral band, the radiance measurement performed by the detector is characterized by a sensitivity function $S(\lambda)$. The *equivalent radiance* in a spectral band, is the total radiance measured by a detector which has a sensitivity in this band:

$$L_{\text{eq}} = \frac{\int_{\lambda_1}^{\lambda_2} L(\lambda)\, S(\lambda)\,\mathrm{d}\lambda}{\int_{\lambda_1}^{\lambda_2} S(\lambda)\,\mathrm{d}\lambda}. \tag{2.2}$$

The MSI uses 13 spectral bands in the visible (`B1`, `B2`, `B3`, `B4`), near infra-red (NIR) (`B5`, `B6`, `B7`, `B8`, `B8A`, `B9`) and short wavelength infra-red (SWIR) (`B10`, `B11`, `B12`) domains, and has a spectral sensitivity in all those bands. The spectral bands are characterized by a central wavelength λ_c, which is the barycenter of the band sensitivity:

$$\lambda_c = \frac{\int_{\lambda_1}^{\lambda_2} \lambda S(\lambda)\,\mathrm{d}\lambda}{\int_{\lambda_1}^{\lambda_2} S(\lambda)\,\mathrm{d}\lambda}. \tag{2.3}$$

The characteristics of the MSI spectral bands are summarized in Table 2.1, and their spectral response functions are provided in Figure 2.1. It can be noted that although both S2 satellites

[1] This spelling is in British English, as opposed to *program* in American English. As the United Kingdom participates to the European Space Agency and used to be part of the European Union (EU), the natural spelling is an *European Space Agency (ESA) programme*, as opposed to a *NASA program*.

[2] The electromagnetic power, the derivative of the received energy E with respect to (w.r.t.) time: $\phi(\lambda) = \frac{\mathrm{d}E(\lambda)}{\mathrm{d}t}$.

are equipped with the same instrument, there is a slight difference between the sensitivity of their spectral bands, and with the theoretical values given in Table 2.1.

Table 2.1: MSI spectral bands

Band	Use	Central wavelength (nm)	Bandwidth (nm)	Spectral domain	Spatial resolution (m)
B1	Coastal aerosols, AOT	443	20	Visible	60
B2	Vegetation, urban area, water	490	65	Visible (blue)	10
B3	Vegetation, urban area, water	560	35	Visible (green)	10
B4	Vegetation, urban area, water	665	30	Visible (red)	10
B5	Vegetation	705	15	NIR (Red edge)	20
B6	Vegetation	740	15	NIR (Red edge)	20
B7	Vegetation	783	20	NIR (Red edge)	20
B8	Vegetation	842	115	NIR	10
B8A	Vegetation	865	20	NIR	20
B9	Water vapour	945	20	NIR	60
B10	Cirrus detection	1375	30	SWIR	60
B11	Vegetation, soil moisture	1610	90	SWIR	20
B12	Vegetation, soil moisture	2190	180	SWIR	20

The MSI is a *push-broom* instrument, for which a linear array of detectors scans the surface perpendicularly to the trajectory. The MSI telescope has an across track field-of-view of 20.88° enabling a wide swath of about 290 km.

2.1.2 Orbital characteristics and angular geometry

The S2 constellation is positioned at 786 km altitude on *heliosynchronous orbit*[3]. This enables to acquire images with near constant sun illumination angles. The two S2 satellites orbit the Earth with a 180° phase, i.e. they are on opposing sides of the Earth. The S2 constellation has a high revisit time. Any location on Earth is sensed at least once every five days, although it can be even more frequent at high latitudes or at the intersection of the satellite swaths between two adjacent orbits.

Each point sensed by a S2 satellite is characterized by an angular configuration described by four angles depending on the satellite (the *observer*) and the solar directions $\overrightarrow{r_O}$ and $\overrightarrow{r_S}$ (see Figure 2.2):

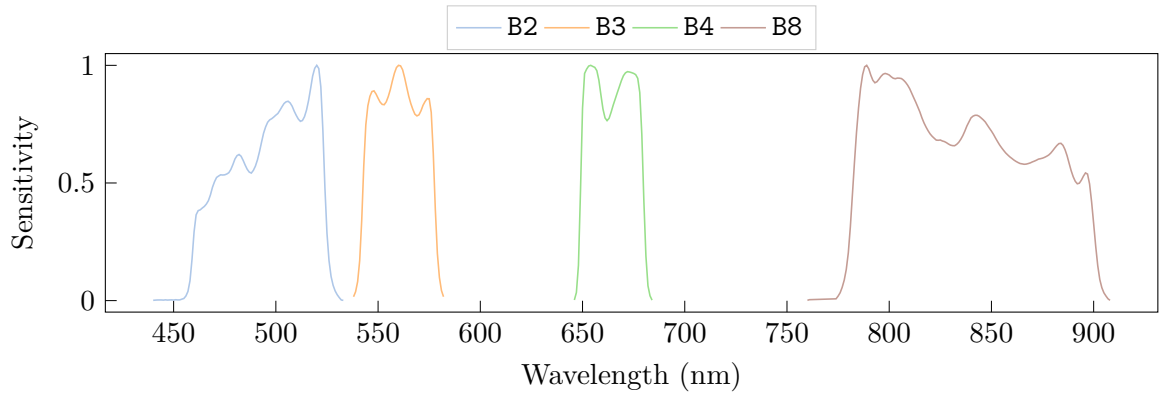
- the sun zenith angle[4] θ_S;
- the sun azimuth angle ψ_S;
- the satellite zenith viewing angle θ_O;
- the satellite azimuth viewing angle ψ_O.

The solar angles are independent from the satellite, they are a function of the geographical location, the season, and the local solar time. As the S2 satellites passes occur during the morning, the Sun always appears east of the sensed areas, i.e. with a azimuth angle $\psi_S \in [0°, 180°]$. The S2 zenith angle is restricted to about $\pm 11.9°$[5]. The S2 azimuth angle can be any value in $[0°, 360°]$, however it becomes indeterminate when the satellite is at the zenith of the sensed point.

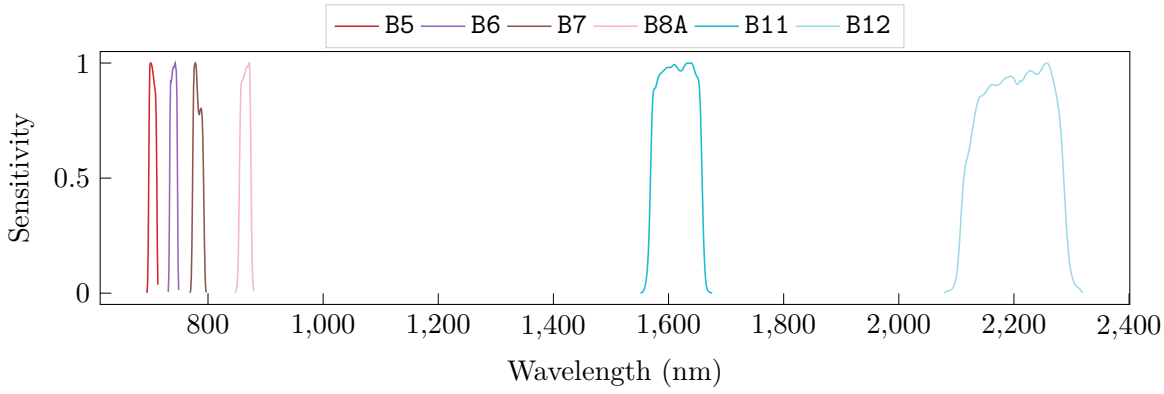
[3] An heliosynchronous orbit is characterized by a nearly polar trajectory around the Earth, in which the orbiter flies over any given point on the surface at the same local solar time.

[4] The zenith angle is not to be mistaken for the *elevation angle*, which is the angle with the horizontal plane, i.e. the elevation angle is the complementary of the zenith angle.

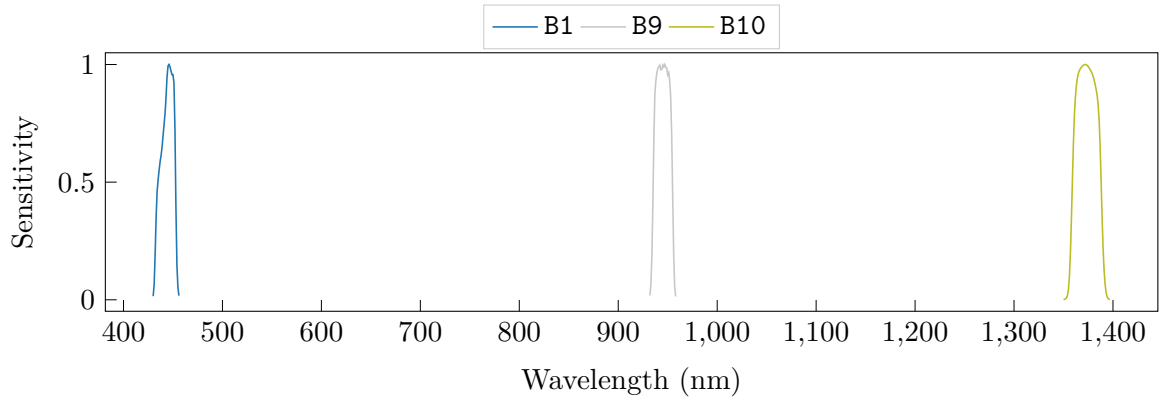
[5] Due to the Earth curvature, the S2 zenith angle range has a maximum value of about 23.9°, wider than the MSI 20.9° field of view.



(a) 10 m resolution bands



(b) 20 m resolution bands



(c) 60 m resolution bands

Figure 2.1: Sensitivity function of S2's MSI spectral bands.

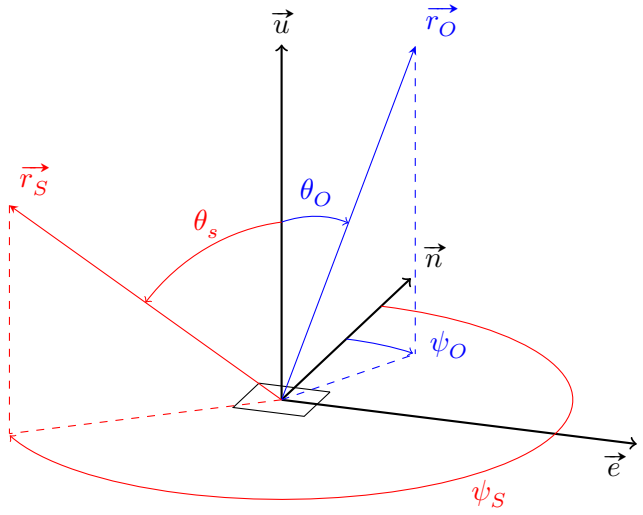


Figure 2.2: Angular geometry of pixel sensed by an observer with relative direction $\vec{r_O}$ and illuminated by the Sun with relative direction $\vec{r_S}$, with local $(\vec{e}, \vec{n}, \vec{u})$ orthonormal basis corresponding to east, north and upward directions.

2.1.3 Products

The data produced by the S2 mission is released to users in the form of different products designated by a certain level of processing [SUHET, 2015]. Each processing level uses the previous processing levels as input. The elementary products are partitioned into *granules* whose fixed size depends on the product level. The granules are the minimum indivisible area that a product covers. The different product level descriptions are summarized in Table 2.2. All Sentinel data up to level-1C are available typically within 3-24 hours of being sensed by the satellite. The most notable image processing tasks are detailed hereafter.

Table 2.2: Sentinel-2 products overview

Level	Production	Description	availability	Granule (km^2)
0	Ground segment	Compressed raw image	not released	23×25
1A	Ground segment	Decompressed level-0 image	not released	23×25
1B	Ground segment	Radiometrically corrected TOA radiance in sensor geometry	released	23×25
1C	Ground segment	Ortho-rectified TOA reflectance in UTM/WGS84	released	110×110
2A	Ground segment / On user side	BOA reflectance (atmospheric corrections)	released	110×110
3A	On user side	Composite or synthesis from several images	released	110×110

The S2 and solar angles are available from level-1C onward at a 5 km resolution, on the same geographical projection than the image.

2.1.3.1 Radiometric and geometric corrections

Radiometric corrections are rectifications applied to the measured radiance. The corrected radiometric effects include:

- The dark signal, that is a residual current in the detector that delivers an erroneous measurement, even when there is no incident photon.

- The pixel response non-uniformity, that is the measurement discrepancy between detectors even when they observe the same incident radiance.
- The defective pixels, which must be identified and mitigated.
- The crosstalk phenomenon, which is caused by the leakage of electrons and photons between adjacent pixels.

The raw image acquired by a spaceborne sensor is a two-dimensional array of pixels of radiometric values. These images need to be georeferenced i.e., the pixels must be linked to ground point coordinates. An ortho-image or ortho-rectified image is a 2-dimensional pixel array so that geometry of the array matches a particular cartographic projection. The accuracy of georeferencing can be limited and the computed position of array element can have too high errors. Image registration attempts to correct georeferencing deviations by comparing successive images of observed scenes. These corrections are performed in Level-1 processing.

2.1.3.2 MGRS tiling

It is fundamentally impossible to perform a projection without distortion of a 2 dimensional map onto the surface of a sphere, and vice-versa. As such, cartographic projection aims at choosing a mapping of the sphere onto 2D maps that minimize an arbitrary distortion at the expense of others. Flat maps are always an imperfect visual *representations* of the three-dimensional Earth. When representing a sufficiently small area on the surface of the Earth, the effect of the curvature becomes small, so representing this 3D surface with a flat projection generates negligible distortions.

This is why Sentinel-2 products are projected onto granules. From Level-1C onward, the granules are a set of *tiles* that subdivide the Earth, defined in universal transverse Mercator (UTM)/World Geodetic System 1984 (WGS84) projection. These tiles are defined as overlapping squares with a surface of 110×110 km^2. The S2 tiling system is based on NATO's Military Grid Reference System (MGRS). Based on the UTM projection, the MGRS divides the Earth into 60 zones of 6° longitude. The MGRS further divides the Earth surface into 8° latitude zones from S80° to N72°, and in a 12° zone from N72° to N84°. The georeferencing of polar regions with MGRS is not discussed here, since S2 offers systematic coverage to land surfaces between S56° to N84°. This partition of the Earth surface produces a grid, for which each element is associated with an identifier made of a natural number n and a letter L_0, or grid zone designator (GZD): nL_0. The number designates the UTM zone, and thus takes values from 1 to 60, while the letter designates the latitude band, ranging from C in the southern hemisphere to X in the north (remaining letters are for the poles). Each GZD is further divided into another grid with square 100×100 km^2 elements (as the GZD becomes closer to the poles, the area covered becomes smaller so there are fewer grid elements). The elements within a GZD do not overlap with each other, however they may overlap with elements from neighboring GZD. Each of these grid elements is specified with two letters L_1 and L_2, following a row and column index logic. For instance, the 100×100 km^2 GZD element in which Toulouse is located is designated as `31TCJ`. These GZD grid elements are the basis for S2 tiling system. S2 tiles are associated with each GZD grid element and are called by the same identifiers. The S2 tiles are larger than their MGRS GZD grid element counterparts so that there is a 10 km overlap between them.

2.1.3.3 Radiance to reflectance

Level-1C transforms the radiance measurement into a reflectance measurements. For wavelengths below 3000 nm the radiance of the terrestrial surface essentially comes from the reflection of solar irradiance[6] $E_s(\lambda)$. The radiance $L(\lambda)$, as an absolute flux of photons,

[6] Aside from rare intense light sources such as projectors oriented towrad the detector, or lava flows.

depends on the scene illumination conditions i.e. the Sun configuration, that varies seasonally depending on its distance to the Earth. The reflectance $\rho(\lambda)$, as the ratio between incident and reflected light on a surface, makes abstraction of the solar irradiance and is a radiometric quantity that solely depends on the surface characteristics:

$$\rho(\lambda) = \frac{\pi L(\lambda)}{E_s(\lambda)\cos(\theta_s)}. \tag{2.4}$$

For a detector with a sensitivity $S(\lambda)$ within a spectral band $[\lambda_1, \lambda_2]$, the equivalent reflectance is defined as:

$$\rho_{\text{eq}}(\lambda) = \frac{\pi L_{\text{eq}}(\lambda)}{E_{s,\text{eq}}(\lambda)\cos(\theta_s)}, \tag{2.5}$$

with $E_{s,\text{eq}}(\lambda) = \frac{\int_{\lambda_1}^{\lambda_2} S(\lambda)E_s(\lambda)\,\mathrm{d}\lambda}{\int_{\lambda_1}^{\lambda_2} S(\lambda)\,\mathrm{d}\lambda}$ the equivalent solar irradiance.

2.1.3.4 Atmospheric corrections

Level-2A converts top-of-atmosphere (TOA) reflectances into bottom-of-atmosphere (BOA) reflectances. To do that, it is necessary to take into account the propagation of light in the atmosphere, which is affected by absorption and scattering, besides line-of-sight interception by clouds. Correcting these effects is a difficult task that requires modeling the radiative transfer in the atmosphere. Two atmospheric correction algorithms are Sen2cor, developed by Telespazio VEGA Deutschland GmbH on behalf of the ESA [Louis, 2021], and MACCS-ATCOR joint algorithm (MAJA) developped by the Centre national d'études spatiales (CNES), the Centre d'études spatiales de la biosphère (CESBIO) and the Deutsches Zentrum für Luft und Raumfahrt (German Aerospace Center) (DLR) [Hagolle et al., 2017; Rouquié et al., 2017].

2.2 Sentinel-2 image data-sets

Within this work, S2 images are used in two ways. Firstly, training data-sets for unsupervised models are constituted with S2 images, and they will be used in Chapter 8. Secondly, S2 images corresponding to field surveys of vegetation are collected, to enable quantitative evaluation of biophysical variable retrieval techniques, as seen in Chapter 5 and Chapter 8. These two image data-sets are respectively detailed in subsection 2.2.1 and subsection 2.2.2.

Both data-sets are assembled from a collection of S2 multi-spectral images, freely available within the THEIA catalog[7]. The images are orthorectified, terrain-flattened, and atmospherically corrected with MAJA [Hagolle et al., 2017; Rouquié et al., 2017], i.e. they are a Level-2A S2 product (see Table 2.2). In this work, only 10 S2 bands among 13 are used: `B2`, `B3`, `B4`, `B5`, `B6`, `B7`, `B8`, `B8A`, `B11` and `B12`. This is because the excluded bands `B1`, `B9` and `B10` are used for cloud detection and atmospheric correction and the corresponding BOA reflectances are not reliably estimated.

2.2.1 Training patch data-set

The unsupervised biophysical variable retrieval method used in Chapter 8 is used to produce maps of biophysical variables of vegetation from S2 images. The calibration (i.e. training in ML terms) of this so-called PROSAIL-VAE model involves reconstructing input images with an autoencoder architecture. This training data-set, which will be denoted $\mathcal{D}_{\text{S2}}$ and is described below, is made of S2 images that contain vegetation areas.

[7] https://www.theia-land.fr/en/product/sentinel-2-surface-reflectance/

2.2.1.1 Data-set image selection

To ensure that the retrieval of vegetation biophysical variables from images is as general as possible and not focused on a single type of vegetation, it is important that many types of vegetation at diverse phenological stages are included in the data-set. To this end, images from various regions of interest (ROIs) with different types of vegetation across Western Europe are collected.

Only spectral bands acquired at 10 and 20 m spatial resolutions are considered and 20 m bands are upsampled using cubic interpolation to 10 m. The satellite viewing and solar illumination angles (see subsection 2.1.2) are also included in the data-set, and are upsampled from 5 km to 10 m resolution to the same grid than reflectances. Three angles are taken into account:

- the solar zenith angle θ_S,
- the satellite zenith angle θ_O,
- the relative azimuth angle, computed as the difference between the solar and satellite azimuth angle: $\psi_{SO} = \psi_S - \psi_O$.

For each tile, 1–3 different ROIs, of size 5120×5120 m^2 describing croplands and forest areas are selected (see Table 2.3). For each tile, multiple acquisitions with dates ranging between January 2016 and December 2019 are considered (see Figure 2.4). Within chosen S2 acquisitions are extracted 512×512 pixels patches corresponding to the selected ROIs, when they are observed (the cloud coverage is below 3%).

2.2.1.2 Data-set splitting for training, validation and testing

As depicted in Figure 2.5, each 512×512 ROI patch of the $\mathcal{D}_{\text{S2}}$ data-set is spatially split into 16 disjoint patches of size 128×128 pixels of 10 m^2: 14 for training, 1 for validation, and 1 for testing. Any 128×128 patch with invalid pixels (due to clouds) is discarded. These sub-patches are aggregated into $\mathcal{D}_{\text{S2,train}}$ training, $\mathcal{D}_{\text{S2,valid}}$ validation and $\mathcal{D}_{\text{S2, test}}$ testing data-sets. The number of patches for each data-set is described in Table 2.3. The splitting scheme of patches is kept constant across each ROI patch of $\mathcal{D}_{\text{S2}}$. This ensures that pixels related to specific locations are not shared between these training, validation and testing data-sets. This also ensures that these pixels are observed at several dates in their associated data-set.

The training data-set $\mathcal{D}_{\text{S2,train}}$ is used for PROSAIL-VAE model training (see Chapter 8), the validation data-set $\mathcal{D}_{\text{S2,valid}}$ is used to monitor the loss during training and ensure that the model does not over-fit. The testing data-set $\mathcal{D}_{\text{S2, test}}$ is used to assess the performances of the trained models[8]. Specifically, the testing data is used to assess reconstructions and parameter inference on unseen samples (see Figure 8.9).

2.2.2 Images of field survey sites

Four field surveys over three different sites (Las Tiesas - Barrax, Wytham Woods and BelSAR), described in section 2.4, have gathered vegetation biophysical parameters data. These data are to be used as reference data to assess retrieval performances. These quantities are to be estimated from S2 remote sensing data. As such, S2 images of the measurement sites are gathered. Since in most cases S2 overpass didn't occurred on the same day as field surveys, or clear data weren't available, for each measurement on the ground, the clear images that are temporally closest before and after the measurements are chosen. The mapping of measurement locations of the field surveys are shown in Figure 2.6, Figure 2.7 and Figure 2.8.

[8]The testing data-set doesn't intervene in either model parameters nor hyper-parameter tuning.

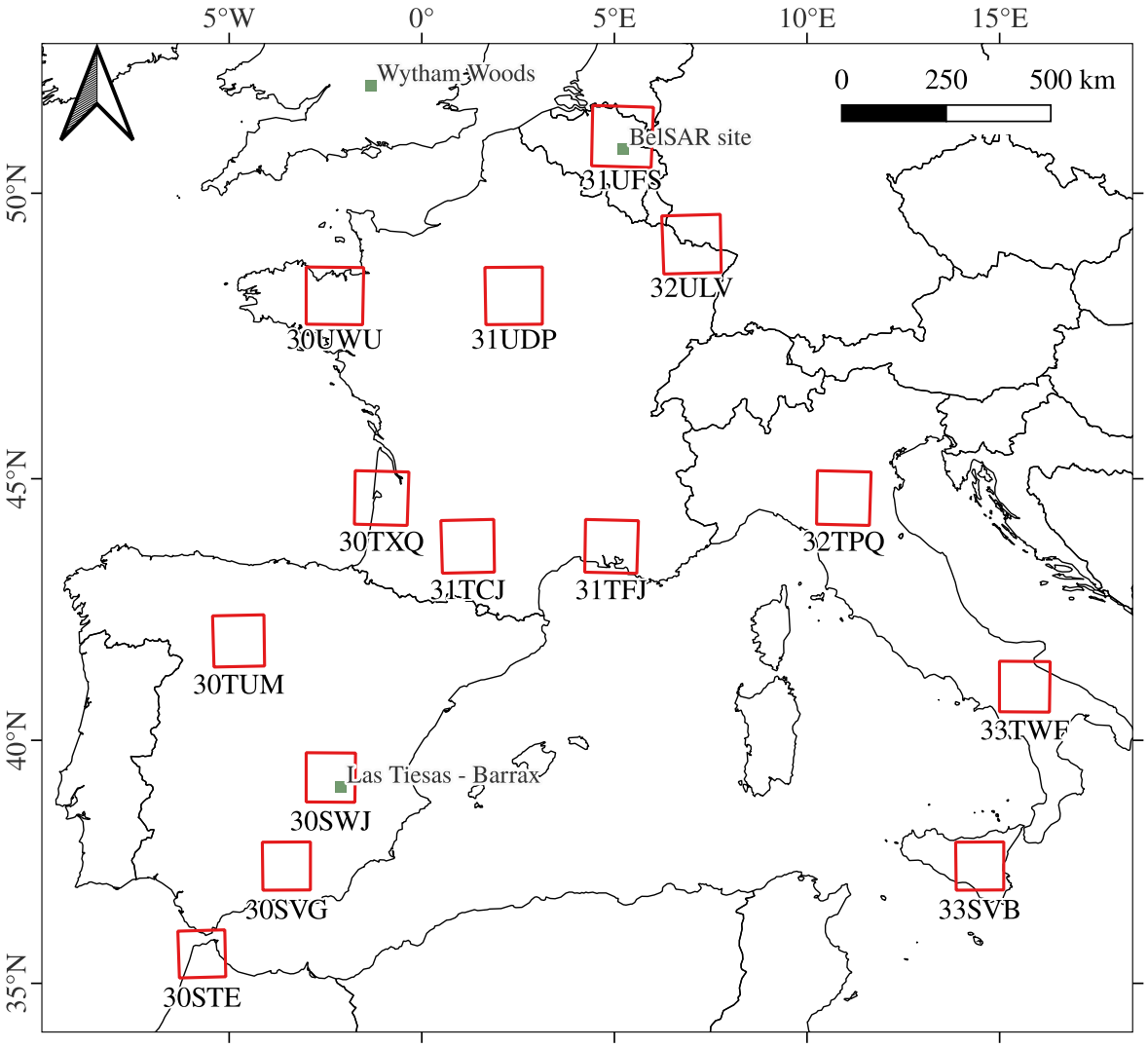


Figure 2.3: Red squares: selected MGRS tiles for the S2 training data-set. Small green squares: location of the in-situ evaluation data sites (Las Tiesas Barrax, Wytham Woods and the BelSAR Site).

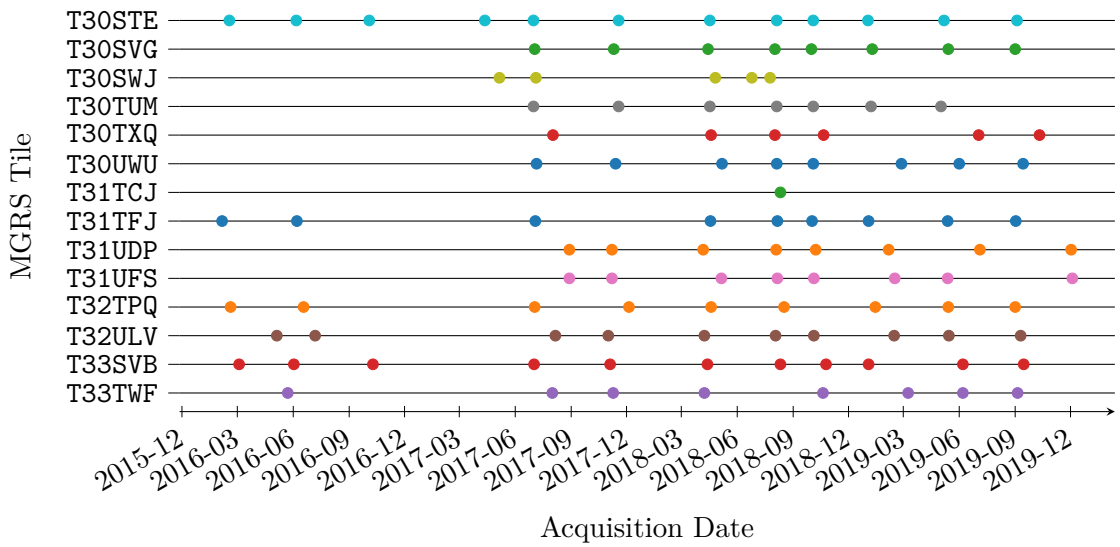


Figure 2.4: Dates of image acquisitions for each MGRS tile in $\mathcal{D}_{S2}$.

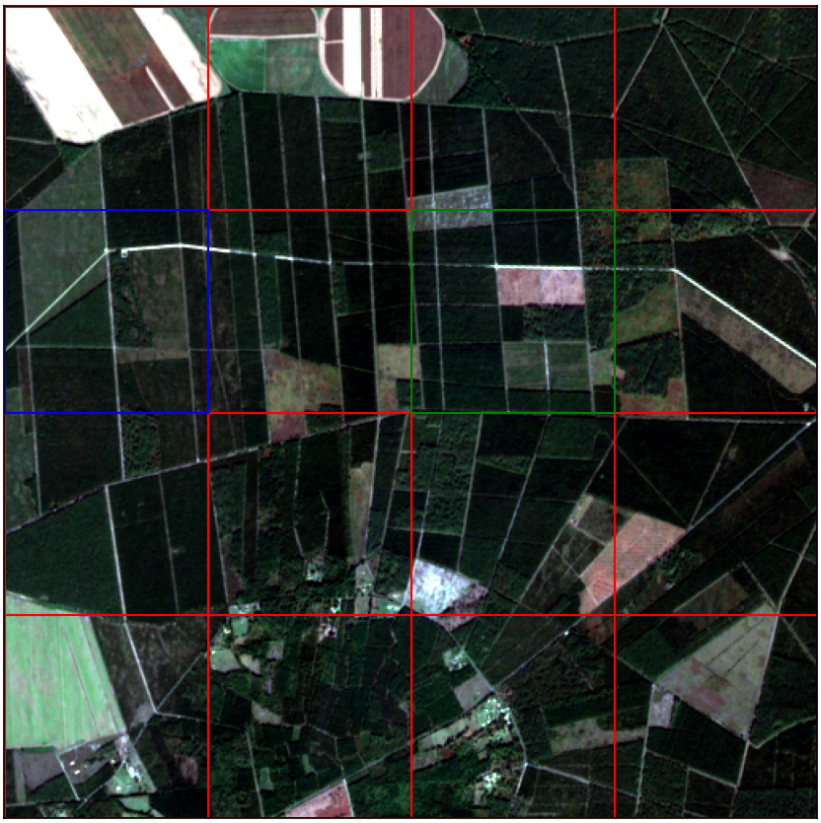


Figure 2.5: Splitting of ROI image in training (red), validation (blue), and test (green) patches (S2 image of an ROI in T30TXQ of 2019-09-11)

Table 2.3: Description of patches and pixels in training, validation and testing S2 image data-sets.

S2 Tile	Acquisitions	ROIs	Training patches (32 x 32)	Training pixels	Validation patches (32 x 32)	Validation pixels	Testing patches (128 x 128)	Testing pixels	Total pixels
T30STE	13	1	2688	2 752 512	192	196 608	12	196 608	3 145 728
T30SVG	8	1	1568	1 605 632	128	131 072	8	131 072	1 867 776
T30SWJ	5	3	896	917 504	64	65 536	4	65 536	1 048 576
T30TUM	7	1	1568	1 605 632	112	114 688	7	114 688	1 835 008
T30TXQ	6	1	1344	1 376 256	96	98 304	6	98 304	1 572 864
T30UWU	8	1	1792	1 835 008	112	114 688	8	131 072	2 080 768
T31TCJ	1	1	224	229 376	16	16 384	1	16 384	262 144
T31TFJ	9	1	2016	2 064 384	144	147 456	9	147 456	2 359 296
T31UDP	8	2	1792	1 835 008	128	131 072	8	131 072	2 097 152
T31UFS	8	2	1792	1 835 008	128	131 072	8	131 072	2 097 152
T32TPQ	9	1	1808	1 851 392	144	147 456	9	147 456	2 146 304
T32ULV	10	1	2240	2 293 760	160	163 840	10	163 840	2621440
T33SVB	11	1	2464	2 523 136	176	180 224	10	163 840	2 867 200
T33TWF	8	3	1792	1 835 008	128	131 072	8	131 072	2 097 152
Total			23 984	24 559 616	1728	1 769 472	108	1 769 472	28 098 560

As shown in Figure 2.3, some tiles of the training data-set (see subsubsection 2.2.1.1) contain the measurement sites. This is the case for the BelSAR site in tile `31UFS` and Las Tiesas - Barrax site in `30SWJ`. However, care was taken to ensure that selected ROIs do not overlap with these test sites where in-situ measurements were collected.

The images of Las Tiesas - Barrax and BelSAR sites were also taken from the THEIA catalogue. However S2 tile `30UXC` which the Wytham site isn't part of this catalogue. Therefore, the related Level-2A S2 product were taken from the CNES PEPS catalogue[9], with on demand MAJA atmospheric correction.

2.3 Biophysical data field surveys

The definitions of the leaf area index (LAI) are reviewed in subsection 2.3.1, and the associated methods of measurement and estimation are discussed in subsubsection 2.3.1.3. Measurement methods for chlorophyll content are described in subsection 2.3.2.

2.3.1 Measuring the Leaf Area Index

The LAI [Watson, 1947] is defined as one half of the total green leaf area per unit horizontal ground surface area. The LAI quantifies the amount of leaf area in an ecosystem and is a critical variable in processes such as photosynthesis, respiration, and precipitation interception [Fang et al., 2019]. It is strongly correlated to crop yield and is a feature frequently used for forecasting [Chen et al., 2018b]. As a fundamental attribute of global vegetation, the LAI is highlighted as an essential climate variable (ECV) by the Global Climate Observing System (GCOS) [GCOS, 2011].

2.3.1.1 Definition scopes

A number of vegetation area indices[10] closely related to the LAI are commonly defined in studies, to account for nuances. In all cases, the vegetation area index designates half the area (or one-sided area) of vegetation elements per unit horizontal ground area. While the LAI takes all leaves into account, the green LAI (GLAI) is restricted to green leaves, which are photo-synthetically active and participate in evapo-transpiration [Broge and Leblanc, 2001]. The counterpart for senescent leaves is the brown LAI (BLAI). Their relationship is simply:

$$\mathrm{LAI} = \mathrm{GLAI} + \mathrm{BLAI}. \tag{2.6}$$

The LAI and GLAI are generally equivalently used in canopy reflectance models, under the hypothesis that BLAI is negligible or accounted for by other variables.

Another LAI related index is the green area index (GAI), defined as the area index of green organs, which includes green leaves, stems, branches, and fruits [Baret et al., 2010; Fang et al., 2019]. Notably, this excludes brown leaves which are typically accounted for in the LAI. It should be noted that GAI is not the area index of only photosynthetic elements. Non-green leaves may also contribute to photosynthesis, and photosynthesis may not be performed in all green tissues, or under extreme conditions.

The broader plant area index (PAI) accounts for the area of the whole plant, and makes no distinction between leaves and non leaves-elements, or green and brown areas, [Weiss et al., 2004]. The LAI is related to the PAI through the woody area index (WAI) [Toda and Richardson, 2018], which accounts for the area of woody elements (i.e. non-leaves):

$$\mathrm{PAI} = \mathrm{LAI} + \mathrm{WAI}. \tag{2.7}$$

[9] `https://www.peps.cnes.fr`

[10] The term *vegetation area indices* defines here the general set of area indices.

The non-leaf elements of vegetation can be divided into stem and branches elements, whose surface is respectively accounted for by the stem area index (SAI) [Duveiller et al., 2011] and branch area index (BAI) [Kucharik et al., 1998]. The WAI is the sum of those indices:

$$\mathrm{WAI} = \mathrm{BAI} + \mathrm{SAI}. \tag{2.8}$$

The contribution of branches through the BAI is usually neglected, and the WAI and SAI may bee seen as equivalent. In some studies, the contribution of dead leaves are added to the SAI [Fang et al., 2019]. In such cases, the SAI is complementary to the GLAI:

$$\mathrm{PAI} = \mathrm{GLAI} + \mathrm{SAI}. \tag{2.9}$$

Depending on the application, some indices such as SAI can be extended to other plant parts, or other vegetation area indices specific to those elements may be defined [Duveiller et al., 2011]. Some vegetation area indices may be designated by other acronyms in some studies, such as the PAI that is designated as VAI, for *vegetation area index* in Fassnacht et al. [1994], or the shoot area index defined in Chen and Cihlar [1995]. The ability to distinguish the various plant elements and their corresponding vegetation area indices depends on the application, and on the sensor used for measurement. The correspondence between plant elements and several vegetation area indices is provided in table 2.4.

The different area indices described here account for the surface area of different parts of plants while discarding explicitly others. Although the different definitions attempt to put a precise meaning to each quantity, it can actually hamper understanding and comparison between studies. The definition of the LAI itself can be blurry: the GCOS actually defines the LAI as half the surface area of green leaves, which is the definition given for the GLAI. There can be scientific interest in distinguishing the various plant parts in different area indices, for given applications. However, it is actually quite difficult and fastidious to measure the LAI precisely, as will be discussed in subsubsection 2.3.1.3. This is one of the main reasons for different works to propose other area indices closely related to the LAI. Specifically, with indirect measurement, separating the contribution of the different plant parts to the global surface area is difficult. As such, to account for inaccuracies of isolating specific plant parts, area indices that encompass all measured surface areas can be proposed alternatively to the LAI. Moreover, distinguishing all plant parts for accurate vegetation area definition is complicated for ground measurements, therefore it is all the more difficult for space-borne remote sensing. LAI, GAI, GLAI and PAI may be seen as roughly equivalent from this perspective.

Table 2.4: Correspondence between vegetation area indices and vegetation elements. (✓) indicates vegetation elements that may or may not be taken into account by the corresponding index, depending on studies

Index	Leaves		Non-leaves			
	Green leaves	Brown leaves	Green stem	Green branches	Non-green stem	Non-green branches
LAI	✓	✓				
GLAI	✓					
BLAI		✓				
GAI	✓		✓	✓		
PAI	✓	✓	✓	✓	✓	✓
WAI			✓	✓	✓	✓
BAI				✓		✓
SAI		(✓)	✓	(✓)	✓	(✓)

2.3.1.2 The canopy clumping effect

The LAI (or some other vegetation area indices described in 2.3.1) is usually not acquired directly when using optical measurement methods (see 2.3.1.3) because of the canopy clumping effect [Ryu et al., 2010]. When using optical field instruments, the gap fraction of the canopy is estimated, and an *effective* LAI (LAI_{eff}) is derived, instead of the *actual* LAI. The LAI_{eff} is defined as the product of the LAI with the clumping index $\Omega(\theta_S)$, which is a function of the solar zenith angle θ_S:

$$LAI_{eff}(\theta_S) = \Omega(\theta_S) \times LAI. \tag{2.10}$$

The clumping index quantifies the canopy clumping effect, and measures the non-randomness of the leaf distribution. The LAI_{eff} assumes that the leaf distribution is uniformly random i.e. that there is no canopy clumping effect Chen et al. [2005]. On the contrary, the LAI is an index that is obtained by correcting the LAI_{eff} with the clumping index.

2.3.1.3 LAI field measurements methods

The study of leaf shapes and dimensions is called *phyllometry*. The LAI, which is defined as half the ratio of leaf surface by ground surface, is a *phyllometric* parameter. To measure the LAI, there are two complementary approaches: direct methods, which measure the LAI from sampled leaves, and indirect methods that rely instead on the measurement of other, more accessible parameters as a proxy.

Direct methods Direct LAI estimation requires harvesting leaves from within an area of study [Jonckheere et al., 2004; Weiss et al., 2004]. Leaves can either be destructively picked from the vegetation, or collected as leaf litter in autumn season during the vegetation senescence with leaf litter traps. The area of detached leaves is then measured in the laboratory. The area of individual leaves can be measured with mechanical or optical planimeters, scanners or even smartphones with dedicated applications, and automated with belt conveyors. Alternatively, the LAI can be assessed from the weight of dry leaf samples w [Baret et al., 2010] by assuming some constant leaf area per unit of dry leaf mass SLA[11]:

$$LAI = SLA \times w. \tag{2.11}$$

Direct methods are the most precise, as they directly measure the value of interest. However they are labor and time consuming, and are difficult to deploy over large areas and for repeated measurements. Therefore, they are only used for local studies and for validating indirect methods.

Indirect methods Indirect methods estimate the LAI from other vegetation variables. Indirect methods can be divided into methods with and without physical contact with the studied vegetation.

One of the earliest LAI measurements techniques involved physical contact with the leaves. A measure of foliage area of a low growing vegetation within a *quadrat*[12], was derived by piercing the leaves with a physical probe made of thin needles. The foliage area was derived by counting the number of contact points between the needles and the leaves: this is the quadrat point method [E.B. and E.A., 1933]. The method was further improved by inclining the quadrat at an optimal 32.5° instead of using it vertically with *inclined point quadrats* [Wilson, 1960], that reduces the measurement error. While the common random uniform leaf distribution is not required contrary to other indirect methods, the inclined points quadrats

[11] specific leaf area

[12] A quadrat is a frame used in ecology, geography, and biology to identify and isolate a standard unit of area for studying of the distribution of an item over a given area.

is work intensive, as it requires a large number of needle insertion. Furthermore, it is hardly applicable to canopies higher than 1.5 m due to the limited length of needles.

Another indirect contact LAI estimation method is vegetation allometry, which provides empirical links between the various dimensions of the plants. In particular, for trees, the leaf area is highly correlated with tree height, sapwood[13] basal area[14], stem basal area, etc [McDowell et al., 2002; Waring et al., 1980]. Allometric techniques are specific to site, species, and phenological stage of the vegetation. Furthermore, they can be destructive, as sapwood measurements requires to expose the tree cross section.

Non-contact indirect methods use the interaction between the leaves and an electromagnetic radiation source and is captured by a sensor. These techniques are also qualified as *optical*. For field measurements discussed here, the sensor is on the ground, whereas for *remote sensing*, the sensor is either *airborne* or *space-borne*. When the radiation source is artificial, the method is *active*. One such indirect non-contact active method is based on the inclined point quadrats method which uses laser beams as probes instead of physical needle [Denison, 1997].

On the contrary, when the scene exposition is natural (i.e. the Sun), the method is *passive*. As Weiss et al. [2004] reviews, non-contact indirect passive methods of field LAI estimation method are based on the measurement of the gap fraction. The gap fraction represents the probability of incident radiation to pierce the canopy and reach the observer. Using gap fraction to retrieve the LAI [Martens et al., 1993] is based on Monsi [1953], which finds that the light attenuation trough the canopy can be approximated with Beer-Lambert law. Specifically, it is the LAI_{eff} that is retrieved, and the canopy clumping effect (see 2.3.1.2) must be estimated to correct for the LAI. For measuring the gap fraction, the technique depends on the sensor used.

2.3.2 Measuring the chlorophyll content

Chlorophyll is an essential photo-synthetic pigment for plants. This molecule is found in the chloroplast organelle within plant cells. The chlorophyll is responsible for absorbing incident sunlight radiation in visible and infra-red (IR) spectrum, and freeing electrons to enable the photosynthesis chemical reaction. Photosynthesis (see Equation 2.12) transforms carbon dioxide and water into sugar and dioxygen molecules and is arguably one of the most important processes for life on Earth.

$$6\,H_2O + 6\,CO_2 \longrightarrow C_6H_{12}O_6 + 6\,O_2 \tag{2.12}$$

There are actually two types of chlorophyll molecules, with very similar composition and structure with complementary role in the photosynthetic process: chlorophyll A ($C_{55}H_{72}MgN_4O_5$) and chlorophyll B ($C_{55}H_{70}MgN_4O_6$). The green color of plants is due to the chlorophyll pigments that absorb short wavelength (violet/blue) and long wavelength (orange/red) visible light. The effect of both types of chlorophyll is commonly apprehended simultaneously with a combined pigment concentration $C_{ab} = C_a + C_b$.

The chlorophyll content in plant elements is influenced by many factors, the vegetation type, phenological stages, the environmental conditions, the geographical location, etc [Li et al., 2018]. As such it is a key quantity to characterize vegetation.

Like the LAI (see subsubsection 2.3.1.3) the chlorophyll content can be measured in-situ with direct destructive techniques (subsubsection 2.3.2.1), or estimated with indirect methods (subsubsection 2.3.2.2).

[13] Soft outer layers of recently formed wood between the heartwood and the bark, containing the functioning vascular tissue.

[14] Cross-sectional area of trees at breast height (≈ 1.4 m).

2.3.2.1 Spectrophotometrical measurement

Most direct measurements of chlorophyll content in plant elements are performed following the protocol of Lichtenthaler [1987]. Photosynthetic plant elements (e.g. leaves) are destructively sampled from vegetation, and are dissolved with acetone or ethanol. The absorbance[15] A at several wavelengths of the solution is then measured with a *spectrophotometer*. Concentrations C can be retrieved from the absorbance, since the *Beer-Lambert law* links those two quantities:

$$A(\lambda) = \epsilon(\lambda) lC \tag{2.13}$$

with l the optical path length and $\epsilon\lambda$ is the *molar attenuation coefficient* (or *absorptivity*) of the molecular species. Measuring the solution absorption at two wavelengths enables to distinguish the concentration of chlorophyll A from chlorophyll B[16], by solving (i.e. inverting, see Chapter 3) a linear system derived from the Beer-Lambert law:

$$\begin{cases} A(\lambda_1) = \alpha_1 C_a + \beta_1 C_b \\ A(\lambda_2) = \alpha_2 C_a + \beta_2 C_b \end{cases} \iff \begin{cases} C_a = \dfrac{1}{\alpha_1\beta_2 - \alpha_2\beta_1}\left(\beta_2 A(\lambda_1) - \beta_1 A(\lambda_2)\right) \\ C_b = \dfrac{1}{\alpha_1\beta_2 - \alpha_2\beta_1}\left(\alpha_1 A(\lambda_1) - \alpha_2 A(\lambda_2)\right) \end{cases} \tag{2.14}$$

with α_1, β_1, α_2 and β_2 coefficients that are function of the species absorptivity at the two wavelengths. It can be noted that a similar protocol can be applied to measure the concentration of other pigments, such as carotenoids.

2.3.2.2 Indirect measurements

Like the LAI, indirect chlorophyll measurements involve some sort of model inversion. Estimating the chlorophyll content with remote sensing relies on optical measurements since it is a pigment whose effect can only be sensed from its spectral signature, although some attempts to combine multi/hyper-spectral measures with synthetic aperture radar (SAR) have been performed Zhang et al. [2018].

As for in-situ measurements, *chlorophyll-meters* are simple, fast and relatively inexpensive tools to estimate this pigment. These small devices non-destructively estimate the absorbance of leaves and produce an estimate of the pigment concentration. One of the most popular chlorophyll-meter is Konica Minolta's SPAD-502 Markwell et al. [1995]. The SPAD-502 produces dimensionless measurements M that are related to the chlorophyll concentration through an experimental relation (usually linear, polynomial, exponential or homographic) that must be calibrated beforehand, by using spectrophotometry measurements (see subsubsection 2.3.2.1), like [Brown et al., 2022]:

$$C = a\mathrm{e}^{bM}. \tag{2.15}$$

Nonetheless, these kind of measurements are less precise than spectrophotometry, and are subject to more variability. Besides, the choice of the calibration equation has a direct impact on the chlorophyll content estimation, which adds to the uncertainty of chlorophyll-meter measurements [Li et al., 2024].

2.4 In-situ data-sets

In order to assess the inference of vegetation biophysical parameters, as will be performed in Chapter 5 and Chapter 8, it is necessary to establish reference databases. Here is described

[15] The absorbance must not be mistaken for the absorbtance. The absorbance is defined as the decimal logarithm of the transmittance : $A = \log_{10} T$

[16] with more precision when these wavelengths are chosen to match absorptivity maxima of the dissolved species.

the composition of an in-situ measurement data-set $\mathcal{D}_{\text{IS}}$, composed of direct measurements collected in different field campaigns under the framework of fiducial reference measurements for vegetation (FRM4Veg) and BelSAR projects, and associated S2 images (see subsection 2.2.2). The complete data-set contains 211 LAI and 121 canopy chlorophyll content (CCC) reference measurements with estimated uncertainties. This data serves as a reliable reference for quantitatively evaluating the accuracy of the biophysical variable estimations. Nonetheless, prudence must be exerted when using this data-set for validation purposes, because it contains a limited number of data points of a few vegetation types at few phenological stages, and because the measurement themselves are affected with an uncertainty. Using more measurement data in future work will be essential to confirm the results obtained here.

2.4.1 Fiducial reference measurements for vegetation

FRM4Veg is an ESA managed project focused on establishing the protocols required for traceable in-situ measurements of vegetation-related parameters, to support the validation of Copernicus products [Origo et al., 2020]. In this project, different field campaigns have been performed over two test sites covering agricultural crops (Las Tiesas-Barrax, Spain) and deciduous broadleaf forest (Wytham Woods, UK). Besides LAI, CCC and bare soil measurements, their associated uncertainties are also available for both test sites.

Considering an elementary sampling unit (ESU) of 20×20m, about 12 to 15 LAI individual measurements were performed using digital hemispheric photography (DHP). Leaf chlorophyll content (LCC) measurements were made on 13 points per ESU with a Konica Minolta SPAD-502 chlorophyll meter. Considering 3 leaves per point with 6 replicates per leaf, 234 measurements were thus performed for each ESU. The relative values provided by the SPAD-502 were converted to absolute units using calibration functions specific to each vegetation type [Origo et al., 2020]. Finally, CCC measurements were obtained by applying $\text{CCC} = \text{LCC} \times \text{LAI}$. Although the measurement of non-destructive chlorophyll can lead to imprecise and unreliable results [Zhang et al., 2022], it must be noted that measurements provided by the FRM4Veg campaigns were performed with rigorous and high standard protocols considering important number of repetitions and uncertainty estimations [Brown et al., 2021a].

The FRM4Veg also estimate the clumping index of the vegetation canopy, which enable to derive LAI_{eff} and *effective* CCC (CCC_{eff}) ($\text{CCC}_{\text{eff}} = \text{LCC} \times \text{LAI}_{\text{eff}}$) measurements (see subsubsection 2.3.1.2).

In this Ph.D., the measurements collected in 2018 and 2021 over the Barrax test site are used (see Figure 2.6). As proposed in Brown et al. [2021b], alfalfa measurements are not considered because these crops had been thinned prior to the S2 acquisitions, but after the in situ measurements were made. By considering the dates of in-situ measurements, satellite images acquired on (2018-05-16, 2018-06-13, 2018-07-22) are considered for the Barrax test site.

In the case of Wytham test site, only data from 2018 is considered due to the lack of clear satellite image acquisitions over the summer of 2021. For this study area, S2 images acquired on 2018-06-29 and 2018-07-06 are used.

2.4.2 BelSAR

In the framework of the BelSAR project [Bouchat et al., 2022, 2023; Orban et al., 2021], field measurements and airborne SAR bistatic acquisitions were collected over a test site in Belgium, near the town of Gembloux during the summer of 2018. This project had the objective of assessing the interest of SAR bistatic acquisitions for vegetation and soil moisture monitoring. It also wanted to validate the capabilities of active-passive satellite configurations by ensuring the performances of L-band SAR bistatic and multistatic imagery.

In the BelSAR campaign, measurements were collected over 10 maize and 10 winter wheat fields larger than 1 hectare (ha) in size (see Figure 2.8).

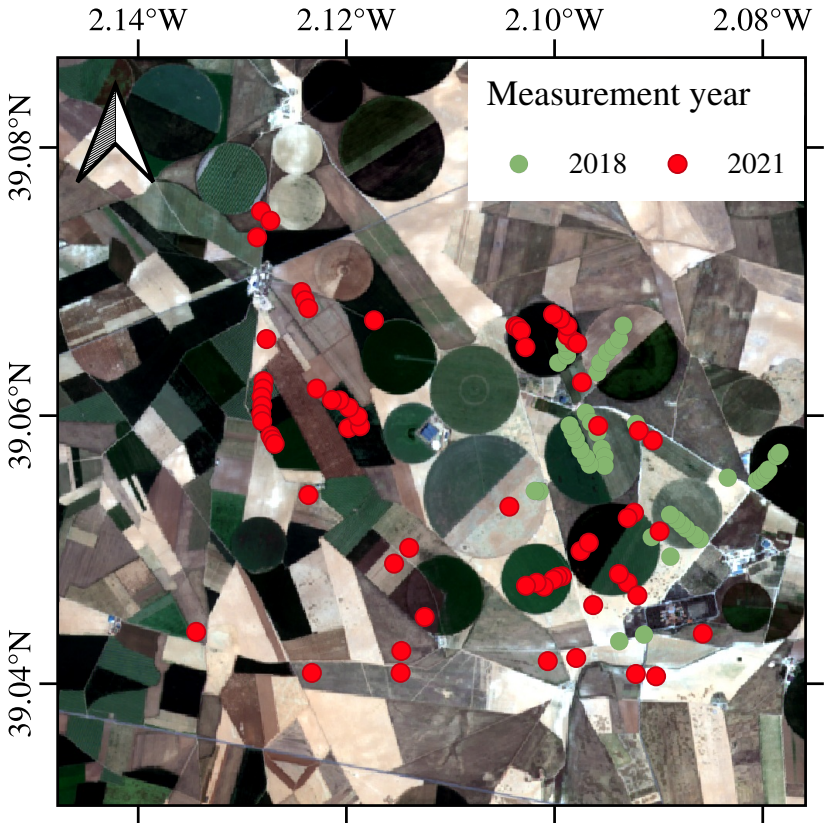


Figure 2.6: In-situ measurements of 2018 and 2021 FRM4Veg in Las Tiesas - Barrax test site (S2 image of 2018-06-13).

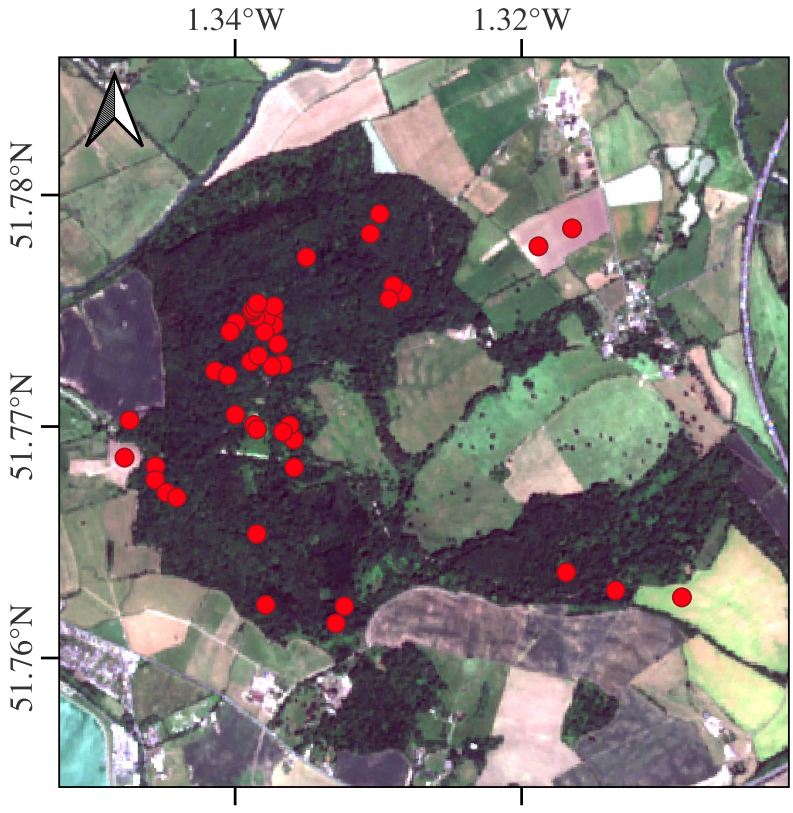


Figure 2.7: In-situ measurements collected over FRM4Veg Wytham area in 2018.

The BelSAR project provides PAI measures for wheat parcels and GAI for maize fields. Considering that PAI and GAI are similar to LAI Fang et al. [2019], both measurements are interpreted as LAI in our study. For each field, 3 measurements were made at each date. Accordingly, the average of the measurements computed at each parcel for each date is considered as reference. Following the same idea, the standard deviation at parcel level is interpreted as an uncertainty measurement.

A timeline of the BelSAR measurement dates and available S2 images is shown in Figure 2.9. It should be noted that the measurements of 2018-08-29 were excluded from our study, as no valid S2 images were available within 24 days before or after. There are three or four acquisitions for each parcel, as field measurements are not carried out for each maize or wheat parcel for each measurement date.

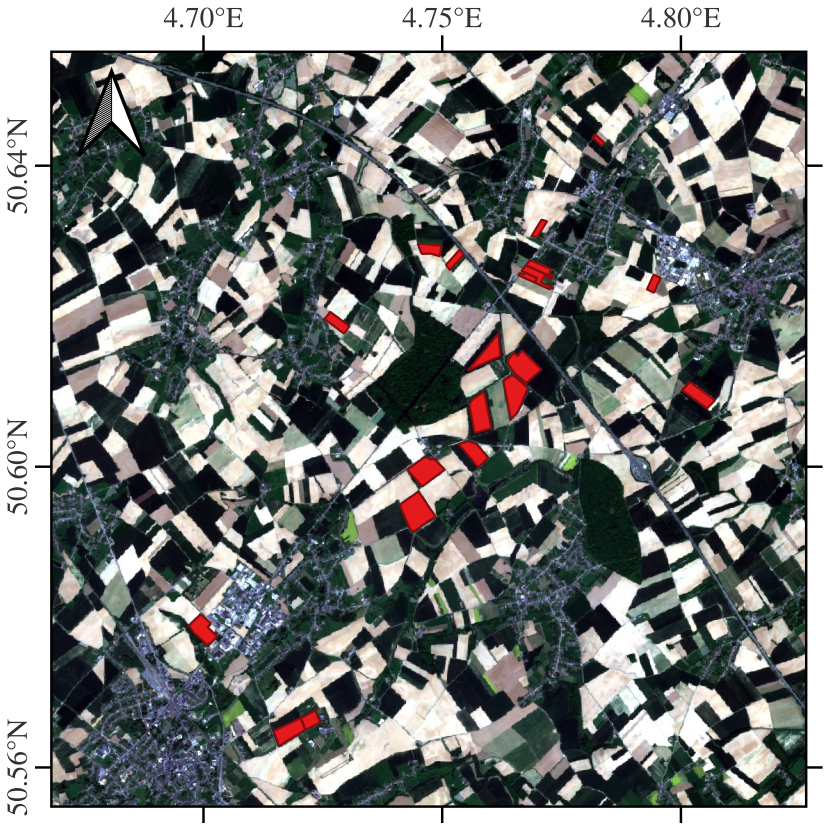


Figure 2.8: Field parcels of BelSAR test site over a S2 image acquired on 2018-05-08).

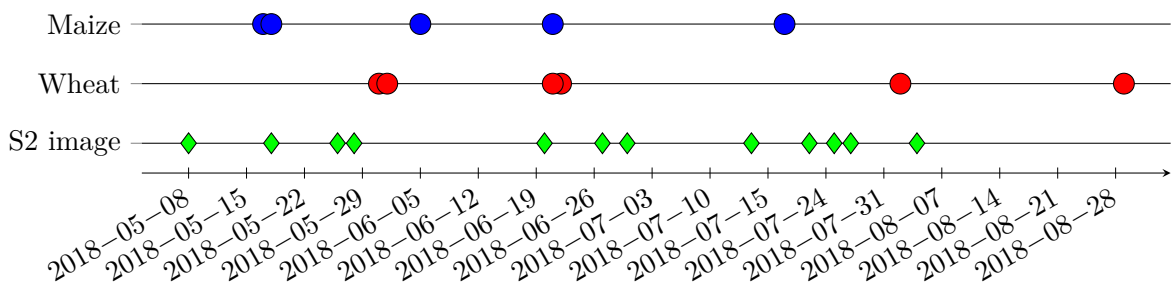


Figure 2.9: Timeline of measurement dates for maize and wheat parcels of BelSAR campaign, and available S2 images.

2.4.3 Validating with the in-situ data-set

The in-situ data-set $\mathcal{D}_{\text{IS}}$ contains reference in-situ measurements of LAI and CCC and the S2 images of the sites that are temporally the closest possible to the measurements dates. For almost all measurements, there is no S2 image captured on the same day. Therefore to assess the performance of a prediction algorithm with some in-situ data on a given day, predictions are made using the first prior available image and first posterior available image. For a given measurement $y^\star$, an estimate y on the same day (at $t = 0$) is derived from predictions y_{before} and y_{after} performed on images respectively sensed t_{before} days before and t_{after} days after, using linear interpolation:

$$y = \frac{t_{\text{after}}}{t_{\text{after}} + t_{\text{before}}} y_{\text{before}} + \frac{t_{\text{before}}}{t_{\text{after}} + t_{\text{before}}} y_{\text{after}}. \tag{2.16}$$

When a given algorithm outputs distributions as predictions, the estimate is the interpolation of expected values, and the uncertainty is derived as the interpolated standard deviation.

Part II

Inversion of vegetation models

Chapter 3

Model inversion and regression

Contents

3.1 Forward and inverse modeling

Scientific models (e.g. mathematical, numerical models) are a representation of physical or theoretical systems, that attempt to explain their behavior. Finding relevant models for those systems is called "modeling". Models can be designed in two ways. *First-principle modeling*, is about using theories, physical or mathematical laws to build models, whereas *empiric modeling* forms models from data. These approaches are not exclusive: models can both have theoretical background and be tuned with data. Many models of interest aim at representing the relationship between two sets of phenomena, of quantities that correlate in the target system.

3.1.1 Model inputs, outputs and parameters

Scientific models usually characterize a *directional*, or *conditional* relationship that leads to describe the related phenomena as *input*, and *output* of the model. Model inputs are commonly thought as model *parameters*, however in this Ph.D., a distinction is made between the two. Parameters, usually denoted $\boldsymbol{\theta}$, are defined as *global* variables, attributes that define the model itself (e.g. artificial neural network (ANN) weights, regression coefficients), whereas model input are *local* variables, that vary along with the model output.

In the remainder of this work, model inputs and outputs (and parameters $\boldsymbol{\theta}$) are assumed to be numerical quantities whose components are put in a vector form, unless specified otherwise[1], and are respectively denoted $\boldsymbol{x}$ and $\boldsymbol{y}$. Their domains are respectively subspaces $\mathbb{X}$ of $\mathbb{R}^n$ and $\mathbb{Y}$ of $\mathbb{R}^m$. Collections of samples $\boldsymbol{x}$ and $\boldsymbol{y}$ are gathered in *data-sets* $\mathcal{D}_{\boldsymbol{x}} \subset \mathbb{X}$ and $\mathcal{D}_{\boldsymbol{y}} \subset \mathbb{Y}$. In this context, the input and output are representations of physical or theoretical phenomena, whereas the model represents the relationship between those phenomena as a mathematical relationship between the inputs and outputs. For instance, as discussed in subsubsection 2.3.2.1, the concentration of a chemical species within a solvent is related to the absorbance of the solution at a given wavelength. These two quantities can be taken as input and output of the Beer-Lambert law chosen as a model.

Additionally, the taxonomy of this work enables to make a straightforward distinction between *model inversion* (discussed in the subsequent subsection 3.1.2) and *model calibration*. Both notions can be qualified as *estimation* of an unknown quantities, however model inversion is about retrieving a model input $\boldsymbol{x}$, whereas calibration estimates $\boldsymbol{\theta}$.

3.1.2 Model inversion

Characterizing the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$ leads to a dual problem. When $\boldsymbol{x}$ and $\boldsymbol{y}$ are chosen as input and outputs, the *direct* (or *forward*) problem is about finding the direct/forward model that describes the outputs $\boldsymbol{y}$ from the inputs $\boldsymbol{x}$. The complementary *inverse* problem aims at inferring the input $\boldsymbol{x}$ from the outputs $\boldsymbol{y}$. More formally, *solving the inverse problem* associated with the model $\mathcal{F}$ (or *inverting the model* $\mathcal{F}$) is, for a given $\boldsymbol{y}$ in a data-set $\mathcal{D}_{\boldsymbol{y}}$, to find the input $\boldsymbol{x}$ whose propagation (or *forwarding*) through $\mathcal{F}$ produces $\boldsymbol{y}$:

$$\forall \boldsymbol{y} \in \mathcal{D}_{\boldsymbol{y}},\ \text{Find } \boldsymbol{x} \in \mathbb{X} \text{ s.t. } \mathcal{F}(\boldsymbol{x}) = \boldsymbol{y}. \tag{3.1}$$

Sometimes, additionally to the numerical value of some input samples $\boldsymbol{x}$, an inverse model $\mathcal{F}^{-1}$ to the forward model $\mathcal{F}$ can be found. In this case the model input $\boldsymbol{x}$ can be found by simply propagating $\boldsymbol{y}$ through $\mathcal{F}^{-1}$:

$$\forall \boldsymbol{y} \in \mathcal{D}_{\boldsymbol{y}},\ \mathcal{F}^{-1}(\boldsymbol{y}) = \boldsymbol{x}. \tag{3.2}$$

[1] In this work, only finite dimensional models, that are described with a finite number of parameters are considered. However in general, there are models that require an infinite number of parameters, countable or uncountable. In the latter case, functions can be used as parameters.

It can be remarked that the qualification of models as forward or inverse is arbitrary, since the associated problems describe the two sides of a bi-directional relationship between two variables $\boldsymbol{x}$ and $\boldsymbol{y}$. Deciding if a model is forward or inverse is a matter of context, depending on which variable can be considered as a cause, which one is a consequence. Usually, one direction of the relationship is more accessible, more straightforward than the other, so it is considered to constitute the direct problem. The inputs of the forward model are usually hidden variables whereas the outputs are observed through measurements. Since forward models can produce outputs that are comparable to measurements, they are data *simulators*.

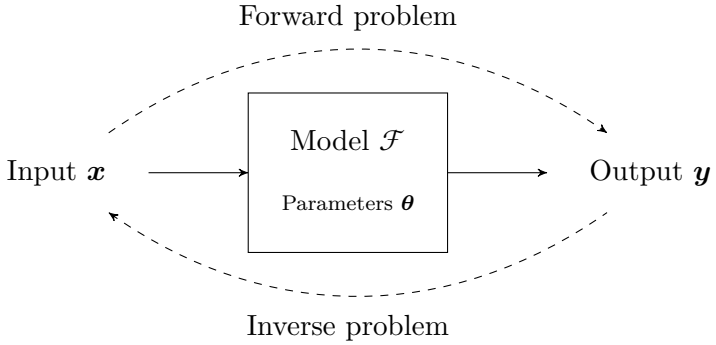


Figure 3.1: Forward and inverse problems. Dashed arrows indicate the direction of inference.

3.1.3 Well-posedness and approximate inversion

The possibility of solving an inverse problem (or more generally a numerical problem) is described with Hadamard's well-posedness conditions:

1. *existence* of a solution,
2. *uniqueness* of the solution,
3. *stability*, i.e. the continuous dependence of the solution on data.

In the case of linear models, these conditions translate to matrix specifications: they must be invertible and *well-conditioned* (see subsection B.4.1). Problems that don't satisfy those conditions are *ill-posed*. As a rule of thumb, all interesting inverse problems are ill-posed, usually because solutions are not unique. Common causes for inverse problem ill-posedness are over-determined[2] and under-determined[3] systems of equations, occurring notably because of a difference in the dimension of input and output space. For instance, the gravitational field outside of a body is uniquely determined by the spatial distribution of mass, however there are (infinitely) many different mass distributions that allow a given gravity. To collapse the set of all possible solutions of an inverse problem to a single solution, additional constraints must be added, i.e. prior information must be provided. Furthermore, for physical systems ill-posedness is amplified because of measurements being noisy, and the model being imperfect. Thus, inverse problems rather aim at finding *plausible* input values of the model. The model inversion problem (Equation 3.1) is then usually relaxed as:

$$\forall \boldsymbol{y} \in \mathcal{D}_{\boldsymbol{y}}, \text{ Find } \boldsymbol{x} \in \mathbb{X} \text{ s.t. } \mathcal{F}(\boldsymbol{x}) \approx \boldsymbol{y}. \tag{3.3}$$

Model inversion intersects with *estimation theory* since it is about estimating unknown variables $\boldsymbol{x}$ from data $\boldsymbol{y}$ that can contain an random component.

[2] More equations than unknown.
[3] Fewer equations than unknown

3.1.4 Model inversion and regression

Regression analysis aims at finding relationships between two sets of variables. It estimates a mapping of *dependent* variables[4] $\boldsymbol{v}$ as a function f[5] of the *independent* variables[6] $\boldsymbol{\xi}$. Usually, the regression function is assumed to belong to a certain family of functions (e.g., linear, polynomial, neural networks), which depends on a set of parameters $\boldsymbol{\beta}$, and is *fit* to the data by optimizing $\boldsymbol{\beta}$. An *error* term $\boldsymbol{\varepsilon}$ accounts for effects non-modeled by f and for randomness.

$$\boldsymbol{v} = f\left(\boldsymbol{\xi}, \boldsymbol{\beta}\right) + \boldsymbol{\varepsilon} \tag{3.4}$$

In this paragraph, the notation of inputs and outputs are distinguished between those $(\boldsymbol{x}, \boldsymbol{y})$ of a forward model $\mathcal{F}$ and $(\boldsymbol{\xi}, \boldsymbol{v})$. As will be explained below, these quantities do not always match. Beyond this section, the notation $(\boldsymbol{x}, \boldsymbol{y})$ will be used again for all models.

Simple regression refers to regression of a scalar dependent variable $v \in \mathbb{R}$ from a scalar independent variable $\xi \in \mathbb{R}$. *Multiple regression* refers to regression of a scalar dependent variable $\boldsymbol{v} \in \mathbb{R}^k$ from a one or several independent variables $\boldsymbol{\xi} \in \mathbb{R}^m$. *Multivariate regression* is a generalization to the case of several dependent variables $\boldsymbol{v} \in \mathbb{R}^k$ and several independent variables $\boldsymbol{\xi} \in \mathbb{R}^m$. *Linear regression* considers the fitting of a model that is linear with respect to (w.r.t.) the parameters $\boldsymbol{\beta}$.

$$\boldsymbol{v} = \sum_{j=1}^{k} f_j\left(\boldsymbol{\xi}\right) \beta_j + \boldsymbol{\varepsilon} = \Xi\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{3.5}$$

Regression methods can be either *parametric* or *non-parametric* (see subsection 3.1.1), depending on the model f being fit to the data (in the non-parametric case, $\boldsymbol{\beta}$ can be a function, or infinite-dimensional).

Regression analysis is relevant for both forward and inverse modeling, depending on what is chosen as dependent and independent variables. Empiric modeling, that produces (forward) models $\mathcal{F}$ from data, can be performed using a regression approach on variables separated into predictors $\boldsymbol{x} = \boldsymbol{\xi}$ and outcomes $\boldsymbol{y} = \boldsymbol{v}$.

Performing the inversion of a model $\mathcal{F}$ with regression can be classified in two categories.

1. A *local* approach, that uses the model $\mathcal{F}$ as the regression function f, and to retrieve the model inputs $\boldsymbol{x}$ as the unknown parameter $\boldsymbol{\beta}$ of the regression. This implies that regression must be performed each time for retrieving model inputs $\boldsymbol{x}_i$ from different output samples $\boldsymbol{y}_i$.

2. The *global* approach is to use regression to estimate an inverse model $\mathcal{F}^{-1} = f$. The forward model is used to generate samples pairs $(\boldsymbol{x}, \boldsymbol{y})$. This is performed by selecting samples of input data $\boldsymbol{x}$ (or drawing them from a prior distribution, as seen in section 5.1), and by propagating these samples through the forward model to produce the outputs $\boldsymbol{y} = \mathcal{F}\left(\boldsymbol{x}\right)$. Then, regression is applied on the synthetic data $(\boldsymbol{x}, \boldsymbol{y})$, by using $\boldsymbol{y}$ as predictors $\boldsymbol{\xi}$, and $\boldsymbol{x}$ as outcomes $\boldsymbol{v}$ — in reverse to that of the forward model $\mathcal{F}$. As such it is only necessary to perform regression once, and then model inversion can be carried out by forwarding the different samples to the estimated model inverse.

Regression involves fitting a model onto the data $(\boldsymbol{x}, \boldsymbol{y})$, usually by optimizing a *criterion* (or *objective function*), commonly denoted J, or $\mathcal{L}$ for ML, in which the criterion is usually called a *loss function*. As such, regression transforms the problem of estimation (of $\boldsymbol{x}$) in inverse modeling into an optimization problem (of a regression model).

[4] Also called outcomes, regressands, or response or *labels* in Machine Learning (ML) literature.
[5] Commonly called the *regression function*.
[6] Also called regressors, covariates, predictors, explanatory variables of *features* in ML

When performing regression of an outcome $\boldsymbol{\upsilon}$ from predictors $\boldsymbol{\xi}$ to produce a regression model f, residuals $\boldsymbol{r}$ need to be minimized.

$$\boldsymbol{r} = \|\boldsymbol{\upsilon} - f(\boldsymbol{\xi})\| \tag{3.6}$$

Residuals are the difference between an observed outcome and the outcome estimated from propagating the covariates through the model. They are different from *regression errors* that are the difference between the observed outcome $\boldsymbol{\upsilon}$ and the true value of the outcome $\boldsymbol{\upsilon}^\star$ which is usually unknown[7]: $\boldsymbol{\varepsilon} = \|\boldsymbol{\upsilon} - \boldsymbol{\upsilon}^\star\|$. The residuals are an observable estimate of an unobservable error. *Least squares* regression explicitly minimizes the sum of squared residuals.

3.1.5 Regression metrics

Once a regression function f has been set for modeling the relationship between two sets of variables, it is crucial to assess *goodness-of-fit*, i.e. how well the model matches the data. Different metrics can be used to evaluate how well the model fits the data, in two situations: either to quantify the fitting of the model to the *training data*, or for estimating how good the model is at modeling the relationship with new, unseen data. In the latter case, the regression model is used as a *predictive model*.

The root mean squared error (RMSE) is the standard deviation (std) of the residuals:

$$\text{RMSE} = \frac{1}{N}\sqrt{\sum_{i=1}^{N}((x_i) - y_i)^2}. \tag{3.7}$$

Due to the residual being squared, the RMSE is more affected by large errors: it is sensitive to outliers. Since the RMSE is related to the mean squared error (MSE) with a square root, it is also minimized by least-square methods (see subsection 3.2.1).

The mean absolute error (MAE) is the average of the absolute residuals:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|f(x_i) - y_i|. \tag{3.8}$$

Contrary to the RMSE, the effect of errors on the MAE is proportional to their magnitude, so large errors affect the MAE less.

The coefficient of determination (R^2)[8] [Wright, 1921] quantifies the proportion of the variance in the dependent variable y that is predictable from the independent variables x. It is defined as one minus the average of squared residuals r over the variance of the independent variable:

$$R^2 = 1 - \frac{\overline{r^2}}{\text{var}(y)} = 1 - \frac{\sum_{i=1}^{N}(f(x_i) - y_i)^2}{\sum_{i=1}^{N}(\overline{y} - y_i)^2}.^{9} \tag{3.9}$$

The coefficient of determination ranges from $-\infty$ (arbitrarily bad fit), to 1 (perfect fit), through 0 (a model that regresses data to the mean $\overline{y}$). Chicco et al. [2021] argues that the R^2 score is the most informative metric for regression analysis, and reports the goodness of fit more truthfully.

All presented metrics assume that the regression model produces deterministic outcomes. To apply them in the context of probabilistic models that estimate distributions, *point estimates* must be derived (see subsection 6.2.2).

[7] This "true outcome value" is related to the unobserved "true relationship" (or *latent generative model*) that links the predictors and the outcomes.

[8] Pronounced "r squared".

[9] $\sum_{i=1}^{N}(f(x_i) - y_i)^2$ is commonly called the *residual sum of squares* and $\sum_{i=1}^{N}(\overline{y} - y_i)^2$ the *total sum of squares*.

3.1.6 Data assimilation

In remote sensing, or geo-sciences in general, the notion of *data assimilation* frequently appears alongside that of *inverse problem*. From a theoretical point of view, these two concepts can be seen as equivalent: they are both about using observed data with a model to estimate hidden variables of a system. In practice, they are nuanced by referring to differently formulated models and applications. With data assimilation, the model of a target system is characterized by internal *state variables* rather than *input variables*. Those state variables are governed with an *evolution model*, which is usually a temporal model that links current or future values with past values. Therefore, data assimilation is performed with *dynamic systems*. The outputs (or observations) are modeled as a function of the internal state with an *observation mapping*. The evolution model enables to extrapolate, or forecast future states of the system. However, the state variable evolution from a known initial value invariably ends up diverging from their true value, because these models can never be perfect (some effects, or external forces may not be taken into account correctly, or at all) and states are never known with infinite precision[10]. Taking measurements into account aims at re-calibrating the state estimate. One of the most widespread data assimilation techniques is the Kalman filter [Kalman, 1960] and its derivatives, which are also ubiquitous in control theory.

Performing data assimilation for retrieving representations of land surfaces in the form of state variables is out of the scope of this Ph.D., although it makes for an interesting perspective.

3.2 Regression methods

In this section, an overview of different regression methods is presented, in the perspective of performing model inversion. There are two ways of performing the inversion of a moel $\mathcal{F}$ using regression (see subsection 3.1.4). Regression is about finding a function f fitting some data. Inversion can be carried out with this function when the inversion target is a parameter of f. This method is mainly performed with least squares methods (see subsection 3.2.1) . Alternatively, f is used to approximate an inverse model $f \approx \mathcal{F}^{(-1)}$, when training data is produced by $\mathcal{F}$. Most methods described here perform regression that better suit this latter inversion regime: k-nearest neighbors in subsection 3.2.2, random forests in subsubsection 3.2.3.1, support vector regression in subsubsection 3.2.3.2 neural networks in subsubsection 3.2.3.3.

3.2.1 Least squares minimization

Least squares are parameter estimation methods that can be used to perform (parametric) regression. These methods aim at adjusting the parameters $\boldsymbol{\beta} \in \mathbb{R}^k$ of a model f on a data-set of N pairs of independent-dependent variables $(x_i, y_i) \in \mathbb{R} \times \mathbb{R}$, based on the minimization of the sum of the squares of the residuals $r_i \in \mathbb{R}$ as a criterion $\mathcal{L}$, like the name suggests.

$$\mathcal{L} = \sum_{i=1}^{N} r_i^2 = \sum_{i=1}^{N} \left(y_i - f\left(x_i, \boldsymbol{\beta}\right)\right)^2 \tag{3.10}$$

Least squares are commonly put into two categories: linear least squares (subsubsection 3.2.1.1) and non-linear least squares (subsubsection 3.2.1.2), depending on the model f considered. Additionally, like regression, least squares can be qualified as *simple*, *multiple* or *multivariate* depending on the dimensions of x_i and y_i (see subsection 3.1.4).

[10] For non-linear models, small or infinitesimal differences in the value of a state, can lead to large differences in later states. This dependence on initial conditions is known as the *butterfly effect*, and is one of the basic principles of *chaos theory*.

3.2.1.1 Linear least squares

Linear least squares assume that the model is a linear combination of the parameters, i.e.:

$$f(\boldsymbol{x}_i, \boldsymbol{\beta}) = \sum_{j=1}^{k} f_j(\boldsymbol{x_i})\,\beta_j \tag{3.11}$$

Linear least squares methods are often used to fit *linear regression* models. In the case of multiple linear regression, for which there are several independent variables:

- the independent variable is a matrix $X \in \mathcal{M}_{N,m}(\mathbb{R})$, whose columns $\boldsymbol{x}_j$ are sets of N observations $x_{i,j}$ of each of the m independent variable components (the rows $\boldsymbol{x}_i$ are the m values of the independent variable components for a given observation i),
- the dependent variable is a vector $\boldsymbol{y} \in \mathcal{M}_{N,1}(\mathbb{R})$, whose columns are the N measurements y_i,
- the m regression parameters are gathered in a vector $\boldsymbol{\beta} \in \mathcal{M}_{m,1}(\mathbb{R})$
- the error is gathered in a vector $\boldsymbol{\varepsilon} \in \mathcal{M}_{N,1}(\mathbb{R})$,

and the linear equations can be written matricially:

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{3.12}$$

In this case, the least square criterion is:

$$\mathcal{L}(\boldsymbol{\beta}) = \|\boldsymbol{r}\|_2^2 = \|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 = \sum_{i=1}^{N} (y_i - \boldsymbol{x}_i\boldsymbol{\beta})^2 . \tag{3.13}$$

The goal is to retrieve the value of the parameters $\boldsymbol{\beta}$ that minimizes $\mathcal{L}$. It can be noted that the multiple linear regression can be extended to the multivariate linear regression, for which there are $k \geq 1$ dimensions to the dependent variables. However, for linear models, the dimensions of the dependent variables are assumed independent, therefore it amounts to solving for each dimension separately.

$\mathcal{L}$ is a positive quadratic function of $\boldsymbol{\beta}$, which is convex and admits a single minimum. To find this minimum, the gradient of the objective function is equated to the zero vector, which leads to the expression of the *normal equations*[11]:

$$\nabla_{\boldsymbol{\beta}}\mathcal{L}(\boldsymbol{\beta}) = \begin{pmatrix} \dfrac{\partial \mathcal{L}}{\partial \beta_1} \\ \vdots \\ \dfrac{\partial \mathcal{L}}{\partial \beta_m} \end{pmatrix} = \mathbf{0} \iff X^\top X\boldsymbol{\beta} = X^\top \boldsymbol{y}. \tag{3.14}$$

Solving the normal equations enable to retrieve the optimal value $\boldsymbol{\beta}_{\text{OLS}}$ of $\boldsymbol{\beta}$[12]:

$$\boldsymbol{\beta}_{OLS} = (X^\top X)^{-1} X^\top \boldsymbol{y}. \tag{3.15}$$

This solution to the normal equations is an estimator of the unknown parameters that corresponds to a specific formulation of linear least squares (LLS), called ordinary least squares (OLS), with a set of associated assumptions are realized. It is an *unbiased*[13] and

[11] Normal equations are equations obtained by equating to zero the gradient (or the partial derivatives) of the least square objective function.

[12] The analytical derivation of β introduces a matrix inverse $(X\top X)^{-1}$. It isn't necessary to compute this matrix, and it is almost always more computationally efficient to just solve the linear system.

[13] An estimator is unbiased, if its bias, which is the difference of the expected value of the estimator and the true value of the estimated parameter, is zero.

consistent[14] estimator provided the that the errors in the model have a finite variance and are *exogeneous*[15]. It is also an *efficient*[16] estimator if the errors are homoscedastic[17]. Furthermore, OLS can be interpreted as maximum likelihood estimation (MLE) (see subsection 6.1.1) with a Gaussian prior with zero mean and constant variance.

When these assumptions are not fulfilled, other formulations of the least square objective enable to improve performances. For instance, weighted least squares (WLS) enables to take heteroscedastic errors into account. General least squares also enables to account for correlated errors. When the regression system is under-determined, thus ill-posed, *regularization* can be applied to the least square objective, such as Tikhonov regularization [Tikhonov and Arsenin, 1977] (also known as ridge regression).

3.2.1.2 Non-linear least squares

A key assumption of LLS regression, is that a linear model of the parameters is an accurate representation of the target system. When the regression model is a non-linear function f of the parameters, least square minimization methods are called non-linear least squares (NLLS). Contrary to LLS, NLLS do not have closed-form nor unique solutions for the least square objective J, aside for specific and simple examples. The gradients of the objective function, that must be equated to zero to find minimums, are function of both the variables and the parameters:

$$\nabla_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta}) = 2\mathrm{J}_{\boldsymbol{\beta}}[\boldsymbol{r}(\boldsymbol{\beta})]^{\top} \boldsymbol{r}, \tag{3.16}$$

with $\mathrm{J}_{\boldsymbol{\beta}}[\boldsymbol{r}(\boldsymbol{\beta})]$ s.t. $\left(\mathrm{J}_{\boldsymbol{\beta}}[\boldsymbol{r}(\boldsymbol{\beta})]\right)_{i,j} = -\dfrac{\partial r_i(\boldsymbol{\beta})}{\partial \beta_j}$ the Jacobian matrix of the residual $\boldsymbol{r}(\boldsymbol{\beta})$ w.r.t. the parameters $\boldsymbol{\beta}$. Since an analytical solution isn't usually available the optimal parameters are estimated iteratively instead:

$$\boldsymbol{\beta}^{(n+1)} = \boldsymbol{\beta}^{(n)} + \Delta\boldsymbol{\beta} \tag{3.17}$$

with $\Delta\boldsymbol{\beta}$ the *shift vector*, which is the update of the parameters between two iterations.

3.2.1.3 Gradient descent

There are several methods that enable to compute the shift vector. The gradient descent method uses the property of the gradient of a function being the direction of *steepest* ascent to compute the shift vector as the negative gradient of the objective function:

$$\Delta\boldsymbol{\beta} = -\alpha_n \nabla_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta}), \tag{3.18}$$

with α_n the learning rate (lr), a coefficient that regulates the update step. Another method, the *Gauss-Newton algorithm* derives it from the linearization of the residual at the next iteration $\boldsymbol{r}\left(\boldsymbol{\beta}^{(n+1)}\right)$ with Taylor expansion:

$$\boldsymbol{r}\left(\boldsymbol{\beta}^{(n+1)}\right) = \boldsymbol{r}\left(\boldsymbol{\beta}^{(n)}\right) - \mathrm{J}_{\boldsymbol{\beta}}\left[\boldsymbol{r}\left(\boldsymbol{\beta}^{(n)}\right)\right]\Delta\boldsymbol{\beta} + \mathcal{O}(\Delta\boldsymbol{\beta}). \tag{3.19}$$

Using this linearization in the gradient of the objective function $\nabla_{\boldsymbol{\beta}}\mathcal{L}\left(\boldsymbol{\beta}^{(n+1)}\right)$ (Equation 3.16) and equating to the zero vector yields normal equations:

$$\mathrm{J}_{\boldsymbol{\beta}}\left[\boldsymbol{r}\left(\boldsymbol{\beta}^{(n)}\right)\right]^{\top}\mathrm{J}_{\boldsymbol{\beta}}\left[\boldsymbol{r}\left(\boldsymbol{\beta}^{(n)}\right)\right]\Delta\boldsymbol{\beta} = \mathrm{J}_{\boldsymbol{\beta}}\left[\boldsymbol{r}\left(\boldsymbol{\beta}^{(n)}\right)\right]^{\top}\boldsymbol{r}\left(\boldsymbol{\beta}^{(n)}\right) = \nabla_{\boldsymbol{\beta}}\mathcal{L}\left(\boldsymbol{\beta}^{(n)}\right), \tag{3.20}$$

[14] An estimator x_N is consistent if it converges in probability to a finite value x_∞ with the number of samples N increasing.

[15] The errors ε are exogeneous if their conditional expectation w.r.t. the predictors X is zero : $\mathrm{E}[\varepsilon|X] = 0$. As a consequence, the errors must have zero expectation ($\mathrm{E}[\boldsymbol{\varepsilon}] = 0$) and be uncorrelated with the independent variables X ($\mathrm{E}[X^{\top}\boldsymbol{\varepsilon}] = 0$).

[16] An efficient estimator has the smallest possible variance among all estimators of the same class, and achieves the Cramér-Rao lower bound.

[17] The Gauss-Markov theorem states that the OLS estimator is a best linear unbiased estimator (BLUE) if errors have zero mean, are homoscedastic and are uncorrelated.

that enable to derive the vector shift[18]:

$$\Delta\boldsymbol{\beta} = \alpha_n \left(\mathrm{J}_{\boldsymbol{\beta}} \left[\boldsymbol{r} \left(\boldsymbol{\beta}^{(n)} \right) \right]^\top \mathrm{J}_{\boldsymbol{\beta}} \left[\boldsymbol{r} \left(\boldsymbol{\beta}^{(n)} \right) \right] \right)^{-1} \nabla_{\boldsymbol{\beta}} \mathcal{L} \left(\boldsymbol{\beta}^{(n)} \right). \tag{3.21}$$

Several issues arise from solving NLLS with such iterative procedures:

- there is a need for an initial estimate of the parameters,
- several local minima may exist, which makes the choice of the initial parameters all the more important, so that the minimum reached is a global minimum,
- computing the Jacobian matrix of the model may be complicated or intractable for certain models.

Finally, there are several variants of the iterative optimization algorithm for solving NLLS. The Levenberg-Marquardt algorithm[19], sometimes called *damped least squares*, proposes a similar procedure that is based on modified normal equations that integrate a *damping factor* λ:

$$\left(\mathrm{J}_{\boldsymbol{\beta}} [n]^\top \mathrm{J}_{\boldsymbol{\beta}} [n] + \lambda \mathbf{I} \right) \Delta\boldsymbol{\beta} = \mathrm{J}_{\boldsymbol{\beta}} [n]^\top \left(\boldsymbol{y} - f \left(\boldsymbol{x}, \boldsymbol{\beta}^{(n)} \right) \right). \tag{3.22}$$

The choice of λ and its evolution throughout optimization enables to tweak the convergence properties of the algorithm. The Levenberg-Marquardt belongs to the class of *trust region algorithms*, that approximate the NLLS objective function with a simpler function (quadratic for Levenberg-Marquardt), within a sub-space of the parameter space: the *trust region*. The size of this trust region changes during optimization (in Levenberg-Marquardt, it is controlled by λ.), it is reduced when the surrogate objective function badly approximates the NLLS objective, and increased otherwise.

Finally, sometimes the search region for the parameters estimated must be restricted to a sub-space of the parameter space $\mathbb{I}$.

$$\boldsymbol{\beta}^\star = \underset{\boldsymbol{\beta}}{\arg\min}\ \mathcal{L} \left(\boldsymbol{\beta} \right) \ \text{s.t.}\ \ \boldsymbol{\beta} \in \mathbb{I} \tag{3.23}$$

The trust region reflective algorithm (TRRA)[20] [Branch et al., 1999], which will be used in Chapter 9, enables to produce a parameter estimate within bounds. Its basic principle is to perform a trust-region estimation of the parameters shift vector $\Delta\boldsymbol{\beta}$, and to accept the parameter update if it falls within the bounds $\mathbb{I}$. Otherwise, the parameters are updated in the direction of the reflection of the shift vector along the crossed boundary.

3.2.2 k-nearest neighbors

The k-nearest-neighbors (KNN) problem is a non-parametric optimization problem of finding the k closest points to a query data-point among a data-set, w.r.t. to a given space metric [Cover and Hart, 1967]. More formally, given N data-points p_i in a data-set $\mathcal{D}_p$, a query data-point q, and a distance metric $\|\cdot,\cdot\|$, a KNN search finds the k nearest points of q in $\mathcal{D}_p$. Usually, KNN returns the set $\mathbb{H}_k(q)$ of the k indices of the nearest data-points:

$$\mathbb{H}_k(q) = \left\{ i_j \in [\![1, N]\!] \ \text{s.t.}\ \ j \in [\![1, k]\!] \right\} \tag{3.24}$$

such that

$$\forall i \in \mathbb{H}_k(q), \forall n \in [\![1, N]\!], n \notin \mathbb{H}_k(q) \Rightarrow \|q, p_i\| \leq \|q, p_n\|. \tag{3.25}$$

[18] The Gauss-Newton and the gradient descent algorithms can be understood as special cases of the *generalized gradient descent algorithm*, that compute the parameter update as $\Delta\boldsymbol{\beta} = -\alpha_n Q \left(\boldsymbol{\beta}^{(n)} \right) \nabla_{\boldsymbol{\beta}} \mathcal{L} \left(\boldsymbol{\beta}^{(n)} \right)$, with $Q \left(\boldsymbol{\beta}^{(n)} \right)$ a symmetrical positive definite matrix.

[19] This algorithm is implemented in the `curve-fit` function of the python library `scikit-learn`, that performs NLLS.

[20] Ibid.

There exist a variety of more or less efficient methods to return the KNN. The most straightforward approach is to compute the distances of all N data-point p from the query q, and retrieve the indices of the k lowest. This approach is relatively inefficient, since it requires iterating throughout the whole data-set, thus a computational complexity of $\mathcal{O}(N)$. More efficient approaches rely on storing data in specific structures such as k-dimensional trees[21]. To improve the query time of KNN, k *approximate* nearest neighbors can be returned instead of the k exact ones.

KNN are commonly used jointly with lookup tables (LUTs), that are arrays that associate an index to data. LUTs replace runtime computation with array indexing, which can be interesting when data computation is relatively expensive. In particular, LUTs can be pre-calculated.

Once nearest neighbors have been retrieved from a data-set, it is possible to use them to perform regression. Assuming data points p_i are made of two components $\boldsymbol{x}_i$ (regressors) and $\boldsymbol{y}_i$ (regressands), KNN w.r.t. to a distance along $\boldsymbol{x}$ is applied on a "partial" query, $\boldsymbol{x}_q$:

$$\mathbb{H}_k(\boldsymbol{x}_q) = \left\{ \left(\boldsymbol{x}_{i_j}, \boldsymbol{y}_{i_j} \right) \text{ s.t. } j \in [\![1, k]\!] \right\}. \tag{3.26}$$

The underlying assumption is that data-points that are neighbors w.r.t. $\boldsymbol{x}$ are also neighbors w.r.t. $\boldsymbol{y}$. Consequently, an estimate $\boldsymbol{y}_q$ associated to the query $\boldsymbol{x}_q$ is derived from the $\boldsymbol{y}$ components of the retrieved k neighbors. For instance, with $k = 1$ the regressand $\boldsymbol{y}_q$ is simply assigned the value $\boldsymbol{y}$ of the closest neighbor. For $k > 1$ there are several possibilities, such as using the median, or the mean of $\boldsymbol{y}_i$ of the closest neighbors. It is also possible to use an average weighted by the distances $\|\boldsymbol{x}_q, \boldsymbol{x}_i\|$. In such case, the regressand is some kind of interpolated values derived from the closest neighbors.

When the data is simulated from a model, KNN regression effectively performs the inversion of this model. For inverting the PROSAIL model (see Chapter 4), a classical approach is to use KNN on data simulated with PROSAIL and organized in a LUT [Schiefer et al., 2021]. The entries of the LUT are the simulated reflectance spectra $\boldsymbol{x}_i$ and the leaf and canopy biophysical parameters $\boldsymbol{y}_i$. Inverting the PROSAIL model with a LUT is performing KNN with a reflectance spectra $\boldsymbol{x}_q$ as a query to retrieve an estimated set of biophysical variables $\boldsymbol{y}_q$.

3.2.3 Machine learning regression

3.2.3.1 Random forests

Random forests (RF) [Ho, 1995] are an *ensemble learning*[22] methodology based on training multiple *decision trees*. Usually, the prediction of RF is chosen as the average prediction of the decision trees. Additionally, the dispersion of individual tree predictions can provide a measure of predictive uncertainty.

Decision trees are a *supervised* ML algorithm based on tree graph[23] models where each node represents a decision (a computation, an assignment) based on a feature. Decision trees and RF can be used for classification and regression problems. The predictions output by the decision trees are contained by nodes that have no children node, that are called the *leaves*[24]. Decision tree learning is about building the tree structure and attributing decision to each node.

[21] k-dimensional trees, or k-d trees are data structures used for organizing and searching points in multi-dimensional spaces efficiently.

[22] Ensemble methods are about using multiple predictive algorithms to produce a global prediction with better performance than individual algorithms. This concept that usually belong to ML or statistics, can be illustrated by the *Wisdom of the crowd* theory, which states that for some problems, the collective opinion of a diverse independent group of individuals is better than that of a single expert.

[23] A tree is an undirected (i.e. edges are not directional and the relation between two nodes is symmetric) graph, for which any two nodes are connected by at most one edge (i.e. acyclic).

[24] They are *degree-1 nodes*.

3.2.3.2 Support vector machines

Support vector machines (SVMs) [Cortes and Vapnik, 1995] are *supervised* learning non-linear models that are commonly used for classification [Camps-Valls and Bruzzone, 2005], although a variant for regression exists: support vector regression (SVR) [Smola and Schölkopf, 2004]. The SVR objective is to fit a model f to some training data $\mathcal{D}_T = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$. For regression, the goal is that this function f maps predictors $\boldsymbol{x}_i$ to regressands $\boldsymbol{y}_i$ of $\mathcal{D}_T$ with a deviation of at most ε (i.e. errors are tolerated up to ε), and that is as *flat* as possible (i.e. a function that is close to a hyperplane). The SVR objective is to find a *maximum margin solution*. There are two basic ideas for SVR.

- *Support vector expansion*: the regression function $f(\boldsymbol{x})$ is expressed as a linear combination of the dot products $\langle \boldsymbol{x} \cdot \boldsymbol{x}_i \rangle$ between $\boldsymbol{x}$ and predictors of the training data-set $\boldsymbol{x}_i$, which are called *support vectors*.
- *Non-linear mapping*: To introduce non-linearity to the fitted function f w.r.t. $\boldsymbol{x}$, input data $\boldsymbol{x}$ are projected into a *feature space* using a non-linear mapping Φ. Support vectors are also mapped with Φ, and the regression function is expressed as a linear combination of the dot products of the mapped features and mapped support vectors $\langle \Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{x}_i) \rangle$ instead of the dot product in their original space. This dot product of mapped features $k(\boldsymbol{x}, \boldsymbol{x}_i) = \langle \Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{x}_i) \rangle$ are called *kernel functions*. It is actually not necessary to compute the coordinates of the regressors in the feature space, nor to explicitely formulate the mapping Φ, there is only a need to define a kernel function.

SVR is useful for high dimensional data, however they do not handle well training with a large data-set (*big data* regime), and are not as efficient and fast as other methods. Additionally, SVR are ill-suited to multi-variable estimations. Nonetheless, SVR has good performance when there are few labeled data points available compared to other supervised methods such as deep learning (DL). Furthermore, it is possible to extend the formulation of SVR to the multi-output case [Tuia et al., 2011].

