Multi-fidelity Gaussian Process Emulation for Atmospheric Radiative Transfer Models

Jorge Vicent, Luca Martino, Jochem Verrelst and Gustau Camps-Valls *Fellow, IEEE*

Abstract

Atmospheric radiative transfer models (RTMs) are widely used in satellite data processing to correct for the scattering and absorption effects caused by aerosols and gas molecules in the Earth's atmosphere. As the complexity of RTMs grows and the requirements for future Earth Observation missions become more demanding, the conventional Look-Up Table (LUT) interpolation approach faces important challenges. Emulators have been suggested as an alternative to LUT interpolation, but they are still too slow for operational satellite data processing. Our research introduces a solution that harnesses the power of multi-fidelity methods to improve the accuracy and runtime of Gaussian Process (GP) emulators. We investigate the impact of the number of fidelity layers, dimensionality reduction, and training dataset size on the performance of multi-fidelity GP emulators. We find that an optimal multi-fidelity emulator can achieve relative errors in surface reflectance below 0.5% and performs atmospheric correction of hyperspectral PRISMA satellite data (one million pixels) in a few minutes. Additionally, we provide a suite of functions and tools for automating the creation and generation of atmospheric RTM emulators.

Index Terms

Atmospheric correction, emulation, Gaussian processes, hyperspectral, muti-fidelity, radiative transfer models

Manuscript submitted March 30, 2023. Luca Martino was supported by the Young Researchers R&D Project, ref. num. F861 (AUTO-BA- GRAPH) funded by Community of Madrid and Rey Juan Carlos University, and by Agencia Estatal de Investigacion AEI (project SP-GRAPH, ref. num. PID2019-105032GB-I00). Jochem Verrelst was supported by the European Research Council (ERC) under the ERC-2017-STG SENTIFLEX project (grant agreement 755617). Gustau Camps-Valls was supported by the ERC under the ERC-CoG-2019 USMILE project (grant agreement 855187) and the Marie Curie ETN iMIRACLI (860100).

J. Vicent, J. Verrelst and G. Camps-Valls are with Image Processing Laboratory, University of Valencia, Spain. E-mail: [jorge.vicent, jochem.verrelst, gustau.camps]@uv.es.

L. Martino is with the University Rey Juan Carlos, Madrid, Spain and the University of Catania, Catania, Italy. Email: luca.martino@unict.it,@urjc.es

I. Introduction

Atmospheric radiative transfer models (RTM) are computer codes that simulate the interaction of electromagnetic radiation with Earth's atmosphere, i.e., gas molecules, aerosols, and clouds. These models describe the absorption, scattering, and emission of radiation by atmospheric constituents based on physical principles and given a configuration of optical properties and geometric conditions. Atmospheric RTMs are used in a variety of applications, such as remote sensing of the atmosphere (e.g., SMART-G [1]), numerical weather prediction (e.g., RTTOV [2]), and atmospheric correction of satellite data (e.g., 6SV [3], libRadtran [4], MODTRAN [5]). These models' increasing complexity and realism result in a computational burden that limits their direct use in operational applications. Therefore, a possible solution is to interpolate look-up tables (LUT) of pre-computed RTM simulations as it is commonly applied in operational atmospheric correction of satellite data [6]. However, the gridded distribution of input variables increases LUT size exponentially, implying an increasing time to generate it and stringent RAM requirements [7], [8]. This is particularly challenging for hyperspectral satellite missions such as CHIME [9], EMIT [10], EnMAP [11], FLEX [12], PRISMA [13] and SBG [14]. With several hundred spectral channels, the data volume of LUTs will increase by one or two orders of magnitude with respect to current multispectral missions [15], limiting the applicability of state-of-the-art atmospheric correction algorithms [16].

Statistical regression models known as *emulators*, or surrogate models, have been proposed as an alternative to LUT interpolation [17], [18]. Emulation approximates the behaviour of a deterministic model at a fraction of its runtime, reducing the LUT size and interpolation errors [19]. In [20], we assessed key configuration elements that affect the performance of an emulator (training dataset size, dimensionality reduction, regression model). However, their runtime is slow for operational satellite data processing, and the combination with LUT interpolation is still needed to achieve the required performances [21]. To further enhance the accuracy of model predictions whilst reducing runtime, multi-fidelity methods have been developed [22]. These methods combine limited simulations of an accurate but computationally expensive model (high-fidelity) with a larger simulation dataset from a fast but less accurate model (low-fidelity) [23]. By merging various fidelities, multi-fidelity methods correct computationally cheap models so that the outputs resemble those of more accurate models. In the context of atmospheric RTMs, multi-fidelity methods have been implemented in MODTRAN's scaled-DISORT method [24] and more

recently in the Cluster Low Streams Regression (CLSR) method [25]. These implementations rely on simplistic approximations or have limited applicability. The scaled-DISORT method scales MODTRAN's 2-stream simulations using a scaling factor obtained from DISORT simulations at fewer wavelengths and interpolates linearly for all remaining wavelengths. Conversely, the CLSR method is only applied on a small spectral range around gaseous absorptions (e.g. O2-A band).

Considering the above limitations, our main objective is to improve the runtime and accuracy of atmospheric RTM emulators by using multi-fidelity techniques. Our study expands upon our previous work [20] by integrating multi-fidelity techniques with well-established Gaussian Process (GP) regression [26] and dimensionality reduction into a unified emulation framework. In particular, we explore various configurations of multi-fidelity emulators in the context of atmospheric correction for operational satellite data processing chains. The implemented approach is generic and valid for multiple RTM and covers the full spectral range (400-2500 nm). Moreover, we provide the scientific community with a practical and automated workflow and tools for generating atmospheric RTM training datasets and developing emulators applicable to real-world applications.

II. Gaussian processes emulation

Based on our previous results [20], GP regression has been selected in this work. In practice, an emulator uses a *training* dataset of precomputed model simulations to infer the output values of a deterministic model for an ensemble of unseen input conditions. The following paragraphs give a description of the implemented approach for multi-fidelity emulation.

A. Gaussian Processes for regression

A GP for regression is a Bayesian non-parametric method for non-linear regression [27]–[31]. Let us consider a training dataset $\{\mathbf{x}_i, y_i\}_{i=1}^n$ with inputs $\mathbf{x}_i = [x_{i1}, \dots, x_{id}]^\top \in \mathbb{R}^d$ (e.g., atmospheric and geometric conditions), and their corresponding scalar outputs $y_i \in \mathbb{R}$ (e.g., the transmittance at a given wavelength). We define the $d \times n$ matrix $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$ and the $n \times 1$ vector $\mathbf{y} = [y_1, \dots, y_n]^\top$. The observation model is defined as $y_i = g(\mathbf{x}_i) + \epsilon_i$, where $g(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ is the unknown underlying function that we want to approximate (i.e., an atmospheric RTM). The parameter ϵ_i is an additive Gaussian noise with zero mean and variance σ^2, i.e., $\epsilon_i \sim \mathcal{N}(\epsilon|0, \sigma^2)$. The model $g(\mathbf{x})$ can also be expressed as $g(\mathbf{x}) = \mathbf{h}(\mathbf{x})^\top \boldsymbol{\beta} + f(\mathbf{x})$.

Here, $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \ldots, h_p(\mathbf{x})]^\top : \mathbb{R}^d \to \mathbb{R}^p$ is a parametric model formed by p basis functions $h_j(\mathbf{x})$ and $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_p]^\top$ their corresponding scalar coefficients β_j [30]. The function $f(\mathbf{x})$ is a non-parametric Gaussian process model with zero mean and a covariance determined by a *kernel* function, $k(\mathbf{x}, \mathbf{x}^*)$. This kernel function models the similarity between input samples calculating the linear dot product in a higher dimensional feature space. Commonly used kernel functions are radial basis functions such as the Gaussian kernel Eq. (1):

$$k(\mathbf{x}, \mathbf{x}^*) = \theta_f^2 \exp\left(-\frac{(\mathbf{x} - \mathbf{x}^*)^\top (\mathbf{x} - \mathbf{x}^*)}{2\theta_l^2}\right), \tag{1}$$

where θ_f and θ_l are hyper-parameters that must be tuned. An extension of the Gaussian kernel is obtained by the so-called automatic relevance determination (ARD) [32] in Eq. (2):

$$k(\mathbf{x}, \mathbf{x}^*) = \theta_f^2 \exp\left(-\sum_{i=1}^{d} \frac{(x_i - x_i^*)^2}{2\theta_{l,i}^2}\right) \tag{2}$$

with a scale-length parameter $\theta_{l,i}$ for each input dimension i.

Considering the Gaussian Process prior over $f(\mathbf{x})$, it is possible to show that the posterior of $g(\mathbf{x})$ given the data $\mathbf{y}$, i.e. $p\left(g(\mathbf{x})|\mathbf{y}\right)$, is a Gaussian distribution with mean $\widehat{g}(\mathbf{x})$ and covariance $\sigma_{\texttt{post}}$ shown in the equations below:

$$\widehat{g}(\mathbf{x}) = \mathbf{h}(\mathbf{x})^\top \boldsymbol{\beta} + \mathbf{k}(\mathbf{x})^\top \boldsymbol{\alpha}, \tag{3}$$

$$\sigma_{\texttt{post}}^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^\top \left(\mathbf{K} + \sigma^2 \mathbf{I}_n\right)^{-1} \mathbf{k}(\mathbf{x}), \tag{4}$$

where $\boldsymbol{\alpha} = \left(\mathbf{K} + \sigma^2 \mathbf{I}_N\right)^{-1} (\mathbf{y} - \mathbf{H}\boldsymbol{\beta})$. Here $\mathbf{K} = [\mathbf{k}(\mathbf{x}_1), \ldots, \mathbf{k}(\mathbf{x}_n)]$ is the $n \times n$ kernel matrix where $\mathbf{k}(\mathbf{x}) = [k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_n)]^\top$ ($n \times 1$ vector) and $\mathbf{H} = [\mathbf{h}(\mathbf{x}_1), \ldots, \mathbf{h}(\mathbf{x}_n)]^\top$ is a $n \times p$ matrix.

A GP model, $\mathcal{GP}(\mathbf{x}|\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma)$, is defined by its hyperparameters $\boldsymbol{\beta}$, $\boldsymbol{\theta}$ and σ, which can be estimated from the training dataset by maximizing the marginal likelihood [30]. Once obtained these hyperparameters, a prediction of y_q at a new query point $\mathbf{x}_q$ is given by Eq. (3) where $\boldsymbol{\alpha}$ is calculated from the training dataset. In this work, we choose a constant basis function $\mathbf{h}(\mathbf{x})$=1 with p=1 (consequently β being a scalar) and the ARD-Gaussian kernel in Eq. (2).

