

Approximate Bayesian neural Doppler imaging

A. Asensio Ramos^{1,2}, C. J. Díaz Baso³, and O. Kochukhov⁴

¹ Instituto de Astrofísica de Canarias, C/Vía Láctea s/n, 38205 La Laguna, Tenerife, Spain
e-mail: andres.asensio@iac.es

² Departamento de Astrofísica, Universidad de La Laguna, 38206 La Laguna, Tenerife, Spain

³ Institute for Solar Physics, Dept. of Astronomy, Stockholm University, AlbaNova University Centre, 10691 Stockholm, Sweden

⁴ Department of Physics and Astronomy, Uppsala University, Box 516, Uppsala 75120, Sweden

Received 16 August 2021 / Accepted 9 December 2021

ABSTRACT

Aims. The non-uniform surface temperature distribution of rotating active stars is routinely mapped with the Doppler imaging technique. Inhomogeneities in the surface produce features in high-resolution spectroscopic observations that shift in wavelength because of the Doppler effect, depending on their position on the visible hemisphere. The inversion problem has been systematically solved using maximum a posteriori regularized methods assuming smoothness or maximum entropy. Our aim in this work is to solve the full Bayesian inference problem by providing access to the posterior distribution of the surface temperature in the star compatible with the observations.

Methods. We use amortized neural posterior estimation to produce a model that approximates the high-dimensional posterior distribution for spectroscopic observations of selected spectral ranges sampled at arbitrary rotation phases. The posterior distribution is approximated with conditional normalizing flows, which are flexible, tractable, and easy-to-sample approximations to arbitrary distributions. When conditioned on the spectroscopic observations, these normalizing flows provide a very efficient way of obtaining samples from the posterior distribution. The conditioning on observations is achieved through the use of Transformer encoders, which can deal with arbitrary wavelength sampling and rotation phases.

Results. Our model can produce thousands of posterior samples per second, each one accompanied by an estimation of the log-probability. Our exhaustive validation of the model for very high-signal-to-noise observations shows that it correctly approximates the posterior, albeit with some overestimation of the broadening. We apply the model to the moderately fast rotator II Peg, producing the first Bayesian map of its temperature inhomogeneities. We conclude that conditional normalizing flows are a very promising tool for carrying out approximate Bayesian inference in more complex problems in stellar physics, such as constraining the magnetic properties using polarimetry.

Key words. stars: atmospheres – stars: activity – line: profiles – methods: data analysis – stars: individual: II Peg

1. Introduction

Different classes of stars with non-uniform surface structures exhibit rotational modulation of their spectra. This variability is produced by the changing visibility of surface spots in the course of stellar rotation. When observed at high spectroscopic resolution, the Doppler-broadened line profiles of these stars show characteristic distortions – bumps or dips – propagating across the line as the star rotates. This Doppler resolution of stellar surfaces enables the user to recover a wealth of information about the topology and evolution of surface spots with the technique known as Doppler Imaging (DI, e.g. Rice 2002; Kochukhov 2016).

The first practical application of the spectroscopic starspot mapping dates back to the work of Deutsch (1958), who used a harmonic expansion method to study chemical abundance distributions on early-type chemically peculiar stars. Other investigations attempted to map spots on these stars with a trial-and-error approach (Khokhlova 1976). In a key development, Goncharskij et al. (1977, 1982) proposed a method to recover unparametrized two-dimensional chemical maps of early-type stars by recasting DI as a regularized ill-posed inversion problem. Subsequently, this version of DI was extended to mapping brightness (Collier Cameron & Horne 1986; Vogt et al. 1987) and temperature (Piskunov et al. 1990; Berdyugina 1998; Rice & Strassmeier

2000) inhomogeneities on the surfaces of active late-type stars and was applied to the reconstruction of isotopic composition maps (Adelman et al. 2002), non-radial pulsational velocity field (Kochukhov 2004), and even clouds in brown dwarf atmospheres (Crossfield et al. 2014).

With the advent of high-resolution spectropolarimetry, it became possible to apply the principles of tomographic mapping not only to intensity but also to polarization spectra in order to reconstruct vector maps of stellar surface magnetic fields (Piskunov & Kochukhov 1983, 2002; Semel 1989; Brown et al. 1991). This technique, known as Zeeman (magnetic) Doppler imaging (ZDI), is actively used today to study magnetic fields of essentially all classes of stars (e.g., Folsom et al. 2018; Kochukhov et al. 2019; Strassmeier et al. 2019) and currently represents the only viable approach for obtaining information about the magnetic field topology of stars other than the Sun.

Similar to almost any inversion problem in physics, DI and ZDI inversions are ill posed. This means that information in observational data is often insufficient for unique parameter inference. To counteract this issue, additional constraints, often based on consideration of simplicity, smoothness, or least amount of information, have to be implemented to ensure convergence of spectral fitting to a unique solution and the stability of this solution against random noise in the data. The

two most popular regularization strategies employed in DI are the Tikhonov regularisation (Tikhonov & Arsenin 1977) and the maximum entropy method (Skilling & Bryan 1984). On the other hand, most modern implementations of ZDI rely on a spherical harmonic decomposition of stellar magnetic field and implement regularization by truncating this harmonic expansion or penalizing its higher-order harmonic terms (e.g., Hussain et al. 2000; Kochukhov et al. 2014).

Despite the enormous success of regularized optimization methods, we still lack a fully probabilistic solution to the tomographic mapping problem. Dealing probabilistically with the very high-dimensional problem of inferring the temperature at all points of the stellar surface turns out to be computationally challenging using standard tools. As a solution, we propose in this work a Bayesian solution to DI that is also extremely efficient thanks to the use of recent tools developed in the field of deep learning. Our focus in this paper is on DI but this can be easily extended to ZDI and abundance mapping (which might even include physically realistic constraints).

2. Amortized neural posterior estimation

Given that the DI problem is ill defined, one needs to resort to a probabilistic description and solve the inference problem in the Bayesian framework (see, e.g., Gregory 2005). Assuming an arbitrary pixelation of the surface of the star, our aim is to obtain the vector \mathbf{T} of all temperatures in the surface of the star that are compatible with the observations. In general, the observations consist of the spectrum of the star – either individual spectral lines, a set of spectral regions, or even an average line profile for cases with reduced signal-to-noise ratio (S/N) – for its different rotational phases. We use the generic notation $I_i(\lambda)$ to refer to the observed spectrum at the i -th rotation phase. For notational convenience, we use $D = \{I_i(\lambda), i = 1 \dots N\}$ to represent the spectra obtained at all N observed rotation phases of the star. The fully Bayesian solution of the DI problem requires the computation of $p(\mathbf{T}|D)$, that is, the posterior distribution for \mathbf{T} conditioned on D . The posterior distribution can be calculated by direct application of the Bayes theorem (Bayes 1764),

$$p(\mathbf{T}|D) = \frac{p(D|\mathbf{T})p(\mathbf{T})}{p(D)}, \quad (1)$$

where $p(D|\mathbf{T})$ is the likelihood, which describes the generation of the observations conditioned on the temperature on the surface of the star, while $p(\mathbf{T})$ is the prior distribution of all possible surface temperatures. The quantity $p(D)$ is the so-called evidence or marginal posterior, which transforms the posterior into a properly normalized probability distribution. The evidence is unimportant in parameter estimation with Bayesian inference, and so we drop it in the following.

All information available for \mathbf{T} is encoded on the posterior distribution. From this posterior, one can extract summaries like the most probable surface temperature, uncertainties, and correlations. However, when the dimensionality of \mathbf{T} is large because one is interested in a very fine description of the surface of the star, the posterior distribution is a very high-dimensional object. Obtaining samples from the posterior distribution using typical Markov chain Monte Carlo techniques (even with the more advanced Hamiltonian Monte Carlo methods) turns out to be extremely difficult and time consuming. On the contrary, obtaining point estimates like the maximum a posteriori (MAP) solution is relatively straightforward. Once a proper likelihood and priors are defined, the MAP solution can be computed with

standard optimization methods. From a practical point of view, it is customary to optimize the log posterior instead (the logarithm is a monotonic operation and does not affect the location of the maxima):

$$\log p(\mathbf{T}|D) = \log p(D|\mathbf{T}) + \log p(\mathbf{T}). \quad (2)$$

The most widespread assumption used to compute the MAP solution is that the observations are corrupted with uncorrelated Gaussian noise. In this case, and assuming that $S_i(\lambda_j, \mathbf{T})$ is the synthetic forward model of choice to fit the observations at phase i and wavelength sample λ_j , the log-likelihood can be written as

$$\log p(D|\mathbf{T}) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^{N_\lambda} \left[\frac{(S_i(\lambda_j, \mathbf{T}) - I_i(\lambda_j))^2}{\sigma_{ij}^2} - \frac{1}{2} \log 2\pi\sigma_{ij}^2 \right], \quad (3)$$

where σ_{ij}^2 is the phase- and wavelength-dependent noise variance. The number of wavelength points we consider is N_λ . Typically, the forward model computes the local profiles in every point in the visible hemisphere of the rotating star at each phase and adds them together to produce the observed spectrum. Additional effects on the spectra produced by the limited spectral resolution of the spectrograph are customarily included via convolution with an instrumental profile.

The MAP solution is then computed by maximizing the log posterior with an appropriate selection of the prior term, $\log p(\mathbf{T})$. All current DI methods (and their magnetic counterparts aimed at modeling polarimetry) can be understood as MAP solutions to the problem and they only differ in the specific details of the observations, the forward model, and the prior assumptions. It is a well-known fact that priors are needed for the solution of this problem because the observations do not necessarily strongly constrain the solution simply through the likelihood. Different types of priors, specifically the maximum entropy prior or the Tikhonov regularization, can be encoded using well-known functional forms for $\log p(\mathbf{T})$.

Despite the success of these approaches, some challenges still remain to be solved. First, point estimates can be biased and might not be fully representative of the solution, especially in complex cases. One of the pathological cases can be multimodal solutions produced by strong ambiguities in the forward problem. Second, it is not always possible to obtain reliable uncertainties or correlations for DI, and so it may be difficult to assess the reliability of the solution. It is important to know, for instance, the uncertainty on the position and size of a predicted starspot. Finally, all MAP approaches use a simplified likelihood under the assumption of uncorrelated noise. However, there might be additional latent variables that we know how to model (e.g., small-scale activity, effects of stellar inclination, projected rotational velocity and differential rotation, and continuum normalization) but that are extremely hard or impossible to incorporate into the likelihood function.