3.2.3.3 Neural Network Regression

Neural networks are ML models that process information through successive transformations through a computation graph organized in *layers*, mimicking the functioning of a living brain's interconnected neurons. They are trained by adjusting the connections weights between the artificial neurons, by using the gradients of an objective function, that are *back-propagated* from the last (output) layer to the innermost layers [Rumelhart et al., 1986]. Neural networks can be used in supervised and unsupervised setting. However for performing regression, neural networks are trained in a supervised way. They are *universal approximators*: provided that the network complexity is high enough, neural networks can approximate arbitrary functions. As such, neural networks can be used in regression problems to approximate a regression function f between regressors $\boldsymbol{x}$, used as input of the model, and targets $\boldsymbol{y}$ output by the model. The loss function of such regression neural networks typically penalizes a distance between the model output $\boldsymbol{y}$ and a true value $\boldsymbol{y}_T$ of a training data-set, for instance squared errors:

$$\mathcal{L}(f, \boldsymbol{x}_T, \boldsymbol{y}_T) = \|f(\boldsymbol{x}_T) - \boldsymbol{y}_T\|_2^2 . \tag{3.27}$$

Neural networks will be discussed more in depth in section 3.3.

3.2.3.4 Bayesian methods

Bayesian approaches are based on the formulation of the quantities of interest, $(\boldsymbol{x}, \boldsymbol{y})$ as random variables $(\mathbf{x}, \mathbf{y})$. As a consequence the models that represent the relationships between these variables are probabilistic in nature: they are probability distributions. Provided

a forward model $p(\boldsymbol{y}|\boldsymbol{x})$ (a so-called *likelihood* model), performing inversion is about retrieving $p(\boldsymbol{x}|\boldsymbol{y})$ (the so-called *posterior distribution*). Bayesian approaches can be broadly divided into two categories:

- Exact posterior approximation with sampling strategies, best exemplified with Markov Chain Monte Carlo (MCMC),
- Approximate posterior estimation, with variational inference.

Bayesian approaches will be discussed at length in Chapter 6, with an emphasis in variational inference.

3.3 Regression with deep learning

deep learning (DL) is an approach of ML that is based on ANN models. This section is dedicated to the presentation of ANN, with well-known classical architectures (subsection 3.3.1), and of their training, with an emphasis to some technical specificities of the ANN models used in this Ph.D.

3.3.1 Artificial neural networks

Artificial neural networks (ANNs), or more simply, neural networks or neural nets are models whose objective is to approximate arbitrary mappings between inputs data $\boldsymbol{x}$ and outputs $\boldsymbol{y}$ (also called *labels*). As their name suggests, ANN were inspired by how information is processed in biological neural networks. Biological neural networks are the groups of neuron cells that are interconnected with *synapses*, that transmit signals chemically (with neuro-transmitters) or electrically. The transmission of a signal from a neuron[25] to other connected downstream neurons is dictated by the synthesis of the signal received by the neuron from connected upstream neurons. Like in these neuronal networks, ANN are made of interconnected *artificial neurons* that mimic the behavior of true neurons: they are abstract computation units that transmit an output value to other neurons as a function of input values provided by upstream neurons. ANN are not as complex as biological neural networks.

As will be explained below, data processing by ANN are essentially operations on matrices. ANN computations (both training and inference) are made significantly more efficient and fast when they are performed on specialized devices such as graphical processing unit (GPU). This is because matrix operations can be parallelized (i.e. divided into a set of independent operations), and GPU are devices that are designed for parallel computations.

3.3.1.1 Multi-layer perceptrons

The artificial neurons are arranged in *layers*, that form a more or less *deep* stack (thus the denotation of *deep* learning). ANN can be represented by a computational graph that models the order of the flow of individual operations, for which the vertices (also called *nodes*) are the artificial neurons and the edges are the neuron connections (see Figure 3.2). The layers between the input and output layers are called the *hidden layers*. For an ANN with n hidden layers (three in the simple example shown), input layer nodes are denoted x_i with $i \in [\![1, k_1]\!]$, the output layers nodes are denoted y_i with $i \in [\![1, k_{n+1}]\!]$ and the nodes of the j-th hidden layer are denoted $h_{i,j}$ with $i \in [\![1, k_j]\!]$. These input and output layers respectively encode the coordinates of an input $\boldsymbol{x}$ and an output $\boldsymbol{y}$.

ANN architectures can be broadly divided into two types:

- *Feed-forward* neural networks only allow the outputs of a given layer to be cast to the inputs of downstream layers. The flow of information is uni-directional, and the

[25] The neuron firing

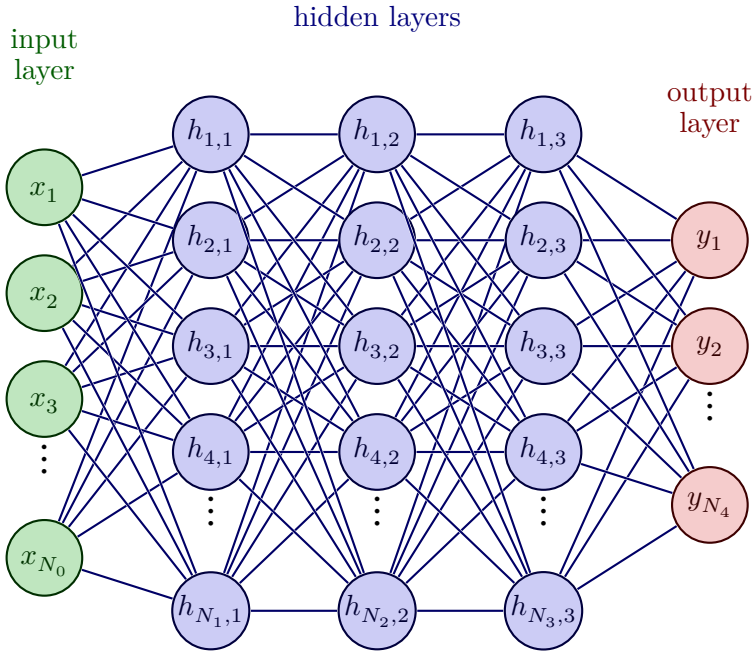


Figure 3.2: Graph of a feed-forward artificial neural network with three hidden layers.

computational graph is acyclic. In signal processing, such a neural network is analogous to a finite impulse response (FIR) filter.

- *Recurrent* neural networks, on the contrary, allow the output of a given layer to affect the nodes of upstream or current layers. The computational graph contains cycles, and the flow of information is bi-directional. They are analogous to infinite impulse response (IIR) filters, and exhibit dynamic behaviors. They also have some internal memory, that stores information from one input to another, as such they are suited to sequential data.

The example shown in Figure 3.2, along with all ANN studied in this Ph.D. are feed-forward ANN. A layer for which all neurons are connected to all neurons of the parent layer is a *dense* or fully-connected layer. An ANN model that consists only of dense layers is also qualified as *fully connected*.

As stated previously, the artificial neurons that occupy the nodes of ANN mimic the behavior of biological neurons. They produce an output that is function of the signal received from upstream neurons, and transmit this output to downstream neurons. For artificial neurons, the signals transmitted between neurons are real numbers. The output $h_{i,l}$ of the i-th neuron of layer l is a non-linear transformation φ of a linear combination of its N_{l-1} inputs $h_{k,l-1}$ (see Equation 3.28 and Figure 3.3).

$$h_{i,l-1} = \varphi\left(\sum_{k=0}^{N_{l-1}} w_{i,l,k} h_{k,l-1}\right) = \varphi\left(\boldsymbol{w}_{\boldsymbol{i,l}}^{\top} \boldsymbol{h}_{l-1}\right) \tag{3.28}$$

The linear combination is weighted by the coefficients $w_{i,l,k}$, that are the parameters optimized in an ANN. For a given architecture, an ANN is entirely determined by these parameters. It can be noted that the summation in Equation 3.28 is set to begin at $k = 0$, despite the layer l nodes being indexed from 1 to N_l. This is to account for an additional term $b_{i,l}$ which is commonly introduced in the sum, the *bias*. As a convention $h_{0,l-1} = 1$ so that $b_{i,l} = w_{i,0,l} h_{0,l-1}$.

The non-linear function φ is called an *activation* function. Activation functions need to be differentiable almost everywhere so that gradients can be computed for updating the weights

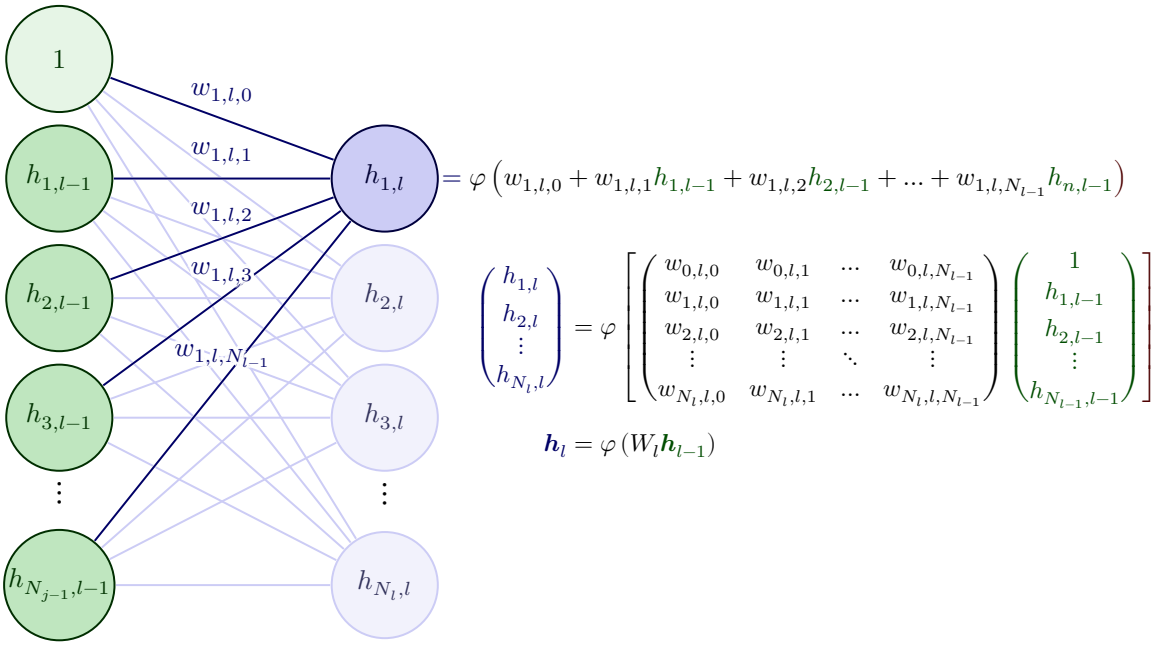


Figure 3.3: Activation of an artificial neuron in a neural network as a function of the previous layer outputs. The activation of all neuron in a layer can be expressed as a vector equation.

W. There is a wide variety activation functions used in the literature [Szandała, 2021], with different properties regarding optimization and the propagation of gradients throughout the model (see subsubsection 3.3.2.1). Here are some of the most commonly used:

- the sigmoid (or logistic) function: $\forall x \in \mathbb{R}, \sigma(x) = \dfrac{1}{1+\mathrm{e}^{-x}}$,
- the hyperbolic tangent $\forall x \in \mathbb{R}, \tanh(x) = \dfrac{\mathrm{e}^{x}-\mathrm{e}^{-x}}{\mathrm{e}^{x}+\mathrm{e}^{-x}}$,
- the rectified linear unit (ReLU): $\mathrm{ReLU}(x) = \max(0, x) = x \cdot \mathbb{1}_{[0,+\infty[}(x)$.

Originally, the formulation of artificial neurons used a Heaviside activation function (or binary step function) $\varphi(x) = \mathbb{1}_{[0,+\infty[}(x)$ [McCulloch and Pitts, 1943], to mirror the state of biological neurons that were either activated or inactivated. This type of neuron is called a *perceptron*. Multi-layer perceptron (MLP) is actually a misnomer, since it designates fully connected feed-forward ANN with any kind of activation function, and not just Heavyside activation.

3.3.1.2 Convolutional neural networks

Convolutional neural networks (CNNs) are ANN with a distinct architecture, that compute activations of the neurons differently than MLP. These architectures are suited to data that have a grid-like structure, such as time series (1-dimensional data) or images (2-dimensional data). In a CNN, the layers are *convolutional*: the output of a particular node i of a layer l (the *feature map*) is the result of the application of a convolution filter (called a *kernel*) K with the activation of the previous layer neuron (called *input* in this context). For each layer node, the output is a discrete convolution[26] between the input and the kernel:

- in 1-D (see Figure 3.4),

$$h_{i,l} = (K * \boldsymbol{h}_l)_i = \sum_r h_{i+r,l}\boldsymbol{k}_r, \tag{3.29}$$

[26] Actually, CNNs do not carry out convolutions, but rather the cross-correlation operation, that uses a flipped kernel/input. The 2-D convolution is actually defined as: $h_{i,j,l} = (K * H_l)_{i,j} = \sum_r \sum_s h_{i-r,j-s,l} K_{i,j}$.

- in 2-D (see Figure 3.5),

$$h_{i,j,l} = (K * H_l)_{i,j} = \sum_r \sum_s h_{i+r,j+s,l} K_{r,s}. \tag{3.30}$$

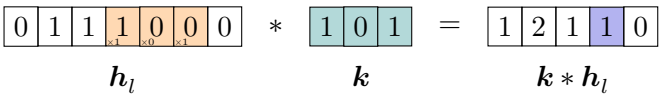


Figure 3.4: 1-D convolution in CNN.

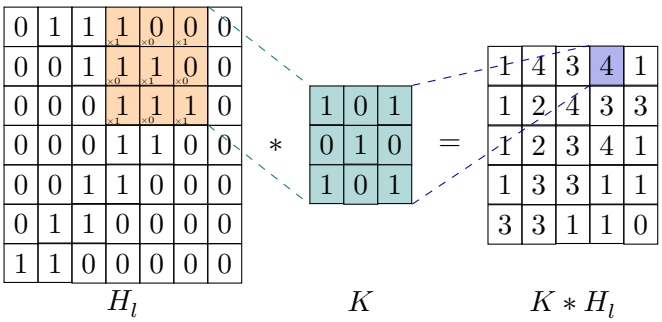


Figure 3.5: 2-D convolution in CNN.

The learnable weights of CNNs are the coefficients of the kernel.

As illustrated in Figure 3.4 and Figure 3.5, kernels are typically of lower size than inputs. This introduces a sparsity in learnable weights for each filter, and a sense of locality, since kernel only processes a small domain of the input at a time. One particular feature of CNNs is that kernels are *shared* among neurons of the same layers, meaning that coefficients of kernels are involved in the activation of all nodes of the layers. This reduces further the number of learnable weights.

Convolutional layers in CNN can be seen as a matrix multiplication[27], and can thus be compared to operations in dense layers. A convolutional layer is equivalent to a dense layer for which all neurons share the same weights, which are zero everywhere except for a few, i.e. a very sparse weight matrix W_h.

The dimensions of a kernel are usually referred as *width* and *height* in 2-D (usually with value of 1, 3 or 5). It is also possible to apply the convolution with a *stride*, which is the number of rows or columns traversed per *slide* of the kernel along the input. The size of the convolution result is a function of the input size, of the kernel size and the stride. To avoid having the convolution result size smaller than the input size, *padding* is commonly applied. It is the augmentation of the input with additional row and columns on the edges, e.g. with zeros (*zero-padding*).

Finally, data is commonly arranged in c *channels* (e.g. $c = 3$ with red green and blue for pixels of an RGB image), which adds a dimension to the data. To process such data, stacked kernels can be used, each associated to a given channel, and the output of the convolution is one channel whose elements are the sum of the convolutions of each kernel with its respective input channel. For instance, with an input data of size $w \times h \times c$, a stack of c kernels of size $w_k \times h_k$ (or a $w_k \times h_k \times c$-sized kernel, i.e. a 3-D *tensor*) is used. To produce multiple channels as output of a convolutional layer, several of these kernels can be used, and stacked in another dimension (i.e. making a 4-D tensor).

[27] Discrete convolution can be expressed as a product with a *Toeplitz matrix*, which have constant diagonals.

3.3.1.3 Residual connections

As will be discussed in subsubsection 3.3.2.1, ANN parameters are updated using the gradients of neuron activations. Unfortunately, a well known problem of deep neural networks is that the gradient that is computed within the innermost layers of the network can have a very low value, effectively not updating these layers: this is the *vanishing gradient* problem. This leads to performance stagnation or even degradation when the model depth increases.

Residual neural networks (also called ResNets) are a type ANN that incorporate *residual connections* (or *skip connections*), and are instrumental to the performance of the seminal ResNet architecture [He et al., 2015a], that enabled to use hundreds of layers effectively. A residual connection combines with a function g the output $\boldsymbol{h}_r$ of a layer r to the output of a downstream weight layer s and bypasses one or several layers denoted $f_{\boldsymbol{\theta}}$ (see Figure 3.6). An activation function φ_s is applied to the result of this combination rather than the output of the weight layer s:

$$\boldsymbol{h}_s = \varphi_s \left(g \left(f_{\boldsymbol{\theta}} \left(\boldsymbol{h}_r \right) + \boldsymbol{h}_r \right) \right) \tag{3.31}$$

Usually, the function g of the residual connection is either a concatenation operation[28]:

$$\boldsymbol{h}_s = \varphi_s \left(\begin{bmatrix} f_{\boldsymbol{\theta}} \left(\boldsymbol{h}_r \right) \\ \boldsymbol{h}_r \end{bmatrix} \right) \tag{3.32}$$

or, as used in this work, a summation:

$$\boldsymbol{h}_s = \varphi_s \left(f_{\boldsymbol{\theta}} \left(\boldsymbol{h}_r \right) + S_r \boldsymbol{h}_r \right), \tag{3.33}$$

with S_r a matrix for performing linear projection. If the dimensions of $\boldsymbol{h}_r$ and $f_{\boldsymbol{\theta}} \left(\boldsymbol{h}_r \right)$ match, S_r is usually simply the identity matrix, and no extra parameter is introduced by the skip connection. A sub-neural network with a skip connection is commonly called a *residual block*.

Residual connections provide a shortcut for information processing between the input and output of a neural network. When training a ResNet, skip connections help to produce a meaningful model output faster, and the short-circuited layers then gradually learn to model residuals. This also helps mitigating the vanishing gradient problem: a part of the gradients can flow directly through the skip connections backwards to the innermost layers, as will be illustrated in Chapter 8.

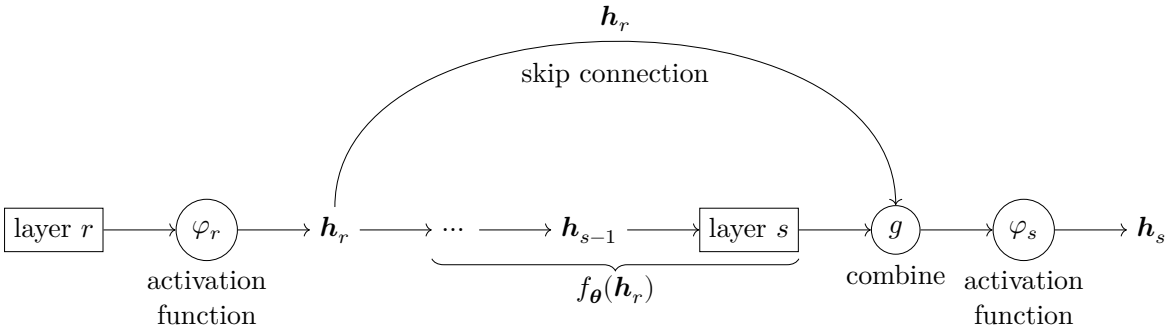


Figure 3.6: Skip connection between two layers r and s within a neural network.

3.3.2 Neural Network training

3.3.2.1 Automatic differentiation and gradient propagation

As seen in subsection 3.3.1, the parameters to optimize in ANN are the *weights*, i.e. the coefficients of the layers (kernel coefficients for CNN). Like many other optimization problems,

[28] Concatenation residual connections as are at the basis of most *U-net* [Ronneberger et al., 2015] implementations, which use skip connections with convolutional layers for processing jointly different layers outputs representing information at different resolutions of the input data.

these parameters $\boldsymbol{w}$ are optimized by minimizing a loss function $\mathcal{L}$ such as Equation 3.27, and performing gradient descent (see subsubsection 3.2.1.3):

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}, \tag{3.34}$$

with the hyperparameter α the lr that tweaks the size of each gradient update. The weights in the successive layers of an ANN are updated thanks to a gradient estimation method called *backpropagation*. Gradient backpropagation makes use of Leibniz derivation chain rule, that computes the gradients of the neurons output in a layer w.r.t. its inputs i.e. the output of the neuron in the previous layer. The chain rule states that for n differentiable functions f_i, the function that is their successive composition $f = f_1 \circ f_2 \circ ... \circ f_i \circ ... \circ f_n$ is also differentiable and can be computed as:

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\mathrm{d}f_n}{\mathrm{d}x} \prod_{i=1}^{n-1} \frac{\mathrm{d}f_i}{\mathrm{d}f_{i+1}}. \tag{3.35}$$

Depending on the neural network architecture and several optimization hyper-parameters, efficient update of the model weights with the chain rule can fail due to two phenomena: *vanishing gradients* and *exploding gradient*

In other ML algorithms that use gradient descent, the gradients of the objective function and the update of the parameters may be expressed analytically and implemented manually. However with deep neural networks, with very large number of neurons and interconnections, this is intractable. The advent automatic differentiation has enabled easy and flexible computation of gradients within DL models. Automatic differentiation is a specific family of techniques that compute derivatives through accumulation of values during code execution to generate numerical derivative evaluations rather than derivative expressions [Baydin et al., 2018].

ANN are commonly optimized using *mini-batches*, that is, the update of the model parameters are performed for the gradients of the loss function aggregated (e.g. summed, averaged) over a subset of training samples of size intermediate between the full training data-set and 1. *Batch* gradient descent computes the loss of the model for each sample in the training dataset, but only updates the model after all training samples have been evaluated. A common assumption is that this allows to update the model using the best possible approximation of the true gradient, thus making the convergence stable and efficient. However this approach is actually slower than other gradient descent approaches (see below), and the model might only reach a sub optimal accuracy when the lr is not well chosen [Wilson and Martinez, 2003]. Furthermore, this method can be memory intensive and become intractable for large data-sets. At the other end of the spectrum *stochastic gradient descent* computes the gradient of the loss and updates the model for each training sample. The model is updated very frequently, which can lead to fast convergence in some cases, and the noisy update can help avoiding getting stuck in local minima. However backpropagating gradients and updating the weights more frequently leads to increased computational cost and slower training, and the noisy gradient descent can make optimization unstable. Mini-batch gradient descent, which introduces the *batch-size* as and additional hyper-parameter, makes a trade-off between these two regimes.

3.3.2.2 Learning rate scheduling

An *optimizer* is an algorithm that updates the weights of an ANN by using a gradient descent variant. Most of these optimizers require the specification of the lr as a hyperparameter. This learning rate may stay constant throughout training, however it can be interesting to have it dynamically change, so that optimization is more efficient, i.e. to ensure convergence, stability of the solution, and to avoid being stuck in local minima.

Some algorithms called *learning rate schedulers* propose different strategies for updating the lr across training epochs t. One of the most basic scheduler is the exponential scheduler:

$$\alpha(t) = \alpha_0 \mathrm{e}^{-\lambda t}. \tag{3.36}$$

In this Ph.D., using *reduce-lr-on-plateau* is considered. It is a piece-wise constant learning rate scheduler, that decays the lr α by a factor η when an choice metric (e.g. the validation loss function) doesn't continue decreasing (with a threshold ε) for a preset time delta (called *patience*). α is allowed to decay until a minimal value $\alpha_{\min}$. The rationale is that if the model doesn't improve after a while, it may be because the gradient updates overshoot the objective function minima. When this happens, reducing the lr can enable to resume the loss function minimization. In this Ph.D., a slightly modified version of this scheduler, which is named cyclical plateau reduction (CPR) is used. When the lr α reaches the minimum values $\alpha_{\min}$, it is reset to the intial value α_0, and the reduce-lr-on-plateau is restarted again. This produces an annealing of the lr, that is proposed to help move away from local minima that low lr cannot help escaping.

Some *adaptive* optimizers, such as the very popular Adam (Adaptive moment estimation) [Kingma and Ba, 2015] propose to forego the use of schedulers by incorporating per-parameter dynamic lr. It can be noted that the Adam optimizer requires a global lr parameter, although optimization is hopefully more robust w.r.t. to its value than other optimizers. It is still unclear if using adaptive optimizer like Adam along with lr scheduler is beneficial or not. Although there are heuristics for choosing an optimization scheme, each ML problem has its own specificities that makes finding all-purpose strategies difficult. In this Ph.D. the Adam optimizer is used jointly with the CPR scheduler.

3.3.2.3 Model initialization

Before an update of the model with training can take place, the weights of the layers must be initialized: this is *model initialization*. Initialization has a significant impact over training and final performance. Ideally, an initialized model would have weights such that the objective function is near a global minima. However in most cases, model weights are initialized randomly, for instance using a centered Gaussian distribution. The variance of such a distribution must be appropriate: too low and most weights have near zero value, and training may be plagued by vanishing gradients, and too high and exploding gradient may occur. A popular initialization scheme that aim at mitigating those effects is the *Xavier initialization* (or *Glorot initialization*), which uses a centered uniform distribution whose symmetrical bounds depend on the size of the ANN layers [Glorot and Bengio, 2010]. In the Pytorch library, *Kaiming He initialization* (weights are distributed with a zero-mean Gaussian with variance $\sqrt{\frac{2}{K}}$ with K the size of the layer) is performed by default [He et al., 2015b]

In this Ph.D., an additional initialization step is performed before pursuing training. Even with proper weight random initialization, convergence and performance can depend on the model instance [Picard, 2021]. As such, to avoid training a model that is plagued with "bad luck" with a bad initialization, the hereby named multiple initialization and best instance training (MIBIT) strategy is applied. MIBIT is about *pre-training* several model instances for very few epochs (e.g.) with relatively high constant lr, and retaining only the model instance that has best minimized the loss function. This model instance is then chosen as "initialization" to start the true training.

3.3.2.4 Data-set splitting

For training ANN, or ML models in general, a good practice is to split the available data into three sub-data sets for distinct purposes. These are the *training* $\mathcal{D}_{train}$, *validation* $\mathcal{D}_{valid}$ and *testing* $\mathcal{D}_{test}$ data-sets.

The training data-set is dedicated to calibrating the model parameters (weights). In ANN, gradient back-propagation is performed with this data-set.

The validation data-set serves two purposes:

- detecting *over-fitting* and
- tuning hyper-parameters[29].

The testing data-set (or test set) is designed to assess the model performance after training, and after hyper parameters have been selected. Ideally this data-set should contain samples that enable to evaluate the model in the various regimes it can face in real applications. It can be noted that in this work, an additional data-set is used to evaluate the model performance: an "evaluation" data-set or "in-situ" data-set. This data-set can occur:

- When the data used to train the model is of different nature than the data it will be applied on. In Chapter 5, simulated data is used to train the model, but performance is assessed on and in-situ data-set with real data.
- When the training task is different from the intended task of the model (i.e. a *proxy task*). In Chapter 8, the model is trained in a self-supervised manner to reconstruct input data, but in-situ data is used to assess how close learned representations are from measurements.

Overfitting The first use of the validation data-set is to provide an unbiased evaluation of a model fit. A ML model *overfits* when it fits too closely the training data, even modeling its noise[30]. As a consequence, it is unable to accommodate new data and performs poorly outside of training samples. Overfitting can be detected by computing the loss function on validation samples: it occurs when the *validation loss* behaves differently than the *training loss* (i.e. when the validation loss increases while the training loss decreases). Overfitting is a consequence of the model being too complex w.r.t. the size of the training data (e.g. a data-set with thousands of samples is not enough to train an ANN with millions of parameters). It can be mitigated by reducing the model complexity, increasing the data-set size, and with *regularization* (e.g. additional loss terms that penalize the weights, such as *ridge regularization*, see subsubsection 3.2.1.1).

Cross-validation In statistical analysis the splitting of a data-set into a training subset and a validation subset for assessing how models generalize with out-of-sample data is known as *cross-validation*. Cross-validation actually goes further than that. Since the performance of the trained model can depend on the specific training and validation subsets, several iterations of the model are tuned using different splitting. Validation results between all model instances can then be averaged to assess how the model generalizes to unseen samples, independently from said samples.

In k-fold cross-validation, the data is shuffled, then partitioned into k subsets with equal sized, called *folds*. k instances of a model are trained on k different associations of these folds as training and validation sets: for a given iteration, $k-1$ folds are used to constitute the training set, and the remaining one is used for validation.

[29] Hyper-parameters are parameters that are not tuned during the training phase. They are not learned, or updated with gradient descent.

[30] Conversely *underfitting* occurs when the models is unable to accurately represent the data, and performance is poor as a consequence. It occurs when the model is too simple, with badly selected or scaled input features, or with excessive regularization.

3.4 Conclusion

In this chapter, different methods for performing model inversion with regression have been introduced, with an emphasis on ANN. The DL methodologies discussed here are used throughout the applications performed in this Ph.D., presented in Chapter 5, Chapter 8 and Chapter 9. Chapter 5 in particular presents the shortcomings of using supervised regression ANN for performing the inversion of the PROSAIL model (Chapter 4). Specifically, it will be shown that regression performances are very sensitive to the choice of the unknown sampling distributions for generating training samples.

Chapter 4

Spectral models of vegetation

Contents

Vegetation has specific spectral features that distinguishes it from other occurring objects on continental surfaces. It has high absorbance in the visible spectrum so as to enable energy absorption for photosynthesis, and high reflectance in the near infra-red (NIR). Just like there is a wide variety of vegetation, there are just as numerous spectral responses possible. The interaction of light with vegetation is governed, at a smaller scale by its chemical components, and at a larger scale, by its spatial organization into a canopy. What is a vegetation made of? How is it organized? These aspects enable to build a representation of a vegetation that characterizes its state. With light being reflected by vegetation towards a remote sensor, it is possible to access the underlying bio-physical properties of the vegetation, as a representation. In this chapter, radiative transfer models (RTMs) that enable to link bio-physical properties of vegetation and observed light are introduced. Specifically, the PROSPECT leaf radiative transfer model (RTM) and the SAIL canopy RTM are respectively presented in section 4.1 and section 4.2. PROSAIL, the composite model of PROSPECT and SAIL is detailed in section 4.3. section 4.4 explains how the PROSAIL model can be used for simulating Sentinel-2 (S2) reflectance bands. Finally, section 4.5 details the particular implementation of PROSAIL developed for the experiments in the remainder of this manuscript.

4.1 The PROSPECT leaf model

Leaf models aim at characterizing the interaction of incident light onto a leaf. The optical properties that are sought after are the leaf reflectance ρ, the transmittance τ and absorbtance A[1], which are respectively defined as the ratios of the reflected (Φ_R), transmitted (Φ_T), absorbed (Φ_A) over the incident (Φ_I) radiant flux (see Equation 4.1). The conservation of energy fluxes (see Equation 4.2) implies that $\rho+\tau+A=1$, and enables to derive one quantity by knowing the other two. It can be noted that this optical model ignores the *chlorophyll fluorescence* phenomenon, in which the chlorophyll pigments can re-emit a fraction of the absorbed light flux [Maxwell and Johnson, 2000].

$$\rho = \frac{\Phi_R}{\Phi_I}, \quad \tau = \frac{\Phi_T}{\Phi_I}, \quad A = \frac{\Phi_A}{\Phi_I} \tag{4.1}$$

$$\Phi_R + \Phi_T + \Phi_A = \Phi_I \tag{4.2}$$

As the light interaction with the leaves depends on the wavelength λ, the reflectance $\rho(\lambda)$, the transmittance $\tau(\lambda)$ and absorbtance $A(\lambda)$ are spectra.

Retrieving $\rho(\lambda)$, $\tau(\lambda)$ and $A(\lambda)$ requires characterizing the internal structure of the leaf, and the light propagation inside it. These quantities are described with the radiative transfer equations established by the seminal work in Chandrasekhar [1960]. They are a set of integral, non-linear equations that can not be solved in the general case, and are still the object of a lot of studies today. The various leaf RTM assume different physical properties and use simplifying assumptions to solve the radiative transfer equations. An overview of the different leaf optical models is provided in Féret [2009]; Jacquemoud and Ustin [2008].

4.1.1 The plate model

The PROSPECT leaf RTM [Jacquemoud and Baret, 1990], models the leaves as identical parallel plate stacks, separated by air layers (see Figure 4.1). These leaf plates are a *turbid* medium, which contain randomly distributed elements that interact with the light. All plates are described simply by a refraction index n and a transmission coefficient θ, but have all a distinct reflectance ρ and transmission coefficient τ. In PROSPECT, the light fluxes are

[1] not to be confounded with *absorbtion*, which is a phenomenon in which light is absorbed, whereas absorbtance is a quantity that measures a ratio of absorption.

assumed to be isotropic between the different layers. Consequently, the reflectance and transmittance are described as *hemispherical*, i.e. integrated over the half-space.

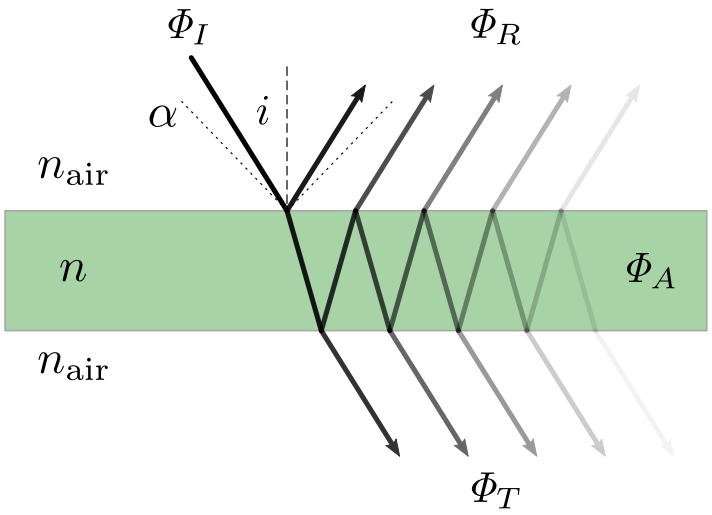


Figure 4.1: Representation of the multiple reflections, refractions and absorptions of an incident beam of light within a turbid leaf plate. i is the radiation incidence angle, with an upper bound α. The leaf and ambient air respective have n and n_{air} as refractive index.

The transmission coefficient is linked to the biophysical content of the leaf, and can be retrieved from the absorption coefficient k of the leaf according to the plate model [Allen et al., 1969; Jacquemoud and Baret, 1990] with Equation 4.3.

$$\theta - (1-k)\,\mathrm{e}^{-k} - k^2 \int_k^{+\infty} \frac{e^{-x}}{x}\,\mathrm{d}x = 0 \tag{4.3}$$

The leaf structure parameter N designates the number of such plates in the stack that makes up a leaf. The radiant flux that propagates in the $N-1$ intermediate air layers between the plates is assumed to be diffuse, and to have no specific direction. The hemispherical reflectance and transmittance coefficients in these layers are identical, and commonly denoted as ρ_{90} and τ_{90}. The first (top) plate receives the incident light, which is not isotropic, but its direction i is assumed to be contained inside a solid angle with a half-apex angle α. The reflectance and transmittance of this layer is ρ_α and τ_α. This angle is linked to the leaf surface roughness. As the incident light is described with a cone, the modeled reflectance (and transmittance) of the whole leaf are described with a *conical-hemispherical reflectance (transmittance) function* [Schaepman-Strub et al., 2006].

The four quantities N, $n\,(\lambda)$, k, α characterize entirely the leaf model, and enable to derive the leaf reflectance ρ_l, transmittance τ_l, and absorbance A_l. In practice, the parameters $n\,(\lambda)$ and α are calibrated from leaf measurements and kept constant within a version of PROSPECT, whereas N and k are considered variable. A representation of the PROSPECT plate model is shown in Figure 4.2.

In the generalized plate model [Allen et al., 1969, 1970] on which PROSPECT is based, N initially designated an integer number of leaf layers. Using the work of Stokes [1862], the leaf structure parameter is extended in PROSPECT to a real number $N \geq 1$. This is conceptually more difficult to represent and understand, however, this makes the model continuous and more easily inversible. A continuous leaf structure parameter may be understood as an average number of air/leaf cell interfaces within leaves. Also, a real N can be used to model the physical properties of leaf with different geometries, rather than just controlling the transversal light propagation. For instance, monocotyledon plants, such as most cereals, are characterized with a relatively low structure parameter $N \in [1, 1.5]$ [Féret et al., 2021]. This is due to a simpler, more compact tissue layout inside the leaves. Dicotyledon plants, with

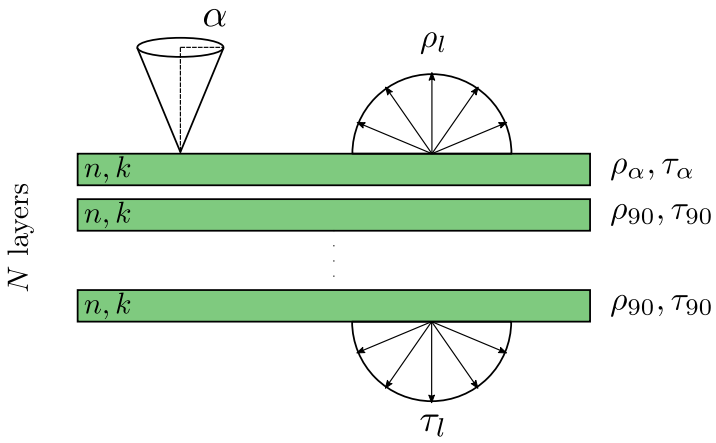


Figure 4.2: Hemispherical leaf reflectance ρ_l and transmittance τ_l in the N layer model of PROSPECT. Each layer of the model has the same refraction index n, absorption coefficient k, reflectance ρ_{90} and transmittance τ_{90}. The first layer has a different reflectance ρ_α and τ_α, with α the maximum radiation incidence angle.

more leaf complexity and inter-cellular air spaces, such as legumes, have higher parameter $N \in [1.5, 3]$.

4.1.2 Modeling the leaf contents

The spectral absorption coefficient $k(\lambda)$ is the sum[2] of the contribution of the leaf plate biochemical contents:

$$k(\lambda) = \sum_i k_i(\lambda) = \sum_i K_i(\lambda) C_i, \tag{4.4}$$

with $k_i(\lambda)$ the absorption coefficient of a component i. $k_i(\lambda)$ is the product of the *specific absorption coefficient* $K_i(\lambda)$ and the concentration[3] C_i of the component i in the plate. Finally, the specific absorption coefficients are experimentally calibrated, then kept constant. The only variables remaining are the leaf components contents C_i.

Between the various versions of PROSPECT, the most notable updates lie in the introduction of different components i whose absorption is taken into account. The first version of PROSPECT [Jacquemoud and Baret, 1990] only considered the absorption contribution of two leaf components: the water content and a leaf pigment content, corresponding to the chlorophyll a and b. The water content C_w is associated with the specific absorption coefficient $K_w(\lambda)$, and the chlorophyll content C_{ab} to $K_{ab}(\lambda)$. In this first version of PROSPECT, $K_w(\lambda)$ and $K_{ab}(\lambda)$ are sampled with a 5 nm resolution over the range 400–2500 nm.

The next early versions of PROSPECT improved the spectral resolution to 1 nm, and introduced the spectral contribution of the dry matter content C_m in leaf cell wall molecules [Jacquemoud et al., 2000]. The version 2 of PROSPECT actually distinguishes protein content C_p and cellulose and lignin content C_c, whereas the following versions use a dry matter content that encompasses both. In PROSPECT-4 all photosynthetic pigments are assumed to be chlorophyll, whereas PROSPECT-5 differentiates chlorophylls from carotenoids [Feret et al., 2008]. In PROSPECT-5, a *brown pigment* content C_{brown} is also introduced. Brown pigments account for the spectral influence of different bio-molecules, such as tannins and polyphenols [Bing Lu and He, 2021]. PROSPECT-D adds the contribution of anthocynanins [Féret et al., 2017, 2019], and PROSPECT-PRO incorporates nitrogen-based proteins and carbon-based constituents [Féret et al., 2021]. The leaf content taken into account by each PROSPECT version is summarized in Table 4.1.

[2]The absorption of a medium is *additive* of the absorption of its components.
[3]Usually a content per unit area of leaf.

Table 4.1: Leaf component taken into account in the different PROSPECT versions

Leaf Component	Chlorophyll $a+b$	Water	Protein	Cellulose & lignin	Dry matter	Carotenoids	Brown pigments	Anthocyanin	Nitrogen-based proteins	Carbon-based constituents
Content	C_{ab}	C_w	C_p	C_c	C_m	C_{car}	C_{brown}	C_{ant}	C_{prot}	C_{cbc}
Unit	µg cm^{-2}	cm	g cm^{-2}	g cm^{-2}	g cm^{-2}	-	µg cm^{-2}	nmol cm^{-2}	g cm^{-2}	µg cm^{-2}
Specific absorption coefficient	K_{ab}	K_w	K_p	K_c	K_m	K_{car}	K_{brown}	K_{ant}	K_{prot}	K_{cbc}
PROSPECT Version										
V1	✓	✓								
V2.01	✓	✓	✓	✓						
V3.01	✓	✓			✓					
V4	✓	✓			✓					
V5	✓	✓			✓	✓	✓			
D	✓	✓			✓	✓	✓	✓		
PRO	✓	✓			✓	✓	✓	✓	✓	✓

The specific absorption spectra that are used throughout the different PROSPECT versions may be subject to a re-calibration. The calibrations are updated with the improvement of calibration data-bases, and of estimation algorithms. Also, the re-definition of the leaf constituents that are taken into account with each absorption coefficient also requires updating the reference spectra (e.g. the distinction of carotenoid from chlorophyll pigments as a new leaf content in PROSPECT-5).

Using PROSPECT-5, the reflectance, transmittance and absorbtance of a maize leaf are simulated and displayed in Figure 4.3. The values of the leaf contents and N used as input are taken from some LOPEX93 data [Hosgood et al., 1993], which is included in the available MATLAB implementation of PROSPECT[4].

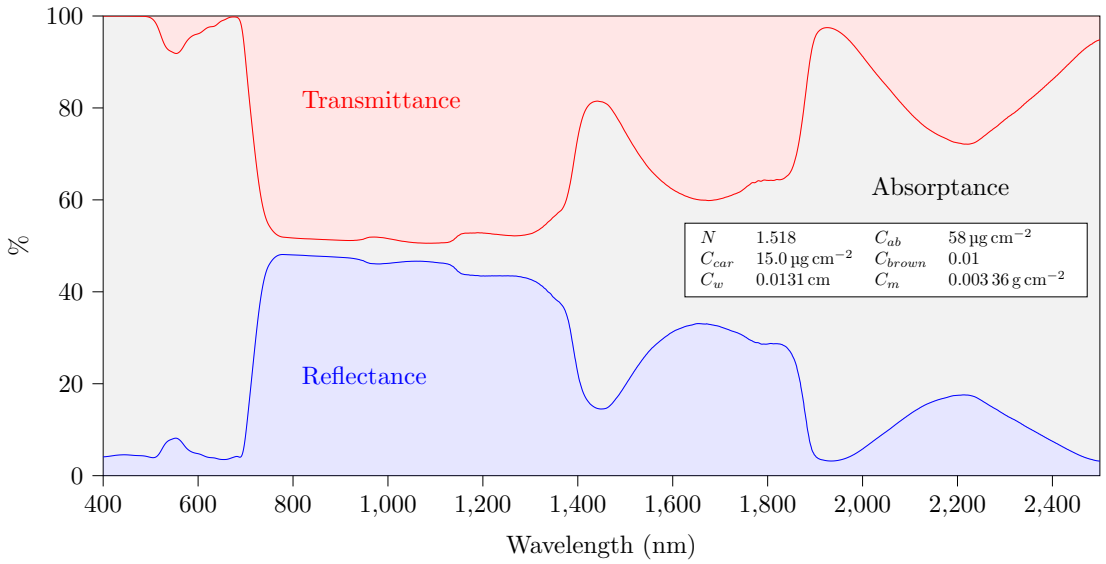


Figure 4.3: Simulated reflectance, transmittance and absorbance of maize leaf with PROSPECT-5.

4.2 The SAIL canopy model

The Scattering by Arbitrary Inclined Leaves (SAIL) model [Verhoef, 1984] is an RTM that simulates vegetation canopy optical properties. It simulates the canopy bi-directional reflectance factor from leaf reflectance and transmittance spectra, and canopy structure parameters. SAIL models the canopy with the following simplifying assumptions:

- the canopy is a single, horizontal, homogeneous and infinitely extended layer in the horizontal plane,
- the leaves are planar and bi-Lambertian[5] surfaces, with infinitesimal size,
- the leaves are the only canopy elements.

SAIL is a *turbid medium* model, the light scattering occurs because of small elements (the leaves) which are assumed to be randomly distributed. SAIL is a *four-stream* model (see Figure 4.4), in that it uses four radiative fluxes:

- the directional incident solar flux e_s,

[4] http://teledetection.ipgp.jussieu.fr/prosail
[5] Incident light is scattered isotropically.

- the directional canopy radiance flux toward the observer e_o,
- the diffuse (semi-hemispherical) downward flux e_-,
- the diffuse (semi-hemispherical) upward flux e_+.

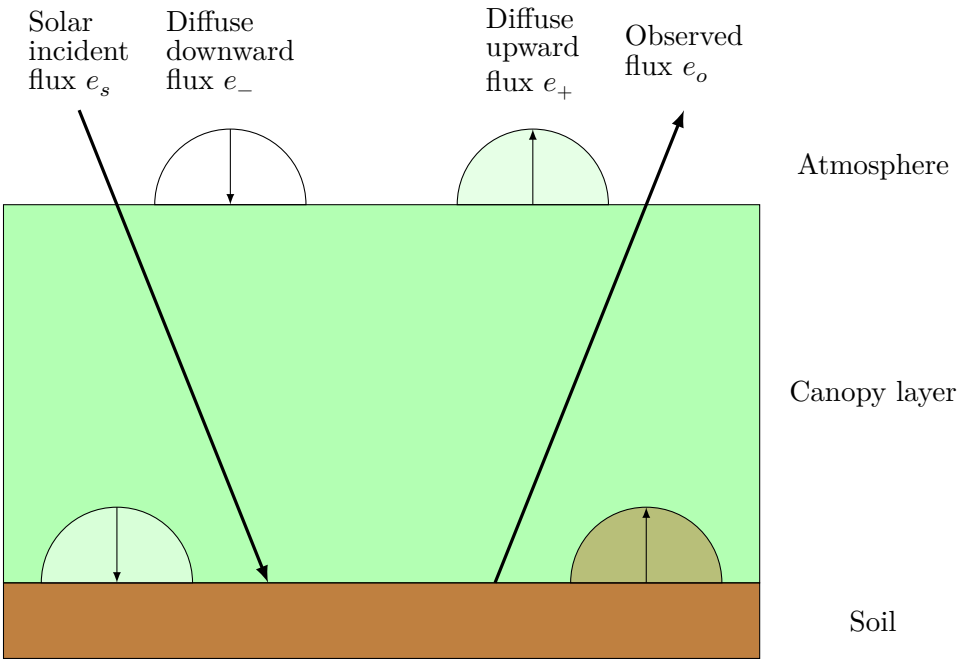


Figure 4.4: The four-stream radiative transfer in SAIL

The four-stream radiative transfer theory describes the interactions between these four fluxes with a set of four differential equations[6]. Also, two additional equations can be provided to take into account the interactions of the fluxes with the soil, to further constrain the radiative transfer model. These equations are function of the leaf area index (LAI) and the *relative optical height*, which designates the relative height within the canopy and ranges from -1 at bottom-of-canopy (at soil level), to 0 at top-of-canopy (TOC) level. To obtain the canopy optical properties, the radiative transfer equations are analytically solved at the boundaries of the canopy relative optical height. The general form of the solution for TOC radiance, is a relation between e_s, e_o, e_-:

$$e_o(0) = \rho_{s,o} e_s(0) + \rho_{-,o} e_-(0) + \sum_i \gamma_i h_i + \sum_j \gamma_j \Delta h_j, \tag{4.5}$$

with $\rho_{s,o}$ the bi-directional reflectance factor at the TOC, $\rho_{-,o}$ the hemispherical-directional reflectance factor. The h_i are hemispherical thermal fluxes associated with the blackbody radiance of different elements in the canopy (the leaves in the Sun or in the shade, the soil, the atmosphere), Δh_j are the differences between a pair of such fluxes, and γ_i, γ_j are associated emissivity coefficients. The Equation 4.5 is derived from equation 11 in Verhoef et al. [2007], which introduced the "unified expression for TOC flux-equivalent radiance", for the improved 4SAIL version of the SAIL model. It can be noted that the SAIL model is constant with respect to (w.r.t.) the wavelength λ (except for thermal fluxes modeling, see subsection 4.2.4). This means that calculations need not be performed on entire spectra.

Depending on the application and available information about the canopy, the different terms in Equation 4.5 are expressed as a function of different input parameters. They can be simplified, or even removed, to account for, or discard certain phenomena. In particular, the

[6]The radiative transfer equations become ordinary when discarding the assumption of infinitesimal leaf size to a finite size, which is performed with SAILH and 4SAIL.

PROSAIL-VAE application discussed in this work (see Chapter 8) only uses the bi-directional reflectance factor from the sun incident light to the observer $\rho_{s,o}$, while other are neglected. In the following parts, the different variables and processes involved in the radiative transfer are discussed.

4.2.1 Leaf optical properties

The SAIL model considers the canopy as an homogeneous layer of identical leaves. Specifically, the leaf reflectance ρ_l and transmittance τ_l spectra are two required model variables. This property enables coupling the SAIL model with leaf optical models, such as PROSPECT introduced in section 4.1.

4.2.2 Soil optical properties

The SAIL model takes into account the influence of the soil over the radiance fluxes inside the canopy. However, SAIL does not compute soil reflectance spectra, but rather treats them as an input. There are several possibilities to compute the soil reflectance spectra prior to inputting it to SAIL. The soil can be considered a Lambertian surface, in which case its properties are simply defined with a hemispherical reflectance spectra ρ_S. The 4SAIL version introduced the possibility of using a non-Lambertian soil, whose properties are expressed with four reflectance spectra (bidirectional, bihemispherical, directional-hemispherical and hemispherical-directional).

Modeling the soil reflectance spectra is difficult, especially in a non-Lambertian scenario, because it requires knowledge about the soil mineral composition, surface roughness, humidity, etc. The Lambertian behavior of the soil depends on the wavelength, and also on environmental variables. In most applications, the soil is considered Lambertian as a simplifying assumption.

A possibility to obtain a soil reflectance spectra, is to use a synthetic soil spectra derived from two reference spectra corresponding to two edge situations: a dry soil ρ_{dry}, with high reflectance and a wet soil ρ_{wet} with a lower reflectance:

$$\rho_S = s_b \left[s_w \rho_{dry} + (1 - s_w) \rho_{wet} \right]. \tag{4.6}$$

The soil reflectance is thus a sum of the two reference soil spectra, weighted by the wet soil coefficient s_w and scaled by a brightness coefficient s_b (see Figure 4.5). This method enables

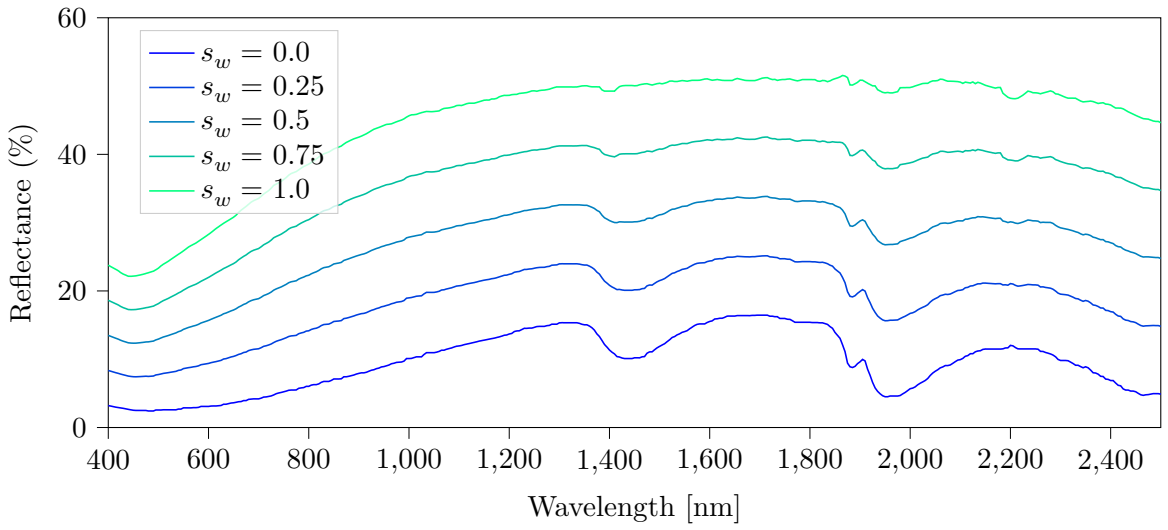


Figure 4.5: Soil reflectance as a weighted sum of dry and wet soil reference spectra.

to model a relatively wide range of soils, with just two input variables. This approach is used in the PROSAIL-VAE application presented in Chapter 8.

Another approach is to make soil measurements and build a data-set of reference soil spectra. The soil spectrum that matches best with a given situation is then provided to the SAIL model. This spectrum is also scaled by a brightness coefficient s_b. This second method was selected for the PROSAIL model used to generate the training data-set of the neural network at the core of Sentinel Application Platform (SNAP)'s Biophysical Processor (BP) in Weiss and Baret [2016], by using the 7 reference spectra provided in Weidong et al. [2002].

4.2.3 Canopy geometrical structure

4.2.3.1 The LAI

The leaf area index (LAI) is defined as half the surface area of the leaves (see subsection 2.3.1) per unit horizontal ground surface area. It is a key variable in the SAIL radiative transfer equations.

4.2.3.2 The leaf inclination distribution function

The SAIL model [Verhoef, 1984] is based on the Suits model [Suits, 1971], which is also a four-stream RTM, with similar idealized canopy assumptions. In the Suits model, to calculate the scattering and extinction coefficients involved in the radiative transfer equation, the leaves were assumed to be only either horizontal or vertical. This assumption in leaf orientation causes singularities in the modeled reflectance w.r.t. the viewing angle zenith. In SAIL, these singularities are mitigated by assuming that the leaf orientation, described with a leaf zenith angle θ_l is random, and bound by a leaf inclination distribution function (LIDF) $g(\theta_l)$.

The choice of a LIDF is arbitrary. The LIDF are commonly classified into four categories:

1. planophile (mostly horizontal leaves),
2. erectophile (mostly vertical leaves),
3. plagiophile (mostly oblique leaves),
4. extremophile (few oblique leaves).

The extremophile LIDF describes leaves that are mostly horizontal and vertical, i.e. it is a bimodal distribution. In the original SAIL paper [Verhoef, 1984], the spherical LIDF distribution $g(\theta_l) = \sin\theta_l$ is used. Modelling the canopy with the spherical LIDF assumes that the angular distribution of leaf area is similar to the distribution of area on the surface of a sphere. This corresponds to a erectophile LIDF.

Other LIDF options were introduced to better model each type of vegetation. For instance, in Verhoef [1981] a LIDF was proposed as a deviation of the uniform distribution. This LIDF is constructed graphically, as a $\frac{\pi}{4}$ rotation of the function $a\sin x + \frac{b}{2}\sin 2x$, with the two parameters a and b such that (s.t.) $|a| + |b| < 1$. The parameter a controls the average leaf inclination, whereas b controls the bimodality. Thus, this function can be tuned to adapt to a variety of cases. There is no analytical formula for this LIDF, so it is approximated with iterative algorithms. This will be referred here as Verhoef's distribution.

Other common LIDF choices are the De Wit distributions [de Wit, 1965] which are four parameter-less trigonometrical functions adapted to each four LIDF types, and Beta distribution fitting, that require two parameters, and fits quite well many cases, except bimodal extremophiles.

Another popular LIDF, are Campbell's ellipsoidal functions [Campbell, 1986, 1990] controlled by the mean leaf angle $\overline{\alpha}$. This distribution is designed to be a simple model of leaf inclination, just as the spherical LIDF, but with more flexibility. Instead of assuming that the leaf angular distributions is similar to the surface repartition on a sphere, Campbell

LIDF assumes that this surface is similar to that of an ellipsoid. The spherical distribution is included as a particular case of the ellipsoidal distribution, for $\overline{\alpha} \approx 58.44$. A few examples of this distribution are plotted in Figure 4.6. This model is well suited to inversion, as it only requires a single parameter. This is the LIDF used in this work (see subsection 4.5.2).

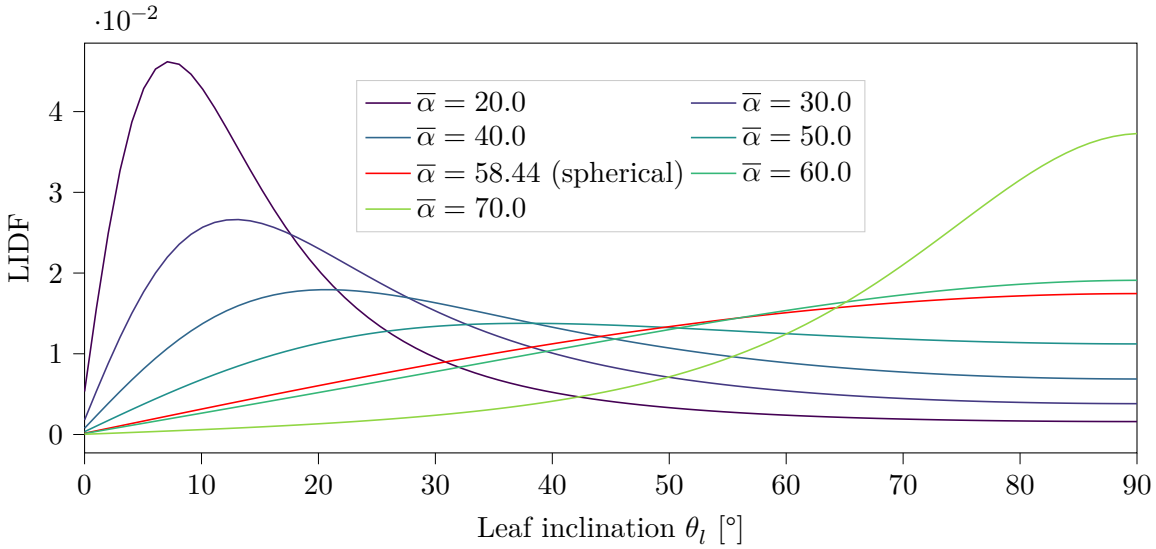


Figure 4.6: Campbell's ellipsoidal LIDF for various mean leaf angles.

4.2.3.3 The hot-spot effect

When observing a 3-D scene illuminated by a directional source, the viewing angle has an influence over the magnitude of the sensed radiance. In particular, there is a particular geometric configuration in which the (BRDF) reaches a maximum: when the viewing angle and the source angle are aligned. This so-called *hot-spot* effect, occurs because of the light backscattering on the scene. It depends on the structure of the scene, and in particular, on the possibility of light being reflected in a particular direction rather others. Notably, when the observer and the source are aligned, all the scene appears to be illuminated, and no shadow is visible.

When observing a canopy from the top, there is also a hot-spot phenomenon, because of the backscattering over the leaves. However, the hot-spot as a single backscattering event isn't well suited in canopies. The theory of Kuusk [1985] describes the hot-spot effect in vegetation canopies, and was used to improve the SAIL model as a updated version, SAILH, as part of Verhoef's Ph.D. work [Verhoef, 1998], and present in the subsequent 4SAIL. In particular, this hot-spot model requires to discard the assumption of infinitesimal leaf size, and to consider it finite. This theory is based on the description of the (BDGP) p_{so}, which is the joint probability of two events: an incident light ray penetrating the canopy (e_s), and being reflected and transmitted inside in the direction of the observer (e_o). These events are described with their own marginal probabilities p_s and p_o, called the *directional gap probabilities*. The BRDF of the canopy is a function of the BDGP. When the events e_s and e_o are independent, the BDGP is simply the product of the marginal gap probabilities: $p_{so} = p_s p_o$. This matches the assumption of a turbid medium. The hot-spot effect occurs because of leaves have a finite size, which introduces a correlation between events e_s and e_o, that can be taken into account in the BRDF as a correction factor c_{hs}: $p_{so} = p_s p_o c_{hs}$. This correction factor depends on a *horizontal correlation length* l, which is a characteristic quantity of the canopy and depends on its architecture.

Within SAILH, the hot-spot effect is taken into account by computing the BDGP, to correct the BRDF. As the BDGP is an analytically un-derivable integral, it is computed

through numerical integration with a fixed number of steps. The horizontal correlation length l is not directly used, but the normalized, dimensionless quantity $h = \frac{l}{h_c}$, with h_c the canopy height, is used instead. h is the *hot-spot size parameter*, or *hot-spot parameter*, and is the only parameter necessary to tune the hot-spot effect within SAILH. In Figure 4.7 is displayed the angular variation of the BRDF simulated with 4SAIL, from given leaf reflectance and transmittance and soil reflectance. Increasing the hot-spot size parameter spreads the hot-spot effect beyond the backscattering direction, i.e. the BRDF increases farther away from the backscattering direction.

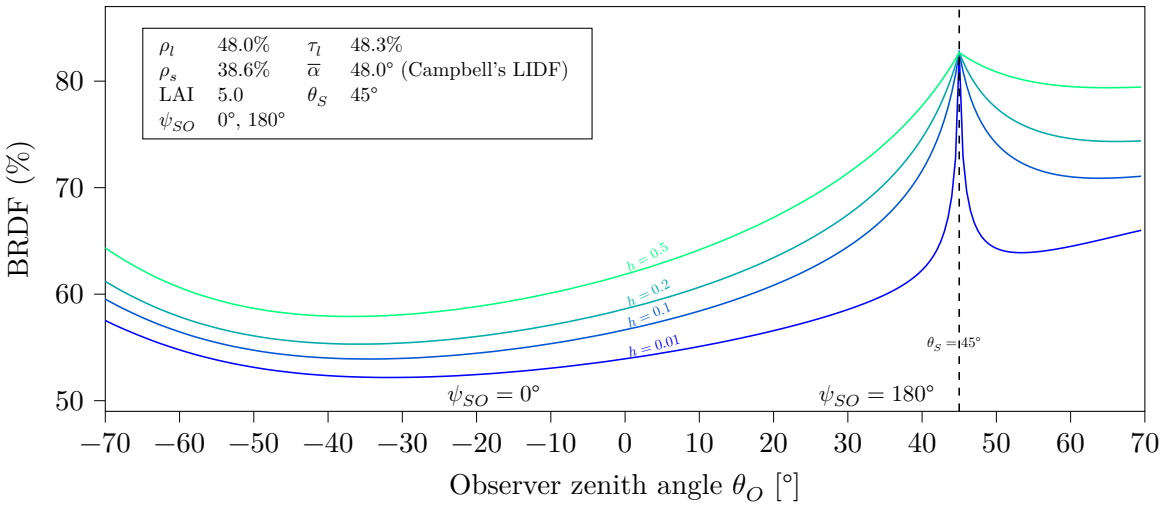


Figure 4.7: Effect of the hot-spot size parameter h on the BRDF of a simulated maize reflectance and transmittance at $\lambda = 800$ nm

The hot-spot parameter characterizes the canopy structure. In Verhoef [1998], h is related to two growth strategies in vegetation:

1. plants that grow taller rather than wider (wheat, maize),
2. plants that grow wider rather than taller (sugar beet).

In both cases, the leaf surface (i.e. the LAI) is increased. In the first case, the plants grow taller as they grow new leaves, whereas in the second case, plants let their leaves grow bigger without increasing the number of leaves. For the first strategy, by assuming that the leaf size is constant, then $h \propto \frac{1}{\text{LAI}}$. Such a relationship between h and the LAI is experimentally found in subsubsection 8.2.3.4. For the second strategy, the hot-spot parameter is constant w.r.t. the LAI.

4.2.4 The thermal fluxes

An addition of 4SAIL compared to previous versions, is that it takes into account the blackbody thermal fluxes into account. The black-body emissions that are considered are those of the shaded and sun-lit soil, shaded and sun-lit leaves. Each of these additional sources are associated with a black-body temperature. In the present work's use of 4SAIL, these thermal fluxes are neglected, and having to provide these temperatures is avoided.

4.3 PROSAIL

As discussed in section 4.2, the SAIL model requires a leaf reflectance and transmittance as inputs. These optical leaf properties can be simulated by a leaf RTM, such as PROSPECT, introduced in section 4.1. The coupling between a PROSPECT leaf RTM and a SAIL canopy RTM is known as the PROSAIL model. This composite RTM simulates canopy BRDF from a set of bio-physical variables (BV) (see Figure 4.8).

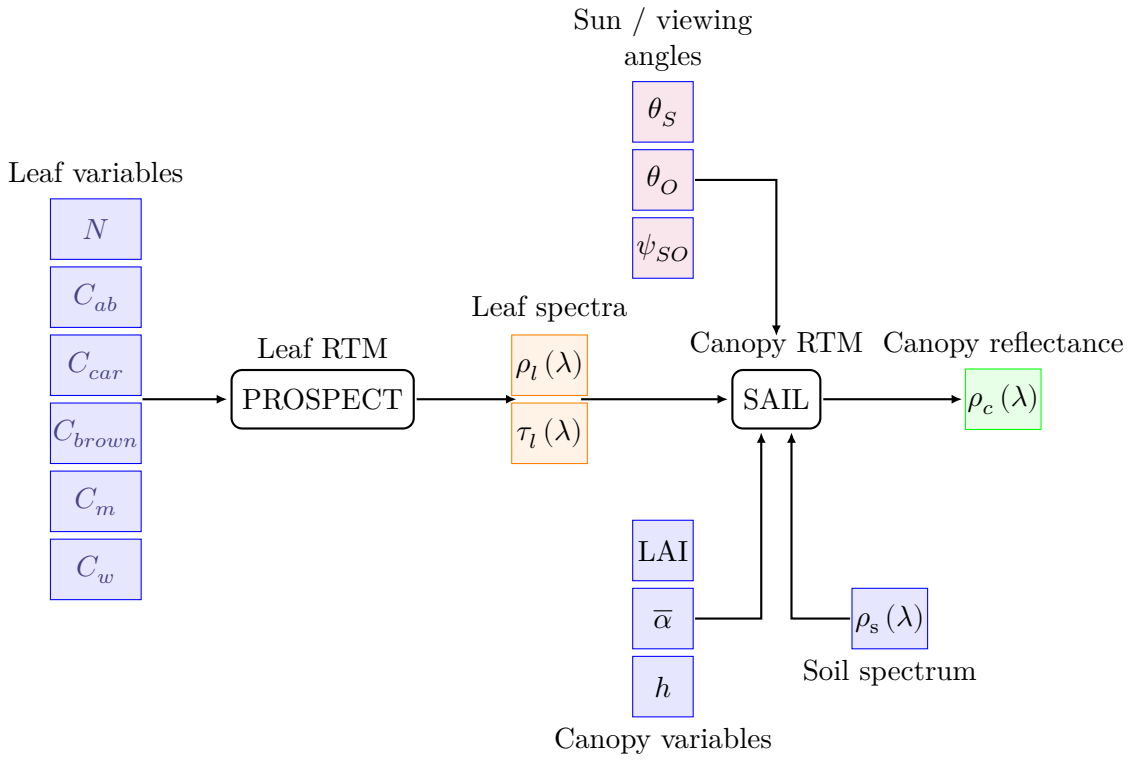


Figure 4.8: The PROSAIL model, fusion between PROSPECT and SAIL.

An example of a PROSAIL simulation is provided in Figure 4.9, which compares a leaf BRDF generated by PROSPECT with the canopy spectra obtained by PROSAIL, with a common set of BV.

The specific PROSAIL model depends on the choice of version for its components, PROSPECT and SAIL. In the present work, we consider for PROSAIL, the combination of PROSPECT-5 and 4SAIL (without incorporating the thermal fluxes into the radiative transfer). Consequently, the variables taken into account to simulate canopy BRDF are detailed in Table 4.2.

[7]Both units are commonly found in the literature, as they are "equivalent", because the density of water is $1.0\,\mathrm{g\,cm^{-3}}$. Thus for the leaf water x g is "equivalent" to x $\mathrm{cm^3}$, so x $\mathrm{g\,cm^{-2}}$ of leaf water content is "equivalent" to x $\mathrm{cm^3\,cm^{-2}} = x$ cm.

[8]Arbitrary unit per surface unit

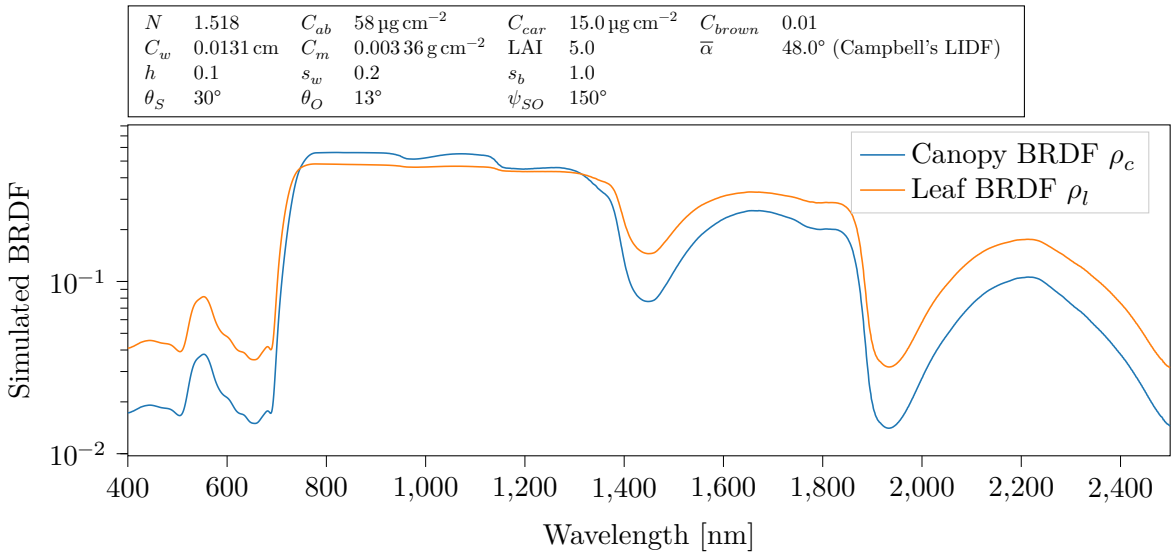


Figure 4.9: Simulated BRDF of a leaf with PROSPECT-5, and a corresponding canopy BRDF with PROSAIL (PROSPECT-5 + 4SAIL).

Table 4.2: PROSAIL input parameters

Model	Input	Description	Unit
PROSPECT-5	N	Leaf structure parameter	-
	C_{ab}	Chlorophyll $a+b$ content	µg cm^{-2}
	C_w	Water equivalent thickness	g cm^{-2} or[7] cm
	C_{car}	Carotenoid concentration	µg cm^{-2}
	C_m	Dry matter content	g cm^{-2}
	C_{brown}	Brown pigments content	a.u.p.s.u[8]
4SAIL	LAI	Leaf Area Index	-
	$\overline{\alpha}$	Mean leaf angle	deg
	h	Hotspot parameter	-
	s_w	Wet soil factor	-
	s_b	Soil brightness factor	-
Geometry	θ_S	Solar zenith angle	deg
	θ_O	Observer zenith angle	deg
	ψ_{SO}	Relative azimuth angle	deg

4.4 Sensor measurements simulation

Using a RTM such as PROSAIL enables to simulate the reflectance spectrum of a canopy, from bio-physical variables. However, the physical quantities that are measured by remote instruments aren't reflectance spectra, but energy integrated over *spectral bands*, proportional to incident equivalent spectral radiance, which are related to equivalent reflectances (see subsubsection 2.1.3.3). To simulate a remote sensor measurement of a canopy, it is necessary to derive the *equivalent reflectance* spectra, from the canopy reflectance spectra ρ_c simulated beforehand.

The bottom-of-atmosphere (BOA) equivalent reflectance ρ_i observed under BOA solar irradiance e_s for a spectral band i, with range $\left[\lambda_{\mathrm{l},i}, \lambda_{\mathrm{u},i}\right]$, and a spectral sensitivity s_i is [Tupin et al., 2014]:

$$\rho_i(\lambda) = \frac{\int_{\lambda_{\mathrm{l},i}}^{\lambda_{\mathrm{u},i}} s_i(\lambda)\, e_s(\lambda)\, \rho_c(\lambda)\, \mathrm{d}\lambda}{\int_{\lambda_{\mathrm{l}}}^{\lambda_{\mathrm{u}}} s(\lambda)\, e_s(\lambda)\, \mathrm{d}\lambda}. \tag{4.7}$$

In the present work, the spectral bands considered are the bands of S2's multi spectral instrument (MSI) except those dedicated to atmospheric correction (i.e. the 60 m resolution bands): `B2`, `B3`, `B4`, `B5`, `B6`, `B7`, `B8`, `B8A`, `B11`, `B12` (see Figure 2.1).

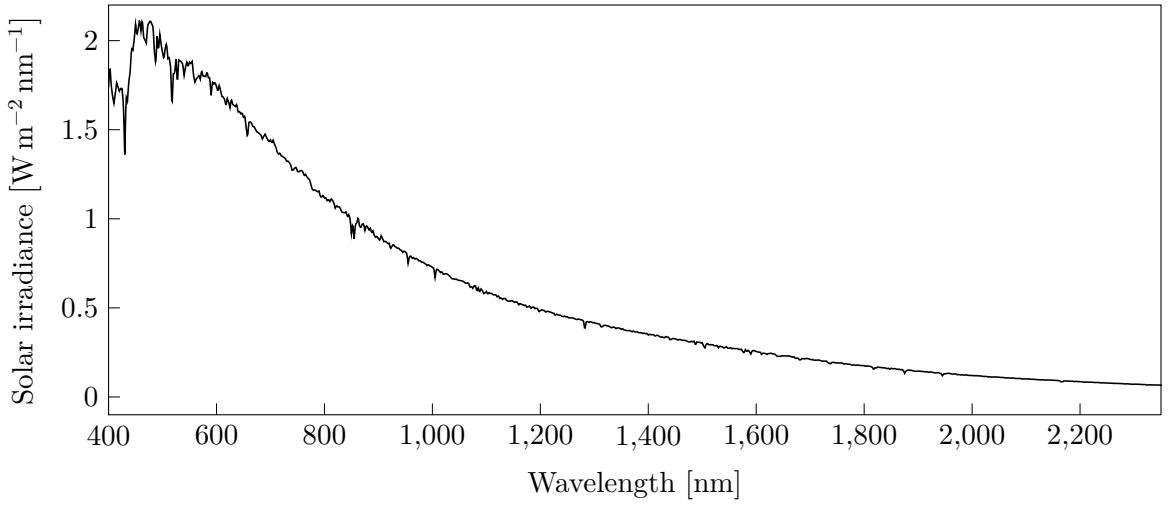


Figure 4.10: Top-of-atmosphere solar irradiance spectrum

It is important to note that the canopy reflectance ρ_c used in Equation 4.7 is a BOA reflectance, such as what is simulated with PROSAIL. Unfortunately, obtaining the required BOA solar irradiance spectrum $e_s(\lambda)$ is complicated. Due to the light scattering in the atmosphere, the BOA solar irradiance spectrum is different to the top-of-atmosphere (TOA) spectrum, which is more easily accessed. To derive the BOA spectrum from the TOA spectrum, it is necessary to apply atmospheric corrections, which requires atmospheric modeling and the computation of corrective factors (see subsubsection 2.1.3.4). Fortunately, the light scattering by the atmosphere is a negative power function of λ, meaning that except for shorter wavelengths, this effect is of low magnitude. For S2's MSI bands, `B2` is the most affected and the difference between $\rho_{\mathtt{B2},t}$ from $\rho_{\mathtt{B2},b}$ is around 1%. As such, in this Ph.D., the BOA equivalent reflectance spectra of simulated S2 bands are approximated by using the TOA solar irradiance spectrum (see Figure 4.10). Besides, the PROSAIL-simulated BOA equivalent reflectances will be compared to Level-2A S2 reflectance bands. These are BOA reflectances obtained from TOA acquisitions by applying atmospheric corrections. As such, even for the reference S2 measurements, there is an unavoidable uncertainty due to correction errors.

4.5 Refactoring PROSAIL for Deep Learning end-to-end optimization

In this work, the above-defined PROSAIL model is to be integrated within a deep learning (DL) framework (see Chapter 8). This brings about two implementation requirements:

1. gradient propagation within PROSAIL must be enabled,
2. PROSAIL must be able to deal with array-based, batched data.

PROSPECT and SAIL models were originally implemented in Fortran, and later ported into Matlab and Python. These codes don't have these two requirements, as they were intended for off-line forward modeling. They offer no gradient-related capabilities. While they do perform canopy reflectance simulation of whole discretized, array-based spectra, they only simulate one spectrum at a time and cannot perform batched computations.

As a consequence, PROSAIL was adapted in this work, and implemented on Python's Pytorch DL library. Pytorch implementations of batched variables as tensors is similar to Numpy's implementations as arrays. Therefore, PROSAIL was adapted from the Numpy code of Domenzain et al. [2019].

Pytorch, like Numpy, meets the second requirement of managing batched data, so long that variables are encoded as tensors. This also enables to perform efficient, parallel computing on graphical processing units (GPUs). Also, the library's automatic differentiation engine `torch.autograd` enables gradient computation, forward and backward propagation, on the condition that all operations are differentiable and written with Pytorch functions.

However, adapting PROSAIL to the Pytorch library is not straightforward, and is not just about "tensorizing" the model (e.g. encoding all variables as multi-dimensional tensors). Some operations involved in PROSAIL are not available in the library and must be adapted into differentiable and batched implementations. The following subsection 4.5.1 and subsection 4.5.2 detail key implementation choices to ensure these requirements. Finally, as the batched implementation of PROSAIL is very memory-intensive, subsection 4.5.4 details how this issue is mitigated, by down-sampling the model.

4.5.1 The exponential integral function

In PROSPECT, the leaf transmission coefficient θ from the leaf absorption coefficient k (see Equation 4.3) involves the exponential integral Ei

$$\theta = (1-k)\,\mathrm{e}^{-k} - k^2 \mathrm{Ei}\,(-k)\,, \tag{4.8}$$

defined as :

$$\forall x \in \mathbb{R}^*, \quad \mathrm{Ei}\,(x) = \int_{-\infty}^{x} \frac{\mathrm{e}^t}{t}\,\mathrm{d}t \tag{4.9}$$

The exponential integral is not an elementary function, and must therefore be approximated. At the time of this work, this function isn't available in Pytorch, and therefore must be implemented manually. Care must be taken that this implementation enables parallel computing with this function, and that automatic differentiation can be applied to it.

This function is defined on the complex plane, but as PROSAIL only deals with real-valued numbers, here the exponential integral will only be approximated on the real set. Besides, as the leaf absorption coefficient is positive, the exponential integral is only used over negative values. The approximation of this function must be computationally efficient, and avoid recursion.

The Ei function is linked to another integral Ein, the *complementary* exponential integral defined in Equation 4.10.

$$\forall x \in \mathbb{R}, \quad \mathrm{Ein}\,(x) = \int_{0}^{x} \frac{(1-\mathrm{e}^{-t})}{t}\,\mathrm{d}t \tag{4.10}$$

Algorithm 1 Iterative computation of the E_1 function with approximation of a continued fraction expansion.

function $\mathrm{E}_1(x, n)$
  $t_0 \leftarrow 0$
  for $i := 1$ **to** n **do**
    $k_i \leftarrow n - i + 1$
    $t_i \leftarrow \frac{k_i}{1+\frac{k_i}{x+t_{i-1}}}$
  end for
  $y \leftarrow \frac{\mathrm{e}^{-x}}{x+t_n}$
  return y
end function

Equation 4.11 links Ei with Ein over the whole of the definition interval:

$$\forall x \in \mathbb{R}^*, \quad \mathrm{Ei}\,(x) = -\,\mathrm{Ein}\,(-x) + \ln|x| + \gamma, \tag{4.11}$$

with $\gamma = \lim_{n\to+\infty}\left(-\ln(n) + \sum_{k=1}^{n}\frac{1}{k}\right) \approx 0.57722$ the Euler-Mascheroni constant. The Ein function can be computed by approximating a power series:

$$\mathrm{Ein}\,(x) = \sum_{k=1}^{+\infty}\frac{(-1)^{k+1}}{kk!}x^k. \tag{4.12}$$

Using this function enables to simply calculate the exponential integral in any evaluation point. However, it must be noted that for large values of x, this power series converges slowly, and approximating it requires many terms. For some intervals, there are different approximations possible with increased precision.

The function Ei is also linked to the function E_1, defined in Equation 4.13.

$$\forall x \in \mathbb{R}_+^*, \quad \mathrm{E}_1\,(x) = \int_1^{+\infty}\frac{\mathrm{e}^{-tx}}{t}\,\mathrm{d}t \tag{4.13}$$

Equation 4.14 shows that the exponential integral can be defined over the negative numbers by just approximating the function E_1.

$$\forall x \in \mathbb{R}_+^*, \quad \mathrm{Ei}\,(-x) = -\,\mathrm{E}_1\,(x) \tag{4.14}$$

The E_1 function has infinite continued fraction representation [Gautschi and Cahill, 1964] shown in Equation 4.15 [9]. A good approximation can be computed from a relatively low number of successive fractions (e.g. $n = 40$), with an iterative algorithm (see Figure 1).

$$\forall x \in \mathbb{R}_+^*, \quad \mathrm{E}_1\,(x) = \cfrac{\mathrm{e}^{-x}}{x + \cfrac{1}{1+\cfrac{1}{x+\cfrac{2}{1+\cfrac{2}{x+\ddots}}}}} = \lim_{n\to+\infty}\mathrm{e}^{-x}\left[x + \underset{k=1}{\overset{n}{\mathrm{K}}}\left(\frac{\lfloor\frac{k+1}{2}\rfloor}{x^{(k+1) \bmod 2}}\right)\right]^{-1} \tag{4.15}$$

By using a fixed number of fractions (e.g. 40), a reasonable approximation can be computed iteratively instead of recursively.

[9] K is Gauss's continued fraction operator, for which $\mathrm{K}_{k=1}^{n}\frac{b_k}{a_k} = \cfrac{b_1}{a_1+\cfrac{b_2}{a_2+\cfrac{b_3}{\ddots+\cfrac{b_n}{a_n}}}}$, with a_k the *partial denominators* and b_k the *partial numerators*.

It can be noted that the proposed implementation of the exponential integral function may be further improved to lessen the computational burden. For instance, instead of computing the exponential integral at each evaluation, an abacus may be created instead. Such case may be interesting, as it reduces the evaluation of the function to a simple interpolation between pre-computed values. In the case of PROSPECT, there isn't really even a need for pre-computing this function over its entire definition interval, because it is applied only over a small interval of negative numbers.

Even-though approximating the exponential integral function requires a bit of calculation, its derivative has an analytical expression. Because Ei is defined as an integral, computing its derivative (see Equation 4.16) is straight-forward.

$$\forall x \in \mathbb{R}^*, \quad \frac{\mathrm{dEi}}{\mathrm{d}x}(x) = \frac{\mathrm{e}^x}{x} \tag{4.16}$$

4.5.2 The leaf inclination distribution function

As detailed in subsubsection 4.2.3.2, the LIDF describes the density of the random leaf surface orientation. Computational constraints guide the choice of a LIDF for 4SAIL. Verhoef's LIDF is very flexible, however this function does not have analytical expressions and must be computed iteratively. As the number of iterations is not constant, it is difficult to parallelize this function. Furthermore, these functions require two parameters, which increases the dimension of inversion problems. Similarly Beta distributions also require two parameters. Alternatively, Campbell ellipsoidal distributions allow a certain flexibility while just requiring a single input parameter. Furthermore, although analytical computation isn't possible, numerical approximation is straightforward and is easily parallelizable. This is why Campbell's ellipsoidal LIDF is chosen for this work.

4.5.3 Gradient-based sensitivity analysis

A differentiable implementation of PROSAIL enables to easily compute the gradients of the outputs of the model w.r.t. to its inputs by using automatic differentiation (see subsubsection 3.3.2.1). By definition, the gradient describes the rate of change of the output for a given change in the input. As such, computing these gradients enables to analyze the sensitivity of the output of PROSAIL (i.e. the canopy reflectance spectra or the S2 bands) to the PROSAIL input variables. *Sensitivity analysis* is the study of how the values of a variable influences the value of a dependent variables. Usually, this encompasses a study of how uncertainties in the independent variables are propagated to the uncertainty of a dependent variable.

The distribution of gradients of S2 bands w.r.t. the PROSAIL input variables can be estimated with the differentiable implementation of the PROSAIL model described in this section. This is performed by sampling $N = 10\,000$ sets of PROSAIL variables (i.e. 14 variables, including S2 and Sun angles), simulating the corresponding S2 reflectance bands and by using automatic differentiation to retrieve the gradients. Contrary to the simulation procedure described in Chapter 5, the variables are sampled with a uniform distribution over their definition range, and no correlation is introduced in sampled variables. This is because the objective is to estimate the gradients for a maximum of different configurations, rather than simulate a data-set that mimics the distribution of variables found in nature. The estimated distribution of gradients is shown in Figure E.4.

A limitation of the above-described sensitivity analysis, is that it is affected by the *curse of dimensionality*. The curse of dimensionality broadly refers to problems that arise when dealing with high dimensional data. For sampling variables in a high-dimensional space, which is performed here for gradient distribution estimation, the problems are two-fold:

- The number of samples required to map the target space with an arbitrary precision increases exponentially with the number of dimensions. For instance, the number of

samples required for sampling all variable combinations on a regular grid of the d-dimensional variable space is n^d, with n the number of sampled values per dimension. In the current PROSAIL setting, for a budget of $N = 10\,000$ samples, with $d = 14$ dimensions, a regular grid would test $n = N^{1/d} \approx 1.9$ values per PROSAIL variable.

- Samples from a given distribution of a high dimensional space are more likely to be located near the boundary. This is a geometrical effect, due to most of the "volume" of a given subspace being distributed closer to the edge of the subspace rather than the origin (e.g. most the volume of a 3D sphere is contained closer to the surface of the sphere rather than the center.). As a consequence, the interior of the sampled subspace is depleted compared to the space near the boundary. This is also called the *edge effect*.

There are approaches that attempt to mitigate the curse of dimensionality for performing sensitivity analysis [Moreau et al., 2013; Sheikholeslami et al., 2019], however they are out of the scope of this work. For the purposes of this Ph.D., a large precision in the estimated distribution of the gradients of the PROSAIL model is not necessary.

4.5.4 Under-sampling PROSAIL

PROSAIL originally simulates canopy spectra over the range 400 – 2500 nm, with a resolution of 1 nm[10]. Therefore, the model output is a vector of size 2101. However, while using PROSAIL in a batched manner, the spectra of multiple samples are simulated simultaneously. Consequently, the total output size increases proportionally to the number of samples. For a single-precision encoding (4 bytes per value), the simulated spectra corresponding to a 32×32 image patch (i.e. 1024 pixels) has a size of over 8 MB.

When using PROSAIL inside a machine learning model with back-propagating gradient, the output spectra and the intermediary variables are saved as tensors. Because each tensor is kept in memory for back-propagation, the memory size increases very quickly with the number of performed simulations. Experiments have shown that for a GPU with a 32 GB, no more than 10 000 samples can be simulated at a time, before the memory is saturated.

To decrease the computational burden, and eventually increase the number of samples within each batch, the different spectra used inside PROSAIL are down-sampled, so that all tensors used within PROSAIL have a reduced size. Besides, in the applications presented here, the output canopy spectra is not directly used. The sensor model of S2 transforms the canopy reflectance spectra into 10 reflectance bands values, essentially performing down-sampling with a factor of more than 200. To simulate those 10 bands, it may be unnecessary to have such a densely sampled canopy spectra. Nonetheless, as will be shown below, each simulated S2 band is affected differently by a down-sampling of the input reflectance spectra, because of their different bandwidth.

To perform down-sampling on PROSAIL, it is necessary to down-sample all the reference spectra for leaf pigments, soil, as well as the sensor *sensitivity spectra* (see Figure 4.11, Figure 4.14 and Figure 2.1). These spectra are constant within PROSAIL, so they only need to be initialized. The down-sampling of spectra only needs to be performed once, and doesn't need to be differentiable. A *down-sampled PROSAIL* will refer in the following to a PROSAIL RTM, for which all leaf content spectra, soil spectra are down-sampled with a given factor r. This down-sampled PROSAIL is associated with the down-sampled response spectra of S2 bands. In practice, as the initial resolution of the down-sampled spectra is 1 nm, the down-sampling factor will correspond to the new spectral resolution in nm.

There are multiple down-sampling techniques for one-dimensional signals such as PROSAIL spectra, such as decimation, anti-aliasing filtering or low-pass filtering. In this work, it was chosen to use a strided moving-average filter, with the down-sampling factor r being the window size, and the stride being equal to the window size (see Figure 4.11). This down-sampling can also be understood alternatively as:

[10] In the versions 4, 5, D and PRO of PROSPECT

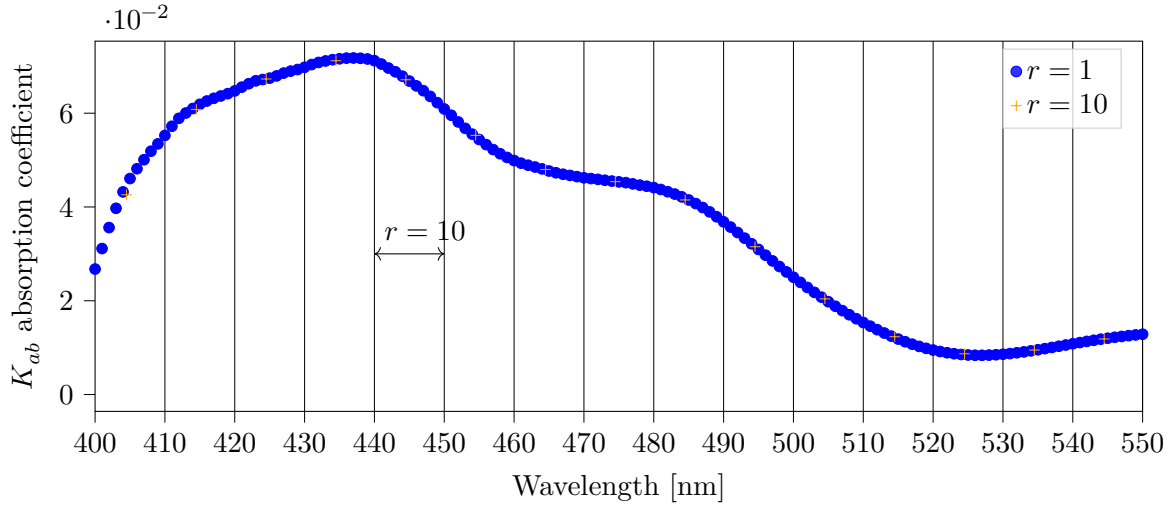


Figure 4.11: Down-sampling of chlorophyll absorption spectra K_{ab} with strided moving average method.

- the application of a simple moving-average filter which is a low-pass finite impulse response (FIR) filter, followed by a decimation step,
- the 1D convolution over the spectra with a kernel of size r, with weights $\frac{1}{r}$, with stride r,
- the 1D average pooling of the spectra, with a pool size of r.

This down-sampling method has the advantage of being simple to implement. Also, unlike some other techniques, it can be applied directly on the spectral dimension of the signal, without relying on Fourier transforms.

As shown in a sample simulation in Figure 4.12, using a down-sampled PROSAIL to simulate canopy spectra and S2 bands introduces some errors. Overall, the larger the down-sampling factor is, the greater is the error w.r.t. a non down-sampled PROSAIL. For some bands, with a large down-sampling factor, the error surpasses the uncertainty caused by MACCS-ATCOR joint algorithm (MAJA) atmospheric correction (see subsubsection 2.1.3.4).

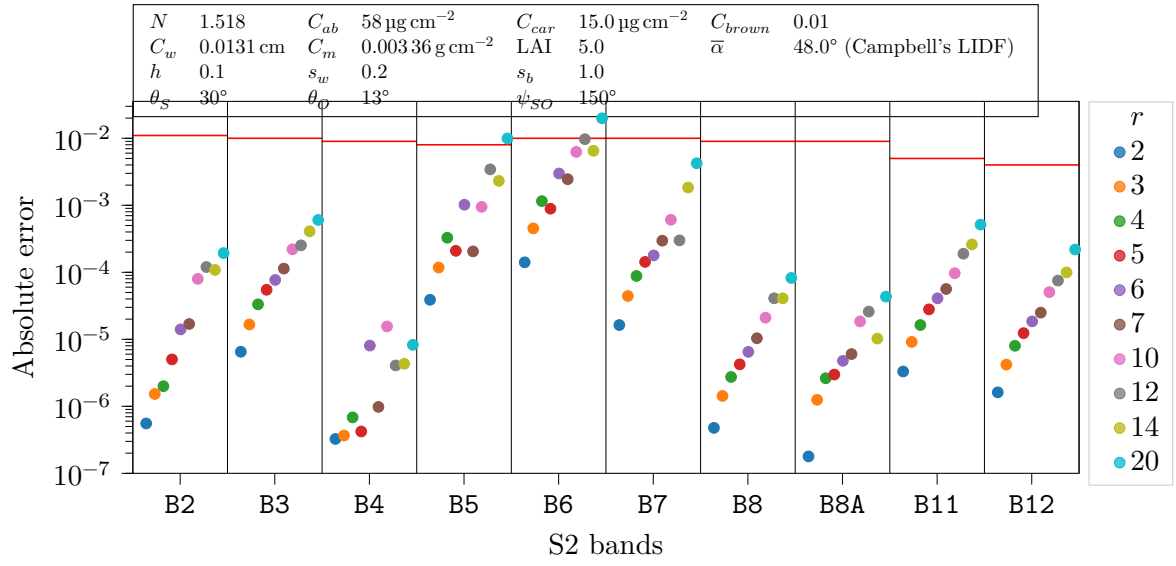


Figure 4.12: Absolute errors of S2 bands simulated with down-sampled PROSAIL as a function of the spectral resolution r. The horizontal red lines are MAJA correction uncertainties.

To select the spectral resolution, the impact of down-sampling PROSAIL over S2 bands simulation must be assessed. In Figure 4.13 are displayed the box-plots of the per-band and per-down-sampling absolute error of simulations w.r.t. a non-down-sampled reference. To do that, $n = 5000$ sets of PROSAIL BV are sampled, using the distributions and procedures described in section 5.1. These sets of BV are then used as input to the down-sampled PROSAIL models with different spectral resolutions, so that the simulations are comparable. Like observed in the sample simulations shown in Figure 4.12, decreasing the spectral resolution increases the simulation error. The comparison of the errors with the MAJA correction uncertainties is used here as a selection criteria for the down-sampling factor. The down-sampling factor is chosen as the maximum values that ensures that the absolute error is negligible (i.e. an order of magnitude below) compared to the MAJA uncertainties, for all S2 bands. For most bands, the down-sampling error is always well below this threshold, even for resolutions up to 20 nm. For the bands `B5` and `B6` however, the error is higher. The resolution factor $r = 7$ nm ensures that the error is 10 times inferior to the MAJA uncertainty for `B5` and `B6`, and all other bands.

The down-sampling error is much higher with `B5` and `B6` than the other bands. This is because these two bands have a narrower spectral support, with a bandwidth of 15 nm (see Table 2.1). This is illustrated by Figure 4.14, in which the spectral response of the band `B5` and several down-samplings are plotted. With increasing down-sampling, the number of points in the discretized sensitivity spectra decreases, and becomes too low to accurately represent the true spectrum. For the band `B5` with $r = 20$, only two points have non zero-sensitivity, and the discretization is significantly different from the original function.

Besides, although the error is overall increasing with the down-sampling factors, for some bands, there are instances where a higher down-sampling yields a lower error (e.g., for `B7`, $r = 12$ has a lower error than $r = 10$). This is because of the mismatch between the sub-sampling grid and the spectrum bandwidth. Some sub-sampling factors lead to a sub-sampling grid that better aligns with the bandwidth.