B. Emulation

Let us consider now the application of GP for the emulation of multi-output model $\mathbf{g}(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^b$, which is used to construct a training dataset $\{\mathbf{x}_i, \mathbf{g}(\mathbf{x}_i)\}_{i=1}^n$, We define the $b \times n$

matrix $\mathbf{Y} = [\mathbf{g}(\mathbf{x}_1), \ldots, \mathbf{g}(\mathbf{x}_n)] = [\mathbf{y}_1, \ldots, \mathbf{y}_b]^\top$ of multi-output data, with the $n \times 1$ vector $\mathbf{y}_i = [y_{i1}, \ldots, y_{in}]^\top$. A straightforward option for emulation would be to train an individual GP model $\{\widehat{g}_\lambda(\mathbf{x})\}_{\lambda=1}^{b}$ for each of the b outputs so that $\widehat{\mathbf{g}}(\mathbf{x}) = [\widehat{g}_1(\mathbf{x}), \ldots, \widehat{g}_b(\mathbf{x})]^\top$, where each $\widehat{g}_\lambda(\mathbf{x})$ is a model described as in the previous section. Although conceptually simple, this option is computationally inefficient as it involves predicting individually hundreds or thousands of spectral outputs calculated by an atmospheric RTM (i.e., $b \approx 10^3$). The alternative method proposed in [18] exploits the large correlations in spectroscopic data [33], combining dimensionality reduction with GP. In this method, an individual GP is trained to predict each component of the spectral data projected in a lower dimensional space. After the prediction of each component, the inverse transformation of the dimensionality reduction is then applied to reconstruct the output spectra at the expense of some loss in accuracy. This work uses principal component analysis (PCA) [34] for dimensionality reduction. PCA has been widely used to accurately reconstruct spectral atmospheric RTM data [35]–[37]. It exploits the correlations in the $b \times n$ matrix $\mathbf{Y}$ to project the original data into a space of orthogonal linearly uncorrelated directions that maximize the variance of the projections, i.e., $\widehat{\mathbf{Y}} = \mathbf{U}\mathbf{Y}$. Here, $\widehat{\mathbf{Y}}$ (same dimensions as $\mathbf{Y}$) and $\mathbf{U}$ ($b \times b$ matrix) are the eigenvectors of the output covariance matrix, $\mathbf{C}_y = \mathbf{Y}\mathbf{Y}^\top$. Since the eigenvector matrix is orthogonal, the original spectral data is reconstructed by the inverse transformation $\mathbf{Y} = \mathbf{U}^\top\widehat{\mathbf{Y}}$. The dimensionality reduction by PCA is possible by selecting fewer components $n_c \ll b$ from the eigenvectors matrix $\mathbf{U}$, i.e., $\widetilde{\mathbf{U}}$ ($n_c \times b$ matrix). Typically, one takes the first n_c <15 components with higher variance to retain 99.99% of total variance [20]. Larger values of n_c only provide marginal improvements in accuracy while slowing down an emulator. With the dimensionality reduction strategy, the process of training a GP emulator and using it for prediction is summarized in pseudo-code 1 and 2.

Algorithm 1 GP emulator (*training*)

Input: Training dataset ($\mathbf{X}$ and $\mathbf{Y}$)
  Obtain PCA eigenvalues/vectors: $\widehat{\mathbf{Y}}$, $\mathbf{U}$
  Select the first n_c components of $\widehat{\mathbf{Y}}$ and $\mathbf{U}$
  for c=1 **to** n_c **do**
    σ_c^2, β_c, $\boldsymbol{\theta}_c \leftarrow$ Train $\mathcal{GP}_c$ model
  end for
Output: GP models and PCA transformation: $\{\mathcal{GP}\}_{c=1}^{n_c}$, $\mathbf{U}$

The training process is only carried out once, storing the GP hyperparameters and PCA eigenvector matrix for later use in the prediction. Notice also that, in the context of atmospheric

Algorithm 2 GP emulator (*prediction*)

Input: Query point(s) ($\mathbf{x}_q$), $\{\mathcal{GP}\}_{c=1}^{n_c}$ and $\mathbf{U}$
  for c=1 **to** n_c **do**
    $\widehat{y}_{c,q} = \widehat{g}_c(\mathbf{x}_q) \leftarrow$ Run $\mathcal{GP}_c$ model
  end for
  • Define the $n_c \times$ 1 vector $\widehat{\mathbf{r}}_q = [\widehat{y}_{1,q}, ..., \widehat{y}_{n_c,q}]^\top$.
  • $\widetilde{\mathbf{y}}_q = \widetilde{\mathbf{U}}^\top \widehat{\mathbf{r}}_q \leftarrow$ Invert PCA transformation
Output: $\widetilde{\mathbf{y}}_q = [\widetilde{y}_{1q}, ..., \widetilde{y}_{bq}]^\top \in \mathbb{R}^b$.

RTMs, we train an emulator for each of the following six atmospheric transfer functions (spectral magnitudes): path radiance ($\mathbf{L}_0$), at-surface direct/diffuse solar irradiance ($\mathbf{E}_{\text{dir/dif}}$), spherical albedo ($\mathbf{S}$), and direct/diffuse target-to-sensor transmittance ($\mathbf{T}_{\text{dir/dif}}$). With the notation in the previous paragraphs, and using the direct transmittance as an example, $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_b]^\top \equiv [\mathbf{T}_{dir,1}, \ldots, \mathbf{T}_{dir,b}]^\top$ is the collection of training data for the direct transmittance.

C. Multi-fidelity emulation

Mathematically, a multi-fidelity model can be expressed through the equation $\widehat{\mathbf{g}}_t(\mathbf{x}) = c \cdot \widehat{\mathbf{g}}_{t-1}(\mathbf{x}) + \boldsymbol{\delta}_t(\mathbf{x})$, where $\widehat{\mathbf{g}}_t$ and $\widehat{\mathbf{g}}_{t-1}$ are two subsequent fidelity layers of the model executed at the input conditions in $\mathbf{x}$, c is a scaling factor of the lower fidelity later, and $\boldsymbol{\delta}_t(\mathbf{x}) \in \mathbb{R}^b$ models the bias between two fidelities. The emulator method proposed in [21] builds up on this idea, setting c=1 and using 6SV as the lowest fidelity model, $\mathbf{g}_0(\mathbf{x})$, with a neural network trained to model the difference $\boldsymbol{\delta}_1(\mathbf{x})$ between 6SV and MODTRAN (high-fidelity) simulations, as shown below. This work extends this approach sequentially by adding higher fidelities layers.

For the lowest fidelity, $\widehat{\mathbf{g}}_0(\mathbf{x})$, we fit a 2$^{\text{nd}}$ order polynomial for the d-dimensional points in $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, and $\mathbf{y}_\lambda = [y_{\lambda 1}, \ldots, y_{\lambda n}]^\top$ i.e., a surface fitting for each wavelength λ=1 to b. The polynomial model is not only fast predicting new outputs but also representative of the main trends that describe the dependencies between input and output spectral data of atmospheric RTMs (mainly exponential, cosine, and power functions). For the first higher fidelity layer, we evaluate the polynomial on the training dataset to get $\widehat{\mathbf{g}}_0(\mathbf{x}_i)$ ($i = 1$ to n) and calculate the difference between the training data and the predictions by the lowest fidelity model, i.e., $\boldsymbol{\delta}_1(\mathbf{x}_i) = \mathbf{g}(\mathbf{x}_i) - \widehat{\mathbf{g}}_0(\mathbf{x}_i)$ ($i = 1$ to n). We then construct a new training dataset $\{\mathbf{x}_i, \boldsymbol{\delta}_1(\mathbf{x}_i)\}_{i=1}^n$ to train a GP emulator $\widehat{\boldsymbol{\delta}}_1(\mathbf{x})$ to approximate $\boldsymbol{\delta}_1(\mathbf{x})$. Accordingly, the first higher fidelity layer is $\widehat{\mathbf{g}}_1(\mathbf{x}) = \widehat{\mathbf{g}}_0(\mathbf{x}) + \widehat{\boldsymbol{\delta}}_1(\mathbf{x})$. This process is repeated for a user-defined number of layers n_l. For example, for the second layer, we apply a GP emulator of the previous layer to construct a new

training dataset $\{\mathbf{x}_i, \boldsymbol{\delta}_2(\mathbf{x}_i)\}_{i=1}^n$, where $\boldsymbol{\delta}_2(\mathbf{x}_i) = \mathbf{g}(\mathbf{x}_i) - \widehat{\mathbf{g}}_1(\mathbf{x}_i)$. This new dataset is then used to train another GP emulator $\widehat{\boldsymbol{\delta}}_2(\mathbf{x})$ to approximate $\boldsymbol{\delta}_2(\mathbf{x})$. The prediction for a 2-layers multi-fidelity emulator would be $\widehat{\mathbf{g}}_2(\mathbf{x}) = \widehat{\mathbf{g}}_1(\mathbf{x}) + \widehat{\boldsymbol{\delta}}_2(\mathbf{x}) = \widehat{\mathbf{g}}_0(\mathbf{x}) + \widehat{\boldsymbol{\delta}}_1(\mathbf{x}) + \widehat{\boldsymbol{\delta}}_2(\mathbf{x})$. The multi-fidelity process is summarized in the following pseudo-code:

Algorithm 3 Multi-fidelity GP emulator (*training*)

Input: Training dataset ($\mathbf{X}$ and $\mathbf{Y}$)
Fit 2[nd] degree polynomials for the data $\{\mathbf{x}_i, \mathbf{g}(\mathbf{x}_i)\}_{i=1}^n$: $\widehat{\mathbf{g}}_0(\mathbf{x})$
for t=1 **to** n_l **do**
- $\{\widehat{\mathbf{g}}_{t-1}(\mathbf{x}_i)\}_{i=1}^n \leftarrow$ Run $t-1$ fidelity layer on training data
- Calculate $\boldsymbol{\delta}_t(\mathbf{x}_i) = \mathbf{g}(\mathbf{x}_i) - \widehat{\mathbf{g}}_{t-1}(\mathbf{x}_i)$ for $i = 1$ to n
- $\widehat{\boldsymbol{\delta}}_t(\mathbf{x}) \leftarrow$ Train emulator considering the pairs input-outputs $\{\mathbf{x}_i, \boldsymbol{\delta}_t(\mathbf{x}_i)\}_{i=1}^n$
- Set $\widehat{\mathbf{g}}_t(\mathbf{x}) = \widehat{\mathbf{g}}_{t-1}(\mathbf{x}) + \widehat{\boldsymbol{\delta}}_t(\mathbf{x})$

end for
Output: Emulators and polynomial fitting: $\{\widehat{\boldsymbol{\delta}}_t(\mathbf{x})\}_{t=1}^{n_l}$, $\widehat{\mathbf{g}}_0(\mathbf{x})$

III. Materials and Methods

A. Simulated dataset and tools

To train the GP models in an emulator, the training dataset should cover the d-dimensional input space so that the ensemble of simulations is representative of the underlying correlations between the RTM inputs and outputs. Following up with our previous work [20], we applied the Latin Hypercube Sampling (LHS) [38] to cover the full variability of the input parameter space with the minimum and maximum boundaries given in Table I. These input variables are typically used in atmospheric correction as they are the main contributors to the scattering and absorption of electromagnetic radiation in the optical spectral range [39]. Without loss of generality, we used MODTRAN6 as the underlying RTM in our emulator training datasets. We configured the RTM simulations in the 400-2500 nm spectral range with a sampling of 5 cm^{-1} (i.e., 0.08 nm to 3 nm), resulting in nearly 4200 wavelengths.

As for the training dataset size, we built three datasets with 100, 500, and 1000 samples using LHS to study the effect of dataset size on accuracy. These sizes were suggested in earlier studies to have a balance between the runtime and accuracy [20], [40], [41]. For GP emulators, larger datasets only provide a residual gain in accuracy at the expense of increasing the runtime proportionally and potentially causing problems during the training phase due to RAM limitations. To compare the performance of all emulators, we simulated a *reference*

TABLE I
RANGE OF RTM INPUT VARIABLES. VIEWING ZENITH IS SET TO 0°.

Input variables	**Units**	**Min.**	**Max.**
O_3 column concentration (O3)	[amt-cm]	0.25	0.45
Columnar Water Vapour (CWV)	[g·cm^{-2}]	0.2	4
Aerosol Optical Thickness (AOT)	unit-less	0.04	0.6
Asymmetry parameter (g)	unit-less	0.5	85
Ångström exponent (α)	unit-less	0.1	2
Single Scattering Albedo (SSA)	unit-less	0.8	1
Surface elevation (h)	[km]	0	3
Solar Zenith Angle (SZA)	[deg]	0	70
Relative Azimuth Angle (RAA)	[deg]	0	180

dataset of m=10000 samples generated with LHS distribution and the same input variables and ranges as in Table I. All the datasets used for analysis in this work are accessible from https://doi.org/10.5281/zenodo.7826005.

