



EmulART: Emulating radiative transfer—a pilot study on autoencoder-based dimensionality reduction for radiative transfer models

João Rino-Silvestre¹ · Santiago González-Gaitán¹ · Marko Stalevski^{2,3} · Majda Smole² · Pedro Guilherme-Garcia⁴ · Joao Paulo Carvalho⁵ · Ana Maria Mourão¹

Received: 20 April 2022 / Accepted: 16 November 2022 / Published online: 22 December 2022
© The Author(s) 2022

Abstract

Dust is a major component of the interstellar medium. Through scattering, absorption and thermal re-emission, it can profoundly alter astrophysical observations. Models for dust composition and distribution are necessary to better understand and curb their impact on observations. A new approach for serial and computationally inexpensive production of such models is here presented. Traditionally these models are studied with the help of radiative transfer modelling, a critical tool to understand the impact of dust attenuation and reddening on the observed properties of galaxies and active galactic nuclei. Such simulations present, however, an approximately linear computational cost increase with the desired information resolution. Our new efficient model generator proposes a denoising variational autoencoder (or alternatively PCA), for spectral compression, combined with an approximate Bayesian method for spatial inference, to emulate high information radiative transfer models from low information models. For a simple spherical dust shell model with anisotropic illumination, our proposed approach successfully emulates the reference simulation starting from less than 1% of the information. Our emulations of the model at different viewing angles present median residuals below 15% across the spectral dimension and below 48% across spatial and spectral dimensions. EmulART infers estimates for $\sim 85\%$ of information missing from the input, all within a total running time of around 20 minutes, estimated to be $6\times$ faster than the present target high information resolution simulations, and up to $50\times$ faster when applied to more complicated simulations.

Keywords Denoising variational autoencoder · Spatial inference · Emulation · INLA · Radiative transfer

1 Introduction

Cosmic dust is ubiquitous in the Universe, particularly present in the interstellar medium (ISM) [56] and in the line of sight towards astrophysical objects such as supernovae remnants [42], galaxies [15] and active galactic nuclei (AGN) (AGN, Haas et al. [21]). Dust grains absorb and scatter UV/optical radiation and re-emit that energy at infrared wavelengths and are thus responsible for both attenuation and reddening of light in the line of sight, which also impact distance measurements for cosmology when using “standard candles” such as supernovae [7]. Moreover, scattering on the dust grains and dichroic absorption in the dusty medium may lead to polarization of the light as it traverses the interstellar medium.

The effect these processes—of dust absorption, scattering and emission—have on the light detected from astronomical objects must be accounted when studying their intrinsic properties. Moreover, the nature of the dust can also inform us on physical and chemical processes related to its own history, from formation, variation in composition, growth and destruction in different astrophysical structures such as accretion disks, clouds and galaxies, as well as its interaction with magnetic fields through dust grain alignment. The analysis of the aforementioned interactions requires performing radiative transfer (RT) calculations [49], allowing us to simulate the light path, from the source to the observer, depending on the physical properties of the emitting source embedded in dust structures of different geometries, sizes and composition. Comparing various RT models, based on different simulated properties, to global and pixel-by-pixel spectral energy distributions (SEDs) obtained in astronomical observations we can infer valuable information on the

Santiago González-Gaitán and Marko Stalevski contributed equally to this work.

Extended author information available on the last page of the article

properties of the light sources, as well as the distribution and properties of the dust [2, 12, 17].

Observations of molecular clouds [6, 16] have shown that dust distributions are often inhomogeneous and complex; consequently, the understanding of their intrinsic properties requires 3D radiative transfer calculations. This type of non-local and nonlinear problem requires calculations which are computationally very costly [49]; this has prompted the search for alternative non-analytic approximate ways to address them. One of the most successful ways is Monte Carlo Radiative Transfer (MCRT, Mattila [34]; Roark et al. [43]).

Monte Carlo Radiative Transfer methods simulate a large number of test photons, which propagate from their emission by a source through their journeys through the dusty medium. At every stage, the characteristics that define their paths are determined by generating numbers from the probability density function most suited for each process they may undergo (absorption, scattering, re-emission, etc.). At the end of the simulation, the radiation field is recovered from a statistical analysis of the photon paths. As an ensemble, the distribution of particles provides a good representation of the radiative transfer, as long as a sufficient number of photons is chosen.

Stellar Kinematics Including Radiative Transfer (SKIRT, Baes et al. [5]; Camps and Baes [9]) is an MCRT suite that offers some built-in source templates, geometries, dust characterizations, spatial grids, and instruments, as well as an interface so that a user can easily describe a physical model. The user can in this way avoid coding the physics that describes both the source (e.g., AGN or galaxy type, observation perspective, emission spectrum) and environment (between the simulated source and observer, such as dust grain type and orientation, dust density distribution, etc.) but instead design a model of modular complexity by following a Q &A prompt (itself adaptable to the user expertise).

MCRT simulations suffer from computational limitations, namely the memory requirement scaling with the volume grid density, and the processing time scaling quasi-linearly with the amount of photons simulated [10]. Autoencoders [55] together with collocation strategies [20] have been applied to solve complex interactions such as the ones described above; our approach differs by attempting to upscale the information density within a simulated data product instead. Considering that the objects and phenomena modeled by MCRT simulations present non-random spatial structures with heavily correlated spectral features, we tackle the computational cost issue through the development of an emulator that can achieve HPN-like MCRT models by exploring and implementing an autoencoder neural network in combination with integrated nested Laplace approximation (INLA, Rue et al. [44]), an

approximate method for Bayesian inference of spatial maps modeled with Gaussian Markov random fields, on LPN-like MCRT simulations. The results are then compared against an analogous implementation employing principal component analysis (PCA).

Section 2 provides a brief highlight of the employed methods. Section 3 describes our pipeline architecture and some of steps that lead to its development. Results and performance evaluation follow in Sect. 4. Section 5 presents our perspective on the significance of the obtained results and provides the steps to follow in order to both improve and generalize them in future developments.

All files concerning this work (SKIRT simulations, R scripts, neural network models, emulation products and performance statistics) can be obtained from our repository.¹

2 Methods

To reduce the computational cost of SKIRT simulations, without compromising, as much as possible, the quality of the resulting models, an autoencoder, i.e., a dimensionality reduction neural network, is implemented to compress the spectral information within LPN spectroscopic data cubes. Then, approximate Bayesian inference is performed with INLA on the spatial information of the compressed feature maps. Lastly, the reconstructed feature maps are decompressed to an HPN emulation.

2.1 SKIRT

As previously state, SKIRT allows the creation of models by prompting a Q &A. Through it the user can configure any one- to three-dimensional geometry and combine multiple sources and media components, each with their own spatial distribution and physical properties, by either employing the built-in library or importing components from hydrodynamic simulations. Media types include dust, electrons and neutral hydrogen; the user can configure their own mixture, including optical properties and size distributions or simply choose from the available built-in mixtures. The included instruments allow the “observation” and recording of spectra, broad-band images or spectroscopic data cubes.

SKIRT uses Monte Carlo method for tracing photon packets through the spatial domain (both regular and adaptive spatial grids area available, as well as some optimized for 1D or 2D models); these packets are wavelength sampled from the source/medium spectrum (the wavelength grids can be separately configured for both

¹ <https://github.com/SN-CRISP/EmulART>.

storing the radiation field and for discretizing medium emission spectra). As they progress through the spatial grid cells, these photon packets can experience different physical interactions, such as multiple anisotropic scattering, absorption and (re-)emission by the transfer medium, Doppler shifts due to kinematics of sources and/or media, polarization caused by scattering off dust grains as well as polarized emission, among others.

The present application is intended for the combination of both spatial and spectral information. The outputs used here will be spectroscopic data cubes; these are in the flexible image transport system (FITS) format [54] and are composed of 2D spatial distributions at the desired wavelength bins.

The simulated spatial flux densities vary according to the amount of photons/photon packets simulated. The simulation starts by assigning some energy to those photons following the spectral energy distribution (SED) of a given astrophysical source, ensuring that no matter how many photons are simulated the spectral information is preserved. Simulations with a lower photon number (LPN) will consequently display fewer spatial positions with information (nonzero flux) and some of these pixels will have higher flux, some lower, i.e., the SED will have lower signal-to-noise ratio than in simulations with higher photon number (HPN). SKIRT has already been employed in the study of various galaxies [13, 51, 52], AGN [47, 48] and other objects.²

2.2 Dimensionality reduction

Dimensionality reduction methods can more familiarly be called as compressors. These are methods that analyze and transform data from a high-dimensional space into a low-dimensional space while attempting to retain as much meaningful properties of the original data as possible. Working in high-dimensional spaces can be undesirable for many reasons; raw data are often sparse as a consequence of the curse of dimensionality,³ and analyzing the data usually becomes computationally intractable. Popular dimensionality reduction techniques in astronomy include PCA [32, 33] and non-negative matrix factorization (NMF, Ren et al. [41]; Boulais et al. [8]).

Autoencoder networks are an alternative method which has been gaining attention within the astronomy, astrophysics and cosmology community [24, 28, 37, 38, 40, 53].

2.2.1 Denoising variational autoencoders

Autoencoders (AEs) are a type of neural network architecture employed to learn compressed representations of data. In such architectures, the input and output layers are equal, and the hidden layers display a funneling in and out scheme in regard to the number of neurons per layer, with the middle layer having the least amount of neurons and the input and output layers having the most. The models built this way can be seen as the coupling of a compressor/encoder and a decompressor/decoder, the first generating a more (ideally) fundamental representation of the data, and the second bringing it back to its initial feature space. After training, the encoder can be coupled at the beginning of other architectures, providing them with more efficient features from which to learn from. Interesting to note that AEs have shown promise as auxiliary tools in citizen science projects,⁴ such as the Radio Galaxy Zoo [40].

Alternative ways of training this kind of network exist, such as:

- Having multiple instances of each data point, resulting from the injection of noise or from a set of transformations. Each instance of the same data point is then matched to the same output with the aim of making the model robust to noise and/or invariant to those transformations.
- Having the mid-layer composed by two complementary layers of neurons (a mean layer and a standard deviation layer) instead of a single layer. Complemented with an appropriate loss function, the model will learn approximate distributions of values instead of single values, making it more robust and allowing for the decoder to also become a generator of different, yet statistically identical, examples.

Such strategies fall under different categories such as denoising autoencoders (DAEs) [31] and variational autoencoders (VAEs) [22], respectively. In this work, we implement both, a denoising variational autoencoder (DVAE).

Im et al. [25] showed that the DVAE, by introducing a corruption model, can lead to an encoder that covers a broader class of distributions when compared to a regular VAE (the corruption model may, however, remove the information necessary for reconstruction). In this context, the loss function, $\mathcal{L}_{\text{DVAE}}$, to be minimized is given by the weighted sum of the Kullback–Leibler divergence (which influences the encoder) and the reconstruction error (see Eq. 1), similarly to a VAE, with the difference that the encoder now learns to approximate a prior $p(z)$ given

² https://skirt.ugent.be/root/_publications_gallery.html.

³ As the number of parameters increases (rapidly increasing the volume of the parameter-space), the density of the data quickly decreases.

⁴ Projects where non-scientists can participate either by collecting or processing data.

corrupted examples and that in this case the reconstruction error can be interpreted as a denoising criterion:

$$\mathcal{L}_{\text{DVAE}} \propto a_1 \mathbb{KL}(q(z | y') | p(z)) + a_2 \ln p(y | z), \quad (1)$$

where \mathbb{KL} refers to the Kullback–Leibler divergence, $y' \sim q(y' | y)$ is a sample of the corruption model, $p(z)$ is the prior for the latent feature, $q(z | y')$ models the encoder, $p(y | z)$ models the decoder/generator and (a_1, a_2) are weights. For more details, the reader is referred to [14, 25].

2.2.2 Principal component analysis

Principal component analysis (PCA) is a method that analyzes the feature space of the training data and creates orthogonal vectors (linear combinations of the initial variables) whose direction indicates the most variability. These new vectors in the transformed data set are called eigenvectors, or principal components, while the eigenvalues represent the coefficients attached to eigenvectors and give the relative amount of variance carried by each principal component.

PCA transforms the original space through rotation of its axes and re-scaling of the axes range. The first new PC is aligned with the direction of largest variance in the data. The second PC should also maximize the variance, while being orthogonal to the first, and, respectively, for the remaining PCs. Mathematically, these directions can be determined through the covariance matrix, as expressed in Eq. (2):

$$\Sigma_{ff'} = \sum_{j=1}^{j=N} \frac{(X_f^j - \bar{X}_f)(X_{f'}^j - \bar{X}_{f'})}{N}, \quad (2)$$

where X_f is the mean of all values at feature f and N is the total number of data points (for more details and a modern review on PCA, we suggest [29]). Once $\Sigma_{ff'}$ is diagonalized, the PCs are its eigenvectors, the first PC being the one with the largest associated eigenvalue and so on.

PCs are uncorrelated and frequently the information is compressed into the first K components, with $K \ll M$ (where M is the total number of features of the original space).

A data point from the original data set can then be reasonably recovered using those K PCs,

$$\hat{\mathbf{X}} \approx \bar{\mathbf{X}} + \sum_{m=1}^{m=K} c_m \mathbf{P}_m, \quad (3)$$

where $\bar{\mathbf{X}}$ represents the mean of all data points, \mathbf{P}_m is the m -th PC and c_m is the projection of the data point on \mathbf{P}_m .

Using all the M PCs, the reconstruction becomes identical to the original data, but with a new basis that captures a large fraction of the variance in a small number of components K , dimensionality reduction is achieved. In

this work, we used two approaches to determine K (see Sect. 3.4).

Further discussion on the importance of PCA and its applications in astronomy can be found in, e.g., [26, 27, 45].

2.3 Spatial approximate Bayesian inference

Bayesian inference (BI) refers to a family of methods of statistical inference where a hypothetical probabilistic model is updated, following Bayes theorem (Eq. 4), whenever new data are obtained. Bayes theorem allows to calculate a posterior distribution $p(\theta | y)$ (the conditional probability of θ occurring given y) by weighting in $p(y | \theta)$ (the likelihood of y occurring given θ), $p(\theta)$ (estimation of the probability distribution of θ before observing y , also designated as a prior), and $p(y)$ (the marginal probability of y , obtained from integrating θ out of $p(y | \theta)$):

$$p(\theta | y) = \frac{p(y | \theta)}{p(y)} p(\theta), \quad (4)$$

$$p(y) = \int p(y | \theta) p(\theta) d\theta. \quad (5)$$

This kind of update is of utmost relevance in the dynamical analysis of data streams or in the analysis of correlated data and has been proposed in astronomy, from the study of variable stars [57] to 3-D mapping of the Milky way [3].

Approximation techniques [23, 35, 50] have been developed over the years in order to help curb the very time-consuming process of sampling the whole likelihood $p(y | \theta)$. To this end, we here implement the integrated nested Laplace approximation (INLA, Rue et al. [44]).

2.3.1 Integrated nested Laplace approximation

INLA is an approximate BI method that accounts for spatial correlations between observed data points to recover an assumed Gaussian latent field and, in doing so, it is not only capable of predicting unobserved points of that field but also of correcting noisy observed ones, as well as associating a variance to those inferences.

Most techniques for calculating posterior distribution rely on Markov chain Monte Carlo (MCMC, Collins et al. [11]) methods. In this class of sampling-based numerical methods, the posterior distribution is obtained after many iterations, which is often computationally expensive. INLA provides a novel approach for faster BI. While MCMC methods draw a sample from the joint posterior distribution, the Laplace approximation is a method that approximates posterior distributions of the model parameters to Gaussians, which is computationally more effective. Within the INLA framework, the posterior

distribution of the latent Gaussian variables \mathbf{x} and hyper-parameters of the model θ is:

$$p(\mathbf{x}, \theta | \mathbf{y}) = \frac{p(\mathbf{y} | \mathbf{x}, \theta) p(\mathbf{x}, \theta)}{p(\mathbf{y})} \propto p(\mathbf{y} | \mathbf{x}, \theta) p(\mathbf{x}, \theta), \quad (6)$$

where $\mathbf{y} = (y_1, \dots, y_n)$ represents a set of observations. Each observation is treated with a latent Gaussian effect, with each x_i (a Gaussian distribution of mean value μ_i and standard deviation σ_i) corresponding to an observation y_i , where $i \in [1, \dots, n]$. The observations are conditionally independent given the latent effect \mathbf{x} and the hyper-parameters θ , and the model likelihood is then:

$$p(\mathbf{y} | \mathbf{x}, \theta) = \prod_i p(y_i | x_i, \theta). \quad (7)$$

The joint distribution of the latent effects and the hyper-parameters, $p(\mathbf{x}, \theta)$ can be written as $p(\mathbf{x} | \theta)p(\theta)$, where $p(\theta)$ represents the prior distribution of hyper-parameters θ . It is assumed that the spatial information can be treated as a discrete sampling of an underlying continuous spatial field, a latent Gaussian Markov random field (GMRF), that takes into account the spatial correlations, and whose hyper-parameters are inferred in the process. For a GMRF, the posterior distribution of the latent effects is:

$$p(\mathbf{x} | \theta) \propto |\mathbf{Q}(\theta)|^{1/2} \exp -\frac{1}{2} \mathbf{x}^T \mathbf{Q}(\theta) \mathbf{x}, \quad (8)$$

where $\mathbf{Q}(\theta)$ represents a precision matrix, or inverse of a covariance matrix, which depends on a vector of hyper-parameters θ . This kernel matrix is what actually treats the spatial correlation between neighboring observations. Using Eq. (6), the joint posterior distribution of the latent effects and hyper-parameters can be written as:

$$\begin{aligned} p(\mathbf{x}, \theta | \mathbf{y}) &\propto p(\theta) |\mathbf{Q}(\theta)|^{1/2} \exp \\ &\quad -\frac{1}{2} \mathbf{x}^T \mathbf{Q}(\theta) \mathbf{x} \prod_i p(y_i | x_i, \theta) \\ &= p(\theta) |\mathbf{Q}(\theta)|^{1/2} \exp \\ &\quad -\frac{1}{2} \mathbf{x}^T \mathbf{Q}(\theta) \mathbf{x} + \sum_i \log(p(y_i | x_i, \theta)). \end{aligned} \quad (9)$$

Instead of obtaining the exact posterior distribution from Eq. (9), INLA approximates the posterior marginals of the latent effects and hyper-parameters, and its key methodological feature is to use appropriate approximations for the following integrals:

$$p(x_i | \mathbf{y}) = \int p(x_i | \theta, \mathbf{y}) p(\theta | \mathbf{y}) d\theta \quad (10)$$

$$p(\theta_j | \mathbf{y}) = \int p(\theta | \mathbf{y}) d\theta_{-j}, \quad (11)$$

where θ_{-j} is a vector of hyper-parameters θ without element θ_j .

INLA constructs nested approximations:

$$\tilde{p}(x_i | \mathbf{y}) = \int \tilde{p}(x_i | \theta, \mathbf{y}) p(\theta | \mathbf{y}) d\theta, \quad (12)$$

$$\tilde{p}(\theta_j | \mathbf{y}) = \int \tilde{p}(\theta | \mathbf{y}) d\theta_{-j}, \quad (13)$$

where $\tilde{p}(\cdot | \cdot)$ is an approximated posterior density. Using the Laplace approximation, the posterior marginals of hyper-parameters $p(\theta | \mathbf{y})$ at a specific value $\theta = \theta_j$ can be written as:

$$\tilde{p}(\theta_j | \mathbf{y}) \propto \frac{p(\mathbf{x}, \theta_j, \mathbf{y})}{p_G(\mathbf{x} | \theta_j, \mathbf{y})} \propto \frac{p(\mathbf{y} | \mathbf{x}, \theta_j) p(\mathbf{x} | \theta_j) p(\theta_j)}{p_G(\mathbf{x} | \theta_j, \mathbf{y})} \Big|_{\mathbf{x}=\mathbf{x}^*(\theta_j)}, \quad (14)$$

$$\tilde{p}_G(\mathbf{x} | \theta, \mathbf{y}) \propto \exp -\frac{1}{2} \mathbf{x}^T \mathbf{Q}(\theta) \mathbf{x} + \sum_i g_i(x_i), \quad (15)$$

where $\tilde{p}_G(\mathbf{x} | \theta, \mathbf{y})$ is the Gaussian approximation to the full conditional of \mathbf{x} , and $\mathbf{x}^*(\theta_j)$ is the mode of the full conditional \mathbf{x} for given θ_j . The posterior marginals of the latent effects are then numerically integrated as follows:

$$\tilde{p}(x_i | \mathbf{y}) \simeq \sum_j \tilde{p}(x_i | \theta_j, \mathbf{y}) p(\theta_j | \mathbf{y}) \Delta_j, \quad (16)$$

where Δ_j represents the integration step.

A good approximation for $\tilde{p}(x_i | \theta, \mathbf{y})$ is required and INLA offers three different options: Gaussian approximation, Laplace approximation and simplified Laplace approximation [44]. In this work, we used the simplified Laplace approximation, which represents a compromise between the accuracy of the Laplace approximation and the reduced computational cost achieved with the Gaussian approximation.

INLA has been shown [44] to greatly outperform MCMC sampling under limited computational power/time conditions, with the estimation error of INLAs results being invariably smaller than those of MCMC. Other approximated inference methods exist, such as variational Bayes [23] and expectation-propagation [35], however these methods are not only slower, but they struggle with estimating the variance of the posterior since they execute iterative calculations instead of analytic approximations, unlike INLA [44].