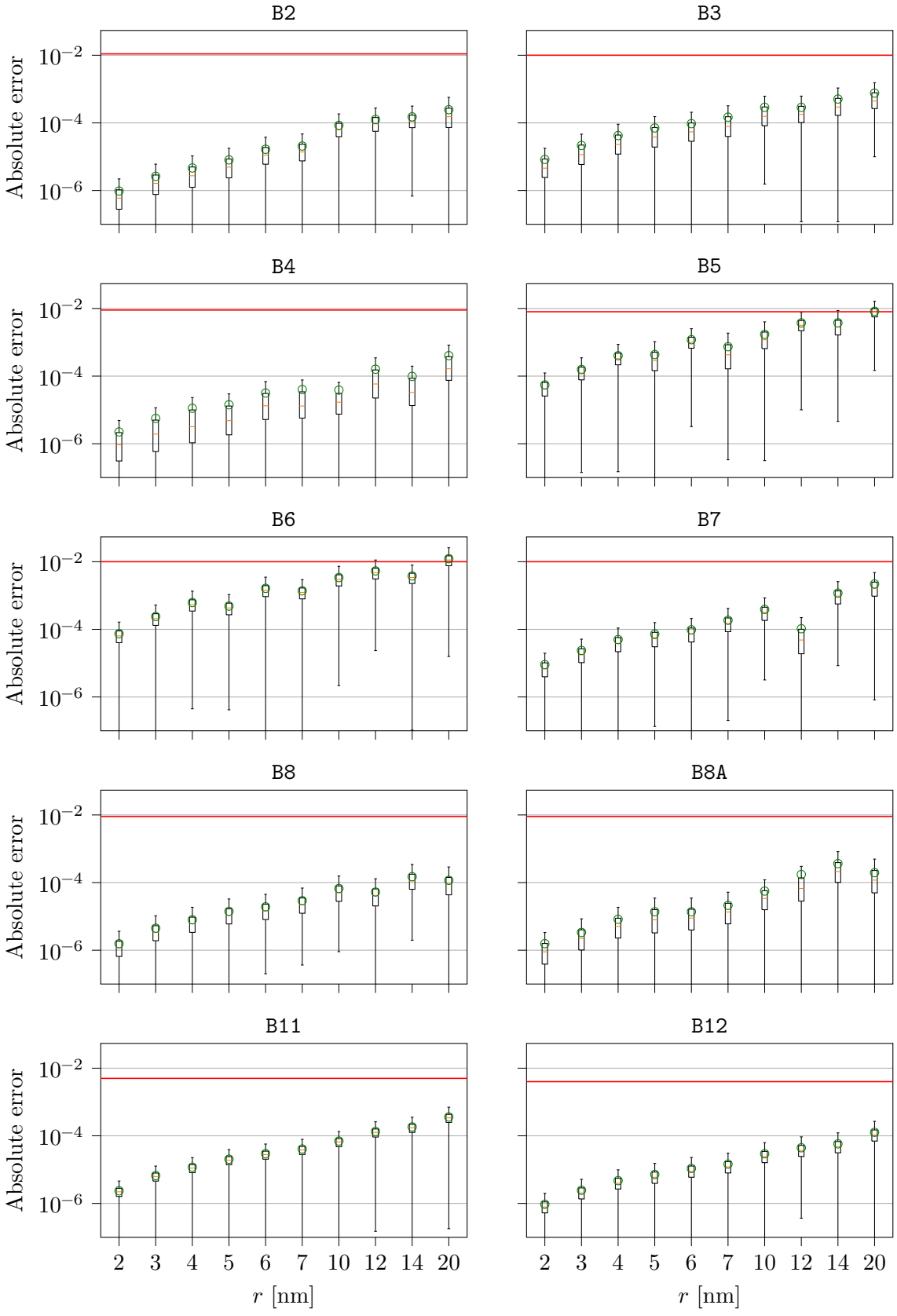


Figure 4.13: Absolute error of S2 bands simulated with a down-sampled PROSAIL as a function of the spectral resolution r ($n = 5000$ samples). Box-plots: absolute error. Green circles: root mean squared error (RMSE). Red line: MAJA correction uncertainty.

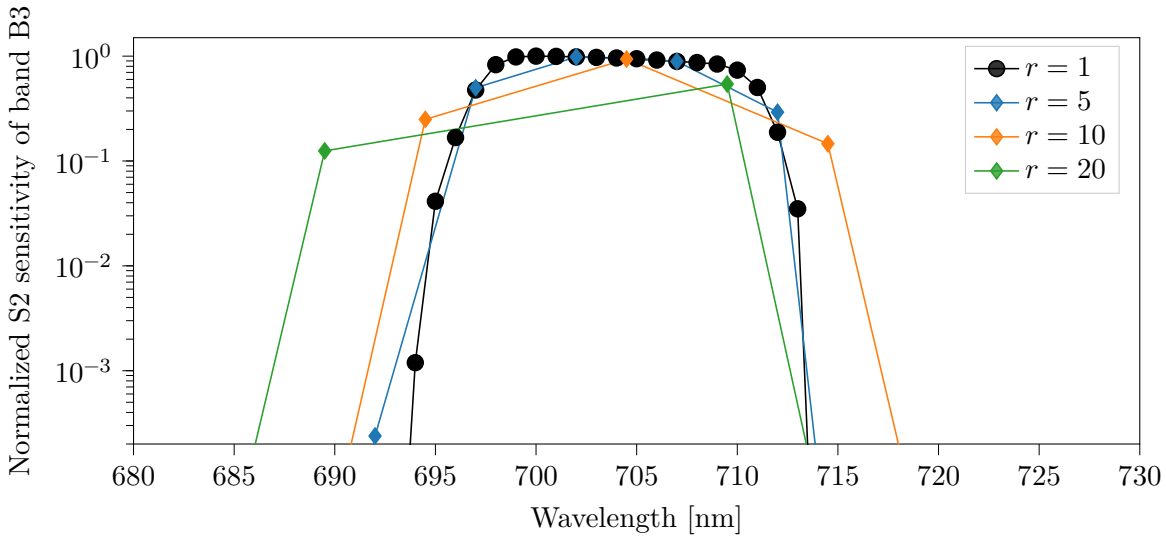


Figure 4.14: Comparison between the different samplings of the S2 MSI spectral response function of the `B5` band.

4.6 Conclusion

PROSAIL, the composite model of the leaf RTM PROSPECT and the canopy RTM SAIL, has been detailed in this chapter. This model enables to link vegetation BV and canopy reflectance, as seen by a sensor. Compared to other canopy RTM, PROSAIL is a relatively simple, 1-D model. However, this enabled to derive a differentiable and parallelized implementation of the model, enabling to integrate it within a DL framework. Also, PROSAIL constitutes the forward model of an inversion problem. Besides, both PROSPECT and SAIL were originally developed with inversion in mind.

Throughout the remainder of this work, the PROSAIL model will be used in two instances. In Chapter 5, PROSAIL will be used as a forward model to generate a training data-base used by artificial neural network (ANN) model for solving the associated inverse problem. In Chapter 8, PROSAIL will be integrated into the physics-informed variational autoencoder (VAE) methodology developed in Chapter 7, to perform the full inversion of the model as an interpretable representation of vegetation.

Chapter 5

Supervised regression with neural networks

Contents

Model inversion is a classical approach for inferring interpretable representations from data, while not having access to sufficient reference data. A very popular approach to model inversion is to use deep learning models to perform supervised regression on simulated data. The model to invert is used to generate the missing reference data to train machine learning methods. The performance of the machine learning model is very dependent on the data it is trained on, and especially, on their distribution. In particular, the end-goal of models trained on synthetic data, is ultimately to be used in real-world scenarios. As such, a mismatch between simulated data and real data may have detrimental consequences on the accuracy of these models. In this chapter, the influence of the distribution of a simulated training data-set over the performance of a machine learning model is investigated. First, section 5.1 presents the simulation procedure to generate a training data-set for an inversion of the PROSAIL model, with simulated Sentinel-2 (S2) reflectances. Then, section 5.2 presents supervised regression experiments where a neural network model is trained with synthetic PROSAIL data. In particular, the influences of the training data simulation process on performances are highlighted.

5.1 Simulation of training data-sets for supervised regression of PROSAIL variables

Training data-set are paramount to neural network learning. In particular, there is a need for data in enough quantity, and diversity, because it must encompass the most cases encountered in real-world applications as possible. While reference data may not be available, fortunately, physical models enable to simulate samples and build a synthetic training data-set. Neural networks trained on synthetic data for supervised regression are *de facto* performing model inversion. In the following, this model is PROSAIL (see Chapter 4). Following the notation introduced in Chapter 3 (see also section H.1), the samples of the training data-set are denoted $(\boldsymbol{x}, \boldsymbol{y})$, with $\boldsymbol{x}$ the vectors of PROSAIL input bio-physical variables (BV) and $\boldsymbol{y}$ the corresponding simulated S2 band reflectances. The inverse model, here a neural network, is trained to infer $\boldsymbol{x}$ from $\boldsymbol{y}$.

This section discusses the simulation with PROSAIL of the data-sets used for training this neural network. In particular, the following subsection 5.1.1 and subsection 5.1.2 details the distribution and sampling procedure of PROSAIL input variables.

5.1.1 Input variable distributions

Generated data-sets are characterized by their distributions, namely, by the joint density between the input $\boldsymbol{x}$ and output $\boldsymbol{y}$ data, each having their own marginal distribution. This synthetic joint density must match the corresponding joint density found in nature, so that a model trained on simulated data can be applied to real data. There is an asymmetrical difficulty in estimating the true distribution of the input $\boldsymbol{x}$ and outputs $\boldsymbol{y}$ of PROSAIL. Indeed, the distribution band reflectance vectors $\boldsymbol{y}$ can be estimated from remote sensing data, much more easily than the distribution of BV vectors $\boldsymbol{x}$ which require field surveys. With a deterministic forward model such as PROSAIL, the distribution of outputs (i.e. reflectance vector) is entirely determined by the distributions of the inputs (i.e. BV vector). Conversely, the distribution of inputs from the distribution of outputs can only be obtained with model inversion, which is more difficult. Therefore, for setting the joint distribution of $(\boldsymbol{x}, \boldsymbol{y})$, the distribution of the input BV vector $\boldsymbol{x}$ is chosen first, and then the distribution of output reflectance vectors $\boldsymbol{y}$, estimated by forward propagation, can be compared to true reflectances.

The distribution of BV is difficult to assess, and it leads to formulating an inverse problem, and using supervised regression to solve it. Yet, creating a synthetic data-set for training a supervised model still requires sampling those distributions. In practice BV individual

range and distribution are roughly estimated from in-situ measurements. Then, the choice of the sampling distribution is made arbitrarily as an informed guess. In this work, the distributions shown in Table 5.1 are used to sample BV and generate training samples with PROSAIL. These distributions are adapted from those ofWeiss and Baret [2016], with both those distributions being qualified as *canonical*.These distributions are almost all truncated normal (TN), and defined by their four parameters. Because their version of PROSAIL (PROSPECT-3 + SAILH) is different from the one used here (PROSPECT-5 + 4SAIL), some variables are different (see Chapter 4), and the associated distributions cannot be directly used.

As the carotenoid content was introduced in PROSPECT-5, the distribution of C_{car} is not available in Weiss and Baret [2016]. In it, the chlorophyll content C_{ab} actually takes carotenoid content into account. Therefore in this work, the distribution C_{car} is chosen to be empirically about $\frac{1}{4}$ of the distribution of C_{ab}, i.e. the lower and upper bounds and the distribution mean are about one fourth of the C_{ab} counterparts. This choice follows an oral advice provided by F. Baret to J. Inglada (supervisor of this work), and is consistent with measured relative carotenoid to chlorophyll concentrations [Thomas and Gausman, 1977].

Similarly, the PROSPECT version of Weiss and Baret [2016] doesn't use equivalent water thickness C_w, but the relative water content ($C_{w,\text{rel}}$) instead. $C_{w,\text{rel}}$ is related to C_w and C_m through the formula:

$$C_{w,\text{rel}} = \frac{C_w}{C_w + C_m} \implies C_w = C_m \frac{C_{w,\text{rel}}}{1 - C_{w,\text{rel}}}. \tag{5.1}$$

Therefore, the C_w samples necessary for the PROSPECT version used here are derived from samples of $C_{w,\text{rel}}$ and C_m, following the distributions defined in Weiss and Baret [2016] and summarized in Table 5.1.

The soil reflectance spectrum input to Scattering by Arbitrary Inclined Leaves (SAIL) is chosen differently than Weiss and Baret [2016]. In their work, a reference soil spectrum is drawn from a catalog, and scaled with a soil brightness factor s_b. Here, a synthetic spectrum is made with the sum of a dry soil and a wet soil spectra, weighted by a soil wetness factor s_w as an additional input variable (see 4.2.2). s_w is simply drawn uniformly. The synthetic spectrum is then scaled with s_b, and the distribution of this parameter is kept identical to that of Weiss and Baret [2016].

Table 5.1: Canonical sampling distributions of PROSAIL parameters

Variable v	Distribution	Range		Distribution parameters	
		$v_{l,0}$ (min)	$v_{u,0}$ (max)	μ_v (mode)	σ_v (std)
N	TN	1.2	2.2	1.5	0.3
C_{ab}	TN	20	90	45	30
$C_{w,\text{rel}}$	TN	0.60	0.85	0.75	0.08
C_{car}	TN	5	23	11	5
C_m	TN	0.003	0.011	0.005	0.005
C_{brown}	TN	0.0	2.0	0.0	0.3
LAI	TN	0	15	2	3
$\overline{\alpha}$	TN	30	80	60	20
h	TN	0.10	0.50	0.25	0.50
s_w	Uniform	0	1	-	-
s_b	TN	0.3	3.5	1.2	2.0

Finally, the angular parameters of each observation θ_S, θ_O and ψ_{SO}, involved in SAIL, are simulated from S2 orbital characteristics, by uniformly drawing random dates and locations, like in Weiss and Baret [2016]. The samples drawn are shown in Figure E.1.

5.1.2 Input variable co-distributions

In nature, the different quantities represented by the input variables of PROSAIL or other models are correlated. This means that these variables are governed by a joint distribution, which can't be described with independent marginal distributions. For example, the sampled observation angles shown in Figure E.1 correspond to real satellite configurations, and their joint values are restricted to a specific domain. However, for other PROSAIL variables, accessing the joint distribution is intractable in practice. In-situ measurements enable to estimate the marginal distribution of a given vegetation variable that is used as a radiative transfer model (RTM) input. However estimating the joint distribution of variables would require measuring jointly the associated quantities in field surveys. While some joint measurements of biophysical variables do exist [Hosgood et al., 1993], they remain quite limited. Indeed, they only collect data about a restricted number of vegetation types, at certain seasonal stages, etc. It is fundamentally impossible to measure exactly the state of vegetation in all of its aspects, so a given model may always have an input BV that hasn't been measured in-situ.

This is why, to perform sampling, correlations between variables are in practice set as empirical, arbitrary relationships. The LAI is a global variable that can be more easily related to other vegetation variables. With a high LAI value that indicates a certain density of vegetation, it is reasonable to assume that other BV, such as the chlorophyll content cannot be low. Linking the LAI with other BV can be performed with a *co-distribution* function, which is a function that restricts the range of variation $[v_{l,0}, v_{u,0}]$ of a BV v along with the value of the LAI. The co-distribution is applied to a given BV v after it has been sampled, and is transformed with a sampled LAI into a variable $v^\star$.

In this work, two linear co-distributions are considered. The first one, hereby designated as co-distribution type 1, is defined in Inglada [2017], with the Equation 5.2:

$$v^\star = f_1\left(v, \mu_v, \mathrm{LAI}, c_{v,\mathrm{LAI}}\right) = \mu_v + (v - \mu_v)\left(1 - \frac{\mathrm{LAI}}{c_{v,\mathrm{LAI}}}\right). \tag{5.2}$$

This co-distribution requires μ_v, the mode of the marginal distribution of v, and $c_{v,\mathrm{LAI}}$, a constant which sets the LAI for which $\forall v, v^\star = \mu_v$. It must be noted that this was erroneously implemented for this work, but it lead to interesting observations. In Inglada [2017], the use of the co-distribution assumes that $\mathrm{LAI} < c_{v,\mathrm{LAI}}$, with, in practice, $\mathrm{LAI} \in [0, \mathrm{LAI}_T]$, $\mathrm{LAI}_T = 5$ and $c_{v,\mathrm{LAI}} = 10$. However, in this work, while sampling sets of PROSAIL BV, the LAI was allowed to be sampled in the wider range $[0, \mathrm{LAI}_{\max}]$, $\mathrm{LAI}_{\max} = 15$. Consequently, there were instances of sampled $\mathrm{LAI} > \mathrm{LAI}_T$. To avoid having the range of $v^\star$ expanding for $\mathrm{LAI} > c_{v,\mathrm{LAI}}$, the co-distribution type 1 is modified, from Equation 5.2 to Equation 5.3:

$$v^\star = f_1\left(v, \mu_v, \mathrm{LAI}, c_{v,\mathrm{LAI}}\right) = \mu_v + (v - \mu_v)\left(1 - \frac{\min\left(\mathrm{LAI}, c_{v,\mathrm{LAI}}\right)}{c_{v,\mathrm{LAI}}}\right). \tag{5.3}$$

This co-distribution is plotted in Figure 5.1. It highlights that the implementation mistake modifies the behavior of the warping function. In the implemented version, the transformed BV $v^\star$ saturates at μ_v for $\mathrm{LAI} \geq c_{v,\mathrm{LAI}}$, whereas in the original version, the range of $v^\star$ is restricted, but never reduced to a point.

In Weiss and Baret [2016], another co-distribution, here called type 2, is defined to sample PROSAIL BV for the training data-set of the regression neural network of Sentinel Application Platform (SNAP)'s Biophysical Processor (BP). This function, defined in Equation 5.4, warps v into $v^\star$ between a lower and upper bounds that vary linearly with the LAI.

$$v^\star = f_2\left(v, v_{l,0}, v_{u,0}, v_{l,M}, v_{u,M}, \mathrm{LAI}\right) = \frac{\left(v - v_{l,0}\right)\left(f_u\left(\mathrm{LAI}\right) - f_l\left(\mathrm{LAI}\right)\right)}{v_{u,0} - v_{l,0}} + v_{l,M} \tag{5.4}$$

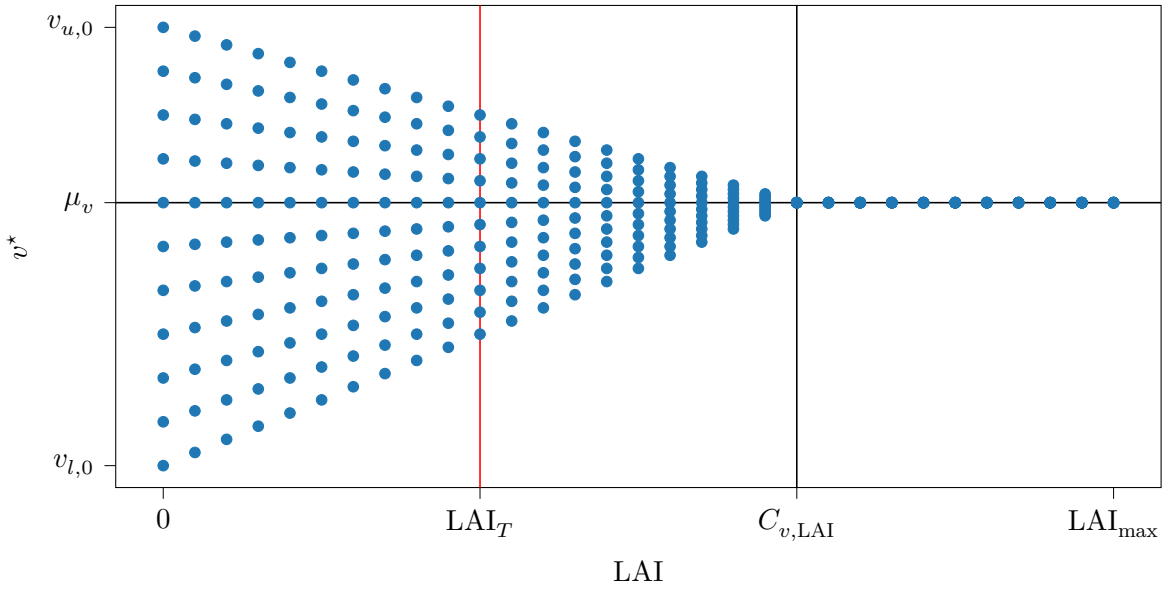


Figure 5.1: Warping of a variable v with the LAI into $v^\star$, with the co-distribution type 1.

The function f_l (resp. f_u) defined in Equation 5.5 (resp. Equation 5.6), is the lower (resp. upper) bound of $v^\star$ as a function of the LAI.

$$f_l\left(\text{LAI}\right) = v_{l,0} + \frac{\text{LAI}}{\text{LAI}_{\text{max}}}\left(v_{l,M} - v_{l,0}\right) \tag{5.5}$$

$$f_u\left(\text{LAI}\right) = v_{u,0} + \frac{\text{LAI}}{\text{LAI}_{\text{max}}}\left(v_{u,M} - v_{u,0}\right) \tag{5.6}$$

This co-distribution requires 4 constants:

- $v_{l,0}$ and $v_{u,0}$, which are the lower and upper bounds of the marginal distribution of v, and are the bounds of $v^\star$ for $\text{LAI} = 0$,
- $v_{l,M}$ and $v_{u,M}$, which are the lower and upper bounds of $v^\star$ for $\text{LAI} = \text{LAI}_{\text{max}}$.

The co-distribution type 2 is a little more flexible than type 1. It can be noted that type 2 is similar to the original type 1, in that it doesn't saturate the LAI after a certain threshold, but warps v with converging bounds. The constants required for both co-distribution types are provided in Table 5.2. Its values are taken from Inglada [2017] for co-distribution type 1 and Weiss and Baret [2016] for co-distribution type 2. As neither C_{car} and s_w weren't used in either of these works, these parameters are excluded from the co-distribution computation.

5.1.3 Simulated reflectance data-sets

As the distribution of PROSAIL input variables are defined, a simulated data-set can be generated. Samples of these data-sets are generated by a three step procedure:

1. Sets of PROSAIL input parameters are drawn from marginal sampling distributions, as described in Table 5.1. Observation angles are drawn separately (see Figure E.1).
2. The PROSAIL input BV drawn are transformed linearly with the LAI, using co-distribution type 1 (see Equation 5.3) or type 2 (Equation 5.4) along with the constants defined in Table 5.2. The distribution of samples after applying co-distribution 1 is shown in Figure E.2, and in Figure E.3 for co-distribution type 2.
3. The sets of sampled and correlated variables are taken as input to PROSAIL to simulate canopy reflectance spectra, then S2 band reflectances.

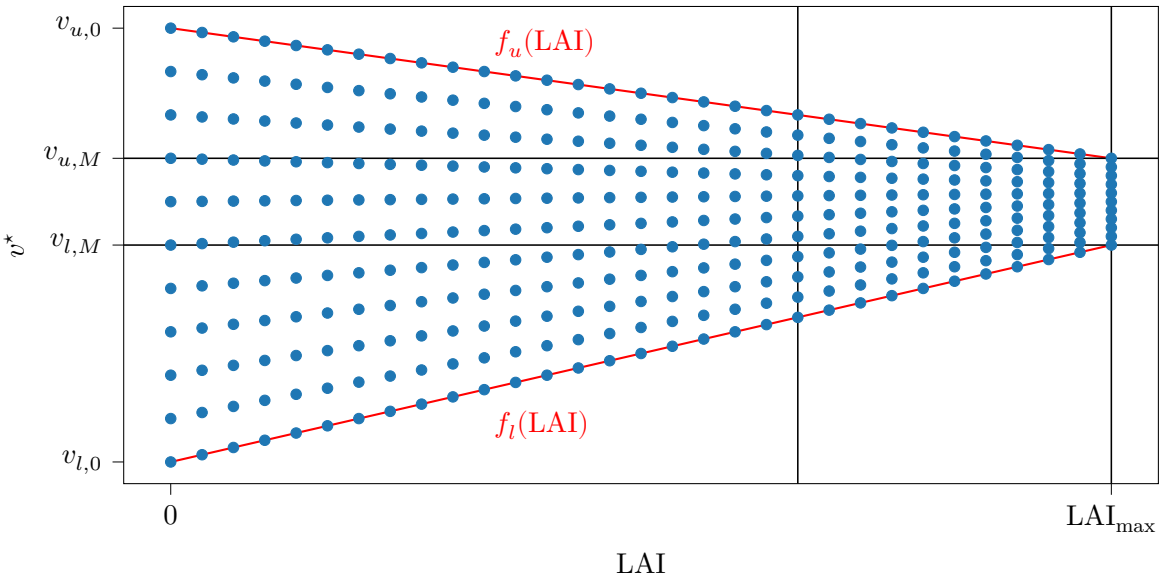


Figure 5.2: Warping of a variable v with the LAI into $v^\star$, with the co-distribution type 2.

Table 5.2: Sampling co-distributions parameters for PROSAIL variables.

Variable v	Co-distribution type 1 parameter C_{LAI}	Co-distribution type 2 parameters $v_{l,M}$	$v_{u,M}$
N	10	1.3	1.8
C_{ab}	10	45	90
$C_{w,\text{rel}}$	10	0.70	0.80
C_{car}	-	-	-
C_m	10	0.005	0.0110
C_{brown}	10	0.0	0.2
LAI	-	-	-
$\overline{\alpha}$	10	55	65
h	10	0.1	0.5
s_w	-	-	-
s_b	10	0.5	1.20

5.2 Limitations of pre-simulation for PROSAIL inversion

Using data-sets generated with the above-defined sampling strategy, a neural network model is proposed in subsection 5.2.1, to invert the PROSAIL model. Such data-sets are *pre-simulated*, because they are generated prior to and independently from the training. Using variations of this synthetic data-set, subsection 5.2.2 highlights the regression performance dependence on data-set distribution.

5.2.1 BVNET

Biophysical variable neural network (BVNET) is originally a MATLAB tool developped by INRA[1], that manages the training of regression neural networks that perform the inversion of PROSAIL to retrieve biophysical variables. In particular, this tool enabled to train the neural network that is used in the BP of SNAP. In this thesis, BVNET will instead refer to the class of neural networks whose architecture is the one used in SNAP's BP. Specifically, BVNET models take eight S2 band reflectances and observation angles cosines as input, and predict a BV: (LAI, canopy chlorophyll content (CCC), canopy water content (CWC), fraction of vegetation cover (F-COVER) or fraction of absorbed photosynthetically active radiation (FAPAR). Each BV is predicted with a different neural network [Weiss and Baret, 2016]. The *canonical* BVNET models in SNAP were trained using PROSPECT-3+SAILH simulations. BVNET models have a very simple two-layered architecture, with only 66 trainable weights (see Figure 5.3). The inputs x of the model are normalized into $\tilde{x}$ as follows:

$$\tilde{x} = 2\frac{x - \min(x)}{\max(x) - \min(x)} - 1, \tag{5.7}$$

and the predicted BV y is obtained after "de-normalizing" the network output $\tilde{y}$ as follows:

$$y = \frac{1}{2}(\tilde{y} + 1)(\max(y) - \min(y)) + \min(y). \tag{5.8}$$

BVNET can hardly be qualified as a *deep neural network*. On the one hand, this puts a limit on the capacity of the model to perform the inversion, and to generalize to a wide variety of cases. On the other hand, the low complexity of the model ensures fast convergence, and prevents overfitting, even with relatively small training data-sets. Specifically, in Weiss and Baret [2016] a data-set of 41472 samples is used for training a BVNET, which is a small number for current deep-learning models, but is enough in this application.

In the remainder of this manuscript, the BVNET models that are used to predict BV in SNAP's BP (with *canonical* weights) will simply be referred as SNAP, whereas "BVNET models" will refer to neural networks that were trained during this thesis. In the following, the training of BVNET models is performed with simulated data-sets. They are optimized with a mini-batch gradient descent strategy. They are initialized with the multiple initialization and best instance training (MIBIT) strategy (see subsubsection 3.3.2.3), and the learning rate (lr) is scheduled with a cyclical plateau reduction (CPR) (see subsubsection 3.3.2.2). The mean squared error (MSE) loss of the BV estimate $\hat{y}$ from the simulated reference BV y is used:

$$\mathcal{L}_{\text{BVNET}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2, \tag{5.9}$$

with N the batch size. The hyper-parameters and configuration for BVNET trainings are provided in Table 5.3.

It can be noted that it is possible to initialize BVNET with the canonical weights of SNAP. This neural network can then be used as-is, or it can be further trained on other data-set. In the latter case, the model benefits from an admittedly good initialization.

[1]Institut National de la Recherche Agronomique, that merged with Institut national de recherche en sciences et technologies pour l'environnement et l'agriculture (IRSTEA) into Institut national de recherche pour l'agriculture, l'alimentation et l'environnement (INRAe) in 2020.

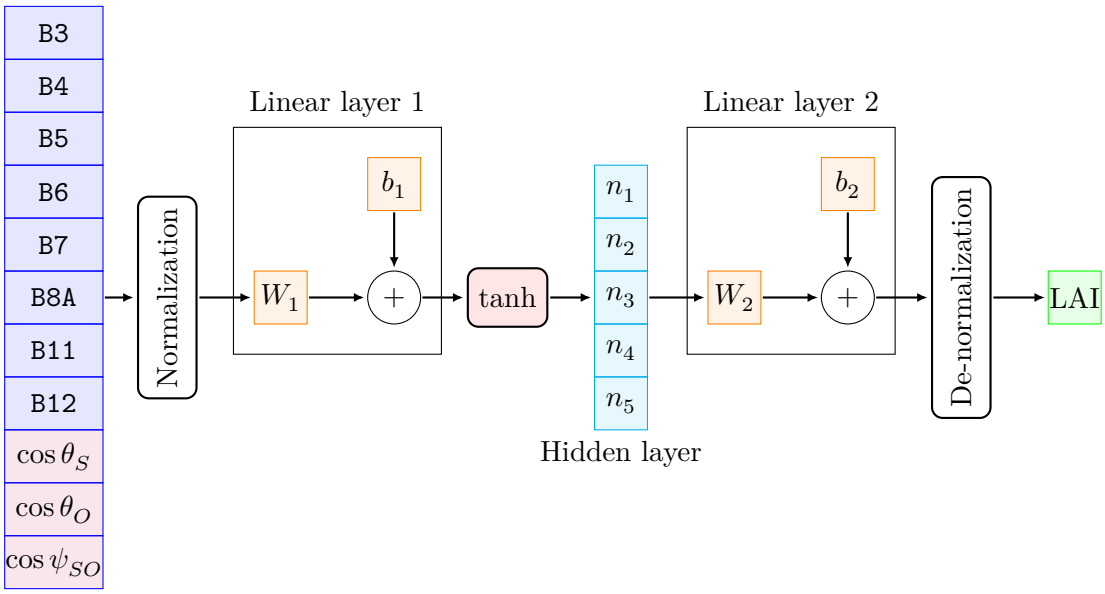


Figure 5.3: BVNET neural network architecture.

Table 5.3: Training configuration and hyperparameters for BVNET.

Training	Optimizer	Adam
	Batch size	5000
	Epochs	500
lr	lr scheduler	CPR
	lr at start of training	10^{-3}
	lr reduction factor	10
	Minimum lr	10^{-8}
Initialization (MIBIT)	Number of initialized models	10
	Number of epochs	20
	lr	10^{-3}

5.2.2 The effect of a LAI distribution mismatch

In the following, the effect of the pre-simulated training data-set on neural-network inversion performances is assessed with respect to (w.r.t.) the marginal distribution of the parameters. First, data-sets with different distributions are generated. Specifically, these data-sets are sampled with the canonical PROSAIL BV distributions, except for the LAI distribution which is made to vary. Then, BVNET models are trained using these data-sets, and their performances on LAI retrieval are compared.

5.2.2.1 Data-sets

To evaluate the effect of a distribution difference between data-sets, a reference *testing data-set* (see subsubsection 3.3.2.4) $\mathcal{D}_{\text{test}}$ is first created. $\mathcal{D}_{\text{test}}$ is generated with PROSAIL (PROSPECT-5 + 4SAIL) as depicted in sec.5.1.3, with the distributions in Table 5.1 and the co-distribution type 2 with the constants of Table 5.2. A number $N_t = 40000$ samples are drawn.

Then, different data-sets $\mathcal{D}_i$ with $N_i = 40\,000$ samples are produced, with identical distributions and co-distributions than $\mathcal{D}_{\text{test}}$, except for the distribution of LAI. The LAI is sampled from a TN distribution p_i with range $[0, 15]$, and with parameters $\mu_i \in \{0, 1, 2, 3, 4\}$, $\sigma_i \in \{0.5, 1, 2, 3, 4\}$. The case $\mu_i = 2$ and $\sigma_i = 3$ matches the LAI distribution of $\mathcal{D}_{\text{test}}$. The data-sets are randomly split into a training data-set $\mathcal{D}_{\text{train},i}$, and validation data-set $\mathcal{D}_{\text{valid},i}$, with $N_{\text{train},i} = 38\,000$ and $N_{valid,i} = 2000$ samples (5%).

For each training data-set, $n = 10$ BVNET models are trained, to ensure that training randomness doesn't affect the results.

Also, the in-situ data-set of section 2.4, with measured LAI and corresponding true S2 bands $\mathcal{D}_{\text{IS}}$ (see section 2.4), is also used as a testing data-set, as a complement to the simulated $\mathcal{D}_{\text{test}}$.

5.2.2.2 Regression performance and distribution divergence

The root mean squared error (RMSE) is the metric used (see Equation 3.7) to quantify the LAI inference performance on testing data-sets. The mismatch between the distribution $p_{\text{train},i}$ of a simulated training data-set and p_{test} of simulated testing data-set is quantified with the Kullback-Leibler divergence (KLD) between the theoretical sampling distributions of LAI of both data-sets: $\mathrm{D}_{\mathrm{KL}}\left(p_{\text{test}} \| p_{\text{train},i}\right)$ (see subsubsection C.4.6.1 for derivation and formula of KLD between TN distributions).

The effect of a difference in LAI sampling distribution between training and testing data-sets over prediction performance is shown 5.4. For both testing data-sets $\mathcal{D}_{\text{test}}$ and $\mathcal{D}_{\text{IS}}$, the results show that there is a sensitivity of performances to the distribution of the training data-set, as the RMSE is not constant. Overall, the more the training distribution diverges from the evaluation data-set, with the KLD between them increasing, and the worse the inference performance is, on both $\mathcal{D}_{\text{test}}$ and $\mathcal{D}_{\text{IS}}$. The best performing BVNET configuration on $\mathcal{D}_{\text{test}}$, is the one that has the same sampling LAI distribution ($\mu_{\text{train}} = 2$, $\sigma_{\text{train},i} = 3$). The behavior of the LAI RMSE over the simulated evaluation data and the in-situ evaluation data-set is similar. It can be remarked that the RMSE increase with the KLD distributions isn't monotonic. This suggests that the KLD as a distribution mismatch quantification may not be the best performance indicator. Perhaps another distribution divergence would show better correlation with the prediction error.

It can be noted that the BVNET trained with a data-set with $\mu_{\text{train},i} = 4$ and $\sigma_{\text{train},i} = 0.5$, with high KLD (8.75), has a similar performance on in-situ data than BVNET with lower KLD. This can be explained by considering that the reference LAI in in-situ data has an average of 3.7 (excluding soil measurements), which is close to the mode $\mu_{\text{train},i} = 4$. With low LAI standard deviation (std) ($\sigma_{\text{train},i} = 0.5$), these models specialize in retrieving LAI that are close to that observed in $\mathcal{D}_{\text{IS}}$. Conversely, these BVNET perform worse on $\mathcal{D}_{\text{test}}$, as

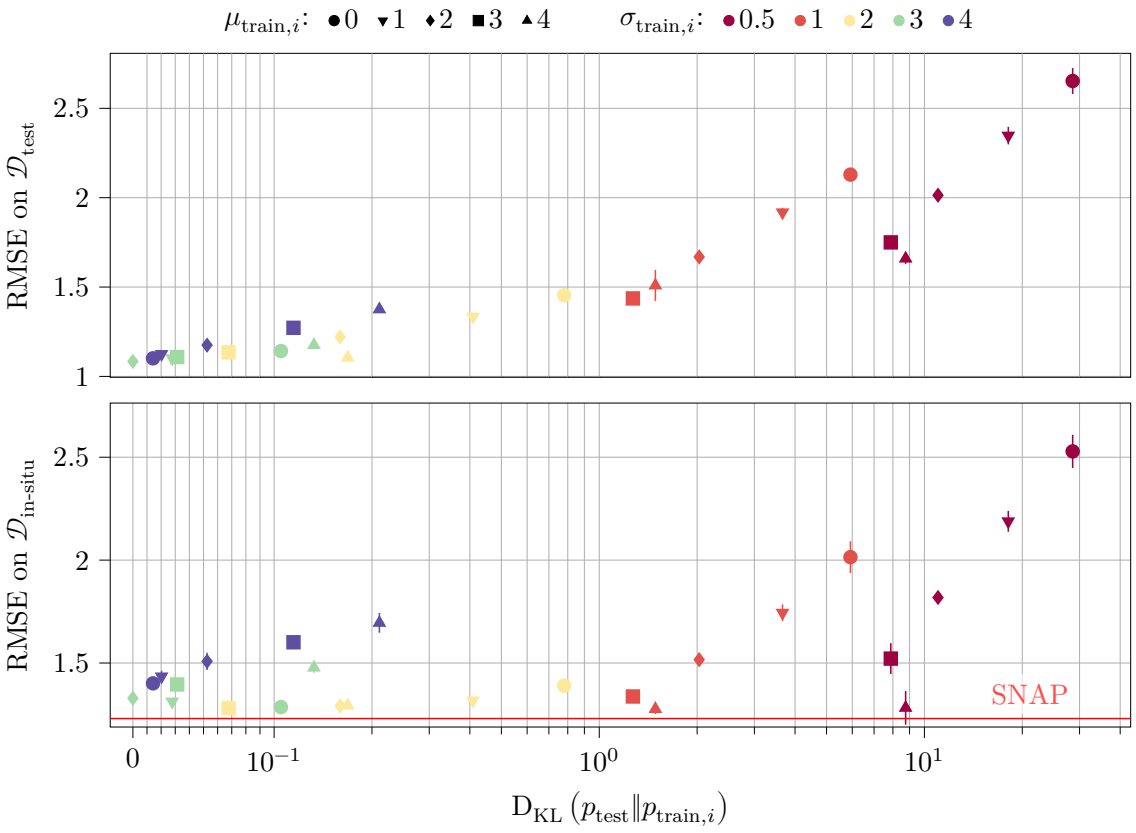


Figure 5.4: RMSE of LAI regression (average over $n = 10$ models) on $\mathcal{D}_{\text{test}}$ and $\mathcal{D}_{\text{IS}}$ as a function of the KLD between the LAI distributions of the simulated training and testing data-sets, and comparison with SNAP (horizontal red lines).

they are too specialized on LAI that are away from the mode $\mu_{\text{test}} = 2$ of $\mathcal{D}_{\text{test}}$. Similarly, all BVNET configurations with $\mu_{\text{train},i} = 4$ have a RMSE on $\mathcal{D}_{\text{IS}}$ that is close to the minimum.

Finally, some BVNET trained here have a comparable LAI RMSE to SNAP on $\mathcal{D}_{\text{IS}}$. With low KLD, these models have the distributions that match best that of Simplified Level 2 Product Prototype Processor (SL2P)'s BVNET. This corroborates that the LAI distribution of the evaluation data-set (that is also the distribution used to train SNAP) is well-chosen, as changing it leads to a decrease in prediction performance on the in-situ data. Interestingly, SNAP is not the best model on $\mathcal{D}_{\text{test}}$. Despite the distribution of training data-sets being close to that of SNAP, there are differences in simulations due to the difference in PROSAIL versions. The influence of the model configuration over performances is further highlighted in the following subsection 5.2.3.

The influence of a mismatch of a variable of interest distribution between the available training data and evaluation data over supervised regression accuracy, is not unique to the retrieval of LAI with BVNET, or even to space-borne remote sensing. For instance, Yang et al. [2022] attempts to perform the inversion of a so-called seismic full-waveform model to retrieve seismic wave velocity from seismic data in a scenario of CO_2 leakage from an underground storage reservoir. They perform inversion with "InversionNet" a supervised regression neural network which can be compared with BVNET. Available (simulated) training data-sets under-represent certain types of CO_2 leakage scenario. By using data-augmentation to add training samples for these scenarios, their InversionNet performance improves significantly.

5.2.3 The effect of model version and variable co-distributions

The previous subsection focused on the influence of the distribution of a single parameter, the LAI, in the inversion of PROSAIL. However, as developed in subsection 5.1.2, model parameters, matching real-world variables, are governed by joint distributions, and not simply their marginal distributions. As these joint distributions are usually intractable, arbitrary co-distributions are used instead. Furthermore, the specific model used for pre-simulating training samples is not unique. The different versions of PROSAIL can be considered as different models altogether. In the following, how the choice of an empirical relationship between variables affect the performance, is investigated for the inversion of PROSAIL. Additionally, the influence of the version of the PROSAIL model used to generate training samples is assessed. Also, the retrieval performance will be estimated on LAI and CCC.

5.2.3.1 Data-sets

Similarly to subsubsection 5.2.2.1, different training data-sets are generated with PROSAIL, this time to evaluate the effect of the choice of a PROSAIL model, and the input variable co-distribution. More specifically, the effect of the PROSPECT leaf RTM is evaluated (see section 4.1). For this goal, four different training data-sets are simulated by following the procedure described in 5.1.3. Each of those data-sets is built with a combination of a PROSPECT model version (5 or D) and with a co-distribution type. Besides, the distribution of all variables is set as described in Table 5.1. These data-sets are denoted $\mathcal{D}_{V,i}$, with $V \in \{5, D\}$ the PROSPECT version and $i \in \{1, 2\}$ the co-distribution type (see subsection 5.1.2).

As discussed in section 4.1, the difference between PROSPECT-5 and PROSPECT-D lies in the introduction of an anthocyanin content C_{ant}, and in the re-calibration of the leaf refraction index and leaf specific absorption spectra. To keep the parameter distribution independent of the model version, the required anthocyanin content is set to $0.0\,\mu g\,cm^{-2}$ when generating a data-set with PROSPECT-D.

After generating the data-sets, they are randomly split into a training data-set $\mathcal{D}_{\text{train},V,i}$, and validation data-set $\mathcal{D}_{\text{valid},V,i}$, with $N_{\text{train},V,i} = 38000$ and $N_{valid,V,i} = 2000$ samples (5%). Using these data-sets, different BVNET models that predict either LAI or CCC are trained.

The CCC samples are simply generated from each data-set as the product of the LAI and C_{ab}. For each data-set $\mathcal{D}_{V,i}$, there are $n = 20$ BVNET models trained.

In this experiment, a synthetic testing data-set is not used. After training, the BVNET tuned on $\mathcal{D}_{V,i}$, denoted $\text{BVNET}_{V,i}$, are compared through their retrieval performance on the in-situ data-set $\mathcal{D}_{in-situ}$ for LAI and CCC (see section 2.4).

5.2.3.2 Regression performance as a function of PROSPECT version and variable co-distributions

The regression performances of LAI and CCC (RMSE) on the in-situ testing data-set $\mathcal{D}_{\text{IS}}$ of BVNET trained on each data-set are provided in Figure 5.5.

A quantitative evaluation is performed using the previously described in-situ LAI and CCC validation data (see section 2.4). Figure 5.5 shows the obtained results which corroborate that the different simulation engineering designs impact the BVNET performances. Unfortunately, concluding which configuration to choose from the results is not straightforward. For instance, co-distribution type 2 seems to improve LAI predictions whereas it deteriorates the predictive performances of CCC. Despite being designed "erroneously" (see subsection 5.1.2), the co-distribution type 1 is better for performing CCC regression. This highlights how carefully sampling variables with a "well-chosen" distribution doesn't necessarily imply optimal performances.

Using PROSPECT-5 instead of PROSPECT-D increases the accuracy of LAI predictions no matter the co-distribution used. PROSPECT-5 obtains slightly better performances for CCC than PROSPECT-D for the co-distribution type 1. Conversely, the predictive accuracies of these models decrease when the co-distribution type 2 is used. It can be noted that the difference in performance observed here for CCC retrieval between PROSPECT versions in training data-sets is corroborated in Hauser et al. [2021].

SNAP obtains the best performance on LAI retrieval, with only $\text{BVNET}_{5,2}$ matching it. However, the situation is very different for the CCC retrieval, as the RMSE of SNAP is much higher than the other. The impact of simulation modeling design on predictive performances explains why results obtained by the $\text{BVNET}_{v,i}$ models are different from the ones reached by SNAP. This is because the simulated data-set used to train SNAP is generated by PROSPECT-3 model (this model doesn't differentiate carotenoid from chlorophyll pigments, see section 4.1). Besides, different strategies are used to characterize soil spectra required by the SAIL model.

Another important remark of the obtained results is that the best performances of LAI and CCC are not reached by the same training data-set.

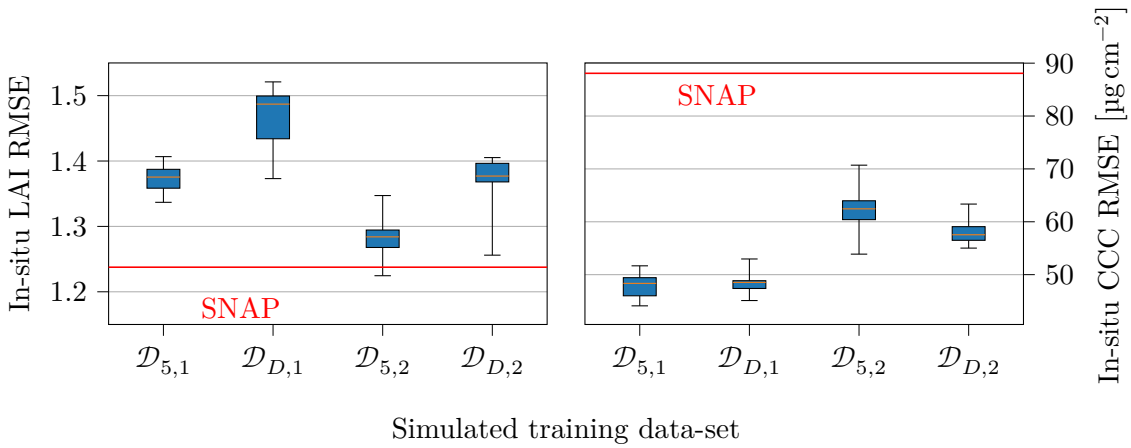


Figure 5.5: Box-plots of RMSE of LAI and CCC on in-situ testing data $\mathcal{D}_{\text{IS}}$ for BVNET trained with 4 different data-sets. The data-sets naming convention is $\mathcal{D}_{v,i}$, with v being the PROSPECT version and i being the co-distribution type used to generate the data-set samples. The horizontal red line indicates SNAP metrics.

5.3 Arbitrary joint distributions and model inversion

In this chapter, the preponderant influence of a training data-set simulation for model inversion with neural networks has been shown. Because the true distribution of the forward model input parameters is unknown, arbitrary choices must be made. When generating a training data-set with a forward model, this choice of the input parameter distributions greatly affects the prediction performance of a model trained on it. The marginal distribution of each parameter has an influence. When the chosen training distribution diverges from the testing distribution the model infers on, retrieval performances decline. Actually, when studying the LAI found in nature, one finds that its distribution is dissimilar to the one proposed for sampling training data-sets. The true LAI distribution is better fitted with a log-normal distribution, heavily skewed toward lower values, whereas a TN centered around $\mu = 2$ is used instead for training.

Moreover, even the correlations between the sampled parameters matters, making the distribution choice even more difficult, because such correlations are even less established. As it could be expected, even with identical input distributions, changing even slightly the forward model used for data-set pre-simulation also has an influence. When contemplating the choice between different forward models for simulation, one must also consider that their optimal input parameter distribution may be different.

Furthermore, when considering the retrieval of several variables in model inversions, the experiments have shown that a training data-set that works best for a given variable isn't necessarily optimal for others. Thus, ideally, each retrieved variable should be predicted by a model trained with a dedicated data-set, to ensure optimal performance. Finally, the distributions and models are only suited for certain cases (e.g. PROSAIL doesn't correctly describe all kinds of vegetation).

Designing and simulating a training data-set for model inversion with neural networks takes a considerable amount of time and effort. Yet, all this work must be re-started over, should the application change ever-so slightly: different observations, different sensors, different models, etc. This approach remains labor-intensive. Therefore, a method that bypasses the strenuous distribution selection for data-set pre-simulation would be interesting. Such an approach, based on the theory introduced in Chapter 7, is presented in Chapter 8.

Part III

Unsupervised Bayesian learning of vegetation representations

Chapter 6

Stochastic modeling and variational inference

Contents

In nature, observed phenomena are not directly accessible, they can not be known with a full certainty. In fact, these phenomena can only ever be apprehended through representations (see section 1.3). Therefore, it is useful to think of these phenomena as *random experiments*, on which the *outcomes*, or *events* are what is measured. The randomness in measuring these phenomena can have multiple sources. The phenomena in itself can be random in nature, such as quantum processes, for which Heisenberg's indeterminacy principle imposes a threshold under which the variance of certain pairs of physical properties cannot be lowered. Stochasticity in the observation of the data itself can appear with measurement noise. Randomness can also stem from the intractability of retrieving the full set of causes of an event. For instance, a remote sensor may observe a sudden change in its observation of a scene, (e.g. clouds obstructing the direction of view, crops being harvested, urban constructions, etc...), that can appear as random because context was missing from the sensor's point of view. Phenomena can be aleatoric in nature. But even if they aren't, knowledge about them isn't perfect and absolute but rather partial and flawed, i.e. they are *uncertain*.

There is also uncertainty in how observed data can be explained, in how much a representation of a process (i.e. a *model*, see subsection 1.3.1) accurately captures its true nature. For instance, Newton's theory of gravitation proposes that gravity is a an attractive force between any two objects that have mass. However, Einstein's theory of general relativity instead proposed that gravity isn't a force *per se*, but a geometric effect of space-time itself being curved by mass. The latter model of gravity is arguably closer to reality, and reduces the uncertainty in our knowledge of the phenomenon. It enabled to take into account other observations and make more accurate predictions. However, this model probably still isn't the *whole truth*. For instance, gravity still hasn't been reconciled with quantum mechanics. Thus the popular saying "all models are wrong, but some of them are useful". While models can never account for all aspects of reality, they may give predictions accurate to a certain degree. Even if "false", Newton's model of gravity is still enough to explain the vast majority of celestial movements. Likewise, PROSAIL (see Chapter 4) is not a perfect canopy radiative transfer model (RTM), but it has been proved to allow accurate simulations.

In this chapter, probabilistic modeling that takes into account uncertainty of studied phenomena into account, is introduced with section 6.1. Then inference methods that attempt to retrieve underlying causes from observed data, are developed with Bayesian theory in section 6.2. Variational inference, as an approximate Bayesian inference method is then discussed in section 6.3, to lay the theoretical foundations of variational autoencoder (VAE) explained in section 6.4. Finally, the disentanglement approaches which attempt to impose particular solution to learned representations are discussed in section 6.5.

6.1 Stochastic modeling

At the basis of probabilistic modeling lies some observed data $\boldsymbol{x}$ (in a vectorized form), part of a data-set $\mathcal{D}_{\boldsymbol{x}}$, which is assumed to be drawn from a random variable $\mathbf{x}$ with domain $\mathbb{X}$. The significance of this probability attached to observations is discussed in section 6.2. A *statistical model* of the data is a set, or family of probability distributions over $\mathbb{X}$ (e.g. exponential, Dirichlet distribution, etc...). A *probabilistic model* (or *stochastic model*) is one such probability distribution, i.e. it is an element of a statistical model. For continuous data, which are exclusively considered here, a stochastic model is a a density function over the data: $f_{\mathbf{x}}(\boldsymbol{x})\,,\ \boldsymbol{x} \in \mathbb{X}$. In the following, random variables are denoted with capital latin letters, whereas associated samples are lowercase. Also, the simpler notation $p(\boldsymbol{x})$ may be used for density functions $f_{\mathbf{x}}(\boldsymbol{x})\,,\ \boldsymbol{x} \in \mathbb{X}$, as is customary in Machine Learning literature.

The data samples $\boldsymbol{x}$ are assumed to be generated from a hidden *underlying process* with an unknown, "true" distribution $p^{\star}(\boldsymbol{x})$. The objective of stochastic modeling is to find a distribution $p(\boldsymbol{x})$ that matches as much as possible the true distribution, among distributions

proposed by a statistical model. *Tuning* a stochastic model is about finding the probability distribution that approximates the true distribution. In the context of machine learning, the model is *learning* when it is able to be tuned directly from the observed data.

Probabilistic models can be used to generate data, therefore, in the field of Machine Learning, they are also commonly called *generative models*. A generative model must be able to generate new data $\widehat{\boldsymbol{x}}$ that is similar to the observed data $\boldsymbol{x}$. Despite not having access to the underlying process of creation of the observed data, it must emulate it. Their goal is not to replicate the true underlying process behind the data, but to find a generative processes that can *explain* the data. This means that given any set of observed data, there are multiple generative processes that can produce it ("all models are wrong, but some of them are useful"). Selecting one generative model among many that can generate the data, takes into consideration criteria that are outside of the generative models' ability to generate data (see section 6.5).

Let's consider a simple setting to illustrate stochastic modeling, with a coin toss, for which the observations x are either heads ($x = 0$) or tails ($x = 1$). Bernoulli distributions $\mathcal{B}$, with parameter θ are trivially proposed as a statistical model:

$$\mathrm{x} \sim \mathcal{B}(\theta). \tag{6.1}$$

This model is determined by its *likelihood function*, i.e. the joint probability density[1] of observed data x as a function of the parameters θ:

$$f_{\mathrm{x}}(\theta, x) = \theta^x (1-\theta)^{1-x}. \tag{6.2}$$

The parameter θ represents the fairness of the coin, with $\theta = 0.5$ for a perfectly fair coin. The tuning of this model is about finding θ from observations. In the following, the density function $f_{\mathbf{x}}(\boldsymbol{\theta}, \boldsymbol{x})$ of a random variable $\mathbf{x}$, with parameters $\boldsymbol{\theta}$ is also denoted $p_{\boldsymbol{\theta}}(\boldsymbol{x})$.

6.1.1 Maximum likelihood estimation

The maximum likelihood estimation (MLE) maximum likelihood estimation is one of the most used methods to tune probabilistic models from data. Assuming a data-set $\mathcal{D}_{\boldsymbol{x}} = \{\boldsymbol{x}_i, \quad i \in [\![1, N]\!]\}$ with independent and identically distributed (i.i.d.) samples $\boldsymbol{x}$ of a random variable $\mathbf{x}$, the joint likelihood function of the observations can be factorized with the observation likelihood:

$$p_{\boldsymbol{\theta}}(\mathcal{D}_{\boldsymbol{x}}) = p_{\boldsymbol{\theta}}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_N) = \prod_{\boldsymbol{x}_i \in \mathcal{D}_{\boldsymbol{x}}} p_{\boldsymbol{\theta}}(\boldsymbol{x}_i). \tag{6.3}$$

The associated MLE is:

$$\hat{\boldsymbol{\theta}}_{\mathrm{MLE}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p_{\boldsymbol{\theta}}(\mathcal{D}_{\boldsymbol{x}}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \prod_{\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{x}}} p_{\boldsymbol{\theta}}(\boldsymbol{x}). \tag{6.4}$$

The logarithm function is strictly increasing, thus for any function f, maximizing f is equivalent to maximizing $\ln f$, or minimizing $-\ln f$. Consequently, maximizing the log-likelihood of the model, or in the machine learning framework, minimizing the negative log-likelihood (NLL) is often preferred:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}}\ \left(-\ln p_{\boldsymbol{\theta}}(\mathcal{D}_{\boldsymbol{x}})\right) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\ \left(-\sum_{\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{x}}} \ln p_{\boldsymbol{\theta}}(\boldsymbol{x})\right). \tag{6.5}$$

The introduction of the logarithm in the MLE enables to transform the product of the likelihood of data-points into the sum of log-likelihoods, which is much easier to differentiate.

[1] Here a *discrete probability density*, or *probability mass function*.

This is particularly interesting when the chosen statistical model is an exponential distribution family since the sum terms simplify. For instance with Gaussians, each term becomes quadratic.

In the simple coin toss setting, the data-set $\mathcal{D}_x$ contains a set of N tossing results, assumed to be i.i.d.[2]. Assuming a number n of tails, the NLL is simply:

$$\mathcal{L}(\theta) = n \ln \theta + (n-1) \ln (1-\theta) . \tag{6.6}$$

The optimal θ is derived from $\frac{\mathrm{d}\mathcal{L}(\theta)}{\mathrm{d}\theta} = 0$, which yields $\theta = \frac{n}{N}$. The best estimate of the fairness of the coin is simply the ratio of tails over the number of throws. In this experiment, the true value of the sought quantity is accessed with a *sufficiently* large number of repetitions. As will be discussed in 6.2, this approach is referred as frequentism.

6.1.2 Parametric models

A particular class of statistical models are *parametric models*. Parametric models are sets $\mathbb{P}$ of probability distributions such that there exists some subset of a finite dimensional Euclidian space whose vectors $\boldsymbol{\theta}$ (the parameters) index the probability distributions :

$$\exists k \in \mathbb{N}^* \text{ such that (s.t.) } \mathbb{P} = \{P_{\boldsymbol{\theta}},\ \boldsymbol{\theta} \in \Theta \subset \mathbb{R}^k\} \tag{6.7}$$

In other words, a parametric model is a statistical model whose elements, the probabilistic models (distributions), are associated with a finite dimensional vector parameter. Therefore, the *parametric distributions* are uniquely determined by their parameters. When the mapping between distributions and vector parameters is a bijection, the parametric model is *identifiable*. Identifiability is usually a requirement when considering parametric models. In the previous example of coin toss, the fairness of the coin is the vector parameter that indexes the parametric Bernoulli statistical model of the experiment.

A distinction must be made when qualifying models as *parametric*, depending on the context. The Machine learning field shares common objects of interest with adjacent disciplines, and in particular with statistics. Nonetheless, despite sharing common tools and methods, the notions of a parametric model, or even of a parameter, are a little different among communities. In statistics, a parameter is a characteristic of the data. As discussed above, statistical parametric models are characterized by their parameter of interest being of finite dimension. Conversely, in Machine Learning, a model parameter is a configuration variable that is internal to the model, and that must be optimized from the data (e.g. the weights of a neural network). A parametric model is a function parameterized by a finite set of those variables, and that learns to map inputs to outputs. These parameters are fixed when training is over. Non-parametric machine learning models make weaker assumptions about the form of the mapping function. These methods may either not use parameters, they may not be fixed after training, or they may not be constant in number. These methods include k-means, decision trees and support vector machines (see section 3.2). This difference in definition tends to blur, because recent Machine Learning (ML) approaches have thrived in incorporating Bayesian methods, as will be seen in section 6.4. Which parameter is discussed shall either be specified, or inferred from the context.

6.2 Bayesian inference

In the previous coin tossing experiment, θ is a parameter of a model of a single coin. The parameter θ was unknown, but was assumed to have a true value that can be approximated. This setup is commonly described as a *frequentist* approach. In Bayesian statistics [Bernardo and Smith, 2009], the unknown parameter is treated as a random variable $\boldsymbol{\theta}$, and therefore

[2]Here, the samples in $\mathcal{D}_x$ can be described by a *binomial* distribution with parameters N and θ

described with a distribution[3]. This approach is about assuming a *prior distribution* $p(\boldsymbol{\theta})$ over the unknown variable, as a first guess, which is then updated from observations $\mathbf{x}$ into a *posterior distribution* $p(\boldsymbol{\theta}|\mathbf{x})$, as an educated guess. This approach revolves around the *Bayes' theorem*:

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\boldsymbol{x})}. \tag{6.8}$$

$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}$ is the *marginal likelihood* of the data, and $p(\boldsymbol{x}|\boldsymbol{\theta})$ is, like before, the conditional likelihood of the data.

Let's resume our previous coin tossing example with a Bayesian approach. The fairness of the coin is set as a random variable θ, and its the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$ which is estimated. This time, only a single toss of the coin is considered. The experiment is still modeled with a Bernoulli distribution : $p(x|\theta) \sim \mathcal{B}(\theta)$. As a prior $p(\theta)$, we choose a beta distribution $\beta(a, b)$, a continuous distribution over the interval $[0, 1]$. For instance, selecting $a = b$, enables the prior to be symmetrical and centered around $\theta = 0.5$.

Selecting a beta distribution as the prior is actually a "good" choice, because the Bernoulli distribution is a conjugate prior for the Bernoulli distribution [George et al., 1993]. This implies that the posterior distribution is also a beta distribution, thus simplifying the computation. In this case, the marginal likelihood integral can be analytically computed, and the right-hand side of Bayes formula (Equation 6.8) can be simplified. The posterior $p(\theta|x)$ follows a beta distribution whose parameters depend on the observed data x:

$$p(\theta|x) \sim \beta(a + x, b + (1 - x)). \tag{6.9}$$

The posterior distribution was obtained by updating a prior distribution with an observation.

As illustrated by this example, the Bayesian approach has several advantages. It enables an uncertainty quantification for the unknown variable. Contrary to a frequentist approach which assumes a certain number of repetitions of the experiment, Bayesian statistics only consider one instance. By using the computed posterior as the new prior, it is possible to iterate and take more observed data into account[4].

Bayesian statistics are often qualified as "subjective". This is because it involves selecting a prior distribution $p(\boldsymbol{\theta})$, and a stochastic model $p(\boldsymbol{x}|\boldsymbol{\theta})$. One could argue that frequentist statistics are just as subjective because they also assume a model for the data. Nonetheless, the influence of the prior distribution over the posterior decreases the more it is updated with new data.

Besides, Bayesianism and frequentism differ in their conception of probability. The frequentist framework only involves probability as an interpretation of frequencies observed in the long run, when experiments are repeated *sufficiently enough*. An important question is then how much repetition should be carried out for the frequencies to be significant. The probability of an event is an idealized, asymptotic proportion of times in which this even occurs in a large number of repeated observations under the same conditions. For Bayesian approaches, the unknown parameter is given a probability distribution for itself, independently of the observed data. Probability is used to quantify uncertainty, and represents a *degree of belief* regarding the system.

6.2.1 Latent variable models

In Bayesian inference introduced in the previous paragraph, the observations $\mathbf{x}$ are conditioned on a variable $\boldsymbol{\theta}$ which is also a random variable. This variable isn't observed (i.e. not

[3]Technically, $\boldsymbol{\theta}$ is unknown, but with a fixed value, and the associated distribution is a measure of uncertainty about the value of this variable, or *degree of belief*.

[4]By modeling the experiment with a binomial distribution with the size of the data-set as parameter N, the final posterior can be obtained without iterating through the data-set: $p(\theta|x) \sim \beta(a + x, b + (N - x))$, with x the number of tails.

present inside the data-set), and it must be estimated indirectly. Retrieving the value of this unobserved variable requires a likelihood model $p(\boldsymbol{\theta}|\boldsymbol{x})$, a prior $p(\boldsymbol{\theta})$, and some observed samples $\boldsymbol{x}$. For instance, in the coin tossing experiment, the fairness of the coin, is an unobserved parameter that can be different for various coins. This variable is directly used as the parameter of a Bernoulli distribution that conditions the coin toss.

Such a unobserved variable is called a *hidden variable*, or a *latent variable*. It is commonly denoted $\mathbf{z}$ in the *variational inference* context (see subsection 6.3.2), and this notation will be used from now on. A point must be made here about the distinction, or lack thereof, between *latent variables* and *model parameters*, which will denote $\mathbf{z}$ and $\boldsymbol{\theta}$. Theoretically, Bayesianism doesn't actually distinguish unobserved random variables as parameters and latent variables for a stochastic model. Bollen [2002] acknowledges the diversity of definitions for latent variablesand gives a general definition as "variables for which there is no sample realization for at least some observations in a given sample". A classification between hidden random variables occurs depending on the context, on the application, and also often depending on the author. In the following paragraph, an arbitrary distinction between parameters and latent variables is proposed to accommodate the context of this work.

A parameter $\boldsymbol{\theta}$ is a tuning parameter of a parametric function that implements a probabilistic model. This parameter may be optimized, however it is supposed to remain constant for all observations $\boldsymbol{x}$, it is a *global* variable. This parameter may be either considered deterministic or random, depending on the needs. In the former case, a parameter can be directly attributed to the machine learning definition of a model parameter (see subsection 6.1.2), and its retrieval is referred as *calibration* rather than *inversion* (see subsection 3.1.1). A latent variable $\mathbf{z}$ is a random variable that is not optimized, and represents an intrinsic property of the observations which they are conditioned on. Its key feature, is that it is a *local* variable, i.e. each observation $\boldsymbol{x}_i$ is associated with a different latent variable $\mathbf{z}_i$. Furthermore, latent variables represent properties of observations, expressed through a formal model. However, latent variables may not be directly identifiable to specific properties, they may just correspond to an arbitrary feature that explains the variability in observed data. This aspect will be further discussed in section 6.5.

Recalling the previous coin example, the model of a toss is a Bernoulli distribution, and the fairness is an unknown distribution parameter. The coin fairness may be computed as a model parameter, or a latent variable depending on the context. If there is a single coin from which all observations are made, then the coin fairness may be better seen as a model parameter (in a frequentist point of view) or as a global latent variable (in a Bayesian point of view). Once an estimate is found for the coin fairness, it can be used to model all observations. However, if we consider the possibility of multiple coins, the fairness is not necessarily identical among the coins. It must be inferred for all coins separately, and is best described as a local latent variable.

A stochastic model can both have model parameters and latent variables. Suppose a slightly more complex model of the coin tossing experiment is made. The coin fairness is still used to model the probability of heads and tails as a Bernoulli distribution. However the variable of interest, which will be used as a latent variable, will be an intrinsic property of the coins. For instance, the coins can be assumed imperfect thin and flat cylinders, with imperfections, they can be bent, or their alloy density is not uniform throughout the coin. We define as latent variable $\mathbf{z}$, some sort of deviation of the mass distribution in the coin from an ideal cylindrical coin. The fairness of the coin will be influenced by this hidden physical property. We then assume that there is a parametric *generative function* (deterministic or not) $f_{\boldsymbol{\theta}}(\boldsymbol{z})$ that links the fairness of a coin with its physical property. The deterministic parameter $\boldsymbol{\theta}$ is an intrinsic property of the model but external factors can be considered as well, such as parameters pertaining to the experiment condition e.g. the local gravity vector, atmospheric conditions, magnetic fields, etc. In this case, the stochastic model likelihood is:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{\theta}) \sim \mathcal{B}(f_{\boldsymbol{\theta}}(\boldsymbol{z})). \tag{6.10}$$

The parametric model $f_{\boldsymbol{\theta}}$ may be arbitrarily complex. In the following of this work in particular, the trainable weights of neural networks will be designated by $\boldsymbol{\theta}$.

A probabilistic model that implements the relationship between latent variables $\mathbf{z}$ and observed data $\mathbf{x}$ is not only the conditional distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$, but the joint distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$. $\boldsymbol{\theta}$ is tuned so that $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$ of the model approaches the true distribution $p(\boldsymbol{x}, \boldsymbol{z})$.

In most cases, $\mathbf{x}$ and $\mathbf{z}$ are not attributed a symmetrical role and meaning. For instance, $\mathbf{x}$ is assumed to have a higher dimension and/or a higher complexity than $\mathbf{z}$.

It is depending on the application, that a *conditional* likelihood between $\mathbf{z}$ and $\mathbf{x}$ is preferred over the other:

- In the context of *data generation*, the conditional distribution $p(\boldsymbol{x}|\boldsymbol{z})$ is used. In this case, the conditional likelihood is a *generative process* and $\mathbf{z}$ is a *generative factor* of $\mathbf{x}$. A parametric *generative function* $f_{\boldsymbol{\theta}}(\boldsymbol{z})$ may govern the conditional distribution of observations $\mathbf{x}$: $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta}) = p_{\boldsymbol{\theta}}(\boldsymbol{x}|f_{\boldsymbol{\theta}}(\boldsymbol{z}))$. $\boldsymbol{\theta}$ are then global *generative* parameters.

- In classification or regression, it is rather $p(\boldsymbol{z}|\boldsymbol{x})$ that is used. In this case, the conditional likelihood is an *inference model* (or *recognition model*), with $\mathbf{z}$ being a *feature*, or a *representation* of $\mathbf{x}$. An *inference function* $f_{\boldsymbol{\phi}}(\boldsymbol{x})$ may govern the conditional distribution of latent variables $\mathbf{z}$: $p_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) = p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\phi}) = p\left(\boldsymbol{z}|f_{\boldsymbol{\phi}}(\boldsymbol{x})\right)$, with ϕ a dedicated global parameter vector.

This leads to introducing a cause-effect relationship, between $\mathbf{x}$ and $\mathbf{z}$. This causality relationship is technically not due to any precedence of $\mathbf{z}$ or $\mathbf{x}$, but rather because it leads to consider $\mathbf{z}$ and $\mathbf{x}$ as the input or the output of the model. Nonetheless, in practice $\mathbf{z}$ is considered a *cause* to $\mathbf{x}$.

In the context of *Bayesian* statistics *inference* is commonly defined as the process of prediction of the cause of a phenomena from the effects. Hence the name of *inference model* for $p(\boldsymbol{z}|\boldsymbol{x})$. Estimation of a latent variable from observations is called *inference* because a latent variable can be thought as the hidden cause that generates some observed data $\mathbf{x}$. In the previous coin tossing example, the fairness of the coin is *inferred* from heads and tails observations.

A stochastic model $p(\boldsymbol{z}, \boldsymbol{x})$ is also called a *generative model*, especially in the Machine Learning field. This is because it enables to generate new data $\mathbf{x}$ by using $p(\boldsymbol{x}|\boldsymbol{z})$, along with a prior $p(\boldsymbol{z})$. The distribution of latent variables is more simply called the *latent distribution*. A realization of this distribution is called a *latent vector*, that belongs to a finite dimensional *latent space*.

6.2.2 Point estimation

When performing statistical estimation, a common goal is to approximate a quantity of interest. Selecting a single value in the parameter space is performing *point estimation*, or choosing a *point estimate*, as a "best estimate", a plausible value of the unknown quantity [Lehmann and Casella, 2006]. Frequentist approaches naturally produce such point estimates, for they are usually concerned with the expected value of the distribution of the parameter of interest.

In the case of Bayesian approaches, the parameter of interest is explicitly modeled as a probability distribution, and the estimated quantity is not a point estimate but the distribution itself. For Bayesian inference, performing point estimation is therefore about choosing a particular value of the computed posterior distribution, and is by essence arbitrary. The most common point estimates of distributions are:

- the *expected value*, as the average, or barycenter of the distribution,

- the *median*, as the "middle" value between the lower and higher half of the distribution,

- the maximum a posteriori (MAP), as the most likely or most frequent value of the posterior distribution.

Once point estimates have been computed, they can be used for evaluating different estimation metrics (see subsection 3.1.5).

The computational context matters a lot when selecting a point estimate. In particular, the dimensionality of the posterior distribution can hamper the computation of some estimates. For multi-dimensional distributions, the expected vector value is the vector of the expected values. However, when the vector components are not independent, the mode (resp. the median) of the distribution cannot be computed as the vector of the mode (resp. the median) of the marginal distributions. In fact, there is no exact algorithm for computing the median of a distribution in dimension greater than 1, it is an optimization problem[5] [Lin and Vitter, 1992]. For sampling methods (see subsection 6.3.1), the estimation of the mode of the posterior distribution is very sensitive to the number of samples, which makes it unreliable for high dimensional posteriors. Finally, the notion of a point estimate falls apart for multi-modal distributions with similarly high likelihood maxima. With such distributions, there is no single best representation of the data.

6.2.3 Measuring uncertainty

One of the main advantages of using statistical models, either in a frequentist or in a Bayesian context, is that uncertainty is taken into account in data analysis. Overall, uncertainty is accounted for with the use of intervals, that are given a certain probability of containing the "true value" of some estimated quantity. However, depending on the context, these intervals are different [Altman et al., 2013].

6.2.3.1 Confidence intervals

Frequentist approaches typically assume that the quantity of interest θ of some data x is unknown, but not random. Given data samples $\boldsymbol{x}_n = (x_i)_{i\in[\![1,n]\!]}$, confidence intervals are commonly defined as intervals $\mathbb{I}(\boldsymbol{x}_n) = [l(\boldsymbol{x}_n), u(\boldsymbol{x}_n)]$, such that the probability of the quantity of interest being in the interval is given by a *confidence level* α:

$$P(\theta \in \mathbb{I}(\boldsymbol{x}_n)) \geq 1 - \alpha \tag{6.11}$$

It can be noted that it is confidence intervals themselves that are random in nature, as they depend on random data samples.

The true definition of confidence intervals is actually a little more subtle. Let there be a procedure f that maps data samples $\boldsymbol{x}_n$ to subsets $f(\boldsymbol{x}_n)$ of the parameter space Θ. The probability $P(\theta \in f(\boldsymbol{x}_n))$ is always defined. When $\forall \boldsymbol{x}_n, \quad P(\theta \in f(\boldsymbol{x}_n)) \geq 1-\alpha$, then the subsets $f(\boldsymbol{x}_n)$ are confidence intervals with confidence level $1-\alpha$. The higher the confidence level, the larger the confidence interval. The definition of confidence intervals is not about one given interval, but about the procedure that generates them. Considering a data-set, from which multiple confidence intervals are computed from data sub data-sets, the long-term frequency of a given confidence interval containing θ is defined by the confidence level. However, in practice, only a single confidence interval is ever computed from data, which commonly leads to shift the definition.

The nuance can be further appreciated by considering the probability of the true parameter belonging to confidence intervals. The actual probability of a given confidence interval $f(\boldsymbol{x}_n)$

[5] The *geometric median* generalizes the notion of median to higher dimension, as the point which minimizes Euclidean distance within the distribution: for samples $(\boldsymbol{x}_i)_{i\in[\![1,m]\!]}$ s.t. $\forall i \quad \boldsymbol{x}_i \in \mathbb{R}^n$, the geometrical median is $\underset{\boldsymbol{x}\in\mathbb{R}^n}{\arg\min} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{x}\|_2$.

containing θ, is the interval's *coverage probability*. The confidence level however, is the lower bound of the coverage probability:

$$1 - \alpha = \inf_{\theta \in \Theta, f(\boldsymbol{x}_n) \subset \Theta} P\left(\theta \in f(\boldsymbol{x}_n)\right) \tag{6.12}$$

Confidence intervals, defined above in the one-dimensional case, are extended into *confidence regions* in the multi-dimensional case.

6.2.3.2 Credible intervals

Credible intervals are the Bayesian counterpart of confidence intervals. In Bayesian statistics, the unknown parameter θ is associated with a probability distribution. A *credible interval* is simply a subset of the parameter space, within which samples of the random parameter θ fall with a certain probability, derived from the distribution of said parameter. For instance, if the distribution of θ is such that $P(\theta \in [a, b]) = 1 - \alpha$, then $[a, b]$ is a $1 - \alpha$ credible interval for θ. Contrary to Frequentism, which considers the unknown parameter as fixed, and the confidence interval as random, Bayesianism defines the unknown parameter as random, and selects a fixed credible interval. It can be noted that the distribution used to construct credible intervals isn't necessarily the posterior distribution $p\left(\theta|x\right)$, but it can be the prior $p\left(\theta\right)$ as well. Credible intervals are non unique, and a selection is inherently arbitrary. Common intervals include:

- Mean-centered intervals, for which the distribution expected value is at the middle of the interval.
- The *equal-tailed intervals*, which ensures identical probability of being above and below the interval. This interval notably includes the median.
- The (HDI), often specified as highest density posterior (or prior) interval, for which the included values have the maximum possible probablity. This interval is the narrowest possible, and contains the maximum likelihood. Variations of the HDI which allow disjoint intervals as subsets that suit well multi-modal distributions, by providing a finite set of intervals of high probability density around each mode.

6.2.3.3 Prediction intervals

Finally, another type of interval that quantifies uncertainty about an unknown quantity are *prediction intervals*. Prediction intervals broadly refer to any interval assumed to contain a yet unobserved variable of interest[6], with a certain probability the prediction interval nominal coverage (PINC) (the analog of the confidence level of confidence intervals) [Landon and Singpurwalla, 2008]. The distinction between prediction intervals and the above-defined intervals is rather blurry. A prediction interval can be considered a confidence interval or a credible interval depending on the context. In the Bayesian context of this work, prediction intervals are taken as credible intervals of posterior distributions. Prediction intervals are at the core of uncertainty quantification [Abdar et al., 2021; Edupuganti et al., 2021].

Like other intervals, the width of a prediction interval depends on the choice of a *coverage probability*. This choice largely depends on the considered application, however it can be noted that imposing high coverage probability (e.g. 95%) usually leads to impractically large, "embarrassingly wide" prediction intervals [Granger, 1996]. Landon and Singpurwalla [2008] argues that the choice of this probability should be motivated by the application context, with the actual use of the prediction interval, in the fashion of an optimization problem. The

[6] In many domains concerned with forecasting, a variable to be observed is often referred as a *future value*. However this terminology only references the type of data that are used, by no mean this variables needs to be produced in the future, it just means that it hasn't yet been taken into account by the model.

size of prediction intervals can be assessed with the mean prediction interval width (MPIW), which is simply the average width of prediction intervals $[a_i, b_i]$:

$$\text{MPIW} = \frac{1}{N}\sum_{i=1}^{N} b_i - a_i. \tag{6.13}$$

The ability of prediction intervals to correctly contain the yet-unknown variable of interest can be assessed after it has been observed. However, this can't be done for a single interval, because then the probability of the interval containing the true value is either 0 or 1, it can only be inside or outside. This is the prediction interval *coverage* $c(\theta)$. However, with multiple prediction intervals related to multiple values of interest, a prediction interval coverage probability (PICP) can be computed as long term coverage frequency with the mean of the prediction interval coverage [Ak et al., 2015]:

$$\text{PICP} = \frac{1}{N}\sum_{i=1}^{N} c_i(\theta_i). \tag{6.14}$$

Ideally, the PICP should be as close as possible to the PINC [Zheng et al., 2022].

6.3 Approximate inference

Let there be a probabilistic model that has observations $\mathbf{x}$ and latent variables $\mathbf{z}$, with its joint density $p(\boldsymbol{x}, \boldsymbol{z})$. Given the observations $\mathbf{x}$, the objective is to perform the inference of $\mathbf{z}$, i.e. to compute $p(\boldsymbol{z}|\boldsymbol{x})$ as the posterior distribution. To do that, the generative process $p(\boldsymbol{x}|\boldsymbol{z})$ is chosen along with a prior distribution on latent variables $p(\boldsymbol{z})$. As introduced in section 6.2, the Bayes theorem (equation (6.8)) states that :

$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p(\boldsymbol{x})}. \tag{6.15}$$

The right hand side denominator $p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z})\,\mathrm{d}\boldsymbol{z}$ is the *marginal likelihood* of the observations, also called *model evidence*, or simply *evidence*. When the marginal likelihood can be computed by marginalizing the stochastic model over latent variables, the posterior distribution can be directly derived from the Bayes formula. In section 6.2 was shown an example of a coin tossing experiment for which the setting allowed to perform exact inference of the posterior distribution.

However in most cases the evidence integral is intractable, mainly in cases of high dimensionality of the latent variable. To retrieve the posterior $p(\boldsymbol{z}|\boldsymbol{x})$ despite $p(\boldsymbol{x})$ being inaccessible, there are two main approaches:

1. The sampling approach: the exact posterior distribution is sampled by using a Markov Chain Monte Carlo (MCMC) algorithm, enabling to estimate various statistics (mean, median, variance, etc...). These methods are usually very accurate, but very computationally expensive, and hardly scale up for large inference problems.

2. The *variational inference* approach [Jaakkola, 1997; Jordan et al., 1998; Saul and Jordan, 1995]: the posterior distribution is approximated by selecting the closest distribution among a chosen distribution family. While less accurate than sampling methods, it is computationally faster, and as detailed in section 6.4, it has been recently used in conjunction with deep learning for a huge performance gain.

6.3.1 Markov Chain Monte-Carlo

To compute the posterior distribution, MCMC sampling methods take a different approach than variational inference (see subsection 6.3.2). MCMC sampling consists in building a

Markov chain over the latent variables, and define its stationary distribution as the posterior distribution [Geman and Geman, 1984; Hastings, 1970; Metropolis et al., 1957]. Once the algorithm reaches equilibrium, samples are drawn from the Markov chain and are used to derive statistics on the posterior distribution. Although very precise, this method has very slow convergence and cannot scale up for large inference problems.

6.3.2 Variational inference

The variational inference methods approximate the true posterior $p(\boldsymbol{z}|\boldsymbol{x})$ with a surrogate distribution $q(\boldsymbol{z}|\boldsymbol{x})$, from a chosen distribution family $\mathcal{Q}$. This surrogate distribution to the true posterior distribution is called the *variational distribution*. Most applications use *fixed-form*, or *structured* variational inference, for which the substitute distribution is a $\boldsymbol{\lambda}$-parameterized distribution $q_{\boldsymbol{\lambda}}$, belonging to $\mathcal{Q}_{\boldsymbol{\lambda}}$. In such case, $\boldsymbol{\lambda}$ are the *variational parameters*. In the remainder of this work, we assume the variational inference to be fixed-form. For instance, if the distribution family is chosen as the beta distributions, then $\boldsymbol{\lambda} = (a, b)$, and $q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x}) \sim \beta(\boldsymbol{\lambda})$.

The variational parameters $\boldsymbol{\lambda}$ are local parameters (see subsection 6.2.1), meaning that a variational parameter vector $\boldsymbol{\lambda}_i$ is associated to each latent variable $\boldsymbol{z}_i$.

A distinction must be made between the terms *variational distribution*, *latent distribution*, and *posterior distribution*. The latent distribution is simply the distribution of latent variables, in a general context. When performing Bayesian inference, the distribution to derive is the posterior distribution. Here, because it is the latent variables that are retrieved, the posterior distribution is the latent distribution. The variational distribution, that is specific to variational inference, is an approximation of the posterior that must be retrieved. In the variational inference context, these terms all refer to the same distribution of latent variables, and in this work, we refer to them interchangeably.

The term *variational Bayes* is often used interchangeably with *variational inference*. Variational inference is the broad approach of using optimization to find surrogate distributions that match with an unknown distribution of interest. Variational Bayes is a variational inference applied in the context of Bayesian inference of a posterior distribution. Non-Bayesian variational inference approximate distributions without relying on a prior distribution Choi and Rim [2023]. Such methods are rare, variational inference is almost always used to infer a posterior distribution in a Bayesian setting, and is assimilated with variational Bayes.

6.3.3 Evidence lower bound

The variational distribution $q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})$ must be optimized (i.e. $\boldsymbol{\lambda}$ must be learned), so that it is the best approximation of the true posterior among the variational distribution family $\mathcal{Q}_{\boldsymbol{\lambda}}$. To do that, the Kullback-Leibler divergence (KLD) (also called *relative entropy*, *relative information content*, or *I-divergence*) is typically used to quantify the similarity between the variational distribution and the true posterior, and optimized:

$$q^*_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x}) = \underset{q_{\boldsymbol{\lambda}} \in \mathcal{Q}_{\boldsymbol{\lambda}}}{\arg\min}\, \mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z}|\boldsymbol{x})\right], \tag{6.16}$$

with $q^*_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})$ the optimized variational distribution. However, this KLD cannot be directly optimized, because it is still a function of the intractable evidence $p(\boldsymbol{x})$:

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z}|\boldsymbol{x})\right] &= \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}\left[\ln q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})\right] - \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}\left[\ln p(\boldsymbol{z}|\boldsymbol{x})\right] \\
&= \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}\left[\ln q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})\right] - \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}\left[\ln \frac{p(\boldsymbol{x},\boldsymbol{z})}{p(\boldsymbol{x})}\right] \\
&= \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}\left[\ln q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})\right] - \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}\left[\ln p(\boldsymbol{x},\boldsymbol{z})\right] + \ln p(\boldsymbol{x}).
\end{aligned} \tag{6.17}$$

Rearranging equation (6.17) yields:

$$\ln p(\boldsymbol{x}) = \mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}|\boldsymbol{x})\right] + \underbrace{\mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}\left[\ln p(\boldsymbol{x},\boldsymbol{z}) - \ln q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})\right]}_{\mathcal{L}_{\mathrm{ELBO}}} \tag{6.18}$$

The D_{KL} is always positive, therefore the term

$$\mathcal{L}_{\mathrm{ELBO}}\left(\boldsymbol{\lambda},\boldsymbol{x}\right) = \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}\left[\ln p(\boldsymbol{x},\boldsymbol{z}) - \ln q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})\right] \tag{6.19}$$

is a lower bound on $\ln p\left(\boldsymbol{x}\right)$. This term is called the evidence lower bound (ELBO)[7]. The evidence is not a function of $\boldsymbol{\lambda}$, therefore minimizing $\mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}|\boldsymbol{x})\right]$ with respect to (w.r.t.) $\boldsymbol{\lambda}$ is equivalent to maximizing the ELBO[8]. Rearranging the ELBO yields:

$$\begin{aligned}
\mathcal{L}_{\mathrm{ELBO}}\left(\boldsymbol{\lambda},\boldsymbol{x}\right) &= \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}\left[\ln p(\boldsymbol{x},\boldsymbol{z}) - \ln q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})\right] \\
&= \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}\left[\ln p(\boldsymbol{x}|\boldsymbol{z}) + \ln p(\boldsymbol{z}) - \ln q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})\right] \\
&= \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}\left[\ln p(\boldsymbol{x}|\boldsymbol{z})\right] - \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}\left[\ln\left(\frac{q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})}\right)\right] \\
&= \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}\left[\ln p(\boldsymbol{x}|\boldsymbol{z})\right] - \mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})\right].
\end{aligned} \tag{6.20}$$

The ELBO is made of two terms with competing effects:

1. $\mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}\left[\ln p(\boldsymbol{x}|\boldsymbol{z})\right]$: the expected log-likelihood of the observed data conditioned with a latent vector that was drawn from the variational distribution. This term quantifies how likely an observed data is generated from the stochastic model from a latent vector.

2. $\mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})\right]$: this term quantifies how far the variational distribution is from the prior on latent variables (which we will also call the *latent prior*).

For a data-set $\mathcal{D}_{\boldsymbol{x}}$ with N data-points, the variational objective function is simply the mean (or alternatively just the sum) of the ELBO over each data-point:

$$\mathcal{L}\left(\boldsymbol{\lambda}_{1:N},\boldsymbol{x}_{1:N}\right) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{\mathrm{ELBO}}\left(\boldsymbol{\lambda}_i,\boldsymbol{x}_i\right). \tag{6.21}$$

6.3.4 ELBO optimization

The denomination of variational inference as "variational" is taken from the mathematical field of *variational calculus*. Variational calculus is about finding the extrema of functionals by using variations (e.g. derivatives). Indeed, the core principle of variational inference methods is to transform the intractable integration problem in the posterior estimation to an optimization problem, with the ELBO.

In many variational inference applications, such as the ones presented in this work, the variational distribution family is a *fully factorized family* (also called *mean-field family* [Neal and Hinton, 1998]), for which each variable in the latent vector is independent:

$$\mathcal{Q}_{\boldsymbol{\lambda}} = \left\{q_{\boldsymbol{\lambda}} \text{ s.t. } q_{\boldsymbol{\lambda}}\left(\boldsymbol{z}\right) = \prod_{n=1}^{N} q_{\boldsymbol{\lambda}_n}\left(\boldsymbol{z}_n\right)\right\}. \tag{6.22}$$

This is because, as shown in equation (6.20), the first term of the ELBO is an expectation w.r.t. the variational distribution. Assuming factorized and independent factors for the variational distribution enables to optimize the ELBO for each of these factors. Methods that

[7] The ELBO is commonly also derived from $\ln p(\boldsymbol{x})$ with Jensen's inequality, which states that for any convex function f and random variable X, $f\left(\mathrm{E}\left[X\right]\right) \leq \mathrm{E}\left[f\left(X\right)\right]$. Furthermore, the difference $\mathrm{E}\left[f\left(X\right)\right] - f\left(\mathrm{E}\left[X\right]\right)$ is called the *Jensen gap*.

[8] $\mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}|\boldsymbol{x})\right]$ is the Jensen gap between $\ln p(\boldsymbol{x})$ and the ELBO.

use a mean-field variational distribution family are called *factorized* or *mean-field* variational inference.

Traditionally, the optimization is performed with the coordinate ascent variational inference (CAVI) algorithm [Blei et al., 2017], which uses every available data sample for each posterior update iteration. This approach is inefficient for large data-sets, as it requires a full pass of the data-set at each iteration, while computing an exact gradient of the objective. To solve this issue *stochastic optimization* [Robbins and Monro, 1951] was combined with variational inference in Hoffman et al. [2013] with stochastic variational inference (SVI). The data is sub-sampled into mini-batches with size $B \geq 1$, for which an estimator $\widehat{\mathcal{L}}$ of the ELBO objective function over the whole data-set $\mathcal{L}$ is computed:

$$\widehat{\mathcal{L}}\left(\boldsymbol{\lambda}_{1:B}, \boldsymbol{x}_{1:B}\right) = \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_{\text{ELBO}}\left(\boldsymbol{\lambda}_i, \boldsymbol{x}_i\right) \xrightarrow[B \to N]{} \mathcal{L}\left(\boldsymbol{\lambda}_{1:N}, \boldsymbol{x}_{1:N}\right). \tag{6.23}$$

Then, a noisy mini-batch gradient of the objective function w.r.t. the variational parameters $\nabla_{\boldsymbol{\lambda}}\widehat{\mathcal{L}}\left(\boldsymbol{\lambda}_{1:B}, \boldsymbol{x}_{1:B}\right)$ can be computed from the ELBO objective function estimate. The classical update of the variational parameters is performed from this noisy gradient :

$$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \rho \nabla_{\boldsymbol{\lambda}}\widehat{\mathcal{L}}\left(\boldsymbol{\lambda}_{1:B}, \boldsymbol{x}_{1:B}\right), \tag{6.24}$$

with ρ the step size.

When the data is sampled independently, the expectation of the noisy mini-batch gradient $\nabla_{\boldsymbol{\lambda}}\widehat{\mathcal{L}}$ equals the true gradient $\nabla_{\boldsymbol{\lambda}}\mathcal{L}$ over the whole data-set, enabling a faster optimization. However, a high variance of the noisy gradient can prevent optimization by making the updates too unstable in the variational parameter space. This is why SVI approaches pay special attention to reducing this stochastic gradient variance. In the original SVI paper [Hoffman et al., 2013], this issue was mitigated by using natural gradients[9], instead of classical gradients.

Nonetheless, a restriction on SVI is that it requires an analytical derivation of the ELBO to estimate the gradients of individual data-points $\nabla_{\boldsymbol{\lambda}_i}\mathcal{L}_{\text{ELBO}}\left(\boldsymbol{\lambda}_i, \boldsymbol{x}_i\right)$. Specifically, differentiating the ELBO (Equation 6.20) yields:

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}_i}\mathcal{L}_{\text{ELBO}}\left(\boldsymbol{\lambda}_i, \boldsymbol{x}_i\right) &= \nabla_{\boldsymbol{\lambda}_i}\mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)}\left[\ln p(\boldsymbol{x}_i|\boldsymbol{z}_i)\right] - \nabla_{\boldsymbol{\lambda}_i}\mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)\|p(\boldsymbol{z}_i)\right] \\ &= \nabla_{\boldsymbol{\lambda}_i}\mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)}\left[f\left(\boldsymbol{\lambda}_i, \boldsymbol{x}_i, \boldsymbol{z}_i\right)\right] - \nabla_{\boldsymbol{\lambda}_i}\mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)\|p(\boldsymbol{z})\right].\end{aligned} \tag{6.25}$$

The expectation $\mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)}\left[f\left(\boldsymbol{x}_i, \boldsymbol{z}_i\right)\right]$, with $f\left(\boldsymbol{x}_i, \boldsymbol{z}_i\right) = \ln p(\boldsymbol{x}_i|\boldsymbol{z}_i)$, on the right-hand side of Equation 6.25 is intractable unless the likelihood $p(\boldsymbol{x}_i|\boldsymbol{z}_i)$ is an exponential distribution w.r.t. $\boldsymbol{z}_i$, which limits use cases to very simple models. To generalize SVI to other more complex statistical models, a workaround is to compute a Monte Carlo (MC) estimate of the term $\nabla_{\boldsymbol{\lambda}_i}\mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)}\left[f\left(\boldsymbol{x}_i, \boldsymbol{z}_i\right)\right]$ as a tractable substitute. [Paisley et al., 2012]:

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}_i}\mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)}\left[f\left(\boldsymbol{x}_i, \boldsymbol{z}_i\right)\right] &= \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)}\left[f\left(\boldsymbol{x}_i, \boldsymbol{z}_i\right)\nabla_{\boldsymbol{\lambda}_i}\ln q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)\right] \\ &\approx \frac{1}{L}\sum_{l=1}^{L} f\left(\boldsymbol{x}_i, \boldsymbol{z}_i^{(l)}\right)\nabla_{\boldsymbol{\lambda}_i}\ln q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i^{(l)}|\boldsymbol{x}_i)\end{aligned} \tag{6.26}$$

MC estimation of the gradient is performed by drawing L samples $\boldsymbol{z}_i^{(l)}$ of the latent (variational) distribution $q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i^{(l)}|\boldsymbol{x}_i)$. The proof for the permutation of the gradient and expectation

[9]The natural gradient is a generalization of standard gradient that accounts for the curvature of the optimized function. This is especially useful when the parameter space is not well characterized by a Euclidean distance, e.g. in this case the dissimilarity between two distributions is not well measured by an l^2 distance but rather with a KLD. Using natural gradient is instrumental to stochastic variational inference because it reduces the variance of the gradient and improves optimization.

operators in Equation 6.26 that enables the MC approximation are provided in section D.2. Unfortunately, this gradient estimate is noisy with high variance, and cannot usually be used as is. This is why additional techniques are required to lower this gradient, among which is the *reparameterization trick* discussed in subsection 6.4.2. Finally, it could be argued that the computation of a gradient estimate such as shown in Equation 6.26 is a convoluted solution. Instead of approximating $\mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)}\left[f\left(\boldsymbol{x}_i,\boldsymbol{z}_i\right)\nabla_{\boldsymbol{\lambda}_i}\ln q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)\right]$ with MC, why not approximating $\nabla_{\boldsymbol{\lambda}_i}\mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)}\left[f\left(\boldsymbol{x}_i,\boldsymbol{z}_i\right)\right]$ directly? The reasons will also be developed in subsection 6.4.2.

6.4 Variational autoencoders

As discussed in section 6.3, variational inference is an interesting approach to retrieve probabilistic representations of data $\boldsymbol{x}$, in the form of latent variables $\boldsymbol{z}$. The core principle of variational inference is to transform an intractable integral computation into an optimization problem, based on the ELBO. However this approach traditionally struggles with both large data-sets and complex statistical models. The former issue is related to a high computational cost due to the need of performing optimization at each data point, whereas the latter is due to ELBO gradients intractability. A novel approach to variational inference that attempts to solve both of these issues was presented in the seminal work Kingma and Welling [2014], during Kingma's Ph.D. [Kingma, 2017]. This has led to the development of the VAE framework, which has seen a large number of contributions and applications.

The two improvements over SVI that brought about VAE are discussed respectively in subsection 6.4.1 and subsection 6.4.2. The derivation and interpretation of VAE is then explained in subsection 6.4.2.

6.4.1 Amortized variational inference

Although faster than sampling methods, variational inference struggles when tasked with large data-set inference. This is because optimization of variational parameters $\boldsymbol{\lambda}_i$ is performed for each new data sample $\boldsymbol{x}_i$ independently. The optimization is *memoryless*, and inference using one observation cannot interfere with others. This means that variational inference cannot take advantage of similarity between data samples to improve or accelerate the posterior computation. It is also the main reason why this method cannot be well parallelized either for large scale problems: the computational cost grows with the data-set.

To solve this issue, the optimization of variational parameters is *amortized*, i.e. the optimization is spread across multiple data instead of being performed independently [Gershman and Goodman, 2014]. Specifically, amortized variational inference uses a parametric *inference function* $\boldsymbol{g}_{\boldsymbol{\phi}}$ that maps observed data points $\boldsymbol{x}_i$ to their variational parameters $\boldsymbol{\lambda}_i$ (or alternatively, a parametric stochastic function that directly maps observations to latent variables $\boldsymbol{z}_i$). It is this mapping function itself that is optimized instead of the variational parameters. Amortization shifts the optimization to global parameters $\boldsymbol{\phi}$ instead of local variational parameters $\boldsymbol{\lambda_i}$.

This approach makes inference on unseen data straightforward, with simply a function evaluation, without needing to perform the optimization process on the whole data-set. Amortized variational inference becomes especially flexible and powerful when the parametric mapping function $\boldsymbol{g}_{\boldsymbol{\phi}}$ is chosen as a neural network [Kingma and Welling, 2014]. In such case, the optimized global parameters $\boldsymbol{\phi}$ are the network's coefficients. With recent deep learning advances, neural network techniques enabled to learn complex relationships of high-dimensional data, which makes them a suitable choice for the inference function. Furthermore, the neural network framework takes advantage of a wide array of efficient optimization techniques. In particular, graphical processing unit (GPU) hardware enables fast learning with large amounts of data. Combining deep learning with amortized variational inference

therefore removes the scalability issues of stochastic variational inference.

Figure 6.1 summarizes the categories of variational and Bayesian inference discussed until now.

