Physics-guided interpretable probabilistic representation learning for high resolution image time series

Yoël Zérah, Silvia Valero, Jordi Inglada

▶ **To cite this version:**

Yoël Zérah, Silvia Valero, Jordi Inglada. Physics-guided interpretable probabilistic representation learning for high resolution image time series. 2022. hal-03837736v2

HAL Id: hal-03837736

https://hal.science/hal-03837736v2

Preprint submitted on 4 Nov 2022 (v2), last revised 16 Aug 2023 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Physics-guided interpretable probabilistic representation learning for high resolution image time series

Yoël Zérah, Silvia Valero, Jordi Inglada

***Abstract*—Learning representations that capture meaningful underlying information of data is a promising solution to reduce the reliance on labeled data for downstream applications. With the advent of the big remote sensing data era, self-supervised deep learning methods have become a valuable tool to extract high-level, complex abstractions and representations from large volumes of data. However, classical methodologies such as Variational Autoencoders are focused on imposing statistical constraints on the latent space and they do not learn generic and interpretable representations of the data.**

To address such limitation, this work presents a generic physics-guided representation learning methodology to discover semantic representations. To address it,the proposed approach constrains the learning process with the incorporation of prior physical knowledge. This study shows through an example how the methodology can be used to solve remote sensing inverse problems. Specifically, the inversion of a crop phenology model derived from NDVI time series is proposed. As a result, the probability distributions of the intrinsic physical model parameters are inferred. The feasibility of the method is evaluated on both simulated and real Sentinel-2 data and compared with different standard algorithms.

***Index Terms*—Generative Models, Autoencoders, Satellite Image Time Series, Self-Supervised Representation Learning, Bayesian physics-guided learning, Inverse problems, Phenology Monitoring, Large Scale.**

I. Introduction

NOWADAYS, vast amount of data are acquired by satellite-borne sensors for Earth Observation. In the last decade, the Sentinel-2 (S2) satellites of EU's Copernicus program have been acquiring optical Satellite Image Time Series (SITS), with high spatial, spectral and temporal resolutions. These data enable large scale applications of Earth monitoring with stunning precision, such as urban sprawl, agricultural cycles, fast disaster analysis and response, climate change impact evaluation, etc. [1], [2].

While remote sensing provides very rich data about Earth's surface state, actual extraction of useful information is not straightforward. Satellite data is complex, with wide variability and information redundancy among its several dimensions. Besides, cloud coverage and sensor overpass (e.g. S2 swath intersections) lead to an irregular temporal and spatial sampling. Traditional statistical methods typically struggle to cope with the huge dimensionality of satellite data. Therefore, there is a need to produce reduced representations that capture the important aspects of the data.

Deep representation learning [3] aims at transforming data into more useful representations and is especially suited to large data volumes. Such a representation should identify the most informative components of input data for downstream applications. Because producing annotated data-sets for satellite data is costly and difficult, self-supervised representation learning is promising.

Generative models are well suited to unsupervised representation learning, in that they learn to generate data from input parameters. These parameters constitute a *de facto* generative representation of data and capture enough information to be able to synthesize data. Representation learning performs then the inverse task by finding the representation that matches data with regards to a given generative model.

Because models are never perfect (*epistemic uncertainty*) and data may be subject to noise (*aleatoric uncertainty*), representations derived from generative models should be described with uncertainties. In generative models, Bayesian inference theory enables inferring probabilistic latent representations capturing these uncertainties. Nonetheless, separating aleatoric and epistemic uncertainties may be intractable: only a *predictive uncertainty* that combines both can be usually quantified [4].

Uncertainty quantification in representations can be performed with Variational Autoencoders (VAE) [5], that are a type of generative model that combines Bayesian theory with deep learning [6], [7]. VAE are successfully used to capture generative factors of data as probabilistic latent variables, yet they face several challenges for producing meaningful representations. They cannot embed a too powerful generative model, for they tend in such cases to reconstruct data while ignoring the latent representation [8]. Furthermore, learned representations aren't interpretable, in the sense that they don't match with human-defined concepts [9]. Interpretable representations would ensure that downstream applications are explainable and justifiable.

Because remote sensing deals with physical data, such as reflectances for Sentinel-2 optical images, there is a large quantity of scientific knowledge about those measurements and the physical processes that they are tied to. There are priors

This work is supported by the Natural Intelligence Toulouse Institute (ANITI) from Université Fédérale Toulouse Midi-Pyrenées under grant agreement ANITI ANR-19-P3IA-0004 (this work is part of a PhD that is co-founded by ANITI and by the Centre National d'Etudes Spatiales (CNES)). This research is also involved in the ANR-JCJC DeepChange project.

Y. Zérah, S. Valero, J. Inglada are with CESBIO, Université de Toulouse, CNES/CNRS/INRAe/IRD/UPS, 31000 Toulouse, France (e-mail: yoel.zerah@univ-toulouse.fr, silvia.valero@cesbio.cnes.fr, jordi.inglada@cesbio.cnes.fr).

and models for physical data derived from human expertise, geographical and natural sciences. Integrating prior knowledge into representation learning has been of great interest recently, with *gray-box* approaches [10].

In this paper, we aim at developing novel approaches to integrate physical knowledge to learn interpretable representations. For instance, variables involved in physical processes, such as moisture, temperature and solar irradiance for vegetation growth, phenological phases dates in vegetation monitoring, etc., are interpretable representations of physical data. More generally, when latent variables are the parameters of physical, or user-defined models used to guide training, they are directly interpretable. To learn such representations, we semantically bind VAE latent variables to user-defined physical model parameters.

The remainder of this article is organized as follows. Section II introduces current approaches for learning interpretable representations of data with VAE frameworks. Section III presents how physical priors can be incorporated into the learning process in different ways to make representations interpretable. Section IV defines a temporal model for satellite optical time series for phenology monitoring, that is integrated in the proposed *pheno-VAE*. Two time series data-sets are also described. At last, the benefits of using the proposed methodology for the inversion of the previous presented model are presented in Section V, where experimental results are shown.

II. Related works

A. Representation learning with Variational Autoencoders

Autoencoders (AE) are self-supervised neural networks that learn low dimensional representations from unlabeled data. An encoder reduces the dimension of the input data into deterministic latent variables that are used by the decoder to reconstruct the input data. Both the encoder and the decoder are neural networks that are trained simultaneously to optimize the compression of the input data. The loss is usually a mean squared error (MSE) of the reconstruction. A Variational Autoencoder (VAE, see Fig. 1) embeds the representation in the latent space, as random variables [6]. Specifically, the encoder outputs the parameters $\boldsymbol{\lambda}$ of a so-called *variational* distribution. The variational distribution belongs to a parametric distribution family, usually a Gaussian distribution. The VAE latent space being a distribution, it fosters regular and continuous representations, making it more suitable to represent high level features. Then the realizations $\mathbf{z}$ of this distribution are taken as input to the decoder. The decoder's output are also distribution parameters, from which reconstructions $\hat{\mathbf{x}}$ are sampled. This is sometimes called *ancestral sampling* [11].

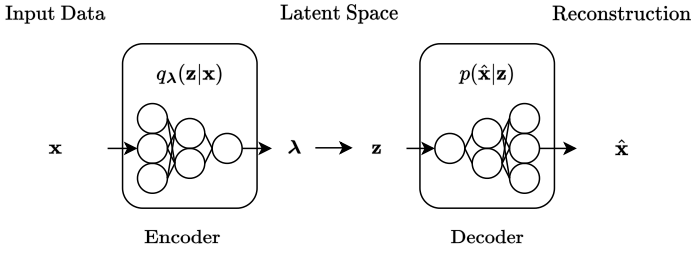


Fig. 1. Classical Variational Autoencoder

A well-known problem of VAEs is the over-pruning of latent variables [12], [13]. When the decoder is a too powerful generative model, it is able to reconstruct the input while ignoring some latent variables. In this case, latent variables can become redundant, and their distributions collapse. Another challenge of representation learning with VAE is the model identification issue related to the classical standard Gaussian prior. Despite the existence of many distribution families, most works only consider Gaussian latent spaces. This can limit the ability of VAE to infer meaningful representations. Finally, as above-mentionned, the latent distributions learned by traditional VAE architectures aren't easily interpretable because associated latent variables are the generative factors of an unknown generative model.

B. Disentanglement

There is a significant amount of research that attempts to improve the interpretability and meaningfulness of latent representations learned by VAE, with *disentanglement* [14]. Disentanglement is a meta-prior about data, that assumes the existence of independent *factors of variation* that generate the data [3], [15]. A disentangled representation should capture these factors into different independent variables. Disentanglement introduces some form of separation so that each component of the representation may encode uncorrelated features at various abstraction levels.

To learn disentangled representations, most approaches rely on a regularization of the latent space to enforce the same properties (closeness to the prior, independence between latent variables, etc.) [16]. Traditionally, a supplementary regularizing term is added to the Evidence Lower BOund (ELBO) objective function to penalize the variational distribution. For instance, β-VAE [17] adds a tunable hyper-parameter to the ELBO loss to control the balance between the reconstruction ability and the regularization of the latent space. Factor-VAE [18] enforces latent variable independence by encouraging the aggregated latent distribution $q_{\boldsymbol{\lambda}}(\mathbf{z})$ to be factorial, by penalizing the total correlations $q_{\boldsymbol{\lambda}}(\mathbf{z})$. In [19], disentanglement is promoted by matching the covariance of the joint latent distribution to that of the prior. While many advances have been made to learn disentangled representations, it is difficult to compare and evaluate results obtained by different representation learning methodologies. The main problem is that, although several metrics exist, it is still unclear how to quantify disentanglement [16].

Furthermore, it is shown theoretically in [20] that there are no disentangled representations without inductive priors. It is underlined that increased disentanglement does not seem to provide better performances on downstream tasks. Besides, existing disentanglement approaches focus on separating independent factors of variations despite the fact that real-world observations are often not structured into meaningful independent causal variables to begin with [21].

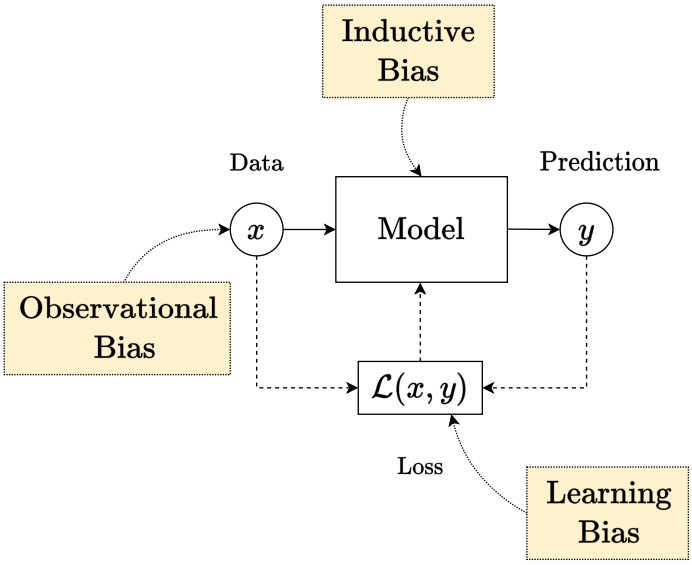


Fig. 2. Incorporation of physical priors in machine learning models.

In particular, there is often significant correlation between factors of variations in physical data. Semantics of physics-based latent variables should correspond to some properties of the observations, that are not necessarily independent. Therefore, disentanglement may not be the best approach to find interpretable representations of such data. Interpretability (the correspondence to human-defined concepts) and explainability (the ability of predictions and decisions to be understood by humans) are mostly secondary in disentanglement studies, behind informativeness and independence criteria [9], [22].

C. Physics-based generative models

Imposing statistical structures with disentanglement latent representations in VAE doesn't ensure which particular properties are uncovered in representations. Abundant literature and models are available to describe and explain the observations of various physical systems. Instead of seeking disentanglement, it can be preferable to enforce specific priors by using scientific knowledge about the observed processes or about the physics of the sensor, .

Integrating physical priors into the learning process has recently been the motivation of works proposing *hybrid methods*.

Theoretically, three types of priors can be introduced to guide learning toward physically consistent predictions [23] (see Fig. 2): "observational biases", "inductive biases" and "learning biases". Observational biases are brought through the choice of data that capture the physical properties of interest. Inductive biases are incorporated by the tailoring of models so that predictions are guaranteed to follow specified physical behaviors. Learning biases are enforced through the choice of loss functions. Several recent methods based on generative models integrate physics with learning biases, and induction biases by specifically tweaking the generative processes (e.g. the decoders in AE) [24]–[26]. For instance, in [27], observational knowledge of galaxy images (point spread function and noise level of images), is used in the decoding process and in the loss function of a VAE. This method can also be used to denoise images, because physical-based latent parameters generate noiseless reconstructions from input images.

Sometimes, knowledge about the physics of the data is available in the form of a parametric physical model, whose parameters are physical variables. In AE's framework, this is performed by replacing the neural network decoder by a user-defined decoder (see Fig. 3), with the code/latent variables being semantically tied to its parameters [28]. In this setup, latent representations become interpretable because they are the parameters of a known generative model.