The generation of training and reference datasets was carried out with the Atmospheric Look-up table Generator (ALG) toolbox [42]. ALG is a software tool that generates datasets of atmospheric transfer functions from a collection of atmospheric RTMs. ALG v3.2 includes all the algorithms described in Section II to automate emulator configuration, training, and validation. Each GP model within an emulator was trained using Matlab's `fitrgp` function [26], and the resulting hyperparameters (σ^2, β and $\boldsymbol{\theta}$) were stored within ALG's emulator object. The ALG tool is freely downloadable at www.artmotoolbox.com. The emulator function is accessible from https://github.com/jorviser/AlgEmulator for standalone use.

The processing time to simulate 1000 samples took 3 hours on a personal computer with the following characteristics: Windows 10 64-bit OS, i7-4710 CPU 2.50 GHz, 16 GB RAM, and using nine parallel executions out of 12 CPUs. The same computer was used for the application scenario described in Section V.

B. Assessment methodology

The simulated atmospheric transfer functions in the *reference* dataset were combined with a vegetation Lambertian surface reflectance, $\boldsymbol{\rho}$, based on Eq. (5) resulting in m TOA radiance spectra, $\{\mathbf{L}\}_{i=1}^{m}$:

$$\mathbf{L}_i = \mathbf{L}_{0,i} + \frac{(\mathbf{E}_{dir,i}\mu_{il,i} + \mathbf{E}_{dif,i})(\mathbf{T}_{dir,i} + \mathbf{T}_{dif,i})\boldsymbol{\rho}}{\pi(1 - \mathbf{S}_i\boldsymbol{\rho})}, \tag{5}$$

where $\mu_{il,i}$ is the cosine of the solar zenith angle (SZA). The sub-index i identifies the i^{th} sample of the *reference* dataset with its input conditions $\mathbf{x}_i$ and spectral outputs, e.g. $\mathbf{T}_{dir,i} \equiv \mathbf{T}_{dir}(\mathbf{x}_i)$. Each trained emulator was used to predict the atmospheric transfer functions at the m input conditions of the *reference* dataset, which were used to derive the surface reflectance by inversion of (5). This produced a dataset of m inverted surface reflectance spectra $\{\widehat{\boldsymbol{\rho}}\}_{i=1}^{m}$ for which we calculated the relative difference against the reference surface reflectance spectrum $\boldsymbol{\rho}$ at every wavelength, i.e., $\boldsymbol{\varepsilon}_i = (\widehat{\boldsymbol{\rho}}_i - \boldsymbol{\rho}_i)/\boldsymbol{\rho}_i$ with i=1 to m. We then calculated the mean relative error (MRE) over the entire *reference* dataset, i.e., MRE $= \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{\varepsilon}_i$ to compare all emulators graphically. We also calculated the spectrally-averaged MRE (MRE$_\lambda$), excluding wavelengths in the deep H2O absorption (1325-1500 nm and 1750-1950 nm), as well as we kept track of the relative error histogram at every wavelength and the emulator execution time of the m samples for further analysis.

We first assessed the multi-fidelity emulation accuracy as a function of the number of layers (n_l=0 to 5) and PCA components (n_c=3 to 15). In this first assessment, the emulators were trained with a dataset of 500 samples as a compromise between accuracy and computation time. The spectral MRE is plotted for all emulators with varying values of n_c and n_l. The MRE$_\lambda$ and execution time as a function of the number of PCA components and multi-fidelity layers are presented as a 2D image to compare all multi-fidelity emulators. Their product is shown in a bar chart to identify the best-balanced multi-fidelity emulator.

Second, we investigated the role of the *training* dataset size in the accuracy of emulators by increasing the number of training samples (n=100, 500, and 1000) to train multi-fidelity emulators. Here, we fixed the number of PCA components to n_c=5 and varied the number of layers (n_l=0 to 5). We plotted the spectral MRE of all emulators with varying values of n_l with one figure for each training dataset size. The MRE$_\lambda$ and emulator execution time for each emulator is presented as a bar chart as a function of the training dataset size. Based on the previous analysis, we identified the best emulator in terms of balance between accuracy and

computation time, and we plotted the histogram of spectral relative errors.

IV. Results

Results are organized as follows. First, we present the performance assessment results of various multi-fidelity emulators, studying the impact of key configuration parameters for predicting spectral atmospheric transfer functions. Then, as proof-of-concept, we applied multi-fidelity emulators for the atmospheric correction PRISMA hyperspectral satellite data.

A. Performance assessment

The performance of the multi-fidelity GP emulators is first analyzed as a function of the number of PCA components (n_c) and multi-fidelity layers (n_l) using a training database of 500 samples. Their accuracy is compared by plotting the spectral mean relative errors (MRE) (Fig. 1), only showing three values of n_c (3, 7, and 15) to illustrate extreme and intermediate configurations. Generally, all results show similar spectral behavior of the relative errors due to several reasons. First, the errors increase inside the deep H_2O and O_2 absorption bands due to divisions by nearly zero during the inversion of surface reflectance from Eq. (5). Second, the nature of the relative error metric implies that wavelengths with lower reflectance values (e.g., 400-690 nm and $>$1500 nm for vegetation) have higher error values. Third, the error values increase towards shorter wavelengths ($<$500 nm) where the aerosol scattering is more relevant and thus where the emulators have to predict the spectral outputs for a larger number of sensitive input parameters [39]. The spectral error plots also show the effect of a poor spectral reconstruction with a low number of PCA components (see higher errors in the O_3 region at $\sim$550 nm with 0 multi-fidelity layers). These results indicate that, in agreement with the results presented in [20], an increasing number of PCA components reduce the errors in the predicted spectral data. This is particularly observed in the case of the simple (i.e., no multi-fidelity, n_l=0) GP emulator, where the errors are reduced by a factor $\sim$5 when passing from three PCA components to 15 components. The higher errors associated with a low number of PCA components (n_c=3) are somewhat compensated by adding extra layers in the multi-fidelity GP emulators (see top plot). This improvement seems to reach a saturation limit after n_l=3 layers. However, this lower limit in the MRE values is achieved with fewer fidelity layers when adding more PCA components. In the extreme case of n_c=15, an emulator of only one fidelity layer obtains the lowest MRE values. This error is still lower than with a simple GP emulator in the 400-1100 nm spectral

range. It is also observed that the addition of new fidelity layers reduces the values of MRE differently depending on the spectral range and the number of PCA components. For instance, with n_c=7, passing from n_l=1 to n_l=2 only reduces the MRE for wavelengths above 1000 nm.

Fig. 2 shows the MRE$_\lambda$ values for various combinations of n_c and n_l, allowing us to have an overview of the accuracy achieved by all emulators. We can observe how an increasing number of PCA components and multi-fidelity layers reduce prediction errors. However, adding extra layers in a multi-fidelity emulator seems to reduce faster the MRE$_\lambda$ values more than adding additional PCA components. For example, we note that an MRE$_\lambda$=0.9% is achieved with an emulator with configuration n_c=3 and n_l=1 or with an emulator with n_c=7 and n_l=0. We also note that the lowest value of MRE$_\lambda$=0.4% is achieved with three multi-fidelity layers regardless of the number of PCA components. This error value can also be achieved with as few as 5 PCA components and three layers or 7 PCA components and two layers.

Similarly, Fig. 3 presents the runtime spent by all emulators to predict the m=10000 points in the *reference* dataset. The figure indicates that the runtime varies linearly as a function of n_c and n_l. As expected, the fastest emulator is also the simplest (n_c=3 and n_l=0) and calculates the prediction in 1.5 s. On the contrary, the slowest emulator is also the most complex (n_c=15 and n_l=5) with a runtime of 18 s. However, a balanced emulator consisting of 5 PCA components and three multi-fidelity layers has a runtime of 5 s and achieves the same accuracy (MRE$_\lambda$=0.4%) as the most complex emulator.

Since the combination of the number of PCA components and multi-fidelity layers can compensate one another and reach similar accuracy and performance, we plot in Fig. 4 a bar chart with the product of runtime and MRE$_\lambda$. Indeed, we seek an emulator that obtains the lowest errors with a competitive runtime. This figure shows that the most balanced emulators are achieved with n_l=1 multi-fidelity layers regardless of the number of PCA components. Among them, the best balanced emulator is achieved with 5 PCA components and one multi-fidelity layer, which results in an MRE$_\lambda$=0.54% and a prediction time of 1.86 s. To achieve the same accuracy-time performance as the fastest emulator (n_c=3 and n_l=0), an emulator with n_c=3 and n_l=4 (or with n_c=7 and n_l=3) should be considered. That is, the gain in accuracy (1.8% to 0.4%) compensates for the increase in runtime (1.5 s to 6.3 s),

Based on these findings, we fixed the number of PCA components to n_c=5 and studied the impact of training dataset size on accuracy and performance. Fig. 5 shows the spectral MRE results for n=100 training samples (top), n=500 (middle), and n=1000 (bottom). We observe

how the MRE values are reduced in all emulator configurations when increasing the size of the training dataset. However, this reduction is driven by the number of fidelity layers. With n=100 training samples, the simple GP emulator (i.e., n_l=0) obtains the lowest MRE compared to any multi-fidelity emulator. The increase of the training dataset size to n=500 reduces the MRE results of this simple emulator by nearly a factor of 10 in the visible spectral range without major improvements at longer wavelengths (>1500 nm). Further increase in the training dataset size (n=1000) does not improve the emulator accuracy. This situation is reversed with emulators of one or more layers after 500 training samples. In these configurations, the MRE values are lower than with n_l=0 for all wavelengths. When increasing n to 1000, no further improvements are observed at wavelengths above 1500 nm in the case of n_l=1. Yet, adding extra layers allows the emulators to reduce error values further by exploiting the data from larger datasets.

Regarding performance, the runtime for predicting 10000 samples is summarised in Tab. II. We observe that the emulators with none or n_l=1 multi-fidelity layers have the lowest runtime (1.8 s). The runtime increases by a factor of 1.3-1.6 when passing from 100 to 500 training samples and an extra factor ∼2 for 1000 samples. Third, adding extra layers in the emulator increase the runtime linearly, as seen in Fig. 3.

TABLE II
RUNTIME (IN S) FOR THE PREDICTION OF m=10000 SAMPLES USING A TRAINING DATASET OF n=100, 500, AND 1000 SAMPLES (ROWS) AND VARIOUS MULTI-FIDELITY LAYERS (COLUMNS).

Layers (n_l):	**0**	**1**	**2**	**3**	**4**	**5**
100	2.2	1.9	2.7	3.7	4.3	5.3
500	1.8	**1.8**	3.5	5.3	7.2	8.7
1000	3.7	4.1	6.4	9.3	13.5	17.6

From these results, we selected the best-balanced emulator in terms of mean accuracy vs runtime (i.e., 5 PCA components, one layer, and 500 training samples), and analysed the relative error histogram (see Fig. 6). The histogram shows higher errors in the 400-500 nm spectral range due higher impact of aerosol scattering and the low reflectance values of the reference spectrum. After 600 nm, except for absorption regions, the errors increase exponentially (linearly in log-scale) for higher wavelengths. The highest relative error values (95% percentile) are typically below 2% outside of deep absorption regions and up to 7% in the O_2-A and 4% in the H_2O band at ∼820 nm. On average, the errors are reduced to 0.2-0.7% outside of absorption regions and down to 2% in the O_2-A absorption. For 10% of the reference conditions, the errors can be

as low as $\sim$0.01%.