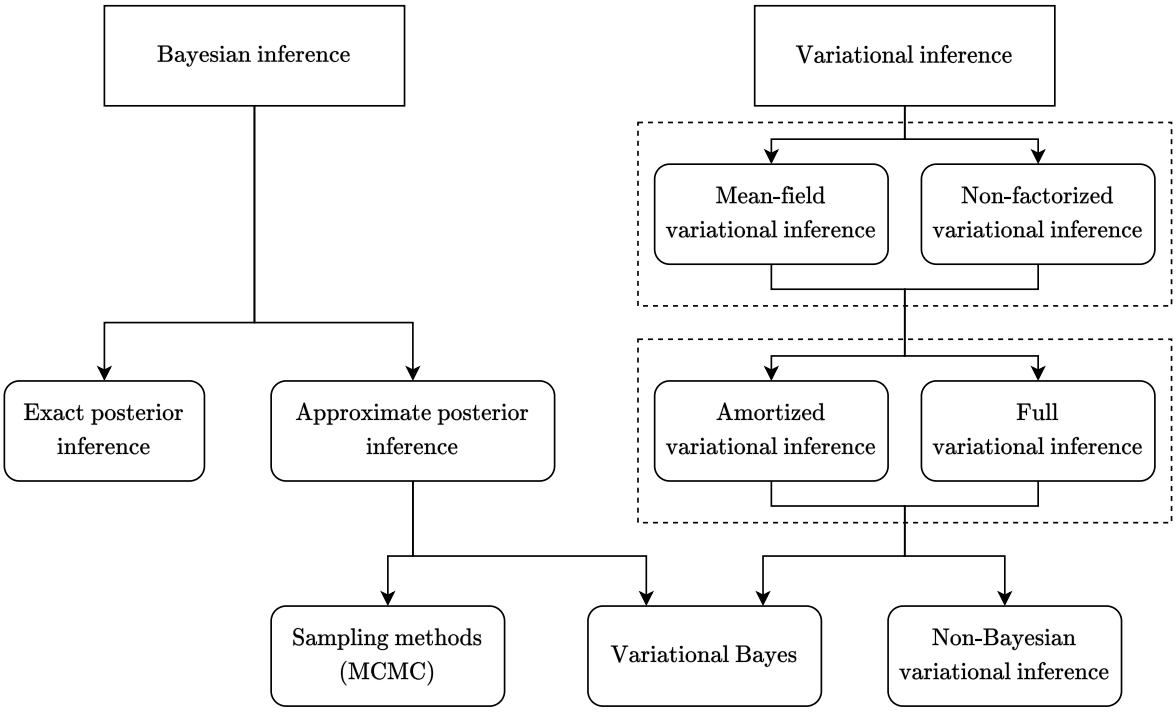


Figure 6.1: Tree of Bayesian and variational inference methods.

6.4.2 Reparameterization trick

As explained in subsection 6.3.4, optimizing the variational parameters requires computing the gradients of the ELBO w.r.t. the variational parameters, for each data-point $\boldsymbol{x}_i$. However in the general case, the term $\nabla_{\boldsymbol{\lambda}_i} \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)} \left[\ln p(\boldsymbol{x}_i|\boldsymbol{z}_i)\right]$ isn't tractable. A solution is to approximate it with MC sampling of the latent variables. However, the simple MC approximation shown in Equation 6.27 cannot be used in the general case.

$$\nabla_{\boldsymbol{\lambda}_i} \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)} \left[f(\boldsymbol{x}_i, \boldsymbol{z}_i)\right] \approx \nabla_{\boldsymbol{\lambda}_i} \frac{1}{L} \sum_{l=1}^{L} f\left(\boldsymbol{x}_i, \boldsymbol{z}_i^{(l)}\right) \tag{6.27}$$

This is because the sampling procedure for latent samples $\boldsymbol{z}_i^{(l)}$ from the variational distribution $q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)$ is usually not differentiable w.r.t. the variational parameters $\boldsymbol{\lambda}_i$. For instance, *rejection sampling* methods are used to design accurate sampling algorithms for most distributions, but are typically not differentiable[10]. An example of one such algorithm, is the Marsaglia and Tsang's algorithm [Marsaglia and Tsang, 2000] for generating gamma distributions samples (which are also used in turn to sample Beta distributions).

To solve this problem, Kingma and Welling [2014] propose to use the *reparameterize* the latent variable, i.e. to perform sampling with a differentiable transformation $\boldsymbol{h}$ of an auxiliary noise variable $\boldsymbol{\epsilon}$, parameterized by the variational parameters:

$$\mathbf{z} = \boldsymbol{h}\left(\boldsymbol{\epsilon}, \boldsymbol{\lambda}\right) = \boldsymbol{h}_{\boldsymbol{\lambda}}\left(\boldsymbol{\epsilon}\right). \tag{6.28}$$

[10] To draw a sample x from a target distribution $\mathcal{X}$, rejection sampling methods draw a sample y from an auxiliary distribution $\mathcal{Y}$. A sample $u \sim \mathcal{U}(0,1)$ is drawn, and y is accepted as the final sample x, following a criterion $f(u)$. Crucially, the criterion depends on the density of the target and auxiliary distribution, and thus on their parameters. However, the sample y accepted into x is not a function of the density parameters of $\mathcal{X}$, and therefore cannot be differentiated w.r.t. them.

This is called the *reparameterization trick*, and it enables to calculate an approximate gradient by further developing Equation 6.27.

$$\nabla_{\boldsymbol{\lambda}_i} \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)} \left[f(\boldsymbol{x}_i, \boldsymbol{z}_i) \right] \approx \nabla_{\boldsymbol{\lambda}_i} \frac{1}{L} \sum_{l=1}^{L} f\left(\boldsymbol{x}_i, g_{\boldsymbol{\lambda}_i}\left(\boldsymbol{\epsilon}^{(l)}\right)\right) \tag{6.29}$$

The approximate gradient as computed with Equation 6.29 rather than Equation 6.26 exhibits lower variance, enabling better optimization. Contrary to Equation 6.26, Equation 6.29 doesn't require permuting expectation and gradient operator, the MC approximation of the expectation can be directly used to approximate the ELBO:

$$\begin{aligned} \mathcal{L}_{\mathrm{ELBO}}\left(\boldsymbol{\lambda_i}, \boldsymbol{x}_i\right) &= -\mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i) \| p(\boldsymbol{z})\right] + \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)}\left[f\left(\boldsymbol{\lambda}_i, \boldsymbol{x}_i, \boldsymbol{z}_i\right)\right] \\ &\approx -\mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i) \| p(\boldsymbol{z})\right] + \frac{1}{L}\sum_{l=1}^{L} \ln p\left(\boldsymbol{x}_i | g_{\boldsymbol{\lambda}_i}\left(\boldsymbol{\epsilon}^{(l)}\right)\right) \\ &= \widehat{\mathcal{L}_{\mathrm{ELBO}}}\left(\boldsymbol{\lambda_i}, \boldsymbol{x}_i\right). \end{aligned} \tag{6.30}$$

The ELBO estimator $\widehat{\mathcal{L}_{\mathrm{ELBO}}}$ is originally called the stochastic gradient variational Bayes (SGVB). The reparameterization trick enables to estimate gradients as the gradient of a MC approximation of the ELBO, instead of an MC approximation of the gradient of the ELBO.

The original reparameterization trick for sampling factorized Gaussian distributions $\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I}\right)$ is denoted by:

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma}\odot\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right) \Rightarrow \mathbf{z} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\sigma}\mathbf{I}\right), \tag{6.31}$$

With $\odot$ the Hadamard (element-wise) product. There are several options to select a particular differentiable transformation for sampling distributions, which are described in Kingma and Welling [2014], and discussed further in section 7.3.

6.4.3 Variational autoencoders, probabilistic autoencoders ?

Auto-encoding variational Bayes (AEVB) is the algorithm that arises when combining amortized variational inference (see subsection 6.4.1) with the SGVB (i.e., the reparameterization trick, see subsection 6.4.2). Specifically, a VAE is an AEVB for which the amortization is performed by using a $\boldsymbol{\phi}$-parameterized neural network $\boldsymbol{g}_{\boldsymbol{\phi}}$ for inferring the variational parameters λ_i.

$$\boldsymbol{\lambda}_i = \boldsymbol{g}_{\boldsymbol{\phi}}\left(\boldsymbol{x}_i\right) \tag{6.32}$$

The combination of this function with the reparameterization trick enables to draw samples from the approximate posterior $q_{\boldsymbol{\lambda}}\left(\boldsymbol{z}|\boldsymbol{x}\right)$ with a parametric, deterministic function of observed data $\boldsymbol{x}$ and an auxiliary random variable $\boldsymbol{\epsilon}$.

$$\mathbf{z} = \boldsymbol{h}_{\boldsymbol{\phi}}\left(\boldsymbol{x}, \boldsymbol{\epsilon}\right) \tag{6.33}$$

The amortized variational distribution will now be denoted $q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}\right)$, as the variational parameters are entirely determined by the observed data $\boldsymbol{x}$ and the neural network $\boldsymbol{g}_{\boldsymbol{\phi}}$. The VAE actually goes a step further by choosing to use $\boldsymbol{\theta}$-parameterized neural networks $\boldsymbol{f}_{\boldsymbol{\theta}}$ for the likelihood model, accordingly denoted $p_{\boldsymbol{\theta}}\left(\boldsymbol{x}|\boldsymbol{z}\right)$. This means that the generative model is actually learned, and not arbitrarily selected.

Finally, the optimization of the ELBO (Equation 6.34) enables to optimize simultaneously the approximate (variational) posterior $q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}\right)$ and the model likelihood $p_{\boldsymbol{\theta}}\left(\boldsymbol{x}|\boldsymbol{z}\right)$, in the form of a neural networks $\boldsymbol{g}_{\boldsymbol{\phi}}$ and $\boldsymbol{f}_{\boldsymbol{\theta}}$.

$$\mathcal{L}\left(\boldsymbol{x}_i, \boldsymbol{\theta}, \boldsymbol{\phi}\right) = -\mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i) \| p(\boldsymbol{z})\right] + \frac{1}{L}\sum_{l=1}^{L} \ln p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i | \boldsymbol{z}_i^{(l)}\right) \tag{6.34}$$

Because of the reparameterization trick, gradients of the ELBO estimator can be directly computed[11].

This specific approximate variational inference setting has been linked to the autoencoder (AE) framework, due to their similarities, thus the name "variational autoencoder". An AE is a two-part neural network, made of an *encoder* and a *decoder*. The encoder transforms input data $\boldsymbol{x}$ into a *code* $\boldsymbol{y}$ usually of lower dimension, whereas the decoder uses the code and attempts to recreate the input data, as a *reconstruction* $\widehat{\boldsymbol{x}}$. An AE goal is to learn compact representations of unlabeled input data. It is trained by minimizing a *reconstruction error*, i.e. a distance between the original input data $\boldsymbol{x}$ and its reconstruction $\widehat{\boldsymbol{x}}$, which is typically a mean squared error (MSE). Because the AE only uses unlabeled data as input, it is a self-supervised method.

As such, VAE can be seen as a variation of AE by seeing the network $\boldsymbol{g}_{\boldsymbol{\phi}}$ that implements the variational distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ as an encoder, and $\boldsymbol{f}_{\boldsymbol{\theta}}$ that models the likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ as a decoder. The ELBO objective (which needs to be maximized), can be seen as a more classical ML loss by considering its opposite (which needs to be minimized). Then, the two terms of the resulting VAE loss (Equation 6.35) can be linked with classical deep learning terminology.

$$\mathcal{L}_{\text{VAE}}(\boldsymbol{x}_i, \boldsymbol{\theta}, \boldsymbol{\phi}) = \underbrace{-\frac{1}{L}\sum_{l=1}^{L} \ln p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i|\boldsymbol{z}_i^{(l)}\right)}_{\mathcal{L}_{\text{rec}}} + \underbrace{\mathrm{D}_{\text{KL}}\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)\|p(\boldsymbol{z})\right]}_{\mathcal{L}_{\text{KLD}}} \tag{6.35}$$

The term $\mathcal{L}_{\text{rec}}$, which is a negative log-likelihood (NLL), represents the ability of the decoder to sample the observed data, and can be thought of as a *reconstruction loss*. In Kingma and Welling [2014], the authors note that the number of latent samples that must be drawn to approximate the ELBO can be set as $L = 1$ when the mini-batch size is large enough. This is because the gradient of the ELBO estimate is averaged between the samples of the batch, so the variance of the gradient estimate is kept low overall. The KLD loss term $\mathcal{L}_{\text{KLD}}$ can be seen as a regularizing term, which encourages the output distribution of the encoder to match the prior distribution. In most applications, including in the original VAE paper, the variational distribution is chosen as a factorized Gaussian and the prior is selected as the isotropic standard Gaussian (i.e. $\mathrm{q}\, p(\boldsymbol{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$). This allows the KLD loss term to have an analytical expression (for a M-dimensional latent space):

$$\mathcal{L}_{\text{KLD}} = -\mathrm{D}_{\text{KL}}\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)\|p(\boldsymbol{z})\right] = \frac{1}{2}\sum_{m=1}^{M}\left[1 + \ln\left(\left(\boldsymbol{\sigma}_i^2\right)_m\right) - \left(\boldsymbol{\sigma}_i^2\right)_m - \left(\boldsymbol{\mu}_i\right)_m^2\right]. \tag{6.36}$$

A synthetic overview of the classical VAE is given in Figure 6.2.

Compared to a classical AE, the "code" of VAE is of probabilistic nature, because it corresponds to the random latent variables that are sampled with the reparameterization trick. Besides, another interpretation of the reparameterization trick can be made: in a deep learning framework that employs automatic differentiation, the reparameterization trick is what enables the propagation of gradients between the encoder and the decoder. As the reconstruction loss $\mathcal{L}_{\text{rec}}$ is computed with the decoder output, gradients must be propagated from there. If the sampling of the latent variable wasn't differentiable, the encoder wouldn't be reached by the gradients, and couldn't be trained.

VAE are commonly seen as "probabilistic AE", with an unusual loss, and a reparameterization trick that enables training. However, this oversimplified definition of VAE that stems from their comparison with AE glosses over key aspects of the approach. It brings about misconceptions that must be dispelled.

[11] It can be noted that the work of Rezende et al. [2014] had a very similar approach to that of the VAE, in that it used neural networks to amortize variational inference. However, because they didn't use a reparameterization trick, they couldn't easily compute gradients of the ELBO, and relied on approximations such as Equation 6.26.

The first common misunderstanding of VAE is about the nature of its decoder. In classical AE, the decoder transforms a code sample $\boldsymbol{y}_i$ into a reconstruction sample $\widehat{\boldsymbol{x}_i}$, on which the loss is directly applied. But in VAE, the decoder network $\boldsymbol{f_\theta}$ actually represents the likelihood distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$. The VAE decoder takes a latent random variable sample $\boldsymbol{z}_i^l$ as input, and outputs the parameters $\boldsymbol{\psi}_i$ of the likelihood and **not reconstructed samples** $\widehat{\boldsymbol{x}_i}$. For instance, the Gaussian decoder as proposed in Kingma and Welling [2014], is a neural network that outputs the parameters of a Gaussian likelihood, i.e. $\left(\boldsymbol{\mu}_i^{(l)}, \boldsymbol{\sigma}_i^{2,(l)}\right) = \boldsymbol{f_\theta}\left(\boldsymbol{z}_i^{(l)}\right)$. A reconstruction can be obtained from the decoder by sampling the obtained likelihood: $\widehat{\mathbf{x}}_i^{(l)} \sim \mathcal{N}\left(\boldsymbol{\mu}_i^{(l)}, \boldsymbol{\sigma}_i^{2,(l)}\right)$.

Another misunderstanding that is related to the first one exposed above, is about the reconstruction loss term of VAE $\mathcal{L}_{\text{rec}}$. This loss term is occasionally wrongly chosen as a MSE between the input of the encoder and the output of the decoder. This is a mistake for two reasons. Firstly, as explained above, it is because the outputs of the VAE decoder are not samples, but parameters $\boldsymbol{\psi}$ of the likelihood distribution $p_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})$. Secondly, an MSE loss doesn't correspond to the NLL term $-\ln p_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})$. A MSE loss is actually only adequate when the likelihood distribution is Gaussian with unit variance, i.e. $p_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{I})$ and the output of the decoder is the $\boldsymbol{\mu}$ parameter. In such case the Gaussian NLL simplifies into the MSE.

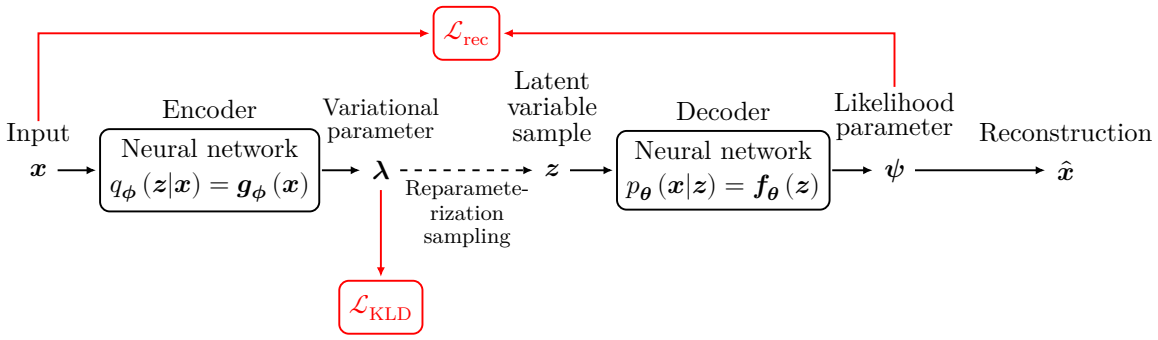


Figure 6.2: Overview of the classical VAE.

6.5 Disentanglement

The VAE framework introduced in section 6.4 offers powerful tools to learn representations of data in an unsupervised manner. These representations are predicted as the probabilistic latent variables. The encoding of VAE has notably some interesting properties. The code of classical AE commonly lacks regularity, because the model is only trained for data dimensionality reduction. In contrast, the latent space of VAE has continuity properties, i.e., two points that are close within the latent space are also close in the input data space. This is because of the probabilistic nature of the encoding. Given an input data $\boldsymbol{x}_i$, the deterministic encoder produces a variational parameter vector $\boldsymbol{\lambda}_i = \boldsymbol{g}_{\boldsymbol{\phi}}(\boldsymbol{x}_i)$. However, what is input to the decoder are not variational parameter vectors, but samples of the variational distribution. Because of the reconstruction loss term, the distribution embedded in the decoder must be able to reconstruct the input data. If the latent space was very irregular, then samples $\boldsymbol{z}_i^{(l)}$ from a given variational distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)$ (thus related to the same input data $\boldsymbol{x}_i$), would produce very different likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ parameters $\boldsymbol{\psi}_i^{(l)}$.

Representation learning with VAE is useful for multiple *downstream tasks*[12]. They can be used for generating new data [Razavi et al., 2019], for clustering [Jiang et al., 2016], classification [Shen et al., 2020], etc. However, the latent representations produced by VAE can lack interpretability and show poor generalizability between applications. Furthermore,

[12]tasks involving the representations of predicted by a model after training.

the inferred latent distributions are often plagued with a phenomenon known as *posterior collapse* that hampers representation quality and robustness, as discussed in subsection 6.5.1.

To improve the quality of latent distributions learned by VAE, additional constraints to enforce desired characteristics are applied. One of the most prolific approaches in representation learning in the last decade has been *disentanglement*. Disentanglement aims at learning representations that match *generative factors*, or *underlying explanatory factors of the data*, as framed by the seminal review Bengio et al. [2013]. Different approaches of *disentanglement* in VAE are discussed in subsection 6.5.2, and their shortcomings are debated in subsection 6.5.3. Finally subsection 6.5.4 concludes on using prior knowledge beyond disentanglement.

6.5.1 Collapse of latent distributions

One common issue with VAE is that occasionally, the latent variables may fail at conveying useful information for the decoder. The decoder may learn to reconstruct the original data without taking a given latent component into account. In such case, the corresponding distribution is no longer constrained by the reconstruction loss to convey information, and is forced by the KLD loss term to match the prior. This phenomenon is commonly known as *latent distribution collapse* Wang et al. [2021], or *information preference* Chen et al. [2017]. It negatively affects the quality of latent representations, because it removes all meaning to one ore more components. This phenomenon occurs in particular when the decoder network is too powerful, and becomes able to map latent variables to observed data space with embeddings of lower dimensions. This is in particular the reason why encoder and decoder network architecture should not be symmetric. Disentanglement approaches therefore must avoid the latent distribution collapse, which concentrates the explainability and representativity of the data in fewer entangled variables.

6.5.2 Imposing structures on latent space

Disentanglement assumes that the observed data possesses independent *factors of variations* y_k, and its goal is to capture those factors with different variables z_j in the learned representation. The reasoning is that these factors of variation are agnostic to any downstream task, and because they capture "the true nature" of observed data they are good general representations. Considering an image of a scene which contains an object, a disentangled representation would for instance be a vector of variables whose coordinates encode independently the position of the object, its size, shape, color, the direction of the light source, etc. Conversely, an entangled representation would mix those aspects between the variables, so that the influence of these factors of variations cannot be clearly separated.

In the numerous works on disentanglement, many different strategies to enforce disentanglement are proposed. As reviews Tschannen et al. [2018], disentanglement (among other constraints on latent representations that will be briefly discussed in subsection 6.5.4), can be imposed by using three main mechanisms:

1. regularization of the latent distribution,
2. the choice of specific model architectures,
3. the choice of the prior distribution and posterior (latent) distribution families.

6.5.2.1 Regularizing the ELBO

This approach is based on reformulating the ELBO loss to achieve some sort of regularization and impose statistical properties on the posterior distribution.

β-VAE [Higgins et al., 2017] is arguably the most straightforward among these approaches. It proposes to improve disentanglement in VAE latent representation by introducing a single

hyperparameter β as a weight of the KLD loss term in the ELBO:

$$\begin{aligned}\mathcal{L}_{\beta\text{-VAE}}(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}) &= -\mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{x})}\left[\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] + \beta \mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z})\right] \\ &= \mathcal{L}_{\text{rec}} + \beta \mathcal{L}_{\text{KLD}}.\end{aligned} \tag{6.37}$$

In this framework, better disentanglement is achieved with $\beta > 1$, i.e. by increasing the role of the KLD loss term. This in turns increases the pressure on inferred posterior distributions $q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)$ to match the prior distribution $p(\boldsymbol{z})$, which is kept as the original factorized unit Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The authors argue that more efficient and disentangled representations are produced this way, because the model is constrained to infer posterior distributions $q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)$ whose overlap between each other is increased [Burgess et al., 2018]. When the overlaps between posterior distributions are increased (when $\mathcal{L}_{\text{KLD}}$ is decreased), decreasing the cost of the log-likelihood term $\mathcal{L}_{\text{rec}}$ is achieved by assigning neighboring latent distributions to data-points that are neighboring in the input data space. It is hypothesized that this locality properties of the latent representation encourages the different latent components to specialize and have a unique independent contribution to the reconstruction.

Nonetheless, the β-VAE notoriously worsens the reconstruction ability compared to the orginal VAE, because the bottleneck on information transmission in the latent space is tightened, and the capacity of the latent variables is diminished. As the inferred posterior distributions overlap a lot between themselves (and with the prior distribution), it becomes more difficult to discriminate between the observed data used as input.

Other regularization approaches introduce more complex loss terms. For instance, the PixelGAN [Makhzani and Frey, 2017] adds to the usual VAE loss a negative mutual information[13] term $\mathrm{I}(\mathbf{x}, \mathbf{z})$. Esmaeili et al. [2019] shows that many approaches that add regularization terms can be explained by re-formulating the original ELBO.

Finally, since disentanglement is about finding independent factors of variations, some approaches have proposed constraining the aggregate posterior distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z})$:

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}) = \int_{x \in \mathcal{D}_{\mathbf{x}}} q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\, p(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}. \tag{6.38}$$

The aggregate posterior distribution is a data-set-wide quantity, that in principle shouldn't be available with mini-batch training strategies. Nonetheless, mini-batch estimates are commonly used, for instance with kernel density estimation. Approaches that constrain the aggregate posterior mostly use a dedicated loss term in the ELBO to enforce independence of the distribution components [Mathieu et al., 2019]. Typically, divergences between the aggregate posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ and the usual factorized unit Gaussian $p(\boldsymbol{z})$ prior can be minimized (KLD, Wasserstein distance, Jensen-Shannon divergences, etc.). This is because the original prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$ promotes independence between the latent components, and the aggregate posterior being close to it would encourage its components independence as well. This approach is used by InfoVAE [Zhao et al., 2019], which adds the term $\mathrm{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}(\boldsymbol{z}) \| p(\boldsymbol{z})\right)$ to the ELBO.

However, Kumar et al. [2018] argues that this term doesn't have a closed-form expression, and using an estimator poses optimization challenges. This is why they propose to match the moments of the two distributions instead of minimizing a distance between them, with the DIP-VAE approach (for "*Disentangled Inferred Prior*"). Specifically, they choose to match the covariance matrix of the aggregate posterior to that of the prior (i.e. the identity matrix), by minimizing the l_2-norm between the matrix components. Their approach assumes a Gaussian variational distribution, i.e. $p_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i) \sim \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}_i), \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\boldsymbol{x}_i)\right)$, so that the covariance matrix

[13] The mutual information between two random variables $\mathbf{x}$ and $\mathbf{z}$ is $\mathrm{I}(\mathbf{x}, \mathbf{z}) = \mathrm{D}_{\mathrm{KL}}\left(p(\boldsymbol{x}, \boldsymbol{z}) \| p(\boldsymbol{x})\, p(\boldsymbol{z})\right)$ and is a measure of the information shared between those variables. Its generalization to more than two random variables is the *total correlation* TC.

of the aggregate posterior can be simplified:

$$\begin{aligned}\mathrm{Cov}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}(z) &= \mathrm{E}_{\mathbf{x}\sim p(\boldsymbol{x})}\left[\mathrm{Cov}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}(\boldsymbol{z})\right] + \mathrm{Cov}_{\mathbf{x}\sim p(\boldsymbol{x})}\left(\mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[\boldsymbol{z}]\right) \\ &= \mathrm{E}_{\mathbf{x}\sim p(\boldsymbol{x})}\left[\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\boldsymbol{x})\right] + \mathrm{Cov}_{\mathbf{x}\sim p(\boldsymbol{x})}\left(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x})\right).\end{aligned} \tag{6.39}$$

From this expression, they propose two loss term variants. DIP-VAE-II minimizes the l^2 distance between components of the aggregate posterior full covariance matrix $\mathrm{Cov}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}(z)$ and the identity matrix. DIP-VAE-I only performs this optimization for the term $\mathrm{Cov}_{\mathbf{x}\sim p(\boldsymbol{x})}\left(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x})\right)$ (see Equation 6.39), by remarking that the cross-correlation between the latent components $\mathbf{z}$ are only expressed through this term when the variational posterior is factorized[14] (i.e. $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\boldsymbol{x})$ is a diagonal matrix).

Alternatively, Factor-VAE [Kim and Mnih, 2018] proposes to minimize the total correlation $\mathrm{TC}\left(q_{\boldsymbol{\phi}}(\boldsymbol{z})\right)$ between the aggregate posterior components. This promotes the factorization of latent variables without explicitly using the prior.

6.5.2.2 Moving away from the Gaussian

The choice of the prior and variational distribution family has a strong impact on learned representations. Therefore, there have been some approaches that have proposed to turn away from the classical Gaussian posterior and the factorized standard Gaussian prior. For instance Casale et al. [2018] proposes to use Gaussian processes (GP) as a prior to improve correlations between latent variables.

Additionally, one of the VAE limitations is that it can originally only handle continuous distributions in the latent space. Some approaches have therefore sought to introduce discrete prior and posterior distributions, such as VQ-VAE and VQ-VAE-2 (VQ standing for "vector quantization") [Razavi et al., 2019; van den Oord et al., 2017]. VQ-VAE introduces discrete latent variables by using a deterministic mapping of the encoder output to a given categorical distribution using learnable embeddings.

It can be noted that the methodology developed in the current work warrants using non-Gaussian distributions, because of the need for bounded variables (see section 7.3). Nonetheless, it can be noted that integrating a non-standard prior/posterior configuration often adds theoretical or computational complexity.

6.5.2.3 Adapting the autoencoder architecture

Designing a specific network architecture can be a way to enforce disentanglement between latent representations. A common approach is to use multiple layers of latent representations, so as to obtain a hierarchy between them [Gulrajani et al., 2017]. In so-called *hierarchical models*, the inference of a given layer of latent representation will depend on the sampling of the parent latent layers, i.e. *ancestral sampling* [Kingma and Welling, 2019].

Noticing that non-Gaussian priors can be difficult to use, as discussed previously, Miao et al. [2022] proposes a simple architecture tweak to that of standard VAE to allow more flexibility. With Intermediary Latent Space VAE, they propose to use a parametric deterministic mapping to samples of variational distribution. Then the image of this mapping (in the mathematical sense) is used as the latent variable to enter the decoder. It enables more general and more flexible priors to be incorporated, by removing the need to express them with an explicit density function. It also allows parts of the mapping to be learned during training. A similar technique is proposed in this Ph.D., as will be discussed in section 7.3.

6.5.3 Challenges of disentanglement

Although there are a wide variety of techniques for imposing disentanglement of representations, they mostly fall short of expectations of a generalized, interpretable, downstream

[14] This is the usual mean-field approximation.

task agnostic representation, beyond simple toy examples or simplistic real-case scenarios. One of the main issues with disentanglement, is that its original definition as the *retrieval of factors of variation within the data* [Bengio et al., 2013] is rather vague. Consequently, all works that attempt to tackle disentanglement end up augmenting this definition with properties that are fulfilled within the framework that they propose. For instance, many posit that disentanglement is characterized by a factorized aggregate posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ (thus the regularization approaches that focus on this term). Another formulation is the independence of features in latent representations.

Some other works attempt to improve on the definition of disentanglement, such as Higgins et al. [2018], which attempt to formalize the concept as the retrieval of symmetries that exist within the data, and links it with group theory. Furthermore they argue that disentangled representations should fulfill three requirements, formulated as:

- *modularity*, that posits that a latent component of a disentangled representation should encode at most one factor of variation of the data,
- *compactness*, that measures whether all factors of variation are each encoded by a single latent component,
- *explicitness*, according to which factors of variations could be retrieved from the disentangled representation up to a linear transformation.

Nonetheless, they note that these requirements do not make consensus.

Mathieu et al. [2019] proposes an alternative extension of disentanglement into *decomposition*, with two requirements. The first is that latent encodings should have an "appropriate" amount of overlap. This is derived from the β-VAE framework, that promotes disentanglement through latent overlapping. The second requirement is that the aggregate posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ should match the prior $p(\boldsymbol{z})$. This condition echoes the regularization techniques introduced in subsubsection 6.5.2.1. Still, this increases the importance of the choice of the prior distribution, that is the factorized unit Gaussian for the majority of works.

A symptom of the lack of a universally accepted view on disentanglement, is the variety of metrics that attempt to measure disentanglement. Many disentanglement papers also propose their own metric, which coincidentally the proposed method is good at.

For instance, the Z-diff score of β-VAE [Higgins et al., 2017], improved in FactorVAE [Kim and Mnih, 2018] propose to use as metric the accuracy of a linear classifier that predicts which generative factor is $\mathbf{y}_k$ kept constant among varying others in batches of latent representations $\mathbf{z}$. In Chen et al. [2018a], the *mutual information gap* measures how much each generative factor y_k is related to a single latent component z_j, and uses the mutual information between pairs of latent components and generative factors:

$$\mathrm{MIG} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{\mathrm{H}(\mathrm{y}_k)}\left(\mathrm{I}\left(\mathrm{z}_{j_k^\star},\mathrm{y}_k\right) - \max_{j\neq j_k^\star}\mathrm{I}\left(\mathrm{z}_j,\mathrm{y}_k\right)\right) \quad \text{s.t. } j_k^\star = \arg\max_j \mathrm{I}\left(\mathrm{z}_j,\mathrm{y}_k\right). \tag{6.40}$$

It can be noticed that these metrics directly measure the ability of latent representations to predict *true* factors of variation. This limits the use of these metrics in real-case scenarios, where the generative factors are usually unknown, or not measured. Eastwood and Williams [2018] proposes metrics of *disentanglement*, *completeness* and *informativeness*, following a description of disentanglement properties that are similar to that of Higgins et al. [2018] discussed above. As Sepliarskaia et al. [2021] remarks, the proposed metrics quantify some properties rather than disentanglement itself. Many metrics even fail at satisfying two basic properties: assigning a high score to representations that are disentangled (according to a given definition) and assigning a low score to those that aren't.

Finally, disentangling latent representations is by essence a subjective enterprise, because it entails enforcing some kind of prior onto the learning process. Locatello et al. [2018] shows

that achieving disentanglement in an unsupervised way (i.e. while only having access to observations $\boldsymbol{x}$ during training) is impossible without incorporating *biases*, i.e. arbitrary assumptions about the system. They also show that a factorized aggregated posterior doesn't guarantee uncorrelated representation components. Also, according to their findings, representations that are deemed disentangled by a given metric do not show improved performance on independent downstream tasks, suggesting that performance gains for disentangled methods were not necessarily due to disentanglement itself.

6.5.4 Introducing prior knowledge in representation learning

As discussed in the previous sub-section, it might be illusory to try to infer general representations that maximize performance in all down-stream tasks, while providing the learning task no insight on what is expected. This is why incorporating prior knowledge is crucial Locatello et al. [2018]. In the end, there is always intention behind representation, there is no good representation in-itself. There are several types of prior knowledge that can be incorporated into representation learning Bengio et al. [2013], besides disentanglement. Such "meta-priors" [Tschannen et al., 2018] include:

- disentanglement, as already discussed,
- hierarchy between explanatory factors, which can describe various levels or abstraction of the data,
- natural clustering, by having representations of different classes being associated with different manifolds within the latent space.

Enforcing those priors is believed to have the potential to improve the learned representations.

Karniadakis et al. [2021] categorizes three types of biases, according to the range of associated techniques, that can be incorporated to guide learning of better representations: *observational biases*, *inductive biases* and *learning biases* (see Figure 6.3).

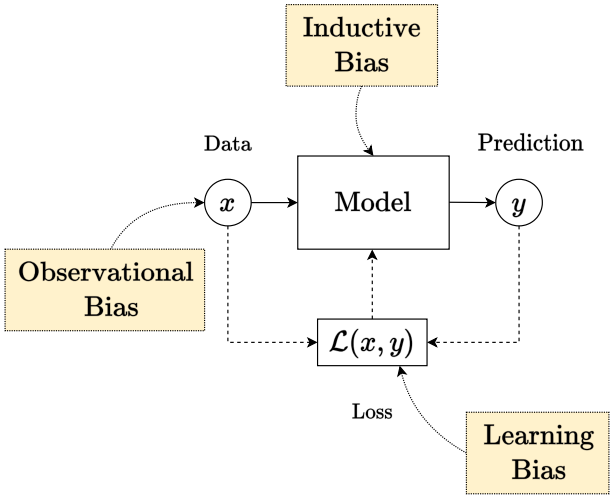


Figure 6.3: Incorporation of prior knowledge in machine learning models.

Observational biases are biases brought through the choice of the data used to train the model. For instance the distribution of samples, is one such bias, and as discussed in section 5.1 and section 5.2, it has a great influence over the model performance. Inductive biases are incorporated by tailoring the machine learning models themselves, their architecture, so that learned representations adopt specified behaviors. Convolutional neural networks (CNNs) typically enforce spatially consistent representations. The techniques discussed in subsubsection 6.5.2.3 and subsubsection 6.5.2.2 can be thought of as inductive biases.

Learning biases are enforced through objective functions. The regularization of the ELBO for achieving disentanglement falls under this category (see subsubsection 6.5.2.1).

In this work representation learning is to be applied on remote sensing data that have a distinctive property, compared to many standard datasets commonly used in the machine learning community (MNIST [Deng, 2012], CelebA [Liu et al., 2015], ImageNet [Deng et al., 2009], etc.). Remote sensing images are measurements of physical quantities, that are bound by the laws of physics. Earth surface processes studied with remote sensing are the objects of a rich literature, that produced many models, as discussed in section 9.1 and Chapter 4. Therefore, an interpretable useful representation of remote sensing data should be a representation that incorporates some of this expert knowledge as priors. If possible, the latent representation itself should match physical quantities. Besides, physical quantities are not necessarily disentangled, on the contrary they may be tied with strong correlations, yet they are arguably good representations of a physical system. In the next chapter, methods of incorporating knowledge about physical data into the VAE framework will be discussed.

Chapter 7

Learning physical representations

Contents

Space-borne Earth observation with remote sensing has for object of study measurements of ground physical processes. As such there is a wide variety of kinds of information that can guide the representation learning to embeddings that are interpretable and physically consistent. Introducing priors pertaining to physical knowledge into machine learning is broadly referred to as *physics-informed machine learning* [Karniadakis et al., 2021], *scientific machine learning* [Rackauckas et al., 2021], or even *hybrid machine learning* [Kurz et al., 2022]. These approaches attempt to bridge the gap between so called *knowledge-driven modeling* (or *first principle modeling*) and *data-driven*, (or *empirical modeling*). While the former reflects physical laws and fundamental properties, the latter is about building models based on the observation of data (with traditional machine learning as the canonical example). These two modeling approaches are also commonly respectively referred as *Newtonian* and *Keplerian* paradigms. Incorporating physical knowledge in the form of biases into data-driven approaches is about leveraging the advantages of both worlds. The powerful modeling and simulating abilities of machine learning can scale up to large problems and data. Including physical knowledge enables to improve robustness, interpretability and explainability of models. Moreover it can help mitigate the lack of reference data, in the small label / big data regime [Karniadakis et al., 2021], which plagues many disciplines including remote sensing [Camps-Valls et al., 2021]. As such, even supervised neural network regression like the biophysical variable neural network (BVNET) (subsection 5.2.1), can be classified as "physic-informed", or "physics-aware". Indeed, training a supervised model to retrieve vegetation variables is impossible with a purely data-driven approach, since reference data is too scarce. Simulating training data-sets with physical models (i.e. PROSAIL, see Chapter 4) allows to train models despite that. Nonetheless, as discussed below, integrating physical knowledge into Machine Learning (ML) models goes beyond simply the simulation of a training data-set. Integrating remote sensing physics into deep learning approaches is framed as "geoscience-aware deep learning" by Ge et al. [2022].

In section 7.1 unsupervised methods for incorporating physical priors into machine learning are reviewed. A methodology to integrate physical model into a variational autoencoder (VAE) framework is proposed in section 7.2. It is based on the idea of replacing the traditional learnable decoder neural network by a physical model. In particular, because many physical models are mechanistic they are usually deterministic, therefore adjustments are required so as to borrow the VAE framework. In section 7.3, the choice of variational distributions and priors is discussed.

7.1 Introducing physical biases into machine learning

As discussed in subsection 6.5.4, priors incorporated into a ML framework can be divided into observational biases (biases related to the learning data), learning biases (related to the learning procedure, and the objective function), and inductive biases (about the architecture of the trained model) [Karniadakis et al., 2021]. Willard et al. [2020] does an extensive survey on the integration of physics in ML models. They classify a sizable literature in terms of applications, such as data generation, uncertainty quantification, and inverse modeling, and in terms of methods used to integrate physics, that can be understood as instances of incorporation of biases. These biases are practically never used separately, because priors are about implicit or explicit assumptions that surround a given model. There are always priors of all three types into each proposed method, although the focus of some works leans towards a particular bias type.

For instance, Yang et al. [2022] proposes a VAE-based data-augmentation method to generate samples of seismic wave velocity maps. To improve physical consistency of generated samples, they consider two additional loss terms: a perception loss[1] (i.e. a learning bias),

[1] A perception loss or perceptual loss measures an error in terms of high-level features extracted from a pre-trained neural network, instead of an error in the original data space.

and a physical regularization loss that encourages temporal consistency between samples generated from a sequence of input data (i.e. a learning bias and observational bias). Arguably, their choice of model architecture as convolutional is an inductive bias, since it favors the prediction of spatial features within seismic velocity maps. It could be noted however, that this work has two misunderstandings about VAE, that were discussed in subsection 6.4.3. Firstly, their decoder outputs are directly samples (thus the decoder is deterministic), when in a classical VAE it should be a distribution. Secondly, they use a mean squared error (MSE) reconstruction loss between input and output samples, when classical VAE would use a negative log-likelihood (NLL) between input samples and distributions of output reconstructions.

One of the most notable advances in physics-informed ML in the last decade is the physics-informed neural networks (PINNs) [Rackauckas et al., 2021; Raissi et al., 2017a,b, 2019]. PINN tackle systems and data that can be described with partial differential equations (PDEs), which are ubiquitous in physics and engineering. PINN approaches have been applied to fluid motion [Cai et al., 2021a], quantum computing [Vadyala and Betgeri, 2023], heat transfer [Cai et al., 2021b], mechanics [Zhang et al., 2024], fusion plasma physics [Rossi et al., 2023], etc. PINN can be used both for solving partial differential equations and performing equation identification (i.e. discovering PDE terms). In both cases, a PINN is a $\boldsymbol{\theta}$-parameterized neural network whose inputs are domain coordinates $\boldsymbol{r} \in \mathbb{D}_{\boldsymbol{r}}$ (time, space, spectral, etc...) and whose outputs $\widehat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{r})$ approximate the solution $\boldsymbol{u}(\boldsymbol{r})$ to a given PDE in the form:

$$\mathcal{P}\left(\boldsymbol{u}(\boldsymbol{r}), \mathcal{N}_{\boldsymbol{\lambda}}[\boldsymbol{u}(\boldsymbol{r})]\right) = f_{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{r})) = \boldsymbol{0}. \tag{7.1}$$

$\mathcal{N}_{\boldsymbol{\lambda}}$ is a non linear, λ-parameterized operator (learnable or not), that can incorporate derivatives of $\boldsymbol{u}(\boldsymbol{r})$. Specifically, PINN make use of modern automatic differentiation techniques to compute the derivatives of $\widehat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{r})$. It is the model loss that enforces the PDE and ensures that the neural network output $\widehat{\boldsymbol{u}}$ is indeed a solution to it (thus a learning bias):

$$\mathcal{L}_{\text{PINN}}(\mathcal{D}_{\boldsymbol{x}}, \boldsymbol{\lambda}, \boldsymbol{\theta}) = \underbrace{\frac{1}{N_j} \sum_{\boldsymbol{r}_j \in \mathbb{D}_{\boldsymbol{r}}} \left\| f_{\boldsymbol{\lambda}}\left(\widehat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{r}_j)\right)\right\|_2}_{\text{MSE}_{\boldsymbol{f}}} + \underbrace{\frac{1}{N_i} \sum_{(\boldsymbol{x}_i, \boldsymbol{r}_i) \in \mathcal{D}_{\boldsymbol{x}} \times \mathbb{D}_{\boldsymbol{r}}} \left\| \widehat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{r}_i) - \boldsymbol{x}_i \right\|_2}_{\text{MSE}_{\boldsymbol{x}}}. \tag{7.2}$$

The $\boldsymbol{r}_j$ are a finite set of N_j evaluation points, associated with the loss term $\text{MSE}_{\boldsymbol{f}}$ that encourages the neural network output $\widehat{\boldsymbol{u}}_{\boldsymbol{\theta}}$ to be solution to the PDE. The N_i data-points $\boldsymbol{x}_i$ of the training data-set (or mini-batch) $\mathcal{D}_{\boldsymbol{x}}$, with associated coordinates $\boldsymbol{r}_i$ are compared to the neural network output $\widehat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$ in the loss term $\text{MSE}_{\boldsymbol{x}}$, to ensure that the approximate solution matches observed data. Boundary conditions can be similarly applied by means of an additional MSE term in the loss. Besides, choosing to represent a solution to a physical PDE with a neural network is a strong inductive bias. Recently, the PINN approach has been combined with VAE in Zhong and Meidani [2023] to enable solving stochastic PDE.

Nonetheless, in the context of VAE, these approaches can be delved deeper. Firstly, although added physical constraints may improve the interpretability of latent representations, it is still not guaranteed that the latent vector will match a specific physical quantity. Secondly, the PINN approach presented above is limited to cases where the associated physical system is described by a differential equation. As it will be discussed in the next section, there is a particularly interesting approach to be taken with autoencoding frameworks: to incorporate inductive biases while tweaking the generative process of the model.

7.2 Physical models as decoders

Autoencoder (AE) and VAE have a remarkable property when it comes to the relation between the encoder and the decoder. The decoder, being a generative model of the data, acts

as a forward simulator with the latent variables as input, and the observed data as output. In this setting, the encoder performs inversion of the model embedded in the decoder. Without additional constraints, the decoder and encoder can be optimized to be any forward and inverse model of the data. The latent embedding is then the inversion of an arbitrary model of the data. With AE and VAE, representation and model inversion are the two sides of the same coin. This in itself can be interesting, as this embedding has been proven to be useful for downstream tasks. With disentanglement (see section 6.5), or physics-informed ML (see section 7.1), additional properties can be given to the encoder, decoder and latent representation of VAE. Still, the latent representation can converge to be arbitrary combinations and rotations of physical variables.

The indeterminacy of latent representations comes from the fact that the latent variables are unspecified and uninterpretable as long as the decoder network is an unknown simulator of the data. What if, instead of a black-box neural-network, the decoder that simulates the reconstructed data was a known physical model? In subsection 7.2.1, the concept of using a user-defined physical model as a VAE is introduced. In order to enable a model likelihood estimation for computing the evidence lower bound (ELBO), a variation of the Gaussian log-likelihood reconstruction loss is proposed in subsection 7.2.2. Finally, subsection 7.2.3 investigates the possibility of estimating a non-factorized Gaussian likelihood to enable taking a structured uncertainty into account.

7.2.1 User-defined decoders

Substituting the neural network in traditional decoder by a user-defined physical model is a strong inductive prior that renders the latent variables entirely interpretable. In such a setup, the latent variables are semantically tied to the variables of the chosen model, and the encoder is forced to produce a semantic encoding (i.e. with a specific meaning) of the input data. Like with a neural network decoder, it is latent realizations that are input to the user-defined decoder (UDD). As discussed above, the trained encoder becomes an estimator of the inverse of the selected model.

This concept is used for instance in Aragon-Calvo [2020] in the case of a deterministic AE, in an approach denoted by "semantic autoencoder". They use an AE to retrieve galaxy characteristics from monochromatic images. By replacing the neural-network in the decoder by a simple exponential model of elliptic galaxies, they force each coordinate of a three-dimensional encoding to match precisely the three parameters of the said model (major semi-axis, ellipticity and position angle).

When applied to VAE instead of deterministic AE, the inversion of the user-defined model becomes probabilistic, because each prediction is associated with a latent distribution. A general framework to incorporate a user-defined decoder into VAE is proposed by Takeishi and Kalousis [2021b]. They propose a decoder that is composed of two parts: a learnable ($\boldsymbol{\theta}$-parameterized neural network) decoder f_A, and a physics-based user-defined, non learnable decoder f_P. To match these decoders, the latent variable vector is split into two parts $\mathbf{z} = [\mathbf{z}_P, \mathbf{z}_A]$, respectively associated with f_P and f_A. This means that the resulting latent representation is made of an interpretable and identified part which matches a physical model, and a non-interpretable part associated with a learned generative process. This is especially interesting to account for effects in the observed data $\boldsymbol{z}$ that are ignored or not well modeled by f_P. It can both act as a complement to the physical model and a correction of residuals or modeling errors. The likelihood model $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ associated with the decoder is a distribution that is a functional[2] of the decoders and the latent space $\mathcal{F}[f_P, f_A, \boldsymbol{z}_P, \boldsymbol{z}_A]$. They choose the likelihood to be a (non-factorized) Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$. As for the variational distribution, they use Gaussian distributions and introduce a hierarchy between

[2]The decoders architectures and hidden features themselves may be accessed, this can be more than a function of inputs and outputs of the decoders.

the latent vectors $\mathbf{z}_P$ and $\mathbf{z}_A$, by conditioning the sampling[3] of the interpretable $\mathbf{z}_P$ on the uninterpretable $\mathbf{z}_A$:

$$q_{\boldsymbol{\lambda}}\left(\boldsymbol{z}_P, \boldsymbol{z}_A | \boldsymbol{x}\right) = q_{\boldsymbol{\lambda}_A}\left(\boldsymbol{z}_A | \boldsymbol{x}\right) q_{\boldsymbol{\lambda}_P}\left(\boldsymbol{z}_P | \boldsymbol{x}, \boldsymbol{z}_A\right). \tag{7.3}$$

Finally, to compute the ELBO, they define the Gaussian priors $p\left(\boldsymbol{z}_A\right) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$ and $p\left(\boldsymbol{z}_P\right) \sim \mathcal{N}\left(\boldsymbol{\mu}_P, \boldsymbol{\sigma}_P^2 \mathbf{I}\right)$, with $\boldsymbol{\mu}_P$ and $\boldsymbol{\sigma}_P^2$ defined as prior knowledge. The choice of prior distributions for variables with physical meaning is further discussed in subsection 7.3.4. The proposed configuration is represented in Figure 7.1. They frame this approach as *physics-integrated VAE*[4]. They consequently apply their proposed framework for the retrieval of human gait characteristics by using a gait engine as the decoder of a VAE [Takeishi and Kalousis, 2021a].

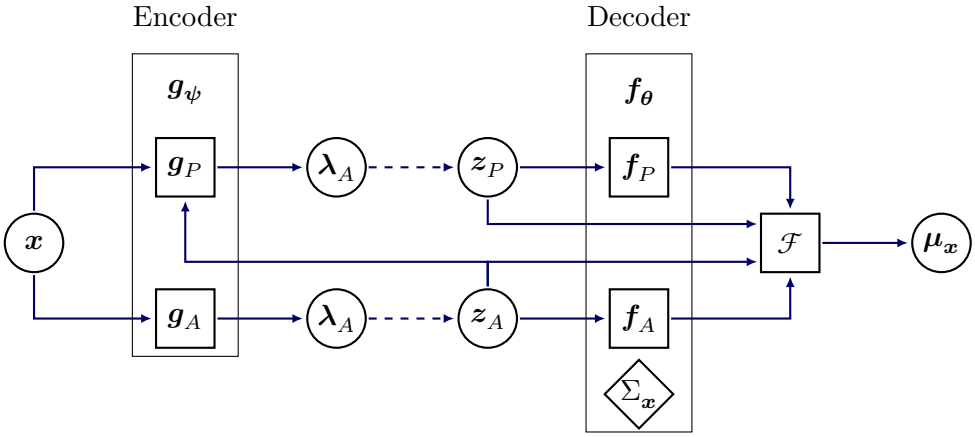


Figure 7.1: Physics-integrated variational autoencoder. Full lines indicate input or output, and dashed lines indicate a reparameterization sampling. The covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ is learned, not predicted.

Overall, there are a few constraints associated with a UDD that can limit its use:

- The model must be *generative*, i.e. it must be a simulator that can produce samples of observed data.
- The model must be a function of some input variables that are used as latent variables.
- As it must be used within a deep-learning architecture as a replacement of a decoder, gradients of the model outputs with respect to (w.r.t.) the model inputs must be computable. Ideally, the model itself should be differentiable. Gradient approximations may be used, such as finite differences [Svendsen et al., 2021], but they are less accurate and computationally efficient.

In this Ph.D., a similar framework to integrate physical models into VAE has been developed. This methodology, on which the remaining part of this manuscript is based on, concentrates on cases where the physical-model takes up the entire decoding function, so that there is no additional learnable decoder, and all latent variables are interpretable. Adding complementary uniterpretable latent variables could be the focus of future work. The applications of this methodology are discussed in Chapter 9 and Chapter 8.

7.2.1.1 Interpretation of user-defined decoders

When using a VAE with UDD, the encoder effectively performs the inversion of the physical model. Discarding the probabilistic nature of retrieved variables, the approach can appear

[3] I.e. ancestral sampling.

[4] Perhaps this naming is more adequate to this method than the more general "physics-informed", that characterizes the whole domain of study. With a UDD, the model is not just "informed" of physics, physics is explicitly part of it.

similar to a supervised regression technique for which simulations are used as a training data-set. Indeed, a supervised regression model is trained on a simulated data-set (see Chapter 5), whereas for VAE with UDD, simulations are performed during training for each encoded sample. Both methods are thus simulation-dependent. Therefore, a VAE with UDD can seem like "supervised regression inversion with extra steps", with the encoder being related to a regression model. However, this interpretation misses a key aspect of the method. In a VAE with UDD input data can be unlabeled real data, unlike supervised regression models. This is because optimization is not directly performed on the retrieved variables, but on the observed data and its simulations/likelihood, i.e. the burden of the loss is transferred to the observed data instead of relying on the label. Even-though both approaches are simulation-driven, their role is not identical: the order of simulation and variable estimation is reversed in VAE with UDD. The VAE with UDD performs *self-supervised inversion*, with one essential advantage: unlike supervised regression, no distribution of the retrieved variables has to be postulated. This necessity of supervised regression has massive influence over the accuracy of the inversion, as shown in section 5.2. Instead of the distributions of the latent physical variables, it is now the distributions of the unlabeled observed data that must be selected, which are arguably a much simpler choice.

A VAE with UDD bridges the gap between the two modeling approaches, the knowledge-driven paradigm and the data-driven paradigm. A physical model as a decoder enables to embed first-principle models into a data-driven approach. Indeed, the data-based representation learning methodology is linked to the physical model inversion.

7.2.1.2 Variance estimation for Gaussian likelihood

Gaussian likelihood models are a reasonable choice for VAE when dealing with continuous real-valued data, all the more so for the output of a physical UDD.

Estimating a Gaussian likelihood for a VAE is about estimating its mean parameter $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$ (or covariance matrix $\boldsymbol{\Sigma}$) for computing the NLL reconstruction term of the ELBO. The $\boldsymbol{\mu}$ is ubiquitously defined as the output of the VAE decoder, and this is still valid for a UDD. However to estimate the parameter σ (or $\boldsymbol{\Sigma}$), several choices are available:

- $\boldsymbol{\sigma}^2$ can be preset to a constant value. This is a very strong assumption that can degrade performances. The classical "error" of mistaking the reconstruction loss for a MSE is equivalent to (unknowingly) setting $\boldsymbol{\sigma} = 1$.
- $\boldsymbol{\sigma}^2$ can be a learnable parameter, optimized simultaneously with the encoder and decoder networks. This option was chosen by the previously discussed physics-integrated VAE methodology [Takeishi and Kalousis, 2021b].
- $\boldsymbol{\sigma}^2$ can be the output of a neural network. This assumes an heteroscedastic[5] likelihood model, and it is the option from the original VAE paper [Kingma and Welling, 2014].

The work in Rybkin et al. [2021] discusses further these different possibilities.

These options for variance estimation are all available for a VAE with UDD. With physical data, uncertainty is reasonably sample dependent, therefore the two first homoscedastic options are discarded. Estimating the likelihood variance with a neural network can be harder to train and doesn't necessarily lead to improved performance. Therefore, for the purposes of this work, a different method to estimate heteroscedastic likelihood without additional neural network has been proposed, and is explained in subsection 7.2.2.

7.2.2 Monte Carlo reconstruction loss

The physics-informed deep learning methodology presented in this work proposes the use of physical-based UDD $\mathcal{F}$ as shown in Figure 7.2. However, many physical models such as

[5] with sample dependent, varying standard deviation.

phenological models (see section 9.1) or PROSAIL (see Chapter 4) are deterministic. The VAE framework requires the $\boldsymbol{\theta}$-parameterized decoder to embed a likelihood model $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$, that can be framed as a "distribution of reconstructions". Specifically, the so-called *reconstruction loss term* of the ELBO objective is approximated with Monte Carlo (MC) sampling (see subsection 6.4.2):

$$\begin{aligned}\mathcal{L}_{\text{rec}}(\boldsymbol{x}_i, \boldsymbol{\phi}, \boldsymbol{\theta}) &= -\mathrm{E}_{\mathbf{z}_i \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)}\left[\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\mathbf{z}_i)\right] \\ &\approx -\frac{1}{L}\sum_{l=1}^{L} \ln p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_i^{(l)}).\end{aligned} \tag{7.4}$$

For mini-batches large enough, L can be set to 1 without hampering training, thus the reconstruction loss term is simplified as $\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_i)$, and the summation is omitted. For deterministic decoders, a model likelihood $p(\boldsymbol{x}|\boldsymbol{z})$ is not available[6][7]. In the original VAE, the decoder outputs the parameters $\boldsymbol{\psi}$ of the model likelihood and uses them to compute the reconstruction loss. But a deterministic decoder can only produce reconstructions $\mathcal{F}(\boldsymbol{z})$[8].

Therefore, it is proposed here to estimate the parameters $\boldsymbol{\psi}$ of a model likelihood by MC sampling. Considering physical models with real-valued simulations, the model likelihood is specified as a continuous distribution. With no additional hypothesis, it is chosen as a factorized Gaussian (like in most VAE implementations), since it is computationally convenient. i.e. $p(\boldsymbol{x}|\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{\psi}), \quad \boldsymbol{\psi} = \left(\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\sigma}^2}\right)$. The Gaussian parameters are estimated by drawing K latent vector samples $\boldsymbol{z}^{(k)}$ and propagating them with the UDD into K reconstructions $\mathcal{F}\left(\boldsymbol{z}^{(k)}\right)$:

$$\widehat{\boldsymbol{\mu}}(\mathbf{z}) = \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\mathcal{F}(\mathbf{z})\right] \approx \frac{1}{K}\sum_{k=1}^{K} \mathcal{F}(\boldsymbol{z}^{(k)}), \tag{7.5}$$

$$\widehat{\boldsymbol{\sigma}^2}(\mathbf{z}) = \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\mathrm{var}\left(\mathcal{F}(\mathbf{z})\right)\right] \approx \frac{1}{K-1}\sum_{k=1}^{K}\left(\mathcal{F}(\boldsymbol{z}^{(k)}) - \widehat{\boldsymbol{\mu}}(\mathbf{z})\right)^2. \tag{7.6}$$

The likelihood parameters $\boldsymbol{\psi}(\mathbf{z}) = \left(\widehat{\boldsymbol{\mu}}(\mathbf{z}), \widehat{\boldsymbol{\sigma}^2}(\mathbf{z})\right)$ are computed as the MC estimate of the expectation w.r.t. $\mathbf{z}$ of some deterministic function $\mathcal{G}$ of $\mathcal{F}(\mathbf{z})$: $\boldsymbol{\psi}(\mathbf{z}) = \mathrm{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\mathcal{G}(\mathbf{z})\right]$. Alternatively, these parameters could be expressed as deterministic functions of K latent samples $\boldsymbol{\psi}\left(\boldsymbol{z}_i^{(1)}, \ldots, \boldsymbol{z}_i^{(K)}\right)$. Comparatively, a neural-network decoder would have directly output the Gaussian parameters $\boldsymbol{\psi}_{\boldsymbol{\theta}}\left(\boldsymbol{z}_i^{(l)}\right)$ by forwarding single latent samples.

The estimated Gaussian likelihood parameters enable to compute the NLL for a given sample i of the batch is:

$$\mathcal{L}_{\text{MCRL}}(\boldsymbol{x}_i, \mathbf{z}_i) = -\ln p(\boldsymbol{x}_i|\mathbf{z}_i) = \frac{1}{2}\left(\frac{\left(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}(\mathbf{z}_i)\right)^2}{\widehat{\boldsymbol{\sigma}^2}(\mathbf{z}_i)} + \ln\left(\widehat{\boldsymbol{\sigma}^2}(\mathbf{z}_i)\right) + \ln(2\pi)\right). \tag{7.7}$$

This NLL is optimized as a *modified* reconstruction loss term for VAE with a user defined decoder. It is coined in this Ph.D. as the Monte Carlo reconstruction loss (MCRL). The MC sampling of latent space brings up the new hyper-parameter K: the number of latent samples drawn from the latent distribution inferred from each input sample $\boldsymbol{x}_i$. The choice of K is a trade-off between accuracy of $\widehat{\boldsymbol{\mu}}(\mathbf{z}_i)$ and $\widehat{\boldsymbol{\sigma}^2}(\mathbf{z}_i)$, and training time, because each latent distribution sample requires a forward pass through the decoder. Although they are both a number of samples, K is fundamentally a different parameter than L (see Equation 6.26 and

[6] It could be chosen as a Dirac distribution, i.e. $p(\boldsymbol{x}|\boldsymbol{z}) = \delta_{\mathcal{F}(\boldsymbol{z})}(\boldsymbol{x})$. Nonetheless, uses would be limited considering that log-likelihood and its gradients are not defined.

[7] $\boldsymbol{\theta}$ is omitted to designate the likelihood associated with a UDD, because it doesn't have learnable parameters.

[8] Contrary to the VAE decoder, the output of a UDD can be qualified as *reconstructions*, because they are samples, and not distributions.

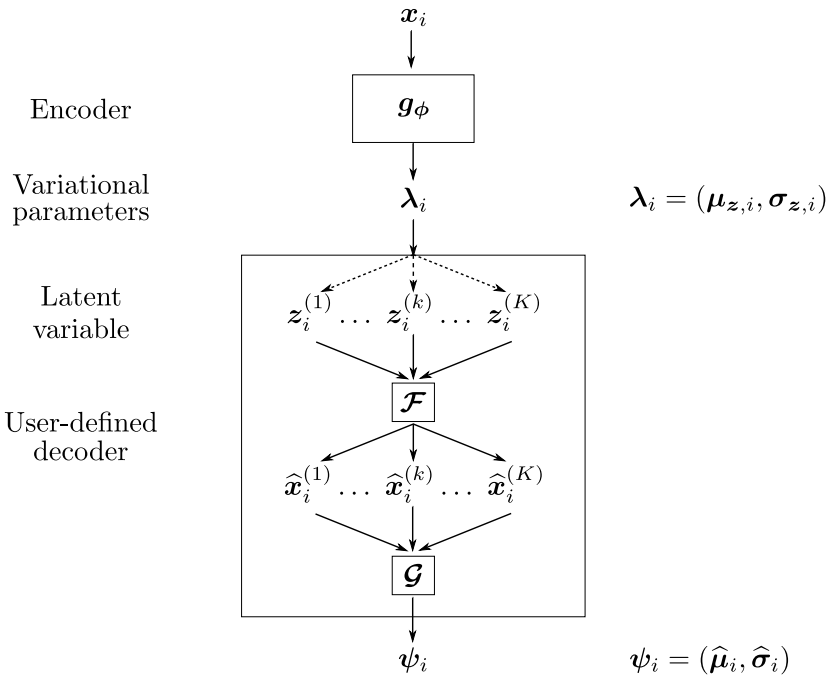


Figure 7.2: VAE with user-defined decoder

Equation 6.34 of Chapter 6), which is the number of MC samples to compute the expectation of the log-likelihood. The parameter $\widehat{\boldsymbol{\sigma}^2}$ is the variance of the model Gaussian likelihood, which is considered factorized, i.e. the covariance matrix is diagonal $\boldsymbol{\Sigma} = \widehat{\boldsymbol{\sigma}^2}\mathbf{I}$. This assumes the independence of the decoder output vector components. The possibility of estimating a non-diagonal covariance matrix for a non-factorized Gaussian likelihood is discussed in subsection 7.2.3.

The MCRL (Equation 7.7) deviates from the classical VAE theory, because the role of MC samples is different. The original log-likelihood terms (see Equation 7.4) $\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_i^{(l)})$ are computed from a single latent vector sample $\boldsymbol{z}_i^{(l)}$. These terms on their own are random in essence, because $\boldsymbol{z}_i^{(l)}$ is sampled (and this is why approximating the expectation w.r.t. $\mathbf{z}_i$ makes sense).

Conversely, the (single) likelihood term $\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\mathbf{z}_i)$ in Equation 7.7 for the MCRL is computed from a set of samples $\left(\boldsymbol{z}_i^{(1)}, \ldots, \boldsymbol{z}_i^{(K)}\right)$. Estimating this likelihood term multiple times (e.g. for deriving an expectation such as in Equation 7.4) would yield similar values each time.

$$-\operatorname{E}_{\mathbf{z}_i \sim q_{\boldsymbol{\lambda}_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)}\left[\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\mathbf{z}_i)\right] \approx -\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\mathbf{z}_i) \tag{7.8}$$

With the proposed MCRL setting, the distribution of reconstructions is inferred by propagating uncertainties of latent variables (i.e. by sampling the latent distribution) into a deterministic model. Perhaps calling the MCRL a negative log "*likelihood*" is inadequate, because it asymptotically doesn't depend on individual samples, but on distributions that are deterministic functions of the sample $\boldsymbol{x}_i$.

Even though the MCRL deviates from the original VAE framework, it has some interesting properties from the optimization point of view. The Gaussian NLL MCRL (Equation 7.7) promotes small sample reconstruction errors (i.e. $\widehat{\boldsymbol{\mu}}_i$ close to $\boldsymbol{x}_i$). It also encourages the reconstruction variance $\widehat{\boldsymbol{\sigma}^2}_i$ to model the decoder uncertainty. If the error isn't small, the variance can be increased to still minimize the loss (e.g. when the error cannot be minimized, uncertainty is increased). The $\ln\left(\widehat{\boldsymbol{\sigma}^2}\left(\mathbf{z}_i\right)\right)$ term prevents the variance from arbitrarily increasing as a trivial way of minimizing the loss.

It is important that the likelihood distribution is chosen in accordance with the selection of the variational distribution (see section 7.3), or more generally on the actual distribution

of the samples outputted by the UDD. In particular, a mono-modal likelihood model (such as the Gaussian used here) isn't adapted to model a multi-modal sample distribution. It can even be detrimental to training. For instance, attempting to estimate the parameters of mono-modal Gaussian likelihood model fails when the reconstruction distribution is bi-modal with modes m_1 and m_2, since a μ parameter MC estimate would neither match m_1 nor m_2 but be placed somewhere in-between. An advantage of the MCRL associated with a UDD, is that it compatible with any likelihood distribution without changing the VAE architecture, as long as the distribution parameters $\boldsymbol{\psi}$ can be reliably and efficiently estimated from samples $\mathcal{F}(\boldsymbol{z})$ of the UDD. By contrast, a VAE with a learnable decoder needs to adapt its architecture to the model parameters.

7.2.3 Multivariate decoder distribution

In subsection 7.2.2, the MC sampling of the latent variables and the forward propagation through the decoder enables the estimation of the parameters of a Gaussian output distribution. However, the underlying assumption is that the decoder output vector $\mathbf{x}$ coordinates are independent and identically distributed (i.i.d.). This grants a simple expression for the NLL reconstruction loss. Nonetheless the decoder model combines the input variables, so that the output vector components are in fact correlated.

The correlations between the decoder output vector components can be taken into account in the reconstruction loss. This is done through the use of a covariance matrix estimate $\widehat{\boldsymbol{\Sigma}}$, instead of using only variance estimates $\hat{\sigma}$ for each component:

$$\begin{aligned}\widehat{\boldsymbol{\Sigma}}(\mathbf{z}) &= \mathrm{E}\left[\left(\mathcal{F}(\mathbf{z}) - \mathrm{E}\left[\mathcal{F}(\mathbf{z})\right]\right)\left(\mathcal{F}(\mathbf{z}) - \mathrm{E}\left[\mathcal{F}(\mathbf{z})\right]\right)^{\top}\right] \\ &\approx \frac{1}{K-1}\sum_{i=1}^{K}\left(\mathcal{F}\left(\boldsymbol{z}^{(k)}\right) - \widehat{\boldsymbol{\mu}}(\mathbf{z})\right)\left(\mathcal{F}\left(\boldsymbol{z}^{(k)}\right) - \widehat{\boldsymbol{\mu}}(\mathbf{z})\right)^{\top},\end{aligned} \tag{7.9}$$

with E the expectation taken w.r.t. $\mathbf{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ The MCRL must then be changed accordingly:

$$\mathcal{L}_{\text{MCRL}}(\boldsymbol{x}_i, \mathbf{z}_i) = \frac{1}{2}\left[\left(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}(\mathbf{z}_i)\right)^{\top}\widehat{\boldsymbol{\Sigma}}(\mathbf{z}_i)^{-1}\left(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}(\mathbf{z}_i)\right) + \ln\left(\left|\widehat{\boldsymbol{\Sigma}}(\mathbf{z}_i)\right|\right)\right]. \tag{7.10}$$

The covariance matrix inverse $\widehat{\boldsymbol{\Sigma}}(\mathbf{z})^{-1}$ and determinant $\left|\widehat{\boldsymbol{\Sigma}}(\mathbf{z})\right|$ are involved in the updated reconstruction loss Equation 7.10, making the computation notably harder. Notably, the inverse of the covariance matrix[9] itself must be computed, and there is no possibility of reducing the computational burden by solving an equivalent linear system. The necessity of inverting this matrix is the reason why a non-factorized Gaussian likelihood isn't used in practice with MCRL. The covariance matrix estimate is very likely to be ill-conditioned, thus preventing the computation of the inverse and hampering training (see subsection B.4.1). Furthermore, even in cases when this matrix is guaranteed to be invertible, the computational cost is high (see subsection B.4.2). For VAE without MCRL, it is possible to use this approach by estimating the matrix inverse directly with a neural network [Dorta et al., 2018].

7.3 The variational distribution as an inductive bias

The use of a physical-based decoder implies that latent variables are tied to physical measurements. Therefore, the usual choices for the prior and posterior distributions may not suit the latent variables with a physical meaning. Informed choices for these distributions must be made depending on the meaning of each individual model variable. In the following, different options for the choice and computation of prior and variational distributions of variables representing physical quantities are investigated.

[9] Also called the *precision matrix*.

7.3.1 Reparameterization sampling techniques

The choice of the variational distribution is limited to distributions that can be sampled in a differentiable way, so that gradients can be propagated through. Three different sampling techniques can be considered to enable various distribution choices Kingma and Welling [2014]:

1. A reparameterization trick to sample *location-scale family* distributions Koike-Akino and Wang [2022], such as the usual Gaussian distribution (see Equation 6.31).

2. The composition of random variables by non-linear functions enables to transform "elementary" distributions into others. For instance, log-normal, logit-normal, Dirichlet, exponential distribution samples can be generated respectively by composing Gaussian with logarithm, Gaussian with sigmoid, Gaussian with softmax Srivastava and Sutton [2017] and uniform with logarithm.

3. The inverse transform sampling (ITS) method described in Equation 7.11 can be used to sample any continuous random variable $\mathrm{z} \sim \mathcal{A}$. This technique can be used when its inverse cumulative distribution function (ICDF) $F_{\mathcal{A}}^{-1}$ is is differentiable almost everywhere. It entails sampling u from $\mathcal{U}(0,1)$ (the uniform distribution), and then calculating the desired z as:

$$\mathrm{z} = F_{\mathcal{A}}^{-1}(\mathrm{u}), \quad \mathrm{u} \sim \mathcal{U}(0,1) \quad \Rightarrow \quad \mathrm{z} \sim \mathcal{A} \tag{7.11}$$

7.3.2 Bounded latent distributions

When using a user-defined physical model as a decoder, the latent variables become associated with a physical meaning, and as such they can have a restricted domain. In particular, these physical variables are often defined on an interval, i.e. bounded. For instance, a concentration can only be a positive quantity below 100%, a normalized difference vegetation index (NDVI) level always belongs between -1 and 1. It is important to ensure that latent variables that are semantically tied to bounded variables are constrained so that their sampling never leaves their definition range. Physical models can be mathematically defined even with out-of-bounds parameters, but samples generated with these parameters would not be realistic. Such reconstructions could still minimize the reconstruction loss, and hamper training while the encoder learns to infer wrong model parameters. This can be especially detrimental when some training samples are not well described by the physical model. As such, the unbounded traditional Gaussian variational distribution cannot be used without performance limitations.

To choose the variational family, the properties that must be imposed on latent variables must be analyzed. Bounded variational distributions which can achieve a variety of densities, with modes reaching the whole intervals, and varying variance, seem especially suited for physical variables. Additional properties can be expected, such as asymmetry (skew), or heavy tailing (kurtosis). Overall, the chosen distributions need as much parameters as the number of distribution moments to be tuned. In the case of mode and spread, two-parameter distributions can be considered.

The sampling techniques described in subsection 7.3.1 can be used to sample bounded distributions. This can be achieved by composing unbounded distribution samples, such as Gaussian samples drawn from the reparameterization trick, with sigmoid functions[10] (logistic[11], hyperbolic tangent, arc-tangent, etc...).

However care must be exerted, because non-linear transformations of random variables changes the dynamic of the distribution. With this method, the transformed distributions are distorted and may become bimodal. For instance, it can be detrimental to apply a

[10] More generally, composing unbounded samples with a monotonic, smooth enough, bounded function.
[11] The composition of a Gaussian distribution with a logistic function is the *logit-normal distribution.*

sigmoid transformation to Gaussian samples obtained with the reparameterization trick to get bounded samples. This is because the resulting logit-normal distribution can be bimodal for some (μ, σ) configurations.

To avoid such limitations, the ITS can be considered instead, because it enables the sampling of distributions that are natively with bounded support such as raised cosine distributions, beta distributions, etc. Still, some computational precautions must be taken: in practice, the gradient of the ICDF computed during training may diverge. In intervals of near-zero density, the cumulative distribution function (CDF) is almost constant at $y = c$ and its reciprocal has infinite derivative at $x = c$. Therefore, uniform sampling of u must be done inside an interval $\mathbb{Y}$ where the CDF is strictly monotonous. In fact, due to the numerical accuracy ϵ, the interval $\mathbb{Y}$ has to be restricted even further (see Equation 7.12).

$$\mathbb{Y} = F_{\mathcal{A}}(\mathbb{X}), \ \mathbb{X} = \{x \in [0,1] \text{ s.t. } \mathrm{d}F_{\mathcal{A}}(x) \geq \epsilon\} \tag{7.12}$$

This means that exceptional (very low probability) events are never observed with the inverse sampling method, although this isn't an issue for the considered applications.

The Kumaraswamy distribution (see section C.2) which has been considered as an option for bounded distributions, can illustrate these considerations. The Kumaraswamy distribution is bounded between 0 and 1, it depends on two parameters a and b , with an analytical simple expression of the probability distribution function (PDF), CDF and ICDF, which makes sampling very easy. This distribution is quite flexible, which can make it useful to model physical variables. Depending on the values of the parameters, the Kumaraswamy distribution has numerical issues because the gradient of the ICDF diverges (see subsubsection C.2.4.1), so the range for the uniform sampling for the ITS must be restricted, as explained above. In the end, Kumaraswamy distributions were discarded for this work because of three reasons:

- Despite a relative flexibility which enables a variety of densities, achieving low variance can require extremely high values of the parameters a and b, diverging beyond the range double precision ($\approx 1.8 \cdot 10^{308}$).
- Analytical expressions of Kullback-Leibler divergence (KLD) between pairs of Kumaraswamy distributions or even between Kumaraswamy and other distributions are intractable.
- The distribution parameters had uninterpretable meaning, because they have a combined effect on all moments.

Truncated normal (TN) distributions were finally chosen as the variational distribution family (see section C.4). These distributions form an exponential family that restricts the usual variational Gaussian into an interval, making it suitable for the considered purposes. A TN is a function of a location $\boldsymbol{\mu}$ and scale parameter $\boldsymbol{\sigma}$ like the usual Gaussian. When the TN has low enough variance, there can be domains within the bounded definition interval that have almost zero density, making gradients of the ICDF diverge. Still, computing a good interval $\mathbb{Y}$ to draw uniform samples for the ITS is straightforward:

$$\mathbb{Y}(\mu, \sigma) = [\max(\mu - n\sigma, 0), \min(\mu + n\sigma, 1)], \tag{7.13}$$

with n the number of standard deviations taken into account. In practice, $4 < n < 5$ is enough. Furthermore, the TN are a *maximum entropy distribution* for bounded probability distributions, meaning that they are one of the least informative distributions among this category. As such, it is an adequate choice because it minimizes the amount of prior information given to the distribution.

One limitation of TN is that there is no expression for the ICDF in the multivariate case. Therefore, the inverse transform method used here cannot be applied. Other sampling methods that are differentiable w.r.t. the distribution parameters (i.e. mean vector and covariance matrix) must be found.

7.3.3 Intermediary latent space

There are additional choices to be made when it comes to the bounds of the physical-based latent variables. First a decision must be made about whether those bounds are pre-set as a prior knowledge, or learned (as an additional output of the neural network, or as an additional learnable parameter). In practice, the former option is easier to implement, doesn't require optimization, and makes use of available knowledge about the physical model in the UDD.

Once the variable bounds $[l, u]$ are fixed, how can they be enforced on the latent variables? One straight-forward option would be to directly predict variational parameters corresponding to distributions bounded on the selected intervals. Alternatively, the variational distributions can be chosen to be bounded all on a common interval $[0, 1]$, and latent samples z are re-scaled with an affine transformation to match the chosen interval $[l, u]$.

$$z \in [0, 1] \Rightarrow (u - l)z + l \in [l, u] \tag{7.14}$$

This configuration has one major advantage compared to simply sampling variables directly on their definition domain. It enables to apply equivalent KLD loss regularization on all the latent variables. The KLD between bounded variational and prior distribution is a function of the domain (e.g. see Equation C.9). Differently-sized intervals could lead to priors being applied unevenly to the latent dimensions. This is why the latter option, which is similar to the *intermediary latent space* approach Miao et al. [2022] (see subsubsection 6.5.2.3), is better.

7.3.4 Prior distribution for physical variables

While the variational distribution should be chosen with physical variables meaning in mind, it also has to be paired with a prior distribution that enables computation of the KLD loss term. It may unfortunately be more complicated to find a meaningful prior whose KLD with the variational distribution admits a closed-form expression[12]. Fortunately, the chosen TN distributions (see subsection 7.3.2) admit analytical KLD (see subsubsection C.4.6.1). This enables to impose a prior similarly to that of the original factorized unit Gaussian prior onto Gaussian variational distributions, or similarly to the *physics-integrated VAE* framework [Takeishi and Kalousis, 2021b] (see subsection 7.2.1).

However, using a TN as prior is akin to promoting a specific mode to the posterior distribution. This is why when the true distribution of a physical quantity associated to the posterior distribution doesn't have the same mode, this prior is detrimental. Furthermore, it has been discussed in Chapter 5 that the true distribution of physical variables often isn't known accurately (justifying the current approach).

To reflect the lack of information on a physical parameter distribution, it is proposed here to use a uniform distribution (over the bounded interval) as a prior. The analytical expression of the KLD between a TN and the uniform distribution is provided in subsubsection C.4.6.2. This prior is the least informative possible, besides using no prior at all. Computing the derivatives of the KLD between a TN and a uniform distribution w.r.t. the TN parameters (μ, σ) (see subsubsection C.4.6.3) enables to assess what kind of distribution is promoted by this prior, which in turn allows to analyze the effect of the KLD loss term in the ELBO. The uniform prior promotes TN posterior with a larger variance, and more weakly encourages a distribution centered on the interval mid point (i.e. $\mu = 0.5$).

7.4 Conclusion

This chapter has presented a methodology for integrating a physical model as the decoder of a VAE. The MCRL was introduced for enabling Gaussian likelihood estimation with a

[12]The KLD between two distributions can be estimated with MC sampling. However KLD estimators can have high variance and require many samples, thus increasing the computational cost.

deterministic decoder. The choice of the variational and the prior distributions for modeling physical variables has been discussed. The different elements of this approach are the building blocks that will enable to predict interpretable representations of land surfaces from remote sensing measurements.

In the next part, this methodology will be applied with the incorporation of two physical models: PROSAIL in Chapter 8 with PROSAIL-VAE, and a temporal, phenological model in Chapter 9 with Pheno-VAE.

Part IV

Applications

Chapter 8

Radiative transfer model inversion

Contents

The methodology developed in Chapter 7 proposes to incorporate a physical model as the decoder of a variational autoencoder (VAE). The inputs of the physical model are semantically tied to the latent variables of the VAE, and as such can be understood from two complementary points of view. The latent variables become a physically consistent representation of the encoded data, and they estimate the input of a model from its output, i.e. they are a solution to a model inversion. The encoder of the VAE is trained to become a inverse model of whatever model is in the decoder. Crucially, it can be trained directly on real data, without having to build simulated training data-sets.

The PROSAIL model introduced in Chapter 4 is a radiative transfer model (RTM) that simulates canopy reflectances from a set of biophysical parameters. When coupled with the model of a remote sensor, it constitutes a physical model that relates ground physical properties to remote sensing observations.

This chapter studies the incorporation of PROSAIL into a VAE as a so-called PROSAIL-VAE, and the training of the model using Sentinel-2 (S2) images. Using in-situ measurement data, the inversion of certain variables of PROSAIL with PROSAIL-VAE can be assessed and compared with other methods. In particular, it is compared with Sentinel Application Platform (SNAP), which is a operational neural-network in SNAP based on the biophysical variable neural network (BVNET) architecture (see Chapter 5).

In section 8.1, the PROSAIL-VAE approach is introduced, alongside two supervised neural-network based deep learning approaches: SNAP and multiple probabilistic supervised regression (MPSR). Then in section 8.2, the performance of one PROSAIL-VAE model is investigated, by comparing its inversion performance against that of SNAP, and by analysing the inference over S2 images. Finally, the section 8.3 discusses different design choices for PROSAIL-VAE.

8.1 PROSAIL inversion methods

8.1.1 PROSAIL-VAE

8.1.1.1 Integrating PROSAIL as the decoder of a VAE

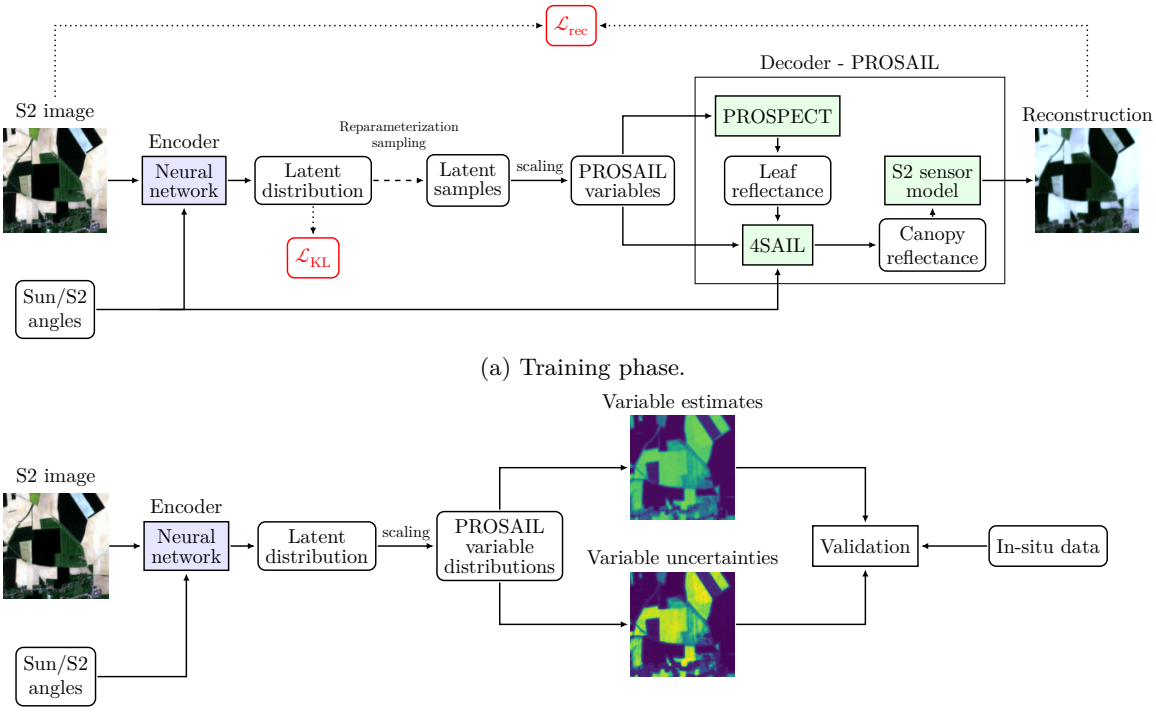
PROSAIL-VAE is defined as a class of VAE in which PROSAIL is integrated as a physics-based user-defined decoder (UDD) (see subsection 7.2.1). This is performed with the developed differentiable implementation of PROSAIL (see section 4.5). The neural-network encoder $f_{\boldsymbol{\phi}}$ takes S2 reflectances and observation angles as input, and outputs the variational parameters $\boldsymbol{\lambda}$ that define the latent distributions, and thereby the approximate posterior approximate $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$. The samples from these latent distributions are semantically tied to the input variables of PROSAIL and taken as input by the PROSAIL-decoder. The decoder then simulates the S2 bands matching the canopy reflectance spectra corresponding to these input variables. These simulations are reconstructions of the encoder input S2 bands.

The training loss of PROSAIL-VAE is the evidence lower bound (ELBO) objective function (see Equation 6.35) weighted by a β hyper-parameter in the manner of β-VAE (see Equation 6.37):

$$\mathcal{L}_{\text{PROSAIL-VAE}} = \mathcal{L}_{\text{rec}} + \beta \mathcal{L}_{\text{KLD}}. \tag{8.1}$$

The reconstruction loss term $\mathcal{L}_{\text{rec}}$ penalizes the ability of the VAE to accurately model its input data. In the present case, it is the ability of the PROSAIL decoder to reconstruct the input S2 bands, from the PROSAIL variables inferred by the encoder. The KLD loss term $\mathcal{L}_{\text{KLD}}$ encourages the approximate posterior to match the prior distribution on latent variables, and acts as a regularizing term. The β coefficient balances both terms in the loss.

The latent variables samples drawn from the inferred variational distribution by the encoder are not directly taken as PROSAIL variables and input to the decoder. PROSAIL-VAE uses an intermediary latent space (see subsection 7.3.3): the latent variables $\boldsymbol{z}$ are constrained to belong to the $[0, 1]$ interval, and PROSAIL variables are obtained by an affine



(a) Training phase.

(b) Inference phase.

Figure 8.1: Description of the proposed PROSAIL-VAE methodology

transformation of the latent variables. The latent distributions are chosen as truncated normal (TN) for ensuring that their samples are bounded (see subsection 7.3.2).

The reconstruction loss doesn't penalize individual reconstructions (i.e. simulated S2 reflectance bands), but a likelihood distribution $p(\boldsymbol{x}|\boldsymbol{z})$. For PROSAIL-VAE, this reconstruction loss is computed as the Monte Carlo reconstruction loss (MCRL) (see subsection 7.2.2). Several sets of latent variables are sampled, transformed into PROSAIL variables and forwarded to PROSAIL for simulating sets of S2 reflectance bands. These sets of reconstructions enable to estimate the parameters of a Gaussian likelihood. Then, the reconstruction loss is computed as the negative log-likelihood (NLL) of the estimated distribution with respect to (w.r.t.) the encoder input S2 bands.

The training workflow of PROSAIL-VAE is summarized in Figure 8.1a. Once trained, the encoder is a fast probabilistic inverser of PROSAIL. During inference, the decoder is discarded, and only the encoder is used for retrieving biophysical variables (see Figure 8.1b).

8.1.1.2 Training of PROSAIL-VAE

PROSAIL-VAE is trained in a self-supervised manner by directly using S2 band reflectances as input. Its training data-set is the S2 patch data-set $\mathcal{D}_{\text{S2}}$, described in section 2.2. This data-set contains patches extracted from S2 images of various locations across Europe and at different dates between 2016 and 2019.

PROSAIL-VAE is initialized by using the multiple initialization and best instance training (MIBIT) approach (see subsubsection 3.3.2.3). It consists in training several deep learning model instances for a few epochs, and then pursuing the training only for the model with the best validation loss. The learning rate (lr) is commanded throughout training with the cyclical plateau reduction (CPR) scheduler (see subsubsection 3.3.2.2). This scheduler reduces the lr when the validation loss doesn't improve for a chosen number of epochs. When the lr crosses a threshold, it is reinitialized to its initial value and the cycle starts over. The Adam optimizer is used to update the weights of the encoder.

8.1.1.3 PROSAIL-VAE base configuration

Table 8.1: Range of the PROSAIL input variables in PROSAIL-VAE

Variable	N	C_{ab}	C_w	C_{car}	C_m	C_{brown}	LAI	$\overline{\alpha}$	h	s_w	s_b
min	1.2	20.0	0.0075	5	0.003	0	0	30	0.0	0	0.3
max	1.8	90.0	0.0750	23	0.011	2	10	80	0.5	1	3.5

The target ranges of the PROSAIL variables for PROSAIL-VAE are shown in Table 8.1. These definition intervals for PROSAIL variables are chosen identical to the canonical sampling intervals used for simulating the training data-set of SNAP in Weiss and Baret [2016], when possible. Since the version of PROSAIL is different from the one used here, the ranges of some variables are unavailable and must be defined separately. The range of the soil parameters s_b and s_w is kept identical to that of the implementation of Domenzain et al. [2019]. The range of the equivalent water thickness C_w is arbitrarily chosen with a very high upper bound. These intervals represent a prior knowledge about the PROSAIL variables. This knowledge is less informative than sampling distributions, and thus much easier to provide.

The prior distribution $p(\boldsymbol{z})$ considered for PROSAIL-VAE are uniform distributions over the range $[0, 1]$. The associated loss term $\mathcal{L}_{\text{KLD}}$ is the KLD of the TN latent distribution $q(\boldsymbol{z}|\boldsymbol{x})$ w.r.t. the uniform prior $p(z)$ (see subsubsection C.4.6.2). This KLD depends on the range of the bounded distributions. Therefore, to ensure that the $\mathcal{L}_{\text{KLD}}$ loss term affects all variables equivalently, the prior is not defined w.r.t. the distribution of PROSAIL variables, but w.r.t. the latent distributions. This is because the latent variables are defined on the same interval, contrary to the PROSAIL variables.

The training of PROSAIL-VAE is relatively random access memory (RAM)-intensive, because of the use of the differentiable version of PROSAIL as a decoder. Applying automatic differentiation throughout the PROSAIL model requires keeping a lot of intermediary variables and their gradients in memory. Furthermore, the use of the MCRL necessitates these computations to occur for several samples at once. As a consequence, the memory of the used graphical processing unit (GPU) (see section A.2) reaches saturation quickly. To lessen the computational burden, the PROSAIL spectra and simulations are down-sampled from an original 1 nm resolution to to 7 nm (see subsection 4.5.4). In the current implementation of PROSAIL-VAE, the used hardware can handle training steps with approximately 7×10^4 total simulations. In practice, this corresponds to simulating $N = 70$ reconstructions (for the MCRL) of a single 32×32 pixels S2 patch (i.e. 1024 pixels). Increasing the mini-batch size (i.e. increasing the number of patches taken into account at each step) requires either reducing the size of the patches or the number N of Monte Carlo (MC) samples for the MCRL. As such, the number N of MC samples for the MCRL is selected jointly with the batch size B, so that the total number of reconstructions at one step is maximized without the GPU memory being saturated, i.e. $N \times B \approx 7 \times 10^4$.

The neural network architecture for the encoder of PROSAIL-VAE is described in Figure 8.2. It is based on a residual network backbone (see subsubsection 3.3.1.3), with three residual connection blocks, and they infer bio-physical variables (BV) for each pixel of the input images. All layers are 32 neurons wide, and are connected with ReLU activation. The residual blocks are two layers deep. Two variants for this architecture are proposed. The first variant is a pixel-wise multi-layer perceptron (MLP) (see subsubsection 3.3.1.1) for which the neurons are fully connected and which handles independently each pixel of the input patch. The second is a convolutional neural network (CNN) (see subsubsection 3.3.1.2), for which the neurons are convolution filters. This architecture is designed to allow capturing the spatial context information within patches. However, only the first layer of this architecture uses 3×3 sized filters, the remaining layers are 1×1. Therefore, only the very close neighborhood of a pixel is taken into account by this architecture.

For each S2 input pixel, 17 features are taken into account by the encoder:

- 10 S2 reflectance bands : `B2`, `B3`, `B4`, `B5`, `B6`, `B7`, `B8`, `B8A`, `B11`, `B12` (all except `B1`, `B9` and `B10`)
- 3 Sun / S2 angles: the Sun zenith angle θ_S, the S2 zenith angle θ_O and the relative azimuth angle ψ_{SO} (see subsection 2.1.2),
- 4 spectral indices derived from S2 bands.

The 4 spectral indices taken as input are:

- the normalized difference vegetation index (NDVI) (see subsection 1.2.1):

$$\text{NDVI} = \frac{\texttt{B8} - \texttt{B4}}{\texttt{B8} + \texttt{B4}} \tag{8.2}$$

- the normalized difference infrared index (NDII) [Klemas and Smart, 1983]:

$$\text{NDII} = \frac{\texttt{B8} - \texttt{B11}}{\texttt{B8} + \texttt{B11}} \tag{8.3}$$

- the normalized difference leaf mass area (ND_{LMA}) [le Maire et al., 2008]:

$$\text{ND}_{\text{LMA}} = \frac{\texttt{B12} - \texttt{B11}}{\texttt{B12} + \texttt{B11}} \tag{8.4}$$

- the leaf area index soil adjusted vegetation index (LAI_{SAVI}) [Bulcock and Jewitt, 2010; Huete, 1988]:

$$\text{LAI}_{\text{SAVI}} = \frac{\log\left(\left|0.371 + 1.5 \times \frac{\texttt{B8}-\texttt{B4}}{\texttt{B8}+\texttt{B4}+0.5}\right|\right)}{2.4} \tag{8.5}$$

These spectral indices are proposed in the literature as quantities that correlate well with the leaf area index (LAI) and chlorophyll content. Therefore, they are relevant input features for the encoder of PROSAIL-VAE.

Finally, the input features are transformed before being input to the encoder. Rather than the raw Sun / S2 angles, the *min-max normalization* (see Equation 8.6) of cosines of these angles are input to the encoder. This normalization is identical to that of BVNET and SNAP (see subsection 5.2.1), and uses the same min and max values for each angle.

$$x_{\text{norm}} = \frac{x - \min_x}{\max_x - \min_x} \tag{8.6}$$

The spectral bands and indices are transformed using *quantile normalization*:

$$x_{\text{norm}} = \frac{x - q_{0.5}}{q_{0.95} - q_{0.05}}, \tag{8.7}$$

with q_α the quantile α. The values of these quantiles are computed from the pixel values of the training S2 data-set $\mathcal{D}_{\text{S2}}$. This ensures that the distributions of the normalized spectral bands and indices have the same dynamic. In particular, the MCRL reconstruction loss is applied to the normalized spectral bands, rather than the raw bands, so that each spectral dimension is penalized equivalently.

The base configuration for PROSAIL-VAE is summarized in Table 8.2. A model trained with this configuration is assessed in section 8.2, and variants are discussed in section 8.3.

Table 8.2: Base configuration for PROSAIL-VAE

Category	Element	Value
Architecture	Encoder	Pixel-wise
	Encoder input features	B2, B3, B4, B5, B6, B7, B8, B8A, B11, B12, NDVI, NDII, ND_{LMA}, LAI_{SAVI}, $\cos\theta_S$, $\cos\theta_O$ and $\cos\psi_{SO}$
	Decoder	PROSPECT-5+4SAIL
	Latent distribution	Truncated Normal
Objective function	loss	$\mathcal{L}_{\text{rec}} + \beta\mathcal{L}_{\text{KLD}}$
	Reconstruction Loss	MCRL
	S2 band penalization in reconstruction loss	B3, B4, B5, B6, B7, B8, B8A B11, B12 (all except B2)
	Number of MC samples	70
	KLD coefficient	$\beta = 2$
	Prior distribution $p(\boldsymbol{z})$	Uniform
	Variable regularized with the prior with $\mathcal{L}_{\text{KLD}}$	LAI
Initialization (MIBIT)	Number of initialized models	10
	Number of epochs	10
	lr	10^{-3}
lr schedule	lr	10^{-3}
	lr scheduler	CPR
	Scheduler patience	5 epochs
	lr_{min}	10^{-8}
	lr decay factor	10
	Scheduler patience	5 epochs
Training	Optimizer	Adam
	Batch size	1
	Patch size	32 x 32
	Epochs	300
	Batches per epoch	50

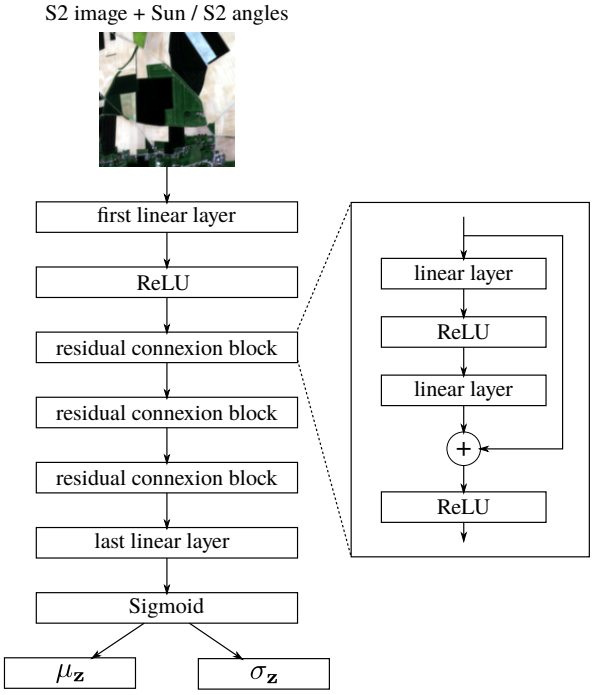


Figure 8.2: Backbone of the encoder of PROSAIL-VAE, with a first input layer followed by 3 blocks of 2 layers with skip connections, and a last layer that outputs the parameters of the distribution of PROSAIL variables. For the pixel-wise version of the encoder, all linear layers are dense layers of size 32. For the spatial version, all linear layers are 2D convolutional layers with size 32 and stride 1. The first layer has a filter size of 3×3, whereas the size of the rest of the layers is 1×1.