Approximate Bayesian computation (ABC; Rubin 1984; Beaumont et al. 2002) offers a way to deal with many of these complications, providing the possibility to carry out Bayesian inference with simulations, which is appropriate for those cases in which a likelihood cannot be properly defined (either intrinsically or because of computational limitations). Although classical approaches to ABC have been successful, they suffer from some problems related to their inefficiency in large-scale problems (Cranmer et al. 2020). Recent advances in machine learning

(especially with the irruption of deep learning) have provided the machinery to efficiently deal with these large-scale problems. Here, we leverage the ideas behind amortized neural posterior estimation (ANPE; Cranmer et al. 2020; Díaz Baso et al. 2022). The aim is to train a complex neural network that directly learns the posterior distribution $p(\mathbf{T}|D)$. We model it using normalizing flows (Kobyzev et al. 2020), which we describe in the following, and use simulations for training. These simulations can incorporate any arbitrary latent process described by generic variables z , so that the posterior distribution of interest is obtained by marginalizing over the latent variables:

$$p(\mathbf{T}|D) = \int p(\mathbf{T}, z|D) dz = \int p(D|\mathbf{T}, z) p(z|\mathbf{T}) p(\mathbf{T}) dz. \quad (4)$$

As said, these latent processes can range from stochastic activity in the star in the form of small-scale dark structures characterized by a probability distribution, to systematic effects in the spectrograph and telescope. As long as proper simulations of these processes can be carried out, they can be incorporated into our approach. The resulting neural network will produce inferences marginalizing over all these uninteresting processes.

3. Synthetic stars and their spectra

We model the surface temperature of the star using Hierarchical Equal Area and isoLatitude Pixelation (HEALPix¹). HEALPix produces pixels of equal area, and efficient software to deal with such grids is available. For this reason, it is now commonly used for pixelation of the sphere, especially in astrophysics (e.g., Planck Collaboration I 2020). We use $N_{\text{side}} = 16$, which results in a total of 3072 pixels distributed over the surface of the star. This number of pixels is high enough to avoid artifacts in the synthetic spectra and low enough to allow affordable computation of the training set. In our case, we generate 100k random surface temperatures.

The surface temperature maps are obtained following a recipe, which then needs to be considered as a prior. As such, the posterior distribution produced by our model depends on this prior. In any case, we point out that this prior can be easily updated by changing the recipe, which can be as complex as needed. Adding complexity is especially relevant for cases such as ZDI, where the additional constraint of zero divergence for the magnetic field has to be fulfilled. Other additional physical constraints can be imposed.

Stars are initialized by randomly choosing a temperature T_{star} uniformly between 4000 and 5500 K. The number of spots is randomly selected between 1 and 10. Each spot is randomly placed on the surface of the star with no preferred latitude. The spot is chosen to be centered on the random position and with a radius randomly extracted from a triangular distribution between 0.1 and 1 rad (6–57°) with a mode equal to 0.1 (this generates more small spots than large ones). The temperature of the spot is chosen randomly between 3200 K and $1.2 T_{\text{star}}$, limited to a maximum of 5800 K. This generates spots with temperatures lower or higher than T_{star} . The ranges for the photospheric and spot temperatures, although somehow arbitrary, cover the expected values for cool stars. They are also limited by the range of temperatures of the grid of models that we use; we explain this grid in the following. However, these ranges can be extended to other regimes if the atmospheric models and the spectral synthesis code are available; they are not a limitation of the method presented here.

¹ <http://healpix.sourceforge.net>

The synthetic spectra of each star at an arbitrary number of N rotation phases $\{\phi_1, \dots, \phi_N\}$ is computed by assuming that the emergent radiation at each point in the observable hemisphere is obtained from a model atmosphere with the temperature of the pixel extracted from the MARCS grid (Gustafsson et al. 2008). Spectra are computed under the assumption of local thermodynamical equilibrium (LTE) using a microturbulent velocity of 2 km s^{-1} . Every star rotates with a rotation velocity v_{rot} randomly selected between 10 and 80 km s^{-1} . The inclination angle of the rotation angle (with $i = 0^\circ$ a star with the rotation axis pointing to the observer) is chosen isotropically between 10° and 85° , distributed uniformly in $\cos i$. The case of i and $180^\circ - i$ cannot be distinguished in most DI problems (with the exception of the ZDI that uses four Stokes parameter spectra, e.g., Kochukhov et al. 2019), and so we choose $i < 90^\circ$. The number of observed phases is also chosen randomly from a minimum of 6 and a maximum of 20.

In order to accelerate the computation of the spectra for the 100k stars, we precompute the MARCS spectra for 19 temperatures between 3000 and 6000 K (in steps of 100 K from 3000 to 4000 K and in steps of 250 K from 4000 to 6000 K). Spectra for each temperature are computed at 15 astrocentric angles (μ) between 0.02 and 1 and 160 Doppler shifts in the range $[-80, 80] \text{ km s}^{-1}$. This allows us to calculate the emergent spectrum in each pixel by a simple tri-linear interpolation, which largely accelerates the process. The synthesis of the 100k stars with our nonoptimized Python code takes $\sim 4 \text{ h}$ on 40 CPU cores.

Given that we apply our model to observations from II Peg in Sect. 6, we degrade the synthetic spectra to mimic the observations with the ESPaDOnS spectropolarimeter (Donati 2003) at the Canada-France-Hawaii Telescope (CFHT). We first synthesize the spectrum for the three Fe I lines and relevant blends in the spectral regions at (5985.1, 5989.0) Å, (6000.1, 6005.0) Å, and (6022.0, 6026.0) Å with a sampling of 0.01 Å. Later, these are degraded, modeling an observation with a spectral resolution of $R = 65\,000$. This is done by convolving the emergent spectra with a Gaussian kernel of appropriate width. The resulting spectra are resampled at a wavelength step of 0.03 Å, which is appropriate for ESPaDOnS. Finally, uncorrelated Gaussian noise with zero mean and standard deviation of 10^{-3} is added to the spectrum.

4. The model

The model proposed in this work for solving the DI problem is shown schematically in Fig. 1. Although we describe all the individual components of the model in detail in this section, let us first discuss it in general. The main component of the model is the normalizing flow (green block), which uses flexible neural networks to transform Gaussian noise into samples from the posterior. We describe it in detail in Sect. 4.1. This flow is conditioned on a context vector that serves as a summary of the observations. This encoder (yellow block) is complex because it needs to deal with spectral lines observed at an arbitrary number of rotation phases. This encoder is described in Sect. 4.3. Given the large dimensionality of the stellar surface temperature maps, we compress them using a pre-trained autoencoder, which is discussed in Sect. 4.2. During the training of the flow and observational encoder, surface temperature maps will be encoded with the pre-trained encoder (violet block) and compared with outputs of the normalizing flow. During evaluation, samples from the normalizing flow will be transformed into surface temperature maps with the pre-trained decoder (blue block).

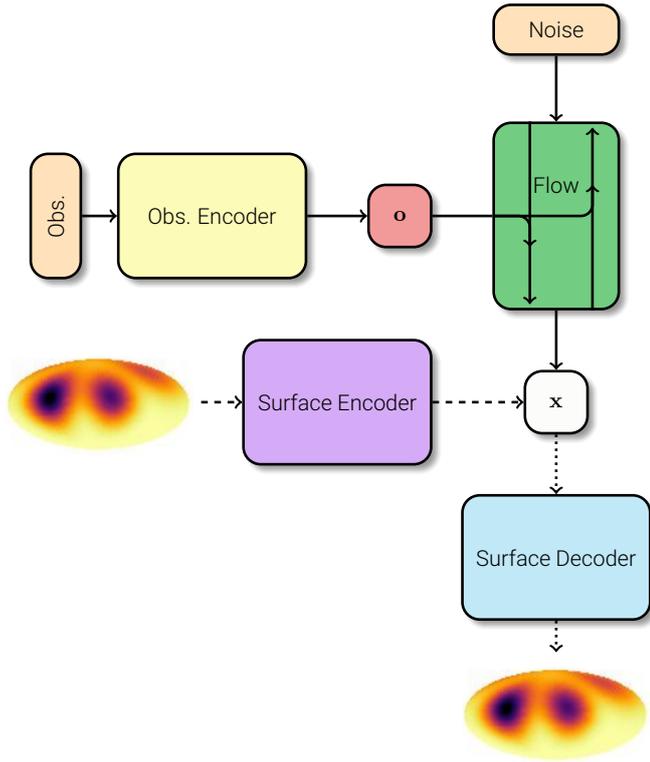


Fig. 1. Schematic representation of the model. The blocks are described in detail in Sect. 4. Solid lines show connections between blocks that propagate gradients during training of the flow and observational encoder. Dashed lines in the autoencoder do not back-propagate gradients during training. Dotted lines are only used in validation to produce surface-temperature maps.

4.1. Normalizing flows

As described in Sect. 2, our aim is to directly model the posterior distribution $p(\mathbf{T}|D)$. To this end, we leverage normalizing flows as a very flexible, tractable, and easy-to-sample family of generative models that can approximate complex distributions. Simply put, a normalizing flow is a transformation of a simple probability distribution (often a multivariate standard normal distribution with zero mean and unit covariance) into the desired probability distribution. Normalizing flows accomplish this by the application of a sequence of invertible and differentiable variable transformations. Let us assume that \mathbf{Z} is a d -dimensional random variable with a simple and tractable probability distribution $q_{\mathbf{Z}}(\mathbf{z})$, with the condition that it is fairly straightforward to sample (that is why standard normal distributions are often the distribution of choice). Let $\mathbf{X} = f(\mathbf{Z})$ be a transformed variable with a function f that is invertible. If this condition holds, then $\mathbf{Z} = g(\mathbf{X})$, where $g = f^{-1}$. The change of variables formula states that the probability distribution of the transformed variable is given by

$$q_{\mathbf{X}}(\mathbf{x}) = q_{\mathbf{Z}}(g(\mathbf{x})) \left| \det \left(\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \right) \right|. \quad (5)$$

The term $\partial g(\mathbf{x})/\partial \mathbf{x}$ is the Jacobian matrix and takes into account the change in probability volume during the transformation. Its role is to force the resulting distribution to be a proper probability distribution with unit integrated probability. As the transformation is invertible, the equality $\partial g(\mathbf{x})/\partial \mathbf{x} = (\partial f(\mathbf{z})/\partial \mathbf{z})^{-1}$ holds, so

that one can rewrite the previous expression as

$$q_{\mathbf{X}}(\mathbf{x}) = q_{\mathbf{Z}}(\mathbf{z}) \left| \det \left(\frac{\partial f(\mathbf{z})}{\partial \mathbf{z}} \right) \right|^{-1}. \quad (6)$$