V. Application example: atmospheric correction

To test the utility of emulators in practical applications, we applied the best emulators to the atmospheric correction of PRISMA Level-1 data (see Fig. 7). PRISMA is an imaging spectrometer satellite mission implemented by the Italian Space Agency (ASI). The instrument acquires the TOA radiance in the 400-2500 nm spectral range with nearly 240 spectral bands of $\sim$10 nm resolution. The Level-1 product has a size of 30$\times$30 km^2 with a spatial resolution of 30 m, resulting in an image of 1000$\times$1000 pixels. More details about the PRISMA mission are available in ASI's dedicated website.

The selected PRISMA product corresponds to the desert of Gobabeb, in Namibia (23°36′S; 15°07′E), acquired on 05/10/2022 at 9:15 with a viewing zenith angle of 0.8°, SZA=28.5° and RAA=75°. The area is characterized by a smoothly varying topography (0.4-0.7 km) and a large spatially homogeneous surface with small temporal and spatial variability of bare soil reflectance. A RadCalNet station in Gobabeb is widely used for radiometric calibration monitoring of satellite instruments [43]. As for the atmospheric conditions, the ozone (0.285 atm-cm) is obtained from CAMS re-analysis data after spatial/temporal interpolation to PRISMA acquisition. The per-pixel values of water vapor (0.7-1 g·cm^{-2}), AOT (0.1), and Angstrom exponent (1.9-2.2) are directly extracted from PRISMA Level-2C (L2C) product. For the remaining aerosol parameters (asymmetry factor and SSA), we assigned the spectrally-averaged values from MACv2 climatology [44] spatially and temporally interpolated at PRISMA observations, i.e. g=0.61 and SSA= 0.91. The atmospheric correction was performed by inverting the surface reflectance from (5). The the best-balanced GP emulator was run to predict atmospheric transfer functions at high spectral resolution using the input geometric and atmospheric conditions at every pixel. These spectra were then convolved by PRISMA's spectral response function, which was approximated with a Gaussian model.

The 10^6 image pixels were processed in less than 4 min using batches of 100000 pixels to avoid RAM saturation. Fig. 8 shows the statistics of surface reflectance values from the PRISMA L2C product (blue) and the inversion with the selected multi-fidelity emulator (red). These surface reflectance statistics are presented with the mean values (lines) and the 10$^{\text{th}}$ and 90$^{\text{th}}$ percentiles (shaded areas). The figure indicates good agreement on the inverted reflectance with nearly overlapping means and percentile values. However, some discrepancies are visible

in the deep H_2O bands (1410 nm and 1850 nm), where the PRISMA L2C reflectance product is strictly zero, and in the spectral range above 2250 nm, where the PRISMA L2C reflectance product is affected by a residual H_2O absorption.

VI. Discussion

Operational atmospheric correction of satellite data relies on computationally efficient yet accurate algorithms to approximate atmospheric RTM output spectral data. Emulators are statistical regression methods used to approximate the relation between input states and output spectral data of a deterministic RTM. They offer an alternative, and quite an advantageous approach, to traditional look-up table interpolation [19]. In this study, we explored multi-fidelity methods to improve the accuracy and performance of emulators. Multi-fidelity essentially places a recursive set of emulators in which one (higher fidelity) layer approximates the prediction residuals by a previous (lower fidelity) layer. We proposed building on GP models and studied the impact on the solution by different configuration parameters: (1) the number of fidelity layers, (2) the number of components after dimensionality reduction, and (3) the training dataset size. In this study, we first investigated the impact of dimensionality reduction and multi-fidelity on the accuracy and runtime of a GP emulator. We found that multi-fidelity compensates for residual errors caused by dimensionality reduction and improves prediction accuracy beyond what is achievable with a simple (i.e., without multi-fidelity) GP emulator. The number of PCA components and fidelities compensate for each other and achieve similar accuracies with a minimum prediction error of MRE_λ=0.4%. This lower error bound is achieved after a maximum of three fidelity layers, independently of the number of PCA components, and as little as one fidelity layer when having a sufficiently large number of PCA components ($n_c \geq$11). The addition of extra fidelity layers does not improve the accuracy. Indeed, for every extra layer, the gain in accuracy is lower since the residuals to be predicted contain less signal characteristics and are harder to predict. The gain in accuracy by a multi-fidelity emulator is more pronounced at shorter wavelengths ($<$1100 nm) due to the richer gaseous absorption features and higher sensitivity to aerosol parameters. For generic applications, we identified the best-balanced emulator consisting of one layer and five PCA components, achieving an MRE_λ=0.56% and runtime of 1.86 s for predicting 10000 new samples. Note that an emulator with zero layers has the same number of GP models as with one layer since the lowest fidelity layer is a regression by polynomial surface fitting. We

note that, depending on the application, the emulator can be optimized to achieve the required accuracy at the expense of runtime.

Second, we explored the impact of training dataset size on emulator accuracy. Our prior research [20] showed that GP emulator models reach accuracy limits with training datasets of 500-1000 samples since adding new samples does not provide additional statistical information for predicting smooth atmospheric transfer functions. However, incorporating the multi-fidelity technique can overcome this limit and reduce prediction errors by approximately 50% when doubling the number of training samples. Multi-fidelity models leverage the statistical information in new samples through the calculated residuals between the previous-layer predictions and the training dataset. Nevertheless, doubling the training samples from 500 to 1000 doubles the runtime. Hence, finding an appropriate training dataset size that satisfies both accuracy and computational requirements for operational use is necessary. We also investigated the possibility of reducing the training dataset size to improve the runtime without sacrificing accuracy. However, using only 100 samples for training led to higher prediction errors in the multi-fidelity emulator compared to the simple GP emulator. This is because the multi-fidelity emulator is constructed with multiple layers that refine the initial prediction from a 2$^{\text{nd}}$ degree polynomial surface fitting (lowest fidelity layer). To fit this polynomial regressor, for example, for a hyper-surface in d=9 dimensions, at least n=$1 + d + d(d + 1)/2$=55 training samples are needed. With this minimum size, the fitted surface will pass precisely through the training points, resulting in prediction residuals of zero by definition. Consequently, the GP emulator on the first (and subsequent) layers will always predict values equal to zero for all query points. Thus, the accuracy of a multi-fidelity emulator trained with this minimum set of training samples will be limited by the accuracy of the polynomial interpolation. This indicates that a minimum training dataset of 100 to 500 samples is needed to train an accurate multi-fidelity emulator. Regarding the sampling of input variables in the training dataset, we adopted the LHS method due to its simplicity and accuracy [39]. However, active learning methods rooted in the Bayesian optimization framework can enhance the performance and accuracy of a multi-fidelity emulator [45]. Active learning identifies regions in the input variable space where the residual errors between fidelities are higher and generate an optimal set of training samples to reduce both the prediction error and the runtime.

Another potential improvement of the proposed multi-fidelity emulation method is regarding the regressor in the lowest fidelity layer. In this work, we used a polynomial surface fitting

due to its speed and simplicity. Yet, alternative options based on atmospheric RTMs have been proposed in similar studies [21], [25]. While these alternative options use a physically-based model for the lowest fidelity layer that contributes to the explainability of the emulator, they are computationally less efficient and require more complex implementation. Semi-empirical atmospheric RTMs (e.g., [46]) could be a suitable option for physics-aware emulation with less complex implementation and a fast runtime.

We finally applied the multi-fidelity emulation technique to invert the surface reflectance from PRISMA L1 data, achieving less than 4 minutes of runtime for processing 1 million pixels. In the context of operational satellite data processing, the runtime is still slow to generate atmospherically corrected products within near-real-time mission requirements (e.g., [9]). Indeed, the maximum half-orbit length of an imaging data take over land (15000 km) implies that a speed-up factor of $\sim$20 is still needed. Profiling the code identified bottlenecks in inverse PCA (50 s), kernel matrix calculation (45 s), and polynomial interpolation (30 s). The inverse PCA is applied for each batch of 10000 query points to recover the six atmospheric transfer functions (each with 4200 wavelengths). This results in large matrix multiplication, suggesting that its implementation in GPUs could significantly improve runtime. The kernel matrix calculation of $n\times$10000 elements (n=500) is repeated in a loop for each transfer function, n_c=5 PCA components, and n_l=1 layer. Accordingly, the runtime of the multi-fidelity emulator could also be reduced by optimizing the number of PCA components on each atmospheric transfer function. Furthermore, atmospheric correction in macro-pixels (image areas with nearly constant atmospheric and geometric conditions) could exponentially reduce the number of emulator predictions and runtime. For example, a macro-pixel of 3$\times$3 pixels would reduce the runtime to 25 s. Regarding surface reflectance, the emulator processor generally agreed well with the PRISMA L2C product [47]. This was expected since the emulators used the same atmospheric RTM (MODTRAN) and atmospheric conditions as the PRISMA L2 processor. However, there were minor differences below 2250 nm, possibly due to differences in the aerosol modeling (optical properties in the emulator, rural aerosol in PRISMA L2C) and the digital elevation model. Above 2250 nm, a significant difference was observed, indicating an incorrect H_2O absorption correction in the PRISMA Level-2 processor. We note that the PRISMA L2 processor might not have accurately retrieved the atmospheric parameters in the L2C product. Indeed, the CAMS re-analysis product (CWV=1 g·cm^{-2}, AOT=0.22 and α=1.64) and RadCalNet station (CWV=1.1$\pm$0.2 g·cm^{-2}, AOT=0.248$\pm$0.002 and α=1.24$\pm$0.02) provide values that disagree with

the PRISMA L2C product. Despite this, using the PRISMA L2C atmospheric products allowed a direct comparison between the L2C surface reflectance product and our inversion based on emulation. Operationally, the proposed emulator methodology faces similar challenges to the traditional LUT interpolation approach in terms of requiring a training dataset that adequately represents the physical processes and natural variability encountered by satellite observations. The presented emulation method provides significant advantages in terms of uncertainty propagation in an atmospheric correction algorithm. From Eq. (4), the mathematical framework of GP regression enables the incorporation of uncertainties in the input TOA radiance as well as uncertainties in the input atmospheric and geometric parameters.

VII. Conclusions

In this study, we demonstrated the effectiveness of multi-fidelity methods in improving the accuracy and performance of atmospheric RTM emulators for satellite data atmospheric correction. We examined several configuration parameters, such as the number of fidelity layers, the dimensionality reduction, and the training dataset size, and how they affect the emulator's accuracy and performance. Our results reveal that multi-fidelity compensates for residual errors caused by dimensionality reduction and significantly improves prediction accuracy beyond what is achievable with a simple Gaussian process (GP) emulation. This is particularly true for shorter wavelengths, where multi-fidelity methods better reproduce the richer gaseous absorption features and higher sensitivity to aerosols. The study also showed that multi-fidelity emulators can overcome the accuracy limit of a simple GP emulator by enhancing the statistical information contained in larger training datasets. This allows emulators to reduce prediction errors by approximately 50% when doubling the number of training samples but at expense of doubling the runtime. We suggested an emulator consisting of one fidelity layer, five PCA components, and 500 training samples, which achieves a minimum prediction error of 0.56% and a runtime of 1.86 seconds for predicting 10000 new samples. This emulator was successfully applied to invert the surface reflectance from PRISMA L1 data, achieving less than 4 minutes of runtime for processing 1 million pixels and a good alignment with the PRISMA L2C product. These results show that multi-fidelity methods improve the accuracy and runtime of GP emulators and can be used in the atmospheric correction of satellite instruments with competitive accuracy and computational efficiency. While we presented this methodology for hyperspectral satellite data processing, it can also be applied to multispectral instruments, where dimensionality reduction is

no longer required. Finally, we proposed several strategies to further improve the accuracy and speed of atmospheric RTM emulators. These strategies include the use of physics-informed models, active learning techniques, and optimized implementation of emulators within atmospheric correction algorithms. We expect that implementing these strategies will improve the accuracy and performance of atmospheric RTM emulators, enabling their widespread use in operational satellite data processing.