8.1.2 A related approach

A similar prior approach to PROSAIL-VAE was proposed in Svendsen et al. [2021]. In it, two different variational inference strategies relying on Monte Carlo Expectation Maximization (MCEM) and VAE are developed to retrieve three BV from Landsat-8[1]. Analogously to PROSAIL-VAE, their latter approach integrated PROSAIL as the decoder of a VAE, so that the latent variables are semantically bound to the PROSAIL variables.

However, their work presents several crucial differences with that of PROSAIL-VAE. Their experimental setup is more limited than that of PROSAIL-VAE. They only used simulated data for their experiments, both for training and evaluating their approach. They predicted three PROSPECT parameters (i.e. leaf parameters): C_{ab}, C_w and C_m. The rest of the necessary PROSAIL variables were set constant throughout the training data-set. In their approach, the latent space was three-dimensional which means that there were fewer latent variables than reconstructed spectral bands, contrary to PROSAIL-VAE.

There are also key differences in the implementation of the VAE model. They neither used a differentiable PROSAIL model nor a differentiable emulator, so they had to estimate gradients of the model without relying on automatic differentiation. They approximated the gradients by using finite differences. For the variational distributions, they used multivariate Gaussians. Since these distributions are unbounded, they ensure that the sampled PROSAIL parameters remain meaningful by mapping negative parameters to 0. They note that this modifies the likelihood and introduces multimodality, since 0 becomes a mode because of this mapping. Their likelihood model $p(\boldsymbol{x}|\boldsymbol{z})$ is Gaussian, however they do not estimate the variance and pre-set it to a constant $\sigma^2 = 10^{-7}$.

[1]Landsat-8 is an optical remote sensing satellite launched by NASA in 2013. With an open data policy, the Landsat-8 images are similar to S2 images: they use nine bands (with two for atmospheric correction) with similar spectral sensitivity to some S2 bands, with 15 m and 30 m spatial resolution. Landsat-8 has 16 days revisit frequency.

Finally, the prior $p(\boldsymbol{z})$ used in the KLD loss term in their model is different from the uniform prior used in PROSAIL-VAE. They use a multivariate Gaussian prior, whose parameters (mean vector and covariance matrix) are learned, instead of pre-set. This means that the prior distribution is first initialized, then updated throughout the training, to reflect a property of the training data-set rather than an arbitrary belief. Each new observation alters the prior and enables to infer more informed posterior distributions. After training, their learned prior approximated the distribution of PROSAIL parameters of their simulated training data-set, i.e. the mean and covariance of the prior matched that of their simulations. This is an interesting approach because it doesn't require much prior knowledge about the data. A comparable prior is investigated for PROSAIL-VAE in subsubsection 8.3.1.2. Since the prior is multivariate, the correlations between the variables in the data-set are estimated through the covariance matrix. In essence, this approach derives the prior as the distribution which minimizes the KLD loss term for all inferred posteriors $q_{\boldsymbol{z}|\boldsymbol{x}}$ over the whole data-set. This definition of the prior is not agnostic to the data: the prior approximates the training data distribution and may not suit data with a different distribution.

Despite the similarity with their approach, PROSAIL-VAE was developed independently, and is more oriented toward a practical application, since it was designed to learn on real data from the start.

8.1.3 Supervised regression strategies

Besides the proposed self-supervised PROSAIL-VAE, two supervised deep learning regression methods are also studied for the inversion of PROSAIL.

SNAP The first strategy is the well-known Biophysical Processor (BP) tool in SNAP [Weiss and Baret, 2016] which uses a BVNET neural network (see subsection 5.2.1). The canonical weights of the operational SNAP model are used and no BVNET is trained for the experiments of this chapter.

MPSR The second approach is the so-called MPSR. It is a supervised deep learning model that uses the architecture of the probabilistic encoder of subsection 8.1.1 (see Figure 8.2). Such models are trained using pre-simulated data-sets generated with PROSAIL. The training data-set for MPSR is simulated with the procedure detailed in section 5.1. The PROSAIL variables are first sampled with chosen distributions, then correlations are applied between each variable and the LAI using a co-distribution function, and these variables are then forwarded to PROSAIL which simulates the corresponding canopy reflectance spectra. For MPSR, 3×10^5 training samples are generated, using the variables distributions of Table 5.1, and using the co-distribution type 2 (see Equation 5.4). This number of simulated pixels is lower than the number 2.4×10^7 real S2 pixels for training PROSAIL-VAE (see Table 2.3), but much higher than the 4×10^4 used for SNAP. The pixel-wise architecture of the encoder is considered rather than the spatial one because PROSAIL doesn't generate images, but individual pixels.

Like the encoder of PROSAIL-VAE, MPSR doesn't output estimates of the PROSAIL variables, but parameters $\boldsymbol{\lambda}$ of distributions that are, up to an affine transformation, the distribution of those variables. Like PROSAIL-VAE, the parameters of independent TN $\boldsymbol{\lambda} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$ and are estimated.

Given sampled PROSAIL variables $\boldsymbol{v}$, associated to simulated S2 reflectance bands $\boldsymbol{x}$, the corresponding latent variables $\boldsymbol{z}$ are derived using the inverse affine transformation to that used for scaling the latent variables in PROSAIL-VAE. The objective function of MPSR is the NLL of the estimated TN distribution w.r.t. the latent variable $\boldsymbol{z}$ (see subsection C.4.3). Like PROSAIL-VAE, MPSR predicts an approximate posterior distribution $q(\boldsymbol{z}|\boldsymbol{x})$. Both methods can be seen as amortized (see subsection 6.4.1) maximum likelihood estimation (MLE) estimation methods (see subsection 6.1.1). Both methods use a NLL as

objective function (the reconstruction loss for PROSAIL-VAE). However, the optimization is not applied to the same element: for MPSR, the predicted distribution is directly optimized whereas for PROSAIL-VAE the NLL is applied on the reconstruction as a proxy.

Compared to BVNET, this strategy has two advantages:

- More complex relationships can be discovered since MPSR is a deeper, more complex neural network.
- Multiple PROSAIL variables are retrieved simultaneously by a single model, while BVNET uses a different network for each variable.
- The predictions of the model are probabilistic, so they can quantify uncertainties.

The training configuration of MPSR (see Table 8.3) is similar to that of the base PROSAIL-VAE (see Table 8.2). MPSR is a pixellic approach trained on individual simulated pixels, rather than patches. Thus, a size of 1024 is used for pixel batch size as an equivalent of the 32×32 patches in PROSAIL-VAE. The MPSR approach is not computationally intensive like PROSAIL-VAE, because there is no simulation with PROSAIL involved at training time, which enables to iterate through the optimizations steps faster.

Table 8.3: Configuration of MPSR.

Category	Element	Value
Architecture	Neural network	Pixel-wise encoder
	Input features	`B2`, `B3`, `B4`, `B5`, `B6`, `B7`, `B8`, `B8A`, `B11`, `B12`, NDVI, NDII, ND_{LMA}, LAI_{SAVI}, $\cos\theta_S$, $\cos\theta_O$ and $\cos\psi_{SO}$
	Output	TN distribution parameters
Objective function	loss	TN NLL
Initialization (MIBIT)	Number of initialized models	10
	Number of epochs	10
	lr	10^{-3}
lr schedule	lr	10^{-3}
	lr scheduler	CPR
	Scheduler patience	5 epochs
	lr_{min}	10^{-8}
	lr decay factor	10
	Scheduler patience	5 epochs
Training	Optimizer	Adam
	Batch size	1024
	Input data	S2 pixel
	Epochs	5000

8.1.4 Variable estimates and prediction intervals

PROSAIL-VAE and MPSR output TN distribution parameters. For each variable characterized by such a distribution, an estimate and a prediction interval are derived. For a given variable, the estimate is taken as the expectation m of the TN distribution. This enables to easily interpolate these estimations to the dates of measurement of in-situ data from the two closest S2 images, with only the assumption of independence of predicted distributions (see subsection 2.4.3). The interpolation of expected values is the expected interpolated value

(as opposed to using the mode or median). The uncertainty for each prediction is characterized by the standard deviation (std), which is derived from the distribution parameter, and interpolated if needed (see subsection 2.4.3). The prediction intervals are derived as 2σ intervals: $[m - 2\sigma, m + 2\sigma]$, with σ the std.

For variables that are inferred by PROSAIL-VAE or MPSR (e.g. the LAI), the expected value m and std σ are directly derived from the inferred TN. For canopy chlorophyll content (CCC), the predicted value is obtained by assuming the independence of LAI and CCC predictions[2]:

$$\mathrm{CCC} = m_{\mathrm{LAI}} \times m_{C_{ab}}. \tag{8.8}$$

The estimated prediction intervals of CCC are derived from the variance of the product of LAI and C_{ab}, by assuming that they were not correlated [Goodman, 1960]:

$$\mathrm{var}\,\mathrm{CCC} = \mathrm{var}\,(\mathrm{LAI} \times C_{ab}) = (\mathrm{var}\,(\mathrm{LAI}) + m_{\mathrm{LAI}}^2)\left(\mathrm{var}\,(C_{ab}) + m_{C_{ab}}^2\right) - m_{\mathrm{LAI}}^2 m_{C_{ab}}^2. \tag{8.9}$$

8.2 Performances of PROSAIL-VAE

In this section, the performances of a single PROSAIL-VAE model, with the base configuration (see Table 8.2) are analyzed.

The training of this model is discussed in subsection 8.2.1. In subsection 8.2.2, its accuracy in retrieving the LAI, CCC and leaf chlorophyll content (LCC) from S2 images is compared to that of SNAP and MPSR, by using in-situ validation data. Using the testing S2 data-set $\mathcal{D}_{\mathrm{S2,test}}$, the reconstruction of input data is discussed in subsection 8.2.2, and the inference of PROSAIL variables is discussed in subsection 8.2.3.

PROSAIL-VAE refers to a class of models, therefore PROSAIL-VAE models are instances of this class (sometimes with different configurations), but with different training results. The presented PROSAIL-VAE model was chosen as one instance with the best in-situ validation performance, and it will serve as a reference for assessing the different design choices for PROSAIL-VAE later in this chapter. It will be denoted PV*.

8.2.1 Training

The evolution of the training and validation losses for PROSAIL-VAE, along with the variation of the lr are shown in Figure 8.3. PROSAIL-VAE shows no sign of overfitting, since the total training and validation loss ($\mathcal{L}_{\mathrm{sum}}$) both decrease at the same rate throughout training.

Thanks to the MIBIT initialization scheme, the optimization of PROSAIL-VAE was given a head-start prior the 300 epochs displayed here. It enabled the loss to be decreased significantly so the subsequent training could be more efficient. Most of the loss decrease occurs within the 50 first epochs, and converged slowly afterwards. The CPR lr scheduler decreased the lr when the validation loss no longer progressed. Variations of the lr coincided with small improvement of the validation loss. Most of the decrease in the validation loss occurred before the lr had changed once, and the scheduled change in lr seems to have help decreasing it a little further.

One specificity about the reconstruction loss $\mathcal{L}_{\mathrm{rec}}$ is that it can be negative. This loss is a NLL, which can be negative when the term $\left(\frac{x-\mu}{\sigma}\right)^2$ becomes small relative to $\log \sigma^2$. This occurs when both $x - \mu$ and σ are small, i.e. when the predicted distribution is narrow and well centered around the target value x. This is a sign that PROSAIL-VAE reconstructs the input S2 bands well.

The KLD loss term varies in reverse to the reconstruction loss, showing a competing effect. This term is a regularizing term, and since it is the KLD between the TN posterior and a

[2]These distributions are independent by design in PROSAIL-VAE and MPSR, since they are sampled independently. However their parameters are derived by a deterministic function of the same input data, and they represent quantities that are correlated in reality. Nonetheless, this assumption is necessary since the correlation between the LAI and CCC is not estimated.

uniform distribution, it promotes latent distributions with larger variance. Conversely, the reconstruction loss term is improved when the variance of reconstruction is low, and promotes low variance latent distributions as a consequence.

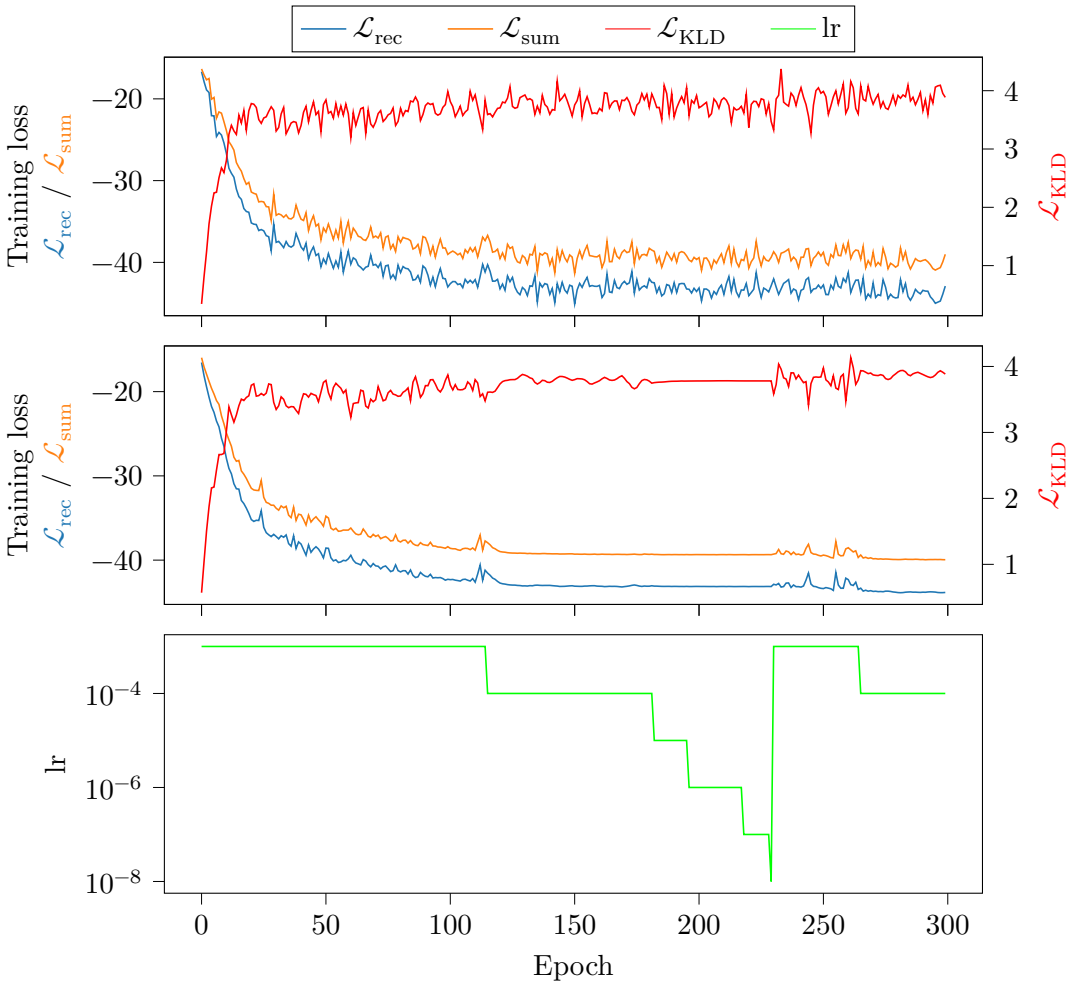


Figure 8.3: Training and validation losses and learning rate of the PV* PROSAIL-VAE model.

It can be observed that the performance of PROSAIL-VAE on retrieving PROSAIL variables varies during the training. In particular, validation with the in-situ data of the LAI retrieval with the in-situ dataset $\mathcal{D}$ is performed at regular intervals during training. The evolution of the root mean squared error (RMSE) of the LAI and CCC retrieval for another instance of PROSAIL-VAE is shown in Figure 8.4. It can be observed that the best LAI and CCC RMSE do not occur simultaneously during training, and do not coincide with a minimum of the validation loss. In particular, the best LAI performance is obtained at the beginning of training, while the total loss is far from convergence.

There are several phenomena at stake during training. PROSAIL-VAE retrieves simultaneously all PROSAIL variables, and as observed, the optima of performance for each variable are reached at different times during training. In the present case, the optimum for LAI was achieved before that of the C_{ab}. Additionally, certain variables can have a competing effect in the simulation of S2 bands by PROSAIL, i.e. on the reconstruction of S2 images. As such, a variable whose optimum was reached first may have its performance decline later in training when another competing variable is optimized. Besides, the discrepancy between the validation loss and the in-situ validation performances has multiple causes:

- First and foremost, the training task (i.e. a regularized reconstruction of an input S2 image) is a proxy task, which is different from the variable retrieval evaluation taken

as the downstream task.

- Second, the images involved during training are different from the images of the measurement sites. This was done on purpose to avoid a bias of the model toward the measurement data. However this also means that a performance on the training data, may not be exactly mirrored by performance on images of the measurements sites.

- Third, the in-situ data is quite limited in quantity and diversity, even more compared to the diversity of the S2 used for training. Therefore, the measurement data is not representative enough of the vegetation found the training images. As such, in-situ validation performance varying during training may be a consequence of the model generalizing to a greater diversity of vegetation than what is found in in-situ data.

Finally, this introduces a limitation in the present study. The validation loss cannot be used as a metric for selecting the PROSAIL-VAE model with the best performance among other. This is simply because models with a similar loss may have different variable retrieval performances on the in-situ data.

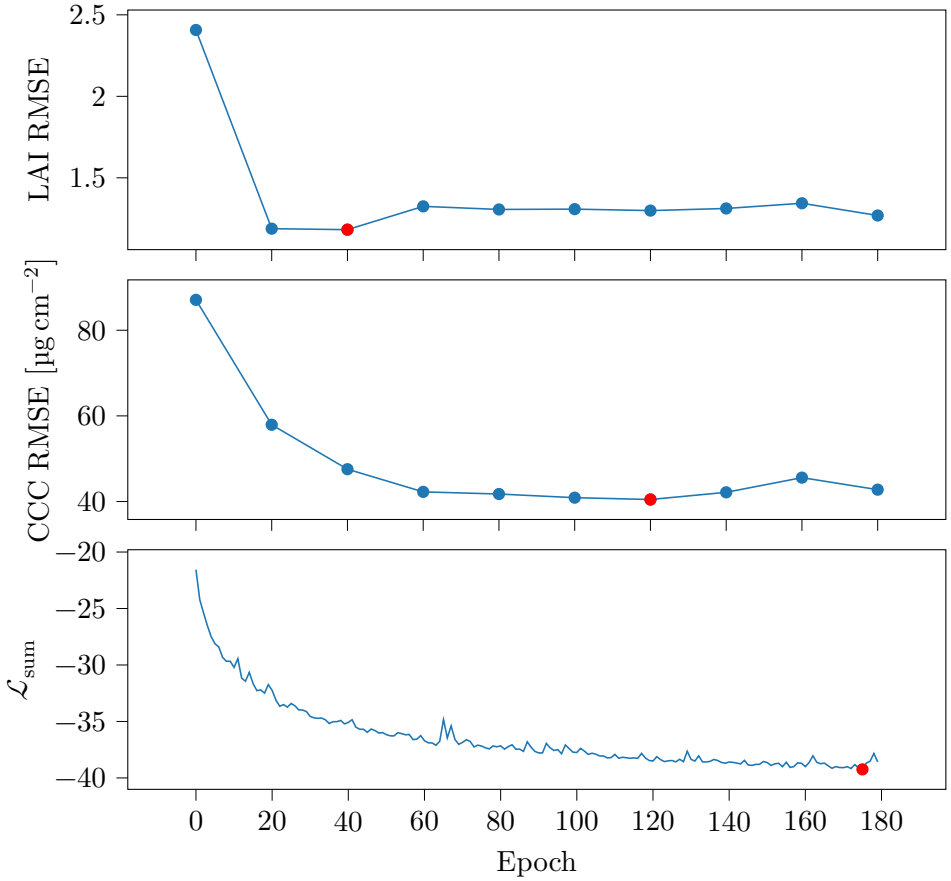


Figure 8.4: Evolution of the RMSE of the LAI and CCC on in-situ data, and of the validation loss during the training of a PROSAIL-VAE model. The red dot on each curve marks the lowest (best) value reached by each metric.

8.2.2 Validation on in-situ data

The performances of the PV* are compared against that of the MPSR and pre-trained SNAP, using the in-situ validation data $\mathcal{D}_{\text{IS}}$. These methods are compared w.r.t. the estimation of LAI, CCC and also *effective* LAI (LAI_{eff}) and *effective* CCC (CCC_{eff}) since the fiducial

reference measurements for vegetation (FRM4Veg) campaign include those measurements (see subsection 2.4.1)

Accuracy of variable retrieval As SNAP is a deterministic approach, only variables estimates can be compared with this method, with regression metrics like the RMSE (Table 8.4 and Table 8.5) and the coefficient of determination (R^2) (Table 8.6 and Table 8.7) metrics (subsection 3.1.5).

Table 8.4: RMSE of the LAI and CCC on in-situ data-sets for PV*, SNAP and MPSR.

Variable	LAI			CCC		
Model	PV*	SNAP	MPSR	PV*	SNAP	MPSR
BelSAR	1.30	**1.22**	1.26			
Barrax 2018	**1.42**	1.43	1.99	**27.60**	83.92	30.35
Barrax 2021	0.72	**0.48**	0.76	**20.51**	84.53	30.31
Wytham 2018	**1.21**	1.77	1.42	**80.78**	101.35	85.77
All	**1.16**	1.24	1.35	**40.06**	89.33	46.53

Table 8.5: RMSE of LAI_{eff} and CCC_{eff} on in-situ data-sets for PV*, SNAP and MPSR.

Variable	LAI_{eff}			CCC_{eff}		
Model	PV*	SNAP	MPSR	PV*	SNAP	MPSR
Barrax 2018	0.77	**0.71**	1.10	**17.45**	112.87	43.05
Barrax 2021	1.16	1.06	**0.56**	**38.85**	125.65	71.81
Wytham 2018	1.82	**0.94**	1.31	**71.85**	135.39	106.99
All	1.25	**0.93**	**0.93**	**42.66**	125.02	74.24

Table 8.6: R^2 of LAI and CCC on in-situ data-sets for PV*, SNAP and MPSR.

Variable	LAI			CCC		
Model	PV*	SNAP	MPSR	PV*	SNAP	MPSR
BelSAR	0.24	**0.33**	0.28			
Barrax 2018	**0.78**	0.77	0.56	**0.91**	0.20	0.90
Barrax 2021	0.86	**0.94**	0.84	**0.94**	-0.08	0.86
Wytham 2018	**0.02**	-1.09	-0.34	**0.08**	-0.44	-0.03
All	**0.75**	0.71	0.66	**0.82**	0.22	0.78

PV* obtained slightly better overall RMSE and R^2 metrics for the LAI than SNAP, and much better for the CCC. The MPSR was worse for the LAI than both other methods, and a little worse than PV* on the CCC. This method did not outperform the others on any retrieved variable in any campaign.

Figure 8.5 corroborates these results by showing the individual predictions of PV* and SNAP. For the LAI, similar predictions were obtained by PROSAIL-VAE and SNAP. For instance, the LAI on the poppy class of the 2018 campaign was underestimated by both methods, although the underestimation was slightly lower for PROSAIL-VAE. Both methods predicted a limited range of LAI values for the Wytham site. PROSAIL-VAE performed better than SNAP. Such an underestimation of the LAI by SNAP on heterogeneous forest canopies is corroborated in Brown et al. [2021b]; Xie et al. [2019]. Besides, PROSAIL-VAE slightly overestimated the prediction of low LAI values. For the CCC, SNAP has an

Table 8.7: R^2 of LAI_{eff} and CCC_{eff} on in-situ data-sets for PV*, SNAP and MPSR.

Variable	LAI_{eff}			CCC_{eff}		
Model	PV*	SNAP	MPSR	PV*	SNAP	MPSR
Barrax 2018	0.88	**0.90**	0.76	**0.93**	-1.52	0.63
Barrax 2021	-0.26	-0.05	**0.71**	**0.26**	-7.08	-1.64
Wytham 2018	-6.71	**-1.05**	-2.98	**-1.47**	-7.80	-4.49
All	0.44	**0.66**	0.62	**0.56**	-2.69	-0.30

overestimation problem. PROSAIL-VAE on the other hand performs much better. However it has a tendency to predict a constant CCC value for forests pixels, and that leads to underestimating a part of these values.

As for effective variables, the LAI_{eff} is better correlated to SNAP and MPSR than PV*, contrary to CCC_{eff} which is better correlated to PV*. The LAI_{eff} is the product of the LAI with the canopy clumping index which is lower than 1. As such, the LAI_{eff} and CCC_{eff} have a lower value than their LAI and CCC counterparts. As a consequence, by comparing predictions to effective variables, underestimation may be mitigated, whereas overestimation is amplified. This explains why using in-situ effective values as reference worsens the accuracy of SNAP on CCC and PV* on LAI. It can be noted that there are PROSAIL-VAE models other than PV* which are better correlated to the LAI_{eff} than even SNAP – but are worse on the LAI on the other hand.

Since FRM4Veg also provides LCC measurements, the retrieval of this variable is also evaluated. A comparison of the predictions of LCC is provided in Figure 8.7. This comparison is detrimental to SNAP: since it doesn't output LCC, estimates are computed from SNAP predictions of LAI and CCC, as LCC = CCC/LAI. Nonetheless, for PV*, the prediction of LCC is rather accurate, except for a few outliers, such as the garlic class. The LCC predictions with the worst accuracy by PROSAIL-VAE are underestimations. As will be discussed in subsubsection 8.2.3.3 with Figure 8.13, the experimental upper bound on predicted C_{ab} with PV* is around $45\,\mu g\,cm^{-2}$. Thus, the retrieval performance of LCC with PV* is probably limited, and worse performance would be observed with more in-situ data with higher LCC values. It is also worth noting that PV* can be limited in the prediction of lower values of LCC, because the allowed lower bound for C_{ab} is $20\,\mu g\,cm^{-2}$ (see Table 8.1). Consequently, predicting a near-zero C_{ab} for non-vegetated pixels (e.g. bare soil) is not possible for PV*. In this configuration, the CCC variable is better for PV*, since in can have a value of zero thanks to the lower bound of the LAI being zero.

The temporal evolution of the LAI predictions of both methods is also studied in Figure 8.6. This figure displays the LAI time series predictions obtained over a maize parcel belonging to the BelSAR site. Both predicted time series show a well-defined summer crop phenology curve, that can be fitted with a double-logistic model (see Chapter 9). As observed, similar predictions for both PROSAIL-VAE and SNAP closely matched the maize in-situ measurements.

Uncertainty quantification with prediction intervals The mean prediction interval width (MPIW) and prediction interval coverage probability (PICP) metrics are derived for the prediction interval produced by PV* and MPIW, and shown in Table 8.8 and Table 8.9. The width of the prediction intervals of both MPSR and PV*, as measured by the MPIW are wide enough that the frequency of them containing the in-situ reference value (PICP) is good. Since the prediction intervals are derived as 2σ intervals, the target for the PICP is approximately 0.95[3]. The PICP reached by PV* and MPSR are close to this target. This

[3]This is known as the *empirical rule*, (or three-sigma rule or 68-95-99.7 rule), which defines the frequency at which some observed data falls within n std of the mean of a normally distributed data. A 1σ intervals contains

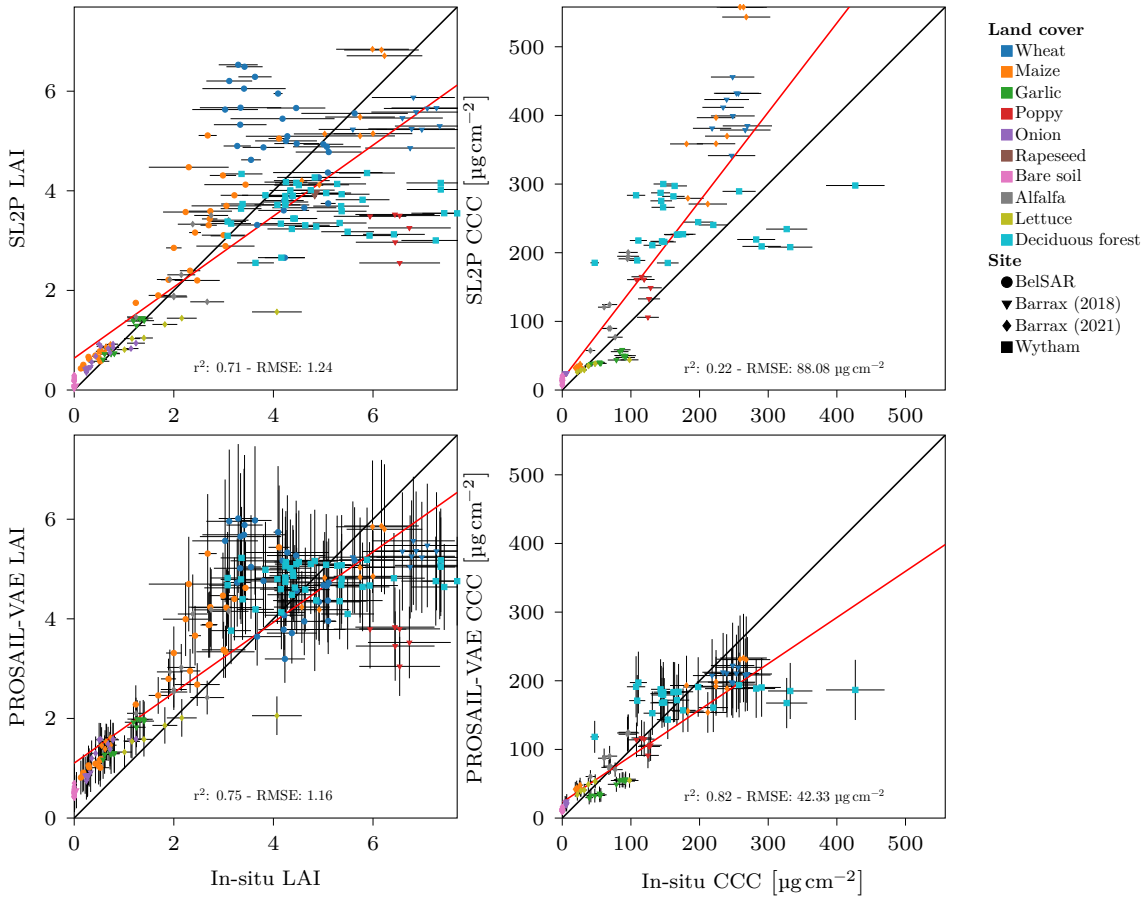


Figure 8.5: Scatter-plots of LAI and CCC predictions from SNAP and PV* PROSAIL-VAE versus in-situ test sites measurements. For each data point, the horizontal black lines correspond to the in-situ uncertainty measures. The vertical black lines indicate PV* 2σ prediction intervals, derived from the inferred PROSAIL variable distributions.

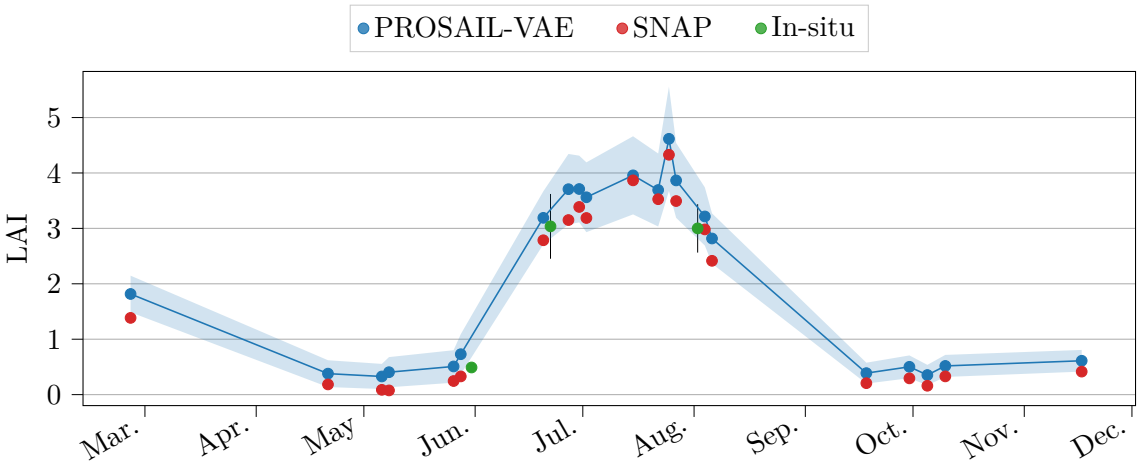


Figure 8.6: LAI time series predictions obtained over a maize parcel belonging to the BelSAR site. PV* PROSAIL-VAE and SNAP predictions are obtained by considering non-cloudy S2 available images acquired on 2018. The blue area is the LAI std predicted by PROSAIL-VAE and the blue line is the interpolated expectation. The vertical black lines are the measurement std of in-situ data.

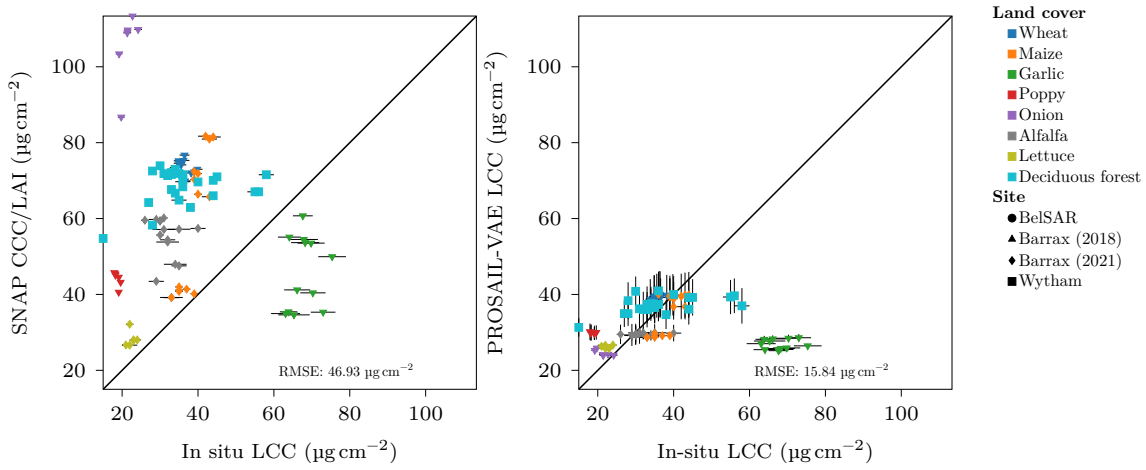


Figure 8.7: Scatter-plots of LCC predictions from SNAP and PV* PROSAIL-VAE versus in-situ test sites measurements. For PROSAIL-VAE, LCC is taken as the infered C_{ab} value, whereas for SNAP it is computed as LCC = CCC/LAI. For each data point, the horizontal black lines correspond to the in-situ uncertainty measures. In contrast, the vertical black lines indicate PROSAIL-VAE 2σ prediction intervals, derived from the inferred PROSAIL variable distributions.

suggests that the uncertainty quantified by prediction intervals is well suited to the level of errors for both methods. For each campaign, the prediction intervals are the same between LAI_{eff} and LAI, and between CCC_{eff} and CCC because they are derived from the same inferred distribution, it is only the reference measurement that changes.

For the LAI, the MPSR achieves similar PICP to PV* while having lower MPIW and despite a larger RMSE. This suggests that MPSR produced better prediction intervals, that quantify uncertainty a little more accurately than PV*. For the CCC however, the prediction intervals are much larger for MPSR, and it leads to overestimating uncertainty, with PICP that overshoot the 0.95 target.

Table 8.8: MPIW of LAI, LAI_{eff}, CCC and CCC_{eff} on in-situ data-sets for PV* and MPSR.

Variable	LAI		LAI_{eff}		CCC		CCC_{eff}	
Model	PV*	MPSR	PV*	MPSR	PV*	MPSR	PV*	MPSR
BelSAR	4.74	4.37						
Barrax 2018	3.74	2.93	3.74	2.93	140.53	177.80	140.53	177.80
Barrax 2021	2.72	2.06	2.72	2.06	94.02	127.17	94.02	127.17
Wytham 2018	5.45	6.76	5.45	6.77	235.20	406.36	235.20	406.36
All	4.04	3.83	3.80	3.67	147.96	222.56	147.96	222.56

8.2.3 Inference on testing data-set

The in-situ validation data $\mathcal{D}_{\text{IS}}$ enabled to quantitatively assess the performance on several vegetation variables for PV*. Using the testing S2 data-set $\mathcal{D}_{\text{S2}}$, the next paragraphs further characterize this model.

about 68% of the data, a 2σ 95% and 3σ 99.7%. This rule is applied here to derive a 95% PICP target, by assuming that the TN distributions can be assimilated to Gaussians (which is a reasonable approximation, except for distributions that are truncated close to the mode).

Table 8.9: PICP of LAI, LAI_{eff}, CCC and CCC_{eff} on in-situ data-sets for $PV^{\star}$ and MPSR.

Variable	LAI		LAI_{eff}		CCC		CCC_{eff}	
Model	$PV^{\star}$	MPSR	$PV^{\star}$	MPSR	$PV^{\star}$	MPSR	$PV^{\star}$	MPSR
BelSAR	**1.0**	**1.0**						
Barrax 2018	**0.88**	0.83	**1.0**	0.86	**0.95**	0.97	**1.0**	**1.0**
Barrax 2021	**0.96**	0.94	0.87	**0.96**	**0.98**	**0.98**	**0.96**	1.0
Wytham 2018	**0.95**	0.98	**0.97**	1.0	0.84	**1.0**	**0.96**	**1.0**
All	**0.95**	0.94	0.93	**0.94**	**0.93**	0.98	**0.98**	1.0

8.2.3.1 Reconstructions

The main driver of PROSAIL-VAE variable retrieval performance is arguably the penalization of reconstructions with MCRL as a proxy task. As such, since the objective of this model is to provide the closest reconstructions possible to real S2 images, it is important to assess how well this task is performed.

Figure 8.8 shows the reconstruction performances obtained by $PV^{\star}$ on the testing part of the S2 data-set $\mathcal{D}_{S2}$. The scatter plots compare original S2 band reflectance values against reflectances reconstructed by $PV^{\star}$. Overall, the reconstructions produced with PROSAIL-VAE match the original spectral S2 bands. All reconstructed bands have a R^2 score greater than 0.9, except for `B2`, which has a lower $R^2 = 0.65$. This is a consequence of not penalizing the `B2` band with the reconstruction loss.

The assessment of the reconstruction can also be done with individual patches. For instance Figure 8.9, illustrates visible and infra-red color compositions and their corresponding reconstructions. These visual results corroborate the accurate reconstruction of crops areas for both color compositions.

8.2.3.2 Prediction of PROSAIL variables on S2 patches

Besides reconstructions, it is also possible to produce patches of inferred PROSAIL variables. In Figure 8.10, the predictions of SNAP and $PV^{\star}$ are compared over the patch shown in Figure 8.9. For both methods, the LAI, CCC and canopy water content (CWC) are well correlated to the presence of vegetation.. The predictions made by $PV^{\star}$ look sharper than the prediction by SNAP, and some structures, such as roads and the shapes of the parcels, are better outlined by $PV^{\star}$. Within the parcels, the predictions of $PV^{\star}$ seem more homogeneous. The CCC and CWC tend to be predicted with higher values within the parcels. For LAI, CCC and CWC, the stds are correlated to high expected values.

In Figure 8.11 are shown the expectations and std of PROSAIL variables, besides the LAI, predicted by $PV^{\star}$ from the example patch of Figure 8.9. This S2 image patch contains both crop vegetation elements, and areas without vegetation: roads, buildings, bare soils. In areas devoid of vegetation, variables besides s_w and s_b are irrelevant, even though some of them exhibit high values (C_{car}, C_m, $\overline{\alpha}$, h). In particular, the soil wetness factor s_w is well correlated to the bare soil areas. For areas with vegetation, N, C_{ab}, C_{brown}, C_w seem correlated to the density of vegetation. Overall, the predicted std are correlated to the expected values.

8.2.3.3 PROSAIL variable distributions

Beyond sample patches, it is of interest to evaluate the distributions of inferred variables over the whole testing data-set $\mathcal{D}_{S2}$. In Figure 8.12, the predictions of the LAI, CCC and CWC variables over all the testing data-set of SNAP and $PV^{\star}$ are compared. A strong correlation was observed between LAI predictions, whereas a different behavior was obtained for CCC and CWC variables.

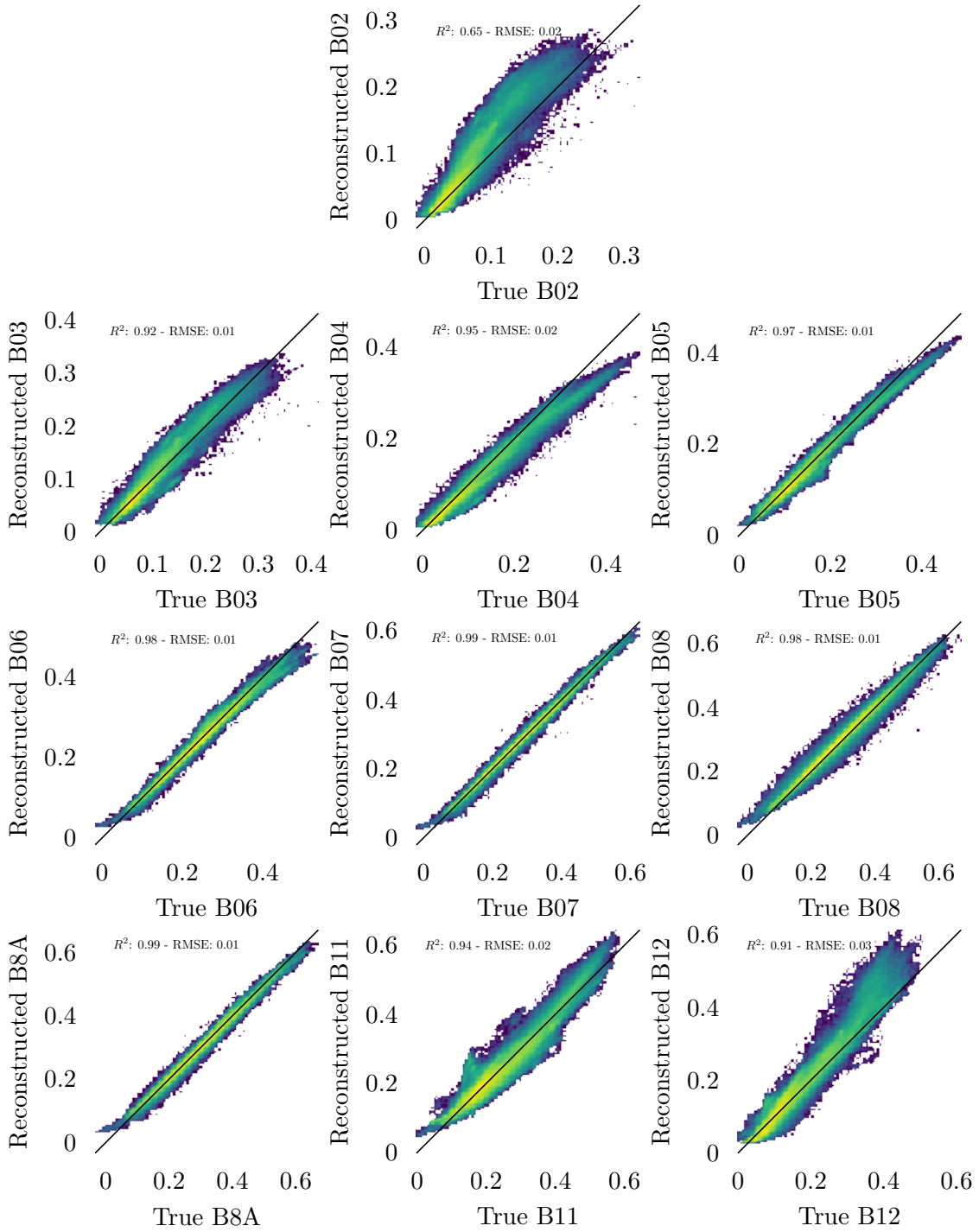


Figure 8.8: Scatter-plots comparing the original S2 band reflectance values against reflectances reconstructed by PV* on S2 test data-set.

Figure 8.9: S2 region of interest (ROI) image from test data-set acquired on 2023-05-13 and located at `T31UFS` tile (Southern Belgium). First column shows true and false color composites constructed by original S2 reflectance values whereas PROSAIL-VAE reconstruction results are displayed at the second column. The red green blue (RGB) true color composite corresponds to bands (`B4`, `B3`, `B2`). The RGB false color composite corresponds to bands (`B11`,`B8`,`B5`)

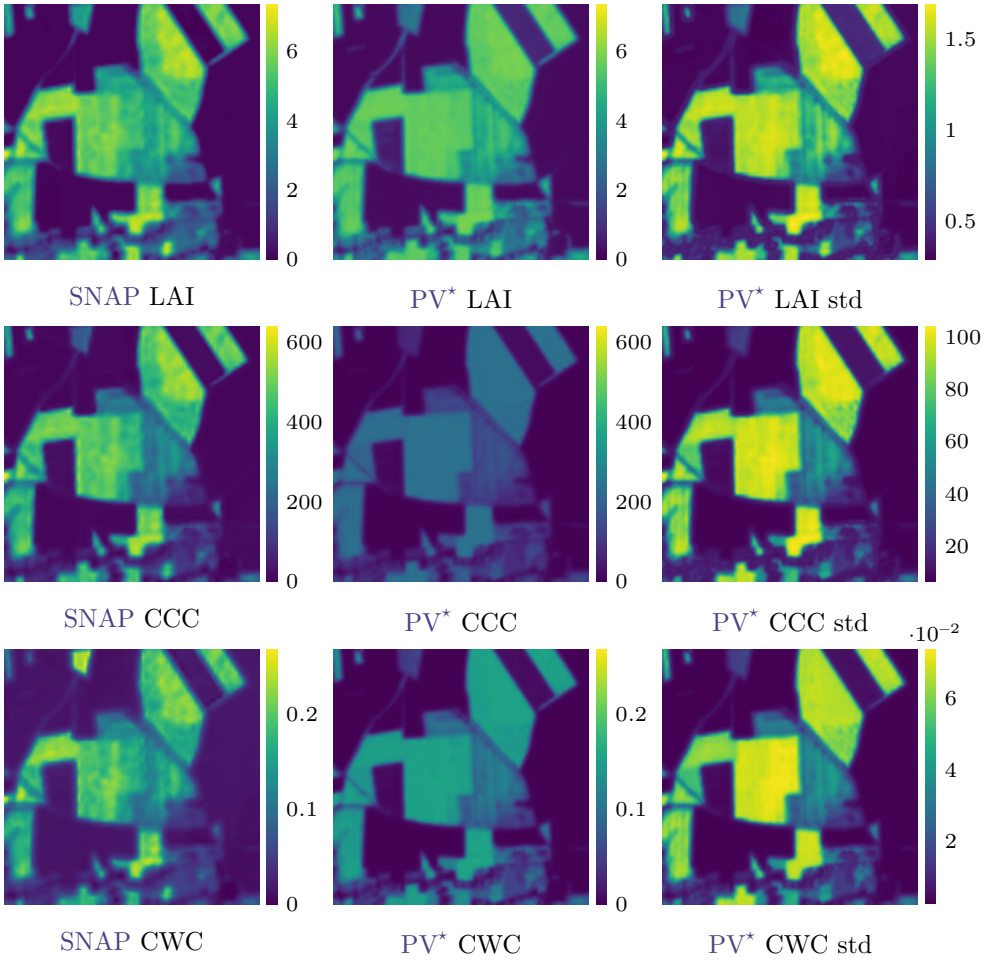


Figure 8.10: Comparison of LAI, CCC and CWC results obtained by SNAP and PV⋆. The image ROI is located at `T31UFS` tile (Southern Belgium) and acquired on 2018-06-01.

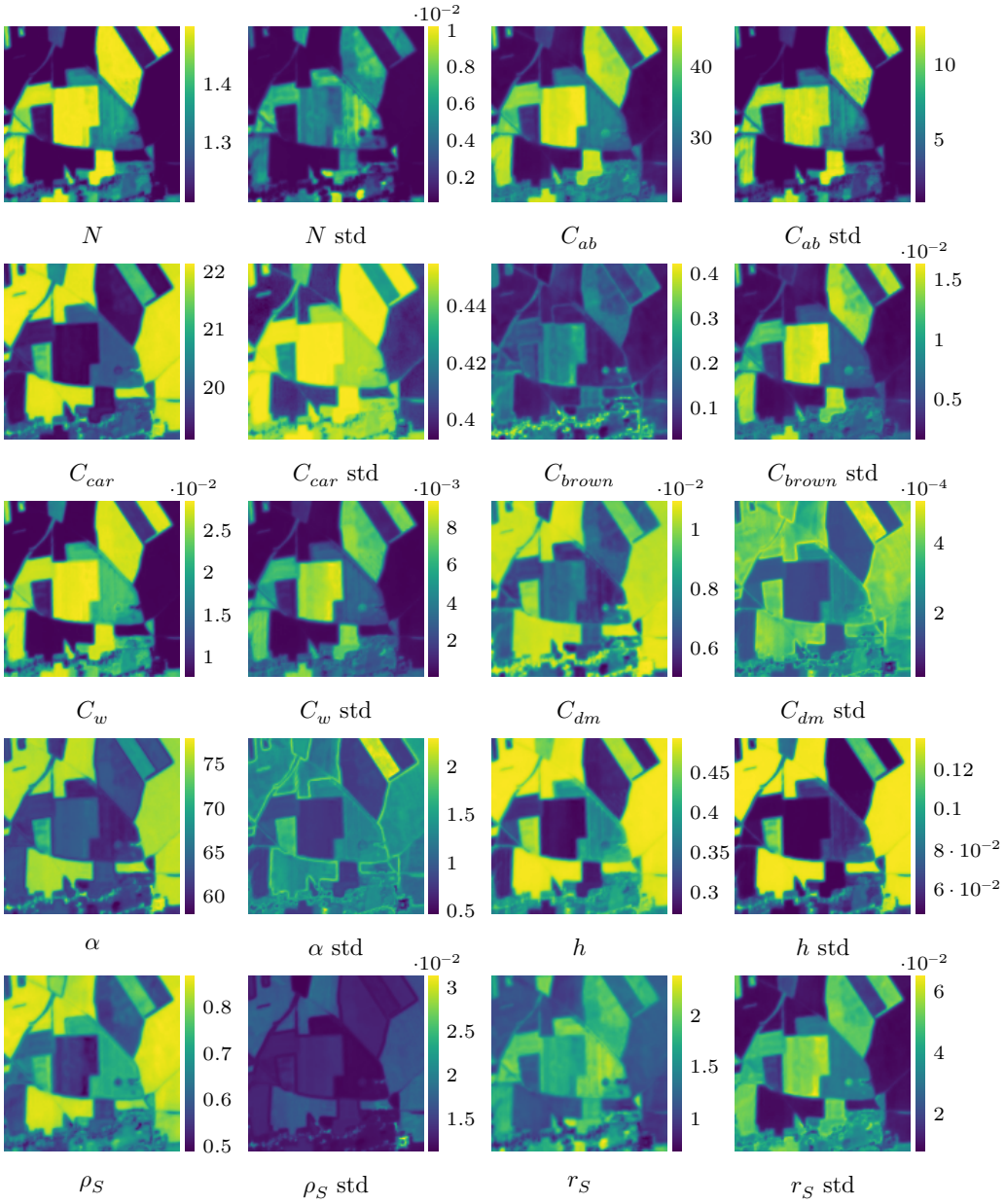


Figure 8.11: Inference of PROSAIL variables (mean and standard deviation) by PV*. BV are predicted at pixel level in a ROI of the Military Grid Reference System (MGRS) tile `T31UFS` (Southern Belgium) on 2023-06-01.

SNAP tended to predict higher CCC values than PV* which saturated at 250 µg cm^{-2}. In this case, the in-situ validation showed that SNAP tended to overestimate the CCC. The same behavior is observed by CWC where PV* predictions saturated at 0.17 cm. To assess these results, more in-situ measurements would be required to corroborate the quality of these two predicted variables. A visual comparison of the results can be found in Figure 8.10, where predicted biophysical variable maps are shown.

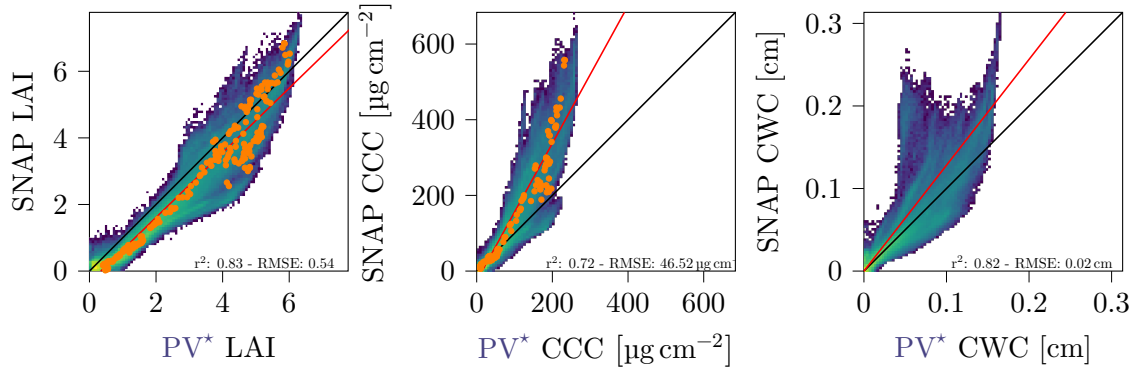


Figure 8.12: Scatter-plots of LAI, CCC, and CWC predictions from SNAP and PV* computed on the testing S2 image data-set $\mathcal{D}_{S2}$. The regression line is plotted in red and orange points correspond to predictions of in-situ measurements.

Despite PV* performing the inversion of all input PROSAIL parameters, not all predicted variables can be compared with SNAP results. Therefore, the distribution of the other variables as predicted by PV* are provided in Figure 8.13. This figure shows the histograms of expectation and std of PROSAIL variables predicted on the testing S2 image data-set. The histograms of the PROSAIL variable expectations are compared to the distribution of variables (see Table 5.1) used to generate training data-sets for BVNET (see subsection 5.2.2).

Figure 8.13 shows that the leaf parameter index N is likely to be poorly estimated. The histogram of the expectation of the leaf parameter N is concentrated on the lower bound of its definition interval at $N = 1.2$, which is associated with monocotyledon vegetation [Féret et al., 2021]. Unfortunately, it is well-known that vegetation with N values significantly higher than 1.3 occurs in real scenarios, and should be present in the images of the testing data-set.

Besides N, looking at the histograms of expected values of C_{ab}, C_{car}, C_{brown}, C_w, $\overline{\alpha}$, it can be observed how predictions do not occur over the full range of their definition intervals described in Table 8.1. For these variables, it is uncertain whether their predictions are flawed, or if they reflect the vegetation observed in the data-set. The low predicted C_{ab} values corroborate the CCC saturation effect observed in Figure 8.12, and the underestimation of in-situ LCC as shown in Figure 8.7. On the contrary, C_{car} is predicted within a small high value range. The chlorophyll and carotenoid pigments are both involved in photosynthesis, and it can be difficult to isolate their respective contribution to the leaf reflectance can be difficult. As such, the range restriction of C_{ab} and C_{car} could be explained by compensation effects between both variables predicted by PV*. For C_m, predicted values saturate on the upper bound (at $C_m = 0.011$ µg^2 cm^{-1}) of its definition interval, suggesting that it is too tight. Similarly C_w saturates on its lower bound (at $C_w = 0.0075$ cm), which may be lowered to 0 cm. On the other hand, the upper saturation of C_w is at 0.03 cm, well below its upper bound that was set at 0.075cm. The set upper bound for C_w is at 0.075 cm, which is an extremely high, arguably rare value. In spite of that, PV* kept C_w predictions in a more reasonable range, below 0.03 cm. This suggests guidelines for setting variables definition intervals for PV* distributions. Bounds for TN distributions should only encompass possible values (e.g. no negative bio-chemical content value). However they could be set rather

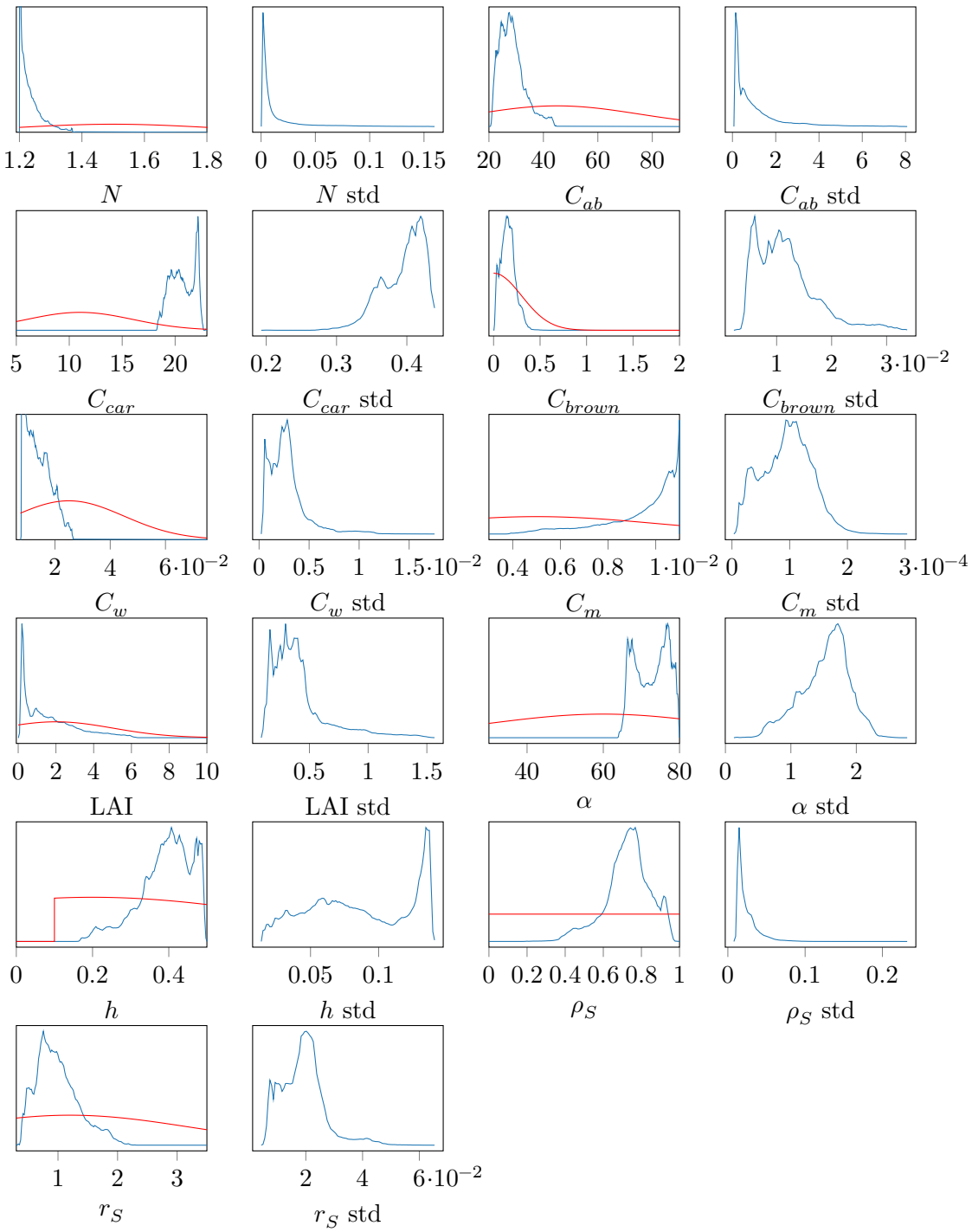


Figure 8.13: Blue: Histograms of expected values and standard deviation of PROSAIL variables inferred by PV* on S2 testing data-set. Red: PROSAIL variables distributions well-established in the literature (see Table 5.1).

loose, since PV* appears to be able to restrict effectively predicted values to a reasonable range. This highlights that the definition intervals for variables in PROSAIL-VAE are a weak prior that can be set roughly, contrary to the parameters for the sampling distributions for supervised approaches. In general, the distributions used in the literature (see Table 5.1) are very different from the variable distributions predicted by PV*. The closest match is observed for LAI and C_{brown} variables.

8.2.3.4 Correlations between PROSAIL variables

The results corroborate that a meaningful correlation between LAI and hot-spot parameters was predicted by PV*. Figure 8.14 shows the scatter plot between these variables, which are inferred for a wheat crop parcel of the BelSAR site at different 2018 dates. The hot-spot parameter effect accounts for the variation of the sensed reflectance as a function of the viewing angle, due to reflections inside the canopy, and is controlled in PROSAIL by the hot-spot parameter. This effect is related to the 3D structure of the canopy. As observed in Figure 8.14, the hot-spot parameter decreases when LAI values are greater than one. This correlation inferred by PROSAIL-VAE follows the theory about the hot-spot parameter of the Scattering by Arbitrary Inclined Leaves (SAIL) model [Verhoef, 1998] suggesting that $h \propto 1/\text{LAI}$ for plants growing taller with constant leaf size, such as wheat (see subsubsection 4.2.3.3).

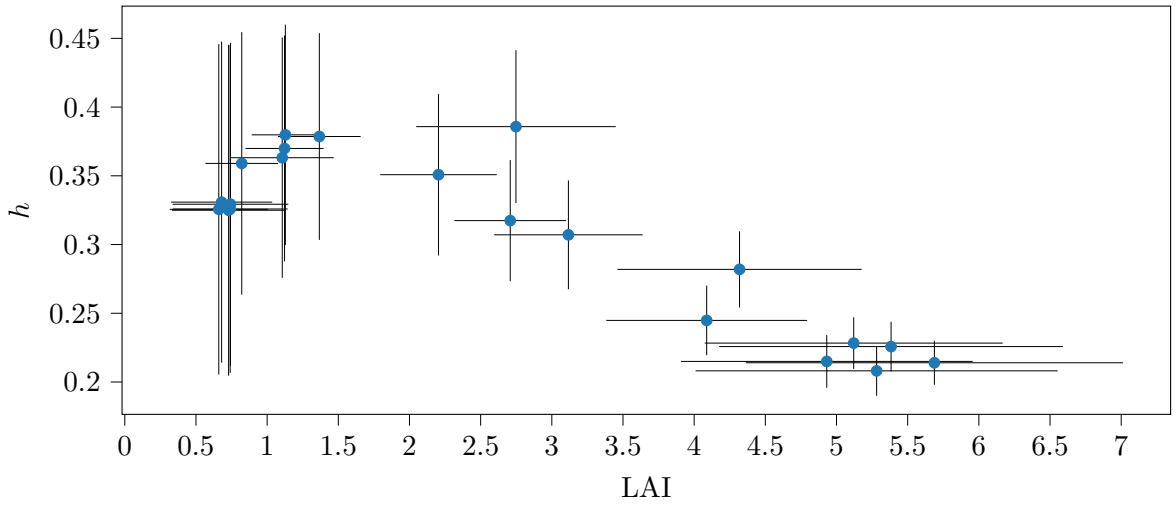


Figure 8.14: Scatter-plot of LAI versus hot-spot parameters predicted by PV*. Blue dots are the expected value of LAI and hot-spot, and horizontal and vertical lines correspond are the std. The predictions are averaged over a wheat crop parcel of the BelSAR campaign. The prediction are performed over the year 2018.

For a further analysis, the scatter-plots between the expectation of all pairs of PROSAIL variables predicted by PV* on the S2 testing data-set are available in Figure 8.15. These figures characterize the co-distributions between PROSAIL variables experimentally discovered by PROSAIL-VAE from remote sensing data. These experimental correlations must be handled carefully because additional validation is required to ensure that each variable relationship is verified with in-situ data. At this stage, the discovered correlations cannot be guaranteed to match the true biophysical variables correlations, let alone the marginal distributions of single variables. For instance, the predicted carotenoid content C_{car} is correlated negatively with the predicted chlorophyll content C_{ab}, contrary to the current understanding [He et al., 2023; Thomas and Gausman, 1977]. In fact, C_{car} is predicted within a restricted range of within the definition interval, showing little variability. This parameter may not be estimated accurately, likely because the carotenoid content has a limited spectral contribution to sensed S2 bands. As shown by the gradient based-sensitivity analysis (see

subsection 4.5.3), the gradient of the S2 bands w.r.t. the carotenoid content is non-zero only for `B2` and `B3`. Furthermore, for PV*, the reconstruction of `B2` less accurate than others, because it is not penalized in the loss, which lowered the influence of the carotenoid content in the training.

Other pairs of variables have unrealistic correlations. It is improbable that PROSPECT leaf parameters are strongly correlated with SAIL canopy parameters. The chlorophyll content C_{ab} should not be correlated to the hotspot parameter h which only depend on the canopy structure. The LAI should also not be correlated with C_{car}, C_w. The soil parameters s_w and s_b should be relatively independent from other PROSAIL variables, contrary to what observed with C_{brown}, C_m and $\overline{\alpha}$. These unlikely correlations corroborate the existence of some mechanism of compensation between the variables in the simulation/reconstruction process in PV*: variables with competing effects in the simulation of S2 bands are not well jointly estimated. It is also likely caused by the ill-posedness of the inversion problem: there is more PROSAIL variables to estimate than S2 bands on which to apply the objective function. A potential mitigation of compensation would be to reduce the problem complexity, either by reducing the number of PROSAIL variables estimated (i.e. reducing the degree of freedom of the solution) or increasing the number of bands penalized in the reconstruction. Besides, different PROSAIL-VAE models with different configurations can display different variable correlations, as will be discussed in the next sections.

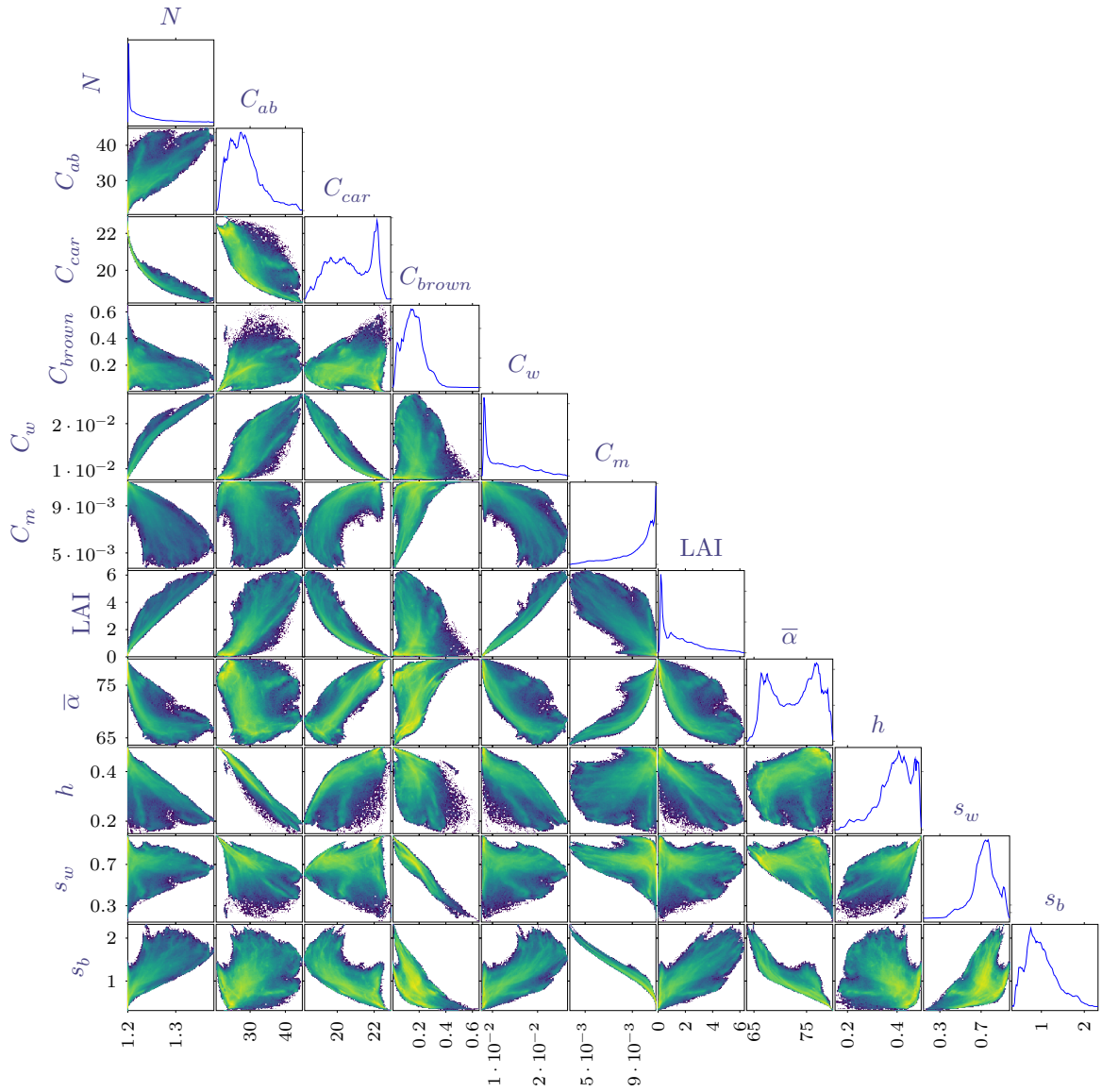


Figure 8.15: Pair-plots between PROSAIL variables retrieved by PROSAIL-VAE from the testing S2 image patch data-set $\mathcal{D}_{\text{S2}}$ (see section 2.2). Each data point is the expected value of a PROSAIL variable predicted by $\text{PV}^{\star}$.

8.3 PROSAIL-VAE variants

In this section, the influence of variations of the base PROSAIL-VAE configuration (see Table 8.2) are discussed. In subsection 8.3.1, changes in the prior distribution are investigated. Different methods for computing the reconstruction loss are discussed in subsection 8.3.2. The architecture of the PROSAIL-VAE encoder is discussed in subsection 8.3.3. Finally, a semi-supervised training strategy for PROSAIL-VAE is introduced in subsection 8.3.4.

8.3.1 The prior distribution

First, subsubsection 8.3.1.1 studies the influence of the hyper-parameter β, and of different sets of latent variables being penalized by the KLD loss term. Then, subsubsection 8.3.1.2 introduces a different kind of prior that is learned rather than pre-set. Finally, subsubsection 8.3.1.3 discusses the use of an external model to provide a prior for PROSAIL-VAE.

8.3.1.1 Uniform prior distribution

PROSAIL-VAE configurations The base PROSAIL-VAE configuration uses a uniform prior distribution for computing the KLD loss term, applied only on the latent dimension related to the LAI and with $\beta = 2$. Two variations of these elements are investigated here:

1. Applying the uniform prior on different sets of PROSAIL variables.
2. Changing balance between the reconstruction and the KLD loss by tweaking the coefficient β. Three different values are considered: 0, 1 and 2.

The configuration $\beta = 0$ is denoted as PV-NP and it indicates that no prior is considered[4] (PROSAIL-VAE "no prior"). To investigate the effects of penalizing different sets of variables with the KLD loss term, three configurations with different sets of variables affected by the uniform prior are considered:

- a single prior on the LAI (PV-L),
- the use of priors on LAI and C_{ab} (PV-LC),
- priors on all PROSAIL variables (PV-AV).

The seven combinations of these configurations are detailed in Table 8.10. They are thereby refered by an acronym that accounts for the variables affected by the uniform prior and the value of β. For instance, the PV* model which has been investigated thoroughly in section 8.2 is a PV-L-B2 model.

Quantitative assessment using in-situ data-sets For each of these configurations described above, 10 PROSAIL-VAE models are trained (see subsubsection 8.1.1.2), enabling to assess the influence of those configurations despite the variability between trained models. Figure 8.16 shows the RMSE, MPIW and PICP for the LAI, whereas these mterics for CCC are shown in Figure 8.17 for the different test sites. These figures also compare the RMSE metrics obtained with SNAP.

In terms of LAI RMSE, PROSAIL-VAE models with all configurations consistently outperformed SNAP on the Wytham site. In contrast, SNAP performed a little better on Barrax's 2021 campaign, although PV* also has high accuracy with low RMSE. On the BelSAR campaign, some PROSAIL-VAE models of all configurations overcame SNAP. The results on this campaign highlight the variability of performance between the different PROSAIL-VAE configurations. The PV-AV configuration forcing the prior on all input PROSAIL

[4] This is equivalent to assuming that the inferred approximate posterior $q(\boldsymbol{z}|\boldsymbol{x})$ is always equal to the prior $p(\boldsymbol{z})$, and therefore their KLD is zero. Such a "prior" therefore brings no information to the model.

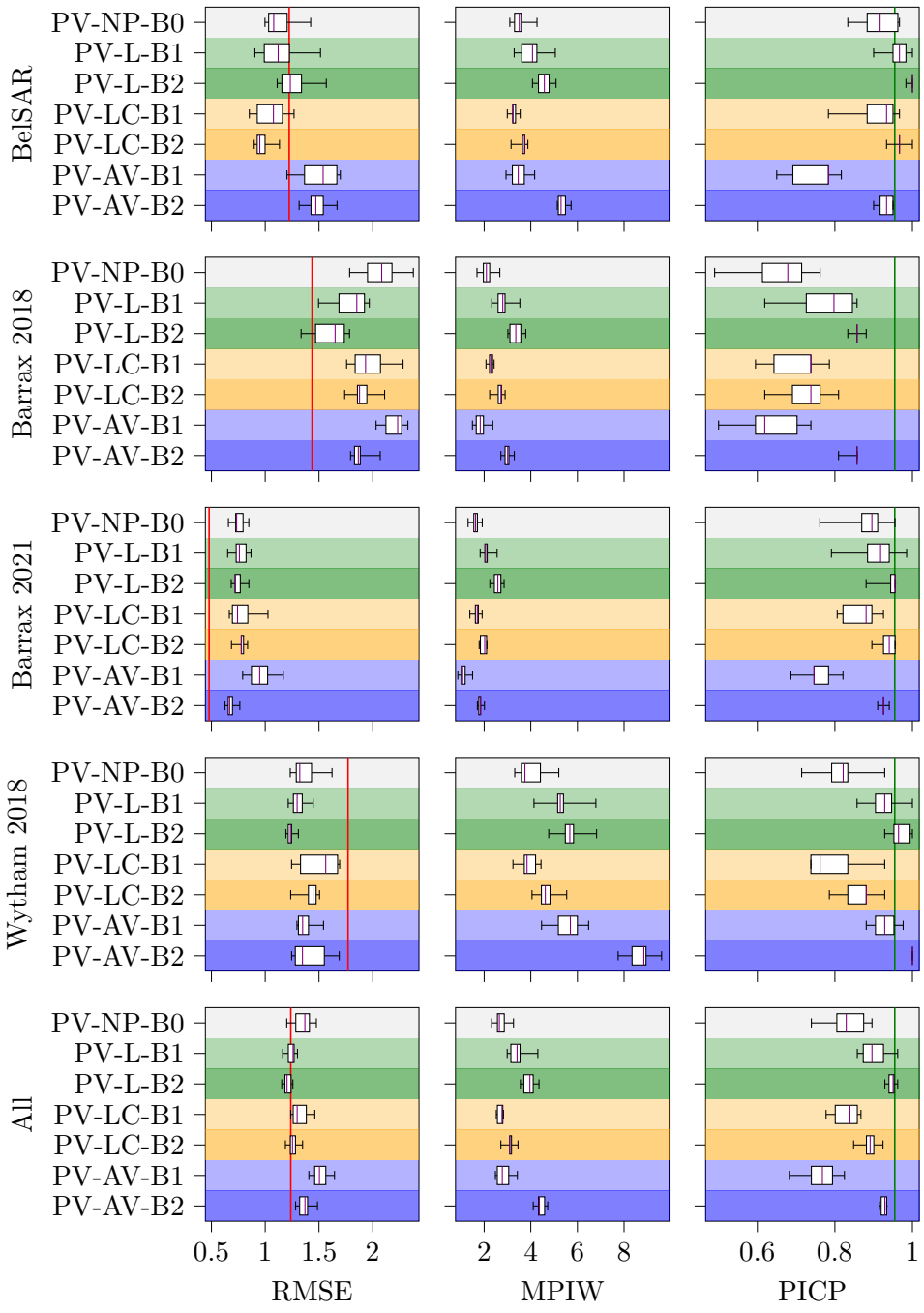


Figure 8.16: Box-plots of LAI metrics obtained on in-situ data-sets with PROSAIL-VAE models and MPSR. For each configuration, 10 models are trained and attained min and max values are displayed by boxplot whiskers. The box sites are the 25^{th} and 75^{th} centiles, and the line inside corresponds to the median. The vertical red lines on the left column indicate SNAP's RMSE. The vertical green line on the right column indicates the target ratio of values that lie within 2-σ prediction intervals (≈ 0.95)

Table 8.10: Studied PROSAIL-VAE configurations which depend on the value of β and on the used variable priors. The configuration acronyms consider : *NP* for "no prior", *L* for "LAI", *LC* for "LAI and C_{ab}", *AV* for "all variables". The acronyms also incorporate the value of the parameter β by appending "-Bβ"

Configuration acronym	Prior type	Variable prior	β
PV-NP-B0	None	None (PV-NP)	0
PV-L-B1	Uniform	LAI	1
PV-L-B2		(PV-L)	2
PV-LC-B1		LAI, C_{ab}	1
PV-LC-B2		(PV-LC)	2
PV-AV-B1		All variables	1
PV-AV-B2		(PV-AV)	2

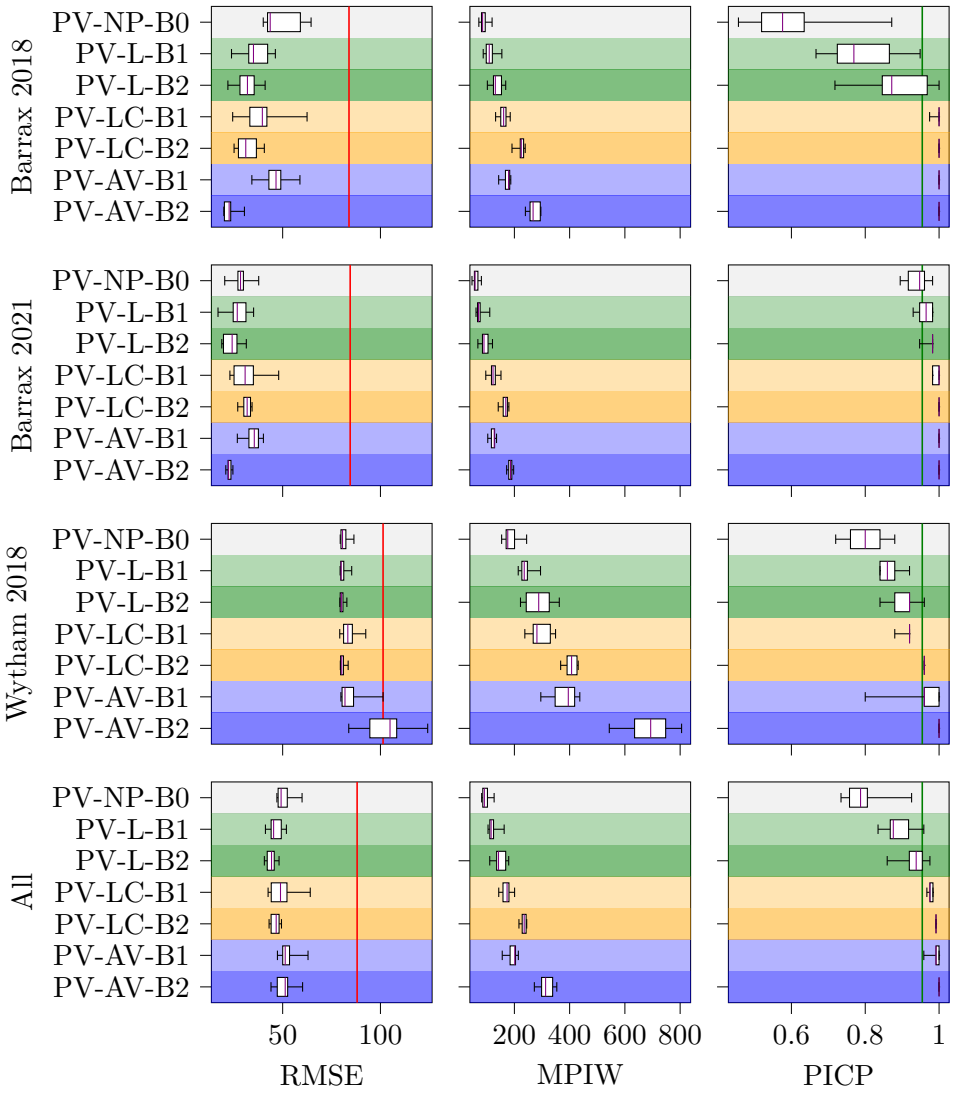


Figure 8.17: Box-plots of CCC metrics obtained on in-situ data-sets with PROSAIL-VAE models and MPSR. For each configuration, 10 models are trained and attained min and max values are displayed by boxplot whiskers. The box sites are the 25^{th} and 75^{th} centiles, and the line inside corresponds to the median. The vertical red lines on the left column indicate SNAP's RMSE. The vertical green line on the right column indicates the target ratio of values that lie within 2-σ prediction intervals (≈ 0.95)

parameters obtained the worst results. In contrast, PV-LC reaches the best performances on BelSAR campaign by only considering LAI and C_{ab} priors. As for the Barrax 2018 campaign, it was the PV-L configuration that performed best. All configurations except PV-AV had PROSAIL-VAE models that achieved a better overall RMSE performance than SNAP. It is noteworthy that even with no prior at all (PV-NP-B0), a performance similar to SNAP could be achieved.

For the CCC RMSE, the PROSAIL-VAE models consistently outperformed SNAP in all in-situ data-sets. The different PROSAIL-VAE configurations obtained similar results, except for PV-AV that performed slightly worse. The results obtained in Wytham shows how the use of a prior on all PROSAIL variables led to the decrease of the accuracy of the results, which was exacerbated by $\beta = 2$.

Predicted uncertainties for PROSAIL-VAE are evaluated with the MPIW and PICP metrics respectively in the center and right column of Figure 8.16 for the LAI, and of Figure 8.17 for CCC.

The MPIW values obtained for the LAI and CCC show that adding KLD regularization terms ($\beta > 0$) increased the variance of LAI and CCC distributions. In general, the prediction intervals become wider when β increases, as theorized in subsection 7.3.4. The narrowest intervals were reached by PV-NP. On the LAI, the MPIW was further increased with PV-L and with PV-AV with $\beta = 2$.

The PICP depends on both the estimation error being low and the prediction intervals being large enough. When the estimation RMSE is similar between models, the PICP increases as the MPIW increases. For instance for the LAI, PV-L had a greater PICP than PV-NP because it had wider prediction intervals (larger MPIW) and a similar RMSE. For the CCC, the RMSE varied relatively little between PROSAIL-VAE models so the ordering of the models in terms of PICP is consistent with the MPIW.

For most models in all configurations, the PICP on the LAI was close to the 0.95 target for all campaigns, except for Barrax 2018. Due to a higher error (RMSE) in this campaign and prediction intervals not wide enough to compensate, the PICP falls short of the target. The uncertainty on the Barrax 2018 LAI was underestimated, leading to most models obtaining an overall PICP lower than the target. Compared to other experiments, the PV-L configuration achieved lower RMSE and had larger MPIW on this campaign. This configuration thus achieved a PICP close to the target.

Concerning CCC, models of all experiments except PV-NP could achieve the 2σ target, in all campaigns and in overall results. PV-NP underestimated uncertainty because of lower MPIW than other experiments. Nonetheless, a PICP of 1 for the PV-LC and PV-AV configurations indicates that the prediction intervals always intercepted the true CCC value. These intervals were too large, they overestimated the uncertainty and the PICP target was overshot.

8.3.1.2 Learned prior distribution

A prior distribution $p(\boldsymbol{z})$ similar to that used in Svendsen et al. [2021] is studied: the use of a learnable prior (see subsection 8.1.2). For PROSAIL-VAE, this prior is a factorized TN distribution whose parameters μ and σ are optimized during training. Subsequently, the associated $\mathcal{D}_{\text{KLD}}$ loss term is the KLD between two TN distributions (see subsubsection C.4.6.1). A PROSAIL-VAE model, denoted PV_{LP} is trained with the same configuration than PV*, but using this learnable prior, applied on all variables and with $\beta = 1$.

Compared to PV*, PV_{LP} did not obtain enhanced in-situ validation results (see Table 8.11). This configuration has an overall lower variable estimation accuracy, and due to smaller prediction intervals, the PICP doesn't reach the 0.95 target.

Nonetheless, it is interesting to compare the values of the learned prior with the predicted distribution of PROSAIL variables over the testing data-set. Specifically, the expected value and variance of each PROSAIL variable prior distribution $p(z)$ is compared to the mean

Table 8.11: Performance on the retrieval of in-situ LAI and CCC with PROSAIL-VAE with a truncated normal learnable prior.

Variable	LAI			CCC		
Metric	RMSE	MPIW	PICP	RMSE	MPIW	PICP
PV_{LP}	1.34	2.30	0.77	50.09	104.55	0.86
$PV^{\star}$	**1.16**	4.04	**0.95**	**40.06**	147.96	**0.93**

and variances of the distribution $q(z)$ of inferred PROSAIL variables over the testing $\mathcal{D}_{S2}$, in Table 8.12. Like with Svendsen et al. [2021], a close matching of the mean is observed between the learned prior $p(\boldsymbol{z})$ and the aggregate posterior PROSAIL variable distribution $q(\boldsymbol{z})$. For the variance however, only the LAI, $\overline{\alpha}$ and s_b had close values. For the other variables, the learned prior $p(\boldsymbol{z})$ had a larger variance than the aggregated posterior $q(\boldsymbol{z})$, sometimes an order of magnitude higher.

Table 8.12: Comparison between the mean and variance of the learned prior of PROSAIL-VAE $p(\boldsymbol{z})$ and the inferred PROSAIL variable distributions $q(\boldsymbol{z})$.

Variable	Mean		Variance	
	$p(\boldsymbol{z})$	$q(\boldsymbol{z})$	$p(\boldsymbol{z})$	$q(\boldsymbol{z})$
N	1.49	1.44	2.81×10^{-2}	8.41×10^{-3}
C_{ab}	39.02	38.94	2.75×10^{1}	1.18
C_{car}	18.62	19.59	7.10	2.68×10^{-2}
C_w	0.095	0.045	6.35×10^{-3}	7.43×10^{-4}
C_m	0.022	0.023	3.85×10^{-5}	5.53×10^{-6}
C_{brown}	0.0093	0.010	1.41×10^{-6}	7.82×10^{-7}
LAI	1.26	1.23	1.26	1.43
$\overline{\alpha}$	69.42	69.32	2.50×10^{1}	2.16×10^{1}
h	0.13	0.13	5.81×10^{-4}	1.01×10^{-5}
s_w	0.89	0.92	8.67×10^{-3}	1.13×10^{-2}
s_b	0.71	0.74	1.23×10^{-1}	1.21×10^{-1}

8.3.1.3 Hyper-prior

Another design choice for the prior distribution that has been considered in this Ph.D. is the use of a dynamic, local prior, that changes for each new encoded sample $\boldsymbol{x}$, as opposed to a global prior that is the same for all data. Such a prior should be provided by a method external to PROSAIL-VAE, e.g. an auxiliary neural network. Since this "prior" would be conditional to the input data: $p(\boldsymbol{z}|\boldsymbol{x})$, thus it is not a prior in the Bayesian sense.

Nonetheless, the idea is to improve or help with the training of PROSAIL-VAE, by introducing the knowledge of an already working algorithm. This "prior" is hereby called *informed prior* or *hyper-prior*. Two possibilities have been considered for producing a hyper-prior.

- A MPSR neural network that predicts TN distributions of the PROSAIL variables could be used. Then, the hyper-prior would regularize the training of PROSAIL-VAE through the KLD between two TN distributions.

- A classical deterministic supervised neural network model such as SNAP could provide an estimate of some PROSAIL variables. This "prior knowledge" would be enforced by using the NLL of PROSAIL-VAE TN distributions w.r.t. this variable estimates. This

option further diverges from the classical use of prior in the VAE framework, since it no longer involves a KLD.

Unfortunately, experiments with the hyper-priors have not been conclusive. Since the PROSAIL-VAE model already performs well on validated variables, no improvement could be observed by using the output of the less accurate models that are MPSR and SNAP.

8.3.2 Likelihood model

Here, different designs for the likelihood model in PROSAIL-VAE (i.e. the distribution of the decoder) are investigated. A deterministic version of PROSAIL-VAE is proposed in subsubsection 8.3.2.2. The effect of penalizing a subset of the reconstructed S2 bands in the loss is discussed in subsubsection 8.3.2.3.

8.3.2.1 Alternative variance computation

For computing a Gaussian NLL reconstruction loss, it is the variance computation that differentiates the different VAE methods (see subsubsection 7.2.1.2). The MCRL approach is compared by computing the reconstruction variance with two different methods:

- The use of a pre-set, constant variance, set to 1×10^{-7} like in Svendsen et al. [2021]. This is a very low variance value, that assumes that the reconstruction are very realistic. The related PROSAIL-VAE is denoted PV_{cst}.
- The use of an auxiliary neural network that takes the sampled latent variables as input and outputs a variance estimate. The architecture of this neural network is similar to that of the encoder of PROSAIL-VAE (see Figure 8.2), with adapted input and output layers and with only one residual connection block to make it simpler. The related PROSAIL-VAE is denoted PV_{NN}.

In both cases, only one latent sample $\boldsymbol{z}$ is drawn, contrary to the MCRL. The reconstruction that is obtained by forwarding this latent sample is taken as the mean μ of the Gaussian likelihood, like with classical VAE. After training, these PROSAIL-VAE variants are assessed using the in-situ validation data. The overall in-situ validation RMSE provided in Table 8.13 are worse than for the reference $\text{PV}^{\star}$ (see subsection 8.2.2) and than even all models of the same PV-L-B2 configuration (see subsubsection 8.3.1.1), suggesting that the MCRL approach was superior. The LAI RMSE is sub-par compared to other PROSAIL-VAE

Table 8.13: Performance on the retrieval of LAI and CCC with PROSAIL-VAE with alternative decoder variance computation PV_{cst} and PV_{NN}.

Variable	LAI			CCC		
Metric	RMSE	MPIW	PICP	RMSE	MPIW	PICP
PV_{cst}	1.45	0.23	0.09	41.84	8.86	0.03
PV_{NN}	1.49	2.04	0.45	44.94	63.26	0.40

models. Nonetheless, the CCC RMSE is comparable to other PROSAIL-VAE configurations. But the main issue with PV_{NN} and PV_{cst} is that they predicted very narrow distributions, as showed by the low MPIW values. As a consequence, uncertainty is underestimated, and the PICP is very far from the 0.95 target.

8.3.2.2 Deterministic PROSAIL autoencoder

The idea of introducing a physical model into the decoder of a VAE can also be applied to a deterministic auto-encoder. As such, an additional experiment is performed: PROSAIL

is integrated as the decoder of a classical autoencoder, which uses the encoder architecture of PROSAIL-VAE. This approach is referred as P-AE. No sample is drawn in the latent space, the encoder's output is the code that is scaled to PROSAIL variables, and input to the decoder. A simple mean squared error (MSE) reconstruction loss is the objective function of P-AE. Since it is not a probabilistic approach, only the LAI and CCC retrieval RMSE are evaluated on in-situ data.

For the LAI, the RMSE is 1.55, and it is 45 µg cm^{-2} for CCC. While the CCC is on par with PROSAIL-VAE models, the LAI is not estimated as well.

The superiority of PROSAIL-VAE can be interpreted by the space of PROSAIL variables being explored more during optimization due to the sampling of the latent space. Additionally, the use of probabilistic objective function terms (the NLL in the reconstruction loss and the KLD loss) introduce regularization effect during training.

8.3.2.3 Penalization of the B2 band

The base PROSAIL-VAE configuration uses the MCRL NLL applied on all reconstructed spectral bands *except* `B2`. This band was involved in the estimation (i.e. it is input to the encoder) but not in the optimization (i.e. it is not taken into account in the reconstruction loss). This design choice for PROSAIL-VAE was made because it significantly improved the accuracy on the retrieval of the LAI, and enabled PV* to have a better RMSE than SNAP. Besides, `B2` is a band for which atmospheric corrections have higher uncertainty (see subsubsection 2.1.3.4). Also, this band neither is used for training nor is taken as input by SNAP (along with `B8`). In Table 8.14 are shown the in-situ validation performances for a PROSAIL-VAE with the base configuration but includes the `B2` into the reconstruction loss, referred as PV$_{\texttt{B2}}$.

Table 8.14: Performance on the retrieval of LAI and CCC with PROSAIL-VAE with the reconstructed `B2` band being penalized in the loss.

Variable	LAI			CCC		
Metric	RMSE	MPIW	PICP	RMSE	MPIW	PICP
Score	1.40	4.88	0.97	46.86	133.39	0.93

The validation performances of PV$_{\texttt{B2}}$ are inferior to that of PV*. Additionally, this configuration changes the inferred distribution of certain the PROSAIL variables on the testing $\mathcal{D}_{\text{S2}}$. In particular, the distribution of the carotenoid content C_{car} is different. In PV*, this distribution was concentrated on the upper bound of the variable allowed range, and had a negative correlation with C_{ab}. On the contrary, for PV$_{\texttt{B2}}$ values are concentrated on the lower bound of the allowed range, and the correlation with C_{ab} is positive. This positive correlation seems to be an improvement of PV$_{\texttt{B2}}$ over PV*. However, it is not possible at the moment to conclude if this variable is correctly predicted in either case without validating with in-situ data, especially considering the small range of the predicted C_{car} in both models.

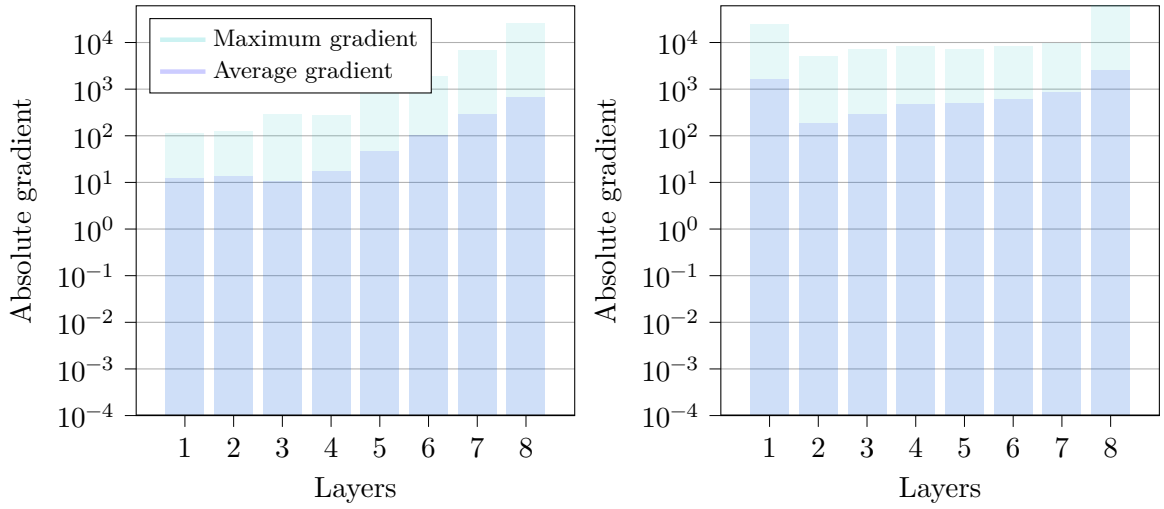
8.3.3 Encoder architecture

8.3.3.1 Gradient propagation and residual connections

The encoder architecture is based on a residual neural network backbone. The rationale behind this choice is to help improve the propagation of gradients throughout the model so that the update of weights is more efficient, and to avoid the vanishing gradient problem (see subsubsection 3.3.1.3).

To evaluate the impact of this architecture choice on PROSAIL-VAE's training, the gradients of the model are retrieved for the first iteration of the first epoch. The values of these gradients are compared layer by layer in Figure 8.18, for a PROSAIL-VAE model

with the base configuration and for a PROSAIL-VAE with the same number of neurons and layers, but no residual connection in the encoder.



(a) Without residual connections

(b) With residual connections

Figure 8.18: Comparison of the gradients in the layers of the encoder PROSAIL-VAE with or without residual connections.

It can be observed that the gradients in the neural networks layers have different behaviors. Without residual connections, the gradient values steadily decreased from the last layer (8) to the input layer (1), with a factor 10^2. With residual connections, the gradient keeps a more constant average value throughout the model layers, and the first layer has a similar gradient level than the last layer.

Since the gradients are estimated at the very beginning of the training, they have quite high values, so the optimization of all layers still occurs. However, when the training progresses and when the loss begins to converge, the gradients become lower, and the decrease due to the depth of the model may prevent the innermost layers from being updated.

8.3.3.2 Spatial encoder

The architecture of the encoder of PROSAIL-VAE has two variants: the pixel-wise variant, which uses classical fully connected neural layers, and the spatial variant which is a CNN. All experiments with the different configurations presented until now have used the pixel-wise variant. In parallel, the same experiments have been carried out using the spatial variant. However, no notable difference between results obtained by pixel-wise and spatial encoder architectures has been observed.