Designing invertible transformation that can be trained to produce generative models over complex datasets is difficult. For this reason, normalizing flows make use of the fact that the composition of invertible transformations is also invertible. Then, if $f = f_M \circ f_{M-1} \circ \dots \circ f_1$, the transformed distribution is

$$q_{\mathbf{X}}(\mathbf{x}) = q_{\mathbf{Z}}(\mathbf{z}) \prod_{i=1}^M \left| \det \left(\frac{\partial f_i(\mathbf{y}_i)}{\partial \mathbf{y}_i} \right) \right|^{-1}, \quad (7)$$

where $\mathbf{y}_i = f_{i-1} \circ \dots \circ f_1(\mathbf{z})$ and $\mathbf{y}_1 = \mathbf{z}$. Compositional invertible transformations have made it possible to define very flexible normalizing flows through the use of deep neural networks. Arguably the simplest invertible transformation is a linear operation, which can introduce correlation among dimensions. However, these are not sufficiently expressive. Perhaps the most successful approach to defining expressive invertible transformations is via coupling layers (Dinh et al. 2014). The idea behind coupling layers is that they randomly split the d -dimensional variable \mathbf{z} at each step of the flow in two disjoint sets, \mathbf{z}^A of dimension p and \mathbf{z}^B of dimension $d - p$, so that $\mathbf{z} = (\mathbf{z}^A, \mathbf{z}^B)$. The transformation is defined as

$$\begin{aligned} \mathbf{y}^A &= h(\mathbf{z}^A; s(\mathbf{z}^B)) \\ \mathbf{y}^B &= \mathbf{z}^B, \end{aligned} \quad (8)$$

where $s(\mathbf{z}^B)$ is, in general, any arbitrary transformation of \mathbf{z}^B , while h is the coupling function. Equation (8) can be trivially inverted if h is invertible, and the ensuing Jacobian of the transformation is simply the Jacobian of h . The power of coupling resides in the complexity of s , which can be arbitrary, so that complex deep neural networks can be leveraged to produce very flexible normalizing flows.

We use the rational quadratic neural spline flows of Durkan et al. (2019) as our coupling functions of choice. They have been demonstrated to produce very good results in approximating high-dimensional complex probability distributions. In this approach, the coupling functions are monotone rational quadratic splines, where the knots defining the splines are obtained by applying a neural network to \mathbf{z}^B .

Our specific normalizing flow model consists of the sequential application of five steps to allow for a more expressive posterior distribution modeling. Each step contains a linear transformation using an LU decomposition (Kingma & Dhariwal 2018) and a neural spline coupling transformation. The neural network s is a fully connected residual network (He et al. 2015) with a hidden dimensionality of 256 and rectified linear units (ReLU; Nair & Hinton 2010) as activation functions. A dropout of 0.4 (chosen after a trial and error process) is applied to prevent overfitting (Srivastava et al. 2014).

As we want to model the posterior distribution $p(\mathbf{T}|D)$, we need to take into account the conditioning on the observed data. Conditional generative models can be modeled with normalizing flows in a straightforward manner by conditioning all previous probability distributions on a latent vector. We discuss how a latent vector can be extracted from the observations for conditioning in Sect. 4.3. From a practical point of view, it is enough to concatenate the conditioning vector \mathbf{o} to the input of the residual network, so that one uses $s(\mathbf{z}^B, \mathbf{o})$ instead of simply $s(\mathbf{z}^B)$.

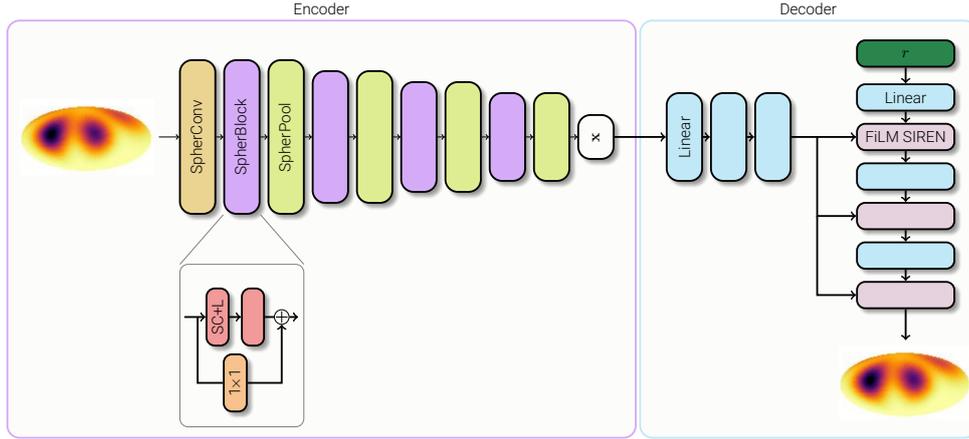


Fig. 2. Autoencoder used for the compression of stellar temperature surface maps in a latent representation of reduced dimensionality. The encoder (violet block) uses spherical convolutions and pooling in the HEALPix representation. The decoder (blue block) uses modulated SIREN layers described by Eqs. (9) and (10).

4.2. Autoencoder

Although normalizing flows have been shown to do a good job even with very high-dimensional spaces (ours is of 3072 dimensions), we find that reducing the dimensionality produces much better predictions and the training of the model is greatly accelerated. For this reason, we employ an autoencoder to compress the surface temperatures in the training set to a latent space of 64 dimensions, with a reduction of a factor 48. An autoencoder can be seen as a nonlinear extension of the linear principal component analysis (PCA; Pearson 1901). It is built by putting together an encoder and a decoder, with a bottleneck between the two. The encoder neural network (E) encodes the input (stellar surface temperature maps) into a latent vector of reduced dimensionality. The decoder neural network (D) then tries to reproduce the output again from the latent vector. The autoencoder is trained with the 100k stellar surface temperature maps by optimizing $|\mathbf{T} - \mathbf{D}(\mathbf{E}(\mathbf{T}))|^2$, which is the mean squared error loss function between the input and the output. A schematic representation of the autoencoder is shown in Fig. 2.

The encoder is a convolutional neural network. However, given that we are dealing with temperatures in the surface of a sphere, we take advantage of the especially simple definition of 3×3 convolutions in the surface of a sphere developed by Krachmalnicoff & Tomasi (2019)². This exploits the spatial information on the sphere to produce expressive intermediate representations. The encoder starts by encoding the input into 32 channels, which then go through four stages of residual spherical convolutional blocks followed by an average pooling. After every pooling, the number of channels is doubled. The residual blocks consist of two spherical convolutions preceded by leaky rectified linear units (LeakyReLU) and a residual connection with a learnable kernel of size 1. The latent vector \mathbf{x} is obtained after applying a fully connected layer that produces a vector of 64 dimensions.

The decoder takes advantage of recent ideas in the field of implicit neural representation. Specifically, we use a SIREN (sinusoidal representation network; Sitzmann et al. 2020), a simple multilayer perceptron (MLP) that takes the xyz coordinates of a point on the surface of the star as input and returns the temperature of this point as output. SIRENs have been shown

to model many complex signals (e.g., images and solutions to partial differential equations) with great flexibility and precision. Given that SIRENs can be trivially extended to produce several outputs, they can be used as decoders in more complex problems like ZDI. As we want this network to model all stars in our training set, we condition the SIREN to the latent vector \mathbf{x} extracted by the encoder following the approach of Chan et al. (2020). The SIREN computes the following function composition for the position on the surface described by the vector \mathbf{r} :

$$T(\mathbf{r}) = \mathbf{W}_3 \phi_2^{\mathbf{x}} \circ \phi_1^{\mathbf{x}} \circ \phi_0^{\mathbf{x}}(\mathbf{r}), \quad (9)$$

where we explicitly state that each layer of the SIREN is conditioned on \mathbf{x} . This conditioning is done via a feature-wise linear modulation (FiLM; Perez et al. 2017), which incorporates the latent information by projecting it via a simple three-layer MLP, using scaling, $\gamma_i(\mathbf{x})$, and bias, $\beta_i(\mathbf{x})$, modifiers. The operation of each SIREN layer is then given by

$$\phi_i^{\mathbf{x}}(\mathbf{y}) = \sin(\gamma_i(\mathbf{x}) \cdot (\mathbf{W}_i \mathbf{y} + \mathbf{b}_i) + \beta_i(\mathbf{x})). \quad (10)$$

The SIREN works internally using a representation of size 128, and a final layer (\mathbf{W}_3) produces a single number from this representation. One of the advantages of using a SIREN as a decoder comes from their inherent interpolation capabilities. Although trained on data sampled at 3072 HEALPix pixels, one can later reproduce the temperature at any arbitrary position on the star. As such, one can potentially use the autoencoder as a prior in standard MAP approaches to DI and use a latent space of low dimensionality during the optimization.

We trained the autoencoder for 200 epochs with a batch size of 128 with the Adam optimizer (Kingma & Ba 2014) with a learning rate of 3×10^{-4} , which is decreased by a factor of 0.5 every 60 epochs. The number of trainable parameters is 1.547M. We check for overfitting using 10k stars as validation and find no evidence of it. The results show that the dimensionality of the latent space is enough to produce good representations of the stellar surface, with a root mean square difference of only ~ 50 K, although with larger differences in some cases. The difference between the spectrum from the original star and that obtained from the star passed through the autoencoder is shown in Fig. 3. For the majority of cases, the difference is small, with a standard deviation of $\sim 10^{-3}$. However, we also find some cases with

² We use our PyTorch implementation found in <https://github.com/aasensio/sphericalCNN>

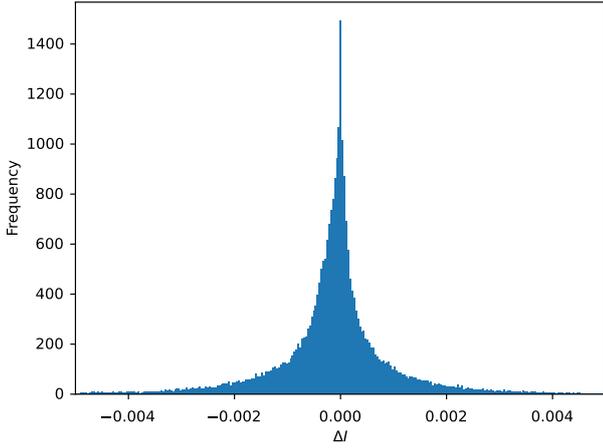


Fig. 3. Histogram for ΔI , the difference between the synthetic line profiles in the original star and those from the star produced by the autoencoder.

larger differences, which may have a small impact on the spectral reconstruction.