In [29], a model of elliptic galaxy images replaces the classical AE's decoder. In [30], different methodologies present the replacement of the decoder by user-defined models $\mathcal{F}$, to learn probabilistic representations with VAE. It is proposed to infer representations that are partially interpretable. Such a latent space has an interpretable part bound to a user-defined decoder, and a non interpretable part bound to a neural network decoder.

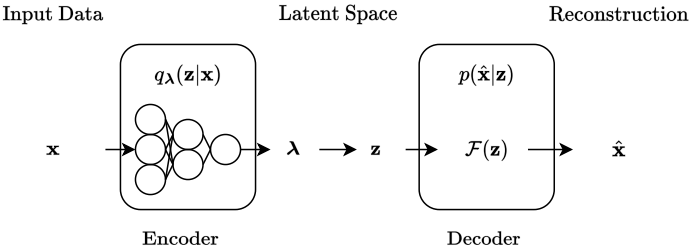


Fig. 3. VAE with user-defined decoder

When replacing the decoder with a user-defined function, all latent variables are forced to be taken into account for the reconstruction, as long as they have impact in the function output. Therefore, it effectively avoids the over-pruning of latent variables. Another interesting point of this setup is that this strategy solves the inverse problem associated with the user-defined model by only training the encoder.

Here, we propose to perform the integration of a user-defined function in a VAE decoder, beyond what is depicted in [30]. We further incorporate inductive and learning biases for representation learning of physical data. We also propose a novel training procedure based on the sampling of the latent distribution.

III. Methodology

The theoretical basis of VAE is firstly introduced through the perspective of variational inference (VI) before presenting the proposed physics-guided methodology. Secondly, different methodological contributions are presented to incorporate physical priors through inductive and learning biases: (i) a new Monte-Carlo reconstruction loss strategy for the incorporation of physical models in VAE decoders, (ii) the possibility and benefit of using variational distributions other than Gaussian to better model physical quantities, (iii) the incorporation of physical priors by imposing complementary relationship constraints on latent distributions.

A. Amortized variational inference with VAE

Let there be a probabilistic model that has observations $\mathbf{x}$ and latent variables $\mathbf{z}$, with its joint density:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z}) \tag{1}$$

with $p(\mathbf{z})$ the *prior* over the latent distribution and $p(\mathbf{x}|\mathbf{z})$ the *likelihood*. It should be noted that $p(\mathbf{x}, \mathbf{z})$ is a *generative model* of observations from latent variables, and $\mathbf{z}$ is then a *generative factor*, and a representation of $\mathbf{x}$. Computing the posterior $p(\mathbf{z}|\mathbf{x})$ is known as the inference problem. Although Bayes theorem,

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{\int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}}, \tag{2}$$

defines a rigorous mathematical formulation for any inference problem, it is not directly applicable. This is because $\int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}$ can become intractable due to the large dimensionality of $\mathbf{z}$. To overcome this issue, instead of calculating the exact posterior, approximation methods are commonly used. In particular, variational inference methods approximate the posterior with a so-called *variational distribution* $q_{\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x})$, that is restricted to belong to a $\boldsymbol{\lambda}$-parameterized distribution family $\mathcal{Q}_{\boldsymbol{\lambda}}$.

To ensure that $q_{\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x})$ is the best approximation of the posterior among $\mathcal{Q}_{\boldsymbol{\lambda}}$, inference methods minimize the Kullback-Leibler (KL) divergence between the posterior and its approximation:

$$q^*_{\boldsymbol{\lambda}}(z) = \underset{q_{\boldsymbol{\lambda}} \in \mathcal{Q}_{\boldsymbol{\lambda}}}{\arg\min}\ \mathbb{KL}\left(q_{\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}|\mathbf{x})\right). \tag{3}$$

The KL-divergence is also untractable here because of the evidence term, $\log p(\mathbf{x}) = \log \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}$.

$$\begin{aligned} \mathbb{KL}\left(q_{\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}|\mathbf{x})\right) = \mathbb{E}\left[\log q_{\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x})\right] \\ - \mathbb{E}\left[\log p(\mathbf{x}, \mathbf{z})\right] \\ + \log p(\mathbf{x}) \end{aligned} \tag{4}$$

The optimization problem can be solved by using the ELBO denoted in (5), by considering a prior distribution $p(\mathbf{z})$ over the variational distribution.

$$\begin{aligned} \mathrm{ELBO}\left(q_{\boldsymbol{\lambda}}\right) &= -\mathbb{KL}\left(q_{\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}|\mathbf{x})\right) + \log p(\mathbf{x}) \\ &= \mathbb{E}\left[\log p(\mathbf{x}, \mathbf{z})\right] - \mathbb{E}\left[\log q_{\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x})\right] \\ &= \mathbb{E}\left[\log p(\mathbf{x}|\mathbf{z})\right] - \mathbb{KL}\left(q_{\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})\right) \end{aligned} \tag{5}$$

Because the evidence is constant with respect to $\boldsymbol{\lambda}$, maximizing the ELBO leads to minimizing the KL divergence term in (3).

In VAE, the likelihood $p(\mathbf{x}|\mathbf{z}, \theta)$ is embedded in the decoder, and the posterior distribution $q_{\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x}, \phi)$ in the encoder, with θ and ϕ the respective networks' parameters. The encoder infers the variational parameters $\boldsymbol{\lambda}$. The variational distribution is typically chosen to be Gaussian: $q_{\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mathbf{z}|\boldsymbol{\lambda})$ with $\boldsymbol{\lambda} = [\boldsymbol{\mu}_{\mathbf{z}}(\mathbf{x}, \phi), \boldsymbol{\Sigma}_{\mathbf{z}}(\mathbf{x}, \phi)]$, with $\boldsymbol{\mu}_{\mathbf{z}}$ and $\boldsymbol{\Sigma}_{\mathbf{z}}$ being the mean vector and the covariance matrix of the latent variables. This choice enables the explicit computation of the KL loss term with a standard Gaussian prior, and a differentiable sampling strategy[1] using the reparameterization trick (see (6)). In practice, $\boldsymbol{\Sigma}_{\mathbf{z}}$ is assumed to be a diagonal matrix, because it prevents having to ensure definite positiveness and it reduces the number of inferred latent parameters.

$$\mathbf{z} = \boldsymbol{\mu}_{\mathbf{z}} + \boldsymbol{\Sigma}_{\mathbf{z}}^{1/2}\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \Rightarrow \quad \mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}}, \boldsymbol{\Sigma}_{\mathbf{z}}) \tag{6}$$

The decoder infers the parameters of the distribution of the reconstructions selected among a chosen parametric distribution family — although this aspect is often overlooked in the literature. In this work, the distribution of the decoder is chosen as Gaussian: $p(\hat{\mathbf{x}}|\mathbf{z}, \theta) = \mathcal{N}(\hat{\mathbf{x}}|\boldsymbol{\mu}_{\hat{\mathbf{x}}}(\mathbf{z}, \theta), \boldsymbol{\Sigma}_{\hat{\mathbf{x}}}(\mathbf{z}, \theta))$, with $\boldsymbol{\mu}_{\hat{\mathbf{x}}}$ and $\boldsymbol{\Sigma}_{\hat{\mathbf{x}}}$ the corresponding mean vector and covariance matrix. The covariance matrix $\boldsymbol{\Sigma}_{\hat{\mathbf{x}}}$ is commonly set as a hyper-parameter (often set to identity matrix). It can also be considered a trainable parameter or estimated from the input's distribution [31].

Traditionnally, the negative ELBO (5) is the loss function minimized during the VAE training process. It has two terms : $\mathcal{L} = \mathcal{L}_{rec} + \mathcal{L}_{KL}$. The reconstruction term $\mathcal{L}_{rec} = -\mathbb{E}\left[\log p(\mathbf{x}|\mathbf{z})\right]$ is the expectation of the Negative Log Likelihood (NLL). It forces decoded samples to match the initial input data. $\mathcal{L}_{rec}$ can be approximated as the average NLL over a number L of Monte-Carlo samples of the latent distribution (see (7)). However in practice with a batch size large enough, $\mathbf{z}$ is typically only sampled once per iteration [6].

$$\mathbb{E}\left[\log p(\mathbf{x}|\mathbf{z})\right] \approx \frac{1}{L}\sum_{i=1}^{L} \log p(\mathbf{x}|\mathbf{z}^{(i)}) \tag{7}$$

This term depends on the distribution chosen in the decoder, commonly a Bernoulli or a Gaussian distribution. Using a MSE loss instead would be equivalent to minimizing the NLL of a unit-variance Gaussian. However, this assumption of constant variance of reconstructions does not allow to accurately estimate the uncertainty of the predictions. Using MSE as a reconstruction loss typically results in VAE being over-regularized [32].

The second term of the ELBO $\mathcal{L}_{KL} = \mathbb{KL}\left(q_{\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})\right)$, is a regularization term that penalizes the mismatch of the variational distributions to the prior $p(\mathbf{z})$. This term has a closed form with Gaussian latent spaces and the usual prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

Because VAE use the ELBO to optimize neural network weights θ and ϕ instead of the variational parameters $\boldsymbol{\lambda}$, it is an *amortized* variational inference approach — VAE learn a function that maps $\mathbf{x}$ to variational distribution parameters $\boldsymbol{\lambda}$.

B. Monte Carlo reconstruction loss for deterministic decoders

The physics-guided learning methodology presented in this work proposes the use of physical-based decoders (see Fig. 3). The outputs of a deterministic user-defined decoder $\mathcal{F}$ can no longer be distribution parameters, but transformations of latent distribution samples. It implies that only samples from $p(\mathbf{x}|\mathbf{z})$ are available for the reconstruction loss computation. Therefore, we propose to approximate $p(\mathbf{x}|\mathbf{z})$ as a Gaussian distribution and to evaluate its parameters (see the mean $\boldsymbol{\mu}_{\hat{\mathbf{x}}}$ in (8) and the covariance matrix $\boldsymbol{\Sigma}_{\hat{\mathbf{x}}}$ in (9)) from reconstructions obtained by Monte-Carlo sampling the latent distribution and

[1] with respect to variational parameters

decoding the samples. The difference between a classical Gaussian decoder and our proposed approach is depicted in Fig. 4).

$$\boldsymbol{\mu}_{\hat{\mathbf{x}}}(\mathbf{z}) \approx \frac{1}{K}\sum_{i=1}^{K}\mathcal{F}(\mathbf{z}^{(i)}) \tag{8}$$

$$\boldsymbol{\Sigma}_{\hat{\mathbf{x}}}(\mathbf{z}) \approx \frac{1}{K-1}\sum_{i=1}^{K}\left(\mathcal{F}(\mathbf{z}^{(i)}) - \boldsymbol{\mu}_{\hat{\mathbf{x}}}\right)\left(\mathcal{F}(\mathbf{z}^{(i)}) - \boldsymbol{\mu}_{\hat{\mathbf{x}}}\right)^{\top} \tag{9}$$

The Monte-Carlo sampling of latent space brings up a new hyper-parameter K: the number of latent samples drawn from the latent distribution inferred from each input sample $\mathbf{x}$. In particular, the latent distribution sampling involved is used to approximate the NLL, and not the expectation of the NLL. The total number of latent samples drawn should be $L \times K$, but we still set $L = 1$. The choice of K is a trade-off between accuracy of $\boldsymbol{\mu}_{\hat{\mathbf{x}}}$ and $\boldsymbol{\Sigma}_{\hat{\mathbf{x}}}$, and training time, because latent distribution sample requires a forward pass through the decoder. Finally, the reconstruction loss term for a Gaussian decoder is described in (10).

$$\mathcal{L}_{rec}(\mathbf{x}) = \frac{1}{2}\left[\left(\mathbf{x}-\boldsymbol{\mu}_{\hat{\mathbf{x}}}\right)^{\top}\boldsymbol{\Sigma}_{\hat{\mathbf{x}}}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}_{\hat{\mathbf{x}}}\right) + \ln\left(|\boldsymbol{\Sigma}_{\hat{\mathbf{x}}}|\right)\right] \tag{10}$$

$\Sigma_{\hat{\mathbf{x}}}$ is approximated as a diagonal covariance matrix by assuming the independence of reconstruction components. Not assuming diagonality of covariance matrix could improve reconstruction quality, and add structure to residuals [33]. However, covariance matrix inversion and determinant computation would become prohibitively expensive for any large dimensional data.

The Gaussian NLL encourages both the reconstruction error of each sample to be small, and the reconstruction variance to model uncertainty, even if the distribution of reconstructions is not Gaussian . If the error isn't small, the variance can be increased to still minimize the loss (e.g. when the error cannot be minimized, uncertainty is increased). The $\ln\left(|\Sigma_{\hat{x}}|\right)$ term prevents the variance from arbitrarily increasing as a trivial way of minimizing the loss.

C. The variational distribution as an inductive bias

To learn latent semantic representations, the incorporation of inductive biases is essential. The use of a physical-based decoder implies that latent variables are tied to physical measurements. Therefore, knowledge about the probability distribution characterizing these measurements can be used to discard the classical prior and posterior Gaussian assumption. We advocate for choosing a variational distribution family that matches assumptions about each semantic latent variables, and a prior that accounts for knowledge about the data-set.