REFERENCES

[1] D. Ramon, F. Steinmetz, D. Jolivet, M. Compiègne, and R. Frouin, "Modeling Polarized Radiative Transfer in the Ocean-Atmosphere System with the GPU-Accelerated SMART-G Monte Carlo Code," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 222-223, pp. 89 – 107, 2019.

[2] R. Saunders, J. Hocking, E. Turner, P. Rayer, D. Rundle, P. Brunel, J. Vidot, P. Roquet, M. Matricardi, A. Geer, N. Bormann, and C. Lupu, "An update on the RTTOV fast radiative transfer model (currently at version 12)," *Geoscientific Model Development*, vol. 11, no. 7, pp. 2717–2737, 2018.

[3] E. Vermote, D. Tanré, J. Deuzé, M. Herman, and J.-J. Morcrette, "Second simulation of the satellite signal in the solar spectrum, 6S: an overview," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, no. 3, pp. 675–686, 1997.

[4] C. Emde, R. Buras-Schnell, A. Kylling, B. Mayer, J. Gasteiger, U. Hamann, J. Kylling, B. Richter, C. Pause, T. Dowling, and L. Bugliaro, "The libradtran software package for radiative transfer calculations (version 2.0.1)," *Geoscientific Model Development*, vol. 9, no. 5, pp. 1647–1672, 2016.

[5] A. Berk, P. Conforti, R. Kennett, T. Perkins, F. Hawes, and J. Van Den Bosch, "MODTRAN6: A major upgrade of the MODTRAN radiative transfer code," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 9088, 2014.

[6] L. Guanter, R. Richter, and H. Kaufmann, "On the application of the MODTRAN4 atmospheric radiative transfer code to optical remote sensing," *International Journal of Remote Sensing*, vol. 30, no. 6, pp. 1407–1424, 2009.

[7] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, 1996.

[8] B. D. Bue, D. R. Thompson, S. Deshpande, M. Eastwood, R. O. Green, V. Natraj, T. Mullen, and M. Parente, "Neural network radiative transfer for imaging spectroscopy," *Atmospheric Measurement Techniques*, vol. 12, no. 4, pp. 2567–2578, 2019.

[9] ESA (2021), "Copernicus Hyperspectral Imaging Mission for the Environment - Mission Requirements Document," ESA ESA-EOPSM-CHIM-MRD-3216 (v3.0), European Space Agency, Noordwijk, The Netherlands.

[10] R. O. Green, D. R. Thompson, and E. Team, "NASA's Earth Surface Mineral Dust Source Investigation: An Earth Venture Imaging Spectrometer Science Mission," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021, pp. 119–122.

[11] L. Guanter, H. Kaufmann, K. Segl, S. Foerster, C. Rogass, S. Chabrillat, T. Kuester, A. Hollstein, G. Rossner, C. Chlebek, C. Straif, S. Fischer, S. Schrader, T. Storch, U. Heiden, A. Mueller, M. Bachmann, H. Muhle, R. Muller, M. Habermeyer, A. Ohndorf, J. Hill, H. Buddenbaum, P. Hostert, S. van der Linden, P. J. Leitao, A. Rabe, R. Doerffer, H. Krasemann, H. Xi, W. Mauser, T. Hank, M. Locherer, M. Rast, K. Staenz, and B. Sang, "The EnMAP Spaceborne Imaging Spectroscopy Mission for Earth Observation," *Remote Sensing*, vol. 7, no. 7, p. 8830, 2015.

[12] ESA (2015), "Report for Mission Selection: FLEX," ESA SP-1330/2 (2 volume series), European Space Agency, Noordwijk, The Netherlands.

[13] S. Cogliati, F. Sarti, L. Chiarantini, M. Cosi, R. Lorusso, E. Lopinto, F. Miglietta, L. Genesio, L. Guanter, A. Damm, S. Pérez-López, D. Scheffler, G. Tagliabue, C. Panigada, U. Rascher, T. Dowling, C. Giardino, and R. Colombo, "The prisma imaging spectroscopy mission: overview and first performance analysis," *Remote Sensing of Environment*, vol. 262, p. 112499, 2021.

[14] R. O. Green, "Global vswir imaging spectroscopy and the 2017 decadal survey," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 2018, pp. 183–185.

[15] M. Main-Knorn, B. Pflug, J. Louis, V. Debaecker, U. Müller-Wilm, and F. Gascon, "Sen2Cor for Sentinel-2," in *Image and Signal Processing for Remote Sensing XXIII*, vol. 10427. SPIE, 10 2017, p. 3.

[16] D. R. Thompson, V. Natraj, R. O. Green, M. C. Helmlinger, B.-C. Gao, and M. L. Eastwood, "Optimal estimation for imaging spectrometer atmospheric correction," *Remote Sensing of Environment*, vol. 216, pp. 355 – 373, 2018.

[17] A. O'Hagan, "Bayesian analysis of computer code outputs: A tutorial," *Reliability Engineering and System Safety*, vol. 91, no. 10-11, pp. 1290–1300, 2006.

[18] J. L. Gómez-Dans, P. E. Lewis, and M. Disney, "Efficient emulation of radiative transfer codes using gaussian processes and application to land surface parameter inferences," *Remote Sensing*, vol. 8, no. 2, p. 119, 2016.

[19] J. Vicent, J. Verrelst, J. Rivera Caicedo, N. Sabater Medina, J. Muñoz, G. Camps-Valls, and J. Moreno, "Emulation as an accurate alternative to interpolation in sampling radiative transfer codes," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 12, pp. 1–14, 10 2018.

[20] J. Vicent Servera, J. P. Rivera-Caicedo, J. Verrelst, J. Muñoz-Marí, N. Sabater, B. Berthelot, G. Camps-Valls, and J. Moreno, "Systematic Assessment of MODTRAN Emulators for Atmospheric Correction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2022.

[21] P. G. Brodrick, D. R. Thompson, J. E. Fahlen, M. L. Eastwood, C. M. Sarture, S. R. Lundeen, W. Olson-Duvall, N. Carmon, and R. O. Green, "Generalized radiative transfer emulation for imaging spectroscopy reflectance retrievals," *Remote Sensing of Environment*, vol. 261, p. 112476, 2021.

[22] M. Kennedy and A. O'Hagan, "Predicting the output from a complex computer code when fast approximations are available," *Biometrika*, vol. 87, no. 1, pp. 1–13, 2000.

[23] K. Cutajar, M. Pullin, A. Damianou, N. Lawrence, and J. González, "Deep Gaussian Processes for Multi-fidelity Modeling," 2019.

[24] A. Berk and F. Hawes, "Validation of MODTRAN6 and its line-by-line algorithm," *Journal of Quantitative Spectroscocy and Radiative Transfer*, vol. 203, pp. 542–556, 2017.

[25] A. del Águila and D. S. Efremenko, "Fast hyper-spectral radiative transfer model based on the double cluster low-streams regression method," *Remote Sensing*, vol. 13, no. 3, 2021.

[26] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. New York: The MIT Press, 2006.

[27] G. Camps-Valls and L. Bruzzone, *Kernel methods for Remote Sensing Data Analysis*, G. Camps-Valls and L. Bruzzone, Eds. UK: Wiley & Sons, Dec 2009.

[28] G. Camps-Valls, D. Sejdinovic, J. Runge, and M. Reichstein, "A perspective on gaussian processes for earth observation," *National Science Review*, vol. 6, no. 4, pp. 616–618, Mar 2019.

[29] G. Camps-Valls, J. Verrelst, J. Muñoz-Marí, V. Laparra, F. Mateo-Jiménez, and J. Gómez-Dans, "A survey on Gaussian processes for earth observation data analysis," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, 2016.

[30] L. Martino and J. Read, "A joint introduction to Gaussian Processes and Relevance Vector Machines with connections to Kalman filtering and other kernel smoothers," *Information Fusion*, vol. 74, pp. 17–38, 2021.

[31] F. Llorente, L. Martino, D. Delgado-Gomez, and G. Camps-Valls, "Deep importance sampling based on regression for model inversion and emulation," *Digital Signal Processing*, vol. 116, p. 103104, 2021.

[32] R. M. Neal, *Bayesian Learning for Neural Networks*, 1st ed., ser. Lecture Notes in Statistics. New York, NY: Springer, 1996, vol. 118.

[33] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55–63, 1968.

[34] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[35] X. Liu, W. L. Smith, D. K. Zhou, and A. Larar, "Principal component-based radiative transfer model for hyperspectral sensors: theoretical concept," *Appl. Opt.*, vol. 45, no. 1, pp. 201–209, Jan 2006.

[36] M. Matricardi, "A principal component based version of the RTTOV fast radiative transfer model," *Quarterly Journal of the Royal Meteorological Society*, vol. 136, no. 652, pp. 1823–1835, 2010.

[37] D. Efremenko, A. Doicu, D. Loyola, and T. Trautmann, "Optical property dimensionality reduction techniques for accelerated radiative transfer performance: Application to remote sensing total ozone retrievals," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 133, pp. 128 – 135, 2014.

[38] M. McKay, R. Beckman, and W. Conover, "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[39] J. Verrelst, N. Sabater, J. P. Rivera, J. Muñoz Marí, J. Vicent, G. Camps-Valls, and J. Moreno, "Emulation of leaf, canopy and atmosphere radiative transfer models for fast global sensitivity analysis," *Remote Sensing*, vol. 8, no. 8, p. 673, 2016.

[40] S. Conti and A. O'Hagan, "Bayesian emulation of complex multi-output and dynamic computer models," *Journal of Statistical Planning and Inference*, vol. 140, no. 3, pp. 640 – 651, 2010.

[41] J. P. Rivera, J. Verrelst, J. Gómez-Dans, J. Muñoz Marí, J. Moreno, and G. Camps-Valls, "An emulator toolbox to approximate radiative transfer models with statistical learning," *Remote Sensing*, vol. 7, no. 7, p. 9347, 2015.

[42] J. Vicent, J. Verrelst, N. Sabater, L. Alonso, J. P. Rivera-Caicedo, L. Martino, J. Muñoz-Marí, and J. Moreno, "Comparative analysis of atmospheric radiative transfer models using the Atmospheric Look-up table Generator (ALG) toolbox (version 2.0)." *Geoscientific Model Development*, vol. 13, no. 4, 2020.

[43] M. Bouvet, K. Thome, B. Berthelot, A. Bialek, J. Czapla-Myers, N. P. Fox, P. Goryl, P. Henry, L. Ma, S. Marcq, A. Meygret, B. N. Wenny, and E. R. Woolliams, "RadCalNet: A Radiometric Calibration Network for Earth Observing Imagers Operating in the Visible to Shortwave Infrared Spectral Range," *Remote Sensing*, vol. 11, no. 20, 2019.

[44] S. Kinne, "The MACv2 aerosol climatology," *Tellus B: Chemical and Physical Meteorology*, vol. 71, no. 1, pp. 1–21, 2019.

[45] D. H. Svendsen, L. Martino, and G. Camps-Valls, "Active emulation of computer codes with Gaussian processes - Application to remote sensing," *Pattern Recognition*, vol. 100, p. 107103, 2020.

[46] F. Seidel, A. Kokhanovsky, and M. Schaepman, "Fast and simple model for atmospheric radiative transfer," *Atmospheric Measurement Techniques*, vol. 3, no. 4, pp. 1129–1141, 2010.