INLA suffers nonetheless from some limitations. The first, already mentioned above, is that to get meaningful results the latent field to be inferred must be Gaussian—which is not always the case—and it must display conditional independence properties; the second is that for fast inference it is required that the number of hyper-parameters (characterizers of the parameter models) should be inferior

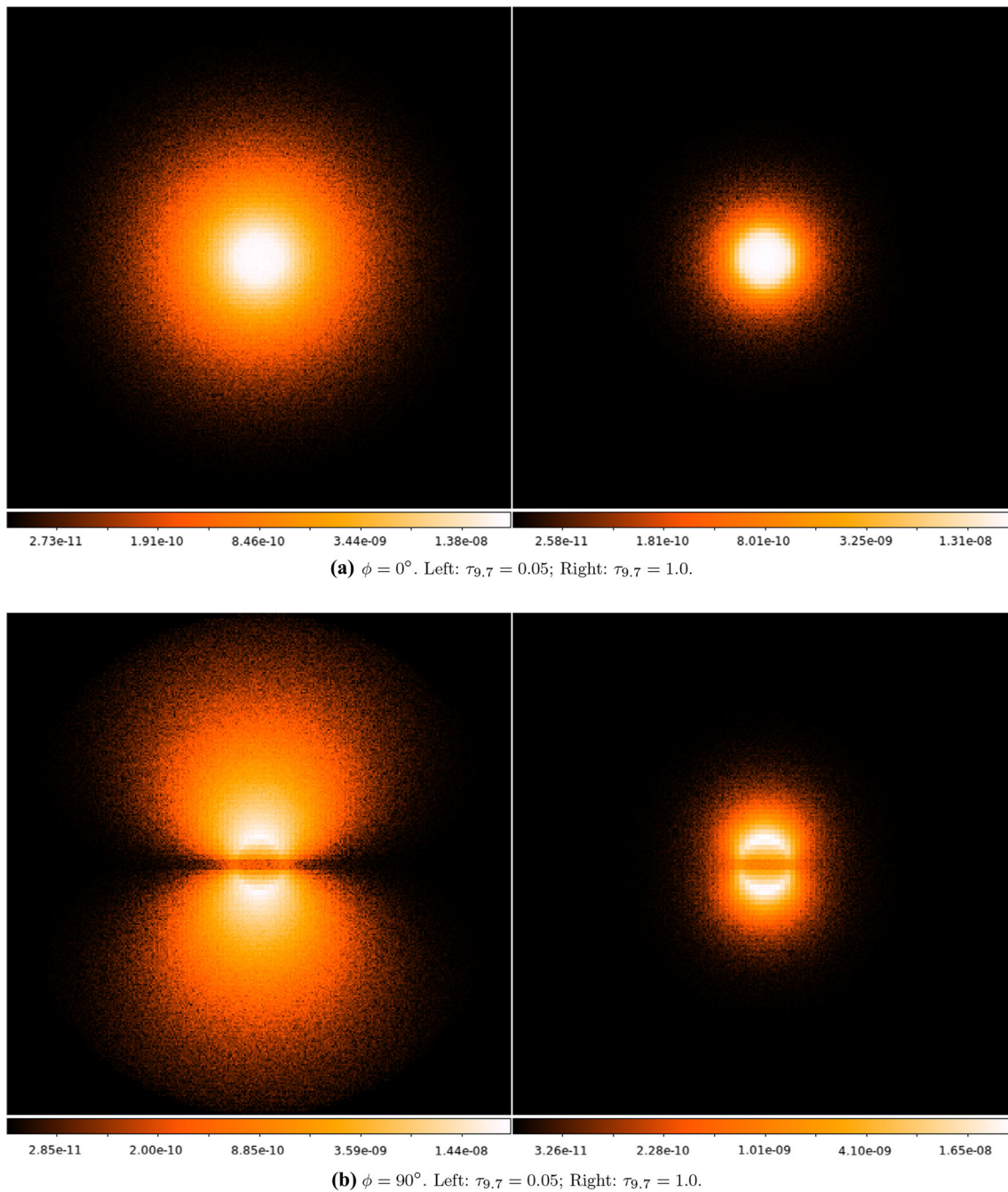


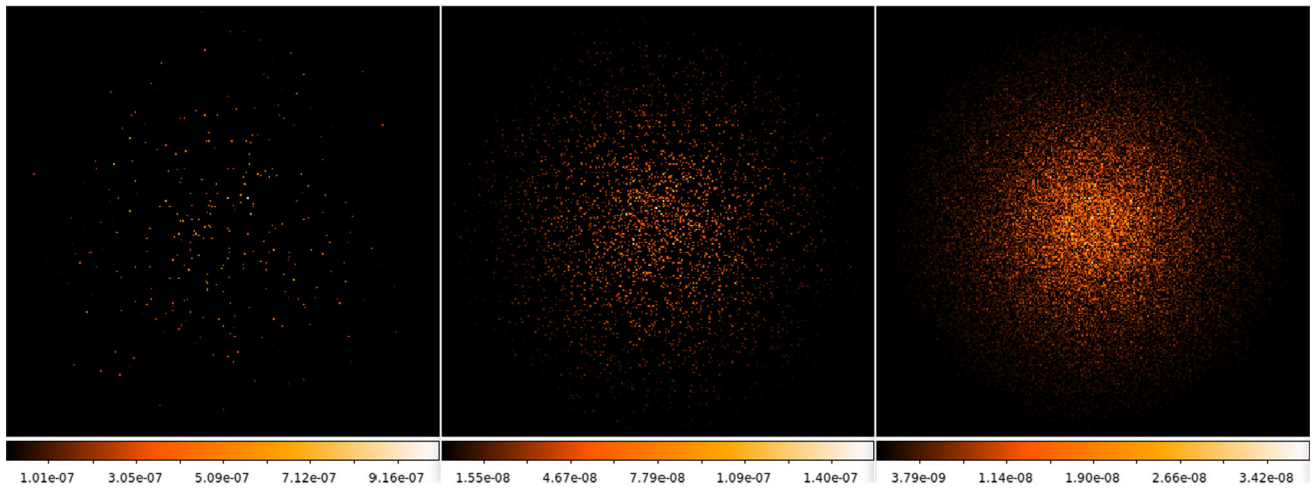
Fig. 1 High photon number references of a spherical dust shell composed of silicates and graphites surrounding a bright anisotropic point source. The present flux density maps represent the simulated

observations at wavelength $1.85 \mu\text{m}$, with $\phi = 0^\circ$ (a) and $\phi = 90^\circ$ (b) for $\tau_{9,7} \in \{0.05, 1.0\}$. Color indicates flux density in W/m^2 (Color figure online)

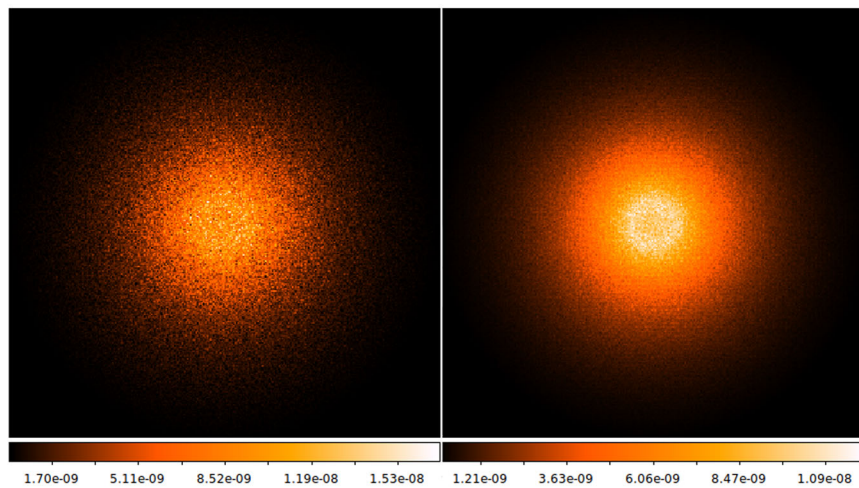
to 6 and that the number of available observations of the field to infer be much smaller than the size of that field.

INLA is freely available as an R package [4], and it has already been shown to: (1) be capable of recovering

structures in scalar and vector fields, with great fidelity, out of sparse sets of observations, and even of inferring structures never seen before; (2) be robust to noise injections [19]. We refer to [18, 44] for more details on the



(a) Left: $N_p = 10^4$; Middle: $N_p = 10^5$; Right: $N_p = 10^6$.



(b) Left: $N_p = 10^7$; Right: $N_p = 10^8$.

Fig. 2 Models of a spherical dust shell composed of silicates and graphites surrounding a bright anisotropic point source, with $\tau_{9.7} = 0.05$ and $\phi = 0^\circ$, realized by simulating different photon amounts. The present flux density maps represent the simulated observations at

wavelength $9.28 \mu\text{m}$. **a** Presents the realization obtained by simulating $N_p \in \{10^4, 10^5, 10^6\}$, while **b** presents the realizations obtained by simulating $N_p \in \{10^7, 10^8\}$. Color indicates flux density in W/m^2 (Color figure online)

Table 1 Statistics regarding how the LPN realizations compare to the HPN reference

| Sim. | Normalized residuals of Sim. (%) | | Information of simulation | | | | |
|--------|----------------------------------|-----|---------------------------|-------------------|---------------------------|-------------------|----------------------|
| | M | MAD | TIR (%) | Null Spxs (N) | Empty Spxs (ΔN) | Full Spxs (N) | Partial Spxs (N) |
| 10^4 | 100 | 0 | 0.2 | 82,041 | 64,185 | 0 | 7959 |
| 10^5 | 100 | 0 | 1.9 | 47,964 | 30,108 | 0 | 42,036 |
| 10^6 | 100 | 0 | 15.4 | 19,115 | 1259 | 0 | 70,885 |
| 10^7 | 100 | 59 | 65.2 | 17,904 | 48 | 2784 | 69,312 |
| 10^8 | 0 | 0 | 100 | 17,856 | 0 | 47,160 | 24,984 |

Median normalized residuals of 100% indicate that missing information permeates the LPN realizations. The quantity and quality of the available information is further described by the total information ratio (TIR), and the amounts of different qualities of spaxels. Null spaxels are spaxels with zero information; empty spaxels are null spaxels that were not tagged as such for the HPN reference; full spaxels are spaxels that have information at every wavelength and, partial are those spaxels that are neither null nor full

mathematical background of INLA and the methods it employs.

3 Implementation

This section describes both the dataset, the combination of a DVAE/PCA with INLA to enhance low information density SKIRT simulation data cubes and the tools to do so.

Our pilot pipeline aims to emulate radiative transfer models, as such we named it EmulART. All scripts were written and executed under R [39] (version 3.6.3) and make use of the Keras API [1].

The DVAE architecture was adapted from the one described in the Keras documentation (https://tensorflow.rstudio.com/guides/keras/making_new_layers_and_models_via_subclassing.html#putting-it-all-together-an-end-to-end-example). The encoder block starts with an input layer of 64 features/neurons (the wavelengths of the SEDs), and each consecutive layer halves the amount of features until the latent space layer is reached. That layer, unlike the ones that precede it, is comprised of a vector doublet, each with 8 neurons. One vector corresponds to the mean value of the latent features and the other to their variance, together these vectors, describe a value distribution for each of the latent features. The input of the decoder will be drawn from those distributions, and each subsequent layer will double the amount of features, decompressing the data, until the output, with the same number of features as the input layer of the encoder, 64, is reached.

Two pipelines using principal component models were used to compare against the DVAE pipeline. One pipeline makes use of the 8 PCs which explained the most variance (the same number of latent variables as available for the DVAE), while the other uses the number of PCs determined by the elbow method.

3.1 Dataset

In this work, 30 SKIRT simulations were used for separate purposes. All simulations model a spherical dust shell composed by silicates and graphites surrounding a bright point source with anisotropic emission [47], as defined by Eq. (17) following Netzer [36]:

$$L(\theta) \propto \cos \theta (2 \cos \theta + 1), \quad (17)$$

where θ is the polar angle of the coordinate system. Each realization is a cube of 300-by-300 pixel maps at 103 distinct wavelength bins. The first 39 wavelength bins were discarded (leaving us with 64) for displaying very low signal of randomly scattered emission (less than 0.0001% of the pixels at these wavelengths display flux density different than 0). The final dataset thus includes 90,000

spaxels⁵ per cube, each spaxel with 64 fluxes, or “features”, at wavelength bins ranging from $\sim 1 \mu\text{m}$ to 1 mm.⁶

The 30 realizations differ from each other by up to three parameters: the tilt angle, ϕ , of the object as seen by the observer (0° , face-on, and 90° , edge-on;⁷) the optical depth,⁸ $\tau_{9.7}$,⁹ of the dust shell (0.05, 0.1 and 1.0); the amount¹⁰ of photon packets simulated, $N_p \in \{10^4, 10^5, 10^6, 10^7, 10^8\}$. For each particular $\tau_{9.7}$ and ϕ combination, the corresponding $N_p = 10^8$ realization was regarded as the HPN reference, or “ground truth”, for the purpose of evaluating the performance of our routines, since those yield the highest information density, while all other simulations, with $N_p \in \{10^4, 10^5, 10^6, 10^7\}$, were considered LPN simulations. Figure 1 illustrates the difference between HPN references through different $\tau_{9.7}$ and ϕ combinations, while Fig. 2 shows LPN models with differing N_p , keeping $\tau_{9.7} = 0.05$ and $\phi = 0^\circ$. Table 1 displays the difference between the quality of the individual spaxels¹¹ that compose each LPN realization and HPN reference; the median, M , and mean absolute deviation (MAD) of the normalized residuals (see Eq. 18) of every pixel within each LPN realization; as well as the total information ratio (TIR) for all realizations, here defined as the ratio of the number of pixels with flux different than 0 of an LPN input or emulation, $N_{X' \neq 0}$, and that same number for the HPN reference, $N_{X \neq 0}$ (see Eq. 19), as an information metric to balance against the normalized residuals¹²:

$$\text{Residuals}(\%) = \left| \frac{X' - X}{X} \right| \times 100\%, \quad (18)$$

⁵ By spaxel, we refer to the array across the spectral dimension of at a given pixel spatial coordinates.

⁶ This wavelength range covers almost completely the infrared part of the light spectrum.

⁷ The face-on/edge-on terminology refers to the shape of a disk-like object as seen by an observer when tilt angle between the plane of the disk and the plane of the observer is, respectively, 0° or 90° .

⁸ The optical depth, τ , describes the fraction of light that is transmitted through a material following the relation $\frac{I_0}{I_T} = e^{-\tau}$, where I_0 is the incoming light and I_T the outgoing light.

⁹ Optical depth at wavelength 9.7 μm corresponding to the peak emission wavelength of silicates.

¹⁰ Throughout this text, the symbol N_X will stand for “amount/number/quantity of X ”.

¹¹ We discriminate spaxels according to their completeness along the spectral dimension. Spaxels that have 0 flux at all wavelength bins are classified as “null spaxel”; spaxels that have all wavelength bins with positive flux are classified as “full spaxel”; spaxels in between the two previous cases are classified as “partial spaxel”; finally, the “empty spaxel” information metric is the difference between the amount of null spaxels within the HPN reference and a given LPN realization.

¹² Should the estimation, X' , be 0 the normalized residual will be 100%, meaning there is no actual estimate for the reference value, X .

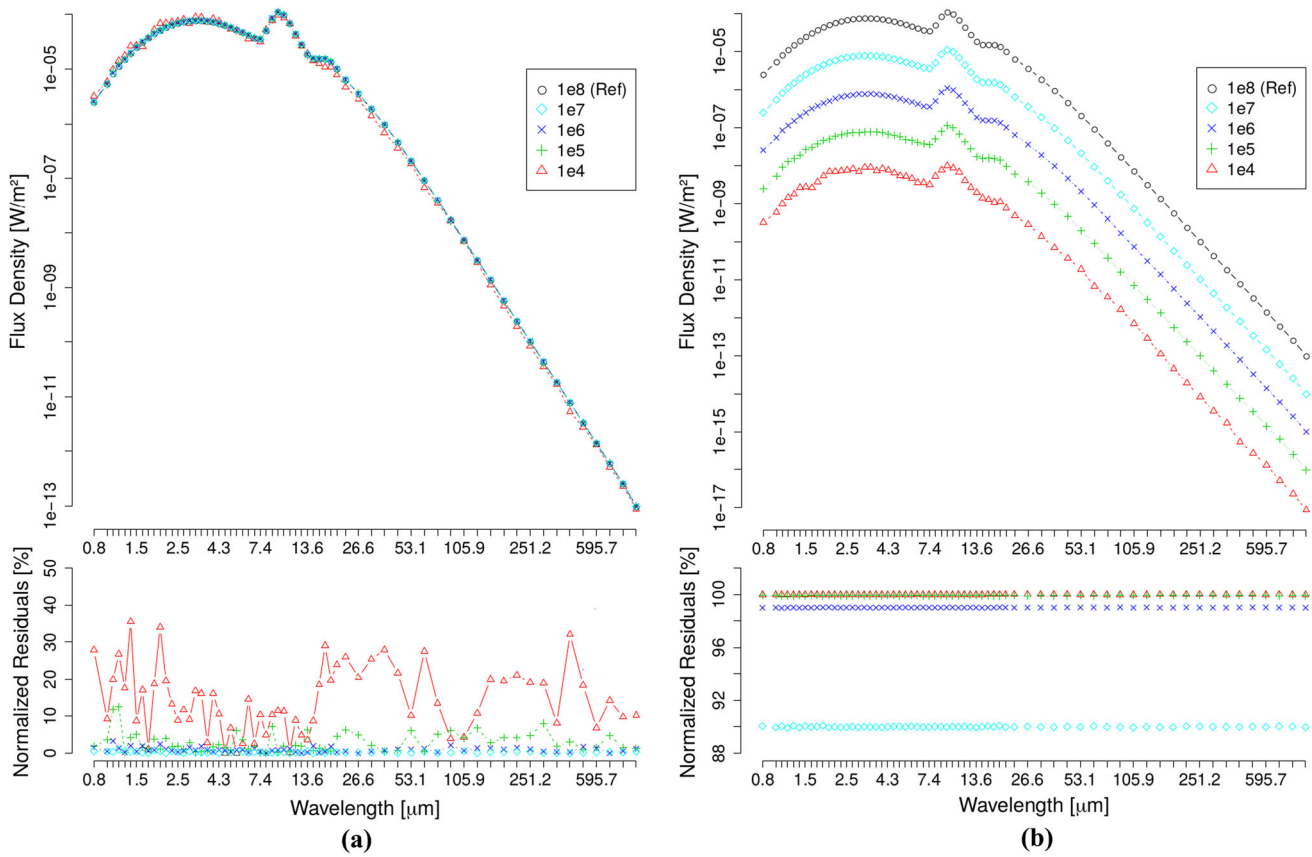
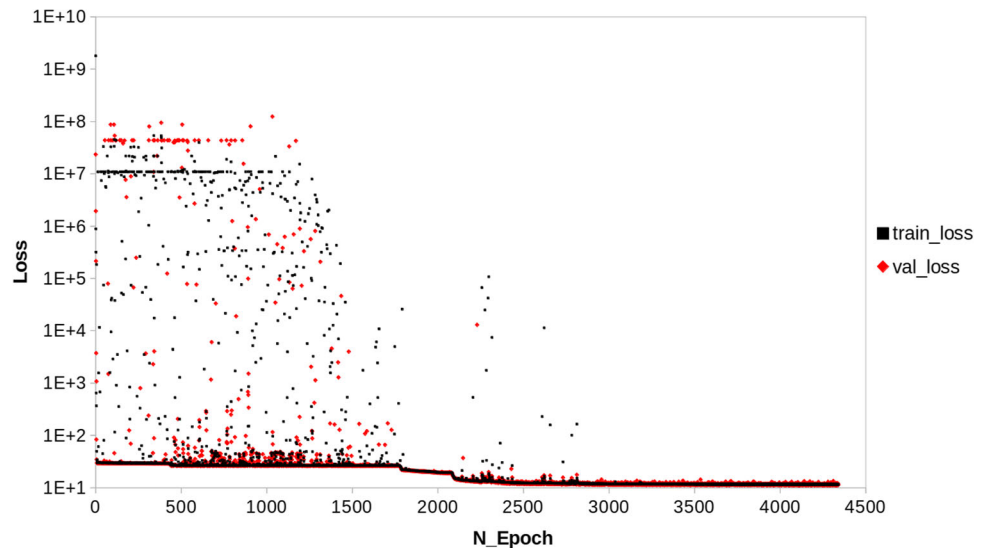


Fig. 3 Integrated SEDs (upper panel) and normalized residuals (lower panel) for each of the five SKIRT simulations in our dataset: **a** due to normalization, realizations with different amounts of photons simulated display very similar integrated SEDs; **b** integrated SEDs of

realizations in AEsSet after de-normalization (total flux is here proportional to N_p). The labels indicate the value of N_p for each realization

Fig. 4 Loss, in log scale, as function of the number of epochs. The training loss is represented by \blacksquare and the validation loss by red \blacklozenge . The plot shows that both metrics converge to comparable values (Color figure online)



$$\text{TIR}(\%) = \frac{N_{X' \neq 0}}{N_{X \neq 0}} \times 100\%. \tag{19}$$

The realizations were split into two subsets: one to train the autoencoder and perform the first batch of tests to the emulation pipeline, labeled AEsSet and described in Sect. 3.1.1; and another, comprised exclusively by data, the

autoencoder did not see during training, to better assess EmulARTs performance, labeled EVASet and described in Sect. 3.1.2.

3.1.1 AESet

This subset is comprised of 5 SKIRT simulation outputs of the same model, a spherical shell of dust composed by silicates and graphites, with optical depth $\tau_{9,7} = 0.05$, surrounding a bright point source with anisotropic emission seen face-on, $\phi = 0^\circ$, making the emission appear isotropic. The only different parameter across the realizations in AESet was $N_p \in \{10^4, 10^5, 10^6, 10^7, 10^8\}$ (see Fig. 2). The $N_p = 10^8$ realization is the HPN and was used as the reference, or “ground truth”, for the purpose of both training the autoencoder model as well as evaluating the performance of EmulART.

Since the goal of our methodology is to reconstruct the reference simulation using LPN realizations as input, the values of each cube were multiplied by the ratio between the amount of photons simulated for the LPN input, N_p^{LPN} and the amount of photons simulated for the HPN reference ($N_p^{\text{HPN}} = 10^8$). In Fig. 3, we can see the impact of this de-normalization on the integrated SEDs¹³ of the LPN realizations: LPN realizations have less flux when not including SKIRTs normalization.

The spaxels of all cubes within AESet were used to train the DVAE model. For this, we split AESet into training set (5/6) and test set (1/6). Spaxels of different cubes but which share the same spatial coordinates were assigned to the same ensemble. This strategy aimed for the DVAE model to achieve a denoising capability and consists on having input spaxels that result from realizations with different N_p always matches the HPN references version on the output layer.

In SKIRT, choosing to simulate fewer photons results in an output with more zero flux density pixels which in turn means that more spaxels will be null at more wavelength bins. Even though this is different than noise, it is akin to missing data whose impact we aim to curb by implementing a DVAE architecture.

Before training the DVAE model, we perform some spaxel selection and preprocessing tasks on AESet which is thoroughly described in Appendix A.

Later, when performing preliminary tests on EmulART, we used the 4 LPN realizations within AESet ($N_p \in \{10^4, 10^5, 10^6, 10^7\}$) as input.

¹³ An integrated SED results from integrating all spatial information at each wavelength.

3.1.2 EVASet

This subset is comprised of the 25 SKIRT simulation cubes that also model a spherical shell of silicates and graphites surrounding a bright anisotropic point source but have different combinations of ϕ , $\tau_{9,7}$ and N_p from those in AESet. To all realizations in EVASet, we performed the same feature selection and flux de-normalization tasks described both in Sect. 3.1.1 and Appendix A.

The 20 LPN realizations within EVASet ($N_p \in \{10^4, 10^5, 10^6, 10^7\}$) were used as input for EmulART for a deeper assessment of the capabilities of our emulation pipeline. The remaining 5 HPN cubes (those with $N_p = 10^8$) were used as references to compute performance metrics.

A list detailing the parameters of the SKIRT simulations used in this work, as well as their split into AESet and EVASet, can be consulted in Appendix B.

3.2 Training the DVAE

To determine which set of hyper-parameters for the DVAE suited our needs best, we performed some tests, grid-searches and explored: amounts of features in the latent space; activation, loss and optimization functions; batch size;¹⁴ bias constraint;¹⁵ learning rate¹⁶; and, patience¹⁷ values. These exploratory tests were performed by training the models during 100, 500 and 2000 epochs, according to the need to differentiate performance between hyper-parameter sets.

We measured performance by the percentage residuals of both the individually reconstructed spaxels, of the test subset of AESet, and of each of AESets cubes integrated SEDs. We selected the set of hyper-parameters listed below for being the most consistent across different tests. Figure 4 shows the validation loss closely following the training loss, indicating successful convergence of the model to our data.

- Latent feature amount: 8
- Activation function: SELU [30] and sigmoid (for the output layer only)

¹⁴ Number of samples that will be passed through the network at one time. An epoch is complete once all samples are passed through the network. Increasing the batch size accelerates the completion of each epoch, but it may also degrade the quality of the model.

¹⁵ Limit, between 0 and 1, for the weight of bias neurons.

¹⁶ Learning rate is a parameter that determines the step size to take, at each iteration, in the direction determined by the optimization function so as to reach the minimum of the loss function.

¹⁷ Number of epochs to wait before implementing a change or stopping the training procedure.

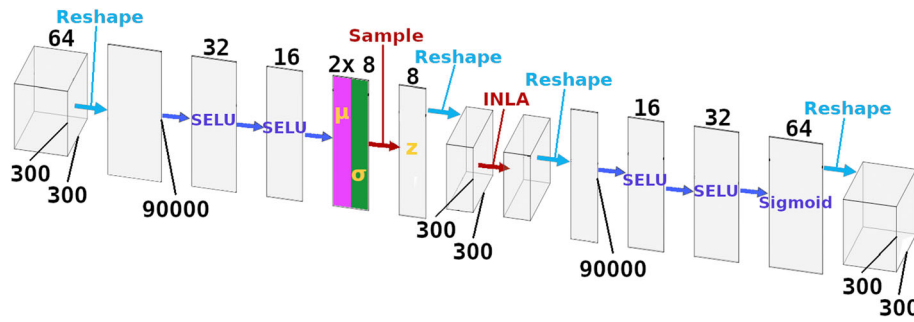


Fig. 5 Scheme with the most relevant operations of the pipeline. Data shaping is represented in cyan; feature combination is in dark blue; and statistical operations are in red. Feature dimensionality is given along the axis of the arrows, at the top of each layer. μ is the vector

holding the mean value of the latent features distribution, while σ is the vector holding their variance; z is the vector built by randomly drawing values from the latent features distributions $G(\mu, \sigma)$ (Color figure online)

- Loss function: weighted sum of the Kullback–Leibler divergence and mean percentual error¹⁸
- Optimization function: Adam¹⁹
- Batch size: 32
- Bias constraint: 0.95
- Train - Validation split: 4/5, 1/5
- Maximum N_{Epoch} : 4500
- Patience 1²⁰: 500 epochs
- Patience 2²¹: 3000 epochs
- Initial learning Rate (LR): 0.001
- Learning rate decrease²²: 0.25

After training, the weights were saved to files which are loaded into the pipeline (see Sect. 3.3).

Appendix C shows the relationship between the compressed (or latent) features of the test set, as well as the Pearson’s correlation coefficients (PCCs) between those features. Based on the analysis of the correlations of the latent features, we decided to stop further compression of the spectral dimensions of the data.

3.3 DVAE Emulation pipeline

Within EmulART, the feature space is first compressed with the variational encoder; the latent space is then sampled and the resulting latent features spatial maps are reconstructed with INLA; finally the reconstructed

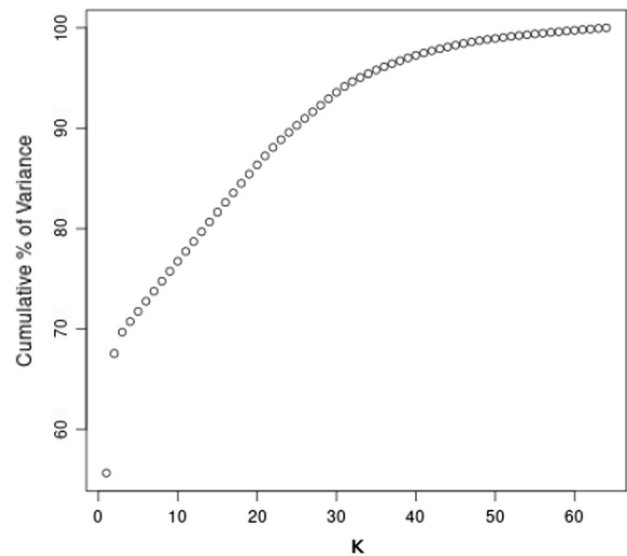


Fig. 6 Plot of cumulative percentage of variance explained as a function of the number of principal components, K used to encode the data. Increasing the amount of PCs used naturally increases the percentage of variance explained, but it also risks over-fitting the model. In this case, the elbow can be found at around $K = 30$

wavelength (original feature space) maps are recovered with the decoder. Additionally, to conform the data to each of the different stages of the pipeline, we perform some operations described below. Figure 5 presents a scheme²³ of the emulation pipeline, while a flowchart including all relevant data pre- and post-processing operations, integrated within the emulation pipeline, can be found in Appendix D.