The choice of the variational distribution is limited to distributions that can be sampled in a differentiable way, so that gradients can be propagated through. Three different sampling techniques can be considered to enable various distribution choices [6]:

1) A reparameterization trick to sample *location-scale family* distributions [34], such as the usual Gaussian distribution (see (6)).

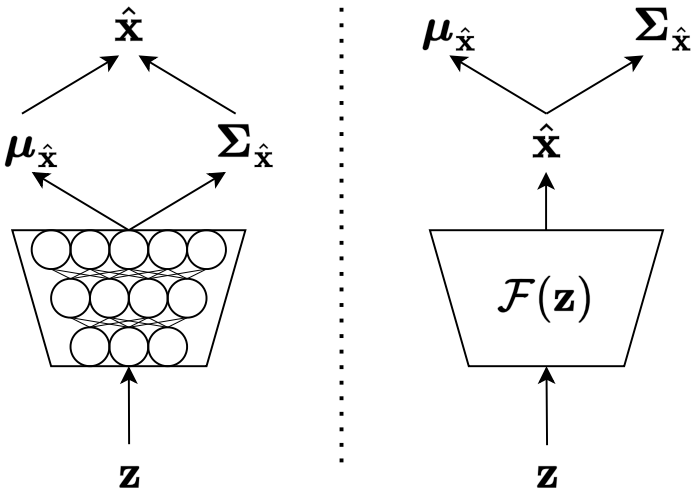


Fig. 4. Predicting parameters of a Gaussian decoder. Left: original Gaussian decoder, with both the mean and the covariance being output by a trained neural network. Right: a non-trainable decoder, where the mean and variance are estimated from a set of K reconstructed samples $\hat{\mathbf{x}}$ obtained by the Monte-Carlo sampling strategy.

2) Composition of auxiliary random variables. For instance, log-normal, logit-normal, Dirichlet, exponential distribution samples may be generated by transforming "elementary" distributions (respectively by composing Gaussian with logarithm, Gaussian with sigmoid, Gaussian with softmax [35] and uniform with logarithm).
3) The inverse transform sampling method described in (11), that can be used to sample any continuous random variable $z \sim \mathcal{A}$. This technique can be used provided its inverse cumulative distribution function (ICDF) $F_{\mathcal{A}}^{-1}$ is differentiable, by propagating a sample from a uniform distribution $\mathcal{U}(0,1)$.

$$z = F_{\mathcal{A}}^{-1}(u), \quad u \sim \mathcal{U}(0,1) \quad \Rightarrow \quad z \sim \mathcal{A} \tag{11}$$

In practice, the gradient of the ICDF computed during training may diverge. In intervals of zero density, the CDF is constant at $y = c$ and its reciprocal has infinite derivative at $x = c$. Therefore uniform sampling of u must be done inside an interval I where the CDF is strictly monotonous. In fact, due to the numerical precision ϵ, the interval I has to be restricted even further (see (12)).

$$I = F_{\mathcal{A}}(X), \quad X = \{x \in [0,1] \text{ s.t. } dF_{\mathcal{A}}(x) \geq \epsilon\} \tag{12}$$

As mentioned above, latent variables are semantically related to the prior knowledge of the physical model, therefore, the physical measurements, corresponding to the parameters may be bounded. If parameters to be estimated are known to belong to a certain bounded interval, their corresponding variational distributions should have closed-support. Physical models can be mathematically defined even with out-of-bounds parameters, but samples generated with these parameters would not be realistic. Such reconstructions could still minimize the reconstruction loss, and hamper training while the encoder learns to infer wrong latent representations. This

can be especially detrimental when some training samples are not well described by the physical model. The VAE would then bend the model instead of increasing latent uncertainty.

The above-described sampling techniques can be used to sample bounded distributions. This can be achieved by composing unbounded distribution samples, such as Gaussian samples drawn from the reparameterization trick, with sigmoid functions[2] (logistic[3], hyperbolic tangent, arc-tangent, etc...). However with this method, the resulting distributions are distorded, asymmetric near the support bounds and may even become bimodal. To avoid such limitations, the inverse transform method enables the sampling of closed support distributions such as raised cosine distributions, Kumaraswamy distributions, etc.

The bounds of the distributions can be inferred by the encoder, or set by the user. In both cases, it may be convenient to sample bounded distributions on the interval $[0, 1]$ and then perform affine scaling to the desired $[a, b]$ interval (see (13)).

$$z \in [0, 1] \Rightarrow (b - a)z + a \in [a, b] \tag{13}$$

While the variational distribution should be chosen with physical variables meaning in mind, it has to be paired with a prior distribution that enables computation of the KL loss term. It may unfortunately be more complicated to find a meaningful prior whose KL divergence with the variational distribution admits a closed-form expression.

D. Incorporation of order constraints into latent distributions

In all previous sampling methods, the independence of latent variables is assumed. However the physical variables of a model may not be independent, and correlations and statistical dependence are usually observed between variables. Different strategies are proposed here to introduce dependence between latent variables, while still performing independent sampling as is done with classical VAE. These strategies propose the ancestral sampling [11] of latent variables whose values are constrained by an order relation. For instance, physical models associated to satellite times series usually have input parameters associated to time, therefore order relationships can be established.When using such models for VAE decoders, order constraints must be enforced to prevent the training from converging to representations that are not physically plausible. As this is not done by sampling independent latent distributions, we propose here methods to ensure order of latent variables.

Let there be n latent variables z_i, on intervals $[a_i, b_i]$ that must be ordered as follows: $z_i < z_{i+1}, \quad \forall i \in [\![1, n-1]\!]$. Two complementary situations can arise:

(i) $\forall i,\ [a_i, b_i] \cap [a_{i+1}, b_{i+1}] = \emptyset$. There is no intersection between the support of each two consecutive latent variable distribution.

(ii) $\exists i$ such that $[a_i, b_i] \cap [a_{i+1}, b_{i+1}] \neq \emptyset$. There is some intersection between the support of two consecutive latent distributions.

[2]More generally, composing unbounded samples with a monotonic, smooth enough, bounded function.

[3]The composition of Gaussian distribution with logistic function is the *logit-normal distribution.*

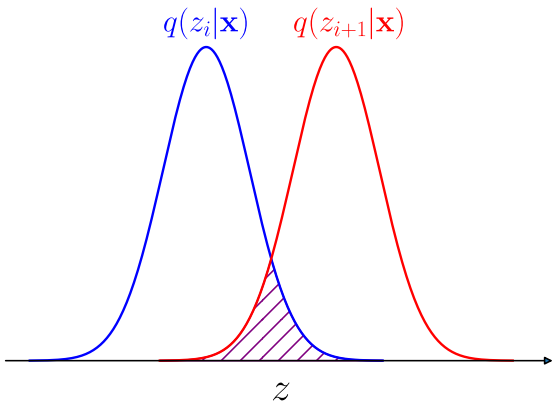


Fig. 5. Marginal densities $q(z_i|\mathbf{x})$ and $q(z_{i+1}|\mathbf{x})$ of latent variables z_i and z_{i+1} with intersecting support. If sampled independently, there is a non-zero probability that $z_i > z_{i+1}$.

In the first case, the independent sampling of each z_i will always yield ordered results. In the second case, there is a non-zero probability of sampling unordered samples z_i and z_{i+1} from two consecutive latent distributions $q(z_i|\mathbf{x})$ and $q(z_{i+1}|\mathbf{x})$ (see Fig. 5).

To ensure that latent samples are always ordered in this second case, we identify the three following strategies:

1) Penalizing out-of-order latent samples: Enforcing ordered constraints by just penalizing latent samples that are out of order would decrease the latent distributions widths and prevent latent distributions from being too close. As a result, the network would arbitrarily infer disjoint marginal distributions. This solution is not applicable in general, because it introduces an inductive prior of distribution disjointness that is not necessarily assumed by the physical-based decoder. Furthermore it would also hamper training by introducing noise into the loss.

2) Inferring the distribution of the difference between two variables: The second way of ensuring the order of samples is to infer positive support distributions of the difference $\Delta_{z_{i+1}}$ between each pair of consecutive variables (z_{i+1}, z_i) in the ordered sequence (see (14)).

$$z_{i+1} = z_i + \Delta_{z_{i+1}}, \quad \forall i \in [\![1, n-1]\!] \tag{14}$$

However this method increases the variance of the summed latent variables. Indeed, the density of the sum of random variables is the convolution of the densities of these variables, and the convolution of two densities results in a wider density.

3) Infering the distribution of the maximum of two variables: To overcome previous methods shortcomings, we propose to use the distribution of the maximum of two consecutive variables (z_{i+1}, z_i) as the distribution of the greater variable z_{i+1}.

To perform that, for consecutive latent distributions, it is necessary to ensure that samples are ordered, and that the expectations of the distributions are ordered. The former is achieved with the rectification of latent samples, while the latter is attained with the rectification of the variational

parameters and with the use an additional loss term on the variational parameters. The sampling procedure of ordered latent variables is illustrated in Fig. 6.

To *rectify* latent samples, the maximum value between a sample of consecutive variables (z_i, z_{i+1}) is attributed to the the greater variable z_{i+1} (see (15)). If the samples are ordered beforehand, the rectification doesn't change the value of the greater variable. If samples were mis-ordered, this sets the value of the greater variable as equal to that of the lower variable z_i. The resulting rectified samples $\underline{z}_{i+1}$ are then used instead of z_{i+1} by the user-defined decoder.

$$\underline{z}_{i+1} = \max\left(z_{i+1}, z_i\right), \quad \forall i \in [\![1, n-1]\!] \tag{15}$$

The distribution of each z_i is then the distribution of the maximum of all previous consecutive variables $z_i = \max_{j \leq i}(z_j), \quad \forall i \in [\![1, n-1]\!]$. The density (PDF) and cumulative distribution function (CDF) of rectified latent variables are available (see appendix D) if the PDF and CDF of all marginal latent distributions are available (marginal distributions can even be from different distribution families). Sample rectification is effective when distributions of consecutive latent variables overlap.

Since the rectification step takes place after the variational parameters inference, the model may rely solely on the rectification step to produce ordered latent variables. When the expectations of two consecutive latent distributions $q(z_i|\mathbf{x})$ and $q(z_{i+1}|\mathbf{x})$ are mis-ordered, the rectification step will mostly make consecutive latent samples identical. The encoder might converge sub-optimally and even though latent samples would technically be ordered, they would never be the right value.

To mitigate this,the expectation of consecutive latent distributions must be ordered as well. The two additional proposed techniques aim at ensuring that the encoder outputs variational parameters that satisfy this constraint. These methods can be applied when a latent distribution parameters λ_i, associated with z_i controls the expectation of the distribution, such as the mean parameters of Gaussians. In the following, we will assume that z_i are Gaussian-based, and denote μ_{z_i} their mean. Similar methods can be designed with other parameters with other distributions.

The rectification of the mean μ_{z_i} of Gaussian-based latent distributions (see (16)) is similar to the rectification of latent samples.

$$\underline{\mu}_{zi+1} = \max\left(\mu_{z_{i+1}}, \mu_{z_i}\right), \quad \forall i \in [\![1, n-1]\!] \tag{16}$$

This hard constraint guarantees that the expectation of the resulting distributions are ordered. However, rectifying latent distribution parameters can again lead to sub-optimal training. The encoder may not learn to output $\mu_{z_{i+1}} > \mu_{z_i} \quad \forall i$, and may always rely on the rectification step to produce distributions that have ordered expectations, leading to $\mu_{z_{i+1}}$ and μ_{z_i} being always equal.

To ensure proper learning, we add a soft constraint in the form of a loss term in (17), that penalizes inference of unordered latent distribution parameters.

$$\mathcal{L}_{\text{order}} = \frac{1}{N}\sum_{i=1}^{N} \underline{\mu}_{z_i} - \mu_{z_i} \tag{17}$$

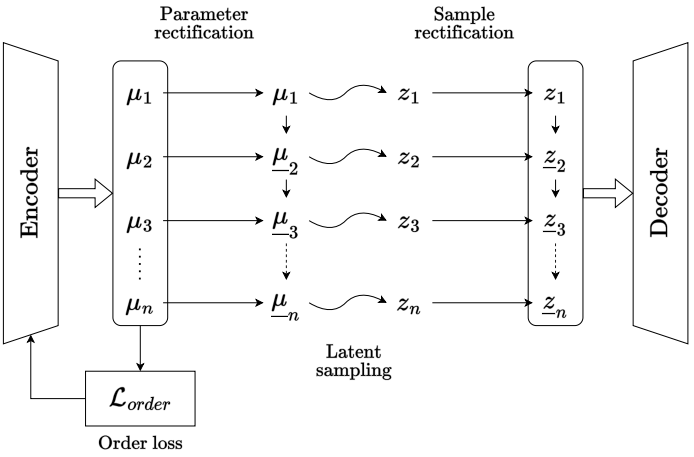


Fig. 6. Procedure of latent samples z_i ordering with maximum of latent distributions, with latent distribution parameters λ_i.