4.3. Context: transformer encoders

Conditioning the normalizing flow on the observations is done via a specific encoder, which summarizes the observations into a vector of 256 dimensions. The encoder we need is nontrivial because it has to deal with two unknowns. First, although we fixed the spectral regions of interest, the number of samples in the spectral direction can change from one observation to another. This is a minor inconvenience because one can always apply interpolation to sample all observations in the same wavelength or velocity axis. The second unknown is the number of observed phases and their specific values, which cannot be fixed a priori, and so dealing with it is crucial. Our proposal is to use Transformer encoders (Vaswani et al. 2017). Transformers can transform a sequence of arbitrary length into a sequence of the same length of features of arbitrary dimensionality using self-attention. The architecture used in this work is displayed in Fig. 4.

The input to the wavelength Transformer encoder for each phase consists of a sequence of N_λ wavelength points containing three features. The first one is a quantity proportional to the wavelength. In our case, we simply use a monotonic vector in the range (0, 1) in the three spectral regions considered, ignoring the wavelength jump between regions. We anticipate that more elaborate features can also be implemented. The second feature is simply the mean spectrum computed for all observed phases, which is obviously the same for all phases. The last one is the residual at each phase with respect to the mean. We concatenate these three features for all phases and wavelengths of a given batch during training, forming a tensor of $(B \cdot T, N_\lambda, 3)$ dimensions, where B is the batch size and T is the largest sequence length in the batch (we apply binary masks to use the appropriate length for each element of the batch). A linear layer produces a feature tensor of size $(B \cdot T, N_\lambda, 256)$. To help with convergence, layer normalization is then applied to the input data (Lei Ba et al. 2016). We note that we follow the most recent practice of normalizing before entering the learnable layers, contrary to the original work of Vaswani et al. (2017), who apply normalizations after the learnable layers. Matrices of values (**V**), queries (**Q**), and keys (**K**) are built by multiplying the inputs for the whole sequence

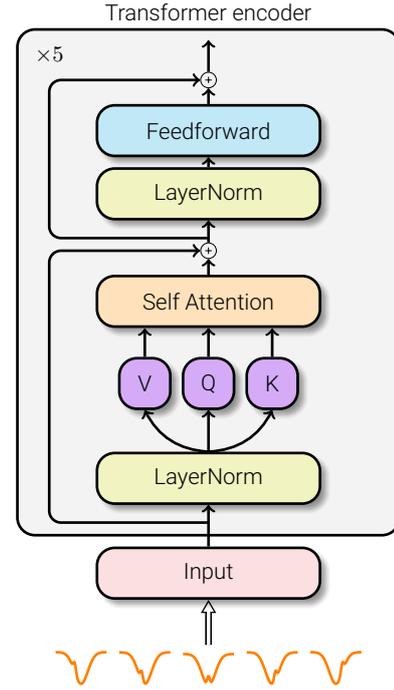


Fig. 4. Internal structure of the Transformer encoder, based on self-attention, to deal with an arbitrary number of phases in time and samples in the spectral direction.

with trainable matrices. The self-attention layer then computes the following operation:

$$\text{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}, \quad (11)$$

where d_k is the dimensionality of the queries and keys, which we set to 256 in our case. The product of the query and key matrices is a score matrix. This matrix provides information on how much focus one element of the sequence has to put on other elements of the sequence, and then shares information of the whole sequence. This score matrix is then scaled down to allow for more stable gradients, and a softmax is applied to transform the scores into probabilities. Finally, these attention weights are applied to the values, which are then normalized again and passed through a two-layer fully connected network with ReLU activation function. We note that the model also contains two residual connections that accelerate training by reducing gradient vanishing effects. This Transformer layer is repeated five times in our case. We additionally use two heads (two such encoders computed in parallel) that will ideally focus on different parts of the sequence, and their outputs are concatenated at the output. The resulting output of the Transformer encoder therefore has $(B \cdot T, N_\lambda, 256)$ dimensions, where all elements of the output sequence have attended to all elements of the input sequence, as explained above.

We then average over the spectral direction and reshape the tensor to dimensions $(B, T, 256)$. These tensors can then be considered to encode the necessary spectral information for every phase of each observation of the batch. It is at this moment where it is crucial to include information about the specific phase at which each observation is done. We do this by using a FiLM layer, which linearly modulates the encoding tensor using a scaling and a bias function obtained from the phase (as a real

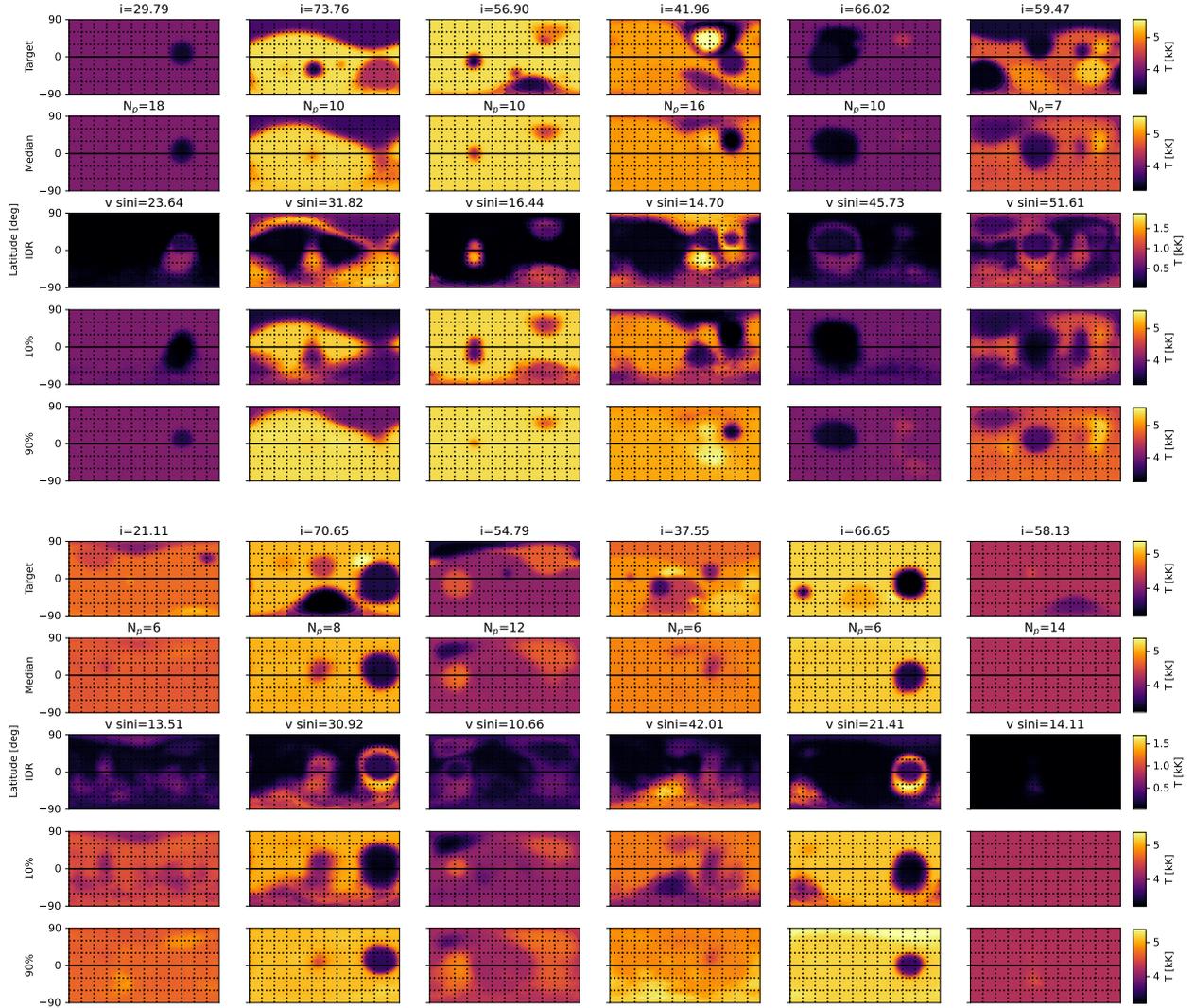


Fig. 5. Results of the Bayesian inference for 12 (6 per panel) validation stars. The rotation velocity, number of phases, and $v \sin i$ of each star are indicated. The first row shows the target surface temperature map. The second and third rows show the median and the interdecile range. The last two rows display the percentiles 10 and 90. All units are given in kK.

number between 0 and 1). A second Transformer encoder following exactly the same description as above then aggregates the temporal information observations at different phases), giving a tensor of size $(B, T, 256)$ as output. This tensor is averaged over all observed phases, resulting in a tensor of $(B, 256)$ dimensions. Finally, we add the information about the rotation velocity (normalized to 80 km s^{-1}) and inclination of the axis of rotation of the star (we use $\sin i$, which is constrained to be between 0 and 1), again with a FiLM layer. The result is the context conditioning vector \mathbf{o} of 256 dimensions, which is then fed into the flow to estimate the posterior distribution.

4.4. Training

Using Eq. (7), and conditioning it on the observations, the normalizing flow produces the following approximation to the posterior (transformed to logarithms):

$$\log q_{\mathbf{X}}(\mathbf{x}|\mathbf{o}) = \log q_{\mathbf{Z}}(\mathbf{z}) + \sum_{i=1}^M \log \left| \det \left(\frac{\partial f_i(\mathbf{y}_i|\mathbf{o})}{\partial \mathbf{y}_i} \right) \right|^{-1}. \quad (12)$$

Our aim is to train the parameters of the model (that we encode in the vector θ) so that the approximate posterior produces

a good reproduction of the real posterior distribution. We do this by minimizing the expected value of the Kullback-Leibler divergence over all possible observations:

$$L(\theta) = \int d\mathbf{o} p(\mathbf{o}) \int d\mathbf{T} p(\mathbf{T}|\mathbf{o}) [\log p(\mathbf{T}|\mathbf{o}) - \log q(\mathbf{T}|\mathbf{o}, \theta)]. \quad (13)$$

The first term is indeed independent of θ and can be considered to be constant. Neglecting this term and applying the Bayes theorem, we can rewrite the loss function as

$$L(\theta) = C - \int d\mathbf{T} p(\mathbf{T}) \int d\mathbf{o} p(\mathbf{o}|\mathbf{T}) \log q(\mathbf{T}|\mathbf{o}, \theta), \quad (14)$$

where C is independent of θ but is difficult to compute because we do not have direct access to the posterior $p(\mathbf{T}|\mathbf{o})$. Dropping this unimportant constant, this expression is especially convenient because one can perform the following Monte Carlo estimation (up to a constant) using a batch of size B :

$$L(\theta) \approx -\frac{1}{B} \sum_{i=1}^B \log q(\mathbf{T}_i|\mathbf{o}_i, \theta), \quad (15)$$

where $\mathbf{T}_i \sim p(\mathbf{T})$ and $\mathbf{o}_i \sim p(\mathbf{o}|\mathbf{T}_i)$. In other words, the Monte Carlo estimation of the loss function simply requires sampling surface temperatures from the prior distribution and then using this to obtain the simulated observation, which includes the noise process (or any necessary latent process as described above).