This similarity in results can be attributed to the fact that only the first layer of the CNN used convolution filters larger than 1 pixel in order to preserve the input data resolution. Therefore, only the first layer of the neural network does take into account the spatial context. Besides, perhaps more importantly, the PROSAIL model is a 1D RTM which doesn't take into account the spatial context, and the reconstruction loss is applied to each pixel independently. Therefore, neither the decoder nor the objective function promote taking the spatial context into account for inference.

8.3.4 Semi-supervised cyclical training

PROSAIL-VAE is a purely self-supervised approach. It only requires some S2 data as input, and learns to reconstruct it while inferring a set of PROSAIL variables. Supervised deep learning approaches however rely on the simulation of a training data-set, and their performance depend on the choice of the distribution made for sampling this data-set.

The PROSAIL-VAE approach still relies on simulations. However those simulations do not play the same role as in supervised approaches. In particular, they are not driven by samples of arbitrary distributions, but by inferred parameters that hopefully match the vegetation observed in real remote-sensing data. The distribution of retrieved parameters may not be a good approximation of the true underlying distribution of vegetation variables. However, at the very least, since PROSAIL-VAE minimizes the a reconstruction loss, the distribution of simulated reflectance bands is by design a good approximation of the distribution of S2 bands found in real images.

This introduces an opportunity for a hybrid, semi-supervised approach. Real S2 images do not have an associated PROSAIL variable reference mapping, which is why supervised approaches must resort to simulation. However, the reconstructions of PROSAIL-VAE have reference PROSAIL variables by design, because they are simulations. What makes those simulations different is that they were made to match the S2 input data. Therefore, an approach that could improve PROSAIL-VAE, would be to use the estimated PROSAIL variables along with their reconstruction as supervised training data. In practice, the encoder of PROSAIL-VAE would be also trained like MPSR, but with data that is generated on-the-fly for each new S2 image rather than pre-simulated data. Once reconstructions $\boldsymbol{x}_1$ are generated by PROSAIL-VAE from estimated PROSAIL variables $\boldsymbol{y}_1$, they are forwarded as a new input to the encoder, which estimates new PROSAIL variables distributions $\boldsymbol{y}_2$ that can compared to the "reference" $\boldsymbol{y}_1$. For instance, the latent distribution produced by the encoder can be penalized with a supervised loss term $\mathcal{L}_{\text{sup}}$, e.g. the NLL of the distribution w.r.t. $\boldsymbol{y}_1$. This approach is hereby referred as *cyclical training*, because it involves feeding the encoder with the output of the decoder.

Implementations of cyclical training can be set in two categories.

- PROSAIL-VAE is optimized jointly with the usual VAE ELBO loss and with the additional supervised loss term, weighted with γ:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \beta\mathcal{L}_{\text{KLD}} + \gamma\mathcal{L}_{\text{sup}}. \tag{8.10}$$

- PROSAIL-VAE is optimized sequentially with either the ELBO or the $\mathcal{L}_{\text{sup}}$. The training would then cyclically be self-supervised, then supervised. When starting with a self-supervised phase, this could enable to train the encoder efficiently so that reconstructions become realistic.

This approach could even be used to train spatial (e.g. CNN) models that require images and not just pixels as input. This is because reconstructions with PROSAIL-VAE are images. Besides, the reconstructions generated by a PROSAIL-VAE model could even be used offline for training a spatial model. In other-words, PROSAIL-VAE could be used perform simulations of images of entire landscapes, even-though PROSAIL is a 1D model.

For now, experiments using such training strategies for PROSAIL-VAE have been inconclusive. Further investigation is required in the future to quantitatively assess the potential of these techniques, for improving the quality of retrieved PROSAIL variables, both in accuracy and in uncertainty quantification.

8.4 Conclusion

PROSAIL-VAE is a self-supervised deep-learning method that performs the probabilistic inversion of PROSAIL. It is based on the use of PROSAIL as the decoder of a VAE, based on the methodology developed in Chapter 7. One of its main advantages is that contrary to supervised models, it can be directly trained on remote sensing data, and doesn't require pre-simulating a training data-set. Therefore, it is not affected by an arbitrary choice of sampling distributions for physical parameters. The model has, to discover this underlying

distribution. On the contrary, supervised methods like SNAP and the MPSR are trained with a pre-simulated data-set, and the distribution of the sampled variables has an influence over performances, as discussed in subsection 5.2.2.

The use of in-situ measurements has enabled to compare the performance between different configurations of PROSAIL-VAE and with SNAP and MPSR. In particular, PV*, a PROSAIL-VAE model with good in-situ validation performance has been deeply investigated. However, it is important to acknowledge that in operational contexts, selecting a model by using validation data can lead to an over-fitting risk since testing data should be never used for setting hyper-parameters or choosing the best configuration. Therefore, reliable solutions based on cross-validation techniques (see 3.3.2.4) could be proposed to compare trained models if a large number of in-situ samples was available. Unfortunately, the experiments have also highlighted that the PROSAIL-VAE objective function is not a good indicator of a model's performance on in-situ data. This is because reconstruction is a training proxy task not the actual intended downstream task. Finding model indicators that do not require measurement data, but correlate well with in-situ validation performance, could enable selecting a model more reliably.

Nonetheless, the PROSAIL-VAE approach introduces an interesting method for identifying correlations between biophysical variables, based only on remote sensing data. The discovered relationship between the LAI and hostpot is one example of that. Furthermore, the arbitrary co-distributions traditionally used in the literature and used to sample biophysical variables for simulations (see subsection 5.1.3) are different than those found with PROSAIL-VAE. This suggests that these co-distributions may not describe real relationships between variables. For instance, a key difference is that with PROSAIL-VAE, most of the predicted variables do not exhibit a simple linear relationship with LAI. Nonetheless, prudence is warranted regarding these experimental correlations, since some pairs of variables had unrealistic relationships. This is likely due to the ill-posed nature of the inverse problem, and compensation mechanisms between PROSAIL variables. Perhaps a way to improve inversion performance would be to enforce known relationships between variables. Such constraints could be applied using disentanglement methods that penalize the aggregate covariance between variables, like DIP-VAE (see subsubsection 6.5.2.1).

Visual results and inferred distributions have suggested that some PROSAIL parameters were better predicted than others. This could be explained by the ill-posedness nature of the inverse problem or by the importance of each input variable in the PROSAIL model. Furthermore, in-situ measurements related to less studied parameters such as carotenoids, brown pigments or dry mater content are necessary to quantitatively assess the performance of predicted variables. More in-situ data is also needed in both quantity and variety (vegetation types, location, season) to further validate the proposed hybrid methodology.

Compared to the existing simulation-assisted regression methods, PROSAIL-VAE requires little prior knowledge about the distribution of input model parameters. For instance, the configuration PV-NP-B0, (see subsubsection 8.3.1.1) has shown that accurate results can be obtained by only setting information about the input PROSAIL parameter's value ranges (only upper and lower bounds, not their distributions). The experiments have also corroborated that the selection of priors to be used in the KLD regularization term is not trivial. Adding some few priors describing well-known variables such as LAI can improve the prediction accuracies. Conversely, straightforwardly applying a prior on all latent variables can be detrimental to performances.

The results have also corroborated that the MCRL approach was better than traditional VAE approaches for computing a reconstruction loss.

Improvements in prediction accuracy could be also obtained by changing the SAIL version considered in our PROSAIL-VAE implementation. The obtained results have shown that a positive bias can exist for low LAI values, which may be explained by the insufficient capacity of the model to simulate realistic soil spectra. Instead of combining only two reference soil

spectra, the use of a soil spectral library could improve the prediction of low LAI values where the ground is visible. In the same direction, using other PROSPECT model versions or other RTM could improve the performances of our hybrid methodology. Using 4SAIL2 instead of SAIL could enable taking into account the canopy clumping effect, i.e. the effect of leaf distribution on the canopy that modifies the apparent LAI (see subsubsection 2.3.1.2). An important remark is that the change of the physical-based decoder to be inverted does not require any additional tuning task, which would be the case for simulation-based approaches as BVNET.

The PROSAIL-VAE used S2 images of vegetated areas for training. Nonetheless these images also contained non-vegetation elements (artificial, mineral or water surfaces), that were not filtered. This suggests that the PROSAIL-VAE approach is relatively robust to out-of-distribution elements in its training data-set. Since no labels are necessary for training, the model could use data-sets with images from all continental surfaces and not only Europe. The distribution of vegetation variables could be estimated in different parts of the world. Also, the approach is based on deep neural networks, whose operations can be easily parallelized on dedicated hardware (GPU). Therefore, the inference, which amounts a forward pass through the neural network model, can be scaled up to a global coverage.

The VAE-based inversion approach developed in this Ph.D. and investigated in this chapter is not specific to the PROSAIL RTM nor to S2 data. Future research efforts may concentrate on the incorporation of different physical models and on the use of different remote sensing data, such as other optical data or radar data. In Chapter 9, it is proposed to apply this methodology to a different problem: the retrieval of phenological parameters from NDVI time series. Notably, in this second application, additional priors are incorporated into the latent space to take into account the physical constraints between the phenological parameters.

Chapter 9

Phenological model inversion

Contents

In Chapter 8, the PROSAIL model is integrated as a user-defined decoder (UDD) of a variational autoencoder (VAE), using the methodology developed in Chapter 7. This enables to retrieve canopy biophysical variables from S2 images as interpretable representations of vegetation. In these experiments, the extracted representations are produced by exploiting the spectral dimension of the S2 imagery, because PROSAIL is a model that links biophysical variables to canopy spectra.

However, S2 data can also be interpreted temporally, because its relatively short revisit time (see section 2.1) enables to build image time series[1]. This chapter proposes another application of the Chapter 7 methodology. A *phenological model*, presented in section 9.2, which relates the normalized difference vegetation index (NDVI) time series with *phenological variables*, is integrated as the decoder of a VAE. The subsequent Pheno-VAE model, presented in section 9.2 incorporates additional constraints to ensure ordering of certain inferred distributions. This enables to invert the phenological model and retrieve phenological variables as a temporal representation of vegetation, in experiments shown in section 9.3.

9.1 Phenological model

Most vegetation goes through cyclical stages across time: this is phenology. Phenology can be represented as a succession of plant temporal phases, such as growth, stagnation, decay, dormancy, and landmark key moments or transition dates, such as sprouting, bud break, flower blooming, leaf onset, growth peak, senescence.

Each plant species has its own phenology. Even between plant individuals of the same species, there exist spatial disparities in phenology, due to different environments (e.g. temperature, humidity, solar illumination). As such, vegetation phenology is very diverse, but also case specific: some phenologies are typical of specific species in specific environments. Furthermore, climate change disturbs environmental conditions, with for instance shorter winters and longer summers, and plant phenology is thereby modified and not constant anymore in-between years [Chen et al., 2022].

Nonetheless, it is possible to propose *phenological models* that can represent a variety of plant phenologies that follow well-defined temporal patterns. Phenological models link measurements on vegetation to physiological stages. For optical remote sensing measurements, phenological models usually relate these *phenophases* to spectral indices, oftentimes the NDVI (see subsection 1.2.1) [Berra et al., 2017; Hall-Beyer, 2003; Zhu et al., 2012]. For S2, the NDVI is computed from the `B4` (red) and `B8` (near infra-red (NIR)) reflectance bands:

$$\text{NDVI} = \frac{\texttt{B8} - \texttt{B4}}{\texttt{B8} + \texttt{B4}}. \tag{9.1}$$

In this work, phenology is derived from S2 NDVI time series.

9.1.1 Logistic model

For deciduous vegetation and many crops, the leaf onset is usually followed by a rapid growth period, then a phenophase of maximum vegetation density (and leaf area index (LAI)). This growth in terms of plant size, leaf surface (i.e. lai), photosynthetic activity (i.e. NDVI, fraction of absorbed photosynthetically active radiation (FAPAR)) can be represented by a temporal *logistic* model:

$$f(t) = \frac{c}{1 + \mathrm{e}^{a+bt}} + d \tag{9.2}$$

with a, b, c, d the model parameters, and t the time, usually in day of years (DOYs). The parameter d is the minimum attained by the logistic model, while c is the amplitude of

[1] S2 data can be arranged into a Satellite image time series (SITS), a four-dimensional array (two dimensions of space, one temporal dimension and one spectral dimension).

change. The parameters a and b respectively control the offset of the growing phenophase, and its slope. Senescence and dormancy phenophases of these vegetation after reaching their peak can also be presented by logistic models. In both cases, logistic models can be used to represent a transition between low and high vegetation activity regimes.

Usually plant phenology considers transition dates t_1 and t_2 such that (s.t.) $t_1 < t_2$, that represent the beginning and end of the transition phenophase, rather than the abstract parameters a and b that controls the transition phenophase in the model. For instance, Zhang et al. [2003] recovers the transition dates from the maxima of the rate of change of the logistic model. In Pelletier et al. [2016b], transition dates t_1 and t_2 are defined as the dates of intersection of the tangent of the transition phenophase inflection point, with respectively the plateau levels $y = c$ and $y = c + d$. Using this definition, the logistic model can be re-parameterized[2] as:

$$f(t) = c\left(1 + \exp\left(2\frac{t_2 + t_1 + 2t}{t_2 - t_1}\right)\right)^{-1} + d. \tag{9.3}$$

When using this re-parameterized model of a growth phenophase, t_1 represents a start of season (SoS)[3] and t_2 a maturity (Mat) date (see Figure 9.1). Conversely, for a decaying phenophase, t_1 and t_2 are a senescence (Sen) date and an end of season (EoS)[4][5].

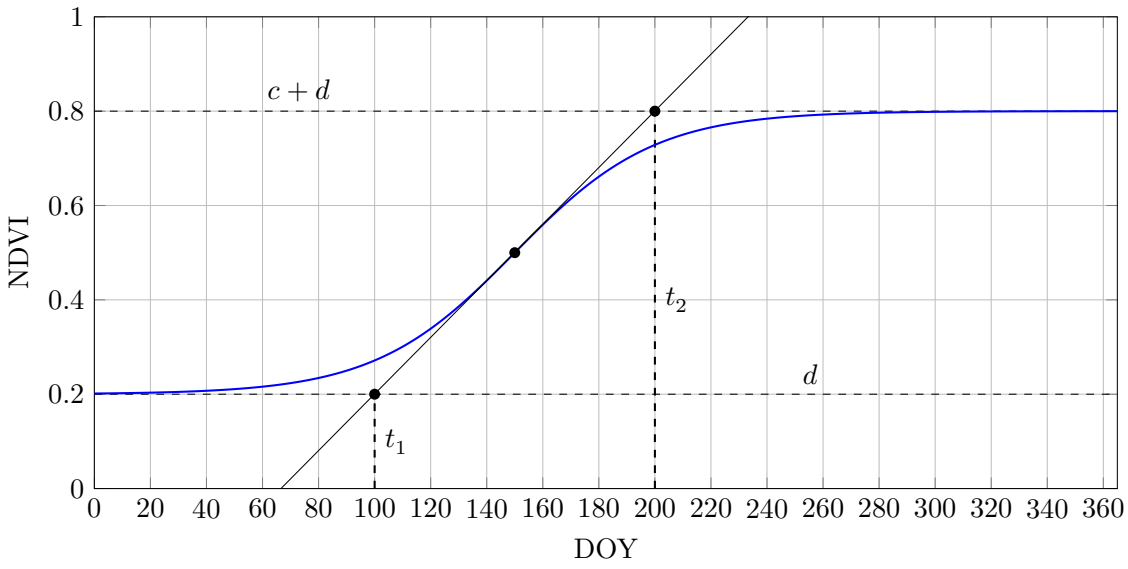


Figure 9.1: Re-parameterized logistic model (blue) representing a growing phenophase of vegetation.

9.1.2 Double-logistic model

When vegetation exhibits different growth and/or decay phenophases throughout a given time frame (e.g. a year), the phenological model can incorporate each phenophase by combining different logistic models, by summation [Pelletier et al., 2016b] or multiplication [Caglar et al., 2018]. The *double-logistic* phenological model used in this work (see Figure 9.2) is defined as:

$$\begin{aligned}\Omega(t, \boldsymbol{\phi}) = (M - m)\left[\left(1 + \exp\left(2\frac{Mat + SoS + 2t}{Mat - SoS}\right)\right)^{-1}\right. \\ \left. - \left(1 + \exp\left(2\frac{EoS + Sen + 2t}{EoS - Sen}\right)\right)^{-1}\right] + m,\end{aligned} \tag{9.4}$$

[2]This reparameterization was published in Zérah et al. [2023a].
[3]Also called *greenup onset*.
[4]Also called *dormancy onset* date.
[5]to obtain a decreasing model for representing a decay, t_1 must be swapped with t_2

with $\boldsymbol{\phi} = (m, M, SoS, Mat, Sen, EoS)$ the *phenological parameters*, summarized in Table 9.1. In this model, depending on the context, (see section 9.3) $\boldsymbol{\phi}$ can be seen either as model parameters or as inputs (see subsection 3.1.1). For instance as described later in this chapter, Pheno-VAE considers $\boldsymbol{\phi}$ as an input whereas curve-fitting algorithm considers $\boldsymbol{\phi}$ as model parameters. This phenological model represents a vegetation index (e.g. the NDVI) that starts at a minimum NDVI level (m), grows up to a maximum NDVI level (M) and then decays back to m.

Table 9.1: Parameters of the double-logistic phenological model and their proposed range for Northern hemisphere vegetation.

Variable	Description	Range $[a, b]$
M	Maximum of double logistic	[0, 1]
m	Minimum of double logistic	[0, 1]
SoS	DOY of start of season, the start of NDVI growth	[-45, 410]
Mat	DOY of maturity, the end of NDVI growth	[-45, 410]
Sen	DOY of senescence, the start of NDVI decay	[-45, 410]
EoS	DOY of end of season, end of NDVI decay	[-45, 410]

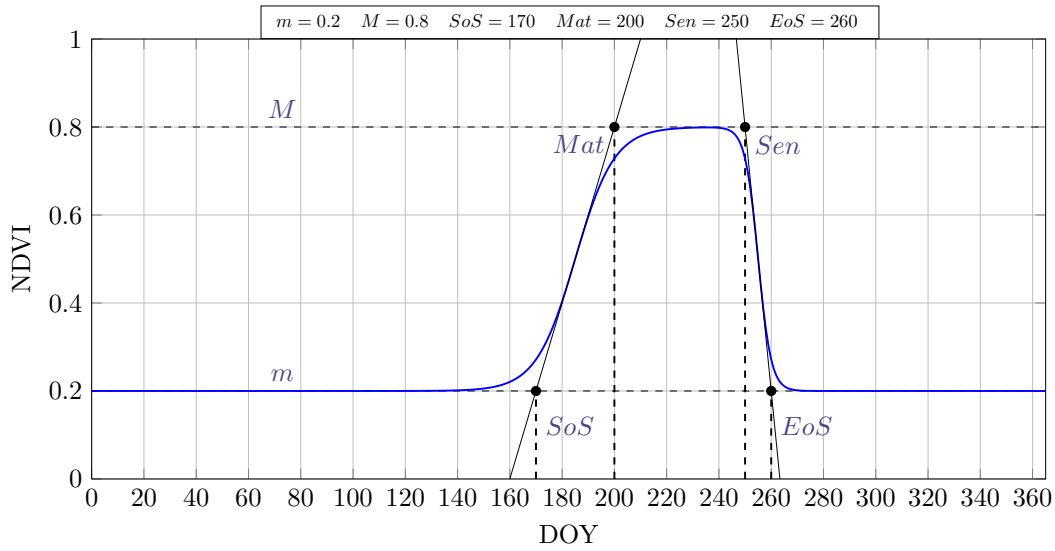


Figure 9.2: Double-logistic model representing an annual phenological cycle.

Phenological parameters are defined as *bounded*. M and m are constrained to the same definition interval than the NDVI index, whereas the *phenological dates* (SoS, Mat, Sen, EoS) are restricted to a range of given calendar year, extended by 90 days. The 45 days considered before January 1st and after December 31st allow less restrictive estimations and take into account vegetation whose cycle started or ended outside the calendar year.

There are many variations of logistic models used in the literature for phenology monitoring [Yang et al., 2012; Zeng et al., 2020; Zhang et al., 2003]. The model defined in Equation 9.4 is relatively simple, with few parameters, which makes it relatively easy to invert. However, it has limitations, which will be apparent in experiments: it assumes that the senescence phase brings the vegetation index back at the starting level, even-though there could be some residuals after senescence (e.g. crop regrowth, inter-cropping, presence of forest undergrowth). Other asymmetrical models such as in Caglar et al. [2018] do not make such an assumption. Furthermore, this double logistic model is limited to one growth phase and one decay phase in vegetation, and therefore cannot accommodate correctly time series with multiple phenological cycles, such as crops with intermediate cover or sequential crops.

9.1.3 Pheno-VAE

Pheno-VAE is defined as a VAE with the phenological model (see Equation 9.4) as a UDD (see subsection 7.2.1). The encoder of Pheno-VAE takes NDVI time series $\boldsymbol{x}$, and the UDD uses the phenological model to generate NDVI time series reconstructions $\hat{\boldsymbol{x}}$. The encoder of Pheno-VAE is a multi-layer perceptron (MLP) (see subsubsection 3.3.1.1) neural network, with a simple architecture shown in Figure 9.3. The input NDVI time series are assumed to be sampled on a regular temporal grid, representing a full year, with 5 days of temporal resolution (i.e. 73 data-points per time series). Similarly the outputs of the decoder are reconstructed time series with data-points on the same temporal grid.

The encoder infers the 6 sets of variational parameters λ_i that define the latent distributions that latent variables z_i are drawn from. Each variable z_i of its 6-D latent space is semantically bounded to a phenological parameter ϕ_i (see Table 9.1). Since all phenological variables are defined on bounded intervals, truncated normal distributions are used as variational distributions (see subsection 7.3.2). Instead of directly sampling the phenological parameters from the inferred latent distributions, an intermediary latent space is used (see subsubsection 6.5.2.3). The domains of the variational distributions are all set as $[0, 1]$, and phenological parameters ϕ_i are derived from latent samples z_i with affine transformations, using the interval bounds u_i, l_i of the associated ϕ_i (see Table 9.1):

$$\phi_i = (u_i - l_i)\, z_i + l_i. \tag{9.5}$$

The loss function for training Pheno-VAE is the evidence lower bound (ELBO) (see Equation 6.35), which is the sum of a reconstruction loss term and a regularization KLD loss term weighted by a coefficient β (see Equation 6.37):

$$\mathcal{L}_{\text{Pheno-VAE}} = \mathcal{L}_{\text{rec}} + \beta \mathcal{L}_{\text{KLD}}. \tag{9.6}$$

$\mathcal{L}_{\text{rec}}$ is computed as a Monte Carlo reconstruction loss (MCRL) (see subsection 7.2.2) which uses multiple latent samples $\boldsymbol{z}$, forwarded through the phenological model UDD for estimating the parameters of a Gaussian decoder distribution $p(\boldsymbol{x}|\boldsymbol{z})$, and compute a negative log-likelihood (NLL) reconstruction loss (see Equation 7.7). The prior distribution $p(\boldsymbol{z})$ used in the KLD loss term $\mathcal{L}_{\text{KLD}}$ is chosen as a uniform distribution on the same range ($[0, 1]$) as the truncated normal output by the encoder (see subsection 7.3.4 and subsubsection C.4.6.3). This prior doesn't promote specific modes in inferred distributions, which avoids favoring any phenological parameter set. The data-set used with Pheno-VAE (see subsection 9.3.1) contains various phenological configurations, making this a reasonable choice.

Aside from the encoder architecture, the data and the model used as UDD, Pheno-VAE is very much like PROSAIL-VAE (see Chapter 8). However, contrary to PROSAIL-VAE, physical variables inferred in the latent space are tied by order relationships, linked to the time temporal aspect of the data and of the physical model. Ensuring that these constraints are fulfilled is crucial, and methods to incorporate ordering between the latent variables of Pheno-VAE is discussed in the section 9.2 below.

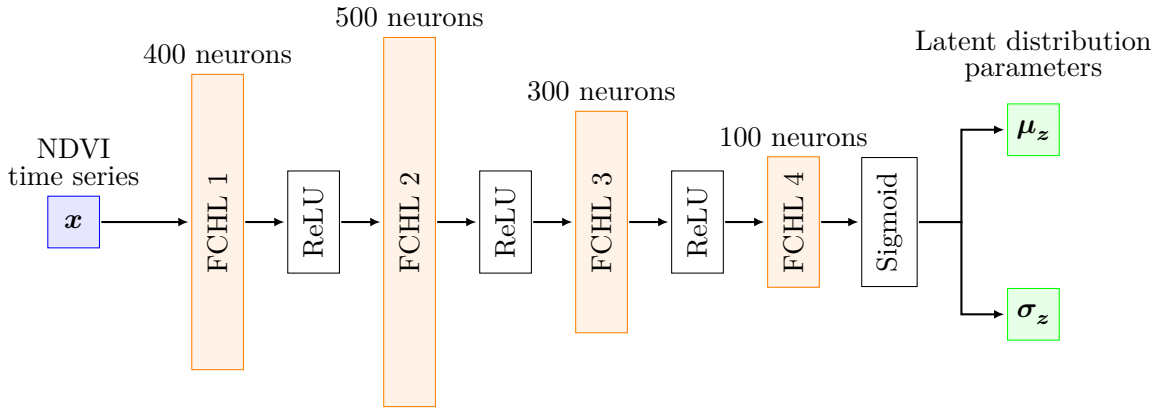


Figure 9.3: Encoder architecture used in Pheno-VAE, with 4 fully connected hidden layers (FCHL).

9.2 Integrating order constraints in latent distributions

The double logistic model requires that the phenological variables provided as input are ordered. Specifically, the phenological dates, and the minimum and maximum NDVI must be two sequences of ordered (*consecutive*) variables:

- $m < M$
- $SoS < Mat < Sen < EoS$.

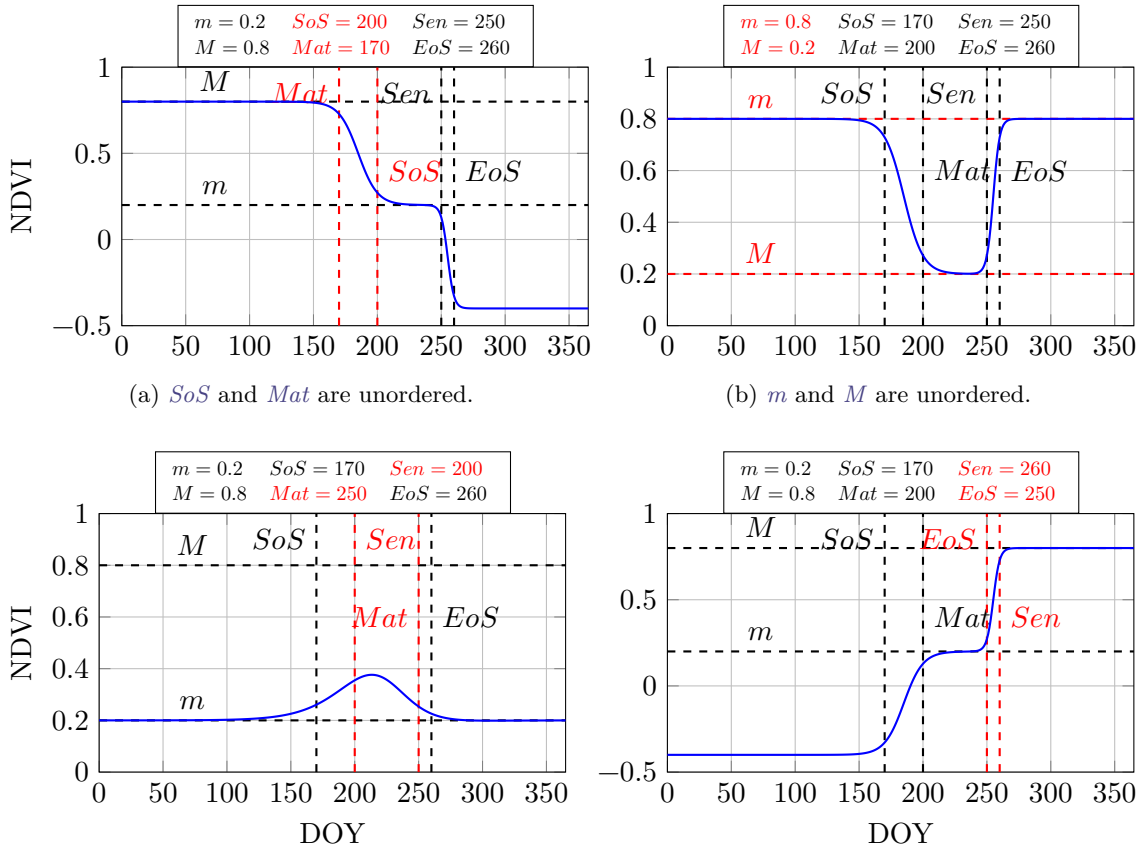
This requirement is implicit, since the function $\Omega(t, \boldsymbol{x})$ (Equation 9.4) is well-defined for $\boldsymbol{x} \in \mathbb{R}^6$ s.t. $SoS \neq Mat$ and $Sen \neq EoS$. However, un-ordered phenological parameters must be excluded, else the phenological model doesn't function as intended and can no longer accurately represent a growth-decay annual cycle of vegetation. Figure 9.4 shows examples of time series simulated with the phenological model with unordered parameters. Some simulated time series completely bend the model and no longer fit a growth/decay phenology type (Figure 9.4a, Figure 9.4b and Figure 9.4d). Other time series such as Figure 9.4c appear correct even though they were generated from unordered time series. In the case of Figure 9.4c, the parameter M is no longer consistent with the true NDVI maximum reached by the time series. Another issue with these kinds of time series is that they represent an alternate possible solution (with unrealistic parameters) to the fitting of the phenological model, i.e. they make the inversion problem ill-posed (see subsection 3.1.3).

As a consequence, additional constraints must be added to the latent space of Pheno-VAE. The phenological variables that are input to the UDD (the phenological model) must be ordered. This means that the latent distributions that these variables are tied to must be tweaked so that relevant samples are ordered. In this work, two random variables z_i and z_{i+1} or their associated distributions are qualified as *ordered* or *consecutive*, if their samples are always ordered: $z_i < z_{i+1}$[6][7]. Two ordered random variables are denoted $\mathrm{z}_i < \mathrm{z}_{i+1}$.

One straightforward method of ensuring that two distributions are ordered is to have the distributions on disjoint domains. Then, the distributions are ordered provided their domains are ordered (see Figure 9.5b). However this solution is too simplistic, and it doesn't suit situations for which distributions have the same domain, such as is the case for phenological

[6] *Order theory* is the branch of mathematics that studies binary relations, which, among other things, give meaning and properties to objects being smaller or greater than others [Dean, 2022; Russell, 1903].

[7] The meaning of *ordered random variables* is different in this work than in the literature. Ordered random variables are usually about characterizing the order of samples drawn from random variables, that are oftentimes of independent and identically distributed (i.i.d.) (e.g. order statistics) [Shahbaz et al., 2016]. In this Ph.D., they refer to a property of the random variables distributions themselves, i.e. it is not about finding the order of samples *a posteriori* to their drawing, but rather *a priori* characterizing their order from their associated distribution.



(a) *SoS* and *Mat* are unordered.

(b) *m* and *M* are unordered.

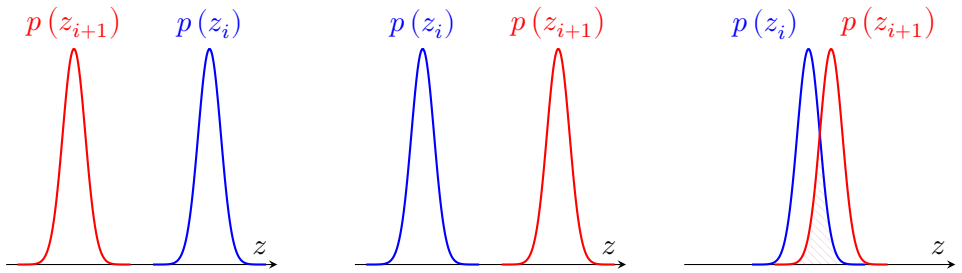
(c) *Mat* and *Sen* are unordered.

(d) *Sen* and *EoS* are unordered.

Figure 9.4: Examples of time series simulated by the phenological model with unordered phenological variables.

variables. In the general case, the marginal distributions of random variables can overlap[8] (see Figure 9.5c). Crucially, random variables with overlapping PDF cannot be ordered if they are independent: the area of their overlap is the probability of drawing unordered samples. As a consequence, to ensure the ordering of samples whose distributions intersect, some sort of dependence must be introduced. Therefore, adaptations are required for VAE, in which latent dimensions are usually independently sampled.

In the following are discussed three strategies for constraining latent distributions of consecutive phenological variables to be ordered appropriately.



(a) Unordered distributions on disjoint domain.

(b) Ordered distributions on disjoint domain.

(c) Distributions on common domain with overlapping densities.

Figure 9.5: Densities of two random variables that are supposed to be ordered $\mathrm{z}_i < \mathrm{z}_{i+1}$.

9.2.1 Penalizing out-of-order latent samples

A first straightforward possibility of enforcing order of latent samples would be to penalize samples that are unordered, e.g. with an additional loss term such as:

$$\mathcal{L}_{\text{unordered}}(z_i, z_{i+1}) = \text{ReLU}(z_i - z_{i+1}). \tag{9.7}$$

However such a solution isn't applicable in practice. Such a loss term doesn't change the independent drawing of latent variables, so it would merely promote latent distributions that minimize their overlap (see Figure 9.5b). As a consequence, the width of consecutive latent distributions would be encouraged to decrease, harming uncertainty quantification. They would also be prevented from being close, hampering the accuracy of the variable retrieval: the encoder would arbitrarily infer disjoint marginal distributions. This solution introduces an inductive prior of distribution disjointedness that is not necessarily assumed by the physical-based decoder. Furthermore it would also hamper training by introducing noise into the loss.

9.2.2 Inferring the distribution of the difference between two variables

A second method for ordering latent variables is to predict the greater variable as the sum of the smaller variable and an auxiliary positive random variable:

$$\mathrm{z}_{i+1} = \mathrm{z}_i + \Delta\mathrm{z}_{i+1}. \tag{9.8}$$

This solution tackles the shortcomings of the previous method: two consecutive random variables are no longer independent, which enables distributions with overlapping densities. Also, this method guarantees that the random variables are properly ordered. It is also compatible with usual VAE implementations. Although z_{i+1} is dependent from z_i, it can be

[8] The overlap can be defined as the area intersected by two probability distribution functions (PDFs). Distributions on the same domain $\mathbb{D}$ overlap if their domain of non-zero density overlap, i.e. $\{z \text{ s.t. } p(z_i) > 0\} \cap \{z \text{ s.t. } p(z_{i+1}) > 0\} \neq \emptyset$.

derived with a deterministic function of two independent random variables. This can be seen as an intermediary latent space (see subsubsection 6.5.2.3):

$$\begin{pmatrix} \mathrm{z}_{i+1} \\ \mathrm{z}_i \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta \mathrm{z}_{i+1} \\ \mathrm{z}_i \end{pmatrix}. \tag{9.9}$$

However, this method has a crucial limitation related to uncertainty quantification. The variance of the sum of two random variables is always equal or greater than the variance of these variables. In the general case the variance of the sum is:

$$\mathrm{Var}\left(\mathrm{x}+\mathrm{y}\right) = \mathrm{Var}\left(\mathrm{x}\right) + \mathrm{Var}\left(\mathrm{y}\right) + 2\mathrm{Cov}(\mathrm{x},\mathrm{y}) \tag{9.10}$$

but for $\mathrm{x} = \mathrm{z}_i$ and $\mathrm{y} = \Delta \mathrm{z}_{i+1}$ which are independent in a VAE latent space, it is simply expressed as:

$$\mathrm{Var}\left(\mathrm{x}+\mathrm{y}\right) = \mathrm{Var}\left(\mathrm{x}\right) + \mathrm{Var}\left(\mathrm{y}\right). \tag{9.11}$$

In fact, the PDF of the sum of random variables is the convolution product of the PDF of these variables, and the convolution of two densities results in a wider density (see Figure 9.6):

$$f_{\mathrm{x}+\mathrm{y}}\left(t\right) = \left(f_{\mathrm{x}} * f_{\mathrm{y}}\right)\left(t\right) = \int f_{\mathrm{x}}\left(\tau\right) f_{\mathrm{y}}\left(\tau - t\right) \mathrm{d}\tau \tag{9.12}$$

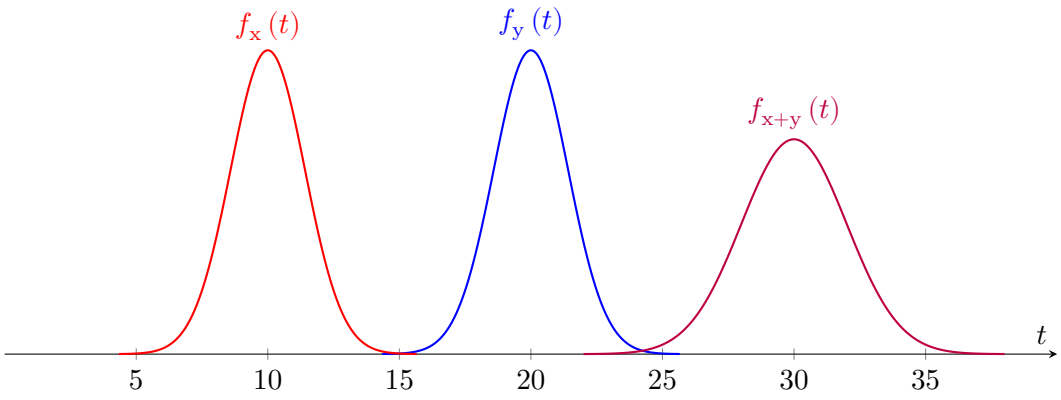


Figure 9.6: Densities of two random variables $\mathrm{x} \sim \mathcal{N}\left(10, 2\right)$ and $\mathrm{y} \sim \mathcal{N}\left(20, 2\right)$ and their sum $\mathrm{x}+\mathrm{y} \sim \mathcal{N}\left(30, 4\right)$.

This is a problem because this means that the predicted variance of the actual variable of interest z_{i+1} is fundamentally larger than the uncertainty of both $\Delta \mathrm{z}_{i+1}$ and z_i. This effect is amplified for multiple consecutive variables, such as is the case for the phenological dates. Using this method would arbitrarily impose that $\mathrm{Var}\left(SoS\right) < \mathrm{Var}\left(Mat\right) < \mathrm{Var}\left(Sen\right) < \mathrm{Var}\left(EoS\right)$.

9.2.3 Inferring the distribution of the maximum of two variables

To overcome the shortcomings of the two previous methods, it is proposed to use the distribution of the maximum of two successive variables as the distribution of the greater variable (see Figure 9.7). Let's assume two random variables z_i and z_{i+1} whose samples must be ordered $z_i < z_{i+1}$. Samples $z_i^\star$ and $z_{i+1}^\star$ are drawn independently from their respective distributions. $z_i^\star$ is first conserved as the sample of z_i. Then if $z_{i+1}^\star > z_i^\star$, $z_{i+1}^\star$ is kept as the sample of z_{i+1}, otherwise $z_i^\star$ is chosen instead:

$$z_{i+1} = \max\left(z_i^\star, z_{i+1}^\star\right). \tag{9.13}$$

To ensure a strict ordering, the following computation can be performed:

$$z_{i+1} = \max\left(z_i^\star + \varepsilon, z_{i+1}^\star\right), \tag{9.14}$$

with $\varepsilon > 0$. This procedure that enables to sample the maximum of two distributions is hereby referred as *sample rectification*. The analytical expression density of the maximum of random variables is provided in section C.1.

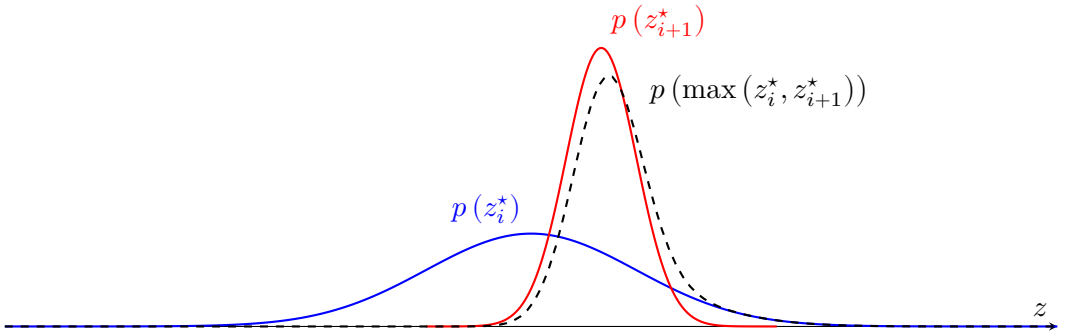


Figure 9.7: Distribution of the maximum of two Gaussian distributions.

This procedure ensures that samples are ordered no matter their distributions, and doesn't arbitrarily increase the variance of the greater variable. However, the inference of meaningful distributions can be hampered because ordered samples can be produced no matter the distributions. This is due to the rectification step taking place after the inference variational parameters and the sampling. For instance, when $\mathrm{z}^{\star}_{i+1} < \mathrm{z}^{\star}_{i}$ (see Figure 9.5a), this procedure will only produce $z_{i+1} = z_i$ (or $z_{i+1} = z_i + \varepsilon$).

Therefore, it is necessary to promote the inference of distributions that are at least *partially ordered*[9] (see Figure 9.5c). This is achieved through the addition on two additional constraints on the latent distribution, that ensure that the encoder outputs variational parameters that allow to satisfy those requirements.

To ensure that the latent distributions are at least partially ordered prior to the sample rectification step, it is proposed to force the expectation m_i of the latent variables z_i (alternatively the median or the mode) to be ordered as well (i.e. $m_{i+1} > m_i$). This amounts to constraining the variational parameters λ_i produced by the encoder so that the expectations m_i are ordered. In practice, this can be performed when one parameter of the variational distribution family controls the expectation, such as the mean parameter μ of Gaussians. In this work, truncated normal (TN) distributions are used, and their parameter μ controls the mode of the distribution, and therefore are suited to this method. The partial ordering of TN distributions is achieved by ordering the parameters $m_i = \mu_i$.

Firstly, the μ_i are rectified, similarly to the latent samples:

$$\mu_{i+1} = \max\left(\mu^{\star}_{i}, \mu^{\star}_{i+1}\right), \tag{9.15}$$

with $\mu^{\star}_{i}$ and $\mu^{\star}_{i+1}$ the mean parameters actually output by the encoder for the latent distribution of consecutive variables, and μ_{i+1} the rectified parameter for the greater variable. This rectification step ensures that the distributions associated with ordered variables are also partially ordered themselves (see Figure 9.5b and Figure 9.5c). This avoids producing ordered latent samples (with the sample rectification step) from unordered distributions (see Figure 9.5a)

Because this rectification step always produces partially ordered distributions, the encoder might never learn to actually output ordered distributions, hampering the ability to infer accurate and meaningful distributions. To encourage the encoder to infer such distributions, while only relying parsimoniously on the expectation rectification, a soft constraint is added

[9]Here, the notion of partial ordering refers to distributions that have a certain degree of overlap, but for which there is a non zero probability of predicting correctly ordered samples.

in the form of a regularization loss term, named the *order loss term*.

$$\mathcal{L}_{\text{order}} = \frac{1}{N_j} \sum_{j=1}^{N_j} \mu_{i_j} - \mu^\star_{i_j}. \tag{9.16}$$

The order loss sums the difference of the rectified parameter μ_{i_j} with the parameter $\mu^\star_{i_j}$ actually produced by the encoder[10], for all N_j variational parameters that have been rectified. This promotes the inference of ordered distributions directly by the encoder, i.e. $\mu_{i_j} = \mu^\star_{i_j}$. This loss term can be interpreted as an additional prior (a *learning bias*, see subsection 6.5.4) on latent distributions that the original KLD term doesn't enforce.

Using the maximum of consecutive variables to order the them does change their distribution (see appendix C.1 for the density of the maximum of random variables). The prior distribution $p(\boldsymbol{z})$ and the KLD loss term can both be expected to become harder to derive for such latent distributions. In the present case, the analytical expression of the KLD between the distribution of the maximum of TN and a uniform distribution is intractable. Therefore, it is advocated here to use the latent distribution without taking the ordering procedure into account in the computation of the prior and the KLD term. Nonetheless, since the analytical expression of the distribution of the maximum of two (or more) distributions is tractable, a NLL of this distribution can be computed.

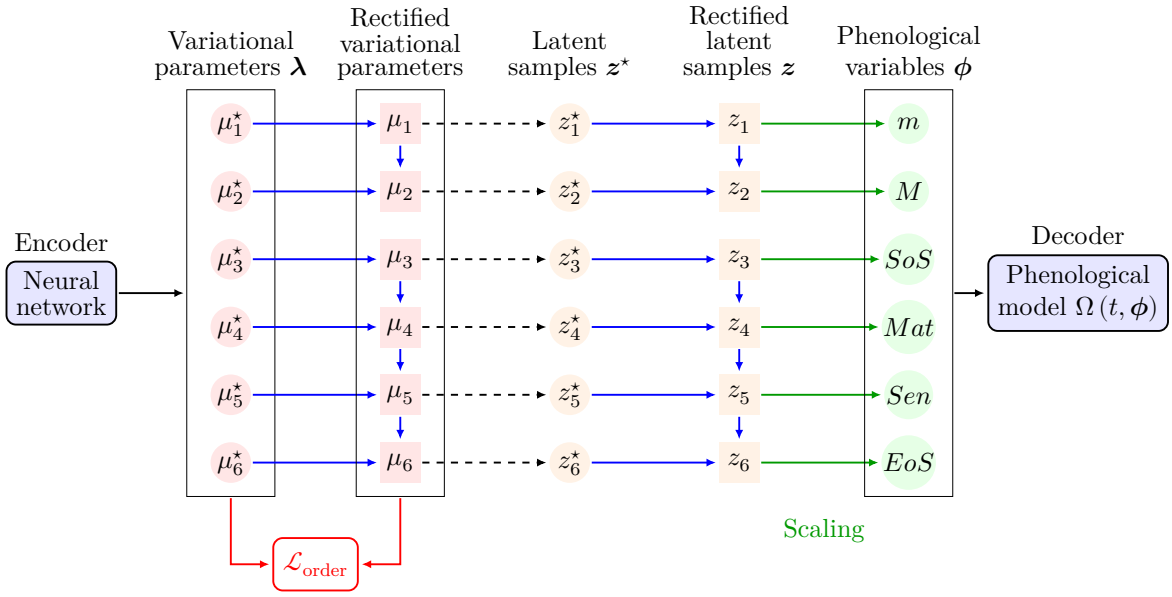


Figure 9.8: Sampling procedure of latent variables $\boldsymbol{z}$ in Pheno-VAE. Blue arrow indicate rectification operations, green arrow indicate a re-scaling and dashed arrow indicate reparameterization sampling. The encoder output $\boldsymbol{\sigma}$ is omitted.

In Pheno-VAE, this method of using the maximum of two distributions as the distribution of the greater of two successive random variables is used for ordering the latent variables tied to the phenological dates and the min/max of NDVI (see Figure 9.8). Thus, the loss function minimized during the training of Pheno-VAE is composed by the next three terms :

$$\mathcal{L}_{\text{Pheno-VAE}} = \mathcal{L}_{\text{rec}} + \beta \mathcal{L}_{\text{KLD}} + \mathcal{L}_{\text{order}}. \tag{9.17}$$

The loss components are:

- $\mathcal{L}_{\text{rec}}$ the Gaussian NLL reconstruction loss,
- $\mathcal{L}_{\text{KLD}}$ the KLD between the TN latent variables and the uniform prior.

[10] $\mu_{i_j} - \mu^\star_{i_j}$ is positive because of Equation 9.15.

- $\mathcal{L}_{\text{order}}$ term to promote ordered latent variables.

In practice, $\mathcal{L}_{\text{order}}$ converges to zero very fast, leaving only the two other terms in most of the training process.

9.3 Experiments

In this section, the experimental setup and the evaluation metrics used to evaluate the quality of the inferred phenological parameters are described, then the obtained results are presented.

9.3.1 Data-sets

Two data-sets are used to evaluate the performances of Pheno-VAE for phenological parameter retrieval. The first data-set is composed of real satellite observations of annual NDVI time series and is used for Pheno-VAE training and qualitative validation. The second data set is composed of simulated crop NDVI profiles. The construction of this data-set is proposed for three main reasons:

1. to perform a quantitative evaluation of parameter retrieval on a large scale data-set,
2. to assess the robustness of Pheno-VAE to the noise of complex satellite observations,
3. to compare the results of Pheno-VAE against supervised methods.

Examples of NDVI time series from both of data-sets are illustrated in Figure 9.11.

9.3.1.1 S2 data-set

The first data-set, denoted $\mathcal{D}_{\text{S2}}$, is composed of 10^6 annual time series of pixels from 31TCJ Sentinel-2 tile (Toulouse area in southern France, see subsubsection 2.1.3.2). The corresponding NDVI time series are computed from the spectral bands `B4` and `B8` (see Equation 9.1). The resulting time series describe different land cover classes which can be associated to the class legend used in the CES OSO land cover map Inglada et al. [2017a] (see subsection 1.2.2). Accordingly, a large number of time series do not represent vegetation classes following the double-logistic phenological model. The ideal behavior for Pheno-VAE on those data would be to have high reconstruction errors but high predicted uncertainty to compensate. Despite the availability of land cover class information, it must be remarked that such information is only used for validation purposes. Land cover classes do not intervene in the training procedure of Pheno-VAE, and all samples are taken into account within a single training. The distribution of the land cover classes in the data-set is detailed in Table 9.2.

The time series are acquired on irregular time intervals for two main reasons. Firstly, the two Sentinel-2 satellites have intersecting ground footprints and some locations get increased coverage. Secondly, cloud cover leads to inconsistent temporal sampling for each pixel on the ground. As a consequence, the number of valid observations in time series varies (see Figure 9.9). For each time series, a validity mask is available to denote the valid satellite observations. Since the encoder of Pheno-VAE learns from regular sampled time series, this mask is used to linearly interpolate raw time series to a common regular temporal grid.

9.3.1.2 Simulated data-set

While the above dataset $\mathcal{D}_{\text{S2}}$ of S2 NDVI time series enables to train Pheno-VAE, the phenological parameters associated with vegetation of observed pixels (when these pixels do contain vegetation) are unavailable. To mitigate the lack of data-set with reference phenology, a simulated data-set $\mathcal{D}_{\text{G}}$ is created. This data-set contains a large number of simulations obtained by the double-logistic model. A high number of combinations of input

Table 9.2: Distribution of the land cover classes composing the Sentinel-2 time series data-set. The class legend is taken from the OSO land cover map product [Inglada et al., 2017a].

Label	Percentage in data-set
Continuous Urban Fabric	0.6%
Discontinuous Urban Fabric	4.1%
Industrial and Commercial Units	3.1%
Road Surfaces	0.3%
Rapeseed	4.5%
Straw Cereals	9.9%
Protein Crops	2.5%
Soy	7.2%
Sunflower	33.0%
Corn	5.8%
Roots	0.2%
Intensive Grasslands	3.4%
Orchards	0.6%
Vineyards	1.8%
Broad-leaved Forests	6.7%
Coniferous Forests	5.5%
Grasslands	5.5%
Woody Moorlands	2.3%
Bare Rock	0.1%
Water Bodies	2.8%

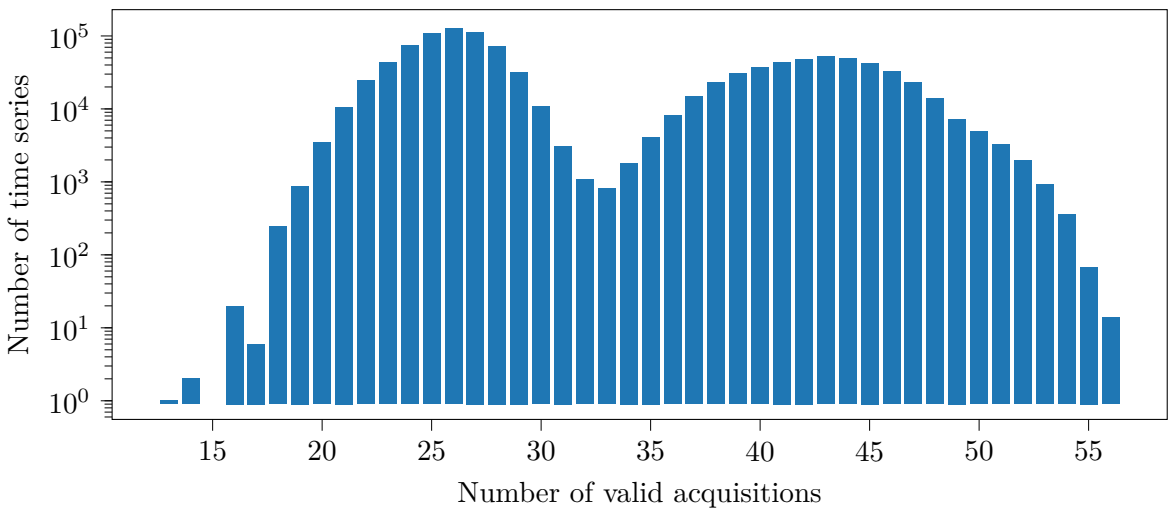


Figure 9.9: Distribution of the temporal acquisitions composing the Sentinel-2 time series data-set.

parameter values are generated by sampling the phenological parameters distributions given in Table 9.3. From these known phenological parameters, this data-set allows to compute quantitative metrics to validate the performances of Pheno-VAE. Additionally, generated data is also used for training Pheno-VAE to assess the influence of the training data on performance. Simulated data is therefore put into different data-sets: $\mathcal{D}_{\mathrm{G},train}$ for training, $\mathcal{D}_{\mathrm{G},valid}$ for validation and $\mathcal{D}_{\mathrm{G},test}$ for testing (see subsubsection 3.3.2.4).

To generate synthetic time series, sets of ordered phenological parameters are firstly sampled using uniform distributions, as depicted below. The double-logistic model is then used to produce the corresponding NDVI temporal profiles. To simulate the irregular temporal sampling, binary validity masks of real S2 time series are considered. These masks are applied on simulated time series to select time series values at certain dates. To generate more realistic time series simulations, a Gaussian noise of randomly sampled standard deviation $\sigma_n \sim \mathcal{U}(0, 0.1)$ is added to the NDVI profile. It accounts for epistemic uncertainty, as no real time series is perfectly described by the phenological model. The resulting time series are finally interpolated at a regular 5-days time grid. The data generation procedure is depicted in Figure 9.10.

The parameter sampling procedure is detailed in the following. The maximum value of the standard deviation σ_n of the noise level is set at 0.1 which corresponds to 10% of the maximum expected range for NDVI values. The interval between 0 (bare soil) and 0.4 (presence of vegetation) is set as for the range of m. The value of M is sampled relatively to m. In general, it can be considered that M is at least 0.3 higher than m for crop classes, and M can not be higher than 1.

A strategy is proposed to enforce the temporal order of the 4 phenological dates. A given phenological date is defined from the previous one. EoS is allowed to occur right after Senescence and up to 90 days later. Sen is defined in the same way with respect to Mat and Mat follows the same rationale with respect to SoS. To improve the plausibility of simulated time series, the SoS parameter is sampled such that it can represent the start of season date of winter and summer crops, instead of allowing it to be any date. To this end, an additional variable SoS_i for modeling the degree to which the simulated time series is a winter or a summer crop. SoS_i is used to adjust the lower bound of the sampling interval of SoS. The earliest DOY for SoS (for a winter crop) is set at 30 (end of January) and its latest DOY summer crop is set at 120 (late April). SoS_i and σ_n are additional variables of the generative process of synthetic data that are not inferred by Pheno-VAE, and not assessed during experiments.

It can be noted that this sampling strategy is equivalent to sampling the deltas between consecutive variables (see subsection 9.2.2).

Table 9.3: Distributions of reference phenological parameters sampled for NDVI time series simulation with the double-logistic model.

Parameter	Sampling interval	Parameter	Sampling interval
m	$\mathcal{U}(0, 0.4)$	M	$\mathcal{U}(m, 1)$
SoS_i	$\mathcal{U}(30, 120)$	SoS	$\mathcal{U}(SoS_i, SoS_i + 90)$
Mat	$\mathcal{U}(SoS, SoS + 90)$	Sen	$\mathcal{U}(Mat, Mat + 90)$
EoS	$\mathcal{U}(Sen, \mathrm{sen} + 90)$	σ_n	$\mathcal{U}(0, 0.1)$

Figure 9.11 shows some examples simulated NDVI time series of $\mathcal{D}_{\mathrm{G}}$ obtained by the proposed generation process, alongside real S2 NDVI time series of $\mathcal{D}_{\mathrm{S2}}$. Even though the $\mathcal{D}_{\mathrm{G}}$ is generated to be as realistic as possible, it is still different from the S2 data-set. Because of the uniform sampling of phenological dates in the synthetic data-set, there is more diversity in the phenology in $\mathcal{D}_{\mathrm{G}}$ than in $\mathcal{D}_{\mathrm{S2}}$. On the one hand, the $\mathcal{D}_{\mathrm{S2}}$ is biased by the samples that have been chosen among available real NDVI time series (observational bias). All samples belong to the same S2 tile so NDVI time series of pixels of the same type are highly correlated,

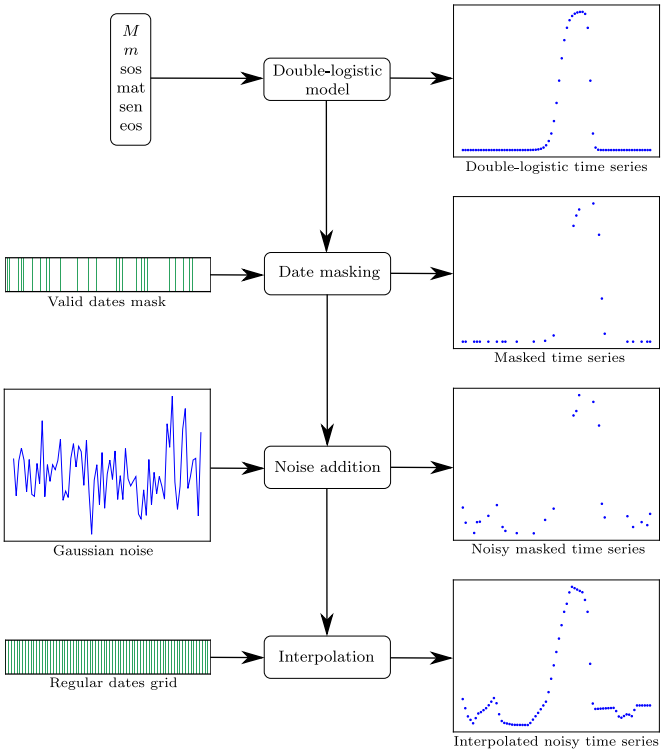


Figure 9.10: Procedure of generation of a data-set of synthetic NDVI Time series.

and cloud coverage similarly affects all time series. On the other hand, the synthetic data-set contains samples whose phenology may not be frequent in reality, or even phenology types that don't exist.

9.3.2 Experimental setup

Different experiments are carried out to assess the performances of Pheno-VAE. Firstly, the reconstructions of NDVI time series obtained by Pheno-VAE trained on the S2 data-set $\mathcal{D}_{\text{S2}}$ are visually evaluated in subsection 9.3.3. Secondly, a quantitative assessment of the performance of inference of phenological parameters Pheno-VAE is carried out using the simulated data-set $\mathcal{D}_{\text{G}}$, through two experiments:

1. the evaluation of the influence of β in subsection 9.3.4,
2. the comparison of Pheno-VAE with different standard parameter retrieval algorithms in subsection 9.3.5.

For the comparison, a multiple probabilistic supervised regression (MPSR) method, a curve fitting (CF) algorithm and a Markov Chain Monte Carlo (MCMC) algorithm (see subsubsection 9.3.2.3) are considered. These methods are compared to Pheno-VAE in terms of parameter estimation performance, and uncertainty quantification. All these methods perform the inversion of the phenological model on the NDVI time series of single pixels. The performances of CF, MCMC and MPSR are provided as an upper bound for parameter retrieval performances.

MCMC and CF strategies have critical computational limitations that limit their use in large-scale parameter retrieval applications. Specifically, these methods are *local* inversion approaches (see subsection 3.1.4) for which optimization must be performed for each pixel time series: inference is not amortized. In particular MCMC approaches are notoriously slow to converge and have a high computational cost. Conversely, the MPSR approach approximates a model inverse with a neural network. Its computational cost at inference is

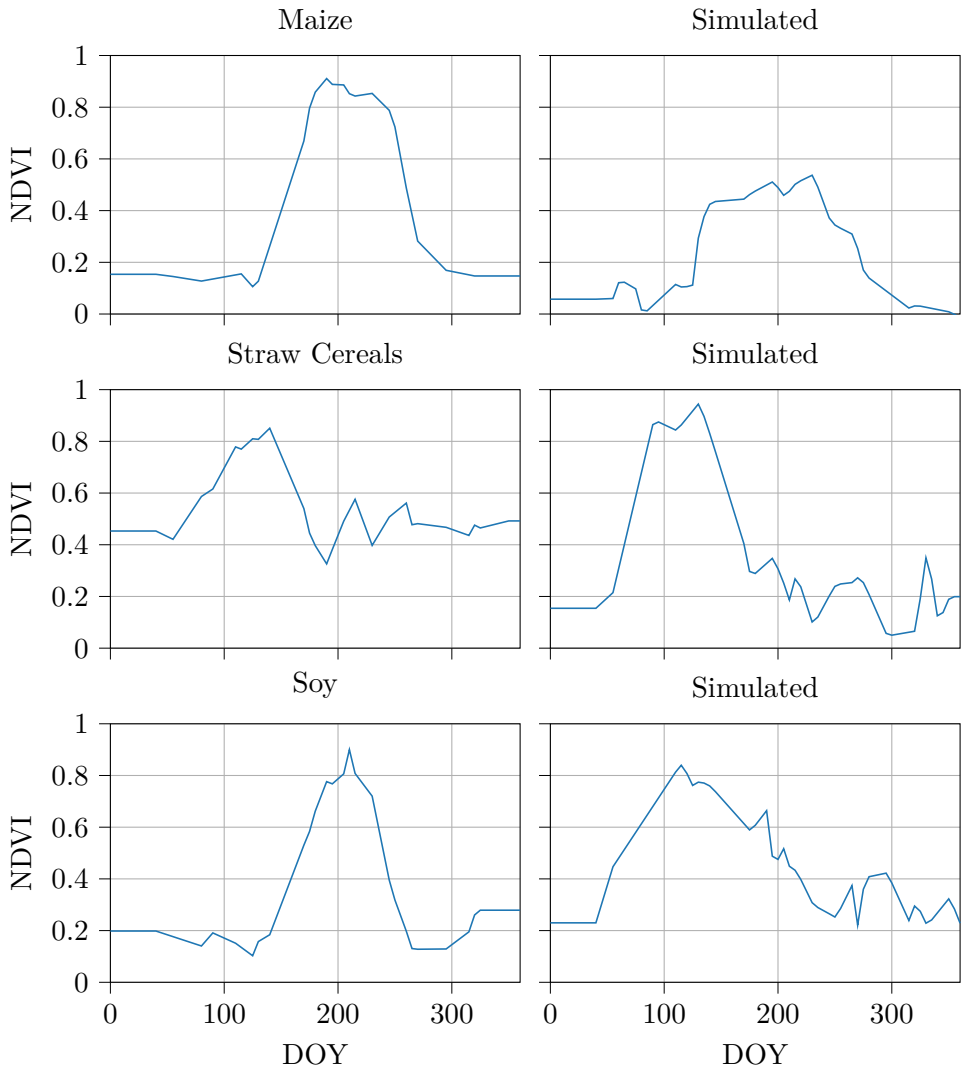


Figure 9.11: NDVI time series of samples of S2 data-set (left) and simulated data-set (right).

lesser than CF and MCMC since it doesn't require optimization, but only a forward pass. However, it is a supervised method that requires a labeled training data-set. This data-set can either be made from measurements (that are usually unavailable in sufficient quantity), or from simulated data such as what is performed in this chapter. But this means that the MPSR is sensitive to the choice of the sampling distributions of the simulations as was shown in Chapter 5. This dependence of MPSR performances on the phenological parameter distribution for the inversion of the phenological model is acknowledged, but not investigated in this chapter.

Besides CF, MPSR and MCMC methodologies, two training scenarios for Pheno-VAE are considered to evaluate the influence of the training data-set. In the first scenario Pheno-VAE models are trained on the S2 data-set, whereas training is performed with the synthetic data-set in the second. The characteristics of the different methods used for the qualitative assessment experiment are summarized in Table 9.4.

Table 9.4: Characteristics and hyper-parameters of each phenological model inversion method.

Method	Supervised	Training	Optimizer	Batch size	Learning Rate	Epochs	Latent samples	Point estimate	Parameter distribution
CF	✗	✗	✗	✗	✗	✗	✗	Deterministic	✗
MCMC	✗	✗	✗	✗	✗	✗	✗	Median	Full posterior approximate
MPSR	✓	Simulated Data-set	Adam	2048	5.10^{-4}	500	✗	Mode	Truncated Normal
Pheno-VAE-G	✗	Simulated Data-set	Adam	2048	5.10^{-4}	200	10	Mode	Truncated Normal
Pheno-VAE-S2	✗	S2 Data-set	Adam	2048	5.10^{-4}	200	10	Mode	Truncated Normal

9.3.2.1 Supervised neural network regression

A supervised neural network, trained on the simulated data-set $\mathcal{D}_{\text{G,train}}$ is proposed to perform regression (see section 3.3). This neural network uses the same architecture as the encoder of Pheno-VAE (see Figure 9.3): it takes interpolated NDVI time series as input and outputs the mean and variance of the TN distributions associated to the 6 phenological parameters (up to an affine transformation). As such, this approach estimates a distribution simultaneously for all phenological variables. It is thus referred as MPSR. The loss function of for training the neural network is the NLL of ordered TNs (see appendix C.1). Knowing the phenological parameter values of the synthetic data sets, the NLL compares the phenological distributions estimated from the regression algorithm against the known phenological parameters. Since the MPSR uses the same architecture as Pheno-VAE encoder, the model complexity will not influence comparative results.

9.3.2.2 Non-linear least squares regression

The CF algorithm solves a non linear least squares problem for each NDVI time series, with a trust region reflective algorithm (TRRA) Coleman and Li [1996] (see subsubsection 3.2.1.3). This method can take the boundaries of the parameters into account. Although it is not a Bayesian approach, the CF algorithm outputs a covariance matrix that can be used to estimate prediction intervals, along with the parameters estimates. Unfortunately, as the inversion is frequently ill-conditioned, the estimated parameters covariances often diverge. Thus prediction intervals estimation is discarded with this method. Contrary to other inversion methods presented here, the CF requires an initial guess z_0 on the parameters (i.e. it requires more prior information). The initial guess used to fit the phenological model on NDVI time series is detailed in Table 9.5. This method was implemented by using the `curve_fit` function of Python's scipy.optimize library.

Table 9.5: Initial guess of the phenological parameters for the curve fitting algorithm.

Parameter	Initial guess	Parameter	Initial guess
m	$\min(y_i)$	M	$\max(y_i)$
SoS	$\arg\max(y_i) - 30$	Mat	$\arg\max(y_i) - 15$
Sen	$\arg\max(y_i) + 15$	EoS	$\arg\max(y_i) + 30$

9.3.2.3 MCMC inference

MCMC algorithms are typically used in Bayesian inference to approximate an intractable posterior distribution by sampling (see subsection 6.3.1). As such, it can be used to estimate the posterior distribution of phenological variables, in the phenological model inversion problem. Following the methodology of [Gao et al., 2021], Hamiltonian Monte Carlo as per the NUTS[11] algorithm [Homan and Gelman, 2014] is used, as implemented in the NumPyro library [Bingham et al., 2019; Phan et al., 2019]. To implement Bayesian inference through MCMC, the likelihood function for the observed data is defined using the double-logistic model. At inference, NDVI time series irregularly sampled are injected into MCMC algorithm i.e. no interpolation to a regular grid is performed. As prior distributions, the uniform distributions described in Table 9.3 are chosen.

9.3.2.4 Evaluation metrics

The accuracy of the retrieved parameters and their predicted uncertainties are evaluated on the synthetic testing data-set $\mathcal{D}_{\text{G}}$, with 10000 samples. The mean absolute error (MAE) (see Equation 3.8) between the inferred phenological parameters and their reference value in the testing data-set is used as estimation metric. While CF directly predicts parameter estimates, Pheno-VAE, MPSR, and MCMC predict distributions, and a point estimate is necessary to compute the MAE. The point estimates for these different methods are detailed in Table 9.4. As MAE is sensitive to outliers, box-plots of the absolute errors are provided in subsection E.3.2 for complementary result interpretation.

Prediction intervals for phenological variables are estimated from inferred distributions: the sampled distributions for MCMC, the truncated Gaussian distributions inferred with MPSR and in the latent space of Pheno-VAE. To assess the quality of these intervals, two prediction intervals metrics are used:

- The mean prediction interval width (MPIW) (see Equation 6.13) — because it is sensitive to outliers, box-plots of the prediction interval widths are provided subsection E.3.5.

- The prediction interval coverage probability (PICP) (see Equation 6.14). It measures the frequency of the model parameters true value being inside the prediction interval, and its value should be as close to the confidence level as possible.

In the following, these metrics are computed for prediction intervals with a selected 90% confidence level, by using the 5^{th}-95^{th} percentile intervals. Results obtained with different confidence levels are shown in subsection E.3.4 and subsection E.3.3.

These three evaluation metrics are computed for Pheno-VAE and MPSR by using a k-fold cross-validation procedure (see 3.3.2.4), with $k = 6$. For MCMC, metrics are independently obtained on k subsets of the testing data-set $\mathcal{D}_{\text{test}}$. The averages and standard deviations of the results on those subsets are computed.

[11] No-U-Turn Sampler.

9.3.3 Evaluation of the reconstruction results

To assess the performances of Pheno-VAE, a visual evaluation is presented in Figure 9.12. This figure shows the reconstruction of different S2 NDVI time series obtained by the Pheno-VAE model trained on S2 data. For each example, the estimated phenological parameter distributions are also illustrated. In most cases shown here, the setting $\beta = 0$ imposes that no prior information from the data-set is incorporated.

In general, the error and variance of reconstructions are both low for temporal profiles well-characterized by the phenological model. The estimated phenological distributions seem well centered on likely phenological parameters. Figure 9.12a shows NDVI time series of a pixel of corn, the inferred phenological distributions and the reconstruction of its mode. The reconstruction curve is observed to accurately match the original time series. The distributions of phenological dates characterize well the growth and decay phases of this summer crop.

The influence of β can be evaluated by comparing the results observed in Figure 9.12a and Figure 9.12b. The same NDVI time series of a corn pixel is taken as input by two Pheno-VAE models with different values of β. The modal reconstructions are very similar. With increasing β, the phenological distributions widen, and the variance of reconstructions increases. This is coherent with the influence of the KLD loss terms, that discourages narrow latent densities. With both results well matching the original NDVI time series, the choice of β is to be made considering the prediction interval metrics.

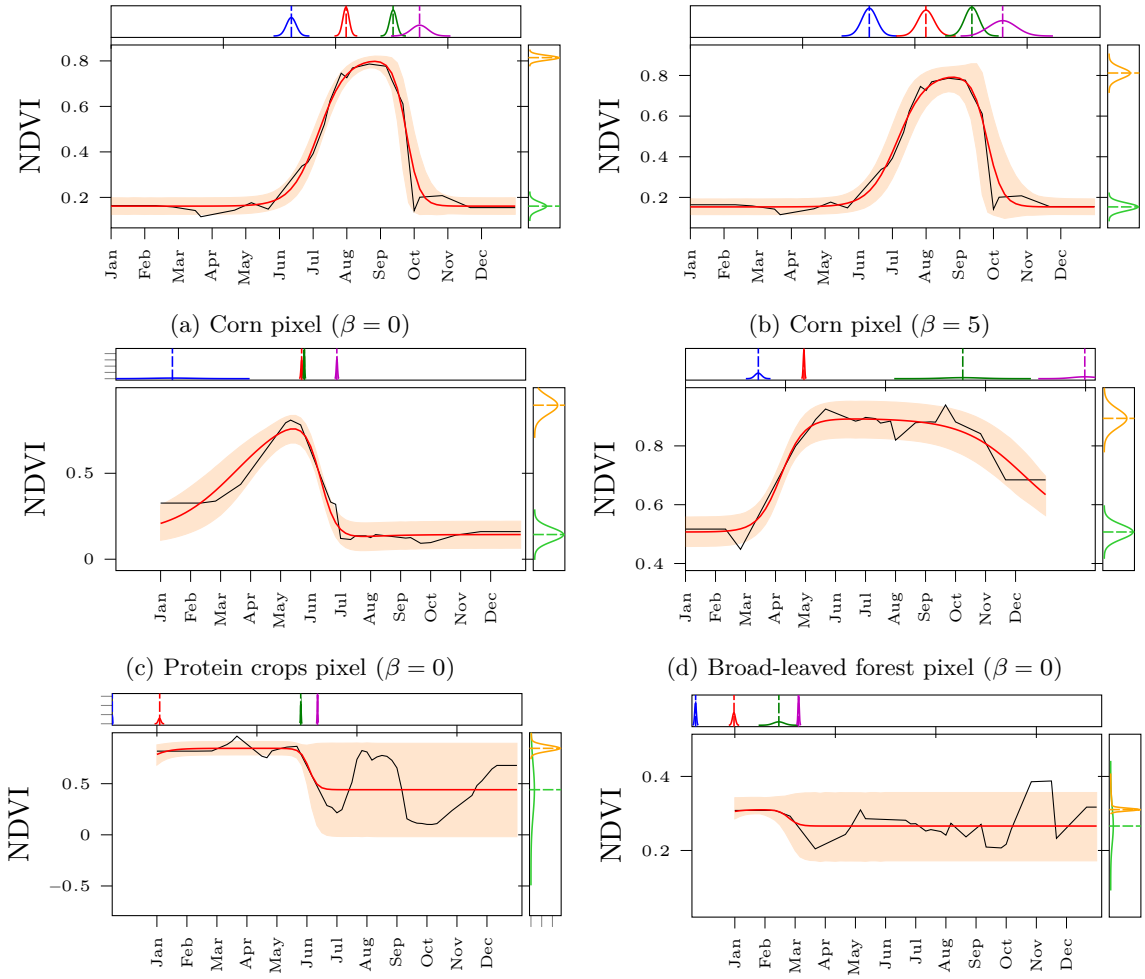
On Figure 9.12c, a protein crop time series shows how the presence of data gaps can lead to bad phenological parameter estimation. In this figure, the phenological cycle is easily identifiable. However, bad weather in winter led to a lack of data points for the first two months, and the backward extrapolation of points at preprocessing has kept the NDVI artificially constant, at a higher value than after harvest. As the encoder of Pheno-VAE doesn't take into account the temporal information, here the reconstruction is disrupted by the gap-filling step. This extrapolation artifact made the input time series not well described by the phenological model at the beginning of the year. The *SoS* estimate is inaccurate, yet the distribution large spread indicates greater uncertainty. This bad inference of the *SoS* seems to have prevented a good estimation of the maturity date as well, with this time a narrow distribution. Nonetheless the senescence and end of season seem well inferred. Similarly with a broad-leaved forest time series (Figure 9.12d), senescence and end of season distributions are not well positioned due to interpolated data points at the end of year. These results show that the gap-filling preprocessing task can lead to wrong parameter estimations when long data gaps include key phenological dates. This highlights the need for encoder architectures that mitigate the need for inputs interpolated on constant grids [Bellet et al., 2024; Dumeur et al., 2024]. However, the study and incorporation of such neural networks is out of the scope of this Ph.D.

In Figure 9.12e, there are several crops in the pixel, and the NDVI time series shows several phenological cycles. As the model can only take one cycle into account, it only fits the largest, and takes the average of the remaining signal. The distribution of the minimum of NDVI is very large, indicating uncertainty.

In Figure 9.12f, the phenological model doesn't suit at all the NDVI time series of a dense urban area pixel. Therefore, reconstruction errors are high. Still, phenological distribution variances increase to take this epistemic uncertainty into account. These results show that large uncertainties could be associated to the model discrepancy with the data.

Another remark is that, inferred marginal phenological distributions sometimes show significant overlap. This highlights the interest of the proposed order constraints on the latent distributions, as reconstructions are consistent with the phenological model, and variables constraints are always respected.

More reconstruction examples are available in subsection E.3.1.



(a) Corn pixel ($\beta = 0$)

(b) Corn pixel ($\beta = 5$)

(c) Protein crops pixel ($\beta = 0$)

(d) Broad-leaved forest pixel ($\beta = 0$)

(e) Rapeseed pixel ($\beta = 0$)

(f) Dense urban area pixel ($\beta = 0$)

Figure 9.12: Reconstruction and distributions of phenological parameters from the encoding of the NDVI time series by Pheno-VAE trained on S2 data-set. Central quadrants, S2 NDVI time series (black), reconstructions from the modes of phenological parameters distributions (red), and reconstruction 5^{th}-95^{th} prediction interval - Upper quadrants: TN distributions of the 4 phenological dates, *SoS* (blue), *Mat* (red), *Sen* (dark green), *EoS* (magenta) - Right quadrants: TN distributions of *M* (orange), and *m* (light green) - Upper and right quadrants: distribution densities are in solid lines, distribution modes are in dashed lines.

9.3.4 Influence of the KLD loss term on Pheno-VAE performances

The impact of the KLD term is studied by comparing results obtained by using different β values. In this experiment, Pheno-VAE is trained with samples from the S2 data-set. The prediction interval metrics presented here are derived for a confidence level of $1-\alpha = 0.9$.

Table 9.6: Evaluation performances obtained on a simulated data-set for different Pheno-VAE models trained on the S2 data-set, and for various KLD loss coefficients β. Prediction intervals are derived from phenological distributions with a confidence level $1-\alpha = 0.9$.

(a) Mean Absolute Error (the lower the better).

Exp.	*M*	*m*	*SoS*	*Mat*	*Sen*	*EoS*
Pheno-VAE-S2, $\beta = 0$	**0.05 ± 0.00**	**0.02 ± 0.00**	**11.13 ± 0.46**	**10.22 ± 0.08**	**11.01 ± 0.47**	**13.35 ± 0.52**
Pheno-VAE-S2, $\beta = 1$	**0.05 ± 0.00**	**0.02 ± 0.00**	11.82 ± 0.27	10.38 ± 0.33	11.61 ± 0.65	13.48 ± 0.69
Pheno-VAE-S2, $\beta = 2$	**0.05 ± 0.00**	**0.02 ± 0.00**	11.93 ± 0.60	10.58 ± 0.25	12.15 ± 0.60	14.75 ± 0.97
Pheno-VAE-S2, $\beta = 5$	0.07 ± 0.00	**0.02 ± 0.00**	14.87 ± 0.21	14.37 ± 0.61	18.37 ± 0.75	18.69 ± 0.47

(b) Prediction Interval Coverage Probability (the closer to 0.9 the better).

Exp.	*M*	*m*	*SoS*	*Mat*	*Sen*	*EoS*
Pheno-VAE-S2, $\beta = 0$	**0.67 ± 0.01**	0.95 ± 0.01	0.34 ± 0.05	0.25 ± 0.03	0.34 ± 0.04	0.58 ± 0.02
Pheno-VAE-S2, $\beta = 1$	0.60 ± 0.01	0.95 ± 0.01	0.53 ± 0.02	0.48 ± 0.02	0.55 ± 0.01	0.71 ± 0.02
Pheno-VAE-S2, $\beta = 2$	0.61 ± 0.02	0.94 ± 0.01	0.64 ± 0.02	0.56 ± 0.01	0.64 ± 0.01	0.76 ± 0.03
Pheno-VAE-S2, $\beta = 5$	0.63 ± 0.03	**0.92 ± 0.01**	**0.77 ± 0.01**	**0.69 ± 0.02**	**0.69 ± 0.02**	**0.83 ± 0.01**

(c) Mean Prediction Interval Width (the lower the better).

Exp.	*M*	*m*	*SoS*	*Mat*	*Sen*	*EoS*
Pheno-VAE-S2, $\beta = 0$	0.12 ± 0.01	0.13 ± 0.00	**14.69 ± 2.85**	**8.81 ± 1.11**	**13.75 ± 1.01**	**30.60 ± 1.83**
Pheno-VAE-S2, $\beta = 1$	**0.11 ± 0.00**	**0.12 ± 0.00**	22.97 ± 1.38	18.24 ± 1.05	23.35 ± 1.18	36.60 ± 2.38
Pheno-VAE-S2, $\beta = 2$	**0.11 ± 0.00**	**0.12 ± 0.00**	27.93 ± 1.54	22.81 ± 0.75	28.43 ± 1.53	43.30 ± 3.10
Pheno-VAE-S2, $\beta = 5$	0.16 ± 0.00	**0.12 ± 0.00**	41.79 ± 1.64	38.24 ± 1.65	42.18 ± 1.36	59.64 ± 2.30

As previously observed, the KLD term tends to increase the dispersion of the phenological parameters distributions. The MPIW (Table 9.6c) and prediction interval width (PIW) (Figure E.12)) increases for the phenological dates along with β and consequently the PICP (Table 9.6b) also increases.

The MAE results (Table 9.6a) tend to increase along with β, decreasing performance, although the distributions of the absolute errors (Figure E.6) only worsen significantly above a certain threshold of β. These results corroborate that the hyper-parameter β must be selected by using an independent validation data-set. For the prediction intervals to be informative, the KLD term needs to be high enough, while keeping it below a certain threshold ensures that precision is acceptable.

Also, different performances are obtained for the different phenological parameters. The minimum of NDVI *m* is the best estimated parameter, as with simulated time series, a large part of available data points are around the value of the minimum — although, it is so well estimated that its prediction interval almost always contains it, overshooting the PICP $= 1-\alpha$ target. The parameter *M* is more challenging to estimate than *m*. The value of the true maximum of the phenological model can differ from the parameter *M* when *Mat* and *Sen* are close. The highest errors are obtained on phenological dates, most certainly because of the gap-filling problem highlighted with reconstruction results (such as with Figure 9.12c and Figure 9.12d). This limitation is more visible in MPIW values obtained for *SoS* and *EoS* than *Mat* and *Sen*. This is because the Pheno-VAE is confronted with more severe extrapolation aberrations at both ends of the time series than in the middle, where interpolation is better, with higher temporal availability in the original time series.

In the following, the setting $\beta = 2$ will be used, as it increases the PICP without degrading too much the MPIW and the MAE.

9.3.5 Quantitative assessment of Pheno-VAE

Quantitative results obtained by Pheno-VAE trained on S2 data-set, Pheno-VAE trained on the synthetic data-set, MCMC and MPSR by inferring the phenological distributions of the simulated data-set are compared here. Obtained results are presented in Table 9.7.

Table 9.7: Evaluation performances obtained on a simulated data-set for different experiments of inversion of the phenological model. Prediction intervals are derived from phenological distributions with a confidence level $1 - \alpha = 0.9$.

(a) Mean Absolute Error (the lower the better).

Exp.	*M*	*m*	*SoS*	*Mat*	*Sen*	*EoS*
Pheno-VAE-S2, $\beta = 2$	0.05 ± 0.00	0.02 ± 0.00	11.93 ± 0.60	10.58 ± 0.25	12.15 ± 0.60	14.75 ± 0.97
Pheno-VAE-G, $\beta = 2$	0.06 ± 0.00	0.02 ± 0.00	8.89 ± 0.53	10.51 ± 0.49	10.59 ± 0.52	9.23 ± 0.26
MCMC	**0.03 ± 0.00**	0.02 ± 0.00	7.18 ± 0.70	9.57 ± 0.95	9.93 ± 1.00	10.42 ± 1.18
MPSR	0.04 ± 0.00	**0.01 ± 0.00**	**6.69 ± 0.03**	**7.54 ± 0.05**	**6.91 ± 0.05**	**6.70 ± 0.07**
CF	0.07 ± 0.00	**0.01 ± 0.00**	7.58 ± 1.07	11.74 ± 1.20	10.75 ± 1.20	7.37 ± 1.25

(b) Prediction Interval Coverage Probability (the closer to 0.9 the better).

Exp.	*M*	*m*	*SoS*	*Mat*	*Sen*	*EoS*
Pheno-VAE-S2, $\beta = 2$	0.61 ± 0.02	0.94 ± 0.01	0.64 ± 0.02	0.56 ± 0.01	0.64 ± 0.01	0.76 ± 0.03
Pheno-VAE-G, $\beta = 2$	0.67 ± 0.01	0.99 ± 0.00	0.67 ± 0.05	0.60 ± 0.01	0.66 ± 0.01	0.77 ± 0.02
MCMC	0.89 ± 0.01	0.86 ± 0.01	0.84 ± 0.01	0.85 ± 0.01	0.83 ± 0.01	0.83 ± 0.01
MPSR	**0.90 ± 0.01**	**0.90 ± 0.01**	**0.89 ± 0.00**	**0.89 ± 0.00**	**0.89 ± 0.01**	**0.88 ± 0.00**

(c) Mean Prediction Interval Width (the lower the better).

Exp.	*M*	*m*	*SoS*	*Mat*	*Sen*	*EoS*
Pheno-VAE-S2, $\beta = 2$	**0.11 ± 0.00**	**0.12 ± 0.00**	27.93 ± 1.54	**22.81 ± 0.75**	28.43 ± 1.53	43.30 ± 3.10
Pheno-VAE-G, $\beta = 2$	0.14 ± 0.01	0.14 ± 0.00	**21.02 ± 0.76**	23.25 ± 1.32	27.09 ± 1.16	25.23 ± 0.80
MCMC	0.13 ± 0.01	0.05 ± 0.00	22.13 ± 1.75	25.03 ± 1.94	**22.74 ± 1.79**	**21.50 ± 2.29**
MPSR	0.16 ± 0.00	0.06 ± 0.00	27.70 ± 0.30	29.91 ± 0.25	27.81 ± 0.43	26.36 ± 0.40

Best overall performances are obtained by the MCMC, for which the distribution of absolute errors is the lowest (Figure E.6), despite having a little higher MAE (Table 9.7a) than MPSR. MCMC also attains PICP that is close to the confidence level α (Table 9.7b and Figure E.9), with prediction intervals significantly narrower than other presented methods. Phenological distribution inference is not limited by a distribution family prior and directly samples phenological distributions, contrary to the other methods studied here. It is also not affected by missing data gaps because MCMC do not require regularly temporal input data. The results of MCMC could be improved by increasing the number of distribution samples and steps, at the expense of greater computation costs. Despite the promising MCMC results, its computing time required is much longer for MCMC than deep learning methods (see Table 9.8). It justifies why such approach can not be applied on operational parameter retrieval applications.

MPSR, has absolute errors that are a little higher and larger prediction intervals, however it has the best PICP, that is the closest to the confidence level α for all phenological variables. Those good results are expected, considering that it is a supervised method, with the training data-set being very similar to the testing data-set. Furthermore its loss doesn't rely on reconstruction, and therefore isn't affected by the irregular temporal sampling of real S2 time series.

The CF approach predicts phenological parameters with a MAE between that of MCMC and MPSR, except for *Mat* and *Sen* which are on par with the inference of Pheno-VAE. However this method was observed to be less reliable than the other presented here, as it didn't converge to a solution for about 5% of the time series (the results presented in Table 9.7a excluded those failed predictions).

The results of Pheno-VAE are less good than MCMC and MPSR. It has higher MAE, and despite similar prediction interval sizes, it underestimates uncertainty with lower PICP.

Results also show different behaviors for the two Pheno-VAE trained on different data-sets. As expected, slightly better results are obtained when Pheno-VAE is trained on simulated data. A greater performance drop is observed for *EoS*. This is because of a discrepancy between both data-sets. In the simulated data-set, there is more diversity in the phenological parameters, because of the uniform sampling to generate it. Even if real validity masks from the S2 data-set are used, they are not correlated to phenology, as it is the case for real data. In the S2 data-set, a smaller diversity of combinations of phenological variables is available. In this data-set, the end of season of real crops can happen when there are clouds, more than in the simulated data-set.

The drop in performances is much less significant compared to regression and MCMC, despite training on samples that don't follow the phenological model. The Pheno-VAE trained on the synthetic data-set benefits from being evaluated on a similar simulated data-set. This unfair advantage could be mitigated by evaluating the performances of Pheno-VAE on real S2 NDVI time series data-set, with available ground truth of phenological stages. Unfortunately, such a data-set was not available at the time of this study.

MCMC and MPSR show similar performances, despite being very different methods. This hints that given the simulated data-set and the double-logistic model, there is not much performance improvement to expect from the inference experiment, even with other setups. The regression yields on phenological dates 7-day MAE, with 90% PICP and 28 days MPIW. These are good results considering irregularly sampled time series that are interpolated to a 5-day grid. For Pheno-VAE to get performances closer to this, there is a need to improve on the ability of the encoder neural network to take temporal structure of time series into account. To minimize the impact of the gap-filling pre-processing step, different solutions could be considered. For instance, the reconstruction loss could be modified to only take valid observations into account. The encoder network architecture could be replaced to allow to learn from irregularly sampled time series such as with transformers.

Table 9.8: Approximate training and inference time for each setup on computing environment described in section A.2.

Method	CF	MCMC	MPSR	Pheno-VAE (Sim)	Pheno-VAE (S2)
GPU usage	✗	✗	✓	✓	✓
Training	✗	✗	15 min	15 min	15 min
Inference per time series	10^{-4} s	10 s	10^{-5} s	10^{-5} s	10^{-5} s

9.3.6 Ablation study of the latent distribution maximum sampling techniques

An ablation study for the strategy presented to incorporate temporal order in latent variables is performed with Pheno-VAE. The three sub-tasks proposed for enforcing the maximum of successive distributions as the distribution of the greater of consecutive variables, are evaluated: the rectification of the variational parameter μ (Equation 9.15), latent samples rectification (Equation 9.13), and the order loss (Equation 9.16). When any of these steps is removed, convergence of the objective function the validation data-set is observed to be slower. It also often leads to sub-optimal models that only order distributions by making them identical. Moreover, simply removing the latent sample rectification leads the Pheno-VAE to infer latent model parameters that fit the data but no longer have physical meaning (with for instance a prediction of the *SoS* date being after the *EoS* date).

9.4 Conclusion

In this chapter, the integration of physical models for representation learning in VAE is applied to perform the inversion of a phenological model from NDVI time series. This application has required the incorporation of additional constraints in the latent space, to account for order relationships between phenological variables. The training of the subsequent Pheno-VAE is robust to samples which do not correspond to the physical model (pixels without vegetation).

Despite using a simple neural network architecture, preliminary results are encouraging. Nonetheless, the inference error and prediction intervals of Pheno-VAE fall behind certain other methods in the current configuration. It is hypothesized here that it is because of the reliance of Pheno-VAE on reconstruction errors for training. The simple MLP architecture of the encoder requires a fixed temporal grid for input time series. The necessary interpolation of irregular time series disturbs the input to the model, and leads to reconstruct an altered signal. Performance could be improved by enhancing the encoder architecture with inductive biases that enable to take into account the temporal structure of the data (attention mechanisms, recurrent architectures), and that allow to take irregularly sampled time series as input [Bellet et al., 2024]. Furthermore, the exploitation of the spatial context of satellite data may improve parameter retrieval on individual pixels, such as with convolutional neural networks, like was performed in Chapter 8 — although additional inductive biases might be necessary.

Part V

Conclusion

Chapter 10

Conclusion and perspectives

10.1 Conclusion

The purpose of this thesis was to develop methodologies for deriving meaningful, interpretable representations of vegetation of continental surfaces from remote sensing observations. In particular, physical variables were identified as ideal representations, since they inform about the nature of ground surfaces. They are general enough to be useful for various downstream tasks, and have an intrinsic value by themselves. Such physical variables are typically the inputs of physical models, and as such, retrieving those variables is an inversion problem. In particular, there are well-known models that link the state of vegetation through bio-physical variables with remote sensing observations.

The contributions of this thesis were detailed in Part II, Part III and Part IV. In Part II, a crucial dependence on simulated data-sets of classical supervised deep learning approaches for vegetation bio-physical variables retrieval has been identified. Part III developed a methodology based on representation learning and variational autoencoders (VAE) for performing unsupervised model inversion. Finally, Part IV presented the results of the application of the proposed methodology on two well-known remote sensing inversion problems: PROSAIL and a phenological model.

In Part II, a differentiable and parallelized implementation of the PROSAIL radiative transfer model (RTM) has been detailed. The computational cost of this model for simulating Sentinel-2 (S2) reflectances was lowered by using under-sampling, without altering simulation accuracy beyond the atmospheric correction accuracy. Then, after introducing classical model inversion methods, the *de facto* standard neural networks based Simplified Level 2 Product Prototype Processor (SL2P) has been discussed. This model enables the large scale retrieval of canopy variables, by preforming the supervised inversion of PROSAIL. As was demonstrated in this part, the performance of supervised regression depends heavily on the simulations used in the training data-set. Specifically, it was shown that the version of PROSAIL used, along with the distribution of the model input variables and even the correlations between them had an influence on the inversion performance. This was identified as a key limitation, because the distributions of all the variables and their correlations involved in the simulation are not well-known.

In Part III, Bayesian inference methods were studied as probabilistic methods for deriving representations from remote sensing data. In particular, a focus was put on VAE, which are at the intersection of deep learning and variational inference. Such models learn to predict generative representations of data in a self-supervised manner. Different methods for incorporating prior information in VAE were investigated. In particular, it was proposed to constrain the latent representations to be physical variables by combining VAE with physical models. By incorporating a physical model into the decoder of a VAE, the latent variables were semantically bound to the input of the model. Such a physics-integrated VAE performed the inversion of the model in the decoder as a representation learning technique.

This methodology can be applied to a wide variety of models, as long as they are differentiable and that forward passes are not too computationally expensive. Furthermore, the retrieved variables are described by a probability distribution rather than a simple estimate. Additionally, a variety of techniques for incorporating a deterministic physical model into the framework of a Bayesian method were proposed. Crucially, once trained, the decoder of the VAE is discarded, and the encoder can be used for probabilistic inference at large scale.

Part IV presented the results of the application of the VAE-based inversion methods with two physical models. In the first application, the differentiable implementation of PROSAIL was integrated into the decoder of a VAE with the so-called PROSAIL-VAE. The latent variables of PROSAIL-VAE are the leaf and canopy variables of the PROSAIL model. As opposed to supervised approaches which must be trained on pre-simulated datasets, PROSAIL-VAE was trained directly on S2 data. Using in-situ measurements of leaf area index (LAI) and canopy chlorophyll content (CCC), the PROSAIL-VAE approach was compared to SL2P and multiple probabilistic supervised regression (MPSR), which are supervised neural network approaches for inverting PROSAIL. PROSAIL-VAE showed superior performance on the retrieval of these variables on the available data, with both accurate estimates and meaningful uncertainty quantification. Additionally, the aggregate posterior distribution of PROSAIL variables was estimated with S2 images with PROSAIL-VAE. This enabled to estimate correlations between PROSAIL variables. Some variables were likely better estimated than others, which highlighted the need for more validation using in situ measurements of vegetation variables. Different configurations of PROSAIL-VAE that incorporated prior knowledge differently were also investigated.

In the second application of the proposed methodology, a double-logistic phenological model was integrated into the decoder of a VAE with the so-called Pheno-VAE. This physical model related phenological dates of vegetation to normalized difference vegetation index (NDVI) times series. In Pheno-VAE, the latent variables matched those phenological dates, which were the target of the inversion problem. Additional inductive biases were incorporated into the latent space of this model for enforcing order constraints between latent variables. Pheno-VAE was trained using NDVI time series computed from real S2 data. It was compared to the MPSR supervised deep learning regression, and to classical Markov Chain Monte Carlo (MCMC) and curve fitting regression methods. While Pheno-VAE showed interesting retrieval capabilities, it also showed limitations due to its reliance on interpolation of input time series. This highlighted that, since VAE methods are based on data reconstruction, it is necessary that the input data can be accurately reconstructed with little alteration. This calls for using more elaborate architectures for the encoder to be able to take irregular time series into account.

10.2 Perspectives

The proposed physics-integrated VAE methodology has shown interesting results and potential. There are several directions in which future research efforts could be undertaken. In particular, future work may focus on improving the performance of the approach for PROSAIL-VAE and Pheno-VAE, and on extending its application to more complex settings while taking more information into account for inference.

10.2.1 Improvements of the physics-integrated VAE methodology

Enhancing the encoder architecture Both PROSAIL-VAE in and Pheno-VAE have highlighted the interest in possible improvements over the encoder architecture. In the former case, although a convolutional neural network (CNN) spatial encoder has been proposed, it didn't show better inference capabilities (subsubsection 8.3.3.2). Using an encoder with increased spatial context, along with introducing a spatial penalization of reconstruction,

and a spatial structure to the latent variables could help improve performances. In the case of Pheno-VAE taking irregular time series as input rather than requiring an interpolation on a regular grid could significantly improve the model. For instance, recurrent neural networks [Metzger et al., 2021], or attention-based [Bellet et al., 2024; Shukla and Marlin, 2021] encoder architectures could enable taking irregular time series as input.