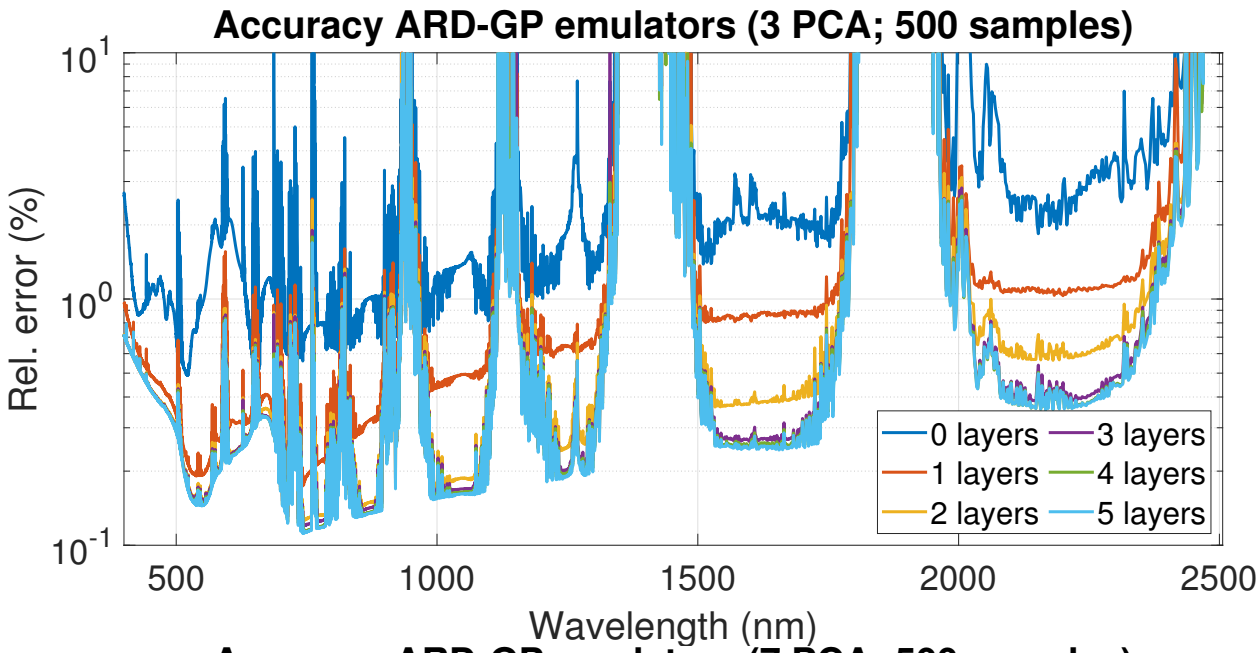
[47] ASI, "PRISMA Algortihm Theoretical Basis Document (ATBD)," 12 2021, v1.0.

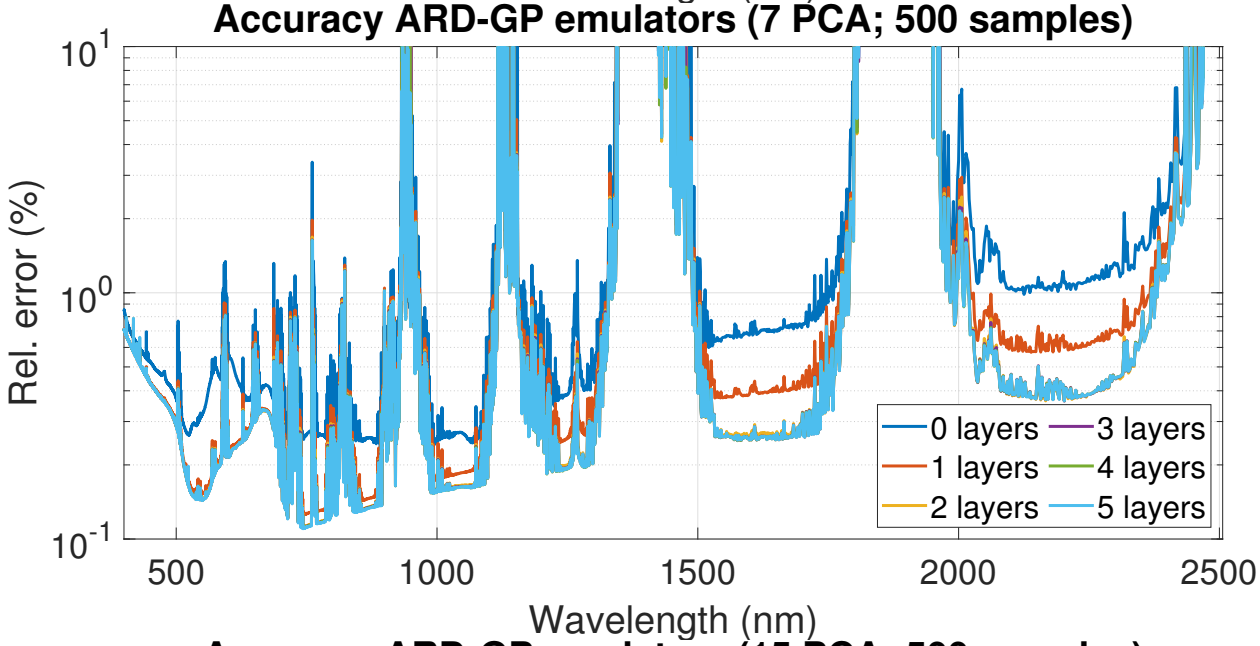
Jorge Vicent Servera received a B.Sc. degree in physics from the University of Valencia, Spain, in 2008; the M.Sc. in physics from the EPFL, Switzerland, in 2010; and the PhD. in remote sensing from the University of Valencia, Spain, in 2016. Since November 2017, he has been with Magellium in the Earth Observation department, Toulouse, France, as an R&D Engineer. He is currently developing the Level-2 processing chain for ESA's FLEX mission. His research interests include the modelling of Earth Observation satellites, radiative transfer modelling, simulation of synthetic scenes, atmospheric correction, and hyperspectral satellite data processing.

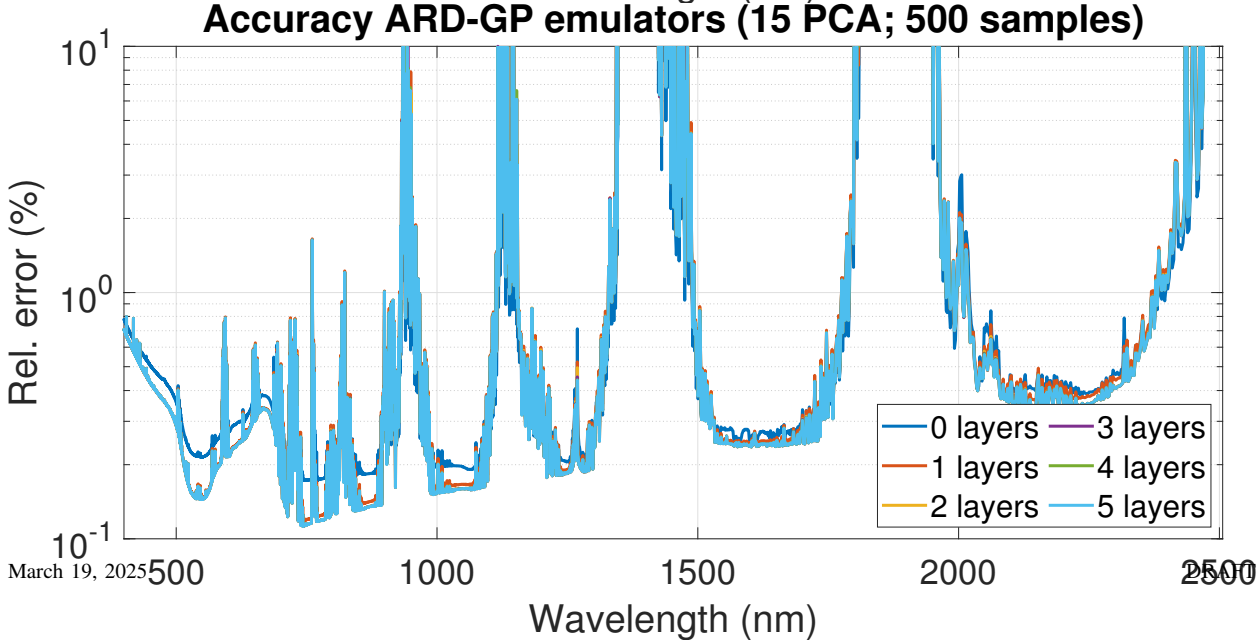
Luca Martino received the M.Sc. degree in electronic engineering from the Politecnico di Milano, Milan, Italy, and the PhD in statistical signal processing from the Universidad Carlos III de Madrid, Madrid, Spain, in 2011. He spent two years with the Department of Statistics, University of Helsinki, Helsinki, Finland. He carried out Post-Doctoral Research at the São Paulo Research Foundation (FAPESP), São Paulo, Brazil, and the University of Valencia, Valencia, Spain. Since January 2020, he has been an associate professor at the Higher Technical School of Telecommunications Engineering, Universidad Rey Juan Carlos, Madrid, Spain. His research interests include Monte Carlo methods (MCMC, particle filtering), general problems and methods for Regression-Filtering-Smoothing, Stochastic Processes (especially Gaussian processes), Multi-label Classification, and Nonlinear Chaotic Systems.

Jochem Verrelst received the M.Sc. degree in tropical land use and geo-information science in 2005 and the PhD in remote sensing in 2010 from Wageningen University, Wageningen, Netherlands. His dissertation focused on the space-borne spectro-directional estimation of forest properties. Since 2010, he has been involved in preparatory activities of FLEX. His research interests include retrieving vegetation properties using airborne and satellite data, canopy radiative transfer modelling and emulation, and hyperspectral data analysis. He is the founder of the ARTMO software package. In 2017 he received an H2020 ERC Starting Grant (#755617) to work on developing vegetation products based on the synergy of FLEX and Sentinel-3 data. See also https://ipl.uv.es/sentiflex/. He is also the co-chair of the SENSECO Cost Action (CA17134), which focuses on optical synergies for spatiotemporal sensing of scalable ecophysiological traits (https://www.senseco.eu/).

Gustau Camps-Valls is a Full professor in the Electrical Engineering Department and coordinator of the Image Signal Processing (ISP) group at the Universitat de València. His research relates to statistical learning for modelling and understanding the Earth and climate systems. More information in https://www.uv.es/gcamps/ and https://isp.uv.es.