All neural networks are implemented in PyTorch (Paszke et al. 2019a). The autoencoder is pretrained as described in Sect. 4.2 and the weights are frozen. The remaining elements of the architecture (normalizing flow, Transformer encoders, and FiLM conditioning layers) that are linked with solid arrows in Fig. 1 are trained end-to-end by minimizing the loss function of Eq. (15). We utilize the nflows library, which is an implementation of normalizing flows in PyTorch developed by Durkan et al. (2020). This library makes it easy to define normalizing flows and to obtain a good implementation of the neural spline flows that we use in this work (see also Díaz Baso et al. 2022). For the training, we apply the Adam optimizer for 100 epochs with a learning rate of 3×10^{-4} , which is decreased by a factor of 0.5 every 60 epochs. The training time per epoch in one single RTX 2080 Ti GPU is of the order of 14 minutes, giving a total training time of around one day. The number of trainable parameters is 11.3 M. Almost 6.5 M correspond to the normalizing flow, with each Transformer encoder having around 2.6 M parameters³.

We carried out a brief analysis of the impact of hyperparameters with the objective of finding the optimal ones. The baseline model we use in this work is definitively large (with more than 13 M parameters) but it can be reduced with a limited impact on the results. We verified that the size of the context vector \mathbf{o} and the hidden dimensionality of the residual network in the normalizing flow can both be reduced to 128 with reduced impact. The number of attention blocks in the Transformers can also be reduced to three. We also verified that increasing the number of steps in the normalizing flow to 15 barely affects the results.

Only 90% of the available 100 k is used for training, and we use the remaining 10% for validation purposes and check for overtraining; we find no evidence of it. The training proceeds in detail as follows. A batch of stars is extracted from the training set. This batch contains the observed spectra, the observed phases for each star, their rotation velocity and inclination of the rotation axis, and the corresponding stellar surface temperature map. The spectra, phases, velocity, and inclination angle are used by the two Transformer encoders and the FiLM layers to produce the context vector \mathbf{o} . At the same time, the surface encoder of the autoencoder is used to produce the latent vector \mathbf{x} for each surface temperature. The log-posterior of the normalizing flow is computed with the context and latent vectors. Backpropagation of this log-posterior allows us to modify the parameters of the normalizing flow, and the Transformer encoders and FiLM layers.

Once the model is trained, samples from the posterior are obtained as follows. The user produces the context vector \mathbf{o} from the observations, uses standard normal noise in the input to the normalizing flow, and computes the associated latent vector \mathbf{x} by applying the variable transformation rule learned by the normalizing flow (see Eq. (7)). Surface temperature maps can then be obtained by plugging the sampled latent vector into the decoder of the autoencoder. As all these operations are very fast, thousands of samples from the posterior can be obtained in less than a second. Although the high-dimensional posterior is the full

result of the inference, we use simple summaries to show graphical representations of our results. For instance, the marginal distribution of temperature in pixel i on the surface is obtained by computing the marginalization over the rest of the pixels:

$$p(T_i|D) = \int dT_1 \cdots dT_{i-1} dT_{i+1} \cdots dT_N p(\mathbf{T}|D). \quad (16)$$

This marginal distribution is simply computed by computing histograms from posterior samples for pixel i .

5. Validation

To validate the model and understand how to interpret the results, we generate a set of new synthetic temperature surface maps and compute the emergent spectra at an arbitrary number of observed phases for different values of the rotation velocity and inclination angle. Figure 5 shows the results for 12 different stars (six stars for each of the two panels) randomly sampled from the validation set. We use a cartesian projection (latitude-longitude) to show the surface. The reconstructed surfaces are obtained with the decoder of Fig. 2 for the coordinates \mathbf{r} of the HEALPix pixels for $N_{\text{side}} = 16$. The upper row in each panel displays the target temperature distribution on the surface of the star. The second row displays the median (percentile 50) of the posterior distribution, while the remaining rows try to summarize the uncertainty. The third row displays the interdecile range (IDR; the range between the percentiles 10 and 90 of the distribution). Given that the posterior can be far from Gaussian, the IDR is a measure of dispersion that better provides information on large excursions from the median than the standard deviation (the standard deviation is approximately given by $\text{IDR}/2.56$ in the Gaussian case). As the distribution in each pixel does not need to be symmetric, the fourth and fifth rows show the percentiles 10 and 90, which can be used to check the expected variability of the temperature in each pixel.

These results show that the median of the distribution is a fairly good representation of the temperature distribution on the surface of the star. Dark and bright spots are correctly obtained, although many small-scale structures are lost and cannot be recovered. Given that the training was done with $i \in [10^\circ, 85^\circ]$ to avoid ambiguities, the results are clearly biased towards a better reconstruction of the northern hemisphere. However, this does not affect the behavior of the algorithm. Despite this bias, we find no artifacts in the southern hemisphere. Of particular relevance are cases in which the rotation axis is pointing to the observer ($\sin i \sim 0$), for which almost no information about this hemisphere is encoded in the observations. The absence of artifacts is a direct consequence of the regularizing effect of the prior and the marginalization of the Bayesian inference. We find that the temperature in the southern hemisphere is roughly similar to the underlying temperature of the star.

Although the position in latitude and size of all inferred structures is correctly recovered on average, there is some ambiguity, as marked by the bright regions in the IDR. The regions in the shape of rings (e.g., the second case in the lower panel) point out to some uncertainty in the specific position of the spot, mainly in the latitude. However, this uncertainty is smaller than the size of the spot, both in latitude and longitude. The recovery of spots is quite robust, both in the regions colder and hotter than the surroundings, as indicated by the percentiles 10 and 90.

Though the percentiles and the IDR shown in Fig. 5 are good summaries of the dispersion of the posterior distribution, it is also good to see samples from the posterior distribution. An

³ A trained model and the tools needed for retraining can be found at <https://github.com/aasensio/bayesDI>

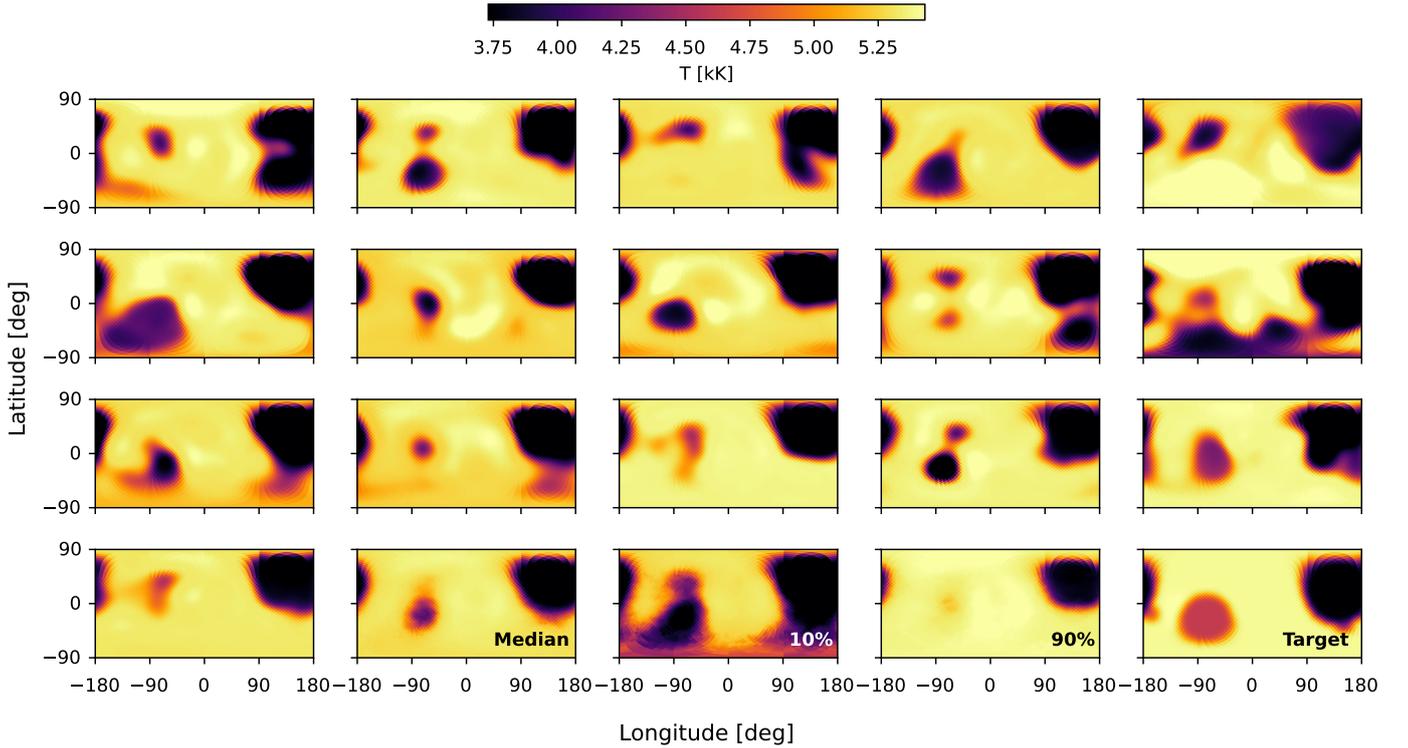


Fig. 6. Samples from the posterior distribution, median, percentiles 10 and 90, and the original target temperature surface map. This case has a rotation velocity of 49 km s^{-1} and an inclination angle of 75° , and the star was observed during nine phases randomly spread over the rotation period.

example is shown in Fig. 6, where we display 13 samples from the posterior, together with the median, percentiles 10 and 90, and the original map (all marked with labels). Samples from the posterior all appear similar, although spots are placed at different positions on the surface. The large cold spot and the slightly smaller, less cold spot are clearly visible in the median. The percentiles indicate that the posterior distribution is fairly asymmetric for the hotter spot, which is clearly visible in percentile 10 but is barely visible in percentile 90. All samples that are compatible with the observations display the large cold spot, although the hotter spot location is more uncertain. The ability to detect hotter spots crucially depends on the specific spectral lines selected, which might not be optimal in our case. The very small-scale spot between both large spots is not visible in the median map, fundamentally because its small size is below the typical resolution element of $2v_{\text{rot}} \sin i / W$ (e.g., Vogt et al. 1987), with W the considered Doppler widths of the lines, but we find hints of it in the percentile 10 map.