Using more training data Pheno-VAE and PROSAIL-VAE were respectively trained with 10^6 time series and 2.4×10^7 pixels, which is a relatively high number considering that both encoder neural networks are not very complex. However, even if this data was produced with a concern for class diversity and variety, it is a very low amount of data compared to the available S2 data. For Pheno-VAE, only data from the `T31TCJ` S2 tile were used, with a total covered surface of $1\,\mathrm{km}^2$ spread across a $100 \times 100\,\mathrm{km}^2$ area. For PROSAIL-VAE, even if data from various tiles were used, patches were extracted from only small regions of interest (ROIs) within those tiles. Therefore, there is a lot more data that can be used for training those models. The data-sets used in this Ph.D. were sufficient to design the presented applications, but improving performance will require using more training data, more spatial and temporal variability. Besides, recent experiments not presented in this manuscript have shown that PROSAIL-VAE can be trained without difficulty with larger data-sets produced by other L2A processors achieving equivalent performances.

Validating the approach using more in-situ data The PROSAIL-VAE application showed accurate LAI and CCC retrieval when compared to in-situ data. However, this data was rather limited in quantity, with less than 300 total data points for each variable, and in diversity, since only few areas with few vegetation types were involved. Besides, the retrieval of other predicted PROSAIL variables were not validated with in-situ data. For Pheno-VAE, only simulated data were used to assess performances.

It is therefore necessary to collect in situ-data in greater quantity, and of biophysical variables that weren't evaluated, to extend the accuracy assessment of the developed approach.

Finding proxy performance metrics Experiments with PROSAIL-VAE highlighted in subsection 8.2.1 that the validation loss cannot be used as a reliable metric for assessing variable estimation performance. As a consequence, for now, validation with in-situ data is unavoidable for estimating the performance of models and for comparing them with others. An important perspective is therefore the development of a metric that enables assessing the performance of the models without relying on evaluation with in-situ data. Ideally, a robust metric would enable selecting the best model while only using available S2 images, and no in-situ measurements.

Improving the correlation between predicted variables As observed in the inversion of PROSAIL with PROSAIL-VAE, the ability of the proposed method to perform full model inversions allows to estimate a joint distribution between all inferred model variables. However, the correlations between variables are simply estimated, and in the case of PROSAIL-VAE, pairs of predicted variables displayed an unrealistic correlation. This specific problem is ill-posed, and could probably be alleviated by reducing the number of variables to predict by the encoder. In the general case, another solution may be to promote known correlations (or absence of correlation) by using learning biases, e.g. an additional loss term that penalizes the correlations between predicted variables in the encoder. This could be enforced, for instance, by adapting disentanglement techniques for regularizing latent variables.

Improving the computational efficiency The Monte Carlo reconstruction loss (MCRL) developed for the physics-integrated VAE approach has shown good performance, but it relied on sampling the latent variables many times and generating many reconstructions. For

Pheno-VAE the phenological model is a simple mathematical formula, so this didn't lead to particular computational issues. However for PROSAIL-VAE, multiple simulations with the more complex PROSAIL generated a memory bottleneck. This was mitigated by down-sampling the resolution of the model. Yet, when using sampling 70 latent samples for the Monte Carlo reconstruction loss (MCRL) the model couldn't handle training with more than one patch of size 32×32 at a time. This limits the ability of the model to train on more data.

To enable training VAE using the MCRL with relatively complex physical models in the decoder, it is necessary to improve the computational efficiency of the approach. First, reliable heuristics for selecting an optimal number of latent samples must be found. Second, if possible, the physical model performing simulation should be computationally optimized. A potential approach would be the use of emulators of the physical models, such as neural networks. Emulators could be used to replicate the model behavior, for parts of the model (e.g. the exponential integral function), or in its entirety.

Perfecting the semi-supervised cyclical training technique The semi-supervised cyclical training strategy was proposed in subsection 8.3.4, and was about directly comparing PROSAIL variable estimates with reference data that is auto-generated by PROSAIL-VAE. For now, experiments with this strategy have not been conclusive, and further investigation is required to assess its potential. Nonetheless, it could enable better estimation of all PROSAIL variables. In particular, the variables whose influence in the reconstructions is lower than others (e.g. the carotenoid content) could be better inferred since the supervised loss term in the cyclical training doesn't involve reconstructions. Also, with a supervised loss term, the estimation of each variable doesn't compete with the others, contrary to the reconstruction loss for which the variables with a greater influence may conceal variables with a lower influence.

10.2.2 Extension of the range of application of the methodology

Using contextual data sources Besides optical measurements, there are additional data sources that could be taken into account by the encoder. Such new inputs could include for instance surface elevation, meteorological data, or other remote measurements such as synthetic aperture radar (SAR). Similarly to the spectral bands used as input to PROSAIL-VAE, these inputs would not be penalized by training (i.e. not taken into account by the loss), but their incorporation could help the model learn to infer better physical variables. Such additional incorporation could help the model inference in the case where multiple sets of variables are likely solutions, since the inversion problem may be ill-posed.

Applying the proposed method to different inversion problems The proposed physics integrated VAE could be applied to more inversion problems than just PROSAIL or the double logistic phenological model. More complex phenological models could be used for modeling vegetation that is not well described by the double logistic model. Different RTM models may be used instead of PROSAIL to model vegetation differently. For instance, the popular Invertible Forest Reflectance Model (INFORM) [Atzberger, 2000] could be used in a model similarly to PROSAIL-VAE. In fact, the PROSAIL-VAE approach has garnered some attention in the community, since the very application of this approach with the INFORM model has been proposed in She et al. [2024].

The interest of such a method may even be found beyond Earth observation in remote sensing, in applications with an abundance of data but with little reference, and some physical model.

Combining multiple physical models As discussed above, it is possible to incorporate different physical models with the approach developed in this thesis. A potential application

for this is to use several models jointly in the decoder of a VAE. Two settings can be proposed.

In a first approach, multiple physical simulators that model differently the same remote sensing observations could be used simultaneously. For instance, PROSAIL is well suited to certain types of vegetation (e.g. crops), whereas other models such as INFORM are specialized on other types of vegetation (i.e. forests). Using jointly both of those models could enable to estimate the variables of the vegetation by using the model that is best suited. One possible implementation could be to estimate the variables, and to predict reconstructions of both models at the same time, and to use the reconstruction likelihood to select which set of variables to save. Alternatively, categorical variables could be added to the latent space for predicting which model is better suited for a given input data.

A second approach could consider performing the fusion of models that involve different dimensions of the input data. For instance PROSAIL and the phenological model both relate S2 remote sensing measurements to vegetation bio-physical variables. However, those variables are about different aspects of the vegetation: the chemical content and structure of vegetation for PROSAIL and the temporal evolution for the phenological model. Both the phenological model and PROSAIL could be used simultaneously with a VAE to predict canopy biophysical variables while constraining them to have a certain temporal behavior imposed by the phenological model. Another possible evolution of the proposed approach could be to use it in data assimilation problems rather than inversion problems.

Multi-modal representations Since a variety of different models may be accommodated by the proposed approach, there could be an opportunity to use it to estimate physical variables from multiple sources. Specifically, physical models that simulate observations of different sensors could be used jointly within the decoder of a VAE. For instance, S2 optical measurements used within PROSAIL-VAE could be complemented by SAR measurements (e.g. Sentinel-1 data). This could be achieved by incorporating a model that relates ground properties to SAR measurements. In the case those different models share common variables, the estimation of those variable could be improved by benefitting from the joint input of their associated sensors.

Residual latent variables Finally, in this Ph.D. physical models have been used as the sole generative component in the decoders of VAE. However, models are never perfect, there are some effects that are not modeled accurately, and some that are not taken into account. A possible future improvement of the approach developed in this thesis could be the incorporation of a non-physical, non-interpretable component to the latent space and to the decoder, as discussed in subsection 7.2.1. Specifically, the physical models in the decoder could be supplemented by auxiliary neural networks, that would be responsible for modeling residuals, and mitigating effects not taken into account in the reconstructions. In such a VAE, the latent variables would be distinguished into an interpretable part that is semantically bound to physical parameters, and a non-interpretable part. Disentanglement approaches, introduced in section 6.5 could be applied to these variables to enforce specific properties and inductive biases.

Conclusion en français

Le but de cette thèse a été de de développer des méthodologies pour calculer des représentations interprétables et pertinentes de la végétation des surfaces continentales, à partir d'observations de télédétection. En particulier, les variables physiques ont été identifiées en tant que représentations idéales, puisqu'elles caractérisent la nature des surfaces observées. Ces variables sont en général suffisantes pour diverses applications en aval, et ont une valeur en-soi. De telles variables sont typiquement les entrées de modèles physiques, ce qui fait de l'estimation de ces variables un problème d'inversion. Il existe notamment des modèles bien connus qui établissent un lien entre l'état de la végétation, en utilisant des variables des variables bio-physiques, et des observations de télédétection.

Les contributions de cette thèse sont détaillées en Partie II, Partie III et Partie IV. En Partie II, une dépendence cruciale des approches d'*apprentissage profond*[1] supervisé à la distribution des jeux de données d'entraînement simulées a été identifiée. La Partie III a développé une méthodologie basée sur l'*apprentissage de représentations*[2] et les *autoencodeurs variationels*[3], qui réalise une inversion de modèle non supervisée. Enfin, la Partie IV a présenté les résultats de l'application de la méthode proposée dans cette thèse, à deux problèmes d'inversion de modèle classiques en télédétection : PROSAIL et un modèle phénologique.

Dans la Partie II, une implémentation différentiable et parallélisable du modèle de transfert radiatif PROSAIL a été détaillée. Afin de diminuer le coût en calculs de la simulation de réflectances Sentinel-2 (S2) avec ce modèle, un sous-échantillonnage a été appliqué, et ce, sans dégrader l'erreur de simulation au delà des erreurs dues aux corrections atmosphériques. Ensuite, après avoir introduit les méthodes classiques d'inversion de modèle, la méthode *de facto* standard Simplified Level 2 Product Prototype Processor (SL2P) basée sur des réseaux de neurones a été présentée. Cette méthode permet l'estimation à grande échelle de variables de canopée, en réalisant l'inversion supervisée de PROSAIL. Ainsi que démontré dans cette partie, la performance des méthodes de régression supervisée dépend fortement des simulations utilisées dans le jeu de données d'entraînement. Plus spécifiquement, il a été montré que la version de PROSAIL utilisée, ainsi que la distribution des variables d'entrée du modèle, et les corrélations entres elles avaient une influence sur la performance d'inversion. Cela a été identifié comme une limitation clé, puisque les distributions de ces variables et leurs corrélations ne sont pas bien connues.

Dans la Partie III, les méthodes d'inférence Bayésiennes ont été étudiées en tant que méthodes probabilistes pour calculer des représentations à partir de données de télédétections. Une attention particulière a été apportée aux VAE, qui sont à l'intersection entre l'apprentissage profond et l'inférence variationelle. Ces modèles apprennent à inférer des représentations générative de données d'une manière auto-supervisée. Différentes méthodes d'incorporation d'informations *a priori* dans les VAE ont été examinées. Il a notamment été proposé de contraindre les représentations latentes pour qu'elles correspondent à des variables physiques, en combinant les VAE avec des modèles physiques. En incorporant un modèle physique dans le décodeur d'un VAE, les variables latentes sont sémantiquement liées à l'entrée d'un

[1] Deep learning.
[2] Representation learning.
[3] Variational autoencoders (VAE).

modèle. Un tel VAE intégrant des contraintes physiques effectue alors une inversion de modèle dans un cadre d'apprentissage de représentations. Cette méthodologie peut être appliquée à une large variété de modèles physiques, pourvu qu'ils soient différentiables et que les simulations ne soient pas trop coûteuses en calcul. D'autre part, les variables estimées sont décrites par une distribution de probabilité, plûtot qu'une estimation ponctuelle. En outre, plusieurs techniques permettant d'incorporer un modèle physique déterministe dans le cadre d'une méthode Bayésienne ont été proposées. Crucialement, une fois entraîné, le décodeur du VAE est écarté, et l'encodeur peut être utilisé pour réaliser une inférence probabiliste à large échelle.

La Partie IV présente les résultats de l'application de la méthode basée sur les VAE à l'inversion de deux modèles physiques. Dans la première application, l'implémentation différentiable de PROSAIL est intégrée dans le décodeur d'un VAE, dans le modèle PROSAIL-VAE. Les variables latentes de PROSAIL-VAE sont les variables de feuille et de canopée du modèle PROSAIL. Contrairement aux approches supervisées qui doivent être entraînées sur des jeux de données pré-simulés, PROSAIL-VAE a été entraîné directement avec des données S2. En utilisant des données terrain d'*indice de surface foliaire*[4] et du *contenu en chlorophylle de la canopée*[5], PROSAIL-VAE a été comparé a SL2P et à une *régression supervisée probabiliste multiple*[6], qui sont des approches supervisées d'inversion de PROSAIL basées sur des réseaux de neurones. PROSAIL-VAE a montré une performance d'estimation supérieure de ces variables sur les données terrain disponibles, à la fois des estimations précises et une quantification d'incertitudes cohérente. De plus, la distribution postérieure agrégée des variables de PROSAIL a été estimée avec des images S2, avec PROSAIL-VAE. Cela a permis d'estimer les corrélations entre les variables de PROSAIL. Certaines variables sont vraisemblablement mieux estimées que d'autres, ce qui souligne le besoin de validations supplémentaires avec des mesures terrain de variables de végétation. Des configurations différentes de PROSAIL-VAE qui incorporent de la connaissance *a priori* on aussi été étudiées

Dans la seconde application de la méthodologie proposée, un modèle phénologique double-logistique a été intégré dans le décodeur d'un VAE, créant le modèle Pheno-VAE. Ce modèle physique met en lien des dates phénologiques de la végétation avec des séries temporelles d'*indice de végétation par différence normalisée*[7]. Les variables latentes de Pheno-VAE correspondent à ces dates phénologiques, qui sont les variables cible du problème d'inversion associé. Des biais inductifs additionnels ont été incorporés dans l'espace latent de Pheno-VAE, afin d'imposer des contraintes d'ordre entre les variables latentes. Pheno-VAE a été entraîné à partir de séries temporelles de NDVI produites à parties de données S2. Pheno-VAE a été comparé à l'approche MPSR, à une inversion bayésienne classique de Markov Chain Monte Carlo (MCMC), et à une régression par *ajustement de courbe*[8]. Alors que Pheno-VAE a montré des capacités d'estimation intéressantes, cette approche possède aussi des limitations dues à sa dépendance à l'interpolation de séries temporelles en entrée. Cela a souligné qu'il était nécessaire que les données d'entrée soient reconstruites le plus précisément possible, puisque VAE sont basées sur la reconstruction de données. Cela suggère l'utilisation d'architectures d'encodeur plus élaborées, afin de pouvoir utiliser directement des séries temporelles irrégulièrement échantillonnées.

[4] Leaf area index (LAI).
[5] Canopy chlorophyll content (CCC).
[6] Multiple probabilistic supervised regression (MPSR).
[7] Normalized difference vegetation index (NDVI).
[8] Curve fitting.

Part VI

Appendices

Appendix A

Data and implementations

This appendix gives pointers to data and code used in this work and briefly describes the computing environment used.

A.1 Repositories

A.1.1 S2 NDVI time series for Pheno-VAE

The real Sentinel-2 (S2) reflectance time series (bands `B4` and `B8`) used for Pheno-VAE experiments presented in chapter Chapter 9 are available at the following repository.
Y. Zérah, S. Valero, and J. Inglada. Sentinel-2 time series for Pheno-VAE, Nov. 2022. URL https://doi.org/10.5281/zenodo.7273500

A.1.2 Pheno-VAE

The Python implementation of the Pheno-VAE models, their training procedure and the generation of simulated normalized difference vegetation index (NDVI) time series, presented in chapter Chapter 9, are available at the following repository:
https://src.koda.cnrs.fr/smrxmlbw/pheno-vae.git.

A.1.3 PROSAIL

The differentiable and refactored Python implementation of PROSAIL developed in this Ph.D. (see Chapter 4), and used in the experiments of Chapter 5 and Chapter 8 is provided in the following repository (branch `downsampled_tensor`):
https://src.koda.cnrs.fr/mmdc/prosailpython.git.

A.1.4 PROSAIL-VAE

The Python implementation of the PROSAIL-VAE models and their training procedure, presented in Chapter 8, is provided in the following repository:
https://src.koda.cnrs.fr/smrxmlbw/prosailvae.git.

A.2 Computing environment

The computational resources of the HPC platform of CNES's Data Processing Centre were used for the experiments in Chapter 5, Chapter 8 and Chapter 9. The description of the graphical processing unit (GPU) nodes of this cluster that were used is given in the following:

- CPU: Intel® Xeon® CPU E5-2698 v4,
- GPU model: NVIDIA® Tesla® V100-SXM2-32GB,

- Allocated RAM: 64 GB.

Appendix B

Linear algebra

Contents

This appendix presents a details of linear algebra for the different methodologies presented in this manuscript. In particular, these different mathematical aspects aim at providing context around matrix inversion methods. section B.1, section B.2 and section B.3 present techniques involved in matrix inversion. section B.4 investigates the case of the inversion of covariance matrices with a focus on covariance matrices estimated using with Monte Carlo (MC) sampling, as a complement to the Monte Carlo reconstruction loss (MCRL) method discussed in subsection 7.2.3.

B.1 Cholesky decomposition

Let $A \in \mathcal{M}_{n,n}(\mathbb{C})$ be a Hermitian matrix[1]. If A is positive-definite, then there is a unique factorization with lower triangular matrices L, called *Cholesky* factorization (or decomposition):

$$A = LL^*. \tag{B.1}$$

L is invertible, and has positive diagonal entries (its eigenvalues).

If A is positive semi-definite with rank $r < n$, then there are non-unique decompositions into lower triangular matrices L. There is a unique such decomposition with r diagonal positive entries, and $n - r$ zeros columns. Neither A nor L are invertible.

There are several algorithms that compute the Cholesky factorization of a matrix, with a computational complexity proportional to that of matrix multiplication (generally $\mathcal{O}(n^3)$, sometimes a little lower for more efficient algorithms).

B.2 LU decomposition

Let $A \in \mathcal{M}_{n,n}(\mathbb{C})$. A lower-upper factorization (or LU factorization) of A is the factorization with a lower triangular matrix L and upper triangular matrix U:

$$A = LU \tag{B.2}$$

The LU factorization is usually not unique. A variant is a LDU (lower-diagonal-upper) decomposition, which introduces a diagonal matrix D, and forces the diagonal entries of L and U to 1:

$$A = LDU. \tag{B.3}$$

The LDU factorization of an invertible matrix is unique. Not all square matrices admit a LU decomposition, however a reordering of the rows of a matrix with a permutation matrix P can enable the decomposition.

$$\forall A \in \mathcal{M}_{n,n}(\mathbb{C}), \quad \exists P \in \mathcal{S}_n, L \in \mathcal{M}_{n,n}(\mathbb{C}), U \in \mathcal{M}_{n,n}(\mathbb{C}) \quad \text{s.t.} \quad PA = LU \tag{B.4}$$

The LU decomposition is the matrix form of the Gaussian elimination algorithm, i.e., they represent the sequence of elementary operations used to solve a linear system of equations.

There are several algorithms that compute the LU factorization. It should be noted that LU factorization is less efficient and stable than Cholesky factorization (see section B.1).

B.3 Singular value decomposition

Let $A \in \mathcal{M}_{m,n}(\mathbb{C})$. The singular value decomposition (SVD) of A is a factorization in the form:

$$A = U\Sigma V^*, \tag{B.5}$$

[1] A Hemitian matrix A is a square matrix that is equal to its own transpose conjugate $A^* = \left(\overline{A}\right)^\top$.

s.t. $U \in \mathcal{M}_{m,m}(\mathbb{C})$ and $V \in \mathcal{M}_{n,n}(\mathbb{C})$ are unitary matrices[2], and $\Sigma \in \mathcal{M}_{m,n}(\mathbb{C})$ is a rectangular matrix with non-negative diagonal entries. The diagonal elements σ_i of Σ are the *singular values* of A, and are a unique set. The number of non-zero singular values is equal to the rank of A. If $A \in \mathcal{M}_{m,n}(\mathbb{R})$, then U and V are real orthogonal matrices. The singular value decomposition (SVD) is not unique. SVD is a generalization of *eigenvalue decomposition*[3] to all matrices — in particular, non-invertible, defective[4] and non-square matrices.

B.3.1 Moore-Penrose inverse

The SVD also extends the notion of inverse to any matrix. For a matrix $A \in \mathcal{M}_{m,n}(\mathbb{C})$ with SVD $A = U\Sigma V^*$, the matrix

$$A^\dagger = U\Sigma^\dagger V^* \tag{B.6}$$

is its unique *Moore-Penrose inverse*, or *pseudoinverse*. The matrix $\Sigma^\dagger$ is the pseudoinverse of Σ, formed by replacing the non-zero singular values in the diagonal by their reciprocal, and transposing. If A is invertible, its pseudoinverse and inverse are the same.

The pseudoinverse is a *weak inverse*:

$$AA^\dagger A = A \tag{B.7}$$

$$A^\dagger AA^\dagger = A^\dagger \tag{B.8}$$

and is hermitian

$$\left(AA^\dagger\right)^* = AA^\dagger \tag{B.9}$$

$$\left(A^\dagger A\right)^* = A^\dagger A. \tag{B.10}$$

B.3.2 Covariance matrix

Performing SVD on a covariance matrix $\mathbf{\Sigma}$ is the basis of principal components analysis (PCA). Indeed, the covariance matrix captures the total variation of some data, and the SVD derives the singular values, which are the magnitude of the variances along orthogonal directions.

B.4 Inversion of covariance matrix Monte-Carlo estimate

B.4.1 Ill-conditionned estimate covariance matrices

A covariance matrix estimated from samples of a multi-variate random variable (such as the reconstruction vectors $\hat{\mathbf{x}} = \mathcal{F}(\mathbf{z})$, see subsection 7.2.3) can be singular[5] or ill-conditionned[6], in particular because:

1. the matrix is computed from a limited number of samples, and the matrix estimate is not accurate,

2. realizations of the random vector has co-linear or nearly co-linear components.

[2]Unitary matrices are invertible and their inverse is their conjugate transpose: $U^{-1} = \left(\overline{U}\right)^\top$.

[3]The eigenvalue decomposition of a square, diagonalizable matrix $A \in \mathcal{M}_{n,n}(\mathbb{C})$ is a factorization $A = Q\Lambda Q^{-1}$, with Q the matrix whose columns are the eigenvectors of A, and Λ is a diagonal matrix of the eigenvectors of A

[4]non-diagonalizable.

[5]A singular matrix, involved in an *ill-posed problem*, has no inverse and its determinant is 0.

[6]An ill-conditionned or *badly-conditionned* matrix, is a non-singular matrix for which the inversion is subject to numerical instability. The conditioning of matrices can be described with a *condition number*, that quantifies the change in the matrix inversion for a small change in coefficients. The ratio of the largest over the smallest *singular values* is a condition number. The determinant is not affected by ill-conditioning

The rank of a covariance matrix estimate can be limited by the number of available samples. In particular, for K number of samples, $K-1$ is an upper bound on the rank of the covariance matrix estimate:

$$\operatorname{rank}\left(\widehat{\boldsymbol{\Sigma}}\right) \leq K-1 \tag{B.11}$$

This means that for an m-dimensional reconstruction space, there is a minimal number of MC samples to draw in the latent space and propagate through the decoder, so that the covariance matrix estimate of the reconstructed vector is full-rank.

$$\operatorname{rank}\left(\widehat{\boldsymbol{\Sigma}}\right) = \dim\left(\boldsymbol{x}\right) = m \Rightarrow K \geq m+1 \tag{B.12}$$

It must be noted that the case of a full-rank $\widehat{\boldsymbol{\Sigma}}$ with $K = \dim\left(\boldsymbol{x}\right)+1$ only occurs if the components of $\boldsymbol{x}$ are linearly independent. Therefore in the general case, a larger MC sample number must be used.

The reconstructed vector components can be co-linear and make the covariance matrix singular. Let's consider a simple situation with a linear model as decoder, i.e. $\mathbf{z} = A\mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^m, A \in \mathcal{M}_{m,n}\left(\mathbb{R}\right), \mathbf{z} \in \mathbb{R}^n$. In the common case $m > n$, the reconstructed vector has linearly dependent components, and

$$\widehat{\boldsymbol{\Sigma}} = \operatorname{Cov}\left(\mathbf{x}\right) = A \operatorname{Cov}\left(\mathbf{z}\right) A^\top \tag{B.13}$$

and it follows that

$$\operatorname{rank}\left(\operatorname{Cov}\left(\mathbf{x}\right)\right) \leq \min\left(\operatorname{rank}\left(A\right), \operatorname{rank}\left(\operatorname{Cov}\left(\mathbf{z}\right)\right)\right) \leq n < m. \tag{B.14}$$

Therefore, in the linear case, when the dimension of the reconstruction is larger than the dimension of the latent space, the covariance matrix of the reconstruction is singular. In the broader case of a non-linear deterministic model in the decoder, $\operatorname{Cov}\left(\mathbf{x}\right)$ cannot simply be expressed as a function of $\operatorname{Cov}\left(\mathbf{z}\right)$, and a relationship between their ranks is usually out of reach. Nonetheless, $\operatorname{Cov}\left(\mathbf{x}\right)$ may, again, not be full rank. For instance, the double-logistic phenological model Ω (see section 9.1) as decoder in Chapter 9 produces reconstructions $\boldsymbol{x}_i$ that are time series, and that can have nearly co-linear components $x_{i,j}$. The reconstruction components are

$$x_{i,j} = \boldsymbol{x}_i\left(t_j\right) = \Omega\left(\boldsymbol{z}_i, t_j\right), \tag{B.15}$$

with $\boldsymbol{z}_i = (m_i, M_i, SoS_i, Mat_i, Sen_i, EoS_i)$. For $t_j < t_k \ll SoS_i$, $x_{i,j} \approx x_{i,k} \approx m_i$, i.e. $\boldsymbol{x}_i$ components corresponding to instants far enough from the SoS are co-linear.

When a covariance matrix or its estimate is singular, no amount of matrix pre-constraining or regularization can enable to compute an accurate inverse, as there is none. In those cases, a pseudo-inverse (see subsection B.3.1), can be computed instead. Also, as a singular covariance matrix will yield a zero determinant, the loss term $\ln\left(\left|\widehat{\boldsymbol{\Sigma}}\right|\right)$ is undetermined. A possible mitigation is the computation of a "pseudo-determinant" as the product of non zero singular values, obtained with SVD. Finally, the influence of approximating a (non-existent) $\widehat{\boldsymbol{\Sigma}}^{-1}$ with $\widehat{\boldsymbol{\Sigma}}^{\dagger}$ on the loss during training remains to be assessed. Regardless of singularity, using SVD to compute an $\widehat{\boldsymbol{\Sigma}}^{-1}$ can also help alleviate the ill-conditioning.

Improving the conditioning of $\widehat{\boldsymbol{\Sigma}}$ can also be performed by using an altered surrogate, such as $\widehat{\boldsymbol{\Sigma}} + \alpha \mathrm{I}_\mathrm{n}$, with $\alpha > 0$ a hyper-parameter. This alteration is called *ridge regularization*[7], and it changes the eigenvalues of $\widehat{\boldsymbol{\Sigma}}$, and improves the matrix conditioning [Hoerl and Kennard, 1970]. The larger α is, the closer to a diagonal matrix $\widehat{\boldsymbol{\Sigma}}$ becomes and the more stability increases. Conversely, the parameter α is an offset to the variance of reconstruction components, and must not set too large so that it is still meaningful.

[7]This matrix regularization is also employed in the so-called *ridge regression*, in particular for solving linear least squares problems (see subsubsection 3.2.1.1).

B.4.2 Computational cost

Matrix multiplication for matrices with size N have an asymptotic computational cost of $\mathcal{O}(n^{\omega})$, with ω ranging between 2.237 for the newest methods [Duan et al., 2023; Williams, 2012][8] and 3 for the traditional method. The computational complexity of matrix inversion, determinant computing and SVD is proportional to that of matrix multiplication [Strassen et al., 1969]. This becomes expensive for any large size matrix, which occurs when the decoder output vector is high dimensional, such as with images. Furthermore, a large number of matrices, proportional to the batch size and number of MC samples, have to be inverted at training time.

Finally, the matrix inversion must be performed as a differentiable operation so that the gradient of the loss can be back-propagated. Gauss pivot inversion associated with matrix decomposition with LU decomposition (see section B.2) or Cholesky decomposition (see section B.1), is one such method, although matrix decomposition still requires non-singularity and a good conditioning. Besides, these decompositions provide a straight-forward and efficient computation of the determinant, as the determinant of a triangular matrix is simply the product of diagonal elements.

Using a covariance matrix in the reconstruction loss could improve reconstruction quality, and add structure to residuals Dorta et al. [2018]. However, covariance matrix estimation, inversion and determinant computation can be impractical, and can become prohibitively expensive for any large dimensional data.

B.4.3 Practical use

In the applications considered here, Pheno-VAE (Chapter 9) and PROSAIL-VAE (Chapter 8), using a full covariance matrix of the reconstruction components instead of their individual variance hasn't been found to improve the results. The negative log-likelihood (NLL) reconstruction loss was less stable, and the optimization had trouble converging. Furthermore, the training had to be stopped and re-started often, because the covariance estimate was often singular and inverse and determinant computing through Cholesky decomposition (see subsection B.4.2 and section B.1) failed. As discussed above, in the case of Pheno-VAE the reconstruction dimension with respect to (w.r.t.) the latent space dimension almost always guaranteed the singularity. For PROSAIL-VAE, covariance matrix singularity was much less frequent, however ill-conditionning still hampered training. When the loss did converge, performances were subpar compared to using individual variances.

Using a full covariance matrix in the loss isn't a bad approach altogether. However, the experiments conducted here seem to point that it cannot be paired with the MCRL (see subsection 7.2.2). Instead of computing a covariance estimate from reconstructed samples like we do, using more classical approaches, of predicting the covariance matrix (with a neural network for instance) may be a better approach. Furthermore, instead of considering the covariance matrix $\widehat{\mathbf{\Sigma}}$, such approaches could predict the lower triangular Cholesky decomposition L of this matrix. Advantages would be twofold:

- the relatively costly Cholesky decomposition can be avoided, and the inverse $\widehat{\mathbf{\Sigma}}^{-1}$ is straightforwardly obtained,
- the matrix L is easier to constrain than $\widehat{\mathbf{\Sigma}}$ to ensure non-singularity, and training stability.

[8] The algorithms achieving the lower bound on asymptotic complexity are in practice unused, as their gain in performance is only perceptible for matrices so large they are never encountered in practical applications. Such algorithms are called *galactic algorithms*.

Appendix C

Distributions

Contents

C.1 Density of maximum of continuous distributions

Let Y be the maximum of n independent continuous random variables X_i. The cumulative distribution function (CDF) of Y is:

$$\begin{aligned} F_Y(y) &= P(Y < y) \\ &= P\left(\max_{i\in[[1,n]]} X_i < y\right) \\ &= P\left(\bigcap_{i=1}^{n} (X_i < y)\right) \\ &= \prod_{i=1}^{n} P\left(X_i < y\right) \\ &= \prod_{i=1}^{n} F_{X_i}(y) \end{aligned} \tag{C.1}$$

The log-derivative of the CDF of Y yields:

$$\begin{aligned} \frac{d\ln F_Y}{dy}(y) &= \frac{d}{dy}\ln\left(\prod_{i=1}^{n} F_{X_i}(y)\right) \\ &= \frac{d}{dy}\sum_{i=1}^{n}\ln\left(F_{X_i}(y)\right) \\ &= \sum_{i=1}^{n}\frac{d}{dy}\ln\left(F_{X_i}(y)\right) \\ &= \sum_{i=1}^{n}\frac{dF_{X_i}(y)}{dy}\frac{1}{F_{X_i}(y)} \\ &= \sum_{i=1}^{n} f_{X_i}(y)\frac{1}{F_{X_i}(y)} \end{aligned} \tag{C.2}$$

Finally, using the log-derivative of the CDF of Y enables deriving its probability distribution function (PDF) as a function of the PDFs and CDFs of X_i:

$$\begin{aligned} f_Y(y) &= \frac{dF_Y}{dy}(y) \\ &= F_Y(y)\frac{d\ln F_Y}{dy}(y) \\ &= \prod_{i=1}^{n} F_{X_i}(y)\sum_{i=1}^{n} f_{X_i}(y)\frac{1}{F_{X_i}(y)} \end{aligned} \tag{C.3}$$

C.2 Kumaraswamy distribution

C.2.1 Probability distribution function

C.2.2 Definition

Definition 1 (Kumaraswamy distribution)**.** *The Kumaraswamy distribution is bounded distribution over the interval* $[0,1]$*. A random variable* X *that follows a Kumaraswamy distribution with parameters* $a, b \in \mathbb{R}_+^{*\,2}$*, denoted* $X \sim \mathcal{K}(a,b)$ *has the following PDF:*

$$f_X(x) = abx^{a-1}\left(1-x^a\right)^{b-1}$$

over the interval $[0,1]$*.*

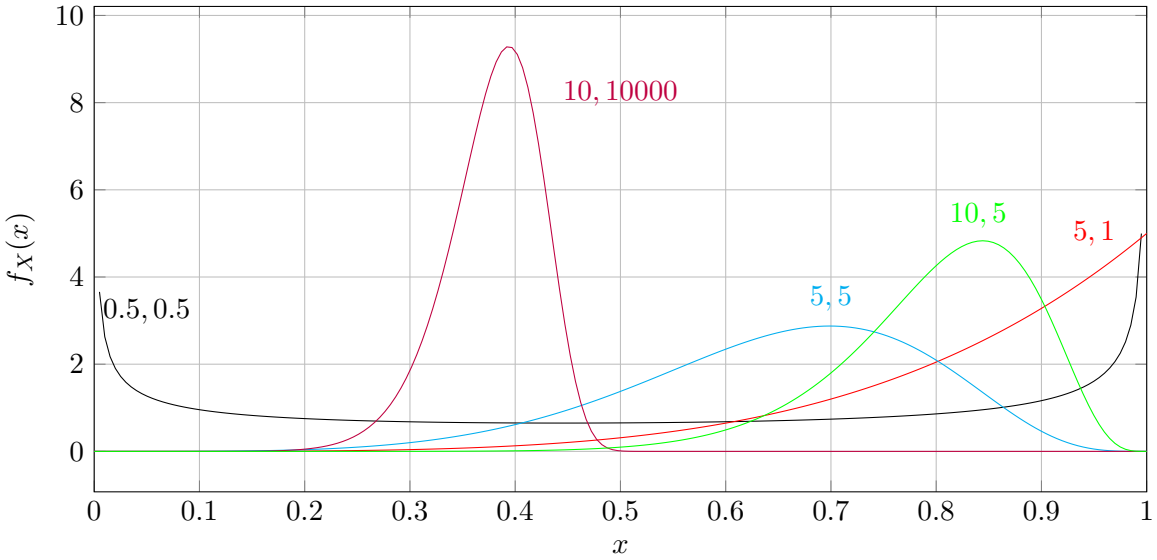


Figure C.1: PDF of Kumaraswamy distributions for 5 different sets of parameters a, b (displayed next to each corresponding curve).

Remark.

- A Kumaraswamy distribution with parameters $a = b = 1$ is a uniform distribution over the interval $[0,1]$.
- A Kumaraswamy distribution with both $a < 1$ and $b < 1$ is bimodal with modes at $x = 0$ and $x = 1$.
- A Kumaraswamy distribution with either $a = 1$ or $b = 1$ has a single mode respectively in $x = 0$ and $x = 1$.

C.2.3 Cumulative distribution function

Definition 2 (CDF of the Kumaraswamy distribution)**.** *If* $X \sim \mathcal{K}(a,b)$*, then :*

$$F_X(x) = 1 - \left(1-x^a\right)^b$$

over the interval $[0,1]$*.*

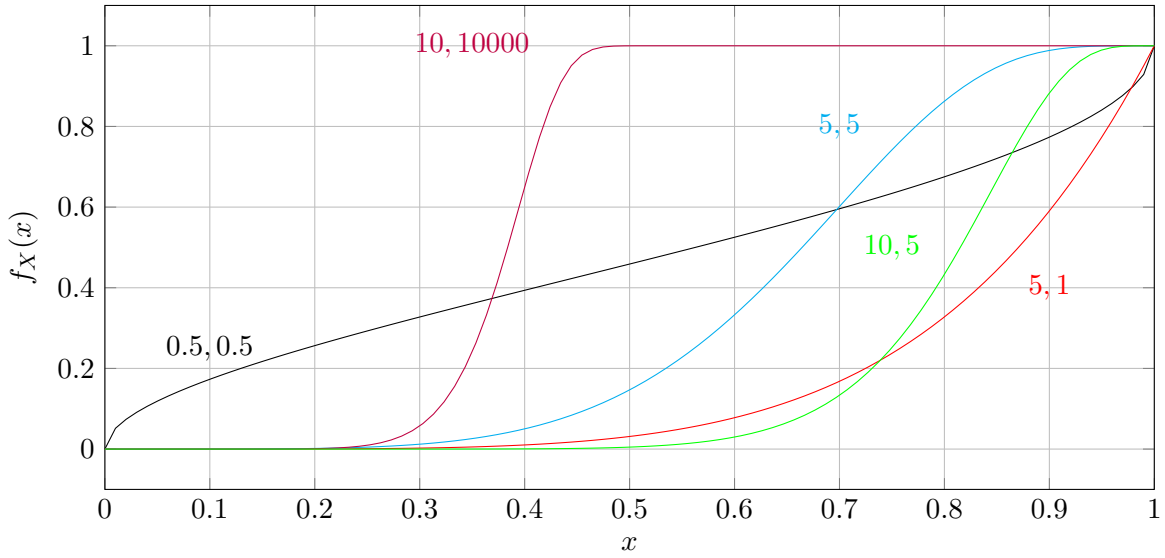


Figure C.2: CDF of Kumaraswamy distributions for 5 different sets of parameters a, b (displayed next to each corresponding curve).

C.2.4 Inverse cumulative distribution function

Definition 3 (ICDF of the Kumaraswamy distribution)**.** *If* $X \sim \mathcal{K}(a, b)$*, then :*

$$F_X^{-1}(x) = \left(1 - (1-x)^{\frac{1}{b}}\right)^{\frac{1}{a}}.$$

over the interval $[0, 1]$.

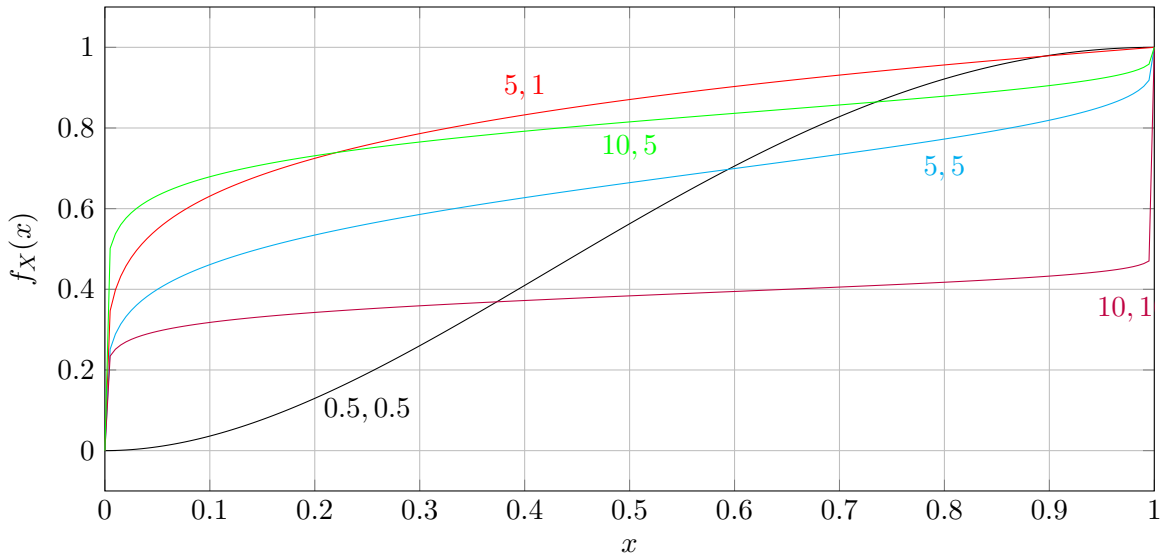


Figure C.3: inverse cumulative distribution function (ICDF) of Kumaraswamy distributions for 5 different sets of parameters a, b (displayed next to each corresponding curve).

C.2.4.1 Derivatives of the ICDF of Kumaraswamy distributions

Definition 4 (Partial derivatives of the ICDF of Kumaraswamy distributions)**.** *Let* $X \sim$

$\mathcal{K}(a,b)$. *Then:*

$$\frac{\partial F_X^{-1}}{\partial x}(x) = \frac{1}{ab}(1-x)^{\frac{1}{b}-1}\left(1-(1-x)^{\frac{1}{b}}\right)^{\frac{1}{a}-1}$$
$$\frac{\partial F_X^{-1}}{\partial a}(x) = \frac{1}{a^2}\log\left[1-(1-x)^{\frac{1}{b}}\right]\left(1-(1-x)^{\frac{1}{b}}\right)^{\frac{1}{a}}$$
$$\frac{\partial F_X^{-1}}{\partial b}(x) = \frac{1}{ab^2}\log\left[1-x\right](1-x)^{\frac{1}{a}}\left(1-(1-x)^{\frac{1}{b}}\right)^{\frac{1}{a}-1}$$

C.2.4.2 Diverging limits of Kumaraswamy ICDF derivatives

$$(1-x)^{\frac{1}{b}-1} \xrightarrow[x\to 1]{} +\infty \Rightarrow \frac{\partial F_{a,b}^{-1}}{\partial x}(x) + \xrightarrow[x\to 1]{} +\infty \tag{C.4}$$

$$\left(1-(1-x)^{\frac{1}{b}}\right)^{\frac{1}{a}-1} \xrightarrow[x\to 0]{} +\infty \Rightarrow \begin{cases} \frac{\partial F_{a,b}^{-1}}{\partial x}(x) + \xrightarrow[x\to 0]{} +\infty \\ \frac{\partial F_{a,b}^{-1}}{\partial x}(b) + \xrightarrow[x\to 0]{} +\infty \end{cases} \tag{C.5}$$

$$\log\left[1-(1-x)^{\frac{1}{b}}\right] \xrightarrow[x\to 0]{} -\infty \Rightarrow \frac{\partial F_{a,b}^{-1}}{\partial a}(x) + \xrightarrow[x\to 0]{} -\infty \tag{C.6}$$

$$\log\left[1-x\right] \xrightarrow[x\to 1]{} -\infty \Rightarrow \frac{\partial F_{a,b}^{-1}}{\partial b}(x) + \xrightarrow[x\to 1]{} -\infty \tag{C.7}$$

C.3 Normal distribution

C.3.1 Probability distribution function

Definition 5 (Normal distribution)**.** *A random variable X that follows a normal (or Gaussian) distribution with parameters $\mu, \sigma \in \mathbb{R} \times \mathbb{R}_+^*$, denoted $X \sim \mathcal{N}(\mu, \sigma)$, has the PDF:*

$$f_X(x) = \frac{e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}}{\sigma\sqrt{2\pi}}$$

Definition 6 (Standard normal distribution)**.** *The standard normal distribution is the normal distribution with parameters $\mu = 0$ and $\sigma = 1$. Its PDF is denoted $\varphi(x) = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}$. A normal distribution with parameters μ, σ can be obtained via an affine transformation from a standard normal distribution:*

$$X \sim \mathcal{N}(0,1) \Rightarrow \mu + \sigma X \sim \mathcal{N}(\mu, \sigma)$$

The PDF of a normal distribution X with parameters μ, σ can be derived from the PDF of a standard normal distribution:

$$f_X(x) = \frac{1}{\sigma}\varphi\left(\frac{x-\mu}{\sigma}\right)$$

C.3.2 Negative Log-Likelihood

Definition 7 (NLL of a normal distribution)**.** *Let $X \sim \mathcal{N}(\mu, \sigma)$. Then, its NLL is*

$$\mathrm{NLL}(x, \mu, \sigma) = \frac{1}{2}\left[\left(\frac{x-\mu}{\sigma}\right)^2 + \log\sigma^2 + \log 2\pi\right].$$

C.3.3 Cumulative distribution function

Definition 8 (Error function)**.** *The error function (erf) is defined as the probability of $X \sim \mathcal{N}\left(0, \frac{1}{2}\right)$ falling in the interval $[-x, x]$:*

$$\mathrm{erf}(x) = P(-x < X < x) = \int_{-x}^{x} f_X(t)\,\mathrm{d}t = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,\mathrm{d}t$$

Remark. There is no closed-form of the error function, it is commonly interpolated from pre-tabulated values, or approximated with logistic functions.

Definition 9 (CDF of the standard normal distribution)**.** *The CDF of the standard normal distribution $X \sim \mathcal{N}(0, 1)$, denoted Φ, is related to the erf function:*

$$\Phi(x) = F_X(x) = \int_{-\infty}^{x} f_X(t)\,\mathrm{d}t = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$$

Definition 10 (CDF of a normal distribution)**.** *The CDF of a normal distribution $X \sim \mathcal{N}(\mu, \sigma)$, is derived from the CDF of the standard normal distribution, and related to the error function:*

$$F_X(x) = \Phi\left(\frac{x-\mu}{\sigma}\right) = \int_{-\infty}^{x} f_X(t)\,\mathrm{d}t = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right]$$

Definition 11 (Gaussian CDF, confidence intervals and σ rule)**.** *Let $X \sim \mathcal{N}(\mu, \sigma)$. Then*

$$P(\mu - n\sigma < X < \mu + n\sigma) = F_X(\mu + n\sigma) - F_X(\mu + n\sigma) = \mathrm{erf}\left(\frac{n}{\sqrt{2}}\right).$$

The interval $[\mu - n\sigma, \mu + n\sigma]$ is a n-σ confidence interval, for which the probability of X samples falling into it (or confidence level) is $\frac{n}{\sqrt{2}}$. The confidence level at n-σ is the probability of samples deviating less than $n \times \sigma$ from the mean μ. For $n = 1, 2, 3$, the confidence level is respectively approximately 0.68, 0.95 and 0.997, and are commonly used values for confidence levels.

C.3.4 Inverse cumulative distribution function

Definition 12 (ICDF of the standard normal distribution)**.** *The CDF of the standard normal distribution $X \sim \mathcal{N}(0, 1)$, denoted Φ, is related to the inverse error function*

(inverf) function:

$$\forall x \in]0,1[, \quad \Phi^{-1}(x) = F_X^{-1}(x) = \sqrt{2}\text{erf}^{-1}(2x-1)$$

Remark. The inverf is tabulated by using its definition: $\text{erf}^{-1}(\text{erf}(x)) = x$.

Definition 13 (ICDF of a normal distribution)**.** *The CDF of a normal distribution $X \sim \mathcal{N}(\mu, \sigma)$, is derived from the ICDF of the standard normal distribution, and related to the inverf:*

$$\forall x \in]0,1[, \quad F_X^{-1}(x) = \mu + \sigma\Phi(x) = \mu + \sigma\sqrt{2}\text{erf}^{-1}(2x-1)$$

C.3.5 Kullback-Leibler divergence

Definition 14 (KLD between two normal distributions)**.** *Let f_1 and f_2 denote the PDF of X_1 and X_2, two normal distributions, such that:*
$X_1 \sim \mathcal{N}(\mu_1, \sigma_1)$ and $X_2 \sim \mathcal{N}(\mu_2, \sigma_2)$. Then

$$\text{D}_{\text{KL}}(X_1 \| X_2) = -\frac{1}{2}\left[\left(\frac{\sigma_1}{\sigma_2}\right)^2 + \frac{(\mu_2-\mu_1)^2}{\sigma_1^2} - 1 + \ln\left(\frac{\sigma_2^2}{\sigma_1^2}\right)\right]$$

C.4 Two-sided truncated normal distribution

C.4.1 Definition

Definition 15 (Two-sided truncated normal distribution)**.** *Let $X \sim \mathcal{N}(\mu, \sigma)$, with $(\mu, \sigma) \in \mathbb{R} \times \mathbb{R}^*$. Then $X|x \in [a,b]$, with $(a,b) \in \mathbb{R}_+^2$ and $a < b$, follows a truncated normal truncated normal (TN) distribution, denoted $\mathcal{TN}(\mu, \sigma, a, b)$.*

Remark. If either $a = -\infty$ or $b = +\infty$, then the distribution is one-sided.

C.4.2 Probability distribution function

Definition 16 (PDF of a two-sided truncated normal distribution)**.** *The PDF of $X \sim \mathcal{TN}(\mu, \sigma, a, b)$ is:*

$$f_X(x) = \frac{\varphi\left(\frac{x-\mu}{\sigma}\right)}{\sigma\eta},$$

with $\eta = \Phi(\beta) - \Phi(\alpha)$, $\alpha = \frac{a-\mu}{\sigma}$, $\beta = \frac{b-\mu}{\sigma}$.

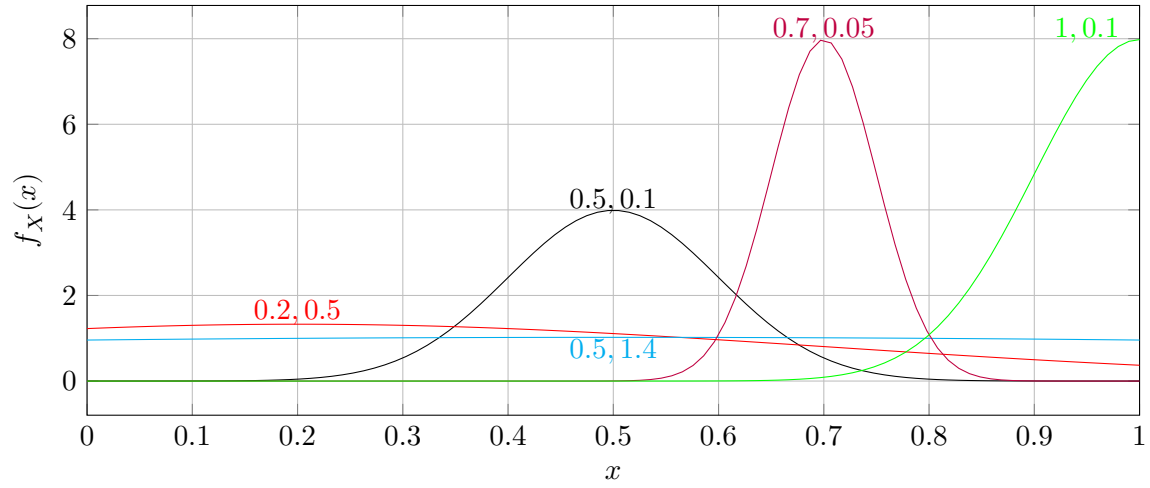


Figure C.4: PDF of TN distributions for 5 different sets of parameters μ, σ (displayed next to each corresponding curve).

C.4.3 Negative Log-Likelihood

Definition 17 (NLL of a two-sided truncated normal distribution)**.** *The NLL of* $X \sim \mathcal{TN}(\mu, \sigma, a, b)$ *is:*

$$\mathrm{NLL}(x, \mu, \sigma) = \frac{1}{2}\left[\left(\frac{x-\mu}{\sigma}\right)^2 + \log\sigma^2 + \log\eta^2 + \log 2\pi\right]$$

with $\eta = \Phi(\beta) - \Phi(\alpha)$, $\alpha = \frac{a-\mu}{\sigma}$, $\beta = \frac{b-\mu}{\sigma}$.

C.4.4 Cumulative distribution function

Definition 18 (CDF of a two-sided truncated normal distribution)**.** *The CDF of* $X \sim \mathcal{TN}(\mu, \sigma, a, b)$ *is:*

$$F_X(x) = \frac{\Phi\left(\frac{x-\mu}{\sigma}\right) - \Phi(\alpha)}{\eta}.$$

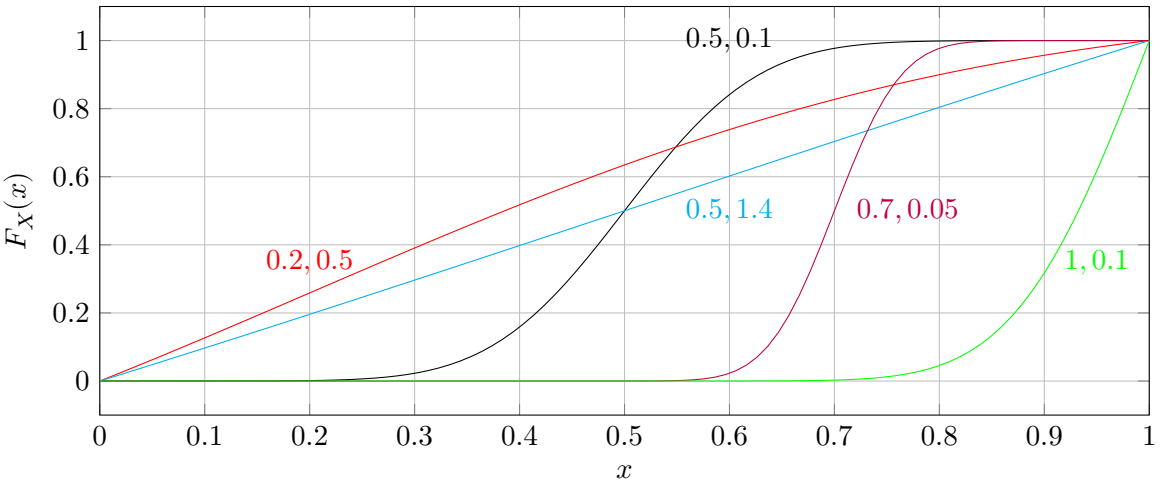


Figure C.5: CDF of TN distributions for 5 different sets of parameters μ, σ (displayed next to each corresponding curve).

C.4.5 Inverse cumulative distribution function

Definition 19 (ICDF of a two-sided truncated normal distribution)**.** *The ICDF of* $X \sim \mathcal{TN}(\mu, \sigma, a, b)$ *is:*

$$\forall x \in]0,1[, \quad F_X^{-1}(x) = \mu + \sigma\Phi^{-1}(\Phi(\alpha) + \eta x)$$

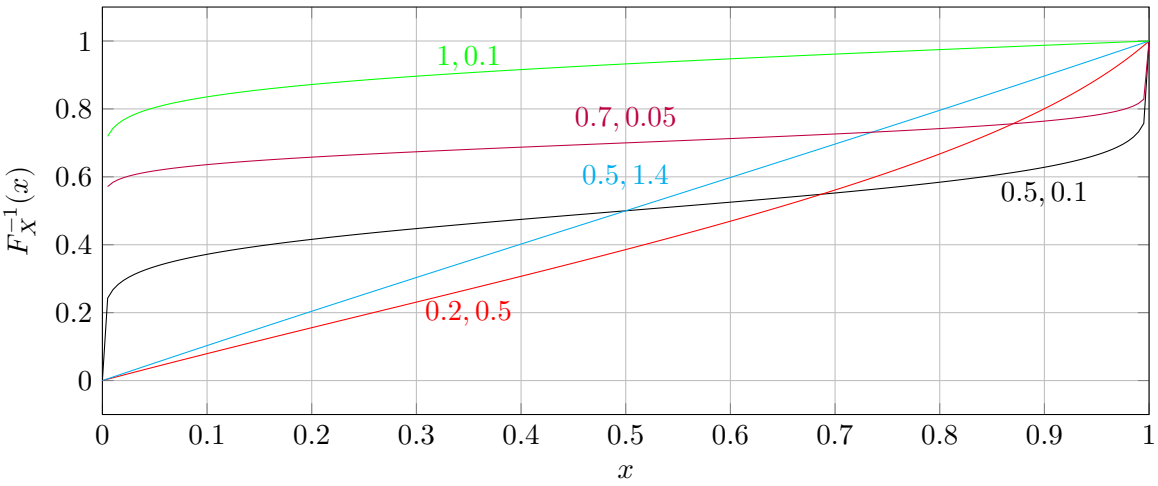


Figure C.6: ICDF of TN distributions for 5 different sets of parameters μ, σ (displayed next to each corresponding curve).

C.4.6 Kullback-Leibler divergence

C.4.6.1 Between two truncated normal distributions

Formula 1 (KLD between two TN distributions)**.** *Let* f_1 *and* f_2 *denote the PDF of* X_1 *and* X_2*, two TN distributions, such that:*
$X_1 \sim \mathcal{TN}(\mu_1, \sigma_1, a, b)$ *and* $X_2 \sim \mathcal{TN}(\mu_2, \sigma_2, a, b)$*. Then:*

$$\begin{aligned} \mathrm{D}_{\mathrm{KL}}(X_1 \| X_2) = -\frac{1}{2} - \ln\left(\frac{\sigma_1\eta_1}{\sigma_2\eta_2}\right) - \frac{K_1}{2\eta_1}\left(1 - \frac{\sigma_1^2}{\sigma_2^2}\right) + \frac{(\mu_1-\mu_2)^2}{2\sigma_2^2} + \frac{\sigma_1^2}{2\sigma_2^2} \\ + (\mu_1 - \mu_2)\frac{\sigma_1}{\sigma_2^2\eta_1}N_1 \end{aligned} \tag{C.8}$$

with: $\eta_i = \Phi(\beta_i) - \Phi(\alpha_i)$, $\alpha_i = \frac{a-\mu_i}{\sigma_i}$, $\beta_i = \frac{b-\mu_i}{\sigma_i}$, $K_1 = \alpha_1\varphi(\alpha_1) - \beta_1\varphi(\beta_1)$ *and* $N_1 = \varphi(\alpha_1) - \varphi(\beta_1)$.

Remark. The derivation of this formula is provided in subsection D.1.1.

C.4.6.2 Between a truncated normal distribution and uniform distribution

Formula 2 (KLD of a TN distribution from a uniform distribution)**.** *Let* f_1 *and* f_2 *denote the PDF of* X_1 *and* X_2*, a TN and a uniform distributions, such that:*
$X_1 \sim \mathcal{TN}(\mu, \sigma, a, b)$ *and* $X_2 \sim \mathcal{U}(a, b)$*. Then*

$$\mathrm{D}_{\mathrm{KL}}(X_1 \| X_2) = -\frac{1}{2} - \frac{1}{2}\ln(2\pi) - \ln(\sigma\eta) - \frac{K}{2\eta} + \ln(b-a) \tag{C.9}$$

Remark. The derivation of this formula is provided in subsection D.1.2.

C.4.6.3 Derivatives of the KLD between TN and uniform distributions

The derivation of the derivatives of the Kullback-Leibler divergence (KLD) between a TN distribution and a uniform distribution enables to assess the influence of the associated loss term in the variational autoencoder (VAE) objective function (see subsection 7.3.4). These derivatives are depicted in Figure C.7. During training, low values of the KLD loss term are promoted, and the derivatives of this term indicate the influence over this loss term of the μ and σ parameters of the TN, and how it is preferentially minimized during optimization. In this case, the derivative w.r.t. σ is always negative, i.e. increasing always decreases the KLD. The derivative w.r.t. μ is positive for $\mu > 0.5$ and negative otherwise, therefore promoting the value $\mu = 0.5$. Comparing the magnitude of the derivative w.r.t. σ is overall greater than the derivative w.r.t. μ, highlighting that σ has a stronger influence on the KLD loss term than μ.

Formula 3 (Partial derivative w.r.t. σ of the KLD of a TN distribution from a uniform distribution)**.** *Let f_1 and f_2 denote the PDF of X_1 and X_2, a TN and a uniform distributions, such that:*
$X_1 \sim \mathcal{TN}(\mu, \sigma, a, b)$ *and* $X_2 \sim \mathcal{U}(a, b)$*. Then*

$$\frac{\partial}{\partial \sigma} \mathrm{D}_{\mathrm{KL}}(X_1 \| X_2) = -\frac{1}{2\sigma\eta}\left(-K - \alpha^2 \dot{\varphi}(\alpha) + \beta^2 \dot{\varphi}(\beta) - \frac{K^2}{\eta}\right) - \frac{K+\eta}{\sigma\eta} \tag{C.10}$$

Formula 4 (Partial derivative w.r.t. μ of the KLD of a TN distribution from a uniform distribution)**.** *Let f_1 and f_2 denote the PDF of X_1 and X_2, a TN and a uniform distributions, such that:*
$X_1 \sim \mathcal{TN}(\mu, \sigma, a, b)$ *and* $X_2 \sim \mathcal{U}(a, b)$*. Then*

$$\frac{\partial}{\partial \mu} \mathrm{D}_{\mathrm{KL}}(X_1 \| X_2) = -\frac{1}{2\sigma\eta}\left(-\frac{KN}{\eta} - N - \alpha\dot{\varphi}(\alpha) + \beta\dot{\varphi}(\beta)\right) - \frac{N}{\sigma\eta} \tag{C.11}$$

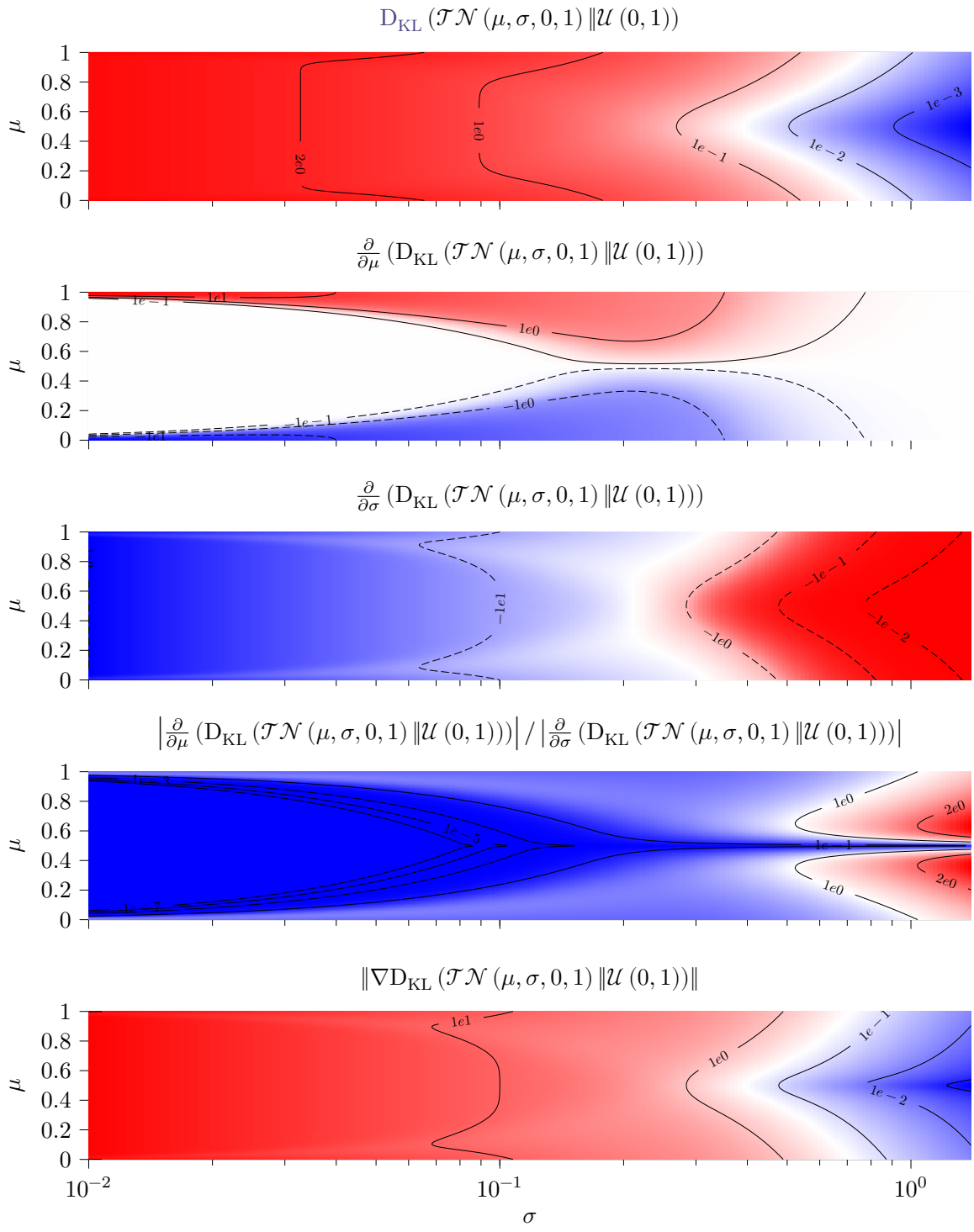


Figure C.7: KLD of a TN from a uniform distribution over the definition interval $[0, 1]$, its derivatives w.r.t. μ and σ, the ratio of these derivatives and the norm of the gradient.

Appendix D

Proofs

Contents

D.1 Kullback-Leibler divergences with the truncated Normal distribution

To derive the KLD of a TN distribution from a TN or uniform distribution, the original integral is split:

$$\mathrm{D}_{\mathrm{KL}}\left(X_1 \| X_2\right) = \int_a^b f_1(x) \ln \frac{f_1(x)}{f_2(x)} \,\mathrm{d}x = \underbrace{\int_a^b f_1(x) \ln f_1(x) \,\mathrm{d}x}_{I_1} - \underbrace{\int_a^b f_1(x) \ln f_2(x) \,\mathrm{d}x}_{I_2} \tag{D.1}$$

and the terms I_1 and I_2 are calculated separately. The computation of these terms involve the two first moments[1] of the TN distribution. For $X_i \sim \mathcal{TN}\left(\mu_i, \sigma_i, a, b\right)$, the first-order moment is:

$$\langle X_i \rangle = \mu + \frac{\sigma}{\eta} N,$$

and the second-order moment is:

$$\langle X_i^2 \rangle = \sigma_i^2 + \frac{\sigma_i^2}{\eta_i} K_i + \mu_i^2 + \frac{2\mu_i \sigma_i}{\eta_i} N_i,$$

with $\eta_i = \Phi(\beta_i) - \Phi(\alpha_i)$, $\alpha_i = \frac{a-\mu_i}{\sigma_i}$, $\beta_i = \frac{b-\mu_i}{\sigma_i}$, $K_i = \alpha_i \varphi(\alpha_i) - \beta_i \varphi(\beta_i)$ and $N_i = \varphi(\alpha_i) - \varphi(\beta_i)$.

D.1.1 Truncated normal distribution and truncated normal distribution

Proof. The KLD is split into two terms as shown in equation (D.1): $\mathrm{D}_{\mathrm{KL}} = I_1 - I_2$. The first term I_1 is:

$$I_1 = \int_a^b f_1(x) \ln f_1(x) dx = -\ln\left(\sigma_1 \eta_1\right) - \frac{1}{2} \ln\left(2\pi\right) - \frac{\mu_1^2}{2\sigma_1^2} - \frac{1}{2\sigma_1^2} \left\langle X_1^2 \right\rangle + \frac{\mu_1}{\sigma_1^2} \left\langle X_1 \right\rangle. \tag{D.2}$$

[1] If it exists, the nth order moment of a random variable X is defined as $\langle X^n \rangle = \int x^n f_X(x) \,\mathrm{d}x$.

Then:

$$
\begin{aligned}
-\frac{1}{2\sigma_1^2}\left\langle X_1^2\right\rangle + \frac{\mu_1}{\sigma_1^2}\left\langle X_1\right\rangle &= -\frac{1}{2\sigma_1^2}\left(\sigma_1^2 + \frac{\sigma_1^2}{\eta_1}K_1 + \mu_1^2 + \frac{2\mu_1\sigma_1}{\eta_1}N_1\right) + \frac{\mu_1}{\sigma_1^2}\left(\mu_1 + \frac{\sigma_1}{\eta_1}N_1\right) \\
&= -\frac{1}{2} - \frac{K_1}{2\eta_1} - \frac{\mu_1^2}{2\sigma_1^2} - \frac{\mu_1}{\sigma_1\eta_1}N_1 + \frac{\mu_1^2}{\sigma_1^2} + \frac{\mu_1}{\sigma_1\eta_1}N_1 \\
&= -\frac{1}{2} - \frac{K_1}{2\eta_1} + \frac{\mu_1^2}{2\sigma_1^2},
\end{aligned}
\tag{D.3}
$$

therefore:

$$
I_1 = -\frac{1}{2}\ln(2\pi) - \frac{1}{2} - \ln(\sigma_1\eta_1) - \frac{K_1}{2\eta_1}. \tag{D.4}
$$

The second term I_2 is:

$$
I_2 = \int_a^b f_1(x)\ln f_2(x)dx = -\ln(\sigma_2\eta_2) - \frac{1}{2}\ln(2\pi) - \frac{\mu_2^2}{2\sigma_2^2} - \frac{1}{2\sigma_2^2}\left\langle f_1^2\right\rangle + \frac{\mu_2}{\sigma_2^2}\left\langle f_1\right\rangle
$$

Then, developing and refactoring yields:

$$
\begin{aligned}
-\frac{1}{2\sigma_2^2}\left\langle f_1^2\right\rangle + \frac{\mu_2}{\sigma_2^2}\left\langle f_1\right\rangle &= -\frac{1}{2\sigma_2^2}\left(\sigma_1^2 + \frac{\sigma_1^2}{\eta_1}K_1 + \mu_1^2 + \frac{2\mu_1\sigma_1}{\eta_1}N_1\right) + \frac{\mu_2}{\sigma_2^2}\left(\mu_1 + \frac{\sigma_1}{\eta_1}N_1\right) \\
&= -\frac{\sigma_1^2}{2\sigma_2^2} - \frac{\sigma_1^2K_1}{2\sigma_2^2\eta_1} - \frac{\mu_1^2}{2\sigma_2^2} - \frac{\mu_1\sigma_1}{\sigma_2^2\eta_1}N_1 + \frac{\mu_1\mu_2}{\sigma_2^2} + \frac{\mu_2\sigma_1}{\sigma_2^2\eta_1}N_1 \\
&= -\frac{\sigma_1^2}{2\sigma_2^2} - \frac{\sigma_1^2K_1}{2\sigma_2^2\eta_1} - \frac{\mu_1^2}{2\sigma_2^2} - (\mu_1-\mu_2)\frac{\sigma_1}{\sigma_2^2\eta_1}N_1 + \frac{\mu_1\mu_2}{\sigma_2^2},
\end{aligned}
$$

and:

$$
\begin{aligned}
I_2 &= -\ln(\sigma_2\eta_2) - \frac{1}{2}\ln(2\pi) - \frac{\mu_2^2}{2\sigma_2^2} - \frac{\sigma_1^2}{2\sigma_2^2} - \frac{\sigma_1^2K_1}{2\sigma_2^2\eta_1} - \frac{\mu_1^2}{2\sigma_2^2} - (\mu_1-\mu_2)\frac{\sigma_1}{\sigma_2^2\eta_1}N_1 + \frac{\mu_1\mu_2}{\sigma_2^2} \\
&= -\ln(\sigma_2\eta_2) - \frac{1}{2}\ln(2\pi) - \frac{\mu_1^2+\mu_2^2}{2\sigma_2^2} - \frac{\sigma_1^2}{2\sigma_2^2} - \frac{\sigma_1^2K_1}{2\sigma_2^2\eta_1} - (\mu_1-\mu_2)\frac{\sigma_1}{\sigma_2^2\eta_1}N_1 + \frac{\mu_1\mu_2}{\sigma_2^2}.
\end{aligned}
$$

Finally:

$$
\begin{aligned}
\mathrm{D_{KL}}(X_1\|X_2) &= I_1 - I_2 \\
&= -\frac{1}{2}\ln(2\pi) - \frac{1}{2} - \ln(\sigma_1\eta_1) - \frac{K_1}{2\eta_1} + \ln(\sigma_2\eta_2) + \frac{1}{2}\ln(2\pi) + \frac{\mu_1^2+\mu_2^2}{2\sigma_2^2} + \frac{\sigma_1^2}{2\sigma_2^2} \\
&\quad + \frac{\sigma_1^2K_1}{2\sigma_2^2\eta_1} + (\mu_1-\mu_2)\frac{\sigma_1}{\sigma_2^2\eta_1}N_1 - \frac{\mu_1\mu_2}{\sigma_2^2} \\
&= -\frac{1}{2} - \ln\left(\frac{\sigma_1\eta_1}{\sigma_2\eta_2}\right) - \frac{K_1}{2\eta_1}\left(1 - \frac{\sigma_1^2}{\sigma_2^2}\right) + \frac{(\mu_1-\mu_2)^2}{2\sigma_2^2} + \frac{\sigma_1^2}{2\sigma_2^2} + (\mu_1-\mu_2)\frac{\sigma_1}{\sigma_2^2\eta_1}N_1.
\end{aligned}
$$

□

D.1.2 Truncated normal distribution and uniform distribution

Proof. The KLD is split into two terms as shown in equation (D.1): $\mathrm{D_{KL}} = I_1 - I_2$. The first term is (see equations (D.2), (D.3), (D.4)):

$$
I_1 = -\frac{1}{2} - \frac{1}{2}\ln(2\pi) - \ln(\sigma\eta) - \frac{K}{2\eta}.
$$

The second term is:

$$\begin{aligned} I_2 = \int_a^b f_1(x) \ln f_2(x) dx &= \int_a^b f_1(x) \ln \frac{\mathbb{1}_{[a,b]}}{b-a} dx \\ &= -\ln(b-a) \int_a^b f_1(x) dx \\ &= -\ln(b-a). \end{aligned}$$

Therefore:

$$\mathrm{D}_{\mathrm{KL}}(X_1 \| X_2) = -\frac{1}{2} - \frac{1}{2}\ln(2\pi) - \ln(\sigma\eta) - \frac{K}{2\eta} + \ln(b-a).$$

□

D.2 Permutation of gradient and expectation operators

Let there be a continuous random variable $\mathbf{x}$ with a $\boldsymbol{\theta}$-parameterized density $p_{\boldsymbol{\theta}}(\boldsymbol{x})$. Let there be $f : \mathbb{X} \mapsto \mathbb{R}$ with sufficient regularity. Then

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \mathrm{E}_{\mathbf{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x})}[f(\mathbf{x})] &= \nabla_{\boldsymbol{\theta}} \int_{\boldsymbol{x} \in \mathbb{X}} f(\boldsymbol{x}) p_{\boldsymbol{\theta}}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \\ &\underset{(1)}{=} \int_{\boldsymbol{x} \in \mathbb{X}} f(\boldsymbol{x}) \nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \\ &\underset{(2)}{=} \int_{\boldsymbol{x} \in \mathbb{X}} f(\boldsymbol{x}) p_{\boldsymbol{\theta}}(\boldsymbol{x}) \nabla_{\boldsymbol{\theta}} \ln p_{\boldsymbol{\theta}}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \end{aligned} \tag{D.5}$$

Proof. The permutation of gradient and integral sign in the vectorial setting in step (1) is derived from the application of the theorem of interchange of integration and differentiation on with scalar random variable and parameter. This theorem requires that for a given function $h : \mathbb{X} \times \boldsymbol{\Theta}$:

1. $\forall \theta \in \boldsymbol{\Theta}$, the function $x \mapsto h(x, \theta)$ is Lebesgue-integrable over x.
2. $\forall x \in \mathbb{X}$ the function $\theta \mapsto h(x, \theta)$ is differentiable. The derivative is $\dfrac{\partial}{\partial \theta}(h(x, \theta))$
3. $\exists g : \mathbb{X} \mapsto \mathbb{R}^+$ measurable and Lebesgue integrable that uniformly dominates this derivative:
$$\forall x \in \mathbb{X}, \forall \theta \in \boldsymbol{\Theta}, \quad \left| \frac{\partial}{\partial \theta}(h(x, \theta)) \right| \leq g(x)$$

Then, $x \mapsto \dfrac{\partial}{\partial \theta}(h(x, \theta))$ is Lebesgues-integrable and

$$\forall x \in \mathbb{X}, \quad \forall \theta \in \boldsymbol{\Theta}, \quad \frac{\partial}{\partial \theta} \int_{x \in \mathbb{X}} (h(x, \theta)) \, \mathrm{d}x = \int_{x \in \mathbb{X}} \frac{\partial}{\partial \theta}(h(x, \theta)) \, \mathrm{d}x$$

In the current setting, with $h(x, \theta) = p_\theta(x) f(x)$, the condition 1 is equivalent to $f(\mathrm{x})$ having a finite expectation (this excludes some pathological cases such as $\mathrm{x} \sim \mathcal{U}(0, 1)$ with $f(x) = \frac{1}{x}$, or Cauchy distributions). Condition 2 only depends on the density of x. Most parametric distributions are differentiable with respect to their parameters. Condition number 3 is case specific.

Although proving the applicability of this theorem in the general case is challenging, it is considered that all conditions are met for the purposes of this work.

In step (2) the identity $\nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\boldsymbol{x}) = p_{\boldsymbol{\theta}}(\boldsymbol{x}) \nabla_{\boldsymbol{\theta}} \ln p_{\boldsymbol{\theta}}(\boldsymbol{x})$ is the generalization of the logarithmic derivative to the multi-dimensional case. □

Appendix E

Complementary results

Contents

E.1 PROSAIL variable joint distributions

E.1.1 Sampling observation angles

In Figure E.1 are shown the histograms and correlations of the observation and solar angular configuration samples obtained by simulating the orbital motion of S2, by uniformly drawing dates and locations within S2 operational range Weiss and Baret [2016]. These samples are used along with PROSAIL input parameters to simulate canopy reflectance spectra (see section 5.1).

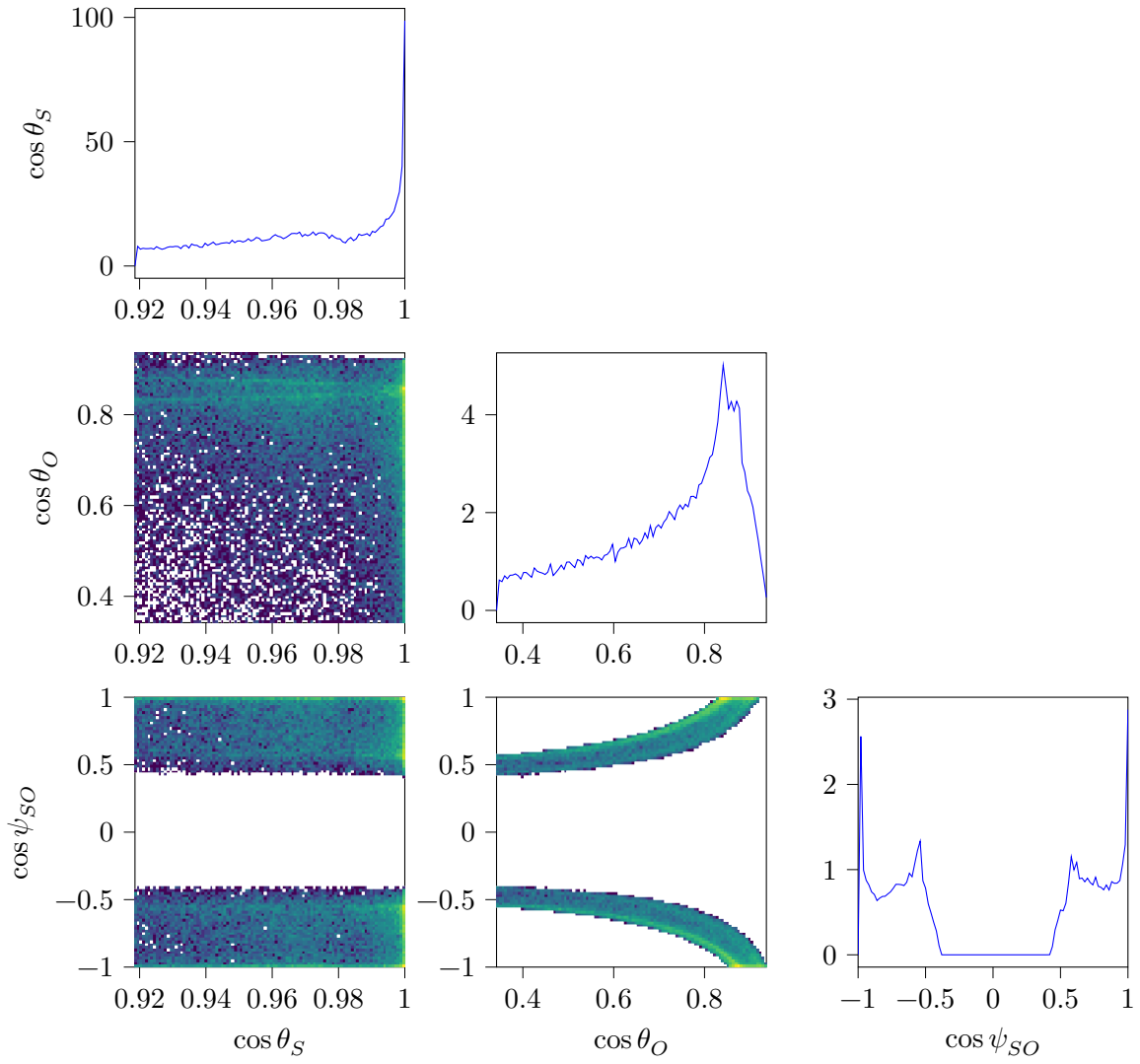


Figure E.1: Distribution of S2 observation angles for simulating a training data-set with PROSAIL.

E.1.2 PROSAIL variable sampling with co-distributions

The Figure E.2 and Figure E.3 show the histograms and correlations of the PROSAIL variables samples that are respectively drawn using the co-distribution functions type 1 and 2 (see subsection 5.1.2). These co-distribution functions introduce a correlation between the PROSAIL variables and the leaf area index (LAI).

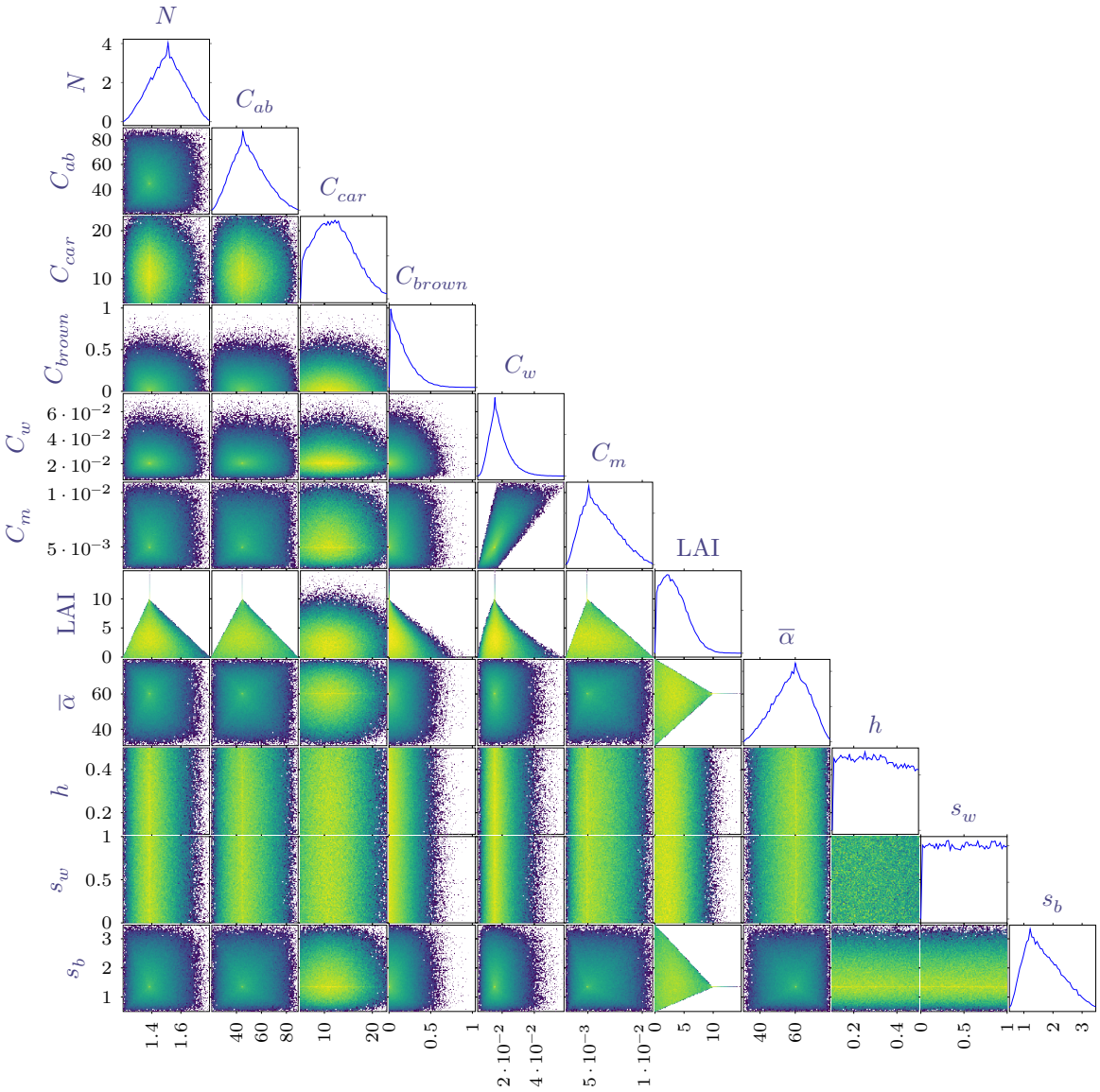


Figure E.2: Pair plot of PROSAIL input variables sampled with co-distribution type 1

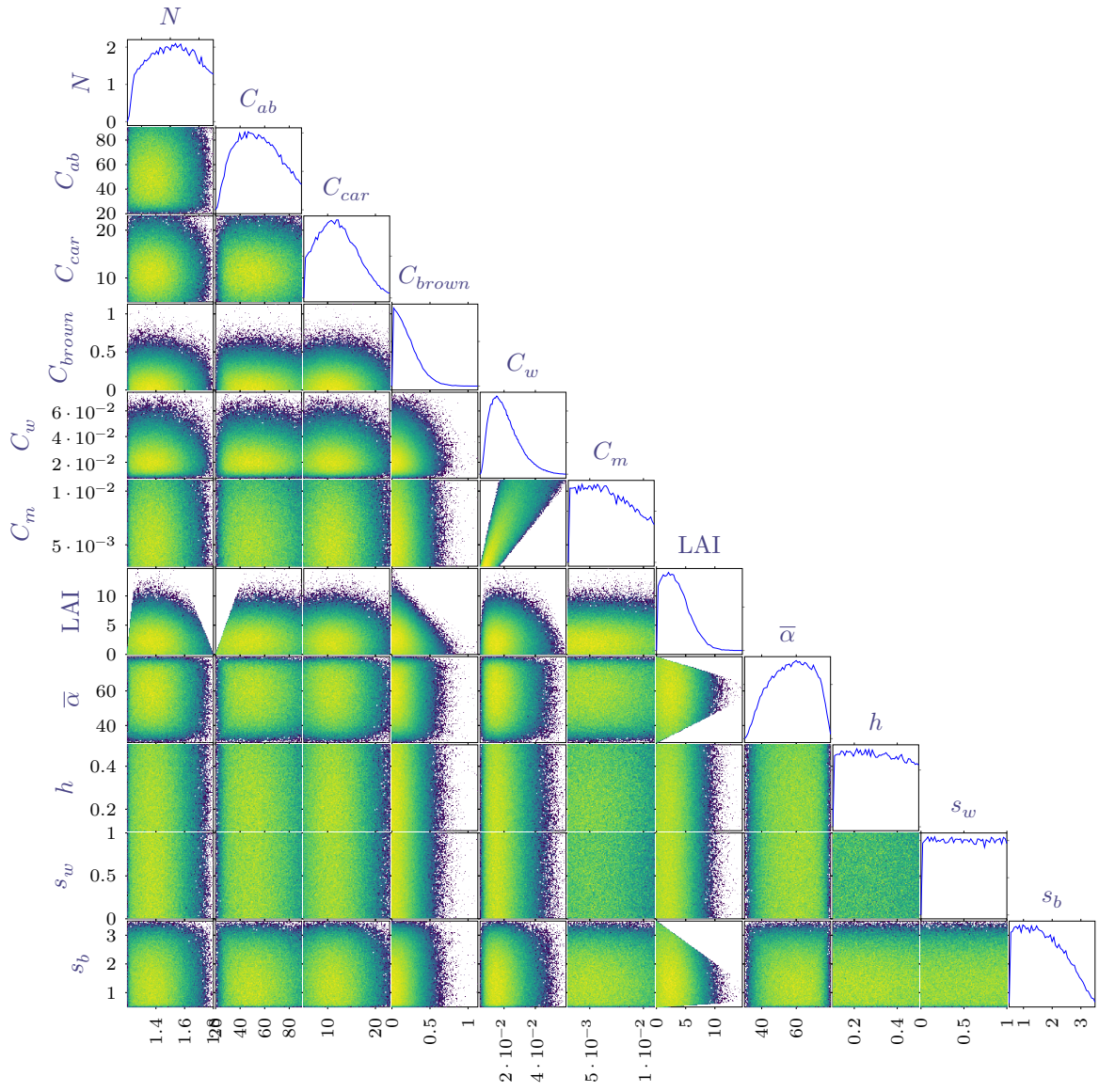


Figure E.3: Pair plot of PROSAIL input variables sampled with co-distribution type 2

E.2 Gradient-based sensitivity analysis of PROSAIL

In Figure E.4 are shown the box-plots of the gradients of S2 bands w.r.t. the input parameters of PROSAIL, computed from simulations. The use of the differentiable implementation of PROSAIL (see section 4.5) enabled to compute these gradients by using automatic differentiation. These results highlight several properties of the input parameters of PROSAIL. The chlorophyll content C_{ab} has no influence (zero gradient) over low frequency bands, beyond the red-edge bands (`B5`, `B6`, `B7`). The carotenoid content C_{car} only influences weakly the `B2` and `B3`, highlighting why this parameter is difficult to estimate. The equivalent water thickness C_w has greater gradient magnitude in short wavelength infra-red (SWIR) bands (`B11` and `B12`), which is expected since water strongly absorbs radiation in this part of the spectrum. Also, the soil wetness factor s_w and brightness factor s_b show the highest range of gradient values. This accounts for different situations in which the soil is either bare and visible, or occulted (partially or fully) by vegetation.

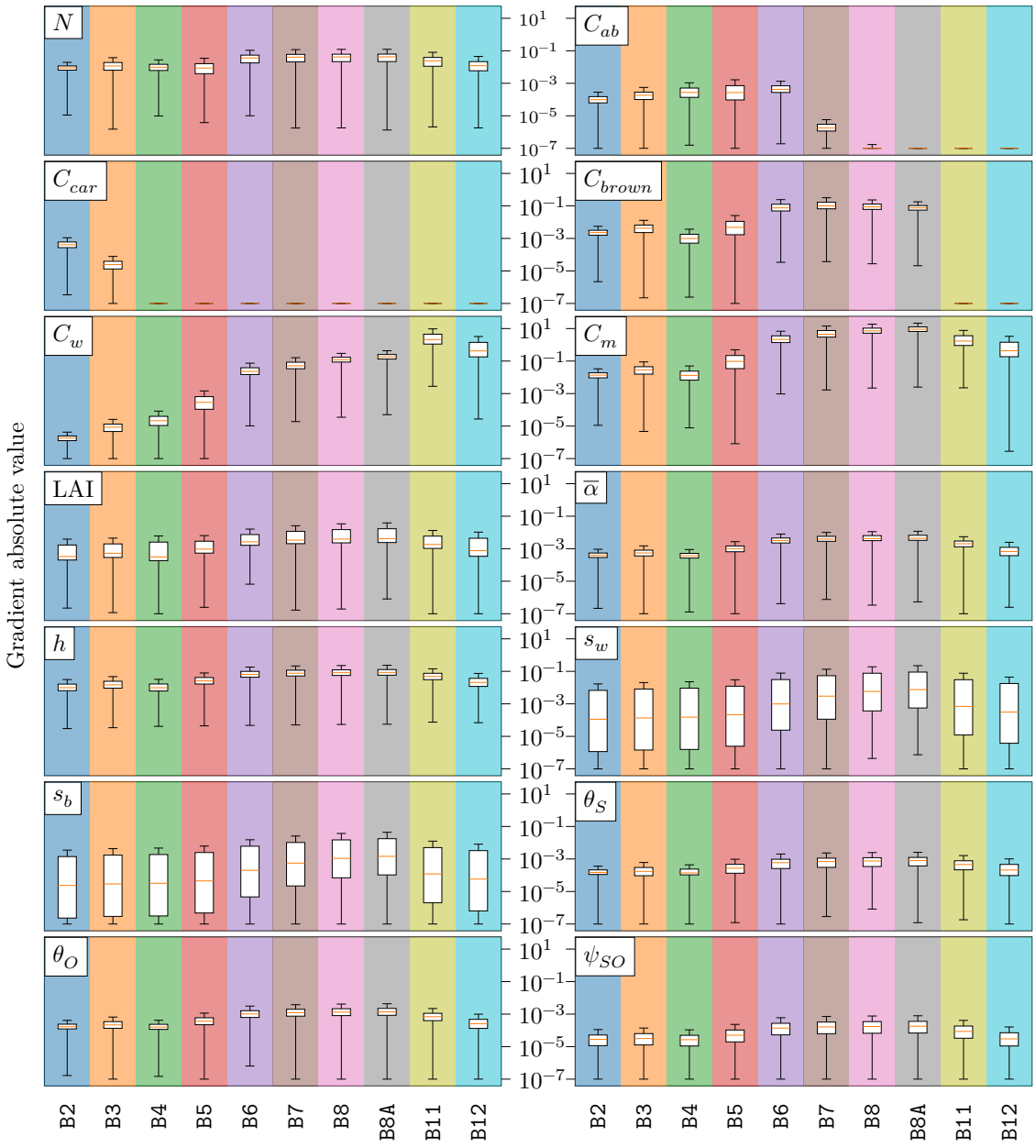


Figure E.4: Box-plots of the absolute value of the gradients of simulated S2 bands w.r.t. PROSAIL parameters (10^4 simulations).

E.3 Inversion of the double-logistic phenological model

E.3.1 Reconstruction of S2 time series with Pheno-VAE

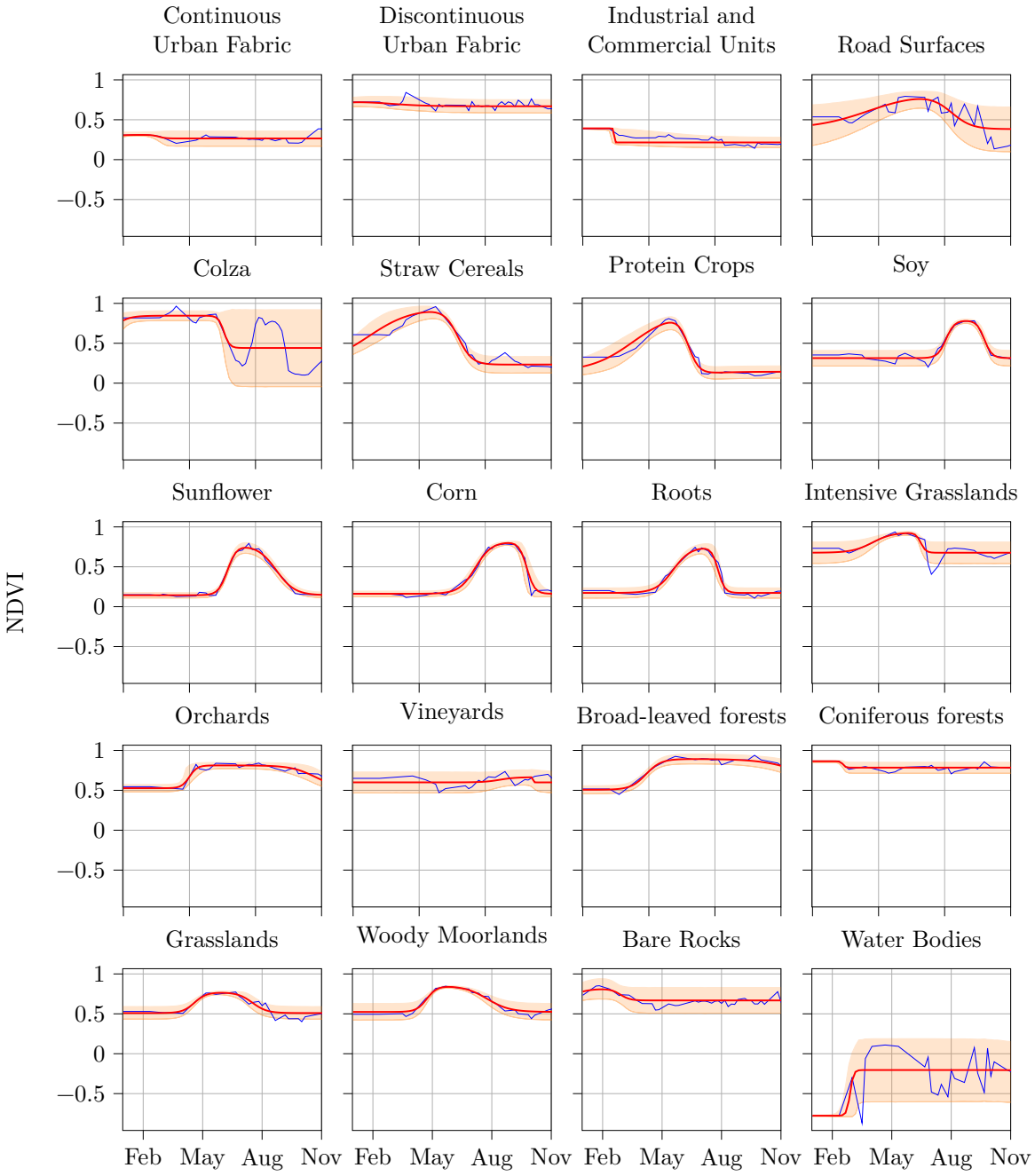


Figure E.5: Examples of reconstructions of Sentinel-2 NDVI time series with Pheno-VAE trained on S2 data-set. Blue: 5-days interpolated S2 time series. Red: Reconstruction of the mode of phenological distribution. Orange: 5^{th}-95^{th} percentile interval.