Prior to being parsed by the encoder network module, the data are initially preprocessed as described in Appendix A. After the data go through the encoding and variational

¹⁸ A custom variation of the one presented in Keras documentation (https://tensorflow.rstudio.com/guides/keras/making_new_layers_and_models_via_subclassing.html#putting-it-all-together-an-end-to-end-example) as the original version of this loss function makes use of the mean squared error instead.

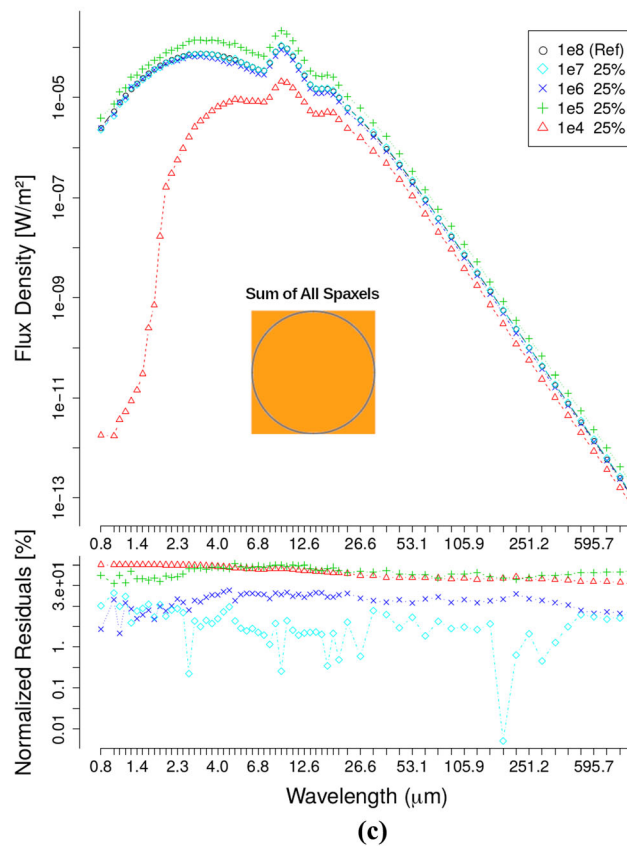
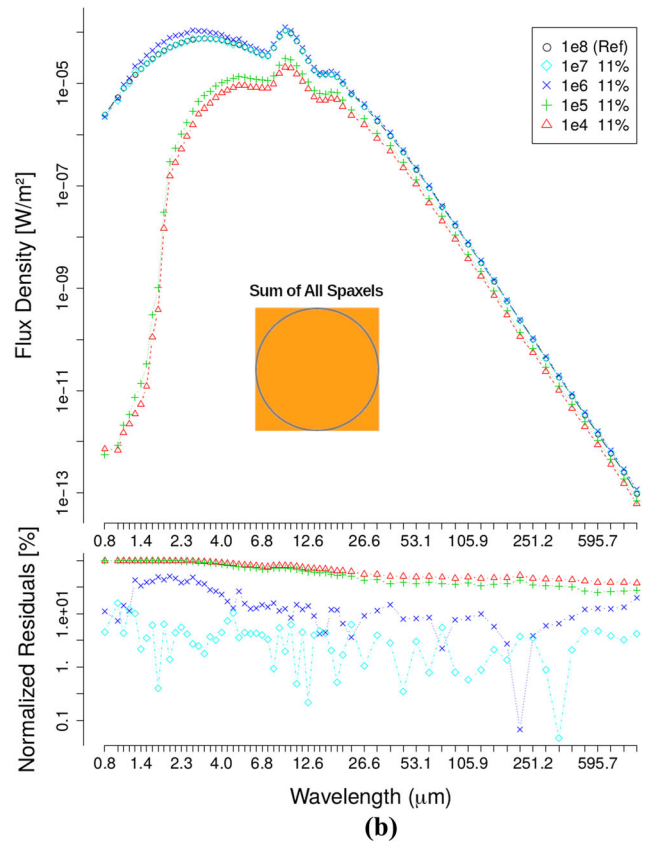
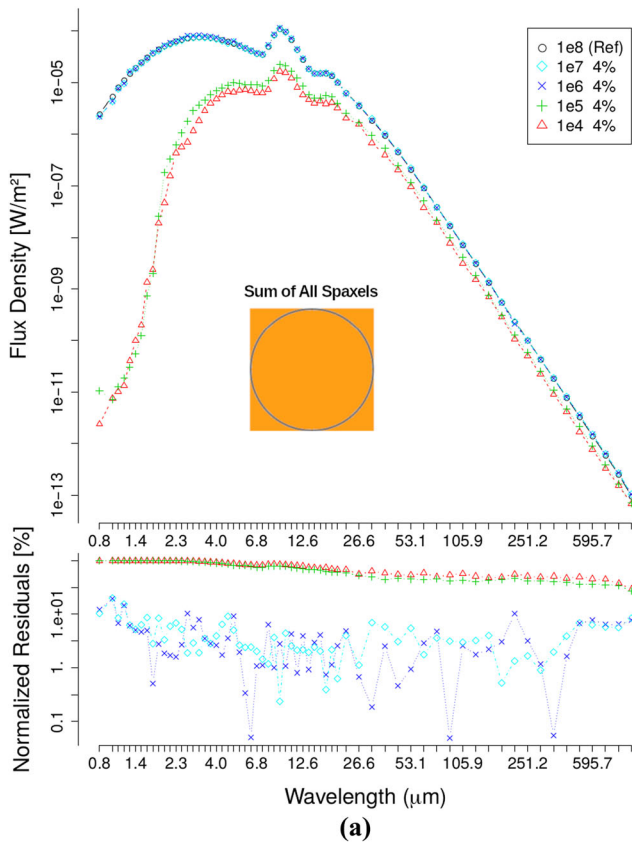
¹⁹ A stochastic gradient descent method based on adaptive estimation of first-order and second-order moments.

²⁰ Number of epochs to wait, with no significant improvement in validation loss, before reducing the learning rate.

²¹ Number of epochs to wait, with no significant improvement in validation loss, before stopping the training process.

²² Ratio between the new and old learning rates.

²³ This illustration was drawn using NNSVG.



◀**Fig. 7** Emulation-integrated SEDs resulting from spatial inference using 4% (a), 11% (b) and 25% (c) samples of the spatial information, and the respective normalized residuals, for the case of a dust shell with $\tau_{9,7} = 0.05$ and $\phi = 0^\circ$. The HPN reference is represented in black (\circ), the emulation based on the $N_p = 10^4$ realization is in red (Δ), on the $N_p = 10^5$ in green (+), on the $N_p = 10^6$ in blue (\times) and on the $N_p = 10^7$ in cyan (\diamond) (Color figure online)

sampling stages (see Fig. 5), the resulting latent features maps, $Z_f(x, y)$,²⁴ are then transformed according to the following steps²⁵ for each feature map, $Z_f(x, y)$:

1. Determine the minimum, m_f^0 , and maximum, M_f^0 , values of the map,

$$m_f^0 = \min(Z_f(x, y)), \quad M_f^0 = \max(Z_f(x, y));$$

2. Determine the value range, R_f , of the map,

$$R_f = M_f^0 - m_f^0;$$

3. Offset the value range of the map so that the new minimum is 0,

$$Z_f'(x, y) = Z_f(x, y) - m_f^0;$$

4. For the offset map, $Z_f'(x, y)$, determine the minimum positive value, m_f^1 , and divide it by R_f^2 to obtain the new minimum, m_f^2 ,

$$m_f^1 = \min(Z_f'(x, y)) : Z_f'(x, y) > 0,$$

$$m_f^2 = m_f^1 / R_f^2;$$

5. Obtain the final map, $Z_f''(x, y)$, by offsetting the value range by m_f^2 ,

$$Z_f''(x, y) = Z_f'(x, y) + m_f^2.$$

These transformations, found by trial and error, are useful in conveying the data to INLA in a value range where its performance is both consistent (across different inference task in this scientific domain), less prone to run-time errors and more accurate. It should be noted that the validity and the improvement on performance granted by this interval transformation have only been empirically verified for our particular case, and it may well be improved upon.

After these transformations, the latent feature maps are reconstructed by INLA. Then, they are transformed back to the original value range and are parsed by the decoder network module.

²⁴ Where $f \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ denotes feature.

²⁵ The reader is reminded that the present, unchanged, dataset, being simulated flux density values, has an inferior limit of 0.

3.4 PCA Emulation pipelines

For the PCA emulation pipeline, a new PCA model of the input feature space is constructed every time (unlike with the EmulART which DVAE model was trained on a subset of AESet, as described in Sect. 3.2). Once that model is constructed, two approaches are followed: the first being the usage of the elbow method (see Fig. 6) to find the threshold number of components, K , that would explain the data without over-fitting it, using INLA to spatially reconstruct those components maps, and then return to them to the original feature space. In the second approach, 8 principal components are used (the same number as the latent features in the DVAE emulation pipeline).

After some preliminary tests, the data range transformations described in the previous section, before and after spatial reconstruction, were also removed as they failed to lead to execution time or reconstruction accuracy improvement. Moreover, spaxel de-normalization (see Sect. 3.1.1) was removed as it leads to the underestimation of the integrated SEDs of the reference model (possible reasons for this are presented in Sect. 4.1.1).

A scheme presenting the most relevant operations for both implementations of the PCA emulation pipeline can be found in Appendix D.

4 Results and discussion

In this section, we present and discuss some of the results from testing EmulART on AESet; these are compared against the results obtained with the PCA emulation implementations, and EVASet. Our goal in first using AESet was to evaluate the performance of the pipeline as a whole, mostly because the decoder network was not trained on latent and INLA reconstructed spaxels. The realizations from EVASet were then used to better gauge its performance.

We created the emulations using different LPN inputs ($N_p \in \{10^4, 10^5, 10^6, 10^7\}$). Because INLA performs faster on sparse maps, for each of those LPN realizations an emulation was performed by sampling different percentages of each latent feature map. These sampling percentages resulted from sampling 1 pixel in each bin of 2×2 , 3×3 and 5×5 pixels, corresponding, respectively, to 25, 11 and 4% of the spatial data. With the intent of reducing the influence of null spaxels in the spatial inference, 90% of the null spaxels were rejected from each map sampling pool.

Our analysis of the results consisted on inspecting how well EmulART reproduces the spectral and spatial features of the reference simulations, as well as the total computational time it took for the emulation to be completed.

Table 2 Comparison of statistics for the residuals of the integrated SEDs, for the case of a dust shell with $\tau_{9,7} = 0.05$ and $\phi = 0^\circ$, for the different LPN realizations (columns 2 and 3), for those same realizations but after they have been de-normalized (columns 4 and

5), as described in Sect. 3.1.1, and for the resulting emulations (columns 7 and 8), while using different sampling amounts (column 6) for the spatial reconstruction

| LPN Input | Residuals of integrated input (%) | | Residuals of integrated de-normalized input (%) | | Sample (%) | Residuals of integrated Emul. (%) | |
|-----------|-----------------------------------|------|---|------|------------|-----------------------------------|-----|
| | <i>M</i> | MAD | <i>M</i> | MAD | | <i>M</i> | MAD |
| 1e4 | 13.4 | 7.4 | 99.99 | 0.00 | 4 | 82 | 18 |
| | | | | | 11 | 78 | 19 |
| | | | | | 25 | 77 | 20 |
| 1e5 | 1.9 | 2.0 | 99.90 | 0.00 | 4 | 75 | 21 |
| | | | | | 11 | 68 | 23 |
| | | | | | 25 | 67 | 16 |
| 1e6 | 0.81 | 0.51 | 99.00 | 0.01 | 4 | 2.8 | 2.5 |
| | | | | | 11 | 12 | 10 |
| | | | | | 25 | 14.3 | 4.3 |
| 1e7 | 0.19 | 0.17 | 90.00 | 0.03 | 4 | 3.3 | 2.4 |
| | | | | | 11 | 3.6 | 2.0 |
| | | | | | 25 | 3.8 | 2.7 |

Table 3 Statistics regarding how the different emulations of the case of a dust shell with $\tau_{9,7} = 0.05$ and $\phi = 0^\circ$, compared to the HPN reference

| LPN input sample | Total Emul. time (min) | Normalized residuals of Emul. (%) | | Information of emulation | | | | |
|---------------------|------------------------|-----------------------------------|-----|--------------------------|------------------------|---------------------------|------------------------|---------------------------|
| | | <i>M</i> | MAD | TIR (%) | Null Spxs (<i>N</i>) | Empty Spxs (ΔN) | Full Spxs (<i>N</i>) | Partial Spxs (<i>N</i>) |
| 10 ⁴ 4% | 7.2 | 77 | 33 | 98.3 | 20,712 | 2856 | 69,288 | 0 |
| 10 ⁴ 11% | 33 | 76 | 35 | | | | | |
| 10 ⁴ 25% | 20 | 76 | 35 | | | | | |
| 10 ⁵ 4% | 6.9 | 76 | 36 | 102.3 | 17,858 | 2 | 72,142 | 0 |
| 10 ⁵ 11% | 22 | 75 | 37 | | | | | |
| 10 ⁵ 25% | 19 | 196 | 272 | | | | | |
| 10 ⁶ 4% | 8.9 | 42 | 45 | 103.6 | 16,969 | – 887 | 73,031 | 0 |
| 10 ⁶ 11% | 25 | 40 | 45 | | | | | |
| 10 ⁶ 25% | 26 | 48 | 45 | | | | | |
| 10 ⁷ 4% | 7.6 | 31 | 32 | 103.7 | 16,866 | – 990 | 73,134 | 0 |
| 10 ⁷ 11% | 31 | 32 | 33 | | | | | |
| 10 ⁷ 25% | 19 | 26 | 26 | | | | | |

The median normalized residuals and TIR display great improvement against all LPN realizations (see Table 1). The emulations that used $N_p = 10^6$ and $N_p = 10^7$ realizations as input display a negative amount of empty spaxels, indicating that beyond reconstructing all empty spaxels it also inferred some of the references nulls

To evaluate the spectral reconstruction, we looked at the normalized residuals (see Eq. 18) between the integrated SEDs¹³ of our emulations and of the HPN reference. We also inspected the spatial maps of the compressed features looking for spatial distributions compatible with physical properties of the simulated model. The spatial reconstruction was also evaluated by the median and MAD of the

normalized residuals of our emulations as well as of their LPN inputs²⁶ at each wavelength. For the statistical analysis of the residuals, reference pixels with value 0 were not

²⁶ When calculating residuals for individual spaxels their flux density were previously de-normalized as described in Sect. 3.1.1. This is, however, not the case when calculating residuals for the SEDs that result from the spatial integration of the pixels at each wavelength.

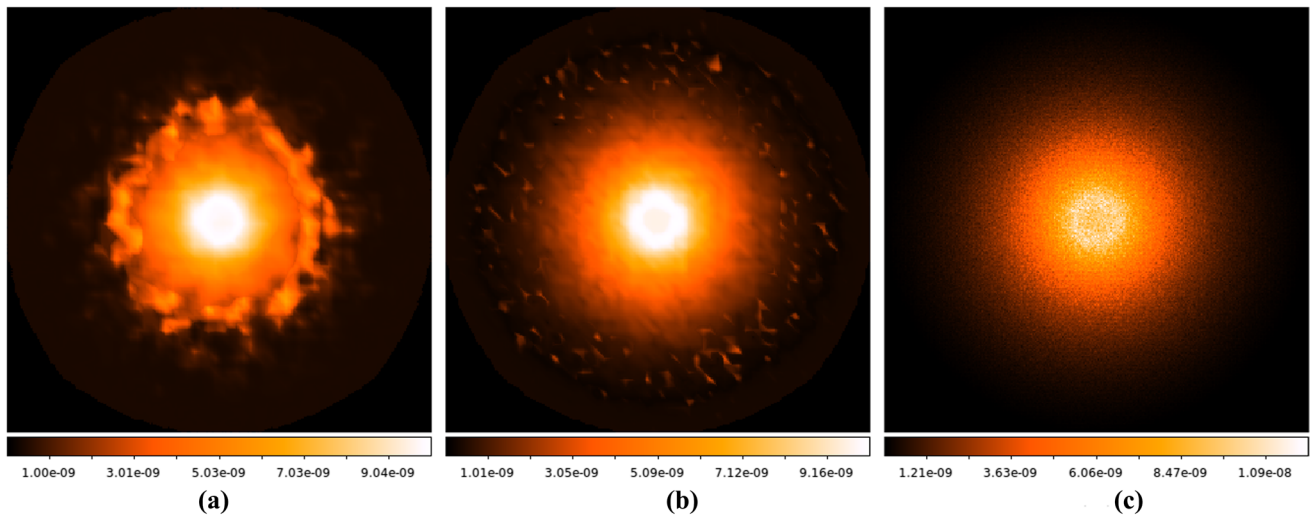


Fig. 8 Spatial maps of emulations, of the case of a dust shell with $\tau_{9.7} = 0.05$ and $\phi = 0^\circ$, at wavelength $9.28 \mu\text{m}$, based on the input of 4% of $N_p = 10^6$ LPN realization spatial information (a), 4% of $N_p = 10^7$ LPN realization spatial information (b) and the HPN reference (c). Color indicates flux density in W/m^2 (Color figure online)

Table 4 Statistics regarding PCA pipeline emulations, using K PCs (where K was determined by the elbow method) of the case of a dust shell with $\tau_{9.7} = 0.05$, $\phi \in \{0^\circ, 90^\circ\}$ and spatial sampling of 11%

| Input | K | Normalized residuals of Emul. (%) | | Residuals of integrated Emul. (%) | | Information of emulation | | | | |
|------------------|-----|-----------------------------------|--|-----------------------------------|-----|--------------------------|-------------------|---------------------------|-------------------|----------------------|
| | | M | | M | MAD | TIR (%) | Null Spxs (N) | Empty Spxs (ΔN) | Full Spxs (N) | Partial Spxs (N) |
| $0^\circ, 10^4$ | 55 | 100 | | 63 | 54 | 53 | 20,712 | 2856 | 0 | 69,288 |
| $0^\circ, 10^5$ | 64 | 99 | | 13 | 13 | 70 | 17,858 | 2 | 10,589 | 61,553 |
| $0^\circ, 10^6$ | 48 | 99 | | 3 | 3 | 77 | 17,051 | - 805 | 2458 | 70,491 |
| $0^\circ, 10^7$ | 31 | 52 | | 1 | 2 | 86 | 16,866 | - 990 | 3820 | 69,314 |
| $90^\circ, 10^4$ | 8 | 97 | | 25 | 44 | 66 | 20,497 | 2641 | 0 | 69,503 |
| $90^\circ, 10^5$ | 55 | 100 | | 18 | 19 | 60 | 18,002 | 146 | 0 | 71,998 |
| $90^\circ, 10^6$ | 48 | 98 | | 3 | 4 | 75 | 16,872 | - 984 | 1322 | 71,806 |
| $90^\circ, 10^7$ | 40 | 48 | | 1 | 2 | 90 | 16,833 | - 1023 | 9346 | 63,821 |

Table 5 Statistics regarding PCA pipeline emulations, using 8 PCs, of the case of a dust shell with $\tau_{9.7} = 0.05$, $\phi \in \{0^\circ, 90^\circ\}$ and spatial sampling of 11%

| Input | Normalized residuals of Emul. (%) | | Residuals of integrated Emul. (%) | | Information of emulation | | | | |
|------------------|-----------------------------------|--|-----------------------------------|-----|--------------------------|-------------------|---------------------------|-------------------|----------------------|
| | M | | M | MAD | TIR (%) | Null Spxs (N) | Empty Spxs (ΔN) | Full Spxs (N) | Partial Spxs (N) |
| $0^\circ, 10^4$ | 96 | | 24 | 52 | 73 | 20,712 | 2856 | 0 | 69,288 |
| $0^\circ, 10^5$ | 100 | | 22 | 15 | 75 | 17,858 | 2 | 3127 | 69,015 |
| $0^\circ, 10^6$ | 64 | | 4 | 6 | 84 | 17,051 | - 805 | 2352 | 70,597 |
| $0^\circ, 10^7$ | 33 | | 1 | 2 | 88 | 16,866 | - 990 | 3122 | 70,012 |
| $90^\circ, 10^4$ | 97 | | 25 | 44 | 66 | 20,497 | 2641 | 0 | 69,503 |
| $90^\circ, 10^5$ | 100 | | 20 | 13 | 70 | 18,002 | 146 | 151 | 71,847 |
| $90^\circ, 10^6$ | 55 | | 2 | 5 | 85 | 16,872 | - 984 | 2884 | 70,244 |
| $90^\circ, 10^7$ | 30 | | 1 | 2 | 91 | 16,833 | - 1023 | 6845 | 66,322 |

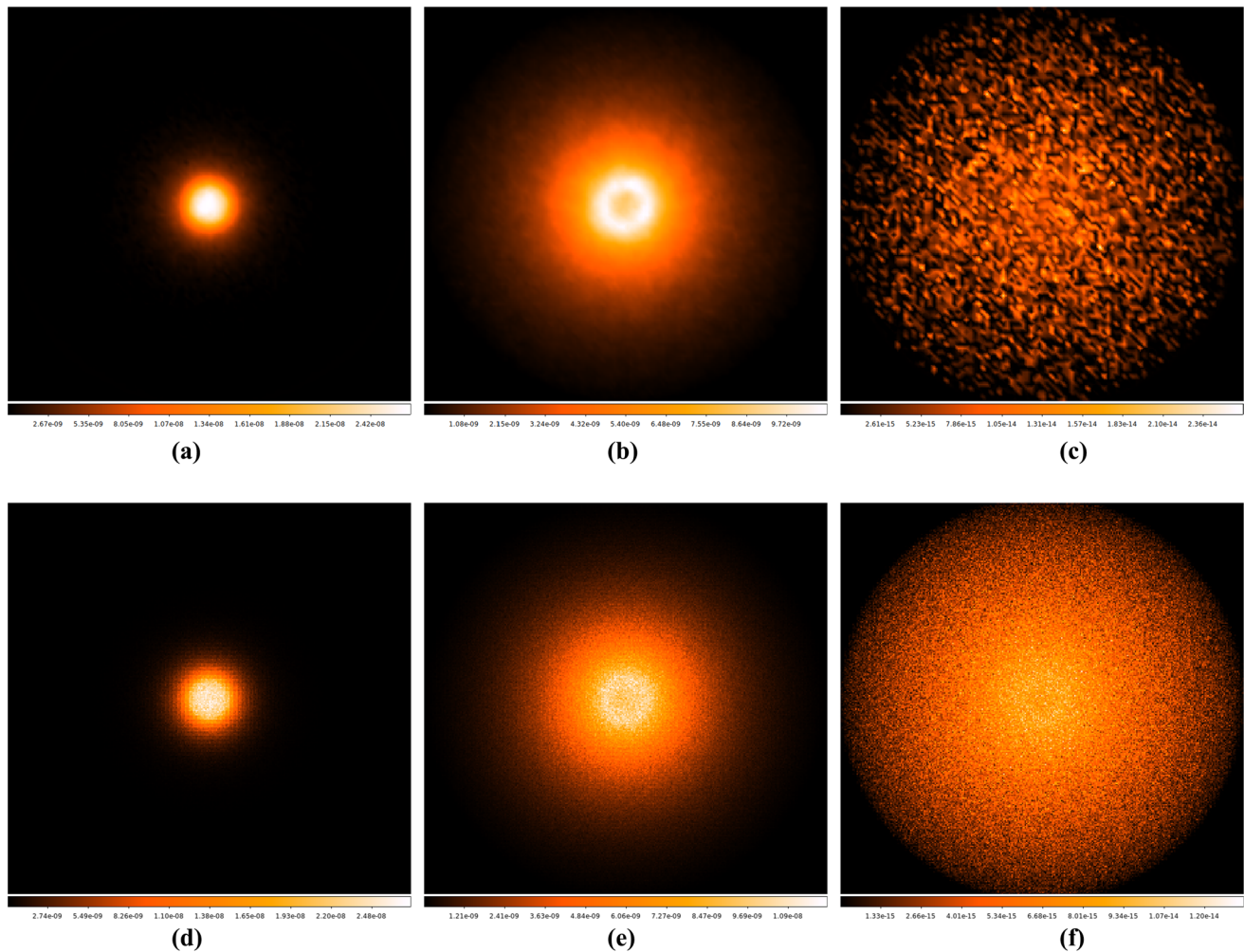


Fig. 9 Spatial maps of emulations (Top), performed with a pipeline including a PCA model with $K = 30$ components based on the input of 11% of the spatial information of the $N_p = 10^7$ LPN realization; and, reference simulation (Bottom) of the case of a dust shell with

$\tau_{9.7} = 0.05$ and $\phi = 0^\circ$, at wavelengths $1.85 \mu\text{m}$ (**a, d**), $9.28 \mu\text{m}$ (**b, e**) and $211.35 \mu\text{m}$ (**c, f**). Color indicates flux density in W/m^2 (Color figure online)

considered since this metric diverges, so the TIR for all emulations and simulations was calculated as well.