The *order loss* in (17) can be interpreted as an additional prior on latent distribution that the original KL term doesn't enforce.

Finally, using the maximum of consecutive variables to order the lhem does change their distribution (see appendix D for the density of the maximum of random variables). The prior distribution and the KL loss term can both be expected to become harder to derive for such latent distributions. In such case, we advocate for using the latent distribution without taking the ordering procedure into account in the computation of the prior and the KL term.

IV. Application: Inferring phenological parameters from NDVI time series

The interest of the proposed physics-guided representation learning methodology is illustrated by a well-known remote sensing inverse problem. Specifically, the goal is to infer the probability distributions of the intrinsic phenological parameters from NDVI[4] times series by considering a vegetation phenological model. Here we present the architecture of phenoVAE that integrates this model, and two data-sets that we use later for training and validation purposes.

A. The phenological model as physics-based decoder

The NDVI quantifies land surface greenness and photosynthetic vegetation vigor [36]. It is derived from Near Infra-Red (NIR) and Red reflectances (R) of a land surface, and its expression is:

$$\text{NDVI} = \frac{\rho_{NIR} - \rho_R}{\rho_{NIR} + \rho_R} \in [-1, 1]\,. \tag{18}$$

This index is typically close to 1 for high densities of vegetation, close to 0 for bare soil, and negative for water.

The characterization of the evolution of vegetation phenology using NDVI derived from remote sensing is widely addressed in the literature [37]–[39]. In general, the annual evolution of NDVI of some vegetation and crops can be well fitted with a double-logistic model [40]–[42]. The inversion

[4]Normalized Difference Vegetation Index.

TABLE I
PARAMETERS OF THE DOUBLE-LOGISTIC PHENOLOGICAL MODEL.

Variable	Description	Range $[a, b]$
M	Maximum of double logistic	[-1, 1]
m	minimum of double logistic	[-1, 1]
sos	DOY[5] of *Start Of Season*, the start of NDVI growth	[-45, 410]
mat	DOY of *Maturity*, the end of NDVI growth	[-45, 410]
sen	DOY of *Senescence*, the start of NDVI decay	[-45, 410]
eos	DOY of *End Of Season*, end of NDVI decay	[-45, 410]

of this *phenological model* from NDVI time series allows extracting *phenological parameters*, that typically characterize *phenophases* of the observed vegetation. This model uses a 6-variable phenological model to characterize seasonal vegetation cycles on yearly time series. The phenological model is described by the following equations:

$$\Omega_{\mathbf{z}}(t) = (M - m)\left(S_{sos,mat}(t) - S_{sen,eos}(t)\right) + m; \tag{19}$$

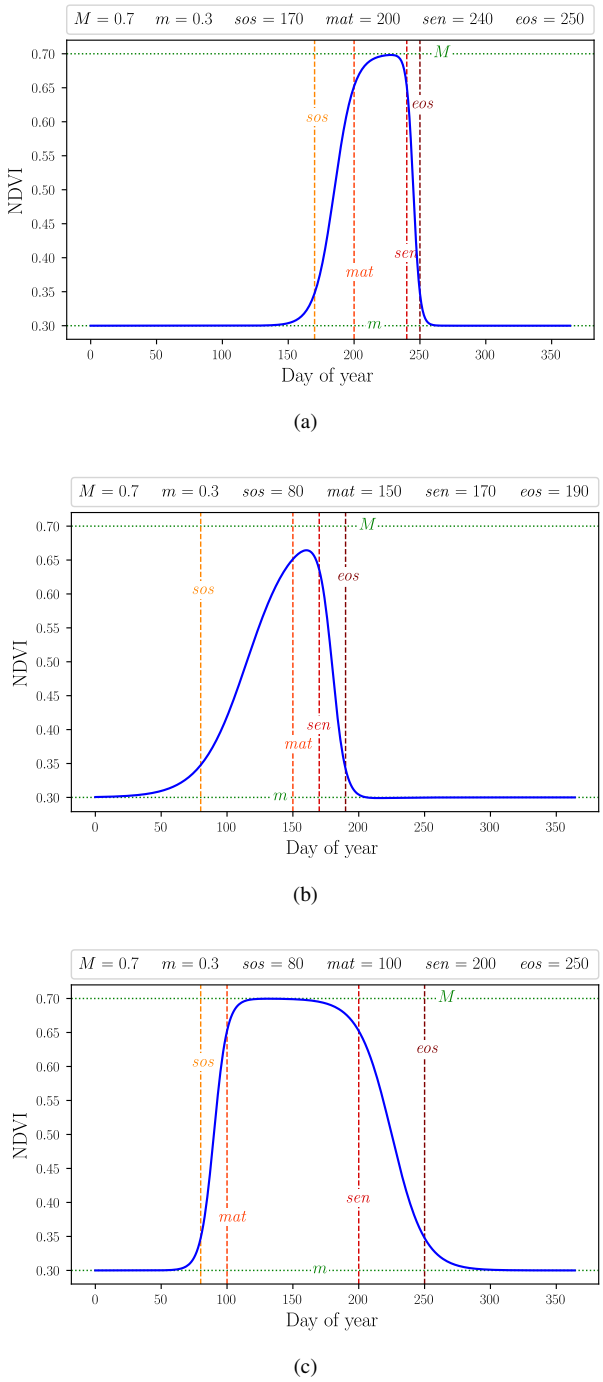
$$S_{\text{sos},mat}(t) = \left(1 + \exp\left(2\frac{sos + mat - 2t}{mat - sos}\right)\right)^{-1}; \tag{20a}$$

$$S_{sen,eos}(t) = \left(1 + \exp\left(2\frac{sen + eos - 2t}{eos - sen}\right)\right)^{-1}. \tag{20b}$$

This model accounts for vegetation annual cycle, with a growth phase, a stagnation phase and a decay/harvest phase. The 6 *phenological parameters* $\mathbf{z} = (M, m, sos, mat, sen, eos)$ are described in Table I, and their effects on the model are shown in Fig. 7. Phenological parameters are all bounded. As observed, M and m have the same bounds as NDVI itself.

The range of *phenological dates* (*sos*, *mat*, *sen*, *eos*) are the days of a given calendar year, extended by 90 days. The 45 days considered before the 1st January and after the 31st December allows less restrictive estimations, and takes into account vegetations whose cycle started or ended outside the calendar year. This range is a prior knowledge about the data, like the double-logistic model itself.

Following the architecture depicted in Fig. 3, the proposed pheno-VAE architecture proposes to use the double-logistic phenological model as a physical-based decoder. Each variable z_i of its 6-dimensional latent space is semantically bounded to a phenological parameter. The reconstruction term is computed as discussed in Section III-B. To take into account that the phenological parameters are bounded, we choose Truncated Gaussians $\mathcal{TN}$ as latent distributions. The latent sampling process is performed with the inverse transform method in (11). The phenological variables are ordered: $m < M$, and $sos < mat < sen < eos$, meaning the associated latent variables must be ordered in the same way. For that, the phenological dates are sampled as the maximum of all previous phenological dates, using the three strategies defined in Section III-D3: (i) the rectification of samples z_i in (15), (ii) the rectification of parameters μ_{z_i} of Truncated Gaussians in (16), (iii) the incorporation of an order term to the ELBO loss in (17).



(a)

(b)

(c)

Fig. 7. Examples of double logistic curves describing different phenology for different vegetation covers, with different phenological parameters.

As the latent variables of pheno-VAE are used as phenological model parameters, their distributions will be referred to as *phenological distributions*.

B. Encoder of Pheno-VAE

Pheno-VAE uses a simple multi-layer perceptron as shown in Fig. 8 that outputs the parameters of truncated Gaussians μ_{z_i} and σ_{z_i} for each variable z_i. The support $[a_i, b_i]$ of each truncated Gaussian is set to $[0, 1]$. Each sample drawn from these distributions is scaled accordingly to the range described in Table I, using the procedure described in (13), before being input to the physics-based decoder.

The neural network is implemented using PyTorch. As there is no temporal encoding of time series, its input layer of size 73 is presented with annual NDVI time series sampled in a 5-days regular grid.

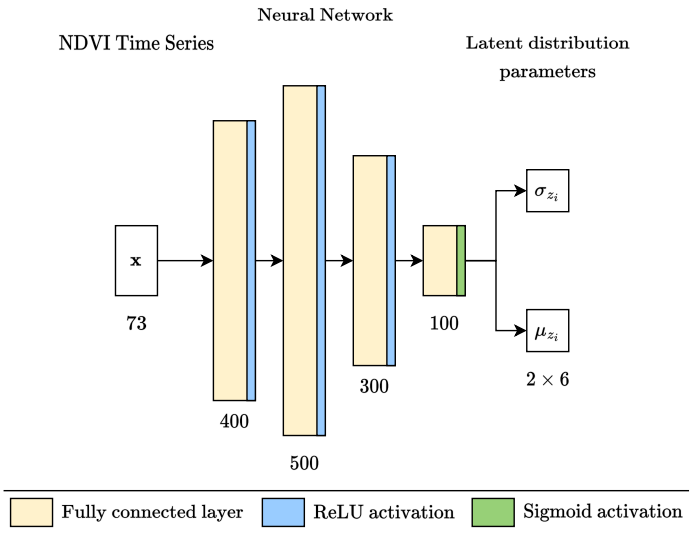


Fig. 8. Encoder architecture used in pheno-VAE, with 4 fully connected hidden layers with ReLU activation.

C. Data-sets

Two data-sets are used to evaluate the performances of pheno-VAE for phenological parameter retrieval. The first data-set is composed of real satellite observations of annual NDVI time series and is used for pheno-VAE training and qualitative validation. The second data set is composed of simulated crop NDVI profiles. The construction of this data-set is proposed for three main reasons: (i) to perform a quantitative evaluation of parameter retrieval on a large scale data-set, (ii) to assess the robustness of pheno-VAE to the noise of complex satellite observations, (iii) to compare the results of pheno-VAE against supervised methods. Examples of NDVI time series from both of data-sets are illustrated in Fig. 10.

1) S2 data-set: It is composed of 10^6 annual time series of pixels from 31TCJ Sentinel-2 tile (Toulouse area in southern France) [43]. The corresponding NDVI time series are computed from the spectral band 4 (Red) and 8 (Near Infra-Red). The resulting time series describe different land cover classes which can be associated to the class legend used on the OSO land cover map [44]. Accordingly, a large number of time series do not represent vegetation classes following the double-logistic phenological model. Despite the availability of land cover class information, it must be remarked that such information is only used for validation purposes. The distribution of the land cover classes in the data-set is detailed in Table VII of appendix A. The times series are acquired on irregular time intervals. The two Sentinel-2 satellites have intersecting ground footprints and some locations get increased coverage. Cloud cover is the main reason for the inconsistent temporal sampling and account for the variability in valid observation number of each pixel on the ground (see Fig. 12 in appendix A). For each time series, a validity mask is available to denote the valid satellite observations. This mask is used to linearly interpolate raw time series to a common regular time grid for pheno-VAE's encoder.

2) Simulated data-set: The corresponding data set is composed by a large number of simulations obtained by the double-logistics model, using a given sampling strategy for the input parameter ranges. The phenological model is used to generate time series samples with reference phenological parameters. This allows to compute metrics on phenological parameter retrieval experiments to validate our approach.

Since we assume that the double-logistic model is an approximation of NDVI time series, we model the observations as a noisy version of such a model. Therefore, we assume that each NDVI observation follows a normal (Gaussian) distribution whose mean is the double logistic function:

$$y(t) \sim \mathcal{N}\left(\mu(t), \sigma_n\right), \tag{21}$$

with $\mu(t) = \Omega_{\mathbf{z}}(t)$, and σ_n the standard deviation of the noise.

To generate synthetic time series, we first sample phenological parameters from uniform distributions. The phenological model is then used to simulate the corresponding NDVI time series. To account for the uneven temporal sampling of real time series, we use binary masks of valid dates of real S2 time series, to only simulate the time series at certain dates t. A Gaussian noise of randomly sampled standard deviation $\sigma_n \sim \mathcal{U}(0, 0.1)$ is added to remaining simulated time series points. It accounts for epistemic uncertainty, as no real time series is perfectly described by the phenological model. The resulting time series are finally interpolated at a regular 5-days time grid. The data generation procedure is depicted in Fig. 9.

The configuration of the parameter sampling procedure is detailed in the following. For σ, which represents the standard deviation of the noise in the observations, we will chose a maximum value of 0.1 which corresponds to 10% of the maximum expected range for NDVI values. For the minimum value of NDVI (m), we define the range between 0 (bare soil) and 0.4 (presence of vegetation). The maximum value of NDVI (M) is defined relative to the minimum value. A crop with a typical phenology value is assumed to have M at least 0.3 higher than m, and that the highest value will not be higher than 1.