E.3.2 Box-plots of the absolute error

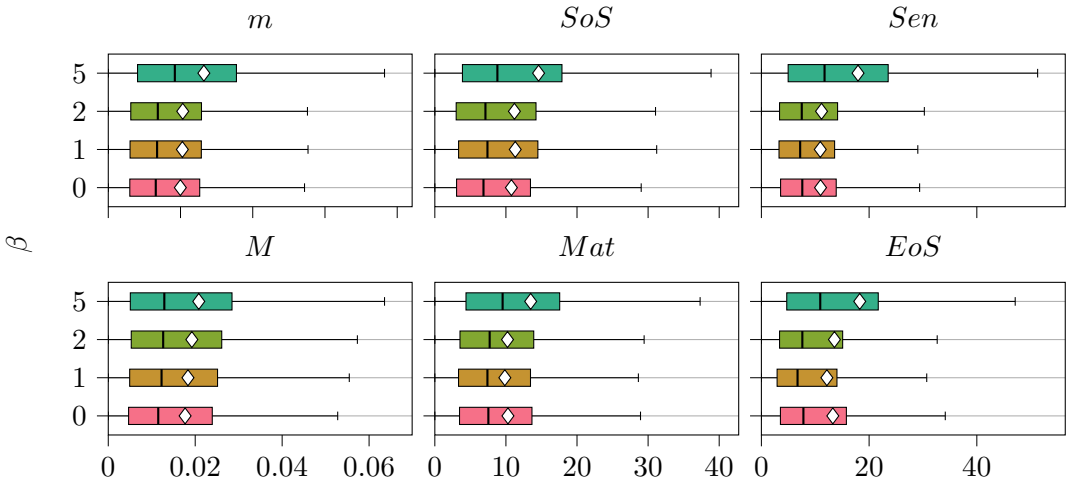


Figure E.6: Box-plot of the absolute error of inference of the 6 phenological parameters for Pheno-VAE trained on S2 Data-set, with various settings of the coefficient β of the KLD loss term. Box-plots are drawn from the results of the best fold of each method, in terms of the end of season (*EoS*) mean absolute error (MAE). The white square for each box plot is the MAE. Absolute errors are comparable, except with $\beta = 5$, with a higher error.

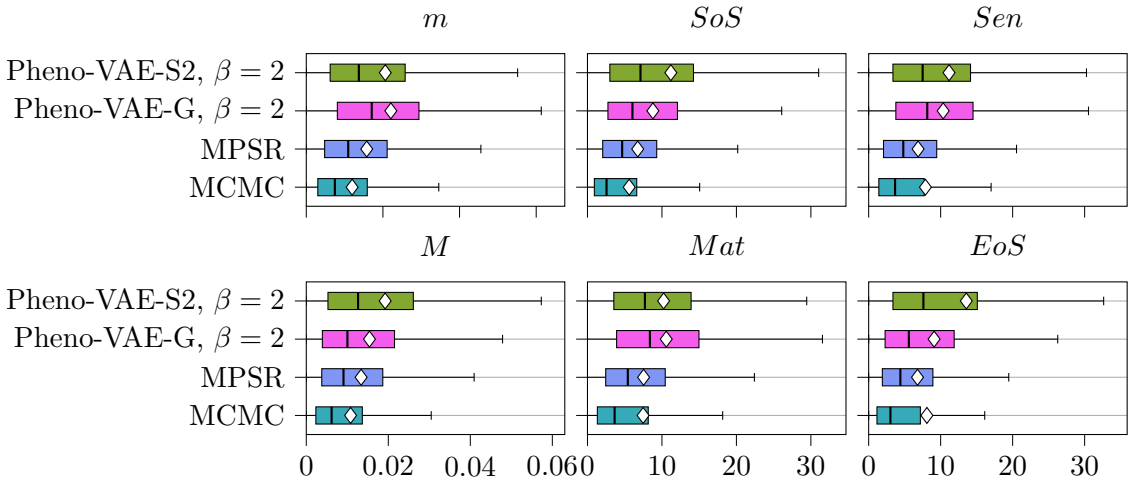


Figure E.7: Box-plot of the absolute error of inference of the 6 phenological parameters for Markov Chain Monte Carlo (MCMC), multiple probabilistic supervised regression (MPSR), curve fitting (CF) and Pheno-VAE (with $\beta = 2$, trained on the S2 or simulated data-set). Box-plots are drawn from the results of the best fold of each method, in terms of the *EoS* MAE. The white square for each box plot is the MAE. Absolute errors are the lowest for MCMC and MPSR, and comparable for both Pheno-VAE.

E.3.3 PICP as a function of the confidence level

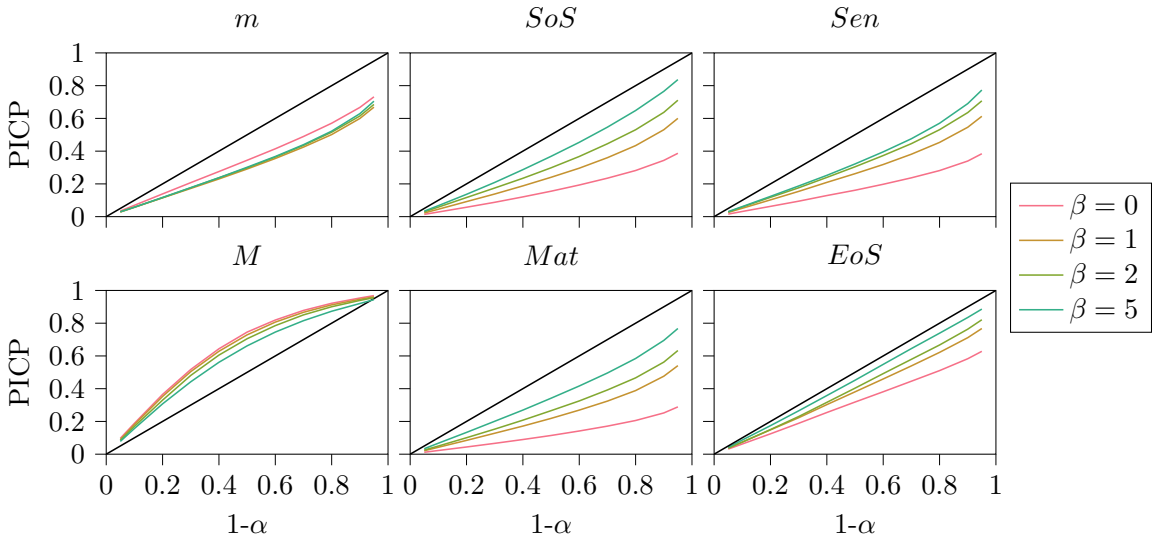


Figure E.8: prediction interval coverage probability (PICP) vs $1-\alpha$ for Pheno-VAE trained on S2 Data-set, with various settings of the coefficient β of the KLD loss term. The more β increases, the more the PICP increases at constant confidence level $1-\alpha$.

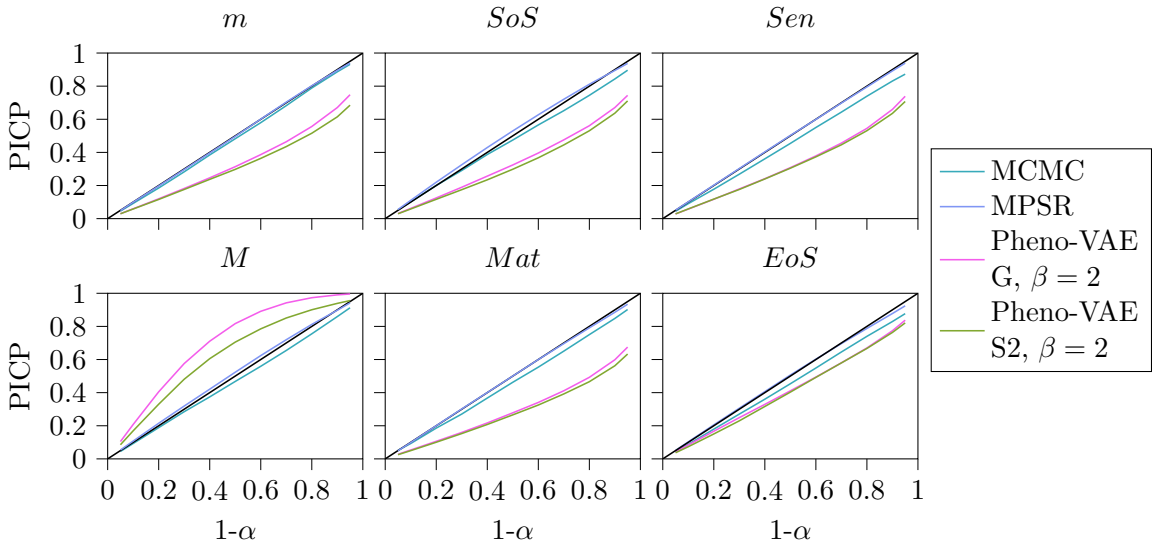


Figure E.9: PICP vs $1-\alpha$ for MCMC, MPSR and Pheno-VAE (with $\beta = 2$, trained on the S2 or simulated data-set). The PICP curves of MPSR and MCMC are very close to PICP=α for all α, while Pheno-VAE underestimates uncertainty for all confidence levels, for all phenological variables, except for minimum NDVI level (m) where uncertainty is overestimated.

E.3.4 MPIW as a function of the confidence level

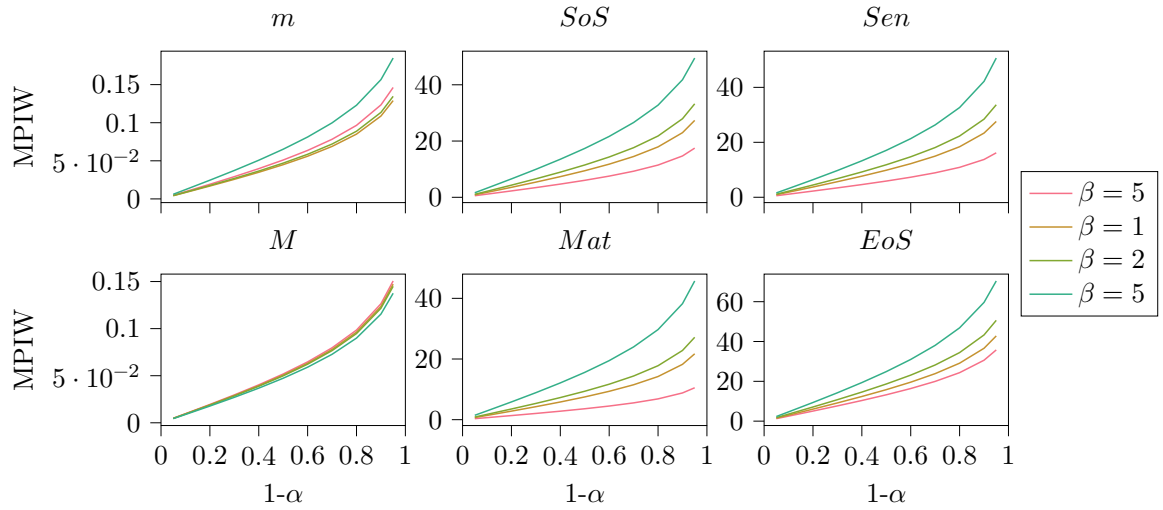


Figure E.10: mean prediction interval width (MPIW) vs $1-\alpha$ for Pheno-VAE trained on S2 Data-set, with various settings of the coefficient β of the KLD loss term. The more β increases, the more the MPIW increases at constant confidence level $1-\alpha$.

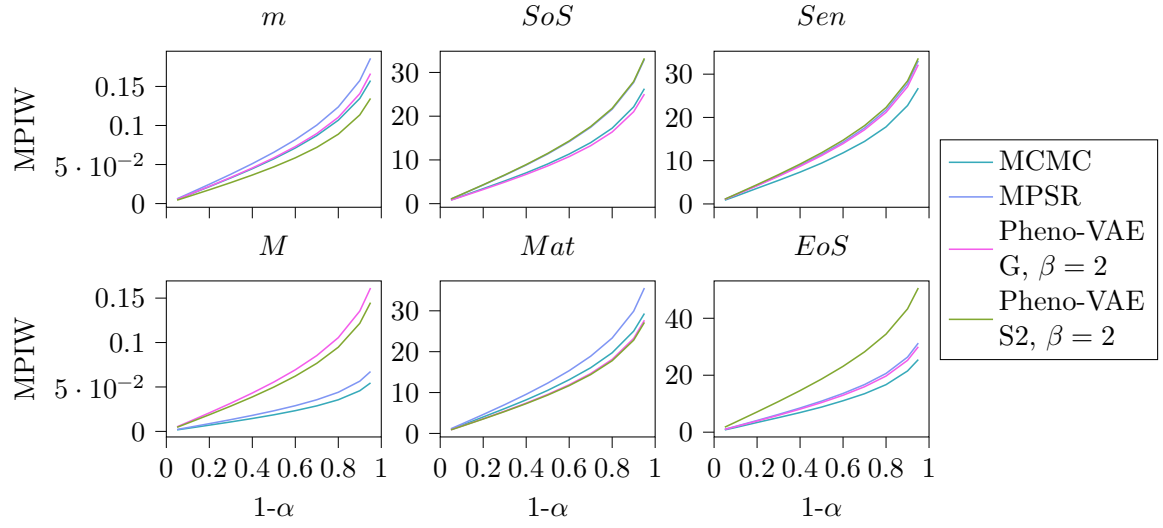


Figure E.11: MPIW vs $1-\alpha$ for MCMC, MPSR and Pheno-VAE (with $\beta = 2$, trained on the S2 or simulated data-set). prediction interval sizes are similar for all methods, except for *m*, where prediction intervals are larger for Pheno-VAE, and for the *EoS* of Pheno-VAE trained on the S2 data-set, that also has larger prediction intervals.

E.3.5 Box-plots of the prediction interval width

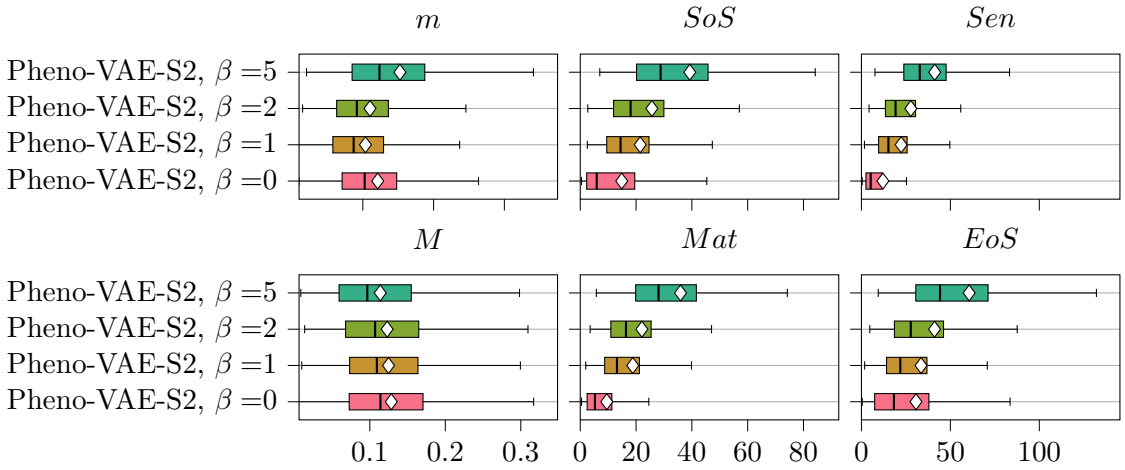


Figure E.12: Box-plot of the prediction interval width (PIW) with a confidence level $1-\alpha = 0.90$ for Pheno-VAE trained on S2 Data-set, with various settings of the coefficient β of the KLD loss term. Box-plots are drawn fro the results of the best fold of each method, in terms of the *EoS* MAE. The white square for each box plot is the MPIW. For phenological dates the PIW increases with β.

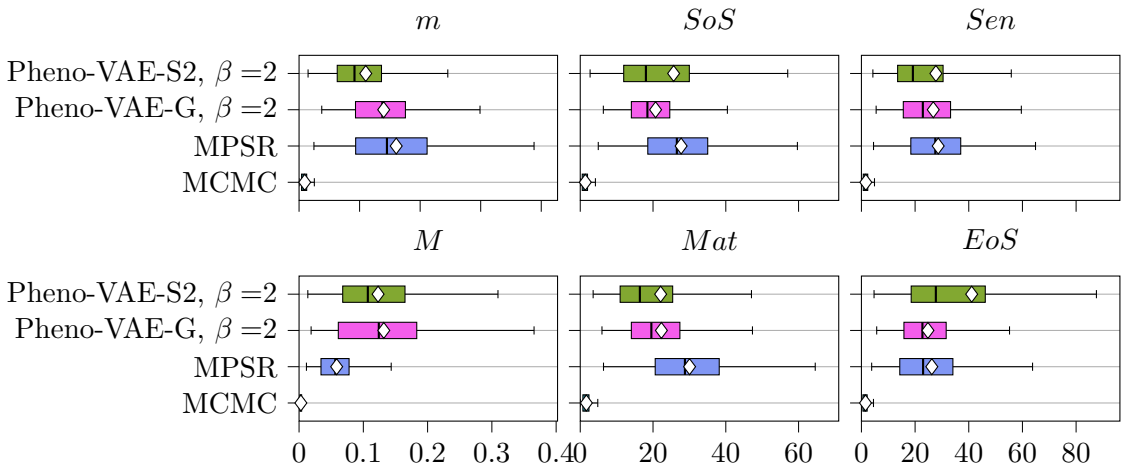


Figure E.13: Box-plot of the PIW with a confidence level $1-\alpha = 0.90$ for the 6 phenological parameters for MCMC, MPSR and Pheno-VAE (with $\beta = 2$, trained on the S2 or simulated data-set). Box-plots are drawn from the results of the best fold of each method, in terms of the *EoS* MAE. The white square for each box plot is the MPIW. MCMC infers significantly smaller PIW than the other methods that are comparable

Appendix F

Glossary

absorbance Decimal logarithm of the ratio between incident and transmitted radiative energy, i.e. of the transmittance.

absorbtance Ratio of incident radiation flux absorbed by a body.

allometry The study of the relationship of body size to shape, anatomy, physiology and finally behaviour in living organisms.

azimuth angle The azimuth angle of an object as seen from the a given point-of-view is the horizontal angle between the object and a cardinal direction, usually the North.

β-VAE A variational autoencoder for which the evidence lower bound (ELBO) objective function has been modified by introducing a coefficient β for the KLD regularization term, so as to improve disentanglement (see subsubsection 6.5.2.1).

bijection A bijection, bijective function, or one-to-one correspondence between two mathematical sets $\mathbb{X}_1$ and $\mathbb{X}_2$ is a function such that each element of the second set $\mathbb{X}_2$ is mapped to from exactly one element of the first set $\mathbb{X}_1$.

canopy clumping effect Non randomness of foliage distribution within the canopy, due to the leaves tending to being grouped (clumped) together. This effect tends to reduce the apparent vegetation elements area, and vegetation indices tend to be underestimated.

dicotyledon Angiosperm plants that have two cotyledons (embryonic leaf). Their leaf veins usually banch from a central vein and interlace. Their flower parts are ususally by multiples of 4 or 5. Also called *dicots*, they include oak, legumes, peas.

gap fraction Fraction of sky visible not obstructed by the canopy.

gradient descent Optimization approach using the gradients of a differentiable objective function to update a target parameter such that (s.t.) this function is minimized w.r.t. this parameter.

heteroscedastic A set of random variables is heteroscedastic if all the random variables do not have the same finite variance.

homoscedastic A set of random variables is homoscedastic if all the random variables have the same finite variance.

Lambertian Qualifies an isotropic light reflection, for which there is no dependance on incident light or observation direction.

monocotyledon Angiosperm plants that have a single cotyledon (embryonic leaf). Their flower petals are in multiples of 3. Their leaves are long and thin with parallel veins. Also called *monocots*, they include wheat, maize, garlic, palm trees.

Pheno-VAE VAE with a phenological model as a deterministic, physics-based decoder, which is trained to infer the posterior distribution of phenological parameter from NDVI time series.

phenology Response of biological organisms to seasonal variations in environmental factors like light, temperature and precipitation.

PROSAIL-VAE VAE with the PROSAIL radiative transfer model (RTM) as a deterministic, physics-based decoder, which is trained to infer the posterior distribution of bio-physical parameters leaf and canopy from S2 pixels and images.

PV* Trained PROSAIL-VAE model selected for its good overall performance in in LAI and canopy chlorophyll content (CCC) estimation and in prediction interval inference.

red-edge The red-edge is a region of the near infra-red in which the reflectance of vegetation changes abruptly. Chlorophyll absorbs most of the light in the frontier between visible and near infra-red (NIR) (low reflectance), but becomes transparent around 700 nm, enabling leaves to reflect more light (higher reflectance).

reflectance Ratio of reflected over incident radiation flux by a body.

residual The difference $\boldsymbol{y} - \mathcal{F}(\boldsymbol{x})$ between an observed value $\boldsymbol{y}$ and the fitted value provided by a model $\mathcal{F}(\boldsymbol{x})$.

scattering Deflection of light in a propagation medium due to small particules.

transmittance Tatio of transmitted over incident radiation flux by a body.

zenith angle The zenith angle of an object as seen from the a given point-of-view is the angle between the object and the local vertical.

Appendix G

Acronyms

AE autoencoder

AEVB auto-encoding variational Bayes

AI artificial intelligence

ANITI Artificial and Natural Intelligence Toulouse Institute

ANN artificial neural network

ANR Agence Nationale de la Recherche

AOT atmosphere optical thickness

BAI branch area index

BDGP bidirectional gap probability

BLAI brown LAI

BOA bottom-of-atmosphere

BOC bottom-of-canopy

BON biodiversity observation network

BP Biophysical Processor

BRDF bidirectional reflectance distribution function

BV bio-physical variables

BVNET biophysical variable neural network

CAP common agriculture policy

CAVI coordinate ascent variational inference

CCC canopy chlorophyll content

CCC_{eff} *effective* CCC

CDF cumulative distribution function

CES OSO Centre d'expertise scientifique sur l'occupation des sols

CESBIO Centre d'études spatiales de la biosphère

CF curve fitting

CNES Centre national d'études spatiales

CNN convolutional neural network

CNRS Centre national de la recherche scientifique

CPR cyclical plateau reduction

CPU central processing unit

CWC canopy water content

DCP digital cover photography

DHP digital hemispheric photography

DL deep learning

DLR Deutsches Zentrum für Luft und Raumfahrt (German Aerospace Center)

DOY day of year

e.g. *exempli gratia*, for instance

EBV essential biodiversity variable

ECV essential climate variable

ELBO evidence lower bound

EO Earth observation

EoS end of season

EOV essential ocean variable

ER epistemic representation

erf error function

inverf inverse error function

ESA European Space Agency

ESU elementary sampling unit

EU European Union

F-COVER fraction of vegetation cover

FAPAR fraction of absorbed photosynthetically active radiation

FIR finite impulse response

FRM4Veg fiducial reference measurements for vegetation

GAI green area index

GCOS Global Climate Observing System

GEO group on Earth observation

GLAI green LAI

GP Gaussian process

GPU graphical processing unit

GZD grid zone designator

HDI highest density interval

HPC high performance computing

i.e. *id est*, to say

i.i.d. independent and identically distributed

ICDF inverse cumulative distribution function

IIR infinite impulse response

INRAe Institut national de recherche pour l'agriculture, l'alimentation et l'environnement

IR infra-red

IRD Institut de recherche pour le développement

ISO International Standards Organization

ITS inverse transform sampling

KLD Kullback-Leibler divergence

KNN k-nearest-neighbors

LAI_{SAVI} leaf area index soil adjusted vegetation index

LAI leaf area index

LAI_{eff} *effective* LAI

laser light amplification by stimulated emission of radiation

LCC leaf chlorophyll content

LiDaR laser imaging detection and ranging

LIDF leaf inclination distribution function

LIDFA leaf inclination distribution function average

LLS linear least squares

lr learning rate

LULC land use and land cover

LUT lookup table

LWIR long wavelength infra-red

MAE mean absolute error

MAJA MACCS-ATCOR joint algorithm

MAP maximum a posteriori

Mat maturity

MC Monte Carlo

MCMC Markov Chain Monte Carlo

MCRL Monte Carlo reconstruction loss

MGRS Military Grid Reference System

MIBIT multiple initialization and best instance training

ML Machine Learning

MLE maximum likelihood estimation

MLP multi-layer perceptron

M maximum NDVI level

m minimum NDVI level

MPIW mean prediction interval width

MPSR multiple probabilistic supervised regression

MSE mean squared error

MSI multi spectral instrument

NATO North Atlantic Treaty Organazation

NDII normalized difference infrared index

$\mathbf{ND_{LMA}}$ normalized difference leaf mass area

NDVI normalized difference vegetation index

NIR near infra-red

NLL negative log-likelihood

NLLS non-linear least squares

OLS ordinary least squares

OMP Observatoire Midi Pyrénées

PAI plant area index

PCA principal components analysis

PDE partial differential equation

PDF probability distribution function

PICP prediction interval coverage probability

PINC prediction interval nominal coverage

PINN physics-informed neural network

PIW prediction interval width

radar radio detection and ranging

RAM random access memory

ReLU rectified linear unit

RF random forest

RGB red green blue

RMSE root mean squared error

ROI region of interest

R^2 coefficient of determination

RTM radiative transfer model

s.t. such that

S2 Sentinel-2

SAI stem area index

SAIL Scattering by Arbitrary Inclined Leaves

SAR synthetic aperture radar

SDU2E Sciences de l'Univers, de l'Environnement et de l'Espace

Sen senescence

SGVB stochastic gradient variational Bayes

SITS Satellite image time series

SL2P Simplified Level 2 Product Prototype Processor

SNAP Sentinel Application Platform

SNNR supervised neural network regression

SoS start of season

std standard deviation

SVD singular value decomposition

SVI stochastic variational inference

SVM support vector machine

SVR support vector regression

SWIR short wavelength infra-red

TN truncated normal

TOA top-of-atmosphere

TOC top-of-canopy

TRRA trust region reflective algorithm

UDD user-defined decoder

UT3 Université Toulouse III

UTM universal transverse Mercator

UV ultra-violet

VAE variational autoencoder

VNIR visible and near infra-red

w.r.t. with respect to

WAI woody area index

WGS84 World Geodetic System 1984

WLS weighted least squares

Appendix H

Notations

H.1 Notations of variables

H.1.1 Typesetting

Notations of variables is based on ISO standard 80000-2:2019. All variables are in italic by default. Scalar values have normal thickness and are written with lowercase letters. Vectors are written as lowercase bold letters. Matrices are uppercase. Tensors are uppercase sans-serif bold italic. Operators are typeset in roman (upright), and usually uppercase (exceptions are the differential operator d and the Gaussian density φ). Mathematical constants are written upright lowercase (e, π, γ). To distinguish a random variable from a sample, a random variable is written upright. This is the single deviation taken from the ISO standard. In the referenced chapters, all random variables are either scalars or vectors (i.e. there is no random matrix), and thus are all lowercase (contrary to a common practice of using uppercase random variables). Thus italic uppercase matrices are not confounded with upright uppercase operators. Finally, named variables from a given literature are exempted from this typesetting e.g. acronyms such as CCC, SoS, or variables identified by indexes such as C_{ab}, θ_S.

H.1.2 Variable indexing

For a matrix, indexing is straightforward. The element on row i and column j of a matrix X is denoted $x_{i,j}$. For vectors and random variables, there are multiple reasons a variable x may need indexing and associated notations:

- being a vector $\boldsymbol{x}$ component i: x_i,
- being a batch or a set element i: x_i,
- being a random variable x sample $x^{(i)}$.
- time t indexing x_t,
- spatial (planar) n, m indexing $x_{n,m}$.

This means that some quantities might need several indexes. When indexing a variable, the textual context usually specifies the associated meaning.

Overall, indexing indicates the context of a variable and doesn't change the typesetting related to the nature of a variable, e.g. a matrix X column j is a vector thus written $\boldsymbol{x}_j$. A vector $\boldsymbol{x}$ component i is a scalar denoted x_i. If the variable being indexed is to be highlighted instead of the particular component, parenthesis can be used. For instance the row i column j element of a matrix X is equivalently written $x_{i,j}$ or $(X)_{i,j}$. Another example is a batched tensor $\boldsymbol{\mathsf{X}}$ of latent vector samples. The batch element i, vector component j sample k is denoted $x_{i,j}^{(k)} = (\boldsymbol{\mathsf{X}})_{i,j}^{(k)}$

H.1.3 Usual notations

In the following Table H.1, the notations and meaning of notable quantities are detailed.

Table H.1: Special variables and their typesetting.

Description	Random variable		Sample / deterministic variable	
	scalar	vector	scalar	vector
Observed data	x	$\mathbf{x}$	x	$\boldsymbol{x}$
Label, encoding or reference data	-	-	y	$\boldsymbol{y}$
Latent variable	z	$\mathbf{z}$	z	$\boldsymbol{z}$
Variational parameter	-	-	λ	$\boldsymbol{\lambda}$
Frequentist or model (decoder) parameters	-	-	θ	$\boldsymbol{\theta}$
Encoder parameters	-	-	-	$\boldsymbol{\phi}$
VAE likelihood parameters	-	-	-	$\boldsymbol{\psi}$

H.2 Physical variables

Sign	Description	Unit
BAI	branch area index	
BLAI	brown LAI	
C_{ab}	Chlorophyll a+b concentration	$\mu g\,cm^{-2}$
C_{car}	Carotenoid concentration	$\mu g\,cm^{-2}$
C_{brown}	Brown pigments content	$\mu g\,cm^{-2}$
CCC	canopy chlorophyll content	$\mu g\,cm^{-2}$
CCC_{eff}	*effective* CCC	$\mu g\,cm^{-2}$
C_m	Dry matter content	$g\,cm^{-2}$
C_w	Water equivalent thickness	$g\,cm^{-3}$ / cm
CWC	canopy water content	$g\,cm^{-3}$ / cm
$C_{w,rel}$	relative water content	
EoS	end of season	DOY
GAI	green area index	
GLAI	green LAI	
h	Hotspot parameter	
LAI	leaf area index	
LAI_{eff}	*effective* LAI	
LCC	leaf chlorophyll content	$\mu g\,cm^{-2}$
$\overline{\alpha}$	leaf inclination distribution function average	
Mat	maturity	DOY
M	maximum NDVI level	
m	minimum NDVI level	
N	Leaf structure parameter	-
NDVI	normalized difference vegetation index	
θ_O	Observer zenith angle	deg
PAI	plant area index	
s_w	Soil brightness factor	
ψ_{SO}	Relative azimuth angle	deg
ρ	reflectance	
s_b	Soil brightness factor	
SAI	stem area index	
Sen	senescence	DOY
SoS	start of season	DOY
ψ_O	Observer azimuth angle	deg
ψ_S	Solar azimuth angle	deg
θ_S	Solar zenith angle	deg
WAI	woody area index	

H.3 Mathematics

$\mathcal{B}$ Bernoulli distribution

β beta distribution

$\mathbf{B}$ beta function

δ Dirac function

$\odot$ element-wise product between equal-sized vector or matrices.

H entropy

erf error function

erf^{-1} inverse error function

E expectation

$\mathcal{E}$ exponential distribution

Ei exponential integral

γ gamma function

∇ Gradient operator

$\mathbf{I}$ Identity matrix

$\mathbb{1}$ indicator function

J Jacobian matrix

D_{KL} Kullback-Leibler divergence

$\mathcal{S}_n$ Group of permutation matrices of size n.

$\mathcal{M}$ $\mathcal{M}_{n,m}(\mathbb{R})$ denotes the set of $n \times m$ real-valued matrices.

I mutual information

$\mathbb{N}$ set natural numbers

NLL negative log-likelihood

$\mathcal{K}$ Kumaraswamy distribution

$\mathcal{N}$ normal distribution

Φ CDF of the standard normal distribution

Φ^{-1} ICDF of the standard normal distribution

φ PDF of the standard normal distribution

$\mathbb{R}$ set of real numbers

ReLU rectified linear unit

R^2 coefficient of determination

$\mathcal{TN}$ truncated normal

$\mathcal{U}$ uniform distribution

$\mathbf{0}$ Zero matrix

H.4 Machine Learning

$\mathcal{D}$ data-set

ELBO evidence lower bound

$\mathcal{L}$ loss

lr learning rate

$\mathcal{D}_{\mathbf{IS}}$ Data-set of in-situ measurements and associated Sentinel-2 images

$\mathcal{D}_{\mathbf{S2}}$ Sentinel-2 images data-set for PROSAIL-VAE.

$\mathbf{x}$ observed (input) data, regressor

$\widehat{\boldsymbol{x}}$ generated, reconstructed (output) data

$\boldsymbol{y}$ label, regressand

$\mathbf{z}$ latent variable

H.5 Metrics

MAE	mean absolute error
MPIW	mean prediction interval width
MSE	mean squared error
PICP	prediction interval coverage probability
PICP	prediction interval nominal coverage
PIW	prediction interval width
RMSE	root mean squared error
std	standard deviation

Bibliography

M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021. ISSN 1566-2535. doi: \url{https://doi.org/10.1016/j.inffus.2021.05.008}.

Global Climate Observing System (GCOS). Systematic observation requirements for satellite-based products for climate 2011 update: Supplemental details to the satellite-based component of the "Implementation plan for the global observing system for climate in support of the UNFCCC (2010 update), 2011.

Sentinel User Handbook and Exploitation Tools (SUHET). Sentinel-2 user handbook. Technical report, ESA, July 2015. URL https://sentinel.esa.int/documents/247904/685211/Sentinel-2_User_Handbook.

R. Ak, V. Vitelli, and E. Zio. An interval-valued neural network approach for uncertainty quantification in short-term wind speed prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 26(11):2787–2800, 2015. doi: 10.1109/TNNLS.2015.2396933.

W. A. Allen, H. W. Gausman, A. J. Richardson, and J. R. Thomas. Interaction of isotropic light with a compact plant leaf. *Journal of the Optical Society of America*, 59(10):1376–1379, Oct. 1969. doi: 10.1364/JOSA.59.001376. URL https://opg.optica.org/abstract.cfm?URI=josa-59-10-1376.

W. A. Allen, H. W. Gausman, and A. J. Richardson. Mean effective optical constants of cotton leaves*. *Journal of the Optical Society of America*, 60(4):542–547, Apr. 1970. doi: 10.1364/JOSA.60.000542. URL https://opg.optica.org/abstract.cfm?URI=josa-60-4-542.

D. Altman, D. Machin, T. Bryant, and M. Gardner. *Statistics with confidence: confidence intervals and statistical guidelines*. John Wiley & Sons, 2013.

M. A. Aragon-Calvo. Self-supervised learning with physics-aware neural networks – i. galaxy model fitting. *Monthly Notices of the Royal Astronomical Society*, 498:3713–3719, 2020.

P. V. Arun and A. Karnieli. Learning of physically significant features from earth observation data: an illustration for crop classification and irrigation scheme detection. *Neural Computing and Applications*, 34(13):10929–10948, Mar. 2022. ISSN 1433-3058. doi: 10.1007/s00521-022-07019-5. URL http://dx.doi.org/10.1007/s00521-022-07019-5.

C. Atzberger. Development of an invertible forest reflectance model: The infor-model. In *A decade of trans-european remote sensing cooperation. Proceedings of the 20th EARSeL Symposium Dresden, Germany*, volume 14(16), pages 39–44, 2000.

H. Balzter, F. Gerard, C. George, G. Weedon, W. Grey, B. Combal, E. Bartholomé, S. Bartalev, and S. Los. Coupling of vegetation growing season anomalies and fire activity with hemispheric and regional-scale climate patterns in Central and East Siberia. *Journal*

of Climate, 20(15):3713–3729, 2007. doi: 10.1175/jcli4226. URL http://dx.doi.org/10.1175/JCLI4226.

F. Baret, B. de Solan, R. Lopez-Lozano, K. Ma, and M. Weiss. Gai estimates of row crops from downward looking digital photos taken perpendicular to rows at 57.5° zenith angle: Theoretical considerations based on 3d architecture models and application to wheat crops. *Agricultural and Forest Meteorology*, 150(11):1393–1401, 2010. ISSN 0168-1923. doi: \url{https://doi.org/10.1016/j.agrformet.2010.04.011}.

B. R. Barricelli, E. Casiraghi, and D. Fogli. A survey on digital twin: Definitions, characteristics, applications, and design implications. *IEEE Access*, 7:167653–167671, 2019. doi: 10.1109/ACCESS.2019.2953499.

L. Baudoux, J. Inglada, and C. Mallet. Multi-nomenclature, multi-resolution joint translation: an application to land-cover mapping. *International Journal of Geographical Information Science*, 37(2):403–437, Feb. 2023. doi: 10.1080/13658816.2022.2120996. URL https://hal.science/hal-03808724.

A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018. URL http://jmlr.org/papers/v18/17-468.html.

A. Bégué, D. Arvor, B. Bellon, J. Betbeder, D. de Abelleyra, R. P. D. Ferraz, V. Lebourgeois, C. Lelong, M. Simões, and S. R. Verón. Remote sensing and cropping practices: A review. *Remote Sensing*, 10(1):99, Jan. 2018. ISSN 2072-4292. doi: 10.3390/rs10010099. URL https://doi.org/10.3390/rs10010099.

V. Bellet, M. Fauvel, and J. Inglada. Land cover classification with gaussian processes using spatio-spectro-temporal features. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–21, 2023. doi: 10.1109/TGRS.2023.3234527.

V. Bellet, M. Fauvel, J. Inglada, and J. Michel. End-to-end learning for land cover classification using irregular and unaligned sits by combining attention-based interpolation with sparse variational gaussian processes. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 17:2980–2994, 2024. doi: 10.1109/JSTARS.2023.3343921.

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. doi: 10.1109/TPAMI.2013.50.

J. M. Bernardo and A. F. Smith. *Bayesian theory*, volume 405. John Wiley & Sons, 2009.

E. F. Berra, R. Gaulton, and S. Barr. Commercial off-the-shelf digital cameras on unmanned aerial vehicles for multitemporal monitoring of vegetation reflectance and NDVI. *IEEE transactions on geoscience and remote sensing*, 55(9):4878–4886, 2017.

C. P. Bing Lu and Y. He. Investigating different versions of prospect and prosail for estimating spectral and biophysical properties of photosynthetic and non-photosynthetic vegetation in mixed grasslands. *GIScience & Remote Sensing*, 58(3):354–371, 2021. doi: 10.1080/15481603.2021.1877435. URL https://doi.org/10.1080/15481603.2021.1877435.

E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman. Pyro: Deep universal probabilistic programming. *Journal of Machine Learning Research*, 20(28):1–6, 2019. URL http://jmlr.org/papers/v20/18-403.html.

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

S. Bojinski, M. Verstraete, T. C. Peterson, C. Richter, A. Simmons, and M. Zemp. The concept of essential climate variables in support of climate research, applications, and policy. *Bulletin of the American Meteorological Society*, 95(9):1431–1443, 2014.

K. A. Bollen. Latent variables in psychology and the social sciences. *Annual Review of Psychology*, 53(1):605–634, 2002. doi: 10.1146/annurev.psych.53.100901.135239. URL https://doi.org/10.1146/annurev.psych.53.100901.135239. PMID: 11752498.

J. Bouchat, E. Tronquo, A. Orban, N. E. C. Verhoest, and P. Defourny. Assessing the potential of fully polarimetric mono- and bistatic sar acquisitions in l-band for crop and soil monitoring. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:3168–3178, 2022. doi: 10.1109/JSTARS.2022.3162911.

J. Bouchat, E. Tronquo, A. Orban, K. A. C. de Macedo, N. E. C. Verhoest, and P. Defourny. The belsar dataset: Mono- and bistatic full-pol l-band sar for agriculture and hydrology, 2023.

M. A. Branch, T. F. Coleman, and Y. Li. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing*, 21:1–23, 1999. URL https://api.semanticscholar.org/CorpusID:8314598.

L. Brocca, S. Barbetta, S. Camici, L. Ciabatta, J. Dari, P. Filippucci, C. Massari, S. Modanesi, A. Tarpanelli, B. Bonaccorsi, et al. A digital twin of the terrestrial water cycle: a glimpse into the future through high-resolution earth observations. *Frontiers in Science*, 1:1190191, 2024.

N. H. Broge and E. Leblanc. Comparing prediction power and stability of broadband and hyperspectral vegetation indices for estimation of green leaf area index and canopy chlorophyll density. *Remote sensing of environment*, 76(2):156–172, 2001.

L. A. Brown, F. Camacho, V. García-Santos, N. Origo, B. Fuster, H. Morris, J. Pastor-Guzman, J. Sánchez-Zapero, R. Morrone, J. Ryder, J. Nightingale, V. Boccia, and J. Dash. Fiducial reference measurements for vegetation bio-geophysical variables: An end-to-end uncertainty evaluation framework. *Remote Sensing*, 13(16), 2021a. ISSN 2072-4292. doi: 10.3390/rs13163194. URL https://www.mdpi.com/2072-4292/13/16/3194.

L. A. Brown, R. Fernandes, N. Djamai, C. Meier, N. Gobron, H. Morris, F. Canisius, G. Bai, C. Lerebourg, C. Lanconelli, M. Clerici, and J. Dash. Validation of baseline and modified sentinel-2 level 2 prototype processor leaf area index retrievals over the united states. *ISPRS Journal of Photogrammetry and Remote Sensing*, 175:71–87, 2021b. ISSN 0924-2716. doi: \url{https://doi.org/10.1016/j.isprsjprs.2021.02.020}.

L. A. Brown, O. Williams, and J. Dash. Calibration and characterisation of four chlorophyll meters and transmittance spectroscopy for non-destructive estimation of forest leaf chlorophyll concentration. *Agricultural and Forest Meteorology*, 323:109059, 2022. ISSN 0168-1923. doi: https://doi.org/10.1016/j.agrformet.2022.109059. URL https://www.sciencedirect.com/science/article/pii/S0168192322002489.

H. H. Bulcock and G. P. W. Jewitt. Spatial mapping of leaf area index using hyperspectral remote sensing for hydrological applications with a particular focus on canopy interception. *Hydrology and Earth System Sciences*, 14(2):383–392, 2010. doi: 10.5194/hess-14-383-2010. URL https://hess.copernicus.org/articles/14/383/2010/.

L. Buonocore, J. Yates, and R. Valentini. A proposal for a forest digital twin framework and its perspectives. *Forests*, 13(4):498, 2022.

C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in β-vae. *CoRR*, abs/1804.03599, 2018. URL http://arxiv.org/abs/1804.03599.

M. U. Caglar, A. I. Teufel, and C. O. Wilke. Sicegar: R package for sigmoidal and double-sigmoidal curve fitting. *PeerJ*, 6:e4251, 2018.

S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis. Physics-informed neural networks (pinns) for fluid mechanics: A review, 2021a.

S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6):060801, 2021b.

C. Callender and J. Cohen. There is no special problem about scientific representation. *Theoria*, 55:67–85, 2006.

G. Campbell. Extinction coefficients for radiation in plant canopies calculated using an ellipsoidal inclination angle distribution. *Agricultural and Forest Meteorology*, 36(4):317–321, 1986. ISSN 0168-1923. doi: https://doi.org/10.1016/0168-1923(86)90010-9.

G. Campbell. Derivation of an angle density function for canopies with ellipsoidal leaf angle distributions. *Agricultural and forest meteorology*, 49(3):173–176, 1990.

G. Camps-Valls and L. Bruzzone. Kernel-based methods for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 43(6):1351–1362, 2005. doi: 10.1109/TGRS.2005.846154.

G. Camps-Valls, D. H. Svendsen, J. Cortés-Andrés, Á. Mareno-Martínez, A. Pérez-Suay, J. Adsuara, I. Martín, M. Piles, J. Muñoz-Marí, and L. Martino. Physics-aware machine learning for geosciences and remote sensing. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 2086–2089. IEEE, 2021.

F. P. Casale, A. Dalca, L. Saglietti, J. Listgarten, and N. Fusi. Gaussian process prior variational autoencoders. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/1c336b8080f82bcc2cd2499b4c57261d-Paper.pdf.

S. Chandrasekhar. *Radiative transfer*. Dover, 1960.

J. Chen, C. Menges, and S. Leblanc. Global mapping of foliage clumping index using multi-angular satellite data. *Remote Sensing of Environment*, 97(4):447–457, 2005. ISSN 0034-4257. doi: \url{https://doi.org/10.1016/j.rse.2005.05.003}.

J. M. Chen and J. Cihlar. Plant canopy gap-size analysis theory for improving optical measurements of leaf-area index. *Applied Optics*, 34(27):6211–6222, Sept. 1995. doi: 10.1364/AO.34.006211. URL https://opg.optica.org/ao/abstract.cfm?URI=ao-34-27-6211.

S. Chen, Y. H. Fu, F. Hao, X. Li, S. Zhou, C. Liu, and J. Tang. Vegetation phenology and its ecohydrological implications from individual to global scales. *Geography and Sustainability*, 3(4):334–338, 2022. ISSN 2666-6839. doi: https://doi.org/10.1016/j.geosus.2022.10.002.

T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *International Conference on Learning Representations*, 2018a. URL https://openreview.net/forum?id=BJdMRoCIf.

X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. Variational lossy autoencoder. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=BysvGP5ee.

Y. Chen, Z. Zhang, and F. Tao. Improving regional winter wheat yield estimation through assimilation of phenology and leaf area index from remote sensing data. *European Journal of Agronomy*, 101:163–173, 2018b. ISSN 1161-0301. doi: \url{https://doi.org/10.1016/j.eja.2018.09.006}.

D. Chicco, M. J. Warrens, and G. Jurman. The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *Peerj computer science*, 7:e623, 2021.

U. J. Choi and K. S. Rim. Variational non-bayesian inference of the probability density function in the wiener algebra, 2023.

T. F. Coleman and Y. Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6(2):418–445, 1996.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967. doi: 10.1109/TIT.1967.1053964.

C. T. de Wit. Photosynthesis of leaf canopies. Technical report, Pudoc, 1965.

M. Dean. Lecture notes for fall 2022 phd class in behavioral economics: Order theory, 2022.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

R. F. Denison. Minimizing errors in lai estimates from laser-probe inclined-point quadrats. *Field Crops Research*, 51(3):231–240, 1997. ISSN 0378-4290. doi: \url{https://doi.org/10.1016/S0378-4290(96)03460-0}.

L. M. Domenzain, J. Gómez-Dans, and P. P. Lewis. jgomezdans/prosail: Pip package bug fix release, Feb. 2019. URL https://doi.org/10.5281/zenodo.2574925.

G. Dorta, S. Vicente, L. Agapito, N. D. Campbell, and I. Simpson. Structured uncertainty prediction networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5477–5485, 2018.

R. Duan, H. Wu, and R. Zhou. Faster matrix multiplication via asymmetric hashing. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2129–2138, Los Alamitos, CA, USA, Nov. 2023. IEEE Computer Society. doi: 10.1109/FOCS57990.2023.00130. URL https://doi.ieeecomputersociety.org/10.1109/FOCS57990.2023.00130.

I. Dumeur, S. Valero, and J. Inglada. Self-supervised spatio-temporal representation learning of satellite image time series. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 17:4350–4367, 2024. doi: 10.1109/JSTARS.2024.3358066.

G. Duveiller, M. Weiss, F. Baret, and P. Defourny. Retrieving wheat green area index during the growing season from optical time series measurements based on neural network radiative transfer inversion. *Remote Sensing of Environment*, 115(3):887–896, 2011. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2010.11.016.

C. Eastwood and C. K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=By-7dz-AZ.

L. E.B. and M. E.A. The point method for pasture analysis. *New Zealand Journal of Agriculture*, 46:267–279, 1933.

V. Edupuganti, M. Mardani, S. Vasanawala, and J. Pauly. Uncertainty quantification in deep mri reconstruction. *IEEE Transactions on Medical Imaging*, 40(1):239–250, 2021. doi: 10.1109/TMI.2020.3025065.

J. Epting, D. Verbyla, and B. Sorbel. Evaluation of remotely sensed indices for assessing burn severity in interior alaska using landsat tm and etm+. *Remote Sensing of Environment*, 96(3):328–339, 2005. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2005.03.002.

B. Esmaeili, H. Wu, S. Jain, A. Bozkurt, N. Siddharth, B. Paige, D. H. Brooks, J. Dy, and J.-W. van de Meent. Structured disentangled representations. In K. Chaudhuri and M. Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 2525–2534. PMLR, Apr. 2019. URL https://proceedings.mlr.press/v89/esmaeili19a.html.

H. Fang, F. Baret, S. Plummer, and G. Schaepman-Strub. An overview of global leaf area index (lai): Methods, products, validation, and applications. *Reviews of Geophysics*, 57 (3):739–799, 2019. doi: \url{https://doi.org/10.1029/2018RG000608}. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018RG000608.

K. S. Fassnacht, S. T. Gower, J. M. Norman, and R. E. McMurtric. A comparison of optical and direct methods for estimating foliage surface area index in forests. *Agricultural and Forest Meteorology*, 71(1-2):183–207, 1994.

J.-B. Feret, C. François, G. P. Asner, A. A. Gitelson, R. E. Martin, L. P. Bidel, S. L. Ustin, G. le Maire, and S. Jacquemoud. Prospect-4 and 5: Advances in the leaf optical properties model separating photosynthetic pigments. *Remote Sensing of Environment*, 112(6):3030–3043, 2008. ISSN 0034-4257. doi: \url{https://doi.org/10.1016/j.rse.2008.02.012}.

L. Finkelstein and K. Grattan. *Concise Encyclopedia of Measurement & Instrumentation*. Advances in systems, control, and information engineering. Pergamon Press, 1994. ISBN 9780080362120. URL https://books.google.fr/books?id=7QIpAQAAMAAJ.

R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pages 3429–3437, 2017.

E. I. Fried. Theories and models: What they are, what they are for, and what they are about. *Psychological Inquiry*, 31(4):336–344, 2020. doi: 10.1080/1047840X.2020.1854011. URL https://doi.org/10.1080/1047840X.2020.1854011.

R. Frigg and J. Nguyen. Scientific Representation. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2021 edition, 2021.

J.-B. Féret. *Apport de la modélisation pour l'estimation de la teneur en pigments foliaires par télédétection*. PhD thesis, Institut de physique du Globe de Paris, 2009. URL http://www.theses.fr/2009PA066419. Thèse de doctorat dirigée par Jacquemoud, Stéphane et François, Christophe Mesures physiques en télédétection Paris 6 2009.

J.-B. Féret, A. Gitelson, S. Noble, and S. Jacquemoud. Prospect-d: Towards modeling leaf optical properties through a complete lifecycle. *Remote Sensing of Environment*, 193: 204–215, 2017. ISSN 0034-4257. doi: \url{https://doi.org/10.1016/j.rse.2017.03.004}.

J.-B. Féret, G. le Maire, S. Jay, D. Berveiller, R. Bendoula, G. Hmimina, A. Cheraiet, J. Oliveira, F. Ponzoni, T. Solanki, F. de Boissieu, J. Chave, Y. Nouvellon, A. Porcar-Castell, C. Proisy, K. Soudani, J.-P. Gastellu-Etchegorry, and M.-J. Lefèvre-Fonollosa. Estimating leaf mass per area and equivalent water thickness based on leaf optical properties: Potential and limitations of physical modeling and machine learning. *Remote Sensing of Environment*, 231:110959, 2019. ISSN 0034-4257. doi: \url{https://doi.org/10.1016/j.rse.2018.11.002}.

J.-B. Féret, K. Berger, F. de Boissieu, and Z. Malenovský. Prospect-pro for estimating content of nitrogen-containing leaf proteins and other carbon-based constituents. *Remote Sensing of Environment*, 252:112173, 2021. ISSN 0034-4257. doi: \url{https://doi.org/10.1016/j.rse.2020.112173}.

X. Gao, J. M. Gray, and B. J. Reich. Long-term, medium spatial resolution annual land surface phenology with a bayesian hierarchical model. *Remote Sensing of Environment*, 261:112484, 2021.

W. Gautschi and W. F. Cahill. *Handbook of mathematical functions*. Dover Publications, Inc, New York, 1964.

Y. Ge, X. Zhang, P. M. Atkinson, A. Stein, and L. Li. Geoscience-aware deep learning: A new paradigm for remote sensing. *Science of Remote Sensing*, 5:100047, 2022. ISSN 2666-0172. doi: https://doi.org/10.1016/j.srs.2022.100047.

S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, 1984. doi: 10.1109/TPAMI.1984.4767596.

E. I. George, U. E. Makov, and A. F. M. Smith. Conjugate likelihood distributions. *Scandinavian Journal of Statistics*, 20(2):147–156, 1993. ISSN 03036898, 14679469. URL http://www.jstor.org/stable/4616270.

S. Gershman and N. Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the annual meeting of the cognitive science society*, volume 36, 2014.

L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89, 2018. doi: 10.1109/DSAA.2018.00018.

A. Gitelson and M. N. Merzlyak. Spectral reflectance changes associated with autumn senescence of aesculus hippocastanum l. and acer platanoides l. leaves. spectral features and relation to chlorophyll estimation. *Journal of Plant Physiology*, 143(3):286–292, 1994. ISSN 0176-1617. doi: https://doi.org/10.1016/S0176-1617(11)81633-0.

A. A. Gitelson, Y. J. Kaufman, and M. N. Merzlyak. Use of a green channel in remote sensing of global vegetation from eos-modis. *Remote Sensing of Environment*, 58(3):289–298, 1996. ISSN 0034-4257. doi: https://doi.org/10.1016/S0034-4257(96)00072-7.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

L. A. Goodman. On the exact variance of products. *Journal of the American Statistical Association*, 55:708–713, 1960. URL https://api.semanticscholar.org/CorpusID:120014324.

C. W. J. Granger. Can we improve the perceived quality of economic forecasts? *Journal of Applied Econometrics*, 11(5):455–473, 1996. ISSN 08837252, 10991255. URL http://www.jstor.org/stable/2285211.

K. Green. Selecting and interpreting high-resolution images. *Journal of Forestry*, 2000. URL https://api.semanticscholar.org/CorpusID:85671598.

A. Gudi. Recognizing semantic features in faces using deep learning, 2016.

L. Gueguen, J. Koenig, C. Reeder, T. Barksdale, J. Saints, K. Stamatiou, J. Collins, and C. Johnston. Mapping human settlements and population at country scale from vhr images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10 (2):524–538, 2017. doi: 10.1109/jstars.2016.2616120. URL http://dx.doi.org/10.1109/JSTARS.2016.2616120.

I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville. PixelVAE: A latent variable model for natural images. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=BJKYvt5lg.

O. Hagolle, M. Huc, C. Desjardins, and S. Auerand Rudolf Richter. Maja algorithm theoretical basis document, Dec. 2017. URL https://doi.org/10.5281/zenodo.1209633.

D. K. Hall and G. A. Riggs. Normalized-difference snow index (ndsi). *Encyclopedia of snow, ice and glaciers*, 2010.

M. Hall-Beyer. Comparison of single-year and multiyear NDVI time series principal components in cold temperate biomes. *IEEE Transactions on Geoscience and Remote Sensing*, 41(11):2568–2574, 2003. doi: 10.1109/TGRS.2003.817274.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, Apr. 1970. ISSN 0006-3444. doi: 10.1093/biomet/57.1.97. URL https://doi.org/10.1093/biomet/57.1.97.

L. T. Hauser, J.-B. Féret, N. An Binh, N. van der Windt, Ângelo F. Sil, J. Timmermans, N. A. Soudzilovskaia, and P. M. van Bodegom. Towards scalable estimation of plant functional diversity from sentinel-2: In-situ validation in a heterogeneous (semi-)natural landscape. *Remote Sensing of Environment*, 262:112505, 2021. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2021.112505.

K. Hayes, J. Dambacher, G. Hosack, N. Bax, P. Dunstan, E. Fulton, P. Thompson, J. Hartog, A. Hobday, R. Bradford, et al. Identifying indicators and essential variables for marine ecosystems. *Ecological Indicators*, 57:409–419, 2015.

C. He, J. Sun, Y. Chen, L. Wang, S. Shi, F. Qiu, S. Wang, and T. Tagesson. A new vegetation index combination for leaf carotenoid-to-chlorophyll ratio: minimizing the effect of their correlation. *International Journal of Digital Earth*, 16(1):272–288, 2023. doi: 10.1080/17538947.2023.2168772. URL https://doi.org/10.1080/17538947.2023.2168772.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015a.

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015b.

V. Henrich, C. Götze, A. Jung, C. Sandow, D. Thürkow, and G. Cornelia. Development of an online indices database: Motivation, concept and implementation. In *6th EARSeL Imaging Spectroscopy SIG Workshop "Innovative Tool for Scientific and Commercial Environment Applications", Tel Aviv, Israel*, Mar. 2009.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner. Towards a definition of disentangled representations, 2018.

T. K. Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1, 1995. doi: 10.1109/ICDAR.1995.598994.

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. ISSN 00401706. URL http://www.jstor.org/stable/1267351.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(40):1303–1347, 2013. URL http://jmlr.org/papers/v14/hoffman13a.html.

M. D. Homan and A. Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, Jan. 2014. ISSN 1532-4435.

B. Hosgood, S. Jacquemound, G. Andreeoli, J. Verdebout, A. Pedrini, and G. Schmuck. Leaf optical properties experiment database (lopex93), 1993. URL http://ecosis.org.

A. Huete. A soil-adjusted vegetation index (savi). *Remote Sensing of Environment*, 25(3):295–309, 1988. ISSN 0034-4257. doi: https://doi.org/10.1016/0034-4257(88)90106-X.

A. Huete, K. Didan, T. Miura, E. Rodriguez, X. Gao, and L. Ferreira. Overview of the radiometric and biophysical performance of the modis vegetation indices. *Remote Sensing of Environment*, 83(1):195–213, 2002. ISSN 0034-4257. doi: https://doi.org/10.1016/S0034-4257(02)00096-2. The Moderate Resolution Imaging Spectroradiometer (MODIS): a new generation of Land Surface Monitoring.

R. I. G. Hughes. Models and representation. *Philosophy of Science*, 64:S325–S336, 1997. ISSN 00318248, 1539767X. URL http://www.jstor.org/stable/188414.

D. Ienco, R. Interdonato, R. Gaetano, and D. Ho Tong Minh. Combining sentinel-1 and sentinel-2 satellite image time series for land cover mapping via a multi-source deep learning architecture. *ISPRS Journal of Photogrammetry and Remote Sensing*, 158:11–22, 2019. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2019.09.016. URL https://www.sciencedirect.com/science/article/pii/S0924271619302278.

J. Inglada. Sentinel-2 Agriculture Vegetation status DPM. Research report, CESBIO, June 2017. URL https://hal.science/hal-02874654.

J. Inglada, A. Vincent, M. Arias, B. Tardy, D. Morin, and I. Rodes. Operational High Resolution Land Cover Map Production at the Country Scale Using Satellite Image Time Series. *Remote Sensing*, 9(1), 2017a. ISSN 2072-4292. doi: 10.3390/rs9010095. URL https://www.mdpi.com/2072-4292/9/1/95.

J. Inglada, A. Vincent, M. Arias, B. Tardy, D. Morin, and I. Rodes. Operational high resolution land cover map production at the country scale using satellite image time series. *Remote Sensing*, 9(1):95, 2017b. doi: 10.3390/rs9010095. URL http://dx.doi.org/10.3390/rs9010095.

T. S. Jaakkola. *Variational methods for inference and estimation in graphical models*. PhD thesis, Massachusetts Institute of Technology, 1997.

S. Jacquemoud and F. Baret. Prospect: A model of leaf optical properties spectra. *Remote Sensing of Environment*, 34(2):75–91, 1990. ISSN 0034-4257. doi: \url{https://doi.org/10.1016/0034-4257(90)90100-Z}.

S. Jacquemoud and S. L. Ustin. Modeling leaf optical properties. *Photobiological Sciences Online*, 2008.

S. Jacquemoud, C. Bacour, H. Poilvé, and J.-P. Frangi. Comparison of four radiative transfer models to simulate plant canopies reflectance: Direct and inverse mode. *Remote Sensing of Environment*, 74(3):471–481, 2000. ISSN 0034-4257. doi: \url{https://doi.org/10.1016/S0034-4257(00)00139-5}.

A. Javed, Q. Cheng, H. Peng, O. Altan, Y. Li, I. Ara, E. Huq, Y. Ali, and N. Saleem. Review of spectral indices for urban remote sensing. *Photogrammetric Engineering and Remote Sensing*, 87(7):513–524, 2021.

Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. In *International Joint Conferences on Artificial Intelligence*, 2016.

I. Jonckheere, S. Fleck, K. Nackaerts, B. Muys, P. Coppin, M. Weiss, and F. Baret. Methods for leaf area index determination. part i: Theories, techniques and instruments. *Agricultural and Forest Meteorology*, 121:19–35, 2004.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Learning in graphical models*, pages 105–161, 1998.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

H. Kim and A. Mnih. Disentangling by factorising. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2649–2658. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kim18b.html.

D. Kingma. Variational inference & deep learning: A new synthesis, 2017.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representation (ICLR) 2015*, 2015.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

D. P. Kingma and M. Welling. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8237. doi: 10.1561/2200000056. URL http://dx.doi.org/10.1561/2200000056.

V. Klemas and R. Smart. The influence of soil salinity, growth form, and leaf moisture on-the spectral radiance of. *Photogrammetric Engineering and Remote Sensing*, 49:77–83, 1983.

T. Koike-Akino and Y. Wang. Autovae: Mismatched variational autoencoder with irregular posterior-prior pairing. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 1689–1694, 2022. doi: 10.1109/ISIT50566.2022.9834769.

C. J. Kucharik, J. M. Norman, and S. T. Gower. Measurements of branch area and adjusting leaf area index indirect measurements. *Agricultural and Forest Meteorology*, 91(1):69–88, 1998. ISSN 0168-1923. doi: \url{https://doi.org/10.1016/S0168-1923(98)00064-1}.

A. Kumar, P. Sattigeri, and A. Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H1kG7GZAW.

S. Kurz, H. De Gersem, A. Galetzka, A. Klaedtke, M. Liebsch, D. Loukrezis, S. Russenschuck, and M. Schmidt. Hybrid modeling: towards the next level of scientific computing in engineering. *Journal of Mathematics in Industry*, 12(1):8, 2022. doi: \url{https://doi.org/10.1186/s13362-022-00123-0}.

A. Kuusk. The hot spot effect of a uniform vegetative cover. *Soviet Journal of Remote Sensing*, 3(4):645–658, 1985.

J. Landon and N. Singpurwalla. Choosing a coverage probability for prediction intervals. *The American Statistician*, 62:120–124, Feb. 2008. doi: 10.1198/000313008X304062.

G. Latombe, P. Pyšek, J. M. Jeschke, T. M. Blackburn, S. Bacher, C. Capinha, M. J. Costello, M. Fernández, R. D. Gregory, D. Hobern, et al. A vision for global monitoring of biological invasions. *Biological Conservation*, 213:295–308, 2017.

G. le Maire, C. François, K. Soudani, D. Berveiller, J.-Y. Pontailler, N. Bréda, H. Genet, H. Davi, and E. Dufrêne. Calibration and validation of hyperspectral indices for the estimation of broadleaved forest leaf chlorophyll content, leaf mass per area, leaf area index and leaf canopy biomass. *Remote Sensing of Environment*, 112(10):3846–3864, 2008. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2008.06.005. URL https://www.sciencedirect.com/science/article/pii/S003442570800196X.

E. L. Lehmann and G. Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.

W. Li, M. Weiss, S. Jay, S. Wei, N. Zhao, A. Comar, R. Lopez-Lozano, B. De Solan, Q. Yu, W. Wu, and F. Baret. Daily monitoring of effective green area index and vegetation chlorophyll content from continuous acquisitions of a multi-band spectrometer over winter wheat. *Remote Sensing of Environment*, 300:113883, 2024. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2023.113883. URL https://www.sciencedirect.com/science/article/pii/S0034425723004340.

Y. Li, N. He, J. Hou, L. Xu, C. Liu, J. Zhang, Q. Wang, X. Zhang, and X. Wu. Factors influencing leaf chlorophyll content in natural forests at the biome scale. *Frontiers in Ecology and Evolution*, 6, 2018. ISSN 2296-701X. doi: 10.3389/fevo.2018.00064. URL https://www.frontiersin.org/articles/10.3389/fevo.2018.00064.

H. K. Lichtenthaler. Chlorophylls and carotenoids: Pigments of photosynthetic biomembranes. In *Plant Cell Membranes*, volume 148 of *Methods in Enzymology*, pages 350–382. Academic Press, 1987. doi: https://doi.org/10.1016/0076-6879(87)48036-1.

J.-H. Lin and J. S. Vitter. Approximation algorithms for geometric median problems. *Information Processing Letters*, 44(5):245–249, 1992. ISSN 0020-0190. doi: https://doi.org/10.1016/0020-0190(92)90208-D.

Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

D. Loaiza, C. Aybar, M. Mahecha, F. Martinuzzi, M. Söchting, and S. Wieneke. A standardized catalogue of spectral indices to advance the use of remote sensing in earth system research. *Scientific Data*, 10, Apr. 2023. doi: 10.1038/s41597-023-02096-0.

F. Locatello, S. Bauer, M. Lucic, S. Gelly, B. Scholkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:54089884.

J. Louis. Level-2a algorithm theoretical basis document. Technical report, ESA, Nov. 2021. URL https://step.esa.int/docs/extra/ATBD_S2ToolBox_L2B_V1.1.pdf.

S. Ma, Y. Zhou, P. H. Gowda, J. Dong, G. Zhang, V. G. Kakani, P. Wagle, L. Chen, K. C. Flynn, and W. Jiang. Application of the water-related spectral reflectance indices: A review. *Ecological indicators*, 98:68–79, 2019.

A. Makhzani and B. J. Frey. Pixelgan autoencoders. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/7e7e69ea3384874304911625ac34321c-Paper.pdf.

D. Marcos, R. Fong, S. Lobry, R. Flamary, N. Courty, and D. Tuia. Contextual semantic interpretability. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

J. Markwell, J. C. Osterman, and J. L. Mitchell. Calibration of the minolta spad-502 leaf chlorophyll meter. *Photosynthesis research*, 46:467–472, 1995.

G. Marsaglia and W. W. Tsang. A simple method for generating gamma variables. *ACM Transactions on Mathematical Software (TOMS)*, 26(3):363–372, 2000.

S. N. Martens, S. L. Ustin, and R. A. Rousseau. Estimation of tree canopy leaf area index by gap fraction analysis. *Forest Ecology and Management*, 61(1):91–108, 1993. ISSN 0378-1127. doi: https://doi.org/10.1016/0378-1127(93)90192-P.

E. Mathieu, T. Rainforth, N. Siddharth, and Y. W. Teh. Disentangling disentanglement in variational autoencoders. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4402–4412. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/mathieu19a.html.

K. Maxwell and G. N. Johnson. Chlorophyll fluorescence—a practical guide. *Journal of Experimental Botany*, 51(345):659–668, Apr. 2000. ISSN 0022-0957. doi: 10.1093/jexbot/51.345.659. URL https://doi.org/10.1093/jexbot/51.345.659.

S. Mazumdar, S. Wrigley, and F. Ciravegna. Citizen science and crowdsourcing for earth observations: An analysis of stakeholder opinions on the present and future. *Remote Sensing*, 9(1), 2017. ISSN 2072-4292. doi: 10.3390/rs9010087. URL https://www.mdpi.com/2072-4292/9/1/87.

W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

N. McDowell, H. Barnard, B. Bond, T. Hinckley, R. Hubbard, H. Ishii, B. Köstner, F. Magnani, J. Marshall, F. Meinzer, et al. The relationship between tree height and leaf area: sapwood area ratio. *Oecologia*, 132:12–20, 2002.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21 (6):1087–1092, Dec. 1957. ISSN 0021-9606. doi: 10.1063/1.1699114. URL https://doi.org/10.1063/1.1699114.

N. Metzger, M. O. Turkoglu, S. D'Aronco, J. D. Wegner, and K. Schindler. Crop classification under varying cloud cover with neural ordinary differential equations. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–12, 2021.

N. Miao, E. Mathieu, S. N, Y. W. Teh, and T. Rainforth. On incorporating inductive biases into VAEs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nzvbBD_3J-g.

L. Miller, C. Pelletier, and G. I. Webb. Deep learning for satellite image time-series analysis: A review. *IEEE Geoscience and Remote Sensing Magazine*, 2024.

M. Monsi. Uber den lichtfaktor in den pflanzengesellschaften und seine bedeutung fur die stoffproduktion. *Japanese Journal of Botany*, 14:22–52, 1953.

D. Montero, C. Aybar, M. D. Mahecha, F. Martinuzzi, M. Söchting, and S. Wieneke. A standardized catalogue of spectral indices to advance the use of remote sensing in earth system research. *Scientific Data*, 10(1):197, 2023.

P. Moreau, V. Viaud, V. Parnaudeau, J. Salmon-Monviola, and P. Durand. An approach for global sensitivity analysis of a complex environmental model to spatial inputs and parameters: A case study of an agro-hydrological model. *Environmental modelling & software*, 47:74–87, 2013.

S. Nativi, P. Mazzetti, and M. Craglia. Digital ecosystems for developing digital twins of the earth: The destination earth case. *Remote Sensing*, 13(11), 2021. ISSN 2072-4292. doi: 10.3390/rs13112119. URL https://www.mdpi.com/2072-4292/13/11/2119.

R. M. Neal and G. E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. MIT Press, 1998.

C. T. Nguyen, A. Chidthaisong, P. Kieu Diem, and L.-Z. Huo. A modified bare soil index to identify bare land features during agricultural fallow-period in southeast asia using landsat 8. *Land*, 10(3):231, 2021.

A. Orban, D. Defrere, and C. Barbier. Belsar : the first belgian airborne campaign for l-band, full polarimetric bistatic and interferometric sar acquisitions over an agricultural site in belgium. In *EUSAR 2021; 13th European Conference on Synthetic Aperture Radar*, pages 1–4, 2021.

N. Origo, J. Gorroño, J. Ryder, J. Nightingale, and A. Bialek. Fiducial reference measurements for validation of sentinel-2 and proba-v surface reflectance products. *Remote Sensing of Environment*, 241:111690, 2020. ISSN 0034-4257. doi: \url{https://doi.org/10.1016/j.rse.2020.111690}.

J. Paisley, D. Blei, and M. Jordan. Variational bayesian inference with stochastic search. In *In Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1367–1374, 2012.

C. Pelletier, S. Valero, J. Inglada, N. Champion, and G. Dedieu. Assessing the robustness of random forests to map land cover with high resolution satellite image time series over large areas. *Remote Sensing of Environment*, 187:156–168, 2016a. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2016.10.010.

C. Pelletier, S. Valero, J. Inglada, N. Champion, and G. Dedieu. Assessing the robustness of random forests to map land cover with high resolution satellite image time series over large areas. *Remote Sensing of Environment*, 187:156–168, 2016b. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2016.10.010.

H. M. Pereira, S. Ferrier, M. Walters, G. N. Geller, R. H. Jongman, R. J. Scholes, M. W. Bruford, N. Brummitt, S. H. Butchart, A. Cardoso, et al. Essential biodiversity variables. *Science*, 339(6117):277–278, 2013.

C. Persello, J. D. Wegner, R. Hänsch, D. Tuia, P. Ghamisi, M. Koeva, and G. Camps-Valls. Deep learning and earth observation to support the sustainable development goals: Current approaches, open challenges, and future opportunities. *IEEE Geoscience and Remote Sensing Magazine*, 10(2):172–200, 2022.

D. Phan, N. Pradhan, and M. Jankowiak. Composable effects for flexible and accelerated probabilistic programming in numpyro. In *Program Transformations for ML Workshop at NeurIPS 2019*, 2019. URL https://openreview.net/forum?id=H1g1niFhIB.

D. Phiri, M. Simwanda, S. Salekin, V. R. Nyirenda, Y. Murayama, and M. Ranagalage. Sentinel-2 data for land cover/use mapping: A review. *Remote Sensing*, 12(14):2291, 2020.

D. Picard. Torch.manual_seed(3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision. *CoRR*, abs/2109.08203, 2021. URL https://arxiv.org/abs/2109.08203.

L. Prévot, I. Champion, and G. Guyot. Estimating surface soil moisture and leaf area index of a wheat canopy using a dual-frequency (c and x bands) scatterometer. *Remote Sensing of Environment*, 46(3):331–339, Dec. 1993. ISSN 0034-4257. doi: 10.1016/0034-4257(93)90053-z. URL http://dx.doi.org/10.1016/0034-4257(93)90053-Z.

C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan, and A. Edelman. Universal differential equations for scientific machine learning, 2021.

M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations, 2017a.

M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations, 2017b.

M. Raissi, P. Perdikaris, and G. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991. doi: \url{https://doi.org/10.1016/j.jcp.2018.10.045}.

A. Razavi, A. van den Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Paper.pdf.

B. Reyers, M. Stafford-Smith, K.-H. Erb, R. J. Scholes, and O. Selomane. Essential variables help to focus sustainable development goals monitoring. *Current Opinion in Environmental Sustainability*, 26:97–105, 2017.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32(2) of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/rezende14.html.

H. Robbins and S. Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951. doi: 10.1214/aoms/1177729586. URL https://doi.org/10.1214/aoms/1177729586.

J. Rogan and D. Chen. Remote sensing technology for mapping and monitoring land-cover and land-use change. *Progress in Planning*, 61(4):301–325, 2004. ISSN 0305-9006. doi: https://doi.org/10.1016/S0305-9006(03)00066-7. URL https://www.sciencedirect.com/science/article/pii/S0305900603000667.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

R. Rossi, M. Gelfusa, A. Murari, and on behalf of JET contributors. On the potential of physics-informed neural networks to solve inverse problems in tokamaks. *Nuclear Fusion*, 63(12):126059, Nov. 2023. doi: 10.1088/1741-4326/ad067c. URL https://dx.doi.org/10.1088/1741-4326/ad067c.

B. Rouquié, O. Hagolle, F.-M. Bréon, O. Boucher, C. Desjardins, and S. Rémy. Using copernicus atmosphere monitoring service products to constrain the aerosol type in the atmospheric correction processor maja. *Remote Sensing*, 9(12):1230, 2017.

J. Rouse, T. A. . M. U. R. S. Center, and G. S. F. Center. *Monitoring the Vernal Advancement and Retrogradation (greenwave Effect) of Natural Vegetation*. Texas A & M University, Remote Sensing Center, 1974. URL https://ntrs.nasa.gov/api/citations/19750020419/downloads/19750020419.pdf.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

B. Russell. *Part IV: Order*, chapter 25. Cambridge University Press, Cambridge, England, 1903.

O. Rybkin, K. Daniilidis, and S. Levine. Simple and effective vae training with calibrated decoders, 2021.

Y. Ryu, T. Nilson, H. Kobayashi, O. Sonnentag, B. E. Law, and D. D. Baldocchi. On the correct estimation of effective leaf area index: Does it reveal information on clumping effects? *Agricultural and Forest Meteorology*, 150(3):463–472, 2010.

L. Saul and M. Jordan. Exploiting tractable substructures in intractable networks. *Advances in neural information processing systems*, 8, 1995.

G. Schaepman-Strub, M. Schaepman, T. Painter, S. Dangel, and J. Martonchik. Reflectance quantities in optical remote sensing—definitions and case studies. *Remote Sensing of Environment*, 103(1):27–42, 2006. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2006.03.002.

F. Schiefer, S. Schmidtlein, and T. Kattenborn. The retrieval of plant functional traits from canopy spectra through rtm-inversions and statistical models are both critically affected by plant phenology. *Ecological Indicators*, 121:107062, 2021. ISSN 1470-160X. doi: https://doi.org/10.1016/j.ecolind.2020.107062.

F. Schrodt, J. J. Bailey, W. D. Kissling, K. F. Rijsdijk, A. C. Seijmonsbergen, D. Van Ree, J. Hjort, R. S. Lawley, C. N. Williams, M. G. Anderson, et al. To advance sustainable stewardship, we must document not only biodiversity but geodiversity. *Proceedings of the National Academy of Sciences*, 116(33):16155–16158, 2019.

A. Sepliarskaia, J. Kiseleva, and M. de Rijke. How to not measure disentanglement, 2021.

M. Q. Shahbaz, M. Ahsanullah, S. H. Shahbaz, and B. M. Al-Zahrani. *Ordered random variables: Theory and applications*. Atlantis Press Paris, 2016. doi: https://doi.org/10.2991/978-94-6239-225-0.

Y. She, C. Atzberger, A. Blake, and S. Keshav. From spectra to biophysical insights: End-to-end learning with a biased radiative transfer model, 2024.

R. Sheikholeslami, S. Razavi, H. V. Gupta, W. Becker, and A. Haghnegahdar. Global sensitivity analysis for high-dimensional problems: How to objectively group factors and measure robustness and convergence while reducing computational cost. *Environmental modelling & software*, 111:282–299, 2019.

X. Shen, B. Liu, Y. Zhou, J. Zhao, and M. Liu. Remote sensing image captioning via variational autoencoder and reinforcement learning. *Knowledge-Based Systems*, 203:105920, 2020. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2020.105920.

S. N. Shukla and B. Marlin. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=4c0J6lwQ4_.

A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14:199–222, 2004.

A. Srivastava and C. Sutton. Autoencoding variational inference for topic models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=BybtVK9lg.

A. Stoian, V. Poulain, J. Inglada, V. Poughon, and D. Derksen. Land cover maps production with high resolution satellite image time series and convolutional neural networks: Adaptations and limits for operational systems. *Remote Sensing*, 11(17), 2019. ISSN 2072-4292. doi: 10.3390/rs11171986. URL https://www.mdpi.com/2072-4292/11/17/1986.

G. G. Stokes. Iv. on the intensity of the light reflected from or transmitted through a pile of plates. *Proceedings of the Royal Society of London*, 11:545–556, 1862.

V. Strassen et al. Gaussian elimination is not optimal. *Numerische mathematik*, 13(4):354–356, 1969.

G. H. Suits. The calculation of the directional reflectance of a vegetative canopy. *Remote Sensing of Environment*, 2:117–125, 1971. ISSN 0034-4257. doi: https://doi.org/10.1016/0034-4257(71)90085-X.

D. H. Svendsen, D. Hernández-Lobato, L. Martino, V. Laparra, Á. Moreno-Martínez, and G. Camps-Valls. Inference over radiative transfer models using variational and expectation maximization methods. *Machine Learning*, 112(3):921–937, June 2021. ISSN 1573-0565. doi: 10.1007/s10994-021-05999-4. URL http://dx.doi.org/10.1007/s10994-021-05999-4.

T. Szandała. Review and comparison of commonly used activation functions for deep neural networks. *Bio-inspired neurocomputing*, pages 203–224, 2021.

N. Takeishi and A. Kalousis. Variational autoencoder with differentiable physics engine for human gait analysis and synthesis. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021a. URL https://openreview.net/forum?id=9ISlKio3Bt.

N. Takeishi and A. Kalousis. Physics-integrated variational autoencoders for robust and interpretable generative modeling. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021b. URL https://openreview.net/forum?id=0p0gt1Pn2Gv.

J. Thomas and H. Gausman. Leaf reflectance vs. leaf chlorophyll and carotenoid concentrations for eight crops 1. *Agronomy journal*, 69(5):799–802, 1977.

A. N. Tikhonov and V. Y. Arsenin. *Solutions of ill-posed problems*. V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York, 1977.

M. Toda and A. D. Richardson. Estimation of plant area index and phenological transition dates from digital repeat photography and radiometric approaches in a hardwood forest in the northeastern united states. *Agricultural and Forest Meteorology*, 249:457–466, 2018. ISSN 0168-1923. doi: \url{https://doi.org/10.1016/j.agrformet.2017.09.004}.

M. Tolomio and R. Casa. Dynamic crop models and remote sensing irrigation decision support systems: a review of water stress concepts for improved estimation of water requirements. *Remote Sensing*, 12(23):3945, Dec. 2020. ISSN 2072-4292. doi: 10.3390/rs12233945. URL https://doi.org/10.3390/rs12233945.

M. Tschannen, O. Bachem, and M. Lucic. Recent advances in autoencoder-based representation learning. In *Third workshop on Bayesian Deep Learning (NeurIPS 2018)*, 2018. URL http://www.nari.ee.ethz.ch/pubs/p/autoenc2018.

C. J. Tucker. Red and photographic infrared linear combinations for monitoring vegetation. *Remote sensing of Environment*, 8(2):127–150, 1979.

D. Tuia, J. Verrelst, L. Alonso, F. Pérez-Cruz, and G. Camps-Valls. Multioutput support vector regression for remote sensing biophysical parameter estimation. *IEEE Geoscience and Remote Sensing Letters*, 8(4):804–808, 2011.

F. Tupin, J. Inglada, and J.-M. Nicolas. *Remote Sensing Imagery*. John Wiley & Sons, 2014.

S. R. Vadyala and S. N. Betgeri. General implementation of quantum physics-informed neural networks. *Array*, 18:100287, 2023. ISSN 2590-0056. doi: https://doi.org/10.1016/j.array.2023.100287.

A. van den Oord, O. Vinyals, and k. kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf.

W. Verhoef. Influence of crop geometry on multispectral reflectance determined by the use of canopy reflectance models. In *Proceedings of the International Colloquium on Spectral Signatures of Objects in Remote Sensing, Avignon*, Sept. 1981.

W. Verhoef. Light scattering by leaf layers with application to canopy reflectance modeling: The sail model. *Remote Sensing of Environment*, 16(2):125–141, 1984. ISSN 0034-4257. doi: \url{https://doi.org/10.1016/0034-4257(84)90057-9}.

W. Verhoef. *Theory of Radiative Transfer Models Applied in Optical Remote Sensing of Vegetation Canopies*. NLR TP.: Nationaal Lucht- en Ruimtevaartlaboratorium. Nationaal Lucht- En Ruimtevaartlaboratorium, 1998. ISBN 9789054858041. URL https://books.google.fr/books?id=HWXXpwAACAAJ.

W. Verhoef, L. Jia, Q. Xiao, and Z. Su. Unified optical-thermal four-stream radiative transfer theory for homogeneous vegetation canopies. *IEEE Transactions on Geoscience and Remote Sensing*, 45(6):1808–1822, 2007. doi: 10.1109/TGRS.2007.895844.

Y. Wang, D. Blei, and J. P. Cunningham. Posterior collapse and latent variable non-identifiability. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 5443–5455. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/2b6921f2c64dee16ba21ebf17f3c2c92-Paper.pdf.

R. H. Waring, W. G. Thies, and D. Muscato. Stem growth per unit of leaf area: a measure of tree vigor. *Forest Science*, 26(1):112–117, 1980.

D. J. Watson. Comparative physiological studies on the growth of field crops: I. variation in net assimilation rate and leaf area between species and varieties, and within and between years. *Annals of botany*, 11(41):41–76, 1947.

L. Weidong, F. Baret, G. Xingfa, T. Qingxi, Z. Lanfen, and Z. Bing. Relating soil surface moisture to reflectance. *Remote Sensing of Environment*, 81(2):238–246, 2002. ISSN 0034-4257. doi: https://doi.org/10.1016/S0034-4257(01)00347-9.

M. Weiss and F. Baret. S2toolbox level 2 products: Lai, fapar, fcover. Technical report, ESA, May 2016. URL http://step.esa.int/docs/extra/ATBD_S2ToolBox_V2.1.pdf.

M. Weiss, F. Baret, G. Smith, I. Jonckheere, and P. Coppin. Review of methods for in situ leaf area index (lai) determination: Part ii. estimation of lai, errors and sampling. *Agricultural and Forest Meteorology*, 121(1):37–53, 2004. ISSN 0168-1923. doi: \url{https://doi.org/10.1016/j.agrformet.2003.08.001}.

A. K. Whitcraft, I. Becker-Reshef, C. O. Justice, L. Gifford, A. Kavvada, and I. Jarvis. No pixel left behind: Toward integrating earth observations for agriculture into the united nations sustainable development goals framework. *Remote Sensing of Environment*, 235: 111470, 2019. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2019.111470. URL https://www.sciencedirect.com/science/article/pii/S0034425719304894.

J. D. Willard, X. Jia, S. Xu, M. S. Steinbach, and V. Kumar. Integrating physics-based modeling with machine learning: A survey. *ArXiv*, abs/2003.04919, 2020.

V. V. Williams. Multiplying matrices faster than coppersmith-winograd. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 887–898, 2012.

D. Wilson and T. R. Martinez. The general inefficiency of batch training for gradient descent learning. *Neural Networks*, 16(10):1429–1451, 2003. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(03)00138-2. URL https://www.sciencedirect.com/science/article/pii/S0893608003001382.

J. W. Wilson. Inclined point quadrats. *The New Phytologist*, 59(1):1–8, 1960. ISSN 0028646X, 14698137. URL http://www.jstor.org/stable/2485037.

S. Wright. Correlation and causation. *Journal of agricultural research*, 20(7):557, 1921.

Q. Xie, J. Dash, A. Huete, A. Jiang, G. Yin, Y. Ding, D. Peng, C. C. Hall, L. Brown, Y. Shi, H. Ye, Y. Dong, and W. Huang. Retrieval of crop biophysical parameters from sentinel-2 remote sensing imagery. *International Journal of Applied Earth Observation and Geoinformation*, 80:187–195, 2019. ISSN 1569-8432. doi: \url{https://doi.org/10.1016/j.jag.2019.04.019}.

X. Yang, J. Mustard, J. Tang, and H. Xu. Regional-scale phenology modeling based on meteorological records and remote sensing observations. *Journal of Geophysical Research*, 117, 2012.

Y. Yang, X. Zhang, Q. Guan, and Y. Lin. Making invisible visible: Data-driven seismic inversion with spatio-temporally constrained data augmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–16, 2022. doi: 10.1109/TGRS.2022.3144636.

M. Zemp, Q. Chao, A. J. Han Dolman, M. Herold, T. Krug, S. Speich, K. Suda, P. Thorne, and W. Yu. Gcos 2022 implementation plan. *Global Climate Observing System GCOS*, page 85, 2022.

L. Zeng, B. D. Wardlow, D. Xiang, S. Hu, and D. Li. A review of vegetation phenological metrics extraction using time-series, multispectral satellite data. *Remote Sensing of Environment*, 237:111511, Feb. 2020. ISSN 0034-4257. doi: 10.1016/j.rse.2019.111511. URL https://doi.org/10.1016/j.rse.2019.111511.

Y. Zérah, S. Valero, and J. Inglada. Méthodes probabilistes d'apprentissage profond avec a priori physiques de représentations interprétables. In *GRETSI 2022: XXVIIIème Colloque Francophone de Traitement du Signal et des Images*, Nancy, France, Sept. 2022a. URL https://hal.science/hal-04186427.

Y. Zérah, S. Valero, and J. Inglada. Méthodes probabilistes d'apprentissage profond avec a priori physiques de représentations interprétables, Sept. 2022b.

Y. Zérah, S. Valero, and J. Inglada. Interpretable representation learning for high resolution satellite image time series, May 2022c.

M. Zhang, T. Guo, G. Zhang, Z. Liu, and W. Xu. Physics-informed deep learning for structural vibration identification and its application on a benchmark structure. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 382(2264):20220400, 2024. doi: 10.1098/rsta.2022.0400. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2022.0400.

R. Zhang, P. Yang, S. Liu, C. Wang, and J. Liu. Evaluation of the methods for estimating leaf chlorophyll content with spad chlorophyll meters. *Remote Sensing*, 14(20), 2022. ISSN 2072-4292. doi: 10.3390/rs14205144. URL https://www.mdpi.com/2072-4292/14/20/5144.

X. Zhang, M. A. Friedl, C. B. Schaaf, A. H. Strahler, J. C. Hodges, F. Gao, B. C. Reed, and A. Huete. Monitoring vegetation phenology using MODIS. *Remote Sensing of Environment*, 84(3):471–475, 2003.

Y. Zhang, M. Hallikainen, H. Zhang, H. Duan, Y. Li, and X. S. Liang. Chlorophyll-a estimation in turbid waters using combined sar data with hyperspectral reflectance data: A case study in lake taihu, china. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(4):1325–1336, 2018. doi: 10.1109/JSTARS.2017.2789247.

S. Zhao, J. Song, and S. Ermon. Infovae: balancing learning and inference in variational autoencoders. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33015885. URL https://doi.org/10.1609/aaai.v33i01.33015885.

Z. Zheng, L. Wang, L. Yang, and Z. Zhang. Generative probabilistic wind speed forecasting: A variational recurrent autoencoder based method. *IEEE Transactions on Power Systems*, 37(2):1386–1398, 2022. doi: 10.1109/TPWRS.2021.3105101.

G. Zhong, L.-N. Wang, X. Ling, and J. Dong. An overview on data representation learning: From traditional feature learning to recent deep learning. *The Journal of Finance and Data Science*, 2(4):265–278, 2016. ISSN 2405-9188. doi: https://doi.org/10.1016/j.jfds.2017.05.001. URL https://www.sciencedirect.com/science/article/pii/S2405918816300459.

W. Zhong and H. Meidani. Pi-vae: Physics-informed variational auto-encoder for stochastic differential equations. *Computer Methods in Applied Mechanics and Engineering*, 403:115664, 2023. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2022.115664.

W. Zhu, Y. Pan, H. He, L. Wang, M. Mou, and J. Liu. A changing-weight filter method for reconstructing a high-quality NDVI time series to preserve the integrity of vegetation phenology. *IEEE Transactions on Geoscience and Remote Sensing*, 50(4):1085–1094, 2012. doi: 10.1109/TGRS.2011.2166965.

Y. Zérah, S. Valero, and J. Inglada. Sentinel-2 time series for Pheno-VAE, Nov. 2022. URL https://doi.org/10.5281/zenodo.7273500.

Y. Zérah, S. Valero, and J. Inglada. Physics-driven probabilistic deep learning for the inversion of physical models with application to phenological parameter retrieval from satellite times series. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–23, 2023a. doi: 10.1109/TGRS.2023.3284992.

Y. Zérah, S. Valero, and J. Inglada. Physics-driven probabilistic deep learning for the inversion of physical models with application to phenological parameter retrieval from satellite times series. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–23, 2023b. doi: 10.1109/TGRS.2023.3284992.

Y. Zérah, S. Valero, and J. Inglada. Physics-constrained deep learning for biophysical parameter retrieval from sentinel-2 images: Inversion of the prosail model. *Remote Sensing of Environment*, 312:114309, 2024. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2024.114309. URL https://www.sciencedirect.com/science/article/pii/S0034425724003274.

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Université de Toulouse

Titre : apprentissage de representations physiques de la végétations à partir de données de télédétection optique.

Mots clés : Apprentissage Profond, Autoencodeurs variationnels, Images Satellite, Modélisation Physique, Séries Temporelles

Résumé : Le changement climatique initié par les activités humaines provoque des transformations drastiques et sans précédent des écosystèmes et des zones habitées dans le monde entier. La télédétection s'impose comme un outil essentiel pour observer la Terre, pour comprendre le fonctionnement de la biosphère ainsi que son altération par les pressions anthropiques. Les capacités d'observation par télédétection spatiale ainsi que les techniques de traitement du signal ont rapidement évolué lors des dernières décennies. L'émergence des techniques d'apprentissage statistique modernes et l'augmentation exponentielle de la puissance de calcul disponible sont cruciaux dans l'exploitation de l'immense volume de données produits par les capteurs de télédétection. En particulier, la mission Sentinel-2 produit des images multi-spectrales à haute résolution spatiale et temporelle depuis une dizaine d'années à une échelle globale, diffusées gratuitement avec une politique d'accès libre. Les produits Sentinel-2 ont permis le développement de diverses applications, telles que l'amélioration des techniques agricoles, la gestion du territoire et la réponse aux catastrophes naturelles. Les données de télédétections sont des mesures de radiations électromagnétiques dont les caractéristiques sont reliées à la nature des élements et aux processus à la surface de la Terre. L'extraction de représentations contenant des informations pertinentes sur ces éléments est fondamental pour les applications de télédétection. L'objectif de cette thèse est de proposer une méthode d'inférence de telles representations à partir de données de télédétection. Plusieurs défis se présentent pour estimer ces représentations. D'abord, ces représentations doivent êtres générales et interprétables, afin d'être utilisables par plusieurs applications. Cela peut être réalisé avec des variables bio-physiques qui caractérisent les systèmes observés, par exemple le contenu minéral et en eau des sols ou la concentration en pigments et la structure de la canopée pour la végétation. Par ailleurs, les representations doivent êtres associés à une incertitudes d'estimation. Le manque de données de référence pose aussi un défi. Contrairement aux acquisitions de télédétection, il est difficile d'obtenir des vérités terrain. Les bases de données qui associent des variables bio-physiques de la végétation et des données de télédétection sont rares. Les approches qui estiment ces variables utilisent donc la modélisation physique et l'inversion. Cette thèse est divisée en trois parties principales qui détaillent ses quatre contributions. La première contribution est la démonstration de la dépendance des modèles de régression supervisée au choix de la distribution d'échantillonnage pour leur jeu de données d'entraînement. La seconde contribution est le développement d'une méthodologie d'estimation de variable physiques non supervisée à partir de données de télédétection, basée sur les Autoencodeurs Variationnels (VAE). Cela consiste en l'incorporation de modèles et de connaissances physiques à priori dans un modèle d'apprentissage profond. Cette approche utilise la reconstruction comme tâche intermédiaire pour estimer une variable physique, plutôt que la comparaison avec une verité terrain indisponible ou une référence simulée. Dans une troisième partie, ce manuscript détaille les deux autres contributions de cette thèse: l'application de la méthodologie proposée à l'estimation de variables physiques dans deux applications. Dans la première le modèle de transfert radiatif PROSAIL est utilisé dans le modèle PROSAIL-VAE afin d'estimer les caractéristiques de feuilles et de canopées à partir d'images Sentinel-2. La validation avec des données in-situ a permit de confirmer les performances de cette approche. Dans la seconde application, des variables phénologiques caractérisant le comportement temporel de la végétation sont estimées à partir de séries temporelles de NDVI, avec le modèle Phéno-VAE.

Title: physics-based representation learning for vegetation from optical remote sensing

Key words: Deep Learning, Variational Autoencoders, Satellites Images, Physical modeling, Time series

Abstract: Human-driven climate change is triggering unprecedented and dire transformations of ecosystems and habitats worldwide. Remote sensing offers precious tools for monitoring the state of the Earth, and for understanding how the biosphere functions and is affected by human activities. Satellite remote sensing capabilities and data processing techniques have rapidly improved over the last decades, and have considerably advanced the study of life processes on land masses. The advent of modern machine learning and exponential development of computational power are crucial for the exploitation of the vast amount of data produced by remote sensors. In particular, the Sentinel-2 mission has been providing for a decade high spatial and temporal resolution multi-spectral images at a global scale. Sentinel-2 products are released with an open-data policy that supports research efforts and various applications, such as the enhancement of agricultural practices, land management and disaster response. Remote sensing data is a measurement of incoming radiation and its properties are related to the nature of elements and processes on the surface of the Earth. Extracting useful representations that contain relevant information is fundamental for applications of remote sensing. The objective of this thesis is to find useful representations from remote sensing data for use in subsequent applications. There are several challenges in the retrieval of such representations. First, in order to be useful to different tasks, the representations need to be general and interpretable. This can be achieved with bio-physical variables that characterize the target system, for instance the water and mineral content in the soil, the pigment concentrations and canopy structure for vegetation. Also, remote sensing data has an intrinsic uncertainty, and representations of this data should be associated with a measure of uncertainty. Another challenge lies in the scarcity of reference data in remote sensing. Although remote sensing measurements are a big data, it is difficult to obtain ground truth data, for instance databases of vegetation bio-physical parameters that can be related to remote sensing measurements are rare. Methods that attempt to retrieve such parameters therefore commonly resort to physical modeling and inversion. This Ph.D. is divided into three main parts, which are associated with its main contributions. Its first contribution is the identification of a key issue of supervised regression models that perform model inversion. Their performance is shown to be very dependent on the choice of the sampling distribution for simulating their training data-sets. The second contribution of this Ph.D. is the development of a self-supervised approach for retrieving physical representations of remote sensing data. This approach is based on the framework of Variational Autoencoders. The proposed methodology is based on the incorporation of a physical model and physical knowledge for a end-to-end deep learning framework. Instead of attempting to optimize the physical variable retrieval from an unavailable ground truth or a biased simulated reference, this method uses input data reconstruction as a proxy task. Finally, in a third part, this thesis reports the results of the application of the proposed approach on the retrieval of physical variables in two settings. In a first experiment, it is used with the PROSAIL radiative transfer model for retrieving leaf contents and canopy structure variables. The resulting PROSAIL-VAE model is trained directly using Sentinel-2 multi-spectral images. Validation with some in-situ data have corroborated the performance of the approach. In a second application, the proposed approach is used to retrieve phenological variables that characterize the temporal behavior of vegetation. The so-called Pheno-VAE is trained on annual NDVI time series extracted from Sentinel-2 data.