Figure 7 shows the synthetic spectra using samples from the posterior distribution of Fig. 6, also known as posterior predictive checks. For convenience, we display the residual between the original synthetic observations and each sample in orange curves. Violet curves show 68% interval estimates. The error bars act as visual guides indicating amplitudes of 0.02, 0.01, and 0.005 for blue, green, and red, respectively. It is clear from these results that, although there is no apparent bias on the inference, the approximated posterior distribution is too broad in some cases. Although the amplitude of the residuals is clearly close to the noise level of 10^{-3} in some phases, much larger amplitudes are found in other rotation phases. We thoroughly explored whether this can be solved by changing the hyperparameters and architecture of the model, with little success. Almost all models we considered successfully converge towards predicting the

correct median but with an overestimation of the uncertainty. A potential solution to this issue is to radically augment the training set, something we have not pursued due to computational limitations, but plan to do in the near future.

As a final validation of the results, we analyze how results behave when the inclination of the rotation axis changes for the same star. To this end, we display in Fig. 8 a tailor-made star of 5500 K with three spots of 3500 K and 0.2 rad (11.5°) of radius. Two of the spots are located at the equator with a longitude difference of 180° . The remaining two are located at longitude 0° and at latitudes 70° and -60° . The star is assumed to be rotating with an equatorial velocity of 30 km s^{-1} and observed at 15 equidistant rotation phases for a whole rotation. The lower panel displays the target temperature map in cartesian projection. The four panels in the left column show the median of the inferred temperature posterior distribution for four values of the inclination angle. The three central columns show the percentiles 10 and 90 and the IDR, respectively. The rightmost column is a simple pictorial representation of the visibility of the active regions with rotation, obtained by simply averaging the temperature over many rotations. The figure shows the position of the dark spots relative to the observer.

The spot located close to the north pole of the star is captured for observations $i = 10^\circ$, 35° and $i = 60^\circ$. The Doppler effect for this spot for observations at $i = 10^\circ$ is weak because it is almost on the plane of the sky, but the spot is correctly located. For observations at $i = 85^\circ$, although the Doppler effect is maximum, the projected area on the plane of the sky is minimum and the results show only a hint of its detection in the lowest percentile. The spot in the southern hemisphere barely appears in the median maps and is only clearly found in the percentile 10% maps. The equatorial spots are very nicely detected in the median maps and in both percentiles, although there are

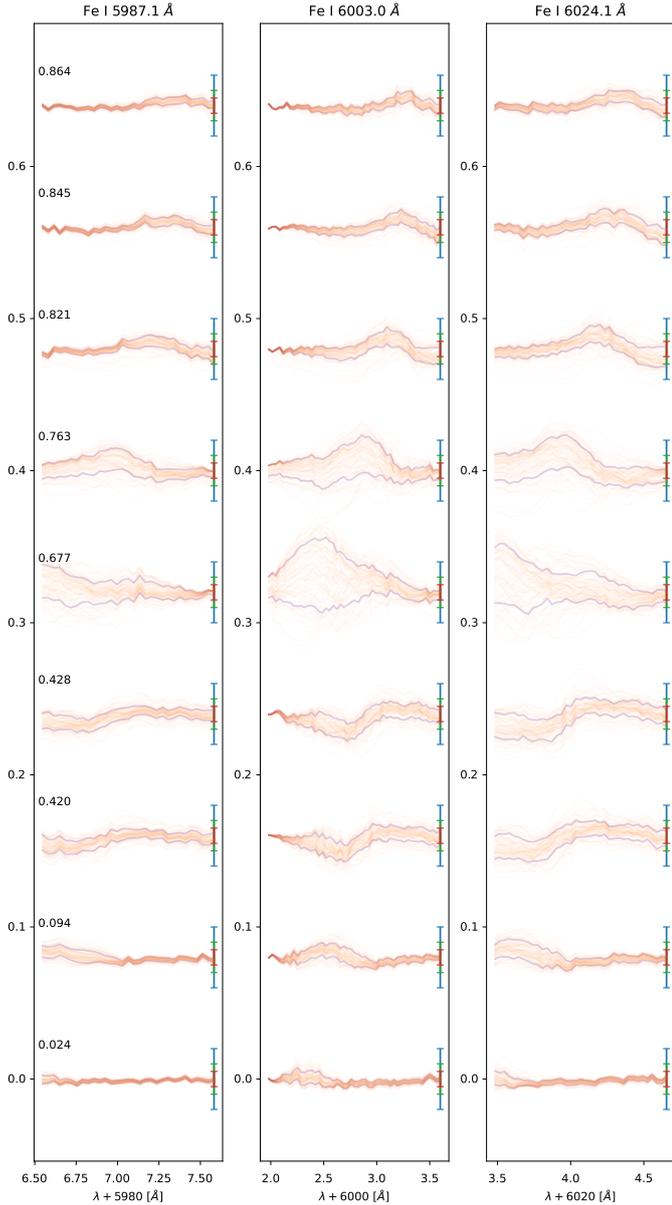


Fig. 7. Residual from the synthesis in the posterior samples and the observations for the case of Fig. 6. Samples are shown in orange, with violet curves indicating the 68% interval estimate. The red, green, and blue error bars show residuals of 0.02, 0.01, and 0.005, respectively.

large uncertainties on the precise position of the spot in the percentile 10% map. However, the specific location of these spots for an observation at $i = 10^\circ$ is clearly incorrect. Such poor reconstruction is not surprising because the projected velocity is $v_{\text{rot}} \sin i \approx 5 \text{ km s}^{-1}$. The percentiles indicate some confusion in the inferred position of these spots and the one in the southern hemisphere. Interestingly, we witness the appearance of a bias towards large temperatures in the areas surrounding the dark spots. These structures are also seen in some of the examples of Fig. 5. This is a consequence of the uncertainty in the exact location in longitude of the spots.

6. Application to II Peg

With the method validated with synthetic observations, we apply it to observations of the moderately fast rotator II Peg. This star

is the primary component in the very active and well-studied RS CVn binary. The observations used here were presented in Rosén et al. (2015) but we give a brief summary for completeness. The observations consist of a total of 12 rotational phases observed between 2013 June 15 and 2013 July 1. The observations were performed with the CFHT using the ESPaDOnS spectropolarimeter. Although the spectra cover the region 3700–10500 Å and the full Stokes vector is available, we only focus on the three spectral regions discussed in Sect. 3 and on Stokes I . The spectral resolution is $R = 65,000$. The spectra were reduced with the standard Libre-ESPRIT package (Donati et al. 1997). The resulting S/N per velocity bin is of the order of 1000. An inclination angle of $i = 60^\circ$ and a projected rotation velocity of $v \sin i = 23 \text{ km s}^{-1}$ are assumed for II Peg (Rosén et al. 2015). We follow Rosén et al. (2015) and add a macroturbulent velocity of 4 km s^{-1} to the spectra to match the stellar parameters of II Peg. However, given the significant rotation velocity of II Peg, we model it by simply reducing the spectral resolution from $R = 65,000$ to $R = 49,120$.

Posterior samples are obtained with the model and we show the summary of the results in Fig. 9. We sample the temperature maps, again using the HEALPix pixelation with $N_{\text{side}} = 16$. We show results when the 12 rotation phases are used for reconstruction. In addition, and to check for the sensitivity to the number of observations, we also show the results obtained when one in every two observations is used (resulting in only six rotation phases) and one in every four (for a total of three rotation phases). When using the full observed data, a spot very close to the equator is found in all percentiles. The median temperature difference between the spot and the surroundings is $\sim 1000 \text{ K}$. However, there is large uncertainty on its specific temperature, between ~ 3600 and $\sim 4400 \text{ K}$ with 80% probability. A second less dark spot is also found close to the northern pole, and is most visible in phases 0.4–0.6. Its median temperature is just a few hundred Kelvin lower than the surroundings, although temperatures as low as $\sim 3800 \text{ K}$ cannot be discarded.

The comparison of our results with those found by Rosén et al. (2015) including all Stokes parameters shows a similar picture of the star. The polar spot is found in both approaches, although the Bayesian analysis places the large spot much closer to the equator than Rosén et al. (2015). In any case, this is compatible with the relatively large uncertainty in the exact location of the low-latitude spot that emerges from the percentiles 10 and 90. We also find clues of hot spots at intermediate-large latitudes, but their statistical relevance is small. The same applies to hot spots close to the equator, which are only found in percentile 90.

A very similar picture is found when half of the observed rotation phases are used. This is, in fact, a consequence of the almost equispaced sampling in rotation phase, as shown in the left panel of Fig. 10. Both dark spots are placed on exactly the same latitude and longitude. The situation is not so positive when only three rotation phases are used as observations. The map of the median temperature shows a spot that is displaced in longitude with respect to the previous inferences. However, the percentile 10 indicates the possible presence of the same equatorial spot found when the full observed material is used. The fact that it does not show up in the median or percentile 90 maps indicates that this prediction is affected by significant bias.

The right panel of Fig. 9 displays samples from the posterior distribution for the three considered cases. The samples fulfill the summary statistics of Fig. 9, but this helps give an idea of the type of variability and spatial correlation that we expect to