4.1 AEsSet predictions

In this section, we present and discuss the results obtained emulating the HPN reference of AEsSet using the different LPN realizations within it.

The upper panels of Fig. 7 show that the emulation-integrated SEDs reproduce the shape of the references: a slow rise in the $1\text{--}8 \mu\text{m}$ range, the two emission bumps in the $8\text{--}20 \mu\text{m}$ range and the steep decline towards longer wavelengths. Moreover, Table 2 displays the median and MAD of the residuals of the integrated SEDs for the LPN input realizations before and after being de-normalized (as we describe in Sect. 3.1.1), as well as those of the different emulations obtained from them. It is clear that using

$N_p \geq 10^6$ realizations as input yields emulations integrated SEDs that closely (median residuals smaller than 15%) follow the references throughout the whole wavelength range, independently (within this subset) of the sampling percentage chosen for the spatial inference task.

From the residuals of the emulations integrated SEDs, shown in the lower panels of Fig. 7, we conclude that: shorter wavelengths yield higher residuals; more input data for the spatial reconstruction does yield a better emulation but at the cost of an increased run time,²⁷ as can be confirmed in Table 3 and that the usage of the $N_p = 10^6$

²⁷ INLAs reconstruction is as influenced by the amount of data it takes as input as by how that data is spatially distributed. Though on average having a larger uniform sample will be better than having a smaller uniform sample, it is possible that a particular smaller sample exists with a distribution that better captures the information of the field and that yields a better reconstruction. In the present case, we avoid this variability by using a regular sampling grid.

realization as input greatly improves the quality of the emulation in relation to the two lowest photon number alternatives.

Looking at Tables 1 and 3, we can also compare the overall performance of the pipeline at estimating information that was not available in the LPN input realizations. The amounts of differently classified spaxels in both LPN inputs and respective emulations show, together with median of the normalized residuals, that EmulART successfully estimates information missing from the input.

Figure 8 shows the comparison, at wavelength 9.28 μm , between the emulations resulting from the LPN inputs (see Fig. 2) and the HPN reference. Once again we can see that with the $N_p = 10^6$ realization the emulations start to display resemblances to the HPN reference not only in the range of flux density values but also in the morphology that emerges from their distribution.

From Tables 2 and 3, as well as from Figs. 7 and 8, it would be natural to conclude that using the $N_p = 10^7$ realization as input would bring the most benefit in terms of the amount of information inferred by the emulation as well as its accuracy. Moreover, as can be seen in the second column of Table 3, the run time of the emulations is more dependent on the amount of information sampled for the spatial reconstruction than on the N_p of the LPN input. Nevertheless, considering how SKIRT's run time for a model scales with the simulated N_p , the choice of LPN input to use with EmulART should weight the time it takes to produce that LPN input as well as the quality of the emulation we expect from it.

4.1.1 PCA Pipeline predictions

The results of testing the PCA pipelines on AESet were processed in the same way as EmulARTs (statistical indicators regarding residuals of the emulation and of its spatial integration were calculated as well as the TIR; the number of different types of spaxels and execution time were measured).

The execution times²⁸ for the pipeline implementing an 8 PC model were indiscernible from the execution times of EmulART on the same input, while the execution time for the pipeline implementing a PCA model with number of components, K , determined by the elbow method was in

²⁸ These do not include the time it took to train the PCA models. Training time for each PCA model was below 10 s, while training the DVAE took around 18 hours ($\sim 6500\times$ more). This was expected due to the methods themselves as well as the differences between training procedures. The DVAE model was trained with $6\times$ the amount of data as each PCA model, with the purpose of being able to generalize across simulations resulting from different photon packet numbers, while each PCA model was trained on LPN simulation that was used as input for emulation, to compared against the results of EmulART.

general much higher since for all except one of the LPN inputs K was larger than 30 (the time per component map was very similar depending mostly on the amount of spatial information sampled).

Tables 4 and 5 present the most significant indicators to be compared to EmulARTs. At first sight, the performance of both PCA models on the emulation pipeline seems to be similar to, and in some cases even better than that of EmulARTs, showing statistically similar residuals, and presenting residuals regarding the spatially integrated SEDs 2 to 3 times lower. They are, however, unable to consistently recover complete spaxel information, which then leads to poor spatial reconstructions at longer wavelengths, even when using $K > 8$ (see Fig. 9), as can be inferred from the TIR values (as well as the number of full and partial spaxels).

In the present application, we thus find the implementation of a DVAE model for spectral compression to be justified. Unlike the PCA models, it not only captures nonlinear relationships between the spectral features but also achieves comparable results. Despite some loss regarding the reconstruction of integrated SED profile, when compared to PCA models, it achieves equal/higher compression rate and lower/equal execution time and most importantly the spatial structure can successfully be reconstructed by the remaining parts of the pipeline.

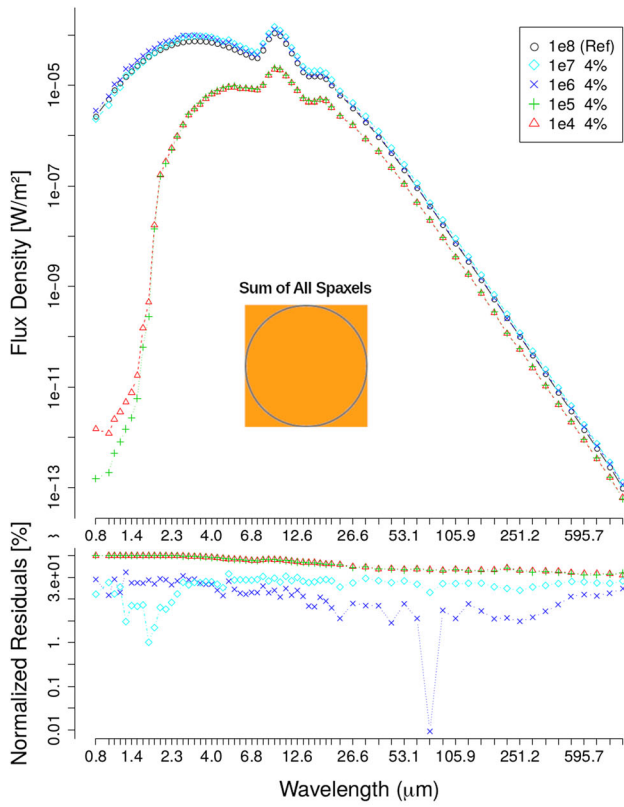
As such, the test results of EmulART on EVASet are compared against its results on AESet. Testing results obtained with PCA emulation pipelines on EVASet did not offer a perspective different from the one above and as such will not be discussed further here (a successful implementation of PCA in an emulation pipeline is described in Smole et al. [46]).

4.2 EVASet Predictions

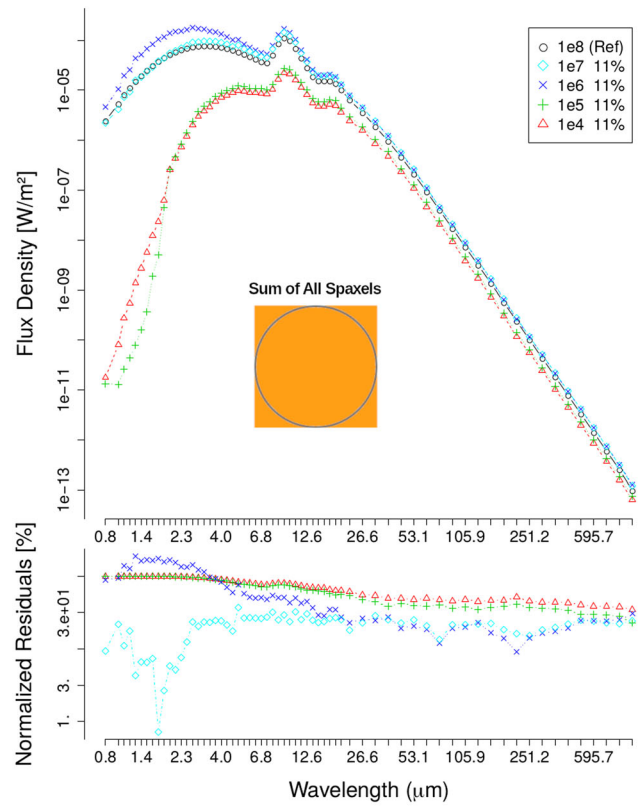
In this section, we present and discuss the results obtained by emulating the HPN references of EVASet using the respective LPN realizations within it. The DVAE model was not trained on any data within this set, which allows us to evaluate whether it manages to accurately predict HPN-like spaxels from LPN spaxels that originate from simulations with different $\tau_{9,7}$ and ϕ values.

First, we tested EmulART on realizations with $\tau_{9,7} = 0.05$ and $\phi = 90^\circ$; we then applied the pipeline to different LPN realizations with $\tau_{9,7} \in \{0.1, 1.0\}$ and $\phi \in \{0^\circ, 90^\circ\}$.

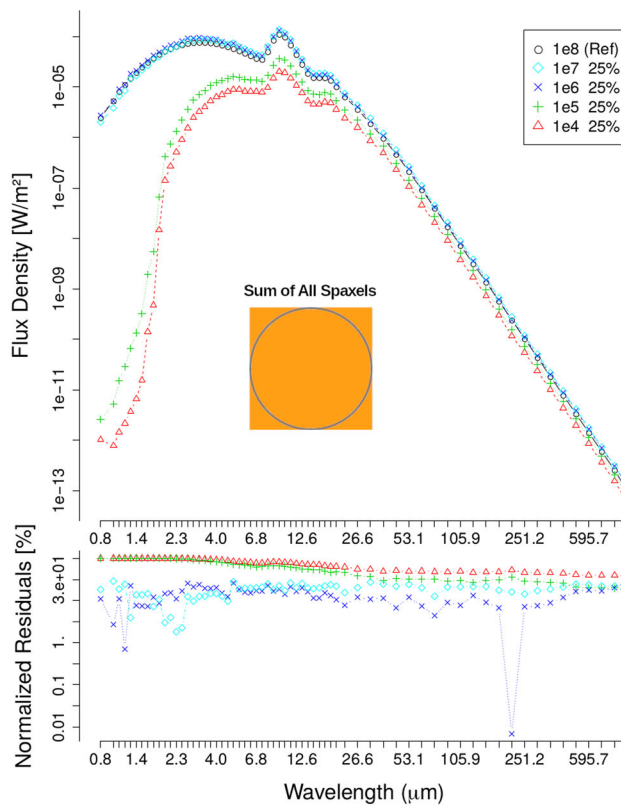
Similarly to the $\tau_{9,7} = 0.05$ and $\phi = 0^\circ$ emulations, the edge-on, $\phi = 90^\circ$, emulations appear to preserve well the spectral information, reproducing the slow rise in the 1–8 μm range, the two emission bumps in the 8–20 μm range, and the steep decline towards longer wavelengths, as can be seen in Fig. 10. Table 6 shows that 4% sampling of the



(a)



(b)



(c)

◀**Fig. 10** Emulation-integrated SEDs, resulting from spatial inference using 4% (a), 11% (b) and 25% (c) samples of the spatial information, and the respective normalized residuals, for the case of a dust shell with $\tau_{9.7} = 0.05$ and $\phi = 90^\circ$. The HPN reference is represented in black (\circ), the emulation based on the $N_p = 10^4$ realization is in red (Δ), on the $N_p = 10^5$ in green (+), on the $N_p = 10^6$ in blue (\times) and on the $N_p = 10^7$ in cyan (\diamond) (Color figure online)

spatial information of the $N_p = 10^6$ realization is enough to get median-integrated residuals below 15%. We note, however, the abnormal performance of the emulations that took as input the $N_p = 10^7$ realizations, displaying higher median integrated residuals than the ones that used different samplings of the $N_p = 10^6$ LPN. This may indicate that one or more of the spatial data manipulation modules, or their interface, should be improved upon.

As for the emulations using as input LPN realizations of simulations with $\tau_{9.7} \in \{0.1, 1.0\}$, at both tilt angles, we observe that both the shape and flux density value range of the integrated SED degrade as $\tau_{9.7}$ increases. As shown in Fig. 11, the emulation-integrated SEDs fail to reproduce the shape of the HPN references, reproducing instead the shape that characterized the realizations present in AEsSet, a clear sign of over-fitting of the DVAE model. This can be solved by expanding the training set of our DVAE architecture to include spaxels originating from simulations with different optical depths.

For $\tau_{9.7} = 1.0$, with both ϕ cases, we observe the influence of the first wavelengths (see Fig. 12) in the overall residuals.²⁹ Figure 13 shows that though the overall morphology of the spatial distribution is well recovered the value range for the emulations flux density value range is drastically underestimated, while the contrast between the central and peripheral regions is considerably higher than what the HPN references display.

These results appear to show that our pipeline is capable of recovering 40–60% of the emergent spatial information of HPN MCRT models from LPN realizations, taking as input as little as 0.04% of the information that would be present in the HPN model, all while preserving 85–95% of the spectral information.

Furthermore, the results also show a clear bias in the performance of the DVAE model as a compressor and decompressor of spectral information, with the performance degrading substantially as the LPN inputs models depart from the optical depth, $\tau_{9.7}$, value present within the training set.

Further details of the results we obtained with AEsSet and EVASet are discussed in Appendix E.

Table 6 Comparison of statistics for the residuals of the spatial integration SEDs, for the case of a dust shell with $\tau_{9.7} = 0.05$ and $\phi = 90^\circ$, for the different LPN realizations (columns 2 and 3), for

those same realizations but after they have been de-normalized (columns 4 and 5), as described in Sect. 3.1.1, and for the resulting emulations (columns 7 and 8)

| LPN Input | Residuals of integrated input (%) | | Residuals of integrated de-normalized input (%) | | Sample (%) | Residuals of integrated Emul. (%) | |
|-----------|-----------------------------------|------|---|------|------------|-----------------------------------|-----|
| | <i>M</i> | MAD | <i>M</i> | MAD | | <i>M</i> | MAD |
| 1e4 | 13.4 | 7.4 | 99.99 | 0.00 | 4 | 77 | 19 |
| | | | | | 11 | 16 | 19 |
| | | | | | 25 | 78 | 19 |
| 1e5 | 1.9 | 2.0 | 99.90 | 0.00 | 4 | 76 | 20 |
| | | | | | 11 | 71 | 23 |
| | | | | | 25 | 63 | 25 |
| 1e6 | 0.83 | 0.51 | 99.00 | 0.01 | 4 | 13.2 | 7.4 |
| | | | | | 11 | 46 | 41 |
| | | | | | 25 | 13.1 | 5.0 |
| 1e7 | 0.19 | 0.17 | 90.00 | 0.03 | 4 | 23.7 | 5.7 |
| | | | | | 11 | 22.7 | 5.7 |
| | | | | | 25 | 19.6 | 4.9 |

²⁹ For more details, consult the data products available at repository.

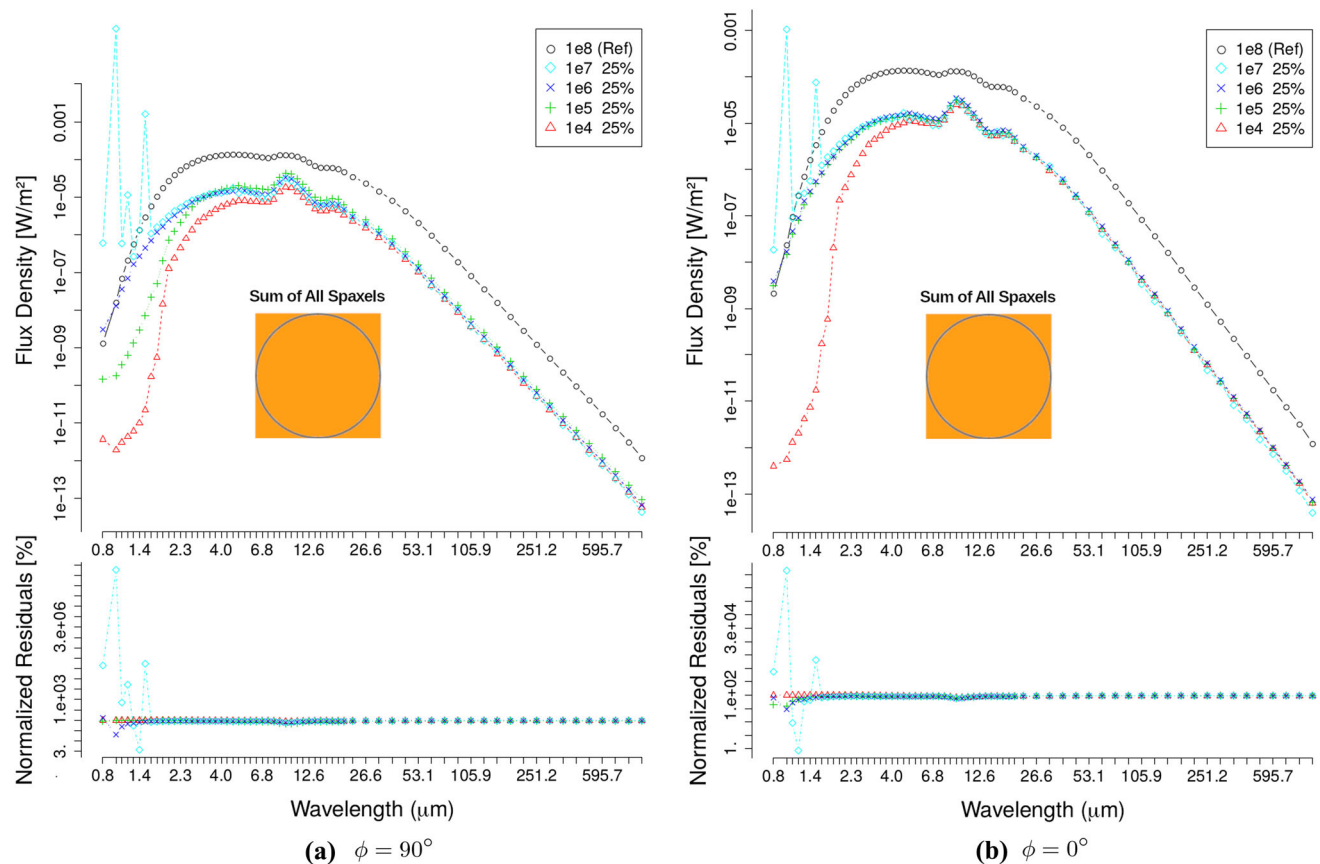


Fig. 11 Emulation-integrated SEDs, resulting from spatial inference using 25% of spatial information, and the respective normalized residuals, for the case of a dust shell with $\tau_{9.7} = 1.0$, $\phi = 90^\circ$ (a) and $\phi = 0^\circ$ (b). The HPN reference is represented in black (o), the

emulation based on the $N_p = 10^4$ realization is in red (Δ), on the $N_p = 10^5$ in green (+), on the $N_p = 10^6$ in blue (\times) and on the $N_p = 10^7$ in cyan (\diamond) (Color figure online)

5 Summary

We report the development of a pipeline that implements in conjunction a denoising variational autoencoder (or alternatively PCA), and a reconstruction method based on approximate Bayesian inference, INLA, with the purpose of emulating high information-like MCRT simulations using LPN MCRT models, created with SKIRT, as input. With this approach, we aim for the hastily expansion of libraries of synthetic models against which to compare future observations. By producing positive preliminary indicators, we show that such a framework is worth pursuing further, with multiple alleys to explore.

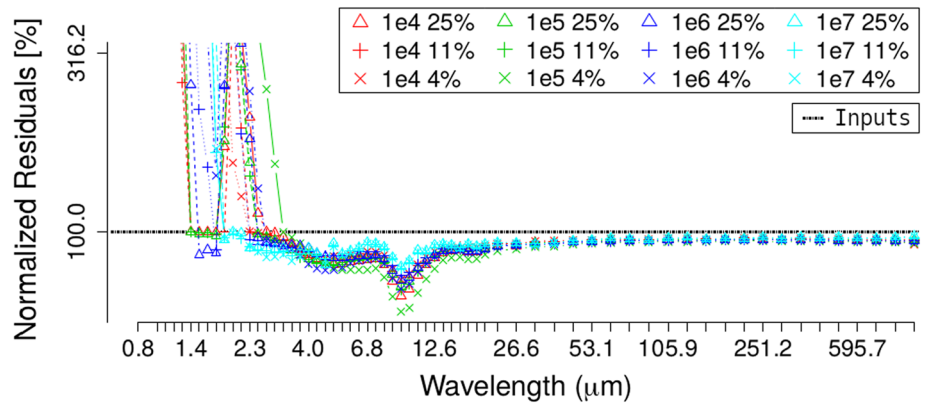
Conditions for systematically measuring the computational cost are necessary to properly evaluate the merit of this approach. However, in this work our aim was to qualitatively assess the potential of this method to be applied to MCRT simulations images. In this pilot study, we chose a very simple model of a centrally illuminated spherical dust shell, which is computationally inexpensive,

whose reference simulations took around two hours to be computed. Thus, in our particular examples we reduced the computational time by approximately $6\times$. Nevertheless, the computational cost of SKIRT simulations scales with the amount of photon packets simulated, the spatial resolution of the grid and the actual geometrical complexity of the model. As for our emulation pipeline, its computational cost is only impacted noticeably when increasing the size and density of the spatial grid to be processed by INLA. This leads us to believe that a generalized version of this pipeline may expedite, by up to $50\times$, the study of dust environments through this kind of radiative transfer models.

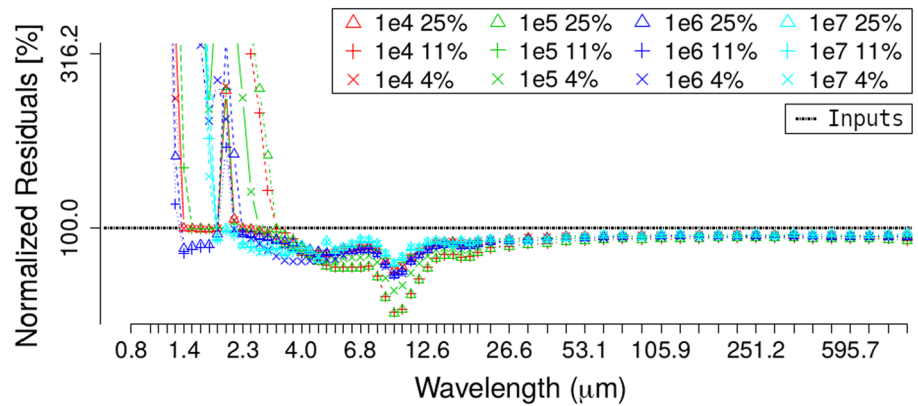
Further exploration of the proposed DVAE architecture is being undertaken, via expansion and diversification of the training set to improve the prediction of the spectral features.

Other approaches to be tested include the use of dropout (cutting the connections, mid training, whose weights are below a given threshold) and incremental learning (training

Fig. 12 Median of the normalized residuals, at each wavelengths spatial map, for every emulation obtained with $N_p = 10^4$ (red), $N_p = 10^5$ (green), $N_p = 10^6$ (blue) and $N_p = 10^7$ (cyan) realizations, for the case of a dust shell with $\tau_{9.7} = 1.0$, with $\phi = 0^\circ$ (a) and $\phi = 90^\circ$ (b). Emulations whose spatial inference was performed using 25% of data are represented by (Δ), 11% by (+) and 4% by (\times). The interrupted black line marks the same metric for the LPN inputs (Color figure online)



(a) $\phi = 0^\circ$



(b) $\phi = 90^\circ$

a model with new data with the starting point of the network being the weights obtained in a previous training session). The first serves as feature selection tool, removing those of little importance, which also prevents the model from over-fitting the training data; the second would be of great use to quickly adapt an already trained model to new data, which is important in the context of emulating simulations.

To improve the reconstruction of the spatial features, using non-uniform sampling methods based on the information spatial density may help improve the reconstruction of the latent features that result from the compression of models simulated with insufficient number of photon ($N_p < 10^6$, in the particular case of our study). Alternative pre-INLA data preprocessing, such as data value range manipulations and sampling grids, may also be worth exploring.