The 4 dates characterizing the phenological stages are each defined in terms of the previous as follows. End of season (*eos*) is allowed to be right after Senescence and up to 90 days later. Senescence (*sen*) is defined in the same way with respect to maturity (*mat*) and *mat* follows the same rationale with respect to start of season (*sos*). For *sos* we would need to give a very wide prior in order to take into account winter and summer crops. Instead of doing that, we introduce an additional (latent)

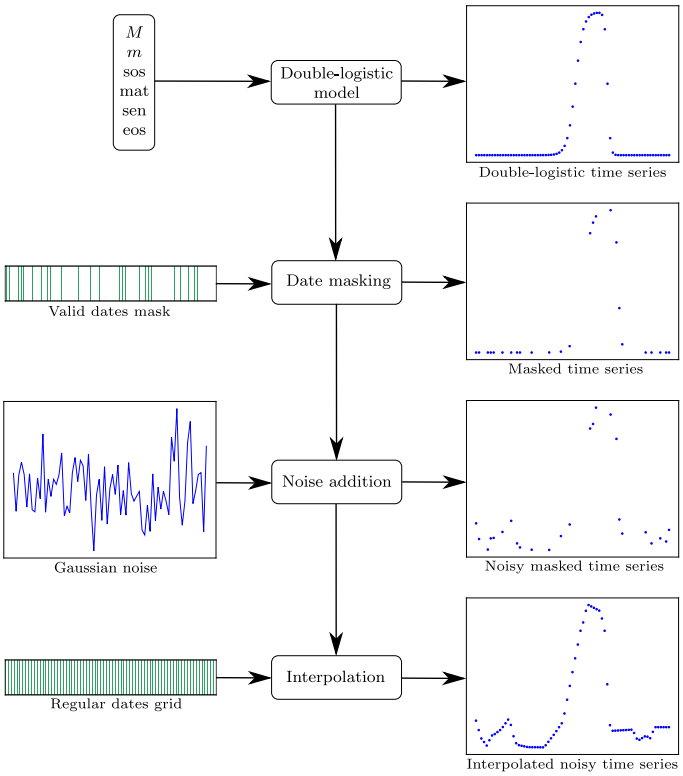


Fig. 9. Procedure of generation of a data-set of synthetic NDVI Time series.

variable sos_i which will model the probability of summer crop. This probability is used to adjust the starting point of the interval of prior values for *sos*. We assume that the earliest *sos* for a winter crop is on day 30 (end of January) and that the earliest summer crop can have an *sos* of 120 (late April). sos_i and σ_n are additional variables of the generative process of synthetic data that will not be inferred during experiments.

Sampling of parameters for synthetic time series generation is summarized in Table II.

TABLE II
DISTRIBUTIONS OF REFERENCE PHENOLOGICAL PARAMETERS SAMPLED FOR NDVI TIME SERIES SIMULATION WITH THE DOUBLE-LOGISTIC MODEL.

Parameter	Sampling interval
m	$\mathcal{U}(0, 0.4)$
M	$\mathcal{U}(m, 1)$
sos_i	$\mathcal{U}(30, 120)$
sos	$\mathcal{U}(sos_i, sos_i + 90)$
mat	$\mathcal{U}(sos, sos + 90)$
sen	$\mathcal{U}(mat, mat + 90)$
eos	$\mathcal{U}(sen, \text{sen} + 90)$
σ_n	$\mathcal{U}(0, 0.1)$

Even though the synthetic data-set is generated to be as realistic as possible, it is still different from the S2 data-set. Because of the uniform sampling of phenological dates in the synthetic data-set, there is more diversity in the phenology than the S2 data-set. On the one hand, the S2 data-set is biased by the samples that have been chosen among available real NDVI time series. All samples belong to the same S2 tile so NDVI time series of pixels of the same type are highly correlated, and

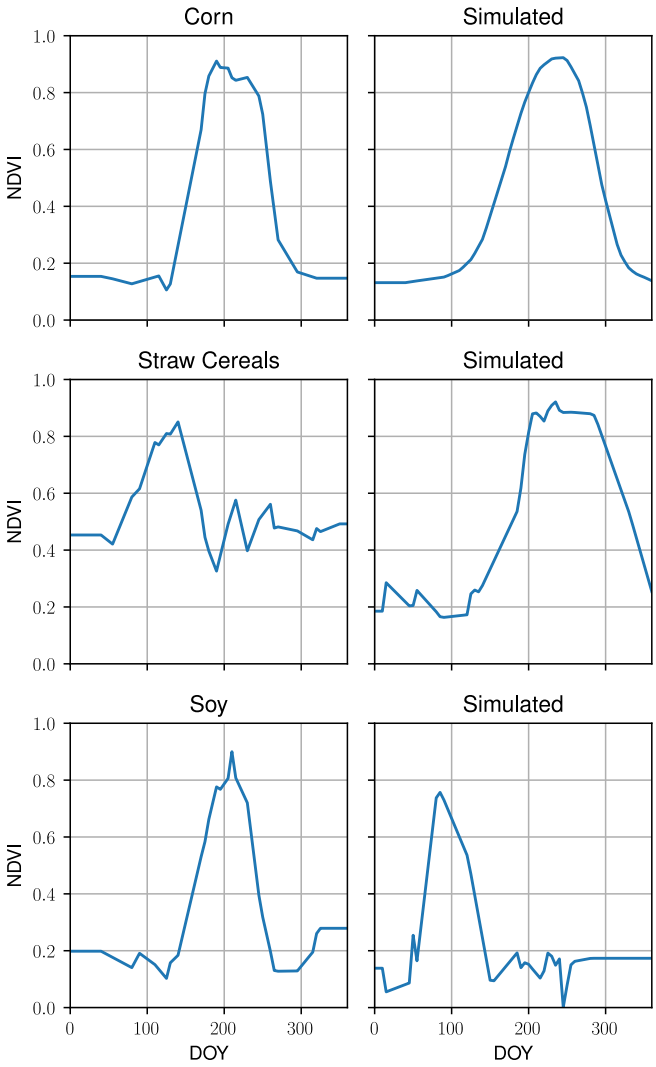


Fig. 10. NDVI time series of samples of S2 data-set (left) and simulated data-set (right)

cloud coverage similarly affects all time series. On the other hand, the synthetic data-set contains samples whose phenology that may not be frequent in reality, or even phenology types that don't exist. These differences will have to be taken into account in the interpretation of the results.

D. Latent prior distribution and KL term

Because of the variety of training samples in both data-sets, in terms of phenology or even in terms of aleatoric and epistemic uncertainty, it is difficult to design a very restrictive prior. We chose a uniform distribution for all latent variables over their respective density support. The KL-divergence between $q_\lambda(z|x) \sim \mathcal{TN}(\mu, \sigma, a, b)$ and $p(z) \sim \mathcal{U}(a, b)$ is given in (22) (see derivation in appendix E) with $\eta = \Psi(\tilde{b}) - \Psi(\tilde{a})$, $\tilde{a} = \frac{a-\mu}{\sigma}$, $\tilde{b} = \frac{b-\mu}{\sigma}$, and ψ and Ψ are respectively the standard Gaussian PDF and CDF.

$$\mathbb{KL}\left(q_\lambda(z|x)\|p(z)\right) = -\frac{1}{2} - \ln\left(\sqrt{2\pi}\sigma\eta\right) - \frac{\tilde{a}\psi(\tilde{a}) - \tilde{b}\psi(\tilde{b})}{2\eta} + \ln(b-a) \tag{22}$$

In practice, this loss promotes the inference of Truncated Gaussian posteriors with larger variances, while not penalizing their locations. Samples of the simulated and S2 data-sets have a wide variety of potential phenological parameters, and this loss doesn't promote any particular value for inference. In the S2 data-set, many samples don't have a phenology (buildings, mineral surfaces). For these time series, the reconstruction error should be high and variance of phenological parameters should increase to express epistemic uncertainty.

E. Loss of pheno-VAE

The loss of pheno-VAE for a single NDVI time series $\mathbf{x}$ is the sum of three terms:

$$\mathcal{L}_{pheno-VAE} = \mathcal{L}_{rec} + \beta\mathcal{L}_{kl} + \mathcal{L}_{order}. \tag{23}$$

The loss components are:

- $\mathcal{L}_{rec} = \frac{1}{2}\left[\left(\mathbf{x} - \boldsymbol{\mu}_{\hat{\mathbf{x}}}\right)^\top \boldsymbol{\Sigma}_{\hat{\mathbf{x}}}^{-1}\left(\mathbf{x} - \boldsymbol{\mu}_{\hat{\mathbf{x}}}\right) + \ln\left(|\boldsymbol{\Sigma}_{\hat{\mathbf{x}}}|\right)\right]$
- $\mathcal{L}_{kl} = \sum_{i=1}^{6} -\frac{1}{2} - \ln\left(\sqrt{2\pi}\sigma_{z_i}\eta_i\right) - \frac{\tilde{a_i}\psi(\tilde{a_i}) - \tilde{b_i}\psi(\tilde{b_i})}{2\eta_i}$, with $\tilde{a_i} = -\frac{\mu_i}{\sigma_i}$ and $\tilde{b_i} = \frac{1-\mu_i}{\sigma_i}$ (as $a_i = 0$ and $b_i = 1$)
- $\mathcal{L}_{order} = \frac{1}{6}\sum_{i=1}^{6} \underset{z_i}{\mu} - \mu_{z_i}$.

In practice, $\mathcal{L}_{order}$ converges to zero very fast, leaving only the two other terms in most of the training. There is a tension between the two remaining terms: the reconstruction loss improves the quality of the reconstructed time series, and the Kullback-Leibler divergence acts as a regularizer of the latent space. The balance between these two terms is shifted with the incorporation of the coefficient β for the KL term, in a similar manner to that of β-VAE [17]. The influence of this hyper-parameter is studied in the following results.

V. Experiences

In this section, we detail our experiments, the evaluation metrics that quantify the quality of the inferred representations, and their associated results.

A. Experimental setup

Given our phenological parameter retrieval application, different experiments are carried out to assess the performances of pheno-VAE. Firstly, we use reconstructions of NDVI time series from trained pheno-VAE to assess the quality of latent representations. Secondly, we evaluate metrics on phenological distributions using a simulated data-set. These metrics are considered in two sets of experiments. The first is a comparison between instances of pheno-VAE with different values of β, so that its influence can be studied and an optimal setting can be achieved. In the second set of metrics assessment, pheno-VAE is compared to two classical parameter estimation methods: a regression neural network and a Bayesian model inversion method.

TABLE III
Characteristics and hyper-parameters of each experiments inference methods.

Exp.	MCMC	NN Regression	pheno-VAE (Sim)	pheno-VAE (S2)
Unsupervised	✓	✗	✓	✓
Training	None	Simulated Data-set	Simulated Data-set	S2 Data-set
Optimizer	None	Adam	Adam	Adam
Batch size	None	2048	2048	2048
Learning rate	None	5.10^{-4}	5.10^{-4}	5.10^{-4}
Epochs	None	500	200	200
Number of latent samples K	None	None	10	10
Point estimate	median	mode	mode	mode

The regression neural network is proposed for comparison with pheno-VAE because parameter retrieval can be considered as a multi-output regression problem. The supervised training of this network is performed using the simulated data-set. The regression network also infers Truncated Gaussian parameters, so that this method also predicts a distribution for the phenological parameters. The training loss is the Negative Log-Likelihood of Ordered Truncated Gaussians (see appendix D). To ensure that model complexity doesn't influence comparative results, the architecture of the regression network is identical to that of pheno-VAE (see Fig. 8). This experiment is proposed to provide a comparison of our unsupervised pheno-VAE against a supervised model of similar architecture.

There exist many techniques for performing Bayesian inference, that is, obtaining the posterior distributions of the model parameters given the observations. Describing them is out of the scope of this paper. For our work, we will use Markov Chain Monte Carlo (MCMC) in order to obtain samples from the posterior distributions of the phenological parameters. Following the methodology of [45], we use Hamiltonian Monte Carlo as per the NUTS algorithm [46] as implemented in the NumPyro library [47], [48].

To implement Bayesian inference through MCMC we need to define the likelihood for the observed data. Here, we use a Gaussian likelihood with the double-logistic function as its mean (see (21)). This model is the same as the one used to generate time series, (see Fig. 9), except that there is no interpolation step involved. At inference, MCMC takes as input uninterpolated time series with the temporal position of each data point. As prior distributions, we choose the same uniform distributions than those in Table II to simulate the NDVI time series data-set.

Finally, two instances of pheno-VAE are considered, one trained with the S2 data-set, and the other with the synthetic data-set, to evaluate the influence of the training data-set. The four experimental setups are summarized in Table III.

B. Evaluation metrics

The three metrics described in the following are used to evaluate the accuracy of the retrieved parameters and their corresponding uncertainties. They are evaluated on a validation data-set of $N = 10000$ samples.

1) Point estimate inference error: The Mean Absolute Error (MAE) from a distribution point estimate:

$$\text{MAE}(i) = \frac{1}{N}\sum_{j=1}^{N}\left|z^*_{i,j} - \hat{z}_{i,j}\right| \tag{24}$$

With $z^*_{i,j}$ the reference value of the phenological parameters i for the time series j in the simulated data-set, and $\hat{z}_{i,j}$ a point estimate of the predicted distribution associated with $z_{i,j}$. We chose the point estimator that gave the best MAE with each setup: the mode of the distributions for the Neural Network Regression, pheno-VAE (S2 & sim), and the median for MCMC.