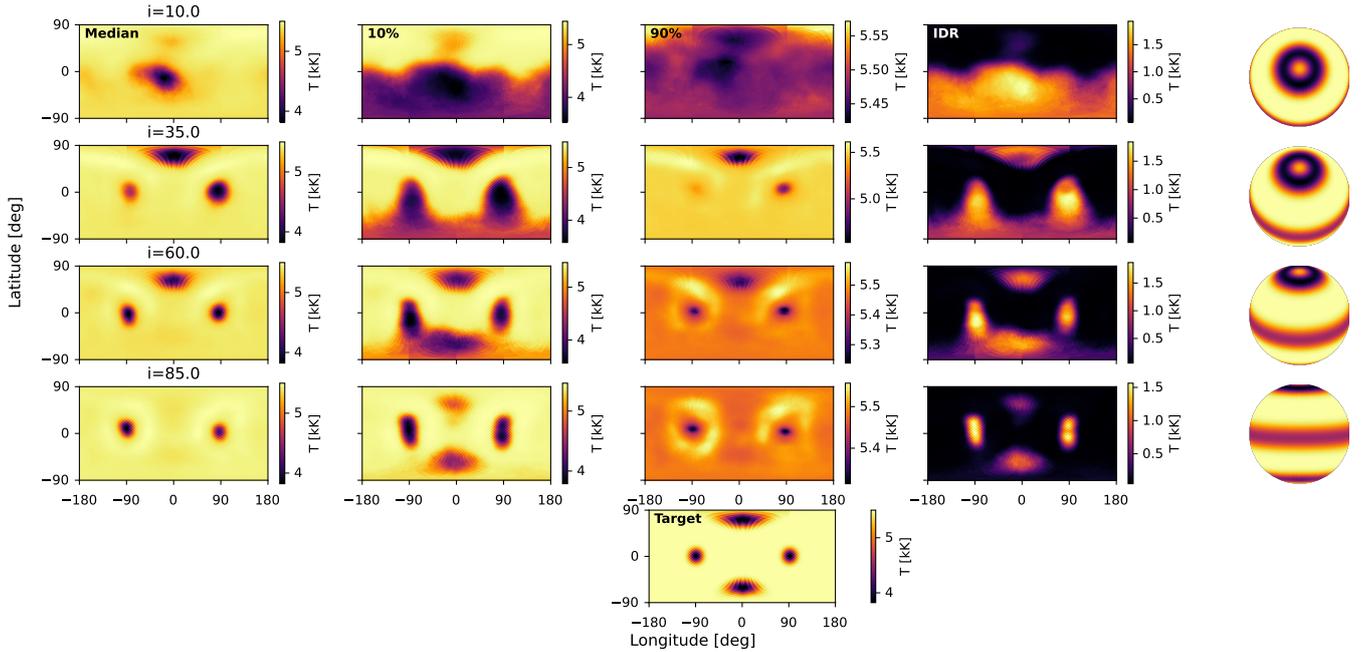


Fig. 8. Reconstruction of a synthetic case for different inclinations of the rotation angle. The star rotates at 30 km s^{-1} and is observed at 15 equidistant rotation phases. *First column:* median of the marginal posterior distribution per pixel. *Second, third, and fourth columns:* percentiles 10 and 90 of the same distribution, and IDR, respectively. *Fifth column:* pictorial representation of the observability of the active regions for each inclination angle. *Lower inset:* original temperature distribution in the star.

have in the inference. The equatorial spots appear in almost all the samples when all available observations are used. The polar spot also appears in many of the samples. However, the rest of the star is plagued with dark/bright spots that differ from one sample to the next and behave almost like white noise. Only those regions with strong posterior correlations produce structures in the percentile maps.

Finally, we carry out a posterior predictive check in Fig. 10, where we show the spectral lines resynthesized in the models from the posterior sampling and compare them with the observations. We only show the results for 12 and 6 observations. The observations are shown in blue. The spectra synthesized in the samples from the posterior are displayed in orange with transparency in order to clearly show that the largest density of profiles fit the observations. The green line displays the synthetic profiles obtained from the median estimation of the surface temperature, while the red line shows the median of all orange profiles. As radiative transfer is a nonlinear process, the two profiles are not the same, despite them being very similar. It is interesting to note that the large majority of the features are correctly reproduced by the inferred models, although some of the smaller scale features are not correctly captured by our approximate posterior, resulting in a somewhat poor quality fit compared to the results presented in Rosén et al. (2015).

7. Conclusions and future work

We have developed, validated, and applied a machine learning model that performs, for the first time, fully Bayesian inference of the surface temperature in rotating stars. The model is based on a normalizing flow that allows us to approximate the posterior distribution in this large-scale computationally heavy problem. The output of the normalizing flow is conditioned on a context vector extracted from the observations by a very flexible

Transformer encoder. Likewise, surface temperature maps are compressed with the aid of a pre-trained autoencoder.

Although training the model is slightly computationally costly, this computation is done only once for a given configuration of observed spectral lines and instrument. Its amortized character allows us to carry out inference for new observations in fractions of a second. The resulting posterior distribution for the temperature on the surface of the star can be exploited to capture uncertainties and correlations. We have used the median of the marginal distribution per pixel as an estimation of one potential solution, together with percentiles 10 and 90 as a summary of the uncertainty. Despite these summaries, all samples returned by the model are to be seen as potential solutions. Regions with small variability in the posterior (small difference between the percentiles 10 and 90) constitute well-constrained values of the surface temperature. Inter-pixel correlations are difficult to represent in surface data, and so we simply show samples from the posterior to give a limited view of which structures are spatially correlated. We find that our approximation overestimates the width of the posterior distribution when dealing with spectra with high S/N. We plan to reduce this effect by increasing the size of the training set.

There are obvious extensions to this work that we are planning to develop in the near future. Although requiring minor changes, having a full Bayesian solution could have a very sizeable impact. The first one is the extension to the magnetic case. The case of the Zeeman DI often requires the use of multi-line techniques because the polarimetric signal is well below the noise amplitude for individual lines. The most widespread multi-line technique is least-squares deconvolution (LSD; Donati et al. 1997; Kochukhov et al. 2010), which produces a pseudo-line with an increased S/N. Our model can be applied seamlessly to this case if the training is done with the LSD profiles of synthetic spectra, in the line with the work of Kochukhov et al. (2014).

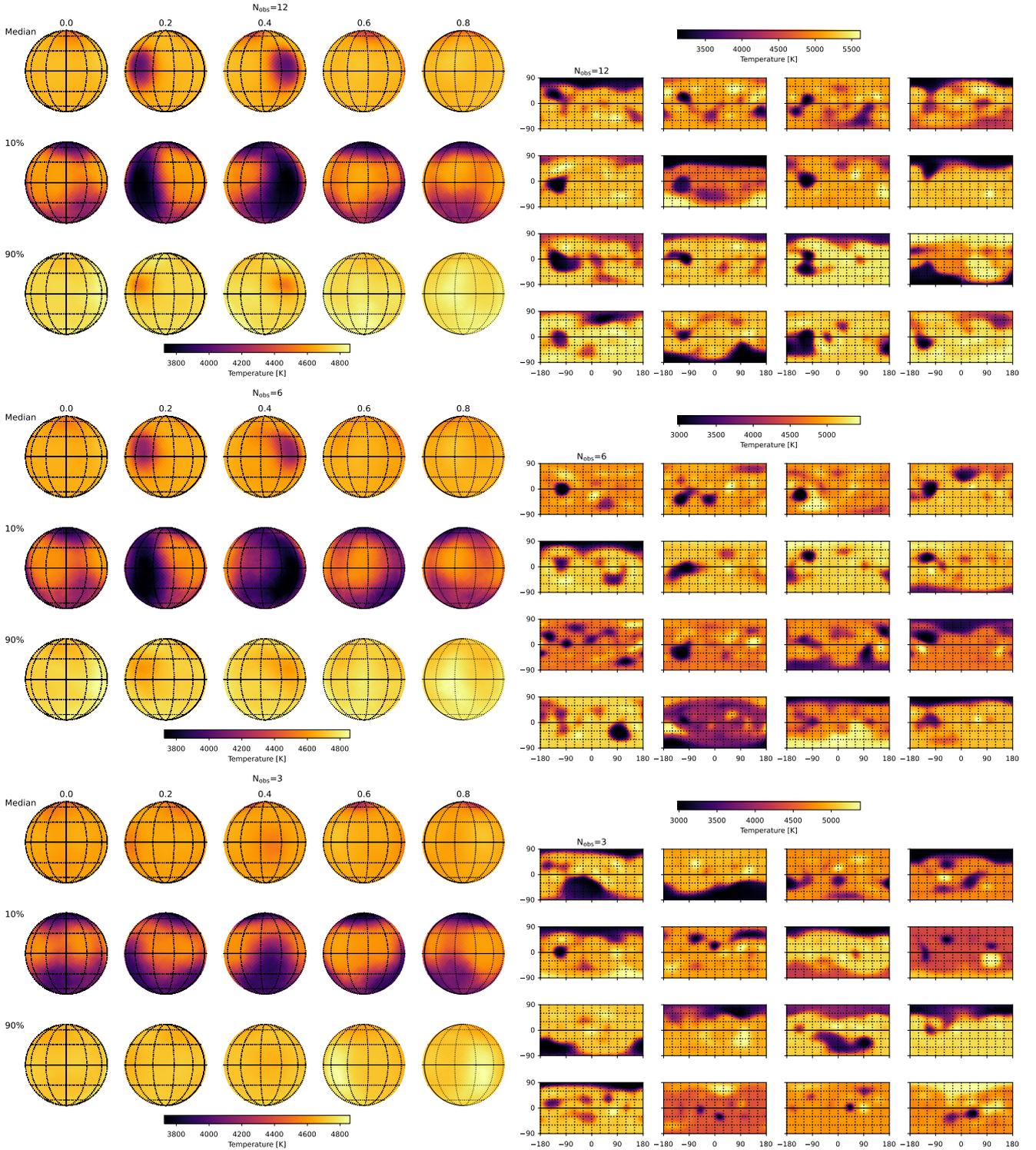


Fig. 9. Bayesian inference for the surface of II Peg. *Left panels:* median and percentiles 10 and 90 of the posterior distribution for the temperature surface map. The star is shown from five points of view in order to obtain access to the inference in all points of the surface. *Upper panel:* results when all available observations are used. *Middle panel:* what happens when only half of the observations are used. *Lower panel:* results when only three observations are used. *Right panels:* 16 posterior samples for each case.

Instead of one single surface map, the Zeeman DI requires the inclusion of the magnetic field vector per pixel in the star. As our machine learning model only requires simulations, the simulated stars can be made as complex as needed or one can even force some constraints to be fulfilled. For instance, one can force the magnetic field to have zero divergence everywhere.

If the autoencoder is properly trained, all solutions will maintain this property. Another possible extension is to deal with chemical abundance spots such as those observed on early-type stars. Again, given that stars are synthesized upfront, constraints can be imposed to make them as realistic as possible, such as constraints based on atomic diffusion theory.