Appendix A: Data preprocessing

Appendix A.1: Identifying and sampling null-spaxels

The SKIRT simulations of the spherical simulated models result in a peripheral region filled with spaxels with value 0 at all wavelength bins; these will be henceforth be referred to as null-spaxels. We perform two routines to identify and filter out null-spaxels, to curb their potential impact in both training the DVAE model and when (within the emulation procedure) reconstructing the spatial information of the latent feature maps:

1. Each feature (wavelength) of each spaxel is checked, if all features of a spaxel have value 0, then that spaxel is temporarily tagged as null-spaxel;
2. An iterative process then checks the immediate neighborhood of each previously tagged spaxel (a $5 \text{ pix} \times$

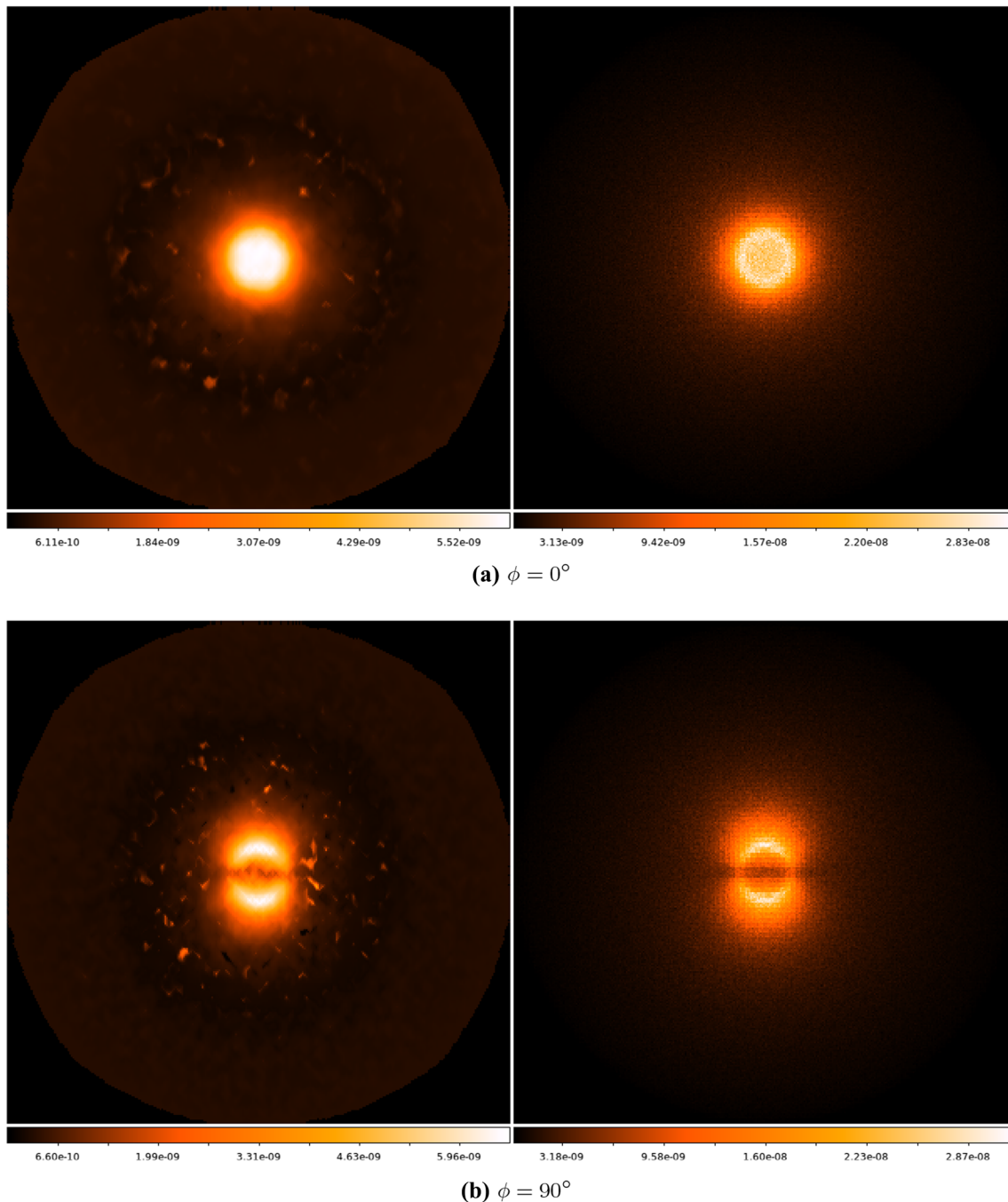


Fig. 13 Emulation produced using as input a 25% sample of the $N_p = 10^6$ realization (left panel) and HPN reference (right panel) at wavelength $9.28 \mu\text{m}$, for the case of a dust shell with $\tau_{9.7} = 1.0$, $\phi = 0^\circ$ (**a**) and $\phi = 90^\circ$ (**b**)

5 pix grid centered at the spaxel being tested) for other tagged spaxels, if a minimum of 4 neighbors share the null-spaxel tag, then that spaxel is tagged as a true null-spaxel, otherwise its temporary tag is removed.

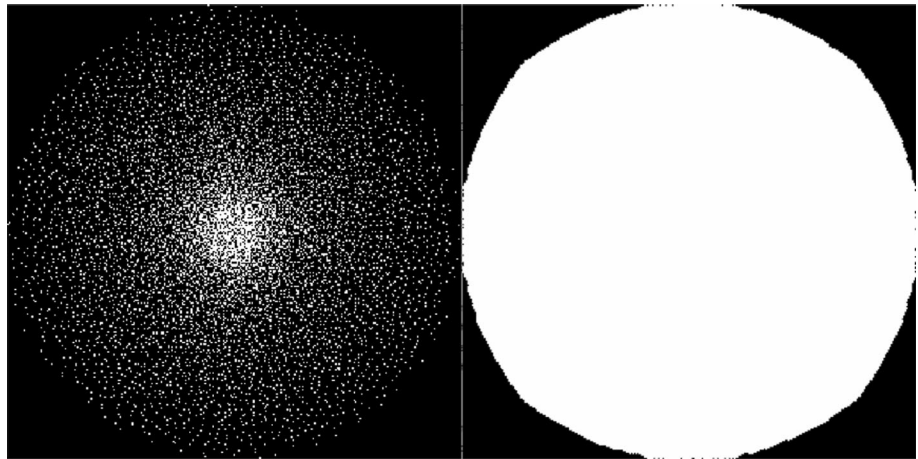
The second routine serves as a way to discriminate spaxels that have zero information at given wavelengths, possibly from the spaxel being the result of a low photon count simulation, from actual null-spaxels—spaxels resulting

from the lack of photons traveling from the source to the observer through that direction, neither by direct emission nor by scattering nor by absorption and re-emission by dust. Figure 14 illustrates the results of routine 1 and 2.

After identifying the null-spaxels, we sample them to incorporate the filtered dataset.

For the DVAE training, complete removal of these null-spaxels could be done, but we decided to reduce their

Fig. 14 On the left, in black, the spaxels temporarily tagged as null-spaxels by routine 1; on the right, also in black, the final selection of null-spaxels that resulted from applying routine 2 with a search grid of $5 \text{ pix} \times 5 \text{ pix}$ around each temporary null-spaxel. The result on the right shows that for this particular case of a spherical shell of dust surrounding an isotropic emitter we only find null-spaxels on the outskirts of the shell, as expected



incidence rate instead. Since there is no guarantee that their latent representation (where the spatial reconstruction will take place) will also be a null-vector we want the DVAE model to learn null-spaxels, too many of these will nevertheless be uninformative and may lead to unintended biases. For this reason, we sample 10% of them, reducing the null-spaxel prevalence in the training dataset from $\sim 25\%$ to $\sim 2.5\%$. Empty spaxels (spaxels tagged as null for LPN realizations but non-null for the respective HPN reference) were completely removed from training.

For the emulation process, and more specifically for the spatial reconstruction of the latent feature maps, we sample 10% of the null-spaxels selected by the spatial sampling grid, for INLA is faster with little data and that amount is sufficient for the null-region to be properly recovered.³⁰ After the emulation process, spaxels that were at this stage identified as being null-spaxels have their values changed to 0.

Appendix A.2: Reshaping data structures

Once all spaxels are selected the data cubes are reshaped into a 2D structure, with a given pixel corresponding to a row with its spectral information on the columns (see Fig. 5), to match the input format of the DVAE. The inverse reshaping, from a 2D to a 3D structure, is performed after the DVAE output layer.

Within the emulation pipeline, the data are reshaped two more times. First, after sampling the latent feature distributions (to obtain the Z arrays, as seen in Fig. 5) the resulting sample is reshaped from a 2D to a 3D structure to obtain latent feature spatial maps (the 8 1D arrays become 8 2D maps) whose spatial information can then be reconstructed with INLA. The second reshape is performed on INLAs reconstructed results to conform that data to the

decoder accepted input format, once again going from a 3D to a 2D structure.

Appendix B: Dataset

Table 7 presents the list of SKIRT simulations used in this work. Every realization models a spherical dust, composed by silicates and graphites, surrounding a bright point source with anisotropic emission, differing in parameters such as the tilt angle, ϕ , the optical depth, $\tau_{9,7}$, and the amount of photon packets simulated, N_p .

Appendix C: Latent Features

In this section, we briefly discuss the relationship between each of the compressed features, as well as how they may relate to some physical attributes of the simulated models.

The PCC measures how strongly linearly correlated different parameters are and is defined as the ratio between the covariance of the parameters and the product of their standard deviations (see Eq. C1):

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}. \quad (\text{C1})$$

Figure 15 illustrates the relationships between our datasets latent features along with their respective PCC. The PCC indicates that only the first feature shows little correlation with the others, ranging from -0.42 to 0.39 , while the remaining features display high correlations (and anti-correlations) between themselves, most either above 0.9 or below -0.8 . This would indicate that the information in those features presents a high level of redundancy.

The point spread in the panels of Fig. 15 shows, however, something unexpected given the PCC values. There are clearly distinct populations, some presenting linear

³⁰ Different simulated objects with different distributions of regions with 0 emission may need a higher sampling of null-spaxels.

Table 7 Parametrical description of the SKIRT realizations used in this work, along with the distribution of these realizations by subset

| N | $\tau_{9,7}$ | ϕ (deg) | N_p | Subset |
|-----|--------------|--------------|--------|--------|
| 1 | 0.05 | 0 | 10^4 | AESet |
| 2 | 0.05 | 0 | 10^5 | AESet |
| 3 | 0.05 | 0 | 10^6 | AESet |
| 4 | 0.05 | 0 | 10^7 | AESet |
| 5* | 0.05 | 0 | 10^8 | AESet |
| 6 | 0.05 | 90 | 10^4 | EVASet |
| 7 | 0.05 | 90 | 10^5 | EVASet |
| 8 | 0.05 | 90 | 10^6 | EVASet |
| 9 | 0.05 | 90 | 10^7 | EVASet |
| 10* | 0.05 | 90 | 10^8 | EVASet |
| 11 | 0.1 | 0 | 10^4 | EVASet |
| 12 | 0.1 | 0 | 10^5 | EVASet |
| 13 | 0.1 | 0 | 10^6 | EVASet |
| 14 | 0.1 | 0 | 10^7 | EVASet |
| 15* | 0.1 | 0 | 10^8 | EVASet |
| 16 | 0.1 | 90 | 10^4 | EVASet |
| 17 | 0.1 | 90 | 10^5 | EVASet |
| 18 | 0.1 | 90 | 10^6 | EVASet |
| 19 | 0.1 | 90 | 10^7 | EVASet |
| 20* | 0.1 | 90 | 10^8 | EVASet |
| 21 | 1.0 | 0 | 10^4 | EVASet |
| 22 | 1.0 | 0 | 10^5 | EVASet |
| 23 | 1.0 | 0 | 10^6 | EVASet |
| 24 | 1.0 | 0 | 10^7 | EVASet |
| 25* | 1.0 | 0 | 10^8 | EVASet |
| 26 | 1.0 | 90 | 10^4 | EVASet |
| 27 | 1.0 | 90 | 10^5 | EVASet |
| 28 | 1.0 | 90 | 10^6 | EVASet |
| 29 | 1.0 | 90 | 10^7 | EVASet |
| 30* | 1.0 | 90 | 10^8 | EVASet |

Realizations marked with (*) were not parsed as input to EmulART while testing the pipeline, being used merely as the HPN reference for the purpose of computing performance metrics

behavior, while others show highly nonlinear relationships (see, for example, Z3 vs Z5) and even dual linear populations (see Z6 vs Z7).

The overall large Pearson correlation values thus indicate that the population density within the linear spreads is higher. But even if the compressed features show at best linear correlations (and anti-correlations) for a greater fraction of our data, we decided not to ignore the nonlinear behavior being expressed for the remaining data. In future implementations, we will tackle this by both expanding the dataset, including models with different dust distributions

and compositions and employing different training strategies (this can be achieved by changing loss function terms and adding different training stages).

Figure 16 shows the spatial reconstruction of 3 latent feature maps³¹ obtained from the $N_p = 10^6$ LPN realization of the case of a dust shell with $\tau_{9,7} = 0.05$ and $\phi = 90^\circ$.

In the map of latent feature Z1 (Fig. 16a), we can relate the spatial distribution to the hot dust emission, at wavelengths 1–3 μm , of the region close to the sublimation zone around the central source. Conversely, in the map of latent feature Z2 (Fig. 16b), the larger structure further away from the center, which corresponds to the colder dust emission at longer wavelengths is outlined.

As mentioned in Sect. 3.1, the model used along this work is that of a bright point like source with anisotropic emission [47], where the emission is defined by Eq. (17), following Netzer [36]. Looking at the map of latent feature Z8 (Fig. 16c), the geometric signature of the model is clearly identifiable.

Appendix D: Emulation flowcharts

This section presents flowcharts for different emulation pipelines used in this work. In these, circles and ellipses represent data cubes at different stages of the pipeline and round cornered squares present lists of indexes. Grey rectangle boxes are operations using the data, purple diamond boxes indicate set operations (\cap for intersection and \cup for union) on indexes, green boxes indicate operations related to dimensionality transformation models. \blacklozenge refers to the de-normalization procedure mentioned in Sect. 3.1.1 and \dagger refers to operations described in Sect. 3.3 (Figs. 17 and 18).

Appendix E: Result analysis

This section presents a detailed view of the results obtained with both AESet and EVASet. As mentioned in Sect. 4, for every $\tau_{9,7}$ and ϕ case, the realization with $N_p = 10^8$ was considered as the HPN reference, or “ground truth”. The remaining LPN realizations and the emulations obtained from those were compared to the HPN reference in regard to their spectral and spatial features.

For the spectral analysis, we computed the normalized residuals (Eq. 18) of the spatially integrated SEDs, compared the shape of those SEDs and inspected the spatial

³¹ Though our pipeline yields 8 latent features for every spaxel, we chose to only show these 3 maps since the remaining latent feature maps either were very similar to these or no other information could be extracted from them.

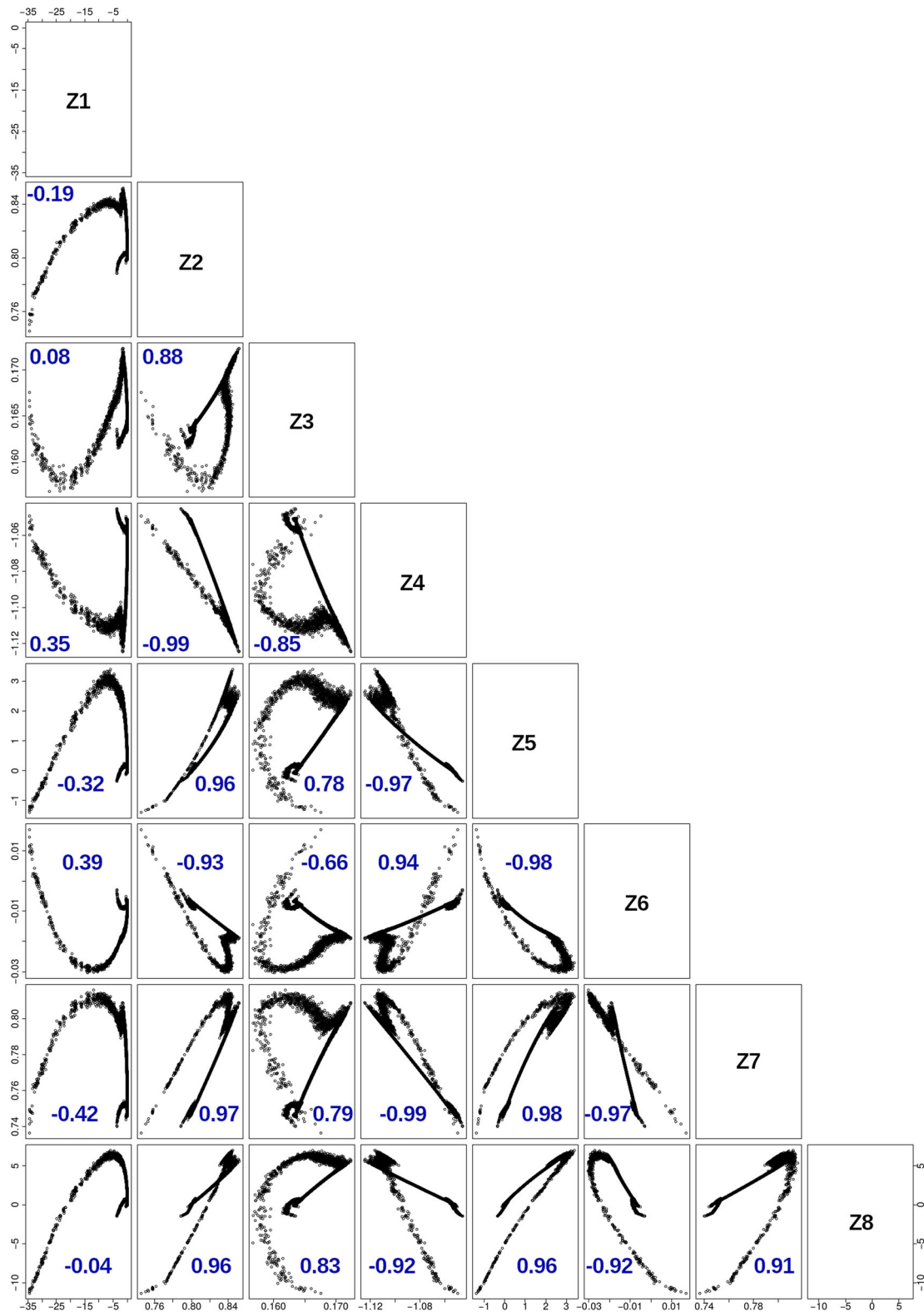


Fig. 15 Corner plot of the latent features, Z_f of the test set. Pearson's correlation coefficients are indicated in navy blue (Color figure online)

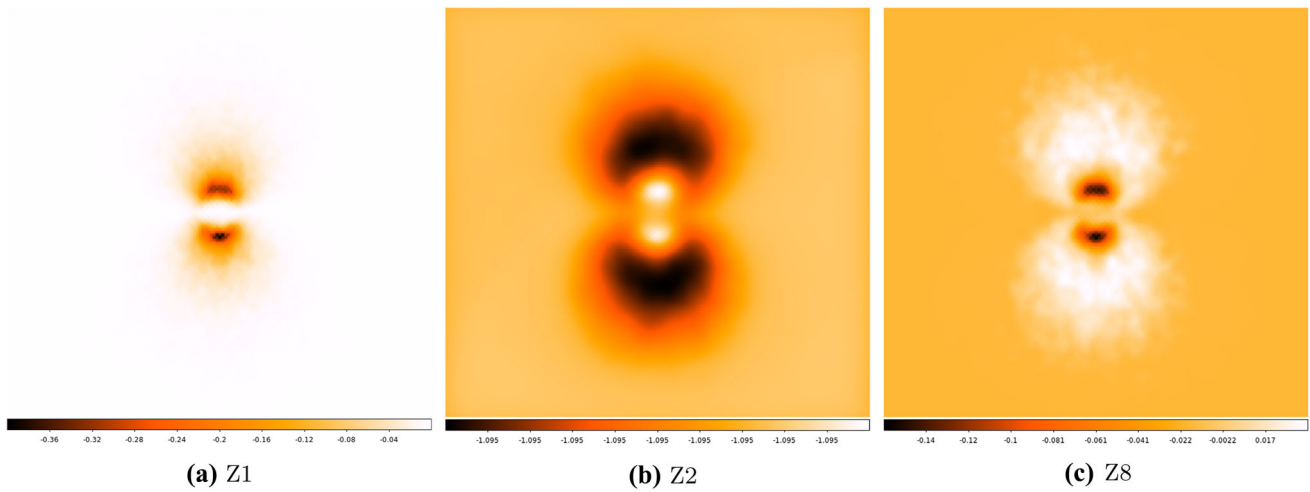


Fig. 16 Spatial reconstruction of 3 latent feature maps of the LPN case $N_p = 10^6$, $\tau_{9,7} = 0.05$ and $\phi = 90^\circ$, using 25% of the available data

Fig. 17 Flowchart for EmulART

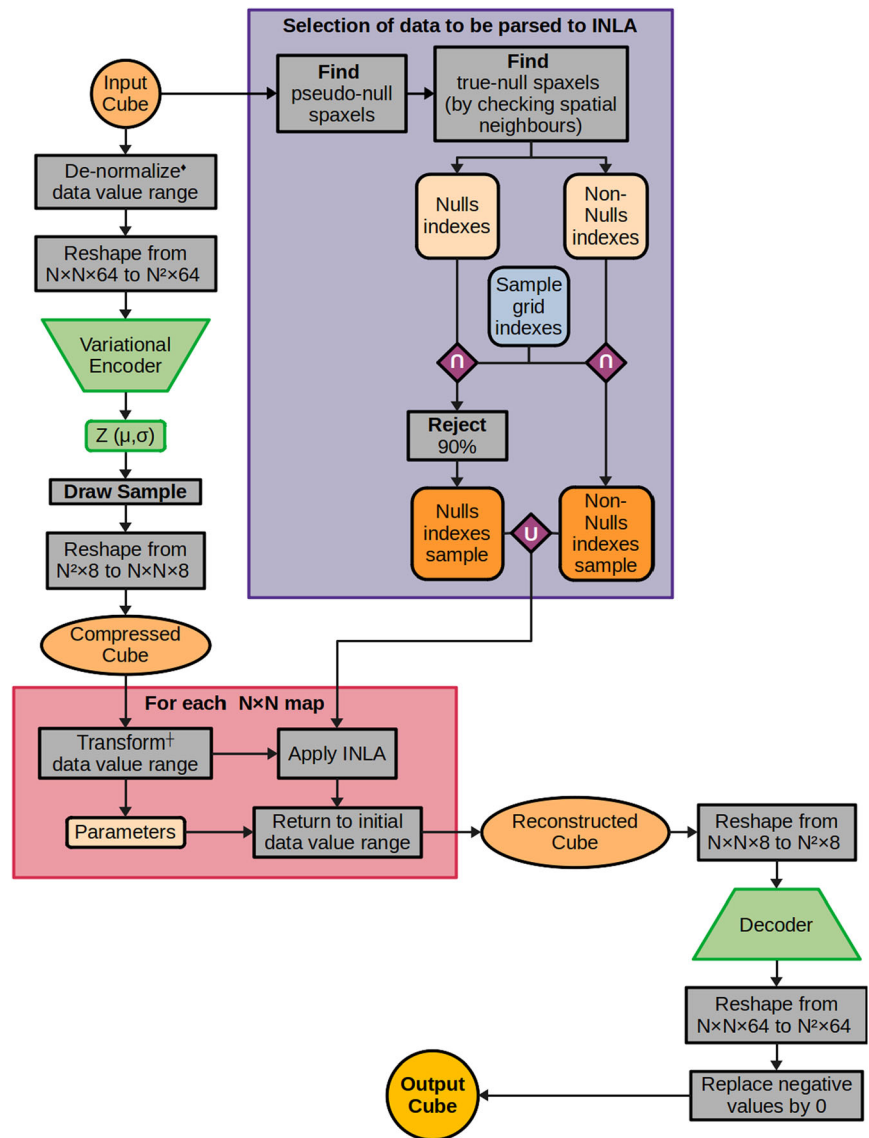
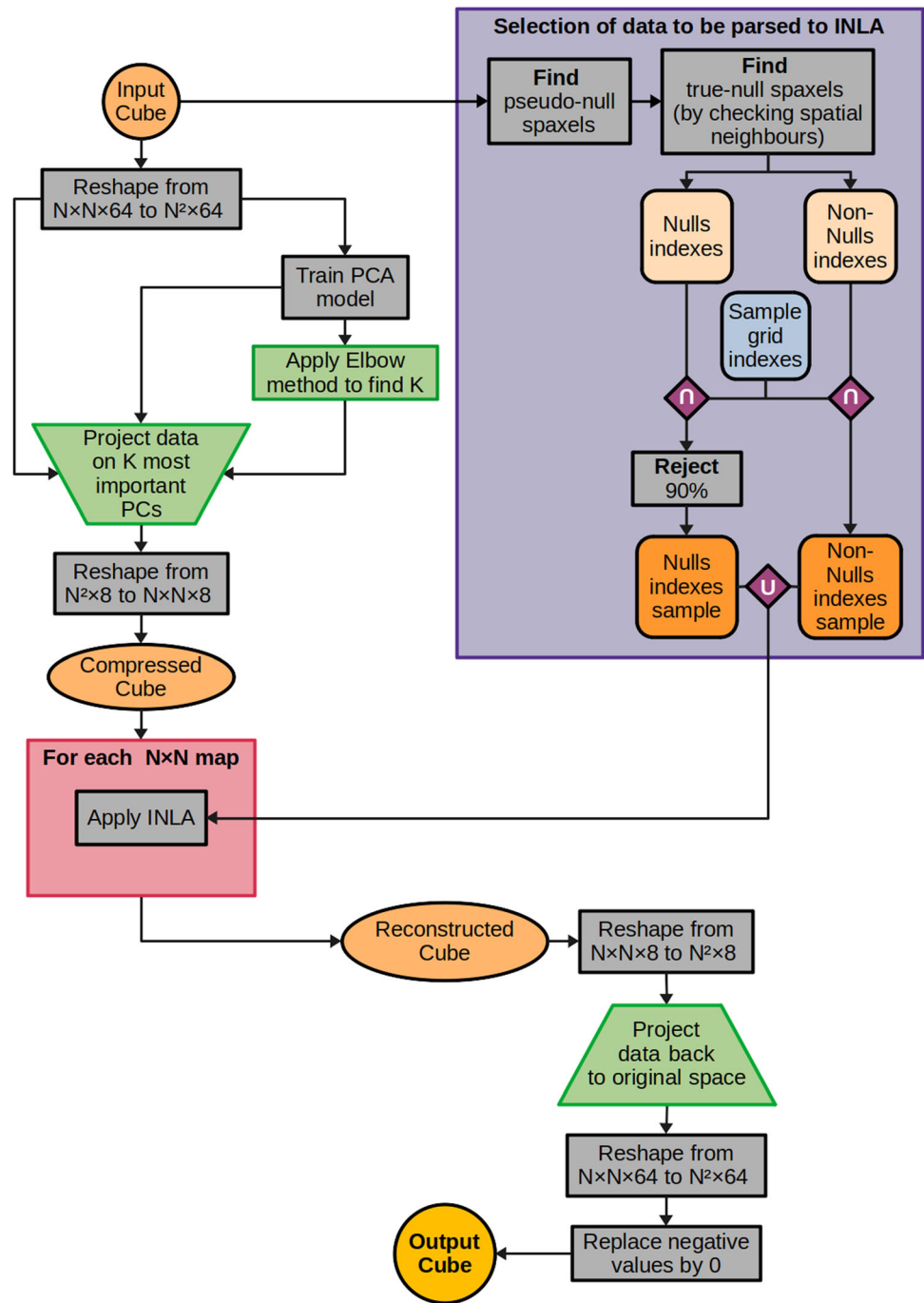


Fig. 18 Flowchart for the emulation pipeline that employs the PCA model with the number of components, K , determined by the elbow method. For the emulation pipeline that employs the DVAE model in EmulART ($K = 8$), the Elbow method block is skipped



maps of the DVAE-compressed features looking for distributions compatible with the physical properties of the HPN reference.