2) Prediction interval metrics: Prediction intervals are at the core of uncertainty quantification [49]. A prediction interval of a predicted variable Z with a confidence level $1-\alpha$ is the smallest range $[l, u]$ that satisfies:

$$P(Z \in [l, u]) \geq 1 - \alpha. \tag{25}$$

In this work, we estimate the prediction intervals $[l_{i,j}, u_{i,j}]$ of latent variables z_i for each data sample x_j, from the inferred latent distributions. Specifically, $l_{i,j}$ is taken as the $\alpha/2$ quantile of the corresponding latent distribution, and $u_{i,j}$ as the $(1-\alpha)/2$ quantile.

In practice, we select $\alpha = 0.1$ for 5th-95th centile intervals, in results pertaining to prediction intervals presented below. Results obtained with different confidence levels are shown in appendix C. We detail in the following two prediction intervals metrics widely used in the literature [50], [51].

- The Mean Prediction Interval Width (MPIW) is the average length of the prediction intervals $[l_{i,j}, u_{i,j}]$ derived from the predicted distributions. The narrower the interval length is, the more confident we can be about the prediction.

$$\text{MPIW}(i, \alpha) = \frac{\sum_{j=1}^{N} u_{i,j} - l_{i,j}}{N} \tag{26}$$

- The Prediction Interval Coverage Probability (PICP) measures the frequency of the model parameters true value being inside the prediction interval:

$$\text{PICP}(i, \alpha) = \frac{\#\left\{j \text{ s.t. } z^*_{i,j} \in [l_{i,j}, u_{i,j}]\right\}}{N} \tag{27}$$

  This metric is an estimate of the probability of time series of a data-set having their true parameters inside the prediction interval. Note that this metric doesn't quantify the probability of a single time series' true parameter of being inside a prediction interval. This probability cannot be accessed, because assessing belonging for a given time series is a binary experience.

  This metric should be as close to α than possible. When the inferred phenological distribution matches perfectly the density of the true phenological parameters, then $\text{PICP} = \alpha,\ \forall\alpha$.

The three evaluation metrics are computed by using a K-fold cross-validation procedure, in which the data-set is divided into K folds. In each round, a model is trained using K − 1 of the folds as training data and tested on the remaining set. Metrics are then measured by averaging the performance values computed on each subset (K models). This strategy is applied to validate deep learning approaches. For MCMC, metrics are independently obtained on K subsets of the total data-set. The averages and standard deviations of the results on those subsets are computed. In the following, K is equal to 6.

C. Evaluation of the reconstruction results

To assess the performances of pheno-VAE, a visual evaluation is presented in Fig. 11. This figure shows S2 NDVI time series samples with their reconstruction by pheno-VAE trained on S2 data, with their corresponding phenological distributions. In most cases shown here, the setting $\beta = 0$ imposes that no prior information from the dataset is incorporated. This is different from our uniform prior that assumes that phenological variables are evenly distributed over their possible range.

Fig. 11a shows NDVI time series of a pixel of corn, the inferred phenological distribution and the reconstruction of its mode. The reconstruction curve is observed to accurately match the original time series. The distributions of phenological dates characterize well the growth and decay phases of this summer crop. The reconstruction error and variance of reconstructions are both low. The estimated phenological distributions seem well centered on likely phenological parameters.

The influence of β can be evaluated by comparing the results observed on Fig. 11(a) and 11(b). The same NDVI time series of a corn pixel is taken as input by two pheno-VAE models with different values of β. The modal reconstructions are very similar. With increasing β, the phenological distributions widen, and the variance of reconstructions increases. This is coherent with the influence of the KL loss terms, that discourages narrow latent densities. With both results well matching the original NDVI time series, the choice of β is to be made considering the prediction interval metrics.

On Fig. 11(c), a protein crop time series shows how the presence of data gaps can lead to bad phenological parameter estimation. In this figure, the phenological cycle is easily identifiable. However, bad weather in winter led to a lack of data points for the first two months, and the backward extrapolation of points at pre-processing has kept the NDVI artificially constant, at a higher value than after harvest. As the encoder of pheno-VAE doesn't take into account the temporal information, here reconstruction is disrupted by the gapfilling step. This extrapolation artifact made the input time series not well described by the phenological model at the beginning of the year. The start of season estimate is inaccurate, yet the distribution large spread indicates greater uncertainty. This bad inference of the start of season seems to have prevented a good estimation of the maturity date as well, with this time a narrow distribution. Nonetheless the senescence and end of season seem well inferred. Similarly with a broad-leaved forest time series (Fig. 11(d)), senescence and end of season distributions are not well positioned due to interpolated data points at the end of year. These results show that the gapfilling pre-processing task can lead to wrong parameter estimations

when long data gaps include key phenological dates. This highlights the need for encoders that don't rely on interpolated inputs. However, that would be out of the scope of our current contribution.

In Fig. 11(e), there are several crops in the pixel, and the NDVI time series shows several phenological cycles. As the model can only take one cycle into account, it only fits the largest, and takes the average of the remaining signal. The distribution of the minimum of NDVI is very large, indicating uncertainty.

In Fig. 11(f), the phenological model doesn't suit at all the NDVI time series of a dense building pixel. Therefore, reconstruction errors are high. However phenological distribution variances increase to take this epistemic uncertainty into account.

The results show that large uncertainties could be associated to the model discrepancy with the data.

Another remark is that, inferred marginal phenological distributions sometimes show significant overlap. This highlights the interest of the proposed order constraints on the latent distributions, as reconstructions are consistent with the phenological model, and variables constraints are always respected.

More reconstruction examples are available in appendix B.

D. Influence of the KL loss term on pheno-VAE performances

The impact of the KL term is studied by comparing results obtained by using different β values. In this experiment, pheno-VAE model is trained with samples from the S2 data-set. The prediction interval metrics presented here are derived for a confidence level of $1 - \alpha = 0.9$.

As previously observed, the KL term tends to increase the dispersion of latent distributions. The MPIW (Table IV(c)) increases for all phenological parameters along with β and consequently the PICP (Table IV(b)) also increases.

The MAE results (Table IV(a)) tend to increase along with β, decreasing performance. These results corroborate that the hyper-parameter β must be selected by using an independent validation data-set. For the prediction intervals to be informative, the KL term needs to be high enough, while keeping it below a certain threshold ensures that precision is acceptable.

Also, different performances are obtained for the different phenological parameters. The minimum of NDVI m is the best estimated parameter, as with simulated time series, a large part of available data points are around the value of the minimum — although, it is so well estimated that its prediction interval almost always contains it, overshooting the PICP $= 1 - \alpha$ target. The parameter M is more challenging to estimate than m. The value of the true maximum of the phenological model can differ from the parameter M when *mat* and *sen* are close. The highest errors are obtained on phenological dates, most certainly because of the gapfilling problem highlighted with reconstruction results (such as with Fig. 11(c) and 11(d)). This limitation is more visible in MPIW values obtained for *sos* and *eos* than *mat* and *sen*. This is because the pheno-VAE is confronted with more severe extrapolation aberrations at both ends of the time series than in the middle, where interpolation is better, with higher temporal availability in the original time series.

In the following, the setting $\beta = 2$ will be used, as it increases the PICP without degrading too much the MPIW and the MAE.

TABLE IV
EVALUATION PERFORMANCES OBTAINED ON A SIMULATED DATA-SET FOR DIFFERENT PHENO-VAE MODELS TRAINED ON THE S2 DATA-SET, AND FOR VARIOUS KL LOSS COEFFICIENTS β. PREDICTION INTERVALS ARE DERIVED FROM PHENOLOGICAL DISTRIBUTIONS WITH A CONFIDENCE LEVEL $1 - \alpha = 0.9$.

Exp	pheno-VAE (S2, $\beta = 0$)	pheno-VAE (S2, $\beta = 1$)	pheno-VAE (S2, $\beta = 2$)	pheno-VAE (S2, $\beta = 5$)
M	$\mathbf{0.05 \pm 0.00}$	$\mathbf{0.05 \pm 0.00}$	$\mathbf{0.05 \pm 0.00}$	0.07 ± 0.00
m	$\mathbf{0.02 \pm 0.00}$	$\mathbf{0.02 \pm 0.00}$	$\mathbf{0.02 \pm 0.00}$	$\mathbf{0.02 \pm 0.00}$
sos	$\mathbf{11.13 \pm 0.46}$	11.82 ± 0.27	11.93 ± 0.60	14.87 ± 0.21
mat	$\mathbf{10.22 \pm 0.08}$	10.38 ± 0.33	10.58 ± 0.25	14.37 ± 0.61
sen	$\mathbf{11.01 \pm 0.47}$	11.61 ± 0.65	12.15 ± 0.60	18.37 ± 0.75
eos	$\mathbf{13.35 \pm 0.52}$	13.48 ± 0.69	14.75 ± 0.97	18.69 ± 0.47

(a) Mean Average Error (mode of phenological distributions)

Exp	pheno-VAE (S2, $\beta = 0$)	pheno-VAE (S2, $\beta = 1$)	pheno-VAE (S2, $\beta = 2$)	pheno-VAE (S2, $\beta = 5$)
M	$\mathbf{0.67 \pm 0.01}$	0.60 ± 0.01	0.61 ± 0.02	0.63 ± 0.03
m	0.95 ± 0.01	0.95 ± 0.01	0.94 ± 0.01	$\mathbf{0.92 \pm 0.01}$
sos	0.34 ± 0.05	0.53 ± 0.02	0.64 ± 0.02	$\mathbf{0.77 \pm 0.01}$
mat	0.25 ± 0.03	0.48 ± 0.02	0.56 ± 0.01	$\mathbf{0.69 \pm 0.02}$
sen	0.34 ± 0.04	0.55 ± 0.01	0.64 ± 0.01	$\mathbf{0.69 \pm 0.02}$
eos	0.58 ± 0.02	0.71 ± 0.02	0.76 ± 0.03	$\mathbf{0.83 \pm 0.01}$

(b) Prediction Interval Coverage Probability

Exp	pheno-VAE (S2, $\beta = 0$)	pheno-VAE (S2, $\beta = 1$)	pheno-VAE (S2, $\beta = 2$)	pheno-VAE (S2, $\beta = 5$)
M	0.12 ± 0.01	$\mathbf{0.11 \pm 0.00}$	$\mathbf{0.11 \pm 0.00}$	0.16 ± 0.00
m	0.13 ± 0.00	$\mathbf{0.12 \pm 0.00}$	$\mathbf{0.12 \pm 0.00}$	$\mathbf{0.12 \pm 0.00}$
sos	$\mathbf{14.69 \pm 2.85}$	22.97 ± 1.38	27.93 ± 1.54	41.79 ± 1.64
mat	$\mathbf{8.81 \pm 1.11}$	18.24 ± 1.05	22.81 ± 0.75	38.24 ± 1.65
sen	$\mathbf{13.75 \pm 1.01}$	23.35 ± 1.18	28.43 ± 1.53	42.18 ± 1.36
eos	$\mathbf{30.60 \pm 1.83}$	36.60 ± 2.38	43.30 ± 3.10	59.64 ± 2.30