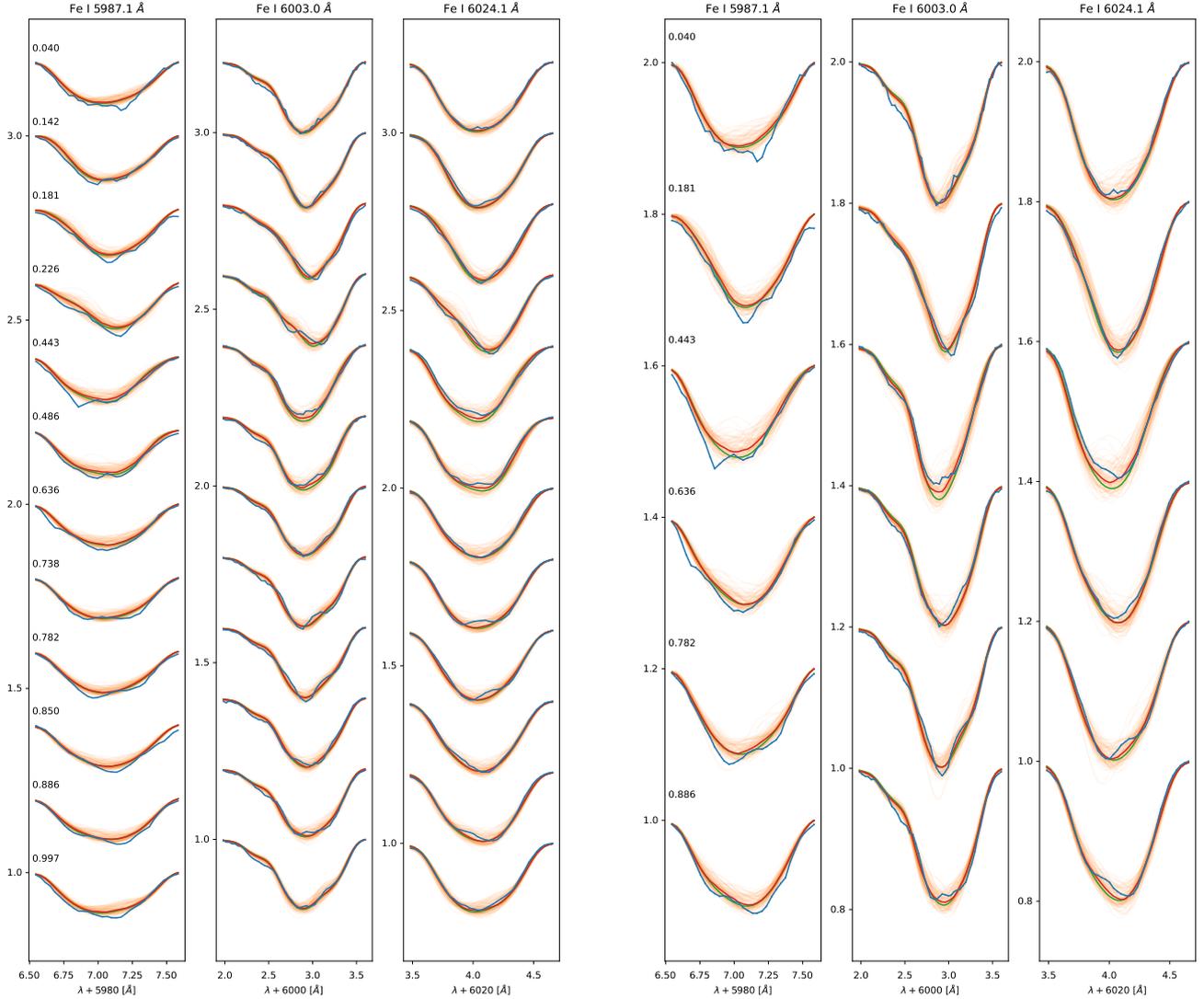


Fig. 10. Spectra of II Peg at all phases. Orange lines: synthesis carried out with samples from the posterior. Blue lines: original observations. Green line: synthesis using the temperatures of the median estimation of the posterior. Red line: median of all spectral lines synthesized. *Left panel:* results for all phases, while *right panel:* results when half the observations are used. We show the three spectral regions used for the inference, marking the specific rotation phases with labels. The lines of each rotation phase are displaced by 0.2 units to avoid crowding.

Acknowledgements. We acknowledge financial support from the Spanish Ministerio de Ciencia, Innovación y Universidades through project PGC2018-102108-B-I00 and FEDER funds. We also acknowledge support by the Swedish Research Council, the Royal Swedish Academy of Sciences and the Swedish National Space Agency. This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (SUNMAG, grant agreement 759548). The Institute for Solar Physics is supported by a grant for research infrastructures of national importance from the Swedish Research Council (registration number 2017-00625). This research has made use of NASA’s Astrophysics Data System Bibliographic Services. We acknowledge the community effort devoted to the development of the following open-source packages that were used in this work: `numpy` (numpy.org, Harris et al. 2020), `matplotlib` (matplotlib.org, Hunter 2007), `PyTorch` (pytorch.org, Paszke et al. 2019b) and `zarr` (github.com/zarr-developers/zarr-python).

References

- Adelman, S. J., Gulliver, A. F., Kochukhov, O. P., & Ryabchikova, T. A. 2002, *ApJ*, **575**, 449
- Bayes, T. 1764, *Phil. Trans. R. Soc. London*, **53**, 370
- Beaumont, M. A., Zhang, W., & Balding, D. J. 2002, *Genetics*, **162**, 2025
- Berdugina, S. V. 1998, *A&A*, **338**, 97
- Brown, S. F., Donati, J. F., Rees, D. E., & Semel, M. 1991, *A&A*, **250**, 463
- Chan, E. R., Monteiro, M., Kellnoher, P., Wu, J., & Wetzstein, G. 2020, ArXiv e-prints [arXiv:2012.00926]
- Collier Cameron, A., & Horne, K. D. 1986, *Maximum Entropy Reconstruction of Starspot Distributions*, eds. M. Zeilik, & D. M. Gibson (Berlin: Springer), 254, 205
- Cranmer, K., Brehmer, J., & Louppe, G. 2020, *Proc. Natl. Acad. Sci.*, **117**, 30055
- Crossfield, I. J. M., Biller, B., Schlieder, J. E., et al. 2014, *Nature*, **505**, 654
- Deutsch, A. J. 1958, *IAU Symp.*, **6**, 209
- Díaz Baso, C. J., Asensio Ramos, A., & de la Cruz Rodríguez, J. 2022, *A&A*, in press, <https://doi.org/10.1051/0004-6361/202142018>
- Dinh, L., Krueger, D., & Bengio, Y. 2014, ArXiv e-prints [arXiv:1410.8516]
- Donati, J. F. 2003, *ASP Conf. Ser.*, **307**, 41
- Donati, J. F., Semel, M., Carter, B. D., Rees, D. E., & Collier Cameron A. 1997, *MNRAS*, **291**, 658
- Durkan, C., Bekasov, A., Murray, I., & Papamakarios, G. 2019, ArXiv e-prints [arXiv:1906.04032]
- Durkan, C., Bekasov, A., Murray, I., & Papamakarios, G. 2020, *nflows: normalizing flows in PyTorch*
- Folsom, C. P., Bouvier, J., Petit, P., et al. 2018, *MNRAS*, **474**, 4956
- Goncharshkij, A. V., Stepanov, V. V., Kokhlova, V. L., & Yagola, A. G. 1977, *Sov. Astron. Lett.*, **3**, 147
- Goncharshkij, A. V., Stepanov, V. V., Khokhlova, V. L., & Yagola, A. G. 1982, *AZh*, **59**, 1146
- Gregory, P. C. 2005, *Bayesian Logical Data Analysis for the Physical Sciences* (Cambridge: Cambridge University Press)

- Gustafsson, B., Edvardsson, B., Eriksson, K., et al. 2008, *A&A*, **486**, 951
- Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, *Nature*, **585**, 357
- He, K., Zhang, X., Ren, S., & Sun, J. 2015, ArXiv e-prints [arXiv:1512.03385]
- Hunter, J. D. 2007, *Comput. Sci. Eng.*, **9**, 90
- Hussain, G. A. J., Donati, J.-F., Collier Cameron, A., & Barnes, J. R. 2000, *MNRAS*, **318**, 961
- Khokhlova, V. L. 1976, *Sov. Astron.*, **19**, 576
- Kingma, D. P., & Ba, J. 2014, ArXiv e-prints [arXiv:1412.6980]
- Kingma, D. P., & Dhariwal, P. 2018, in Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, 10236–10245
- Kobyzev, I., Prince, S., & Brubaker, M. 2020, *IEEE Trans. Pattern Anal. Mach. Intell.*, **1**
- Kochukhov, O. 2004, *A&A*, **423**, 613
- Kochukhov, O. 2016, *Lect. Notes Phys.* (Berlin: Springer), 914, 177
- Kochukhov, O., Makaganiuk, V., & Piskunov, N. 2010, *A&A*, **524**, A5
- Kochukhov, O., Lüftinger, T., Neiner, C., Alecian, E., & MiMeS Collaboration 2014, *A&A*, **565**, A83
- Kochukhov, O., Shultz, M., & Neiner, C. 2019, *A&A*, **621**, A47
- Krachmalnicoff, N., & Tomasi, M. 2019, *A&A*, **628**, A129
- Lei Ba, J., Kiros, J. R., & Hinton, G. E. 2016, ArXiv e-prints [arXiv:1607.06450]
- Nair, V., & Hinton, G. E. 2010, in Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel, 807
- Paszke, A., Gross, S., Massa, F., et al. 2019a, ArXiv e-prints, [arXiv:1912.01703]
- Paszke, A., Gross, S., Massa, F., et al. 2019b, in *Advances in Neural Information Processing Systems 32*, ed. H. Wallach, H. Larochelle, A. Beygelzimer, F. d'É. Buc, E. Fox, & R. Garnett (Curran Associates, Inc.), 8024
- Pearson, K. 1901, *Philos. Mag.*, **2**, 559
- Perez, E., Strub, F., de Vries, H., Dumoulin, V., & Courville, A. 2017, ArXiv e-prints [arXiv:1709.07871]
- Piskunov, N. E., & Khokhlova, V. L. 1983, *Sov. Astron. Lett.*, **9**, 346
- Piskunov, N., & Kochukhov, O. 2002, *A&A*, **381**, 736
- Piskunov, N. E., Tuominen, I., & Vilhu, O. 1990, *A&A*, **230**, 363
- Planck Collaboration I. 2020, *A&A*, **641**, A1
- Rice, J. B. 2002, *Astron. Nachr.*, **323**, 220
- Rice, J. B., & Strassmeier, K. G. 2000, *A&AS*, **147**, 151
- Rosén, L., Kochukhov, O., & Wade, G. A. 2015, *ApJ*, **805**, 169
- Rubin, D. B. 1984, *Annals Stat.*, **12**, 1151
- Semel, M. 1989, *A&A*, **225**, 456
- Sitzmann, V., Martel, J. N., Bergman, A. W., Lindell, D. B., & Wetzstein, G. 2020, ArXiv e-prints [arXiv:2006.09661]
- Skilling, J., & Bryan, R. K. 1984, *MNRAS*, **211**, 111
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014, *J. Mach. Learn. Res.*, **15**, 1929
- Strassmeier, K. G., Carroll, T. A., & Ilyin, I. V. 2019, *A&A*, **625**, A27
- Tikhonov, A. N., & Arsenin, V. Y. 1977, *Solution of Ill-posed Problems* (Wiley: New York)
- Vaswani, A., Shazeer, N., Parmar, N., et al. 2017, in Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, 6000–6010
- Vogt, S. S., Penrod, G. D., & Hatzes, A. P. 1987, *ApJ*, **321**, 496