For the spatial analysis, we computed the normalized residuals and measured statistical descriptors—median and mean absolute deviation (MAD)—for each wavelength bin spatial map, for both LPN inputs and resulting emulations. To measure the impact of INLA on the reconstruction of input spaxels, we count how many full, partial and null

spaxels are present in the HPN reference, LPN input and the resulting emulations, and compute the TIR of each.

Appendix E.1: AEsSet predictions

Appendix E.1.1: Spectral predictions

In this section, we discuss how well the spectral information is preserved, after the LPN inputs of AEsSet are

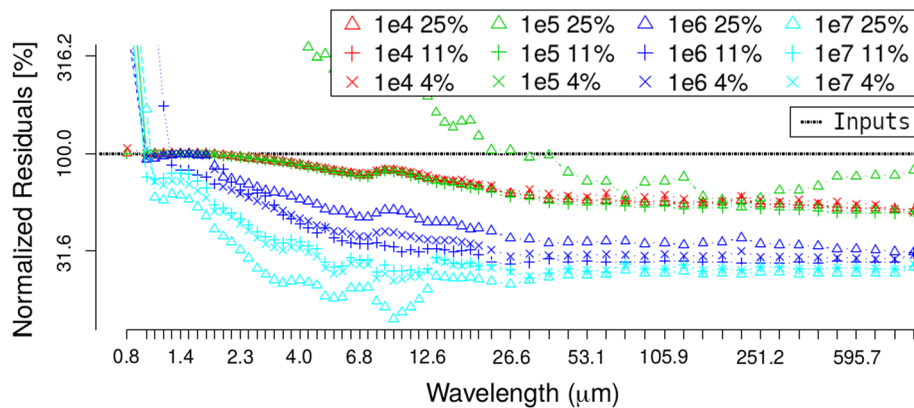


Fig. 19 Median of the normalized residuals, at each wavelength's spatial map, for every emulation obtained with $N_p = 10^4$ (red), $N_p = 10^5$ (green), $N_p = 10^6$ (blue) and $N_p = 10^7$ (cyan) realizations, for the case of a dust shell with $\tau_{9.7} = 0.05$ and $\phi = 0^\circ$. Emulations whose

spatial inference was performed using as input 25% of data are represented by (Δ), 11% by (+) and 4% by (\times). The interrupted black line marks the same metric for the LPN realizations (Color figure online)

processed by EmuLART, by looking into the emulations spatially integrated SEDs.

From both Fig. 7 and Table 2, we determine the $N_p = 10^6$ realization, with spaxel sampling of 4%, to be the most favorable input for our pipeline. This simulation is the result of $100\times$ less photons simulated than the reference, and with the corresponding emulation using as input 4% of its spaxels which results in $2500\times$ less input information than the present in the HPN reference. This emulation presents median residuals of $\sim 85\%$ between 1 and 4 μm , $\sim 33\%$ between 4 μm and 1 mm and $\sim 42\%$ over the whole spectral range, and 2.8% when considering the SED that results from the spatial integration of each wavelength map (less than 2% worse than the realization which was used as input). It also reconstructs over 71,000 spaxels presenting almost two times the amount of full spaxels of the reference. Appendix F shows how the pipeline goes even further with the emulated individual spaxels presenting smoother profiles than the references.

Appendix E.1.2: Spatial predictions

This section looks at the reconstructions of sparse samples of pixels from latent feature maps and the spatial distributions that result after returning those to the original spectral space.

The impact of INLAs spatial reconstruction is evidenced by the emulations residuals having lower median values than all realizations but the reference, and mean deviations of the same order of magnitude. In the case of the emulations performed using the $N_p = 10^7$ realizations as input that is even clearer, with median of the spatial residuals at wavelengths greater than 5 μm going below 30%. Furthermore, as Fig. 19 shows, there are two transitions in quality evidenced by the gaps between the median

residuals. It is also shown that all lower photon count simulations have their residuals dominated by lack of information (the reader is reminded that a 100% normalized residual means that pixel holds no information), and that the spatial inference at lower wavelengths results in large over-estimations. The absence of information which heavily influences the statistical descriptors of the simulations residuals is thus curbed. This is confirmed when comparing the TIR values, where we can see that the emulation based on the lowest N_p realizations has a TIR of 98.3% compared to the reference realization (see Table 2), while the $N_p = 10^7$ realization's is 65.2% (see Table 1).

Figure 20 shows the spatial distribution of the emulation residuals at some wavelengths; it is clear that:

- The residuals are correlated with the distance towards the center of the object, with the highest residuals populating the outer edge and the lowest in the center.
- Shorter wavelengths display higher residuals.

The above results were expected, given that: we performed a uniform spatial sampling of an object central to a spherically symmetric dust distribution; more spatial information is available at the center of the simulated object than in the outskirts; and, at shorter wavelengths there is overall less information, as seen in the previous section, from which to build the spatial reconstruction.

It is harder to assess how accurate INLAs inference of the outskirts regions is. Those regions, given the spatial distribution of the simulated model, tend to be less dense. Nevertheless, at most wavelengths and most of the spatial distribution of the object, the residuals place the emulated value at the same order of magnitude as the reference values. Moreover, it is in these regions that INLA recovers more information, filling in the gaps.

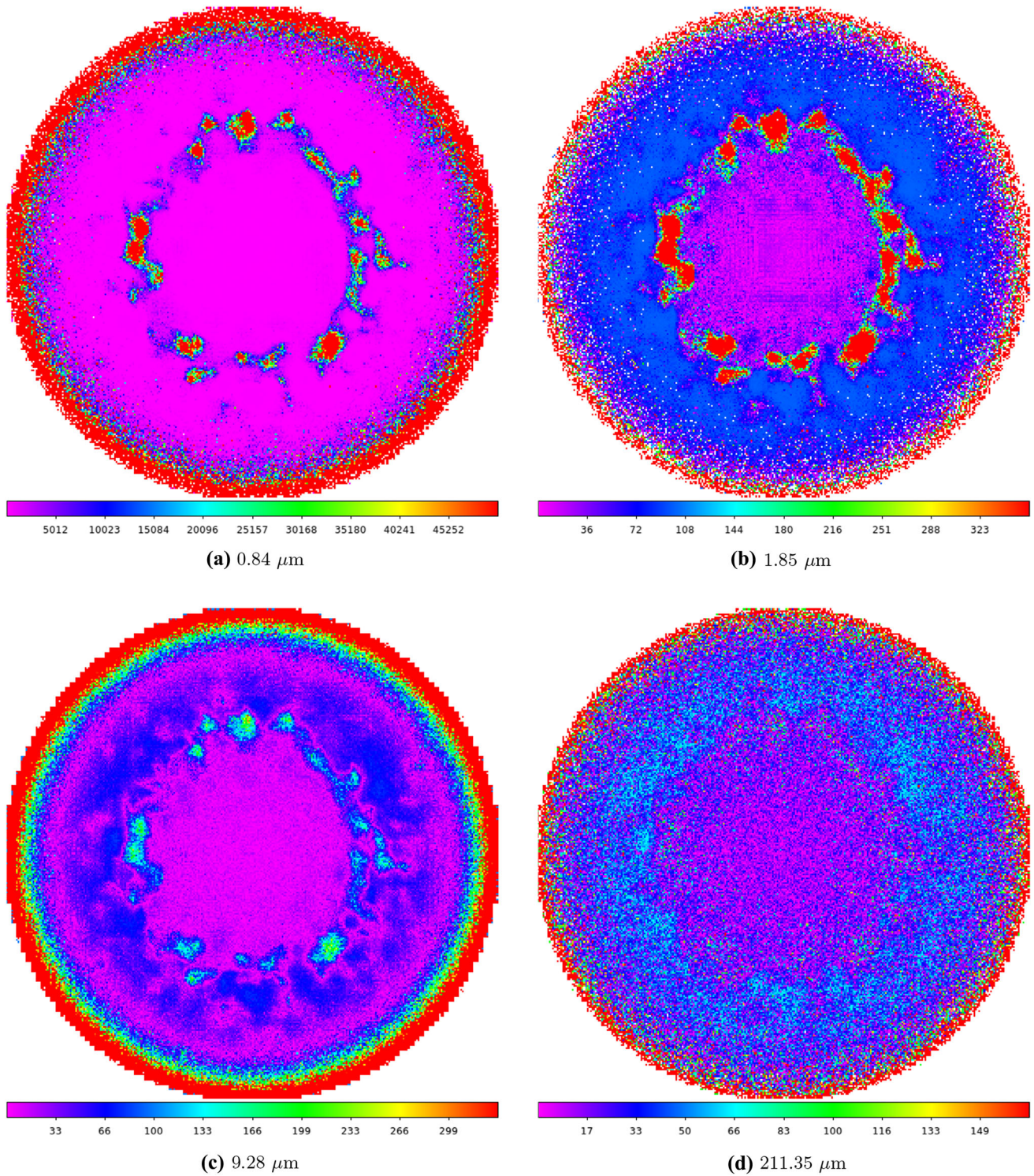


Fig. 20 Spatial distribution of residuals, as defined in Eq. (18), for the emulation produced using as input a 4% sample of the $N_p = 10^6$ realization, for the case of a dust shell with $\tau_{9.7} = 0.05$ and $\phi = 0^\circ$, at wavelengths 0.84 μm (a), 1.85 μm (b), 9.28 μm (c) and 211.35 μm (d)

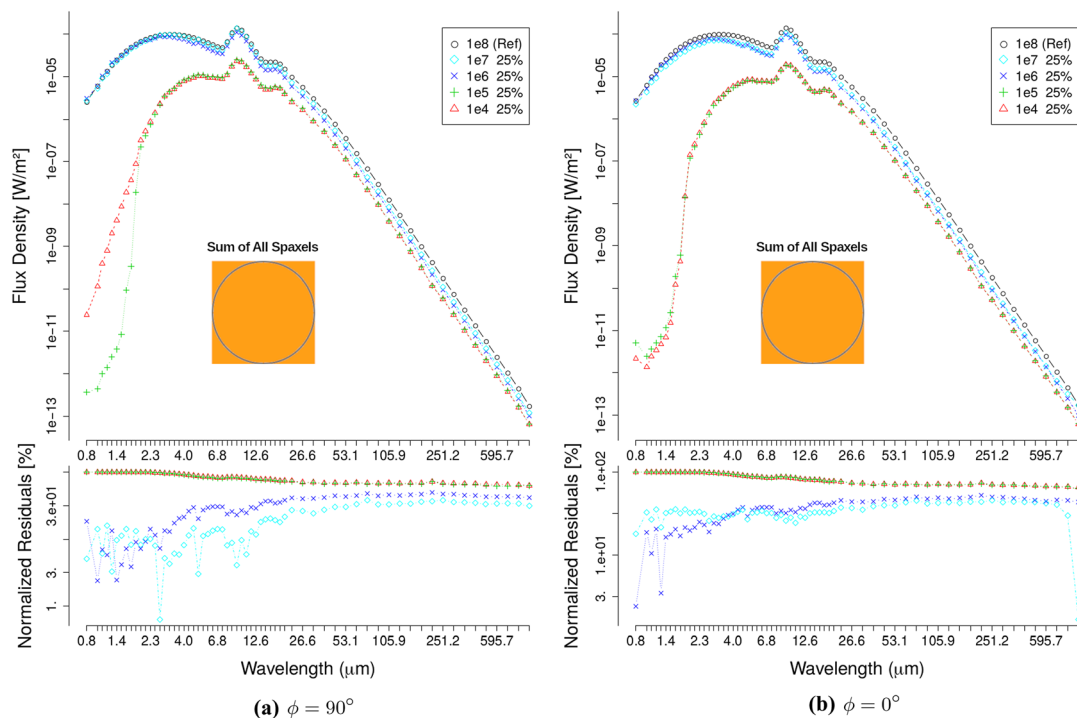


Fig. 21 Emulation-integrated SEDs, resulting from spatial inference using 25% of spatial information, and the respective normalized residuals, for the cases of dust shells with $\tau_{9.7} = 0.1$, $\phi = 90^\circ$ (a) and $\phi = 0^\circ$ (b). The HPN reference is represented in black (o), the

emulation based on the $N_p = 10^4$ realization is in red (Δ), on the $N_p = 10^5$ in green (+), on the $N_p = 10^6$ in blue (\times) and on the $N_p = 10^7$ in cyan (\diamond) (Color figure online)

Table 8 Comparison of statistics for the residuals of the spatial integration SEDs (columns 4 and 5) and of the residuals overall wavelengths (columns 6 and 7), for the emulations that took as input LPN realizations of dust shells with $\tau_{9.7} \in \{0.1, 1.0\}$, $\phi \in \{0^\circ, 90^\circ\}$, and $N_p \in \{10^6, 10^7\}$

| Case | | LPN input sample | Residuals of integrated Emul. (%) | | Normalized residuals of Emul. (%) | |
|--------------|--------------|------------------|-----------------------------------|------|-----------------------------------|-----|
| $\tau_{9.7}$ | ϕ (deg) | | M | MAD | M | MAD |
| 0.1 | 0 | 10^6 11% | 279 | 256 | 561 | 805 |
| | | 10^6 25% | 37 | 10 | 56 | 32 |
| | | 10^7 11% | 33.2 | 5.9 | 46 | 31 |
| | | 10^7 25% | 32.6 | 5.3 | 48 | 35 |
| | 90 | 10^6 11% | 33 | 12 | 58 | 35 |
| | | 10^6 25% | 30 | 13 | 54 | 34 |
| | | 10^7 11% | 15 | 10 | 36 | 29 |
| | | 10^7 25% | 14.1 | 9.9 | 35 | 28 |
| 1.0 | 0 | 10^6 11% | 89.8 | 5.2 | 92.1 | 9.8 |
| | | 10^6 25% | 89.8 | 5.3 | 92 | 10 |
| | | 10^7 11% | 91 | 2e+9 | 93.5 | 9.4 |
| | | 10^7 25% | 91 | 7e+5 | 94.1 | 7.8 |
| | 90 | 10^6 11% | 89.6 | 7.8 | 92.3 | 8.1 |
| | | 10^6 25% | 90.6 | 5.8 | 92.6 | 7.7 |
| | | 10^7 11% | 91 | 4e+8 | 93.0 | 9.1 |
| | | 10^7 25% | 91 | 3e+7 | 93.5 | 8.1 |

Table 9 Statistics regarding how the different emulations, of the case of a dust shell with $\tau_{9,7} = 0.05$ and $\phi = 90^\circ$, compared to the HPN reference

| LPN input sample | Total Emul. time (min) | Normalized residuals of Emul. (%) | | Information of emulation | | | | |
|------------------|------------------------|-----------------------------------|-----|--------------------------|------------------------|---------------------------|------------------------|---------------------------|
| | | <i>M</i> | MAD | TIR (%) | Null Spxs (<i>N</i>) | Empty Spxs (ΔN) | Full Spxs (<i>N</i>) | Partial Spxs (<i>N</i>) |
| 10^4 4% | 7.2 | 74 | 36 | 98.1 | 20,497 | 2641 | 69,503 | 0 |
| 10^4 11% | 22.2 | 74 | 36 | | | | | |
| 10^4 25% | 16.3 | 75 | 36 | | | | | |
| 10^5 4% | 6.2 | 73 | 36 | 101.6 | 18,002 | 146 | 71,998 | 0 |
| 10^5 11% | 21.7 | 70 | 40 | | | | | |
| 10^5 25% | 13.4 | 67 | 44 | | | | | |
| 10^6 4% | 7.2 | 47 | 43 | 103.2 | 16,872 | − 984 | 73,128 | 0 |
| 10^6 11% | 22.7 | 48 | 47 | | | | | |
| 10^6 25% | 15.9 | 42 | 40 | | | | | |
| 10^7 4% | 8.0 | 36 | 38 | 103.2 | 16,833 | − 1023 | 73,167 | 0 |
| 10^7 11% | 19.1 | 36 | 39 | | | | | |
| 10^7 25% | 15.2 | 32 | 34 | | | | | |

The results are similar to those displayed in Table 3

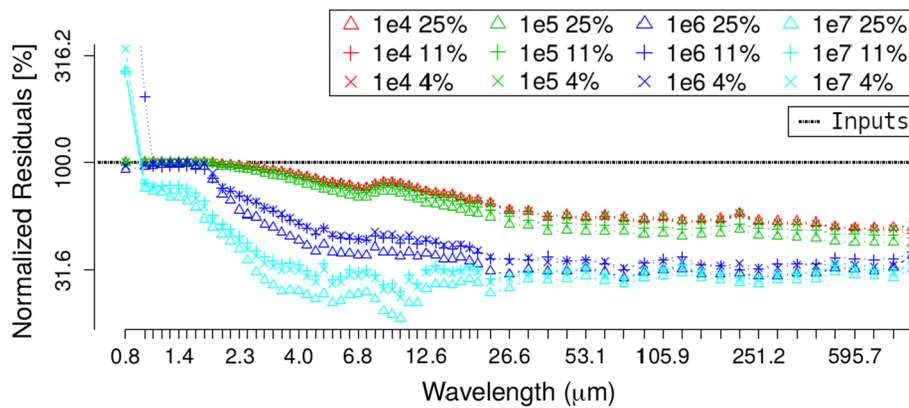


Fig. 22 Median of the normalized residuals, at each wavelengths spatial map, for every emulation obtained with $N_p = 10^4$ (red), $N_p = 10^5$ (green), $N_p = 10^6$ (blue) and $N_p = 10^7$ (cyan) realizations, for the case of a dust shell with $\tau_{9,7} = 0.05$ and $\phi = 90^\circ$. Emulations whose

spatial inference was performed taking as input 25% of the data are represented by (Δ), 11% by (+) and 4% by (\times). The interrupted black line marks the same metric for the realizations used as input (Color figure online)

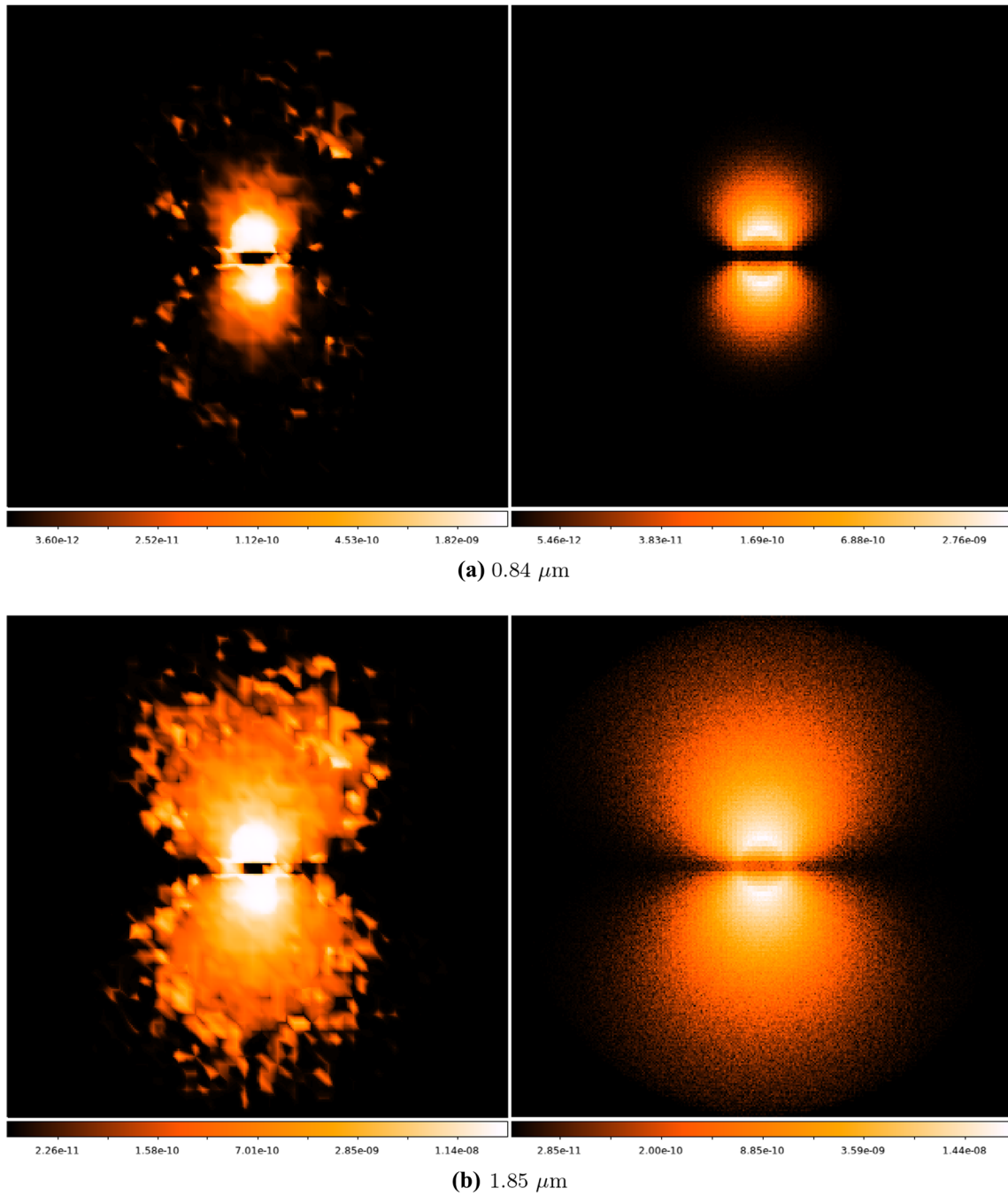


Fig. 23 Emulation produced using as input a 4% sample of the $N_p = 10^6$ realization, left, and HPN reference, right, for the case of a dust shell with $\tau_{9.7} = 0.05$ and $\phi = 90^\circ$, at wavelengths 0.84 μm (a) and 1.85 μm (b)