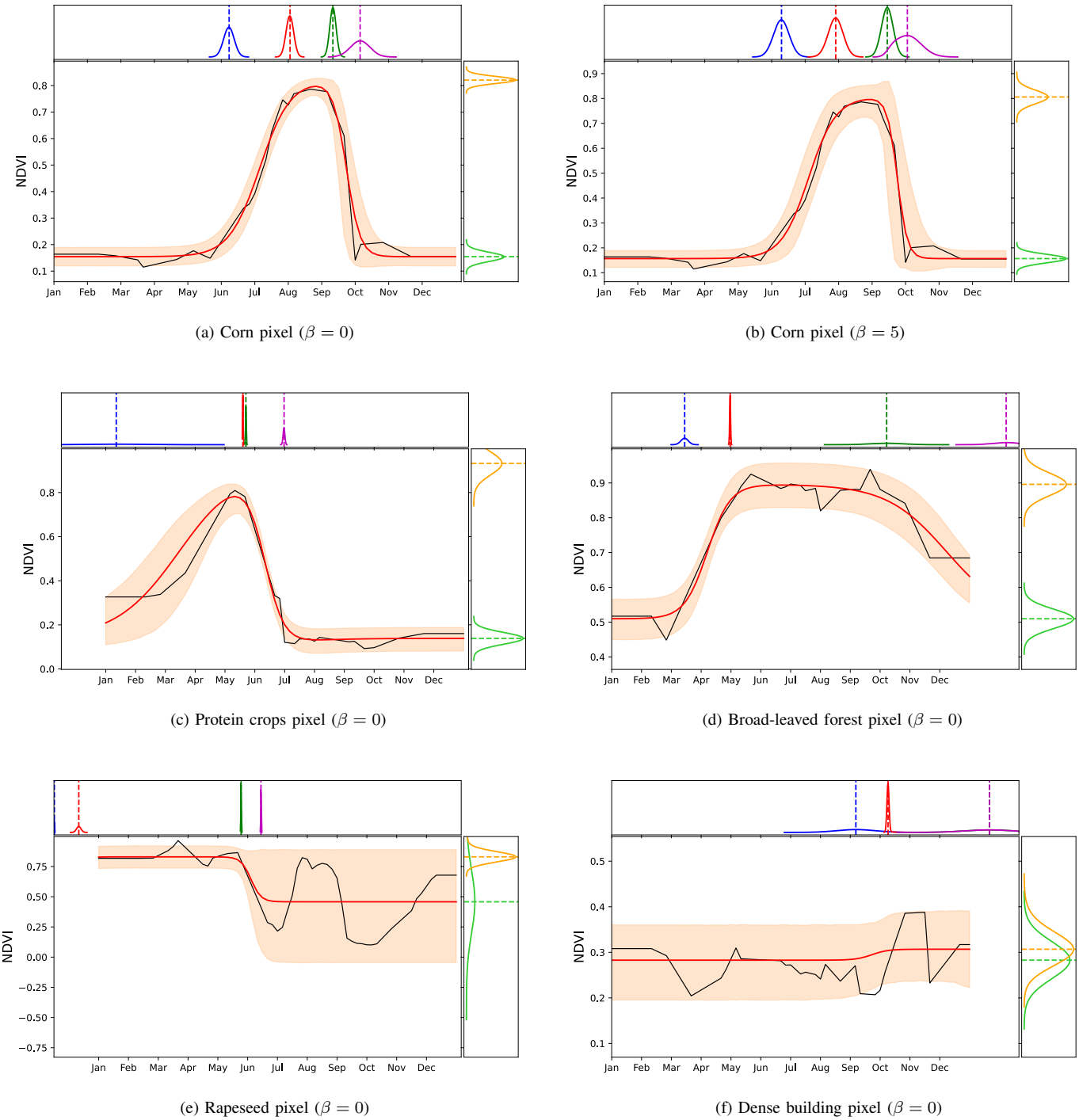
(c) Mean Prediction Interval Width

E. Comparing pheno-VAE to other inversion methods and influence of training data-set

The performances of inference of phenological distributions on a simulated validation data-set are compared between pheno-VAE trained on S2 data-set, pheno-VAE trained on the synthetic data-set, MCMC and Neural Network regression. The results are presented in Table V.

The experiment with the best overall performances is the Neural Network regression, which has the lowest MAE (Table V(a)), the PICP (Table V(b)) closest to $\alpha = 0.9$, with consistent MPIW (Table V(c)) on phenological dates. This is expected considering that it is a supervised method, with the training data-set being very similar to the testing data-set. Furthermore its loss doesn't rely on reconstruction, and therfore isn't affected the same way than pheno-VAE by interpolation.

MCMC has performances that are a little worse than regression, with a little higher error, and it compensates



(a) Corn pixel ($\beta = 0$)

(b) Corn pixel ($\beta = 5$)

(c) Protein crops pixel ($\beta = 0$)

(d) Broad-leaved forest pixel ($\beta = 0$)

(e) Rapeseed pixel ($\beta = 0$)

(f) Dense building pixel ($\beta = 0$)

Fig. 11. Reconstruction and latent distributions from the encoding of the NDVI time series by pheno-VAE trained on S2 data-set. Central quadrants, S2 NDVI time series (black), reconstructions from the modes of latent distributions (red), and reconstruction 5th-95th prediction interval - Upper quadrants: Truncated Gaussian distributions of the 4 phenological dates, *sos* (blue), *mat* (red), *sen* (dark green), *eos* (magenta) - Right quadrants: Truncated Gaussian distributions of M (orange), and m (light green) - Upper and right quadrants: distribution densities are in solid lines, distribution modes are in dashed lines.

smaller prediction intervals with lower PICPs. Phenological distribution inference is not limited by a distribution family prior and directly samples phenological distributions, contrary to the other methods studied here. It is also not affected by gapfilling problems because MCMC do not require regularly temporal input data. The results of MCMC could be improved by increasing the number of distribution samples and steps, at the expense of greater computation costs.

Besides, the computing time required to perform inference is orders of magnitude larger for MCMC than deep learning methods (see Table VI), justifying the use of the latter for large scale problems.

TABLE V
EVALUATION PERFORMANCES OBTAINED ON A SIMULATED DATA-SET FOR DIFFERENT EXPERIENCES INVERSION OF THE PHENOLOGICAL MODEL. PREDICTION INTERVALS ARE DERIVED FROM PHENOLOGICAL DISTRIBUTIONS WITH A CONFIDENCE LEVEL $1-\alpha = 0.9$.

Exp.	MCMC	NN Regression	pheno-VAE (Sim, $\beta = 2$)	pheno-VAE (S2, $\beta = 2$)
M	$\mathbf{0.03 \pm 0.00}$	0.04 ± 0.00	0.06 ± 0.00	0.05 ± 0.00
m	0.02 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.02 ± 0.00	0.02 ± 0.00
sos	7.18 ± 0.70	$\mathbf{6.69 \pm 0.03}$	8.89 ± 0.53	11.93 ± 0.60
mat	9.57 ± 0.95	$\mathbf{7.54 \pm 0.05}$	10.51 ± 0.49	10.58 ± 0.25
sen	9.93 ± 1.00	$\mathbf{6.91 \pm 0.05}$	10.59 ± 0.52	12.15 ± 0.60
eos	10.42 ± 1.18	$\mathbf{6.70 \pm 0.07}$	9.23 ± 0.26	14.75 ± 0.97

(a) Mean Absolute Error (mode of phenological distributions)

Exp.	MCMC	NN Regression	pheno-VAE (Sim, $\beta = 2$)	pheno-VAE (S2, $\beta = 2$)
M	0.89 ± 0.01	$\mathbf{0.90 \pm 0.01}$	0.67 ± 0.01	0.61 ± 0.02
m	0.86 ± 0.01	$\mathbf{0.90 \pm 0.01}$	0.99 ± 0.00	0.94 ± 0.01
sos	0.84 ± 0.01	$\mathbf{0.89 \pm 0.00}$	0.67 ± 0.05	0.64 ± 0.02
mat	0.85 ± 0.01	$\mathbf{0.89 \pm 0.00}$	0.60 ± 0.01	0.56 ± 0.01
sen	0.83 ± 0.01	$\mathbf{0.89 \pm 0.01}$	0.66 ± 0.01	0.64 ± 0.01
eos	0.83 ± 0.01	$\mathbf{0.88 \pm 0.00}$	0.77 ± 0.02	0.76 ± 0.03

(b) Prediction Interval Coverage Probability

Exp.	MCMC	NN Regression	pheno-VAE (Sim, $\beta = 2$)	pheno-VAE (S2, $\beta = 2$)
M	0.13 ± 0.01	0.16 ± 0.00	0.14 ± 0.01	$\mathbf{0.11 \pm 0.00}$
m	$\mathbf{0.05 \pm 0.00}$	0.06 ± 0.00	0.14 ± 0.00	0.12 ± 0.00
sos	22.13 ± 1.75	27.70 ± 0.30	$\mathbf{21.02 \pm 0.76}$	27.93 ± 1.54
mat	25.03 ± 1.94	29.91 ± 0.25	23.25 ± 1.32	$\mathbf{22.81 \pm 0.75}$
sen	22.74 ± 1.79	27.81 ± 0.43	$\mathbf{27.09 \pm 1.16}$	28.43 ± 1.53
eos	$\mathbf{21.50 \pm 2.29}$	26.36 ± 0.40	25.23 ± 0.80	43.30 ± 3.10

(c) Mean Prediction Interval Width

The worst results are shown by pheno-VAE. It has higher MAE, and despite similar prediction interval sizes, it underestimates uncertainty with lower PICP. Results also show different behaviors for the two pheno-VAE trained on different data-sets. As expected, slightly better results are obtained when pheno-VAE is trained on simulated data. A greater performance drop is observed for *eos*. This is because of a discrepancy between both data-sets. In the simulated data-set, there is more diversity in the phenological parameters, because of the uniform sampling to generate it. Even if real validity masks from the S2 data-set are used, they are not correlated to phenology, as it is the case for real data. In the S2 data-set, a smaller diversity of combinations of phenological variables is available. In this data-set, the end of season of real crops can happen when there are clouds, more than in the simulated data-set.

The drop in performances is much less significant compared to regression and MCMC, despite training on samples that don't follow the phenological model. The pheno-VAE trained on the synthetic data-set benefits from being evaluated on a similar simulated data-set. This unfair advantage could be mitigated by evluating performances of pheno-VAE on real Sentinel-2 NDVI time series data-set, with available ground truth of phenological stages. Unfortunately, such a data-set was not available to us at the time of this study.

TABLE VI
APPROXIMATE TRAINING AND INFERENCE TIME FOR EACH SETUP, ON GPU (TESLA V100-SXM2-32GB)

Exp.	MCMC	NN Regression	pheno-VAE (Sim)	pheno-VAE (S2)
Training	-	15 min	15 min	15 min
Inference per time series	10 s	10^{-5} s	10^{-5} s	10^{-5} s

MCMC and NN regression show similar performances, despite being very different methods. This hints that given the simulated data-set and the double-logistic model, there is not much performance improvement to expect from the inference experiment, even with other setups. The regression yields on phenological dates 7-day MAE, with 90% PICP and 28 days MPIW. These are good results considering irregularly sampled time series that are interpolated to a 5-day grid. For pheno-VAE to get performances closer to this, there is a need to improve on the ability of the network to take temporal structure of time series into account. To minimize the impact of the gapfilling pre-processing step, different solutions could be considered. For instance, the reconstruction loss could be modified to only take valid observations into account. The encoder network architecture could be replaced to allow to learn from irregularly sampled times series such as with transformers or recurrent networks. However selecting the best architecture to get state-of-the art performances on this limited experiment is beyond the scope of this study.

F. Ablation study of the latent distribution maximum sampling techniques

An ablation study for the method of ordering latent variables is performed with pheno-VAE, with the μ-rectification in (16), latent samples rectification in (15), and the order loss in (17). When any of these steps is removed, we observe that training convergence takes longer. It also often leads to sub-optimal models that only order distributions by making them identical. Moreover, simply removing the latent sample rectification leads the pheno-VAE to infer latent representations that fit the data but no longer have physical meaning (with for instance the *sos* date being after the *eos* date).

VI. CONCLUSION

The work presented here has proposed a new physics-guided methodology to learn probabilistic interpretable representations of satellite image time series. Different strategies are presented to incorporate physical knowledge in VAE by considering physical-based decoders.

Semantic latent variables bound to physical model parameters are learnt by incorporating prior knowledge and order constraints in the learning process. Monte Carlo sampling of the latent space was introduced to generate a reconstruction distribution from deterministic decoders. The classical pair of prior and posterior distributions was changed. Order constraints were added to better model the properties of physical

variables in a semantic latent space. A new KL loss term was calculated, whose weight in the loss enable to adjust the performance of the model. The training wasn't hampered by noisy Sentinel-2 data, with some of it not fitting the model. The feasibility and the interest of the proposed methodology are corroborated through a well-known remote sensing inverse problem, the phenological parameter retrieval from Sentinel-2 NDVI time series. This physics-guided representation learning approach can be applied to large scale remote sensing problems where reference data is scarce. Applying these methodologies to different models of more complex data will be the focus of future research efforts. These physics-guided learning methods are an important step toward the large scale production of interpretable representations of data, which is of great interest in remote sensing, where a wealth of literature on modeling of the observed processes is available. Despite using a simple neural network architecture, preliminary results are encouraging. Enhancing the encoder architecture with inductive biases taking into account the temporal structure of the data (attention mechanisms, recurrent architectures) can improve the inference error and predicted prediction intervals that fall behind other methods in the current configuration.

In an attempt to enable reproducible research, our implementation of the methods developped in this paper are available at the following: https://gitlab.cesbio.omp.eu/zerahy/pheno-VAE.git.

ACKNOWLEDGMENTS

The authors would like to thank CNES for the provision of its high performance computing (HPC) infrastructure to run the experiments presented in this paper and the associated help. We are especially grateful to Mathieu Fauvel and Julien Michel for their feedback on a preliminary version of this paper.

References

[1] A. Braun, F. Fakhri, and V. Hochschild, "Refugee camp monitoring and environmental change assessment of kutupalong, bangladesh, based on radar imagery of sentinel-1 and alos-2," *Remote Sensing*, vol. 11, no. 17, 2019, ISSN: 2072-4292. DOI: 10.3390/rs11172047. [Online]. Available: https://www.mdpi.com/2072-4292/11/17/2047.

[2] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, "Inceptiontime: Finding alexnet for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1936–1962, 2020.

[3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013. DOI: 10.1109/TPAMI.2013.50.

[4] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," *Advances in neural information processing systems*, vol. 31, 2018.

[5] V. Edupuganti, M. Mardani, S. Vasanawala, and J. Pauly, "Uncertainty quantification in deep mri reconstruction," *IEEE Transactions on Medical Imaging*, vol. 40, no. 1, pp. 239–250, 2021. DOI: 10.1109/TMI.2020.3025065.

[6] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2014. [Online]. Available: http://arxiv.org/abs/1312.6114.