Accuracy ARD-GP emulators (3 PCA; 500 samples)
Rel. error (%)
Wavelength (nm)
0 layers
1 layers
2 layers
3 layers
4 layers
5 layers


Accuracy ARD-GP emulators (7 PCA; 500 samples)
Rel. error (%)
Wavelength (nm)
0 layers
1 layers
2 layers
3 layers
4 layers
5 layers


Accuracy ARD-GP emulators (15 PCA; 500 samples)
Rel. error (%)
Wavelength (nm)
0 layers
1 layers
2 layers
3 layers
4 layers
5 layers

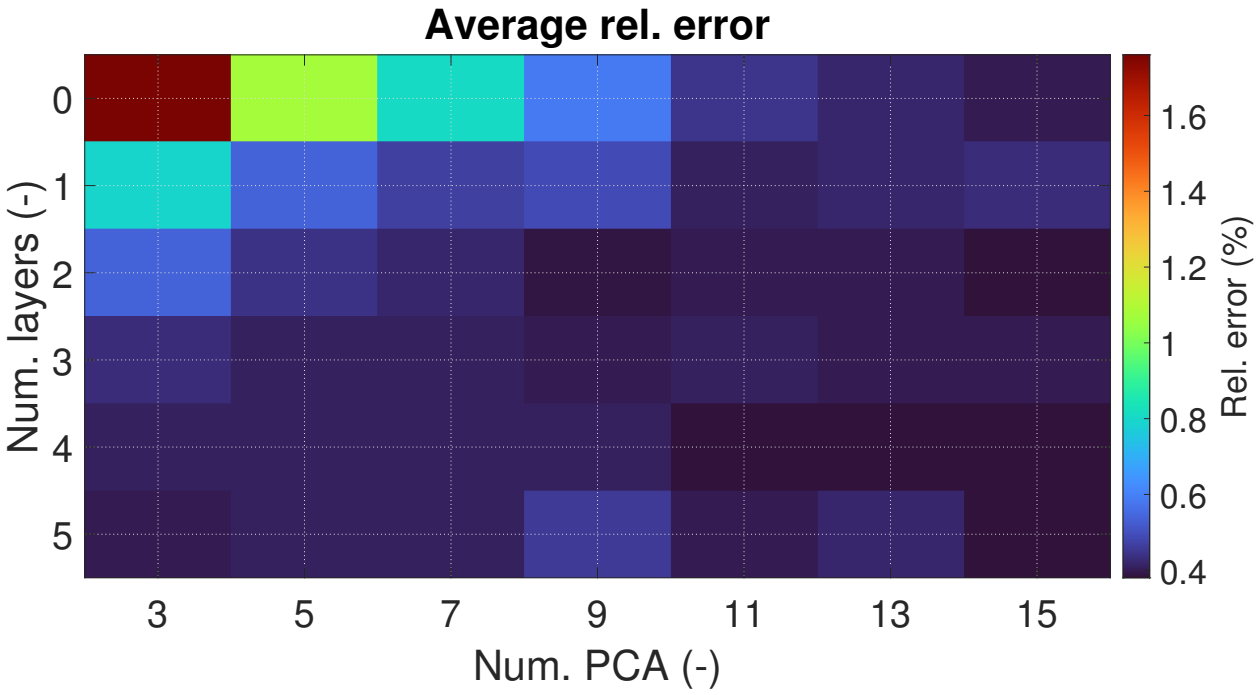


Fig. 2. Average mean relative error (MRE_λ, in %) as a function of the number of PCA components and multi-fidelity layers. Training dataset size: n=500.

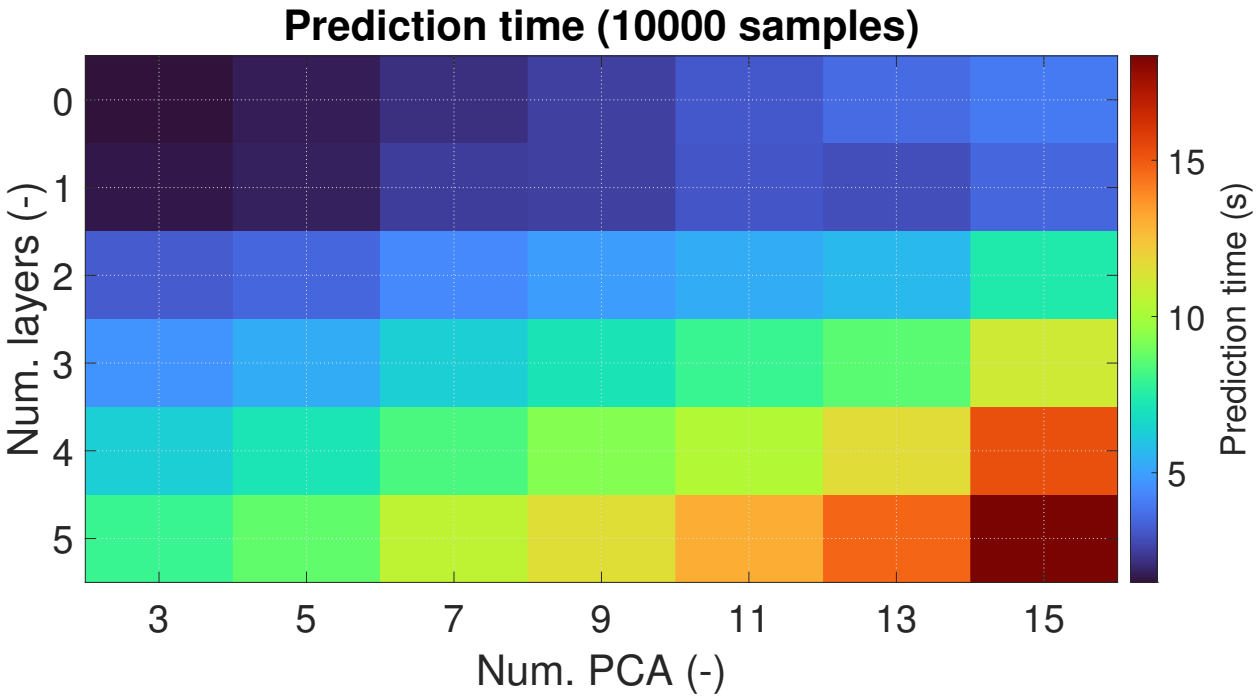


Fig. 3. Same as Fig. 2 but for prediction time (in s) for the m=10000 samples in the *reference* dataset.

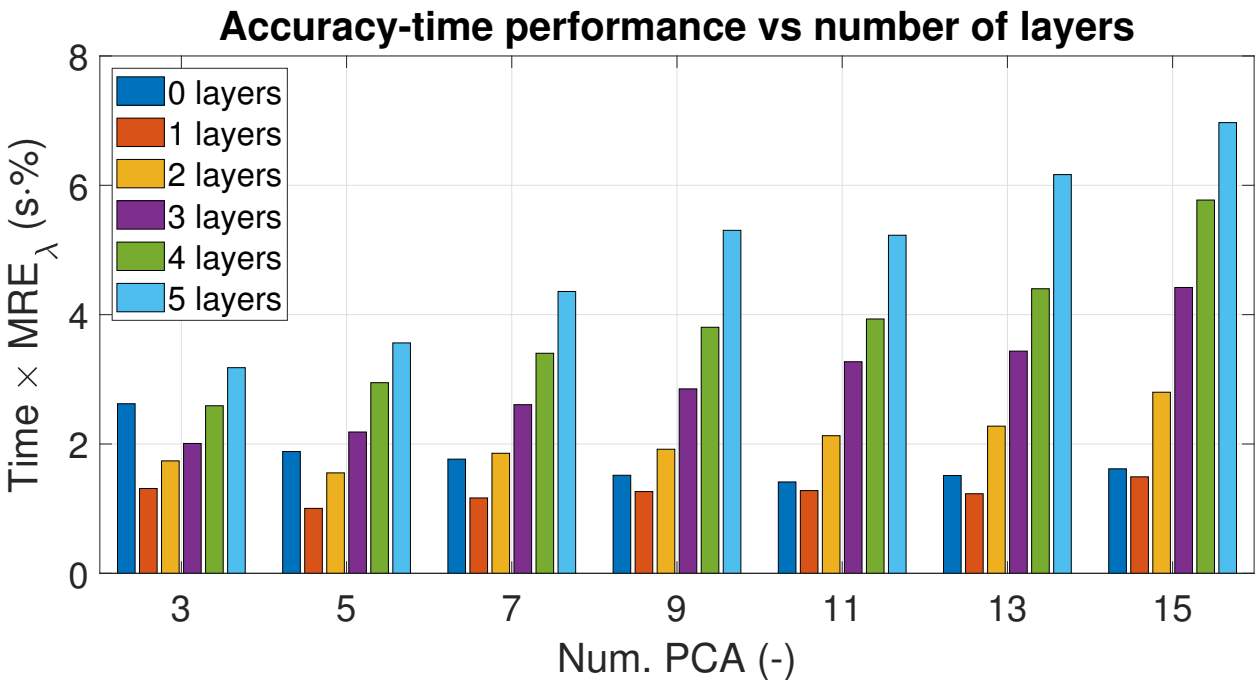
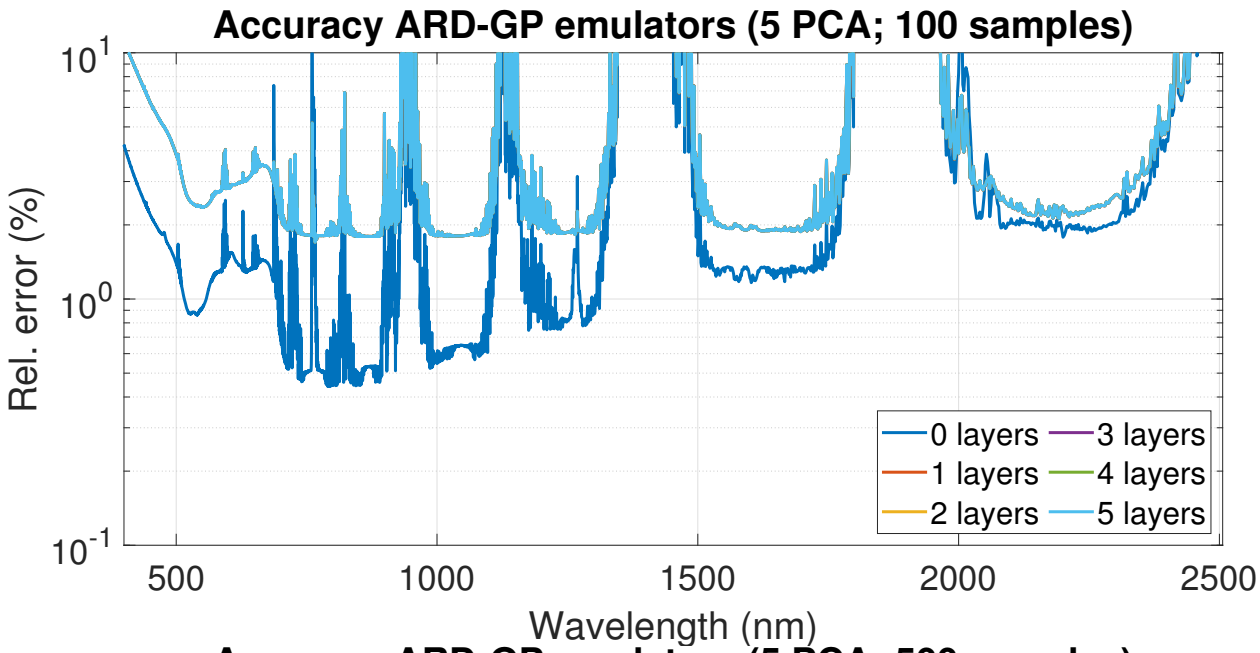
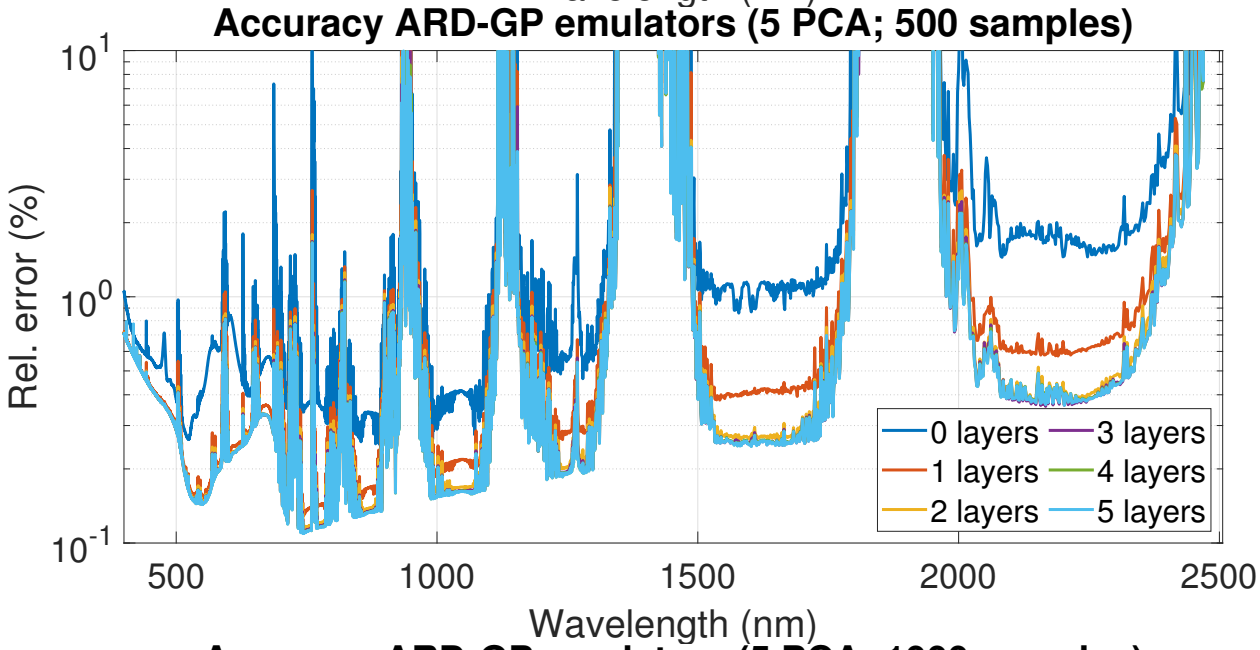
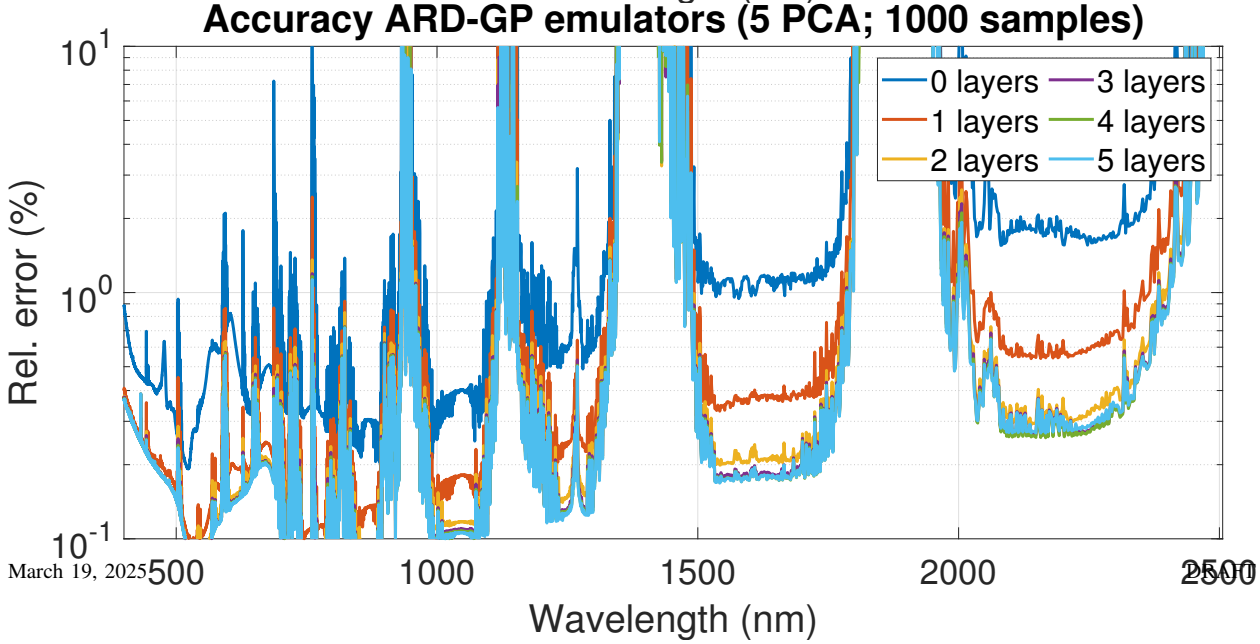