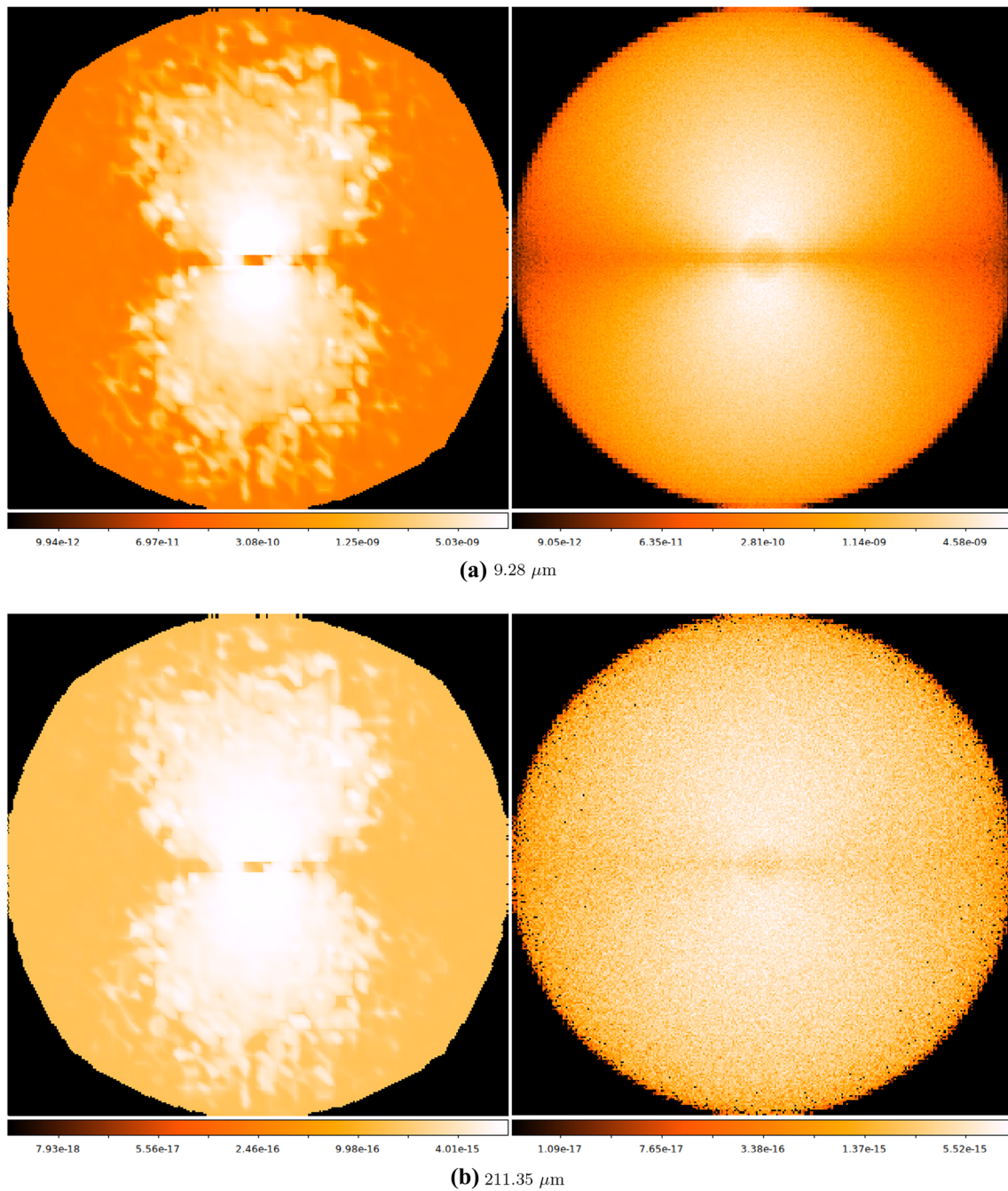


Fig. 24 Emulation produced using as input a 4% sample of the $N_p = 10^6$ realization, left, and HPN reference, right, for the case of a dust shell with $\tau_{9.7} = 0.05$ and $\phi = 90^\circ$, at wavelengths 9.28 μm (a) and 211.35 μm (b)

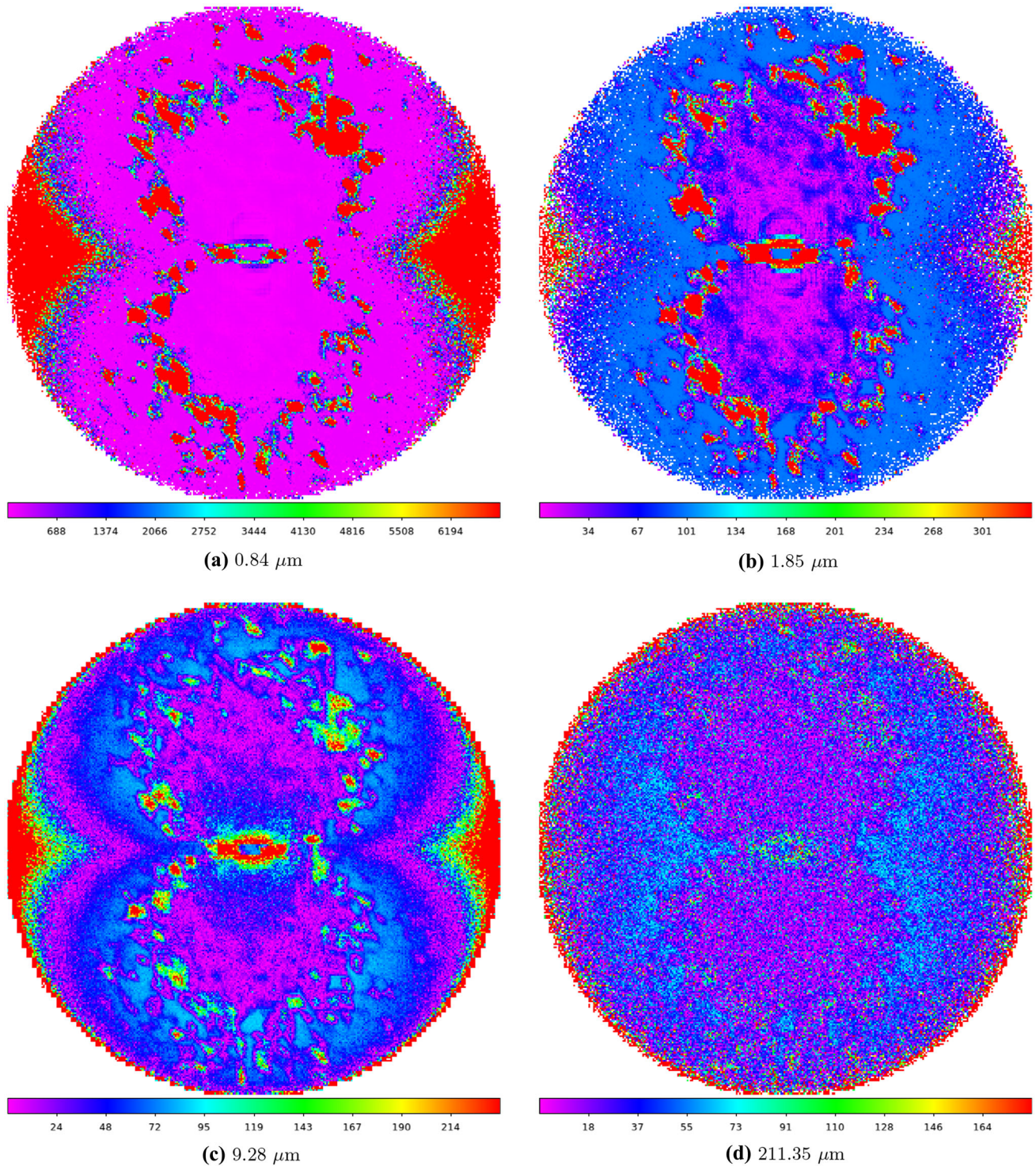
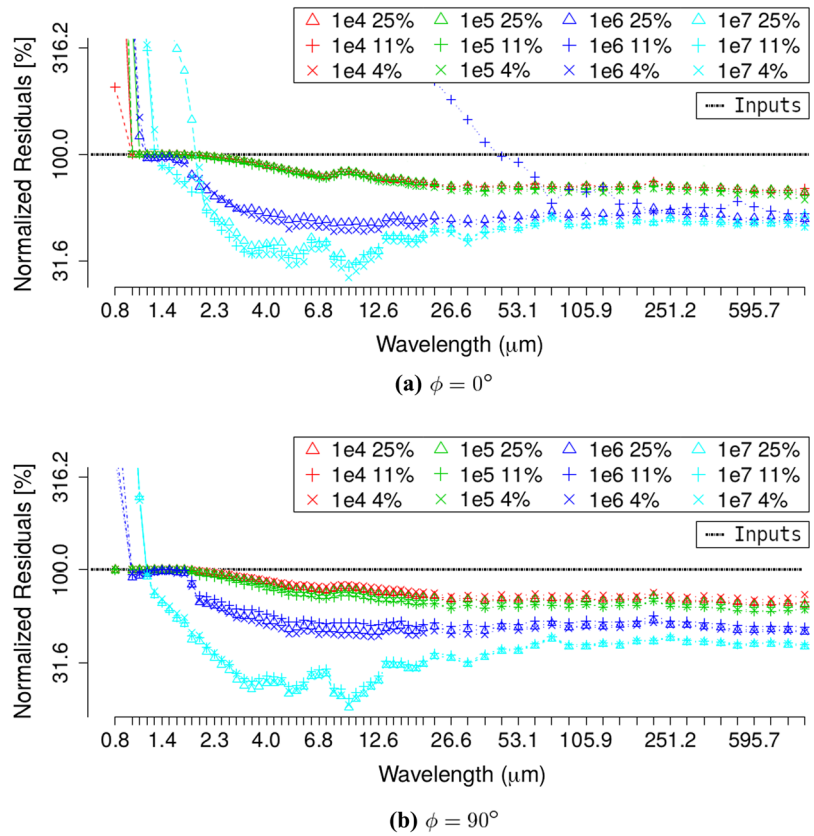


Fig. 25 Spatial distribution of residuals, as defined in Eq. (18), for the emulation produced using as input a 4% sample of the $N_p = 10^6$ realization, for the case of a dust shell with $\tau_{9.7} = 0.05$ and $\phi = 90^\circ$, at wavelengths 0.84 μm (a), 1.85 μm (b), 9.28 μm (c) and 211.35 μm (d)

Fig. 26 Median of the normalized residuals, at each wavelengths spatial map, for every emulation obtained with $N_p = 10^4$ (red), $N_p = 10^5$ (green), $N_p = 10^6$ (blue) and $N_p = 10^7$ (cyan) realizations, for the case of a dust shell with $\tau_{9.7} = 0.1$, with $\phi = 0^\circ$ (a) and $\phi = 90^\circ$ (b). Emulations whose spatial inference was performed using 25% of data are represented by (Δ), 11% by (+) and 4% by (\times). The interrupted black line marks the same metric for the LPN inputs (Color figure online)



Appendix E.2: EVASet predictions

Appendix E.2.1: Spectral predictions

In this section, we discuss how well the spectral information is preserved, after the LPN inputs of EVASet, with $\tau_{9.7} \neq 0.05$, are processed by EmulART, by once again looking into the emulations spatially integrated SEDs.

Regarding the case with $\tau_{9.7} = 0.1$, Fig. 21 and Table 8 show that, for both tilt angles, while the shape of the emulations integrated spectra still follows closely that of the references, the residuals are now over $2\times$ higher than with $\tau_{9.7} = 0.05$. The results for the cases with $\tau_{9.7} = 1.0$ confirm this trend in degrading quality of the pipeline’s spectral predictions. As seen in Fig. 11, the emulation-integrated spectrum completely misses the shape of the reference spectrum, keeping the shape displayed for the cases with $\tau_{9.7} = 0.05$ and $\tau_{9.7} = 0.1$, a sign of over-fitting which is confirmed when inspecting individual spaxels (see Appendix F). It also underestimates the flux density and overestimates the relative magnitude of the two emission bumps in the 8–20 μm range. The median residual of the total flux density per wavelength, displayed in Table 8, further confirms the poor quality of the spectral predictions.

Appendix E.2.2: Spatial predictions

Regarding spatial information, for emulations with LPN inputs of $\tau_{9.7} = 0.05$ and $\phi = 90^\circ$ we observe the same behavior in the performance metrics as with the $\phi = 0^\circ$ case. Table 9 shows that the overall median residuals decrease with the increase of N_p . The median residuals per wavelength, displayed in Fig. 22, present a similar behavior to those of the $\phi = 0^\circ$ case.

Figure 23 shows that the emulation of models with excessive emission within the central regions in comparison with the reference, while in Fig. 24 the prediction for the peripheral region is more isotropic than the reference; this is even clearer when consulting the spatial distribution of residuals for these emulations in Fig. 25 (Fig. 26).

As with the $\tau_{9.7} = 1.0$ cases, for both tilt angle cases of $\tau_{9.7} = 0.1$ we observe a similar trend in the overall residuals. Figure 27 shows that while the overall morphology of the spatial distribution is well recovered, the flux density value range for the emulations is underestimated, though not as considerably as with the $\tau_{9.7} = 1.0$ cases, and the contrast between the central and peripheral regions is higher than in the HPN references. This indicates a bias within the pipeline. Appendix F further details into the

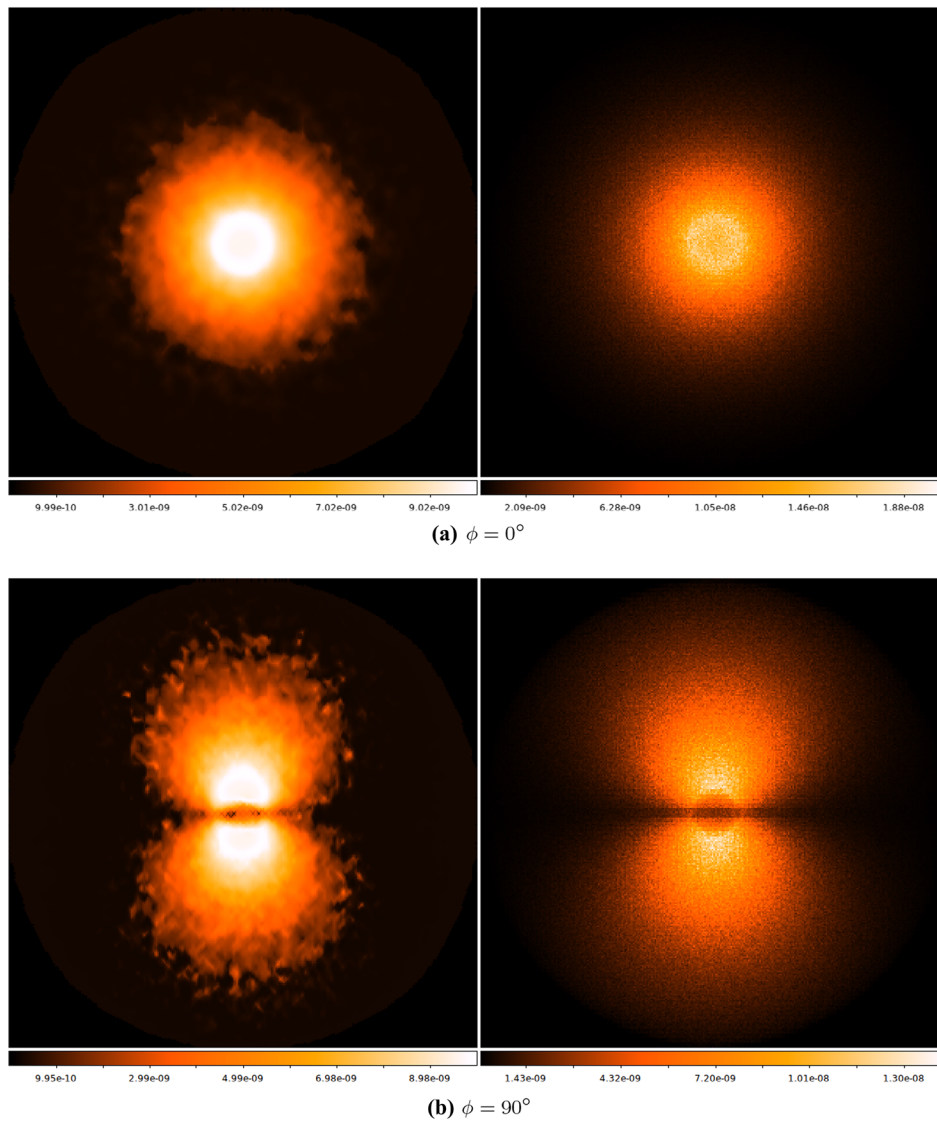


Fig. 27 Emulation produced using as input a 25% sample of the $N_p = 10^6$ realization, left, and HPN reference, right, for the case of dust shells with $\tau_{9.7} = 0.1$, at wavelengths $9.28 \mu\text{m}$, with $\phi = 0^\circ$, (a) and $\phi = 90^\circ$ (b)

likely source of such bias by comparing individual spaxels of emulations and HPN simulations.

Appendix F: Single spaxels

Figures 28, 29 and 30 compare 3 different spaxels between their versions in the LPN input, HPN reference and emulation based on the LPN input. These spaxels are radially aligned: one at the center of the object, another on the outskirts and another in-between those two. The emulations were achieved using 25% of the corresponding source cube spaxels.

Some things easily noticeable are the underestimation of the shorter wavelengths for the versions of these spaxels taken from emulations based on $N_p = 10^4$ and $N_p = 10^5$ realizations. Nevertheless, in those cases the LPN input spaxel is not complete, i.e., there are zero values at some wavelengths, if not all, meaning the pipeline is reaching what it was designed to do: take a sparse compressed model, infer the missing information and then decompress it.

Figures 31, 32 and 33 compare 3 different spaxels from the same spatial locations as before. The emulations were achieved using 11% of the corresponding LPN input cube spaxels, which are the $\tau_{9.7} = 1.0$ and $\phi \in \{0^\circ, 90^\circ\}$ cases. Since the simulated bright central source has anisotropic

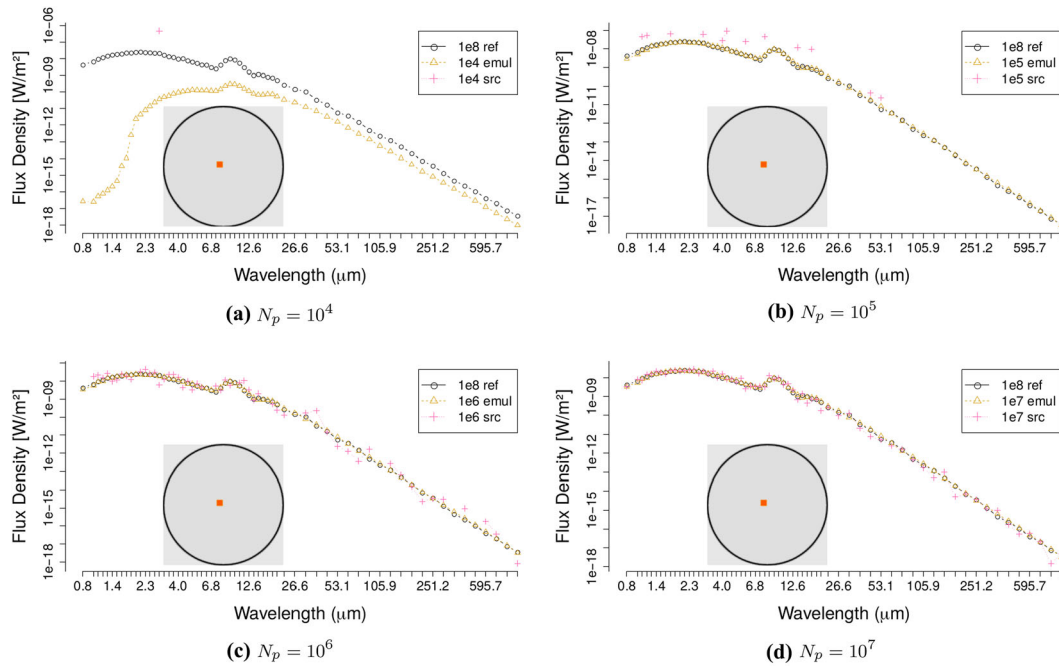


Fig. 28 Comparison of some emulations spaxels located in row = 150, col = 150, against the HPN references and the LPN input (indicated in each plot), for the case of a dust shell with $\tau_{9,7} = 0.05$ and $\phi = 0^\circ$. The emulations were performed using 25% of the LPN input spatial

information. HPN references spaxel in black (\circ), emulations in gold (Δ) and LPN inputs, upon which the emulation was performed, in pink ($+$) (Color figure online)

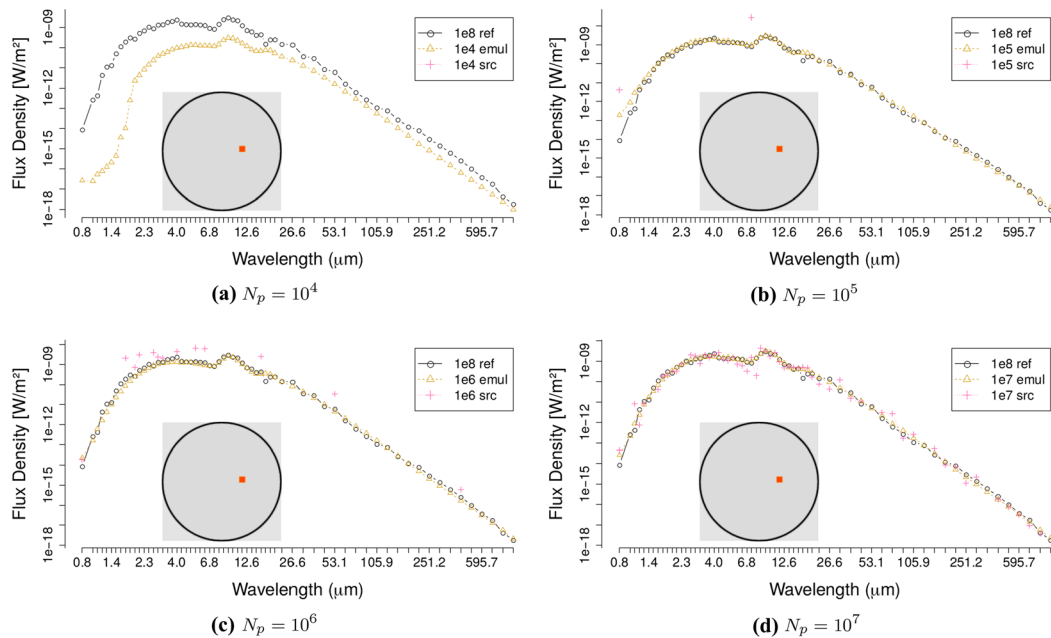


Fig. 29 Comparison of some emulations spaxels located in row = 150, col = 210, against the HPN references and the LPN input (indicated in each plot), for the case of a dust shell with $\tau_{9,7} = 0.05$ and $\phi = 0^\circ$. The emulations were performed using 25% of the LPN input spatial

information. HPN references spaxel in black (\circ), emulations in gold (Δ) and LPN inputs, upon which the emulation was performed, in pink ($+$) (Color figure online)

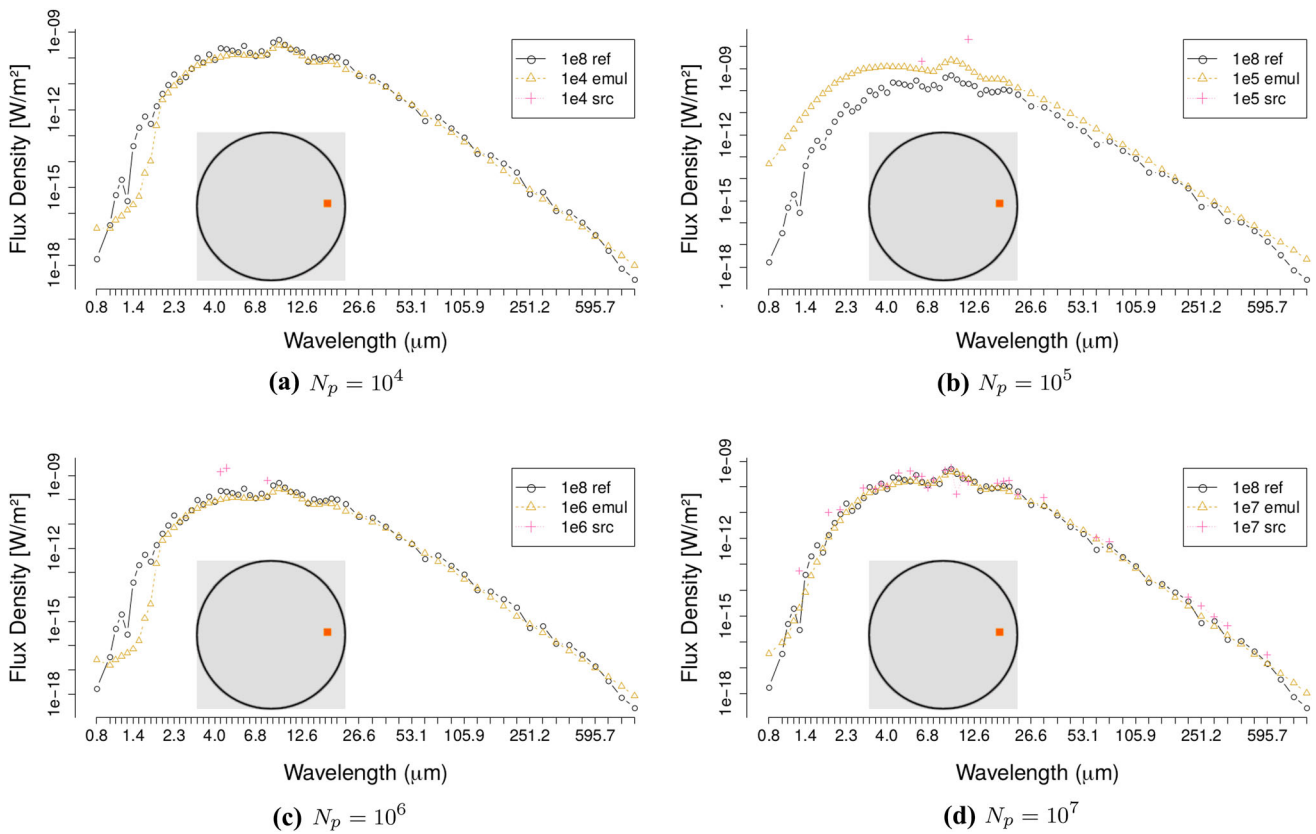


Fig. 30 Comparison of some emulation spaxels located in row = 150, col = 270, against the HPN references and the LPN input (indicated in each plot), for the case of a dust shell with $\tau_{9.7} = 0.05$ and $\phi = 0^\circ$. The emulations were performed using 25% of the LPN input spatial

information. HPN references spaxel in black (\circ), emulations in gold (Δ) and LPN inputs, upon which the emulation was performed, in pink ($+$) (Color figure online)

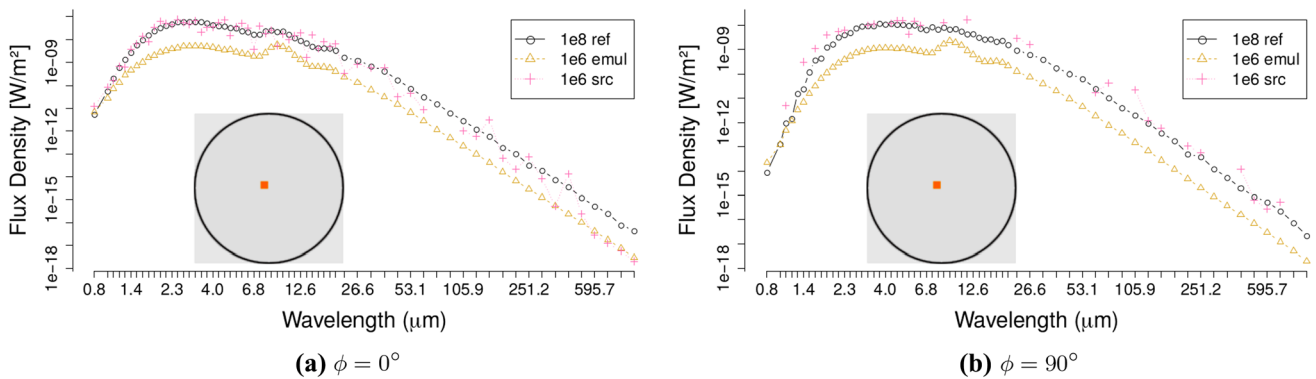


Fig. 31 Comparison of emulations spaxels, located in row = 150, col = 150, against the HPN references and the $N_p = 10^6$ realization, for the case of a dust shell with $\tau_{9.7} = 1.0$, with $\phi = 0^\circ$ (a) and $\phi = 90^\circ$

(b). HPN references spaxel in black (\circ), emulations in gold (Δ) and LPN inputs, upon which the emulation was performed, in pink ($+$) (Color figure online)

emission at greater optical depths, $\tau_{9.7}$, we can expect that for different viewing angles, ϕ , the shapes of spaxels near the central region of the spatial distribution³² will present

differences as can be seen in the reference curves of Fig. 31.

For both the face-on and edge-on cases, at all displayed positions, we can make the same assessment that the shape

³² This is relevant because light coming from near the center of the spatial distribution is less likely to have been scattered towards the observer, unlike light coming from the edges which is mostly light

Footnote 32 continued that also came from the center, where the bright source is located, and was scattered towards the observer.

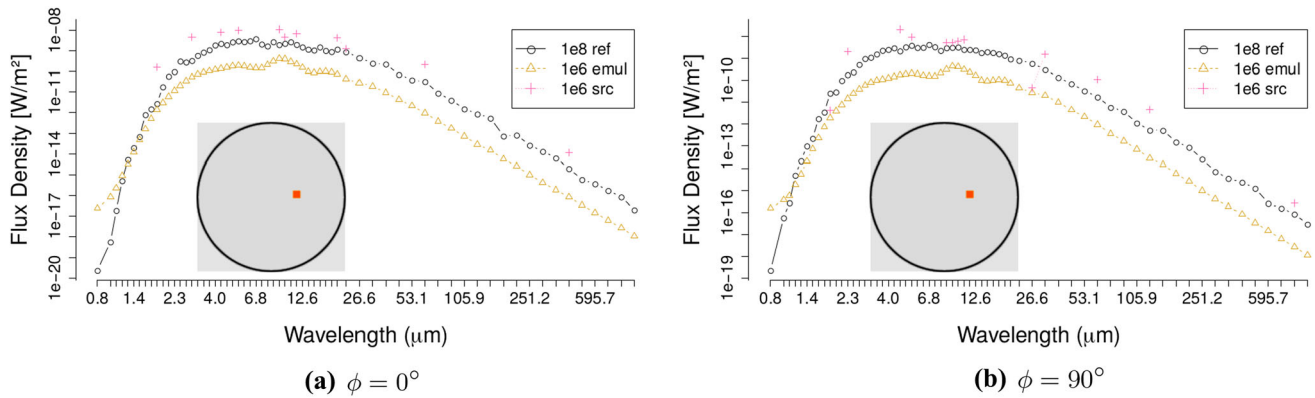


Fig. 32 Comparison of emulations spaxels, located in row = 150, col = 210, against the HPN references and the $N_p = 10^6$ realization, for the cases of dust shells with $\tau_{9.7} = 1.0$, with $\phi = 0^\circ$ (Fig. 31a) and

$\phi = 90^\circ$ (Fig. 31b). HPN references spaxel in black (\circ), emulations in gold (Δ) and LPN inputs, upon which the emulation was performed, in pink (+) (Color figure online)

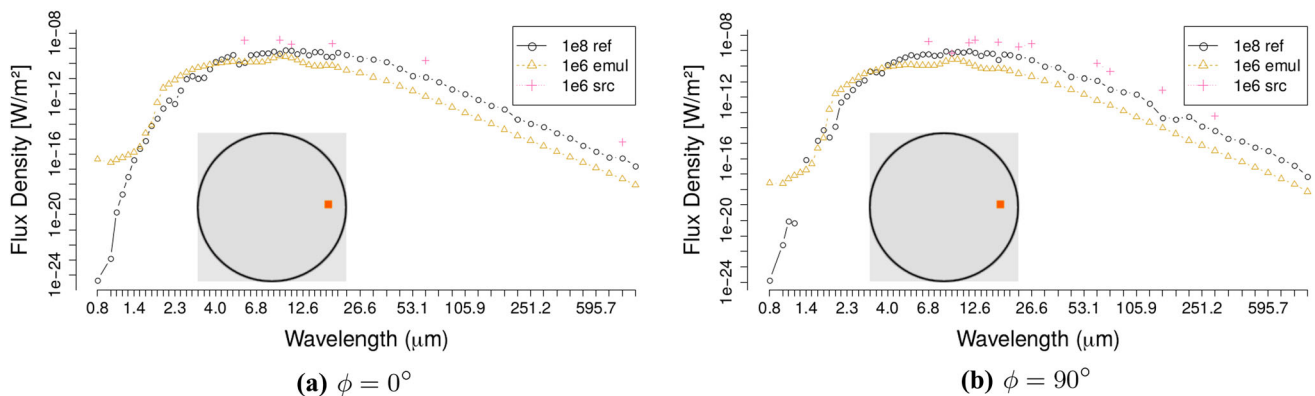


Fig. 33 Comparison of emulations spaxels, located in row = 150, col = 270, against the HPN references and the $N_p = 10^6$ realization, for the cases of dust shells with $\tau_{9.7} = 1.0$, with $\phi = 0^\circ$ (Fig. 31a) and

$\phi = 90^\circ$ (Fig. 31b). HPN references spaxel in black (\circ), emulations in gold (Δ) and LPN inputs, upon which the emulation was performed, in pink (+) (Color figure online)

of the emulated spaxels does not follow the shape of the HPN references. As the simulated models dust density increases so does the likelihood of scattering and absorption of light by the medium, consequently the peak emission features are flattened into the continuum. Figures 31 and 32 show that the emulation of the spaxels in those regions is one to two orders of magnitude too low while the relative flux between the emission peak, at $9.7 \mu\text{m}$, and the continuum is over-estimated, the allocation of flux along the spaxels does not match that of the reference. For the outer spaxels, Fig. 33, the flux mismatch is greater for longer wavelengths, but still the emulation spaxels present a peak emission feature while the references no longer show it. This is a clear sign that the DVAE model is biased.

Acknowledgements Computations were performed at the cluster “Baltasar-Sete-Sóis” and supported by the H2020 ERC Consolidator Grant “Matter and strong field gravity: New frontiers in Einstein’s theory” grant agreement no. MaGRaTh-646597. J. R.-S. is funded by Fundação para a Ciência e a Tecnologia (FCT) (PD/BD/150487/2019), via the International Doctorate Network in Particle Physics, Astrophysics and Cosmology (IDPASC). The work here described is

included in the CRISP project, which is also funded by the FCT (PTDC/FISAST/31546/2017). M. S. and M. S. acknowledge support by the Ministry of Education, Science and Technological Development of the Republic of Serbia through the contract no. 451-03-9/2021-14/200002 and by the Science Fund of the Republic of Serbia, PROMIS 6060916, BOWIE. J. P. C acknowledges the support of the FCT, through Portuguese national funds (UIDB/50021/2020).

Funding Open access funding provided by FCTIFCCN (b-on). The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Declarations

Conflict of interest The authors declare no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless

indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Allaire J, Chollet F (2021) keras: R Interface to 'Keras'. <https://CRAN.R-project.org/package=keras>, r package version 2.6.1
- André P, Men'shchikov A, Bontemps S et al (2010) From filamentary clouds to prestellar cores to the stellar IMF: initial highlights from the Herschel Gould Belt Survey. *Astron Astrophys* 518:L102. <https://doi.org/10.1051/0004-6361/201014666>. arXiv:1005.2618 [astro-ph.GA]
- Babusiaux C, Fourtune-Ravard C, Hottier C et al (2020) FEDReD. I. 3D extinction and stellar maps by Bayesian deconvolution. *Astron Astrophys* 641:A78. <https://doi.org/10.1051/0004-6361/202037466>. arXiv:2007.04455 [astro-ph.GA]
- Bachl FE, Lindgren F, Borchers DL et al (2019) inlabru: an R package for Bayesian spatial modelling from ecological survey data. *Methods Ecol Evol* 10(6):760–766. <https://doi.org/10.1111/2041-210X.13168>
- Baes M, Verstappen J, De Looze I et al (2011) Efficient three-dimensional NLTE dust radiative transfer with SKIRT. *Astrophys J* 196(2):22. <https://doi.org/10.1088/0067-0049/196/2/22>. arXiv:1108.5056 [astro-ph.CO]
- Beech M (1987) Are lynds dark clouds fractals? *Astrophys Space Sci* 133(1):193–195. <https://doi.org/10.1007/BF00637432>
- Betoule M, Kessler R, Guy J et al (2014) Improved cosmological constraints from a joint analysis of the SDSS-II and SNLS supernova samples. *Astron Astrophys* 568:A22. <https://doi.org/10.1051/0004-6361/201423413>. arXiv:1401.4064 [astro-ph.CO]
- Boulais A, Berné O, Faury G et al (2021) Unmixing methods based on nonnegativity and weakly mixed pixels for astronomical hyperspectral datasets. *Astron Astrophys* 647:A105. <https://doi.org/10.1051/0004-6361/201936399>. arXiv:2011.09742 [astro-ph.IM]
- Camps P, Baes M (2015) SKIRT: an advanced dust radiative transfer code with a user-friendly architecture. *Astron Comput* 9:20–33. <https://doi.org/10.1016/j.ascom.2014.10.004>. arXiv:1410.1629 [astro-ph.IM]
- Camps P, Baes M (2020) SKIRT 9: redesigning an advanced dust radiative transfer code to allow kinematics, line transfer and polarization by aligned dust grains. *Astron Comput* 31:100381. <https://doi.org/10.1016/j.ascom.2020.100381>. arXiv:2003.00721 [astro-ph.GA]
- Collins JD, Hart GC, Haselman TK et al (1974) Statistical identification of structures. *AIAA J* 12(2):185–190. <https://doi.org/10.2514/3.49190>
- Cox NLJ, Kerschbaum F, van Marle AJ et al (2012) A far-infrared survey of bow shocks and detached shells around AGB stars and red supergiants. *Astron Astrophys* 537:A35. <https://doi.org/10.1051/0004-6361/201117910>. arXiv:1110.5486 [astro-ph.GA]
- De Looze I, Fritz J, Baes M et al (2014) High-resolution, 3D radiative transfer modeling. I. The grand-design spiral galaxy M 51. *Astron Astrophys* 571:A69. <https://doi.org/10.1051/0004-6361/201424747>. arXiv:1409.3857 [astro-ph.GA]
- Doersch C (2016) Tutorial on variational autoencoders. arXiv e-prints arXiv:1606.05908 [stat.ML]
- Dunne L, Gomez HL, da Cunha E et al (2011) Herschel-ATLAS: rapid evolution of dust in galaxies over the last 5 billion years. *Mon Not R Astron Soc* 417(2):1510–1533. <https://doi.org/10.1111/j.1365-2966.2011.19363.x>. arXiv:1012.5186 [astro-ph.CO]
- Falgarone E, Phillips TG, Walker CK (1991) The edges of molecular clouds: fractal boundaries and density structure. *Astrophys J* 378:186. <https://doi.org/10.1086/170419>
- Fritz J, Gentile G, Smith MWL et al (2012) The Herschel Exploitation of Local Galaxy Andromeda (HELGA). I. Global far-infrared and sub-mm morphology. *Astron Astrophys* 546:A34. <https://doi.org/10.1051/0004-6361/201118619>. arXiv:1112.3348 [astro-ph.CO]
- Gómez-Rubio V (2021) Bayesian inference with INLA. Chapman & Hall/CRC Press, Boca Raton
- González-Gaitán S, de Souza RS, Krone-Martins A et al (2019) Spatial field reconstruction with INLA: application to IFU galaxy data. *Mon Not R Astron Soc* 482(3):3880–3891. <https://doi.org/10.1093/mnras/sty2881>. arXiv:1802.06280 [astro-ph.IM]
- Guo H, Zhuang X, Rabczuk T (2019) A deep collocation method for the bending analysis of Kirchhoff plate. *Comput Mater Continua* 59(2):433–456. <https://doi.org/10.32604/cmc.2019.06660>
- Haas M, Müller SAH, Chini R et al (2000) Dust in PG quasars as seen by ISO. *Astron Astrophys* 354:453–466
- Hinton G, Salakhutdinov R (2006) Reducing the dimensionality of data with neural networks. *Science* 313:504–507. <https://doi.org/10.1126/science.1127647>
- Hinton GE, van Camp D (1993) Keeping the neural networks simple by minimizing the description length of the weights. In: Proceedings of the sixth annual conference on computational learning theory, COLT'93. Association for Computing Machinery, New York, pp 5–13. <https://doi.org/10.1145/168304.168306>
- Ichinohe Y, Yamada S (2019) Neural network-based anomaly detection for high-resolution X-ray spectroscopy. *Mon Not R Astron Soc* 487(2):2874–2880. <https://doi.org/10.1093/mnras/stz1528>. arXiv:1905.13434 [astro-ph.IM]
- Im DJ, Ahn S, Memisevic R et al (2015) Denoising criterion for variational auto-encoding framework. arXiv e-prints arXiv:1511.06406 [cs.LG]
- Ishida EEO, de Souza RS (2011) Hubble parameter reconstruction from a principal component analysis: minimizing the bias. *Astron Astrophys* 527:A49. <https://doi.org/10.1051/0004-6361/201015281>. arXiv:1012.5335 [astro-ph.CO]
- Ishida EEO, de Souza RS, Ferrara A (2011) Probing cosmic star formation up to $z = 9.4$ with gamma-ray bursts. *Mon Not R Astron Soc* 418(1):500–504. <https://doi.org/10.1111/j.1365-2966.2011.19501.x>. arXiv:1106.1745 [astro-ph.CO]
- Jia P, Li X, Li Z et al (2020) Point spread function modelling for wide-field small-aperture telescopes with a denoising autoencoder. *Mon Not R Astron Soc* 493(1):651–660. <https://doi.org/10.1093/mnras/staa319>. arXiv:2001.11716 [astro-ph.IM]
- Jolliffe IT, Cadima J (2016) Principal component analysis: a review and recent developments. *Philos Trans R Soc Lond Ser A* 374(2065):20150202. <https://doi.org/10.1098/rsta.2015.0202>
- Klambauer G, Unterthiner T, Mayr A et al (2017) Self-normalizing neural networks. arXiv e-prints arXiv:1706.02515 [cs.LG]
- Kopf A, Fortuin V, Somnath VR et al (2021) Mixture-of-experts variational autoencoder for clustering and generating from similarity-based representations on single cell data. *PLoS Comput Biol*. <https://doi.org/10.1371/journal.pcbi.1009086>
- Krone-Martins A, Moitinho A (2014) UPMASK: unsupervised photometric membership assignment in stellar clusters. *Astron Astrophys* 561:A57. <https://doi.org/10.1051/0004-6361/201321143>. arXiv:1309.4471 [astro-ph.IM]
- Logan CHA, Fotopoulou S (2020) Unsupervised star, galaxy, QSO classification. Application of HDBSCAN. *Astron Astrophys* 633:A154. <https://doi.org/10.1051/0004-6361/201936648>. arXiv:1911.05107 [astro-ph.GA]

34. Mattila K (1970) Interpretation of the surface brightness of dark nebulae. *Astron Astrophys* 9:53
35. Minka TP (2013) Expectation propagation for approximate Bayesian inference. arXiv e-prints [arXiv:1301.2294](https://arxiv.org/abs/1301.2294) [cs.AI]
36. Netzer H (1987) Quasar discs. II—a composite model for the broad-line region. *Mon Not R Astron Soc* 225:55–72. <https://doi.org/10.1093/mnras/225.1.55>
37. O’Brian T, Ting YS, Fabbro S et al (2020) Interpreting stellar spectra with unsupervised domain adaptation. arXiv e-prints [arXiv:2007.03112](https://arxiv.org/abs/2007.03112) [astro-ph.SR]
38. Portillo SKN, Parejko JK, Vergara JR et al (2020) Dimensionality reduction of SDSS spectra with variational autoencoders. *Astron J* 160(1):45. <https://doi.org/10.3847/1538-3881/ab9644>. [arXiv:2002.10464](https://arxiv.org/abs/2002.10464) [astro-ph.IM]
39. R Core Team (2021) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna. <https://www.R-project.org/>
40. Ralph NO, Norris RP, Fang G et al (2019) Radio galaxy zoo: unsupervised clustering of convolutionally auto-encoded radio-astronomical images. *Publ Astron Soc Pac* 131(1004):108011. <https://doi.org/10.1088/1538-3873/ab213d>. [arXiv:1906.02864](https://arxiv.org/abs/1906.02864) [astro-ph.IM]
41. Ren B, Pueyo L, Zhu GB et al (2018) Non-negative matrix factorization: robust extraction of extended structures. *Astrophys J* 852(2):104. <https://doi.org/10.3847/1538-4357/aaa1f2>. [arXiv:1712.10317](https://arxiv.org/abs/1712.10317) [astro-ph.IM]
42. Rho J, Reach WT, Tappe A et al (2009) Spitzer observations of the young core-collapse supernova remnant IE0102-72.3: infrared ejecta emission and dust formation. *Astrophys J* 700(1):579–596. <https://doi.org/10.1088/0004-637X/700/1/579>
43. Roark T, Roark B, Collins IGW (1974) Monte Carlo model of reflection nebulae: intensity gradients. *Astrophys J* 190:67–72. <https://doi.org/10.1086/152847>
44. Rue H, Martino S, Chopin N (2009) Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *J R Stat Soc Ser B (Stat Methodol)* 71(2):319–392. <https://doi.org/10.1111/j.1467-9868.2008.00700.x>
45. Sasdelli M, Hillebrandt W, Aldering G et al (2014) A metric space for Type Ia supernova spectra. *Mon Not R Astron Soc* 447(2):1247–1266. <https://doi.org/10.1093/mnras/stu2416>
46. Smole M, Rino-Silvestre J, González-Gaitán S et al (2022) Spatial field reconstruction with INLA: application to simulated galaxies. arXiv e-prints [arXiv:2211.02602](https://arxiv.org/abs/2211.02602) [astro-ph.IM]
47. Stalevski M, Ricci C, Ueda Y et al (2016) The dust covering factor in active galactic nuclei. *Mon Not R Astron Soc* 458(3):2288–2302. <https://doi.org/10.1093/mnras/stw444>. [arXiv:1602.06954](https://arxiv.org/abs/1602.06954) [astro-ph.GA]
48. Stalevski M, Tristram KRW, Asmus D (2019) Dissecting the active galactic nucleus in Circinus—II. A thin dusty disc and a polar outflow on parsec scales. *Mon Not R Astron Soc* 484(3):3334–3355. <https://doi.org/10.1093/mnras/stz220>. [arXiv:1901.05488](https://arxiv.org/abs/1901.05488) [astro-ph.GA]
49. Steinacker J, Baes M, Gordon KD (2013) Three-dimensional dust radiative transfer*. *Ann Rev Astron Astrophys* 51(1):63–104. <https://doi.org/10.1146/annurev-astro-082812-141042>. [arXiv:1303.4998](https://arxiv.org/abs/1303.4998) [astro-ph.IM]
50. Tierney L, Kadane JB (1986) Accurate approximations for posterior moments and marginal densities. *J Am Stat Assoc* 81(393):82–86. <https://doi.org/10.1080/01621459.1986.10478240>
51. Verstocken S, Nersesian A, Baes M et al (2020) High-resolution, 3D radiative transfer modelling. II. The early-type spiral galaxy M 81. *Astron Astrophys* 637:A24. <https://doi.org/10.1051/0004-6361/201935770>. [arXiv:2004.03615](https://arxiv.org/abs/2004.03615) [astro-ph.GA]
52. Viaene S, Baes M, Tamm A et al (2017) The Herschel Exploitation of Local Galaxy Andromeda (HELGA). VII. A SKIRT radiative transfer model and insights on dust heating. *Astron Astrophys* 599:A64. <https://doi.org/10.1051/0004-6361/201629251>. [arXiv:1609.08643](https://arxiv.org/abs/1609.08643) [astro-ph.GA]
53. Wang YC, Xie YB, Zhang TJ et al (2021) Likelihood-free cosmological constraints with artificial neural networks: an application on Hubble parameters and SNe Ia. *Astrophys J* 254(2):43. <https://doi.org/10.3847/1538-4365/abf8aa>. [arXiv:2005.10628](https://arxiv.org/abs/2005.10628) [astro-ph.CO]
54. Wells DC, Greisen EW, Harten RH (1981) FITS—a flexible image transport system. *Astron Astrophys Suppl* 44:363
55. Zhuang X, Guo H, Alajlan N et al (2021) Deep autoencoder based energy method for the bending, vibration, and buckling analysis of Kirchhoff plates with transfer learning. *Eur J Mech A Solids* 87(104):225. <https://doi.org/10.1016/j.euromechsol.2021.104225>
56. Zhukovska S, Gail HP, Tieloff M (2008) Evolution of interstellar dust and stardust in the solar neighbourhood. *Astron Astrophys* 479(2):453–480. <https://doi.org/10.1051/0004-6361/20077789>. [arXiv:0706.1155](https://arxiv.org/abs/0706.1155) [astro-ph]
57. Zorich L, Pichara K, Protopapas P (2020) Streaming classification of variable stars. *Mon Not R Astron Soc* 492(2):2897–2909. <https://doi.org/10.1093/mnras/stz3426>. [arXiv:1912.02235](https://arxiv.org/abs/1912.02235) [astro-ph.IM]

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

João Rino-Silvestre¹  · Santiago González-Gaitán¹ · Marko Stalevski^{2,3} · Majda Smole² · Pedro Guilherme-Garcia⁴ · Joao Paulo Carvalho⁵ · Ana Maria Mourão¹

✉ João Rino-Silvestre
joao.silvestre@tecnico.ulisboa.pt

Santiago González-Gaitán
gongsale@gmail.com

Marko Stalevski
mstalevski@aob.rs

Majda Smole
msmole@aob.rs

Pedro Guilherme-Garcia
pguilhermegarcia@gmail.com

Joao Paulo Carvalho
joao.carvalho@inesc-id.pt

Ana Maria Mourão
amourao@tecnico.ulisboa.pt

¹ CENTRA, Instituto Superior Técnico, Av. Rovisco Pais, 1049-001 Lisbon, Portugal

² Astronomical Observatory, Volgina 7, 1060 Belgrade, Serbia

³ Sterrenkundig Observatorium, Universiteit Gent, Krijgslaan 281-S9, 9000 Ghent, Belgium

⁴ CENTRA, Faculdade de Ciências da Universidade de Lisboa, 1749-016 Lisbon, Portugal

⁵ INESC-ID, Instituto Superior Técnico, Rua Alves Redol, 9, 1000-029 Lisbon, Portugal