[7] C. Doersch, *Tutorial on variational autoencoders*, 2016. DOI: 10.48550/ARXIV.1606.05908. [Online]. Available: https://arxiv.org/abs/1606.05908.

[8] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel, "Variational lossy autoencoder," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=BysvGP5ee.

[9] X. Zhu, C. Xu, and D. Tao, "Where and what? examining interpretable disentangled representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5861–5870.

[10] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, "Integrating physics-based modeling with machine learning: A survey," *arXiv preprint arXiv:2003.04919*, vol. 1, no. 1, pp. 1–34, 2020.

[11] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019, ISSN: 1935-8237. DOI: 10.1561/2200000056. [Online]. Available: http://dx.doi.org/10.1561/2200000056.

[12] A. Makhzani and B. J. Frey, "Pixelgan autoencoders," in *NIPS*, 2017.

[13] S. Yeung, A. Kannan, Y. Dauphin, and L. Fei-Fei, *Tackling over-pruning in variational autoencoders*, 2017. DOI: 10.48550/ARXIV.1706.03643. [Online]. Available: https://arxiv.org/abs/1706.03643.

[14] E. Mathieu, T. Rainforth, N Siddharth, and Y. W. Teh, "Disentangling disentanglement in variational autoencoders," in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., ser. Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 4402–4412. [Online]. Available: https://proceedings.mlr.press/v97/mathieu19a.html.

[15] H. Zhang, Y.-F. Zhang, W. Liu, A. Weller, B. Schölkopf, and E. P. Xing, "Towards principled disentanglement for domain generalization," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8014–8024. DOI: 10.1109/CVPR52688.2022.00786.

[16] M. Tschannen, O. F. Bachem, and M. Lučić, "Recent advances in autoencoder-based representation learning," in *Bayesian Deep Learning Workshop, NeurIPS*, 2018.

[17] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "Beta-VAE: Learning basic visual concepts with a constrained variational framework," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=Sy2fzU9gl.

[18] H. Kim and A. Mnih, "Disentangling by factorising," in *International Conference on Machine Learning*, PMLR, 2018, pp. 2649–2658.

[19] A. Kumar, P. Sattigeri, and A. Balakrishnan, "Variational inference of disentangled latent concepts from unlabeled observations," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=H1kG7GZAW.

[20] F. Locatello, S. Bauer, M. Lučić, G. Rätsch, S. Gelly, B. Schölkopf, and O. F. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *International Conference on Machine Learning*, Best Paper Award, 2019. [Online]. Available: http://proceedings.mlr.press/v97/locatello19a.html.

[21] F. Träuble, E. Creager, N. Kilbertus, F. Locatello, A. Dittadi, A. Goyal, B. Schölkopf, and S. Bauer, "On disentangled representations learned from correlated data," in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 10401–10412. [Online]. Available: https://proceedings.mlr.press/v139/trauble21a.html.

[22] K. Do and T. Tran, "Theory and evaluation metrics for learning disentangled representations," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=HJgK0h4Ywr.

[23] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[24] Y. Yang, X. Zhang, Q. Guan, and Y. Lin, "Making invisible visible: Data-driven seismic inversion with spatio-temporally constrained data augmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022. DOI: 10.1109/TGRS.2022.3144636.

[25] P. Y. Lu, S. Kim, "Extracting interpretable physical parameters from spatiotemporal systems using unsupervised learning," *Phys. Rev. X*, vol. 10, p. 031 056, 3 2020. DOI: 10.1103/PhysRevX.10.031056. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

[26] A. Desai, C. Freeman, Z. Wang, and I. Beaver, *Timevae: A variational auto-encoder for multivariate time series generation*, 2021. DOI: 10.48550/ARXIV.2111.08095. [Online]. Available: https://arxiv.org/abs/2111.08095.

[27] F. Lanusse, R. Mandelbaum, S. Ravanbakhsh, C.-L. Li, P. Freeman, and B. Póczos, "Deep generative models for galaxy image simulations," *Monthly Notices of the Royal Astronomical Society*, vol. 504, no. 4, pp. 5543–5555, 3 Jul. 2021, 14 pages, submitted to MNRAS. Comments most welcome. DOI: 10.1093/mnras/stab1214. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03451832.

[28] N. Takeishi and A. Kalousis, "Variational autoencoder with differentiable physics engine for human gait analysis and synthesis," in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, vol. 10, American Physical Society, 2021, p. 031 056. DOI: 10.1103/PhysRevX.10.031056. [Online]. Available: https://openreview.net/forum?id=9ISlKio3Bt.

[29] M. A. Aragon-Calvo, "Self-supervised learning with physics-aware neural networks – i. galaxy model fitting," *Monthly Notices of the Royal Astronomical Society*, vol. 498, pp. 3713–3719, 3 2020. DOI: 10.1103/PhysRevX.10.031056. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

[30] N. Takeishi and A. Kalousis, "Physics-integrated variational autoencoders for robust and interpretable generative modeling," *CoRR*, vol. 10, p. 031 056, 3 2021. DOI: 10.1103/PhysRevX.10.031056. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

[31] O. Rybkin, K. Daniilidis, and S. Levine, *Simple and effective vae training with calibrated decoders*, 2020. DOI: 10.1103/PhysRevX.10.031056. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

[32] R. Yu, *A tutorial on vaes: From bayes' rule to lossless compression*, 2020. DOI: 10.48550/ARXIV.2006.10273. [Online]. Available: https://arxiv.org/abs/2006.10273.

[33] G. Dorta, S. Vicente, L. Agapito, N. D. Campbell, and I. Simpson, "Structured uncertainty prediction networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 10, American Physical Society, 2018, pp. 5477–5485. DOI: 10.1103/PhysRevX.10.031056. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

[34] T. Koike-Akino and Y. Wang, "Autovae: Mismatched variational autoencoder with irregular posterior-prior pairing," in *2022 IEEE International Symposium on Information Theory (ISIT)*, vol. 10, American Physical Society, 2022, pp. 1689–1694. DOI: 10.1109/ISIT50566.2022.9834769. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

[35] A. Srivastava and C. Sutton, "Autoencoding variational inference for topic models," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, vol. 10, OpenReview.net, 2017, p. 031 056. DOI: 10.1103/PhysRevX.10.031056. [Online]. Available: https://openreview.net/forum?id=BybtVK9lg.

[36] F. J. Kriegler, W. A. Malila, R. F. Nalepka, and W. Richardson, "Preprocessing Transformations and Their Effects on Multispectral Recognition," in *Remote Sensing of Environment, VI*, vol. 10, American Physical Society, Jan. 1969, p. 97. DOI: 10.1103/PhysRevX.10.031056. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

[37] W. Zhu, Y. Pan, H. He, L. Wang, M. Mou, and J. Liu, "A changing-weight filter method for reconstructing a high-quality ndvi time series to preserve the integrity of vegetation phenology," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 4, pp. 1085–1094, 3 2012. DOI: 10.1109/TGRS.2011.2166965. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

[38] M. Hall-Beyer, "Comparison of single-year and multiyear ndvi time series principal components in cold temperate biomes," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 11, pp. 2568–2574, 3 2003. DOI: 10.1109/TGRS.2003.817274. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

[39] E. F. Berra, R. Gaulton, and S. Barr, "Commercial off-the-shelf digital cameras on unmanned aerial vehicles for multitemporal monitoring of vegetation reflectance and ndvi," *IEEE transactions on geoscience and remote sensing*, vol. 55, no. 9, pp. 4878–4886, 3 2017. DOI: 10.1103/PhysRevX.10.031056. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

[40] X. Yang, J. Mustard, J. Tang, and H. Xu, "Regional-scale phenology modeling based on meteorological records and remote sensing observations," *Journal of Geophysical Research*, vol. 117, p. 031 056, 3 2012. DOI: 10.1103/PhysRevX.10.031056. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

[41] X. Zhang, M. A. Friedl, C. B. Schaaf, A. H. Strahler, J. C. Hodges, F. Gao, B. C. Reed, and A. Huete, "Monitoring vegetation phenology using MODIS," *Remote Sensing of Environment*, vol. 84, no. 3, pp. 471–475, 3 2003. DOI: 10.1103/PhysRevX.10.031056. [Online].

Available: https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

[42] L. Zeng, B. D. Wardlow, D. Xiang, S. Hu, and D. Li, "A review of vegetation phenological metrics extraction using time-series, multispectral satellite data," *Remote Sensing of Environment*, vol. 237, p. 111 511, 3 2020, ISSN: 0034-4257. DOI: 10.1016/j.rse.2019.111511. [Online]. Available: https://doi.org/10.1016/j.rse.2019.111511.

[43] Y. Zérah, S. Valero, and J. Inglada, *Sentinel-2 time series for pheno-vae*, Nov. 2022. DOI: 10.5281/zenodo.7273500. [Online]. Available: https://doi.org/10.5281/zenodo.7273500.

[44] J. Inglada, A. Vincent, M. Arias, B. Tardy, D. Morin, and I. Rodes, "Operational High Resolution Land Cover Map Production at the Country Scale Using Satellite Image Time Series," *Remote Sensing*, vol. 9, no. 1, p. 031 056, 3 2017, ISSN: 2072-4292. DOI: 10.3390/rs9010095. [Online]. Available: https://www.mdpi.com/2072-4292/9/1/95.

[45] X. Gao, J. M. Gray, and B. J. Reich, "Long-term, medium spatial resolution annual land surface phenology with a bayesian hierarchical model," *Remote Sensing of Environment*, vol. 261, p. 112 484, 3 2021. DOI: 10.1103/PhysRevX.10.031056. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

[46] M. D. Homan and A. Gelman, "The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo," *J. Mach. Learn. Res.*, vol. 15, no. 1, 1593–1623, 3 2014, ISSN: 1532-4435. DOI: 10.1103/PhysRevX.10.031056. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

[47] D. Phan, N. Pradhan, and M. Jankowiak, *Composable effects for flexible and accelerated probabilistic programming in numpyro*, 2019. DOI: 10.48550/ARXIV.1912.11554. [Online]. Available: https://arxiv.org/abs/1912.11554.

[48] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman, "Pyro: Deep universal probabilistic programming," *Journal of Machine Learning Research*, vol. 20, no. 28, pp. 1–6, 3 2019. DOI: 10.1103/PhysRevX.10.031056. [Online]. Available: http://jmlr.org/papers/v20/18-403.html.

[49] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, 3 2021, ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2021.05.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253521001081.

[50] R. Ak, V. Vitelli, and E. Zio, "An interval-valued neural network approach for uncertainty quantification in short-term wind speed prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2787–2800, 3 2015. DOI: 10.1109/TNNLS.2015.2396933. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

[51] Z. Zheng, L. Wang, L. Yang, and Z. Zhang, "Generative probabilistic wind speed forecasting: A variational recurrent autoencoder based method," *IEEE Transactions on Power Systems*, vol. 37, no. 2, pp. 1386–1398, 3 2022. DOI: 10.1109/TPWRS.2021.3105101. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

APPENDIX A
S2 DATA-SET

Label	Percentage in data-set
Continuous Urban Fabric	0.6%
Discontinuous Urban Fabric	4.1%
Industrial and Commercial Units	3.1%
Road Surfaces	0.3%
Rapeseed	4.5%
Straw Cereals	9.9%
Protein Crops	2.5%
Soy	7.2%
Sunflower	33.0%
Corn	5.8%
Roots	0.2%
Intensive Grasslands	3.4%
Orchards	0.6%
Vineyards	1.8%
Broad-leaved Forests	6.7%
Coniferous Forests	5.5%
Grasslands	5.5%
Woody Moorlands	2.3%
Bare Rock	0.1%
Water Bodies	2.8%

TABLE VII
DISTRIBUTION OF THE LAND COVER CLASSES COMPOSING THE SENTINEL-2 TIME SERIES DATA-SET. THE CLASS LEGEND IS TAKEN FROM THE OSO [44] LAND COVER MAP PRODUCT.

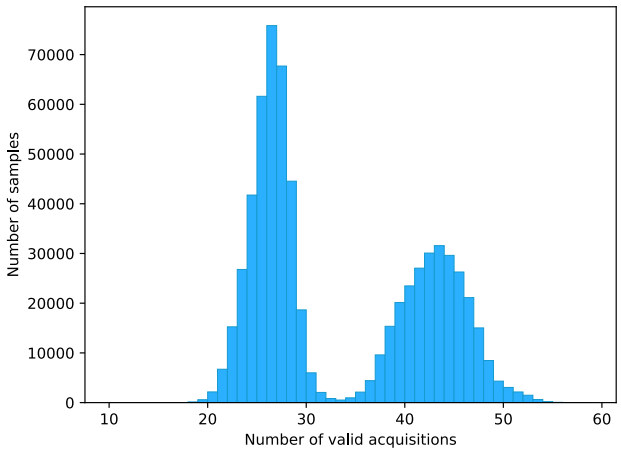


Fig. 12. Distribution of the temporal acquisitions composing the Sentinel-2 time series data-set.

APPENDIX B
RECONSTRUCTION OF S2 TIME SERIES