Fig. 4. Product of MRE_λ and prediction time as a function of the number of PCA components and multi-fidelity layers. Training dataset size: n=500.


Accuracy ARD-GP emulators (5 PCA; 100 samples)
Rel. error (%)
Wavelength (nm)
0 layers
1 layers
2 layers
3 layers
4 layers
5 layers


Accuracy ARD-GP emulators (5 PCA; 500 samples)
Rel. error (%)
Wavelength (nm)
0 layers
1 layers
2 layers
3 layers
4 layers
5 layers


Accuracy ARD-GP emulators (5 PCA; 1000 samples)
Rel. error (%)
Wavelength (nm)
0 layers
1 layers
2 layers
3 layers
4 layers
5 layers

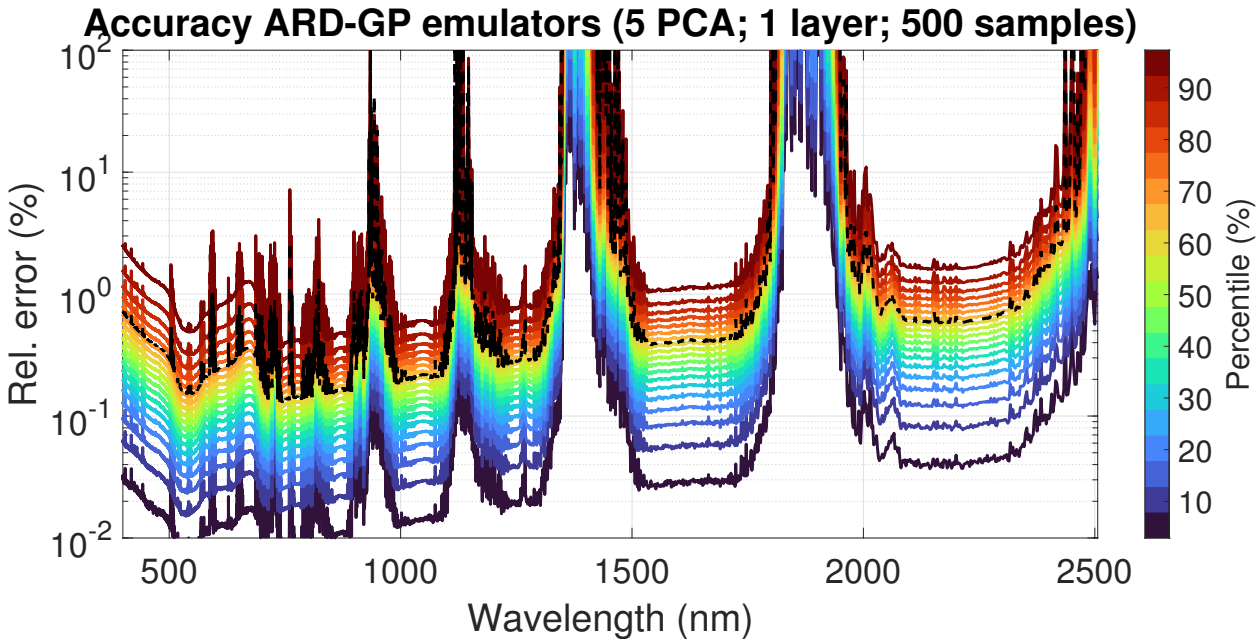


Fig. 6. Relative error histogram (in %) (see colour bar) and mean error (black dashed line) between reference and inverted surface reflectance with a multi-fidelity emulator with n=500, n_c=5 and n_l=1.

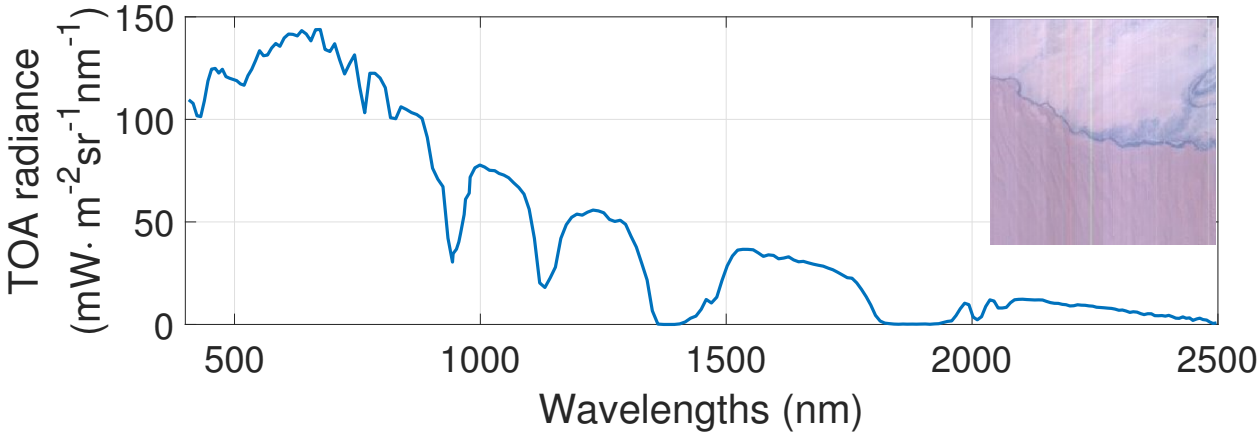


Fig. 7. Sample TOA radiance spectrum and quickview of PRISMA Level-1 product acquired over Gobabeb (Namibia) on the 05/10/2022 at 9:15.

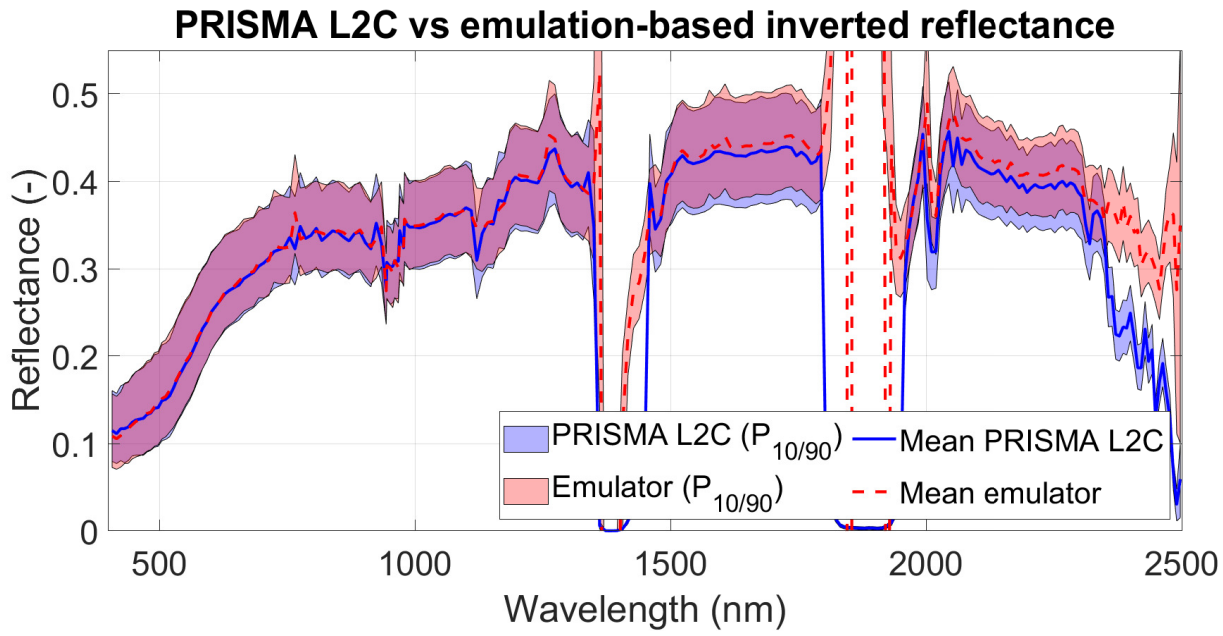


Fig. 8. Average surface reflectance from PRISMA L2C product and inverted from emulation (blue and red dashed lines respectively) and percentiles (P_x) 10% and 90% (shaded areas). Overlapping areas are seen in purple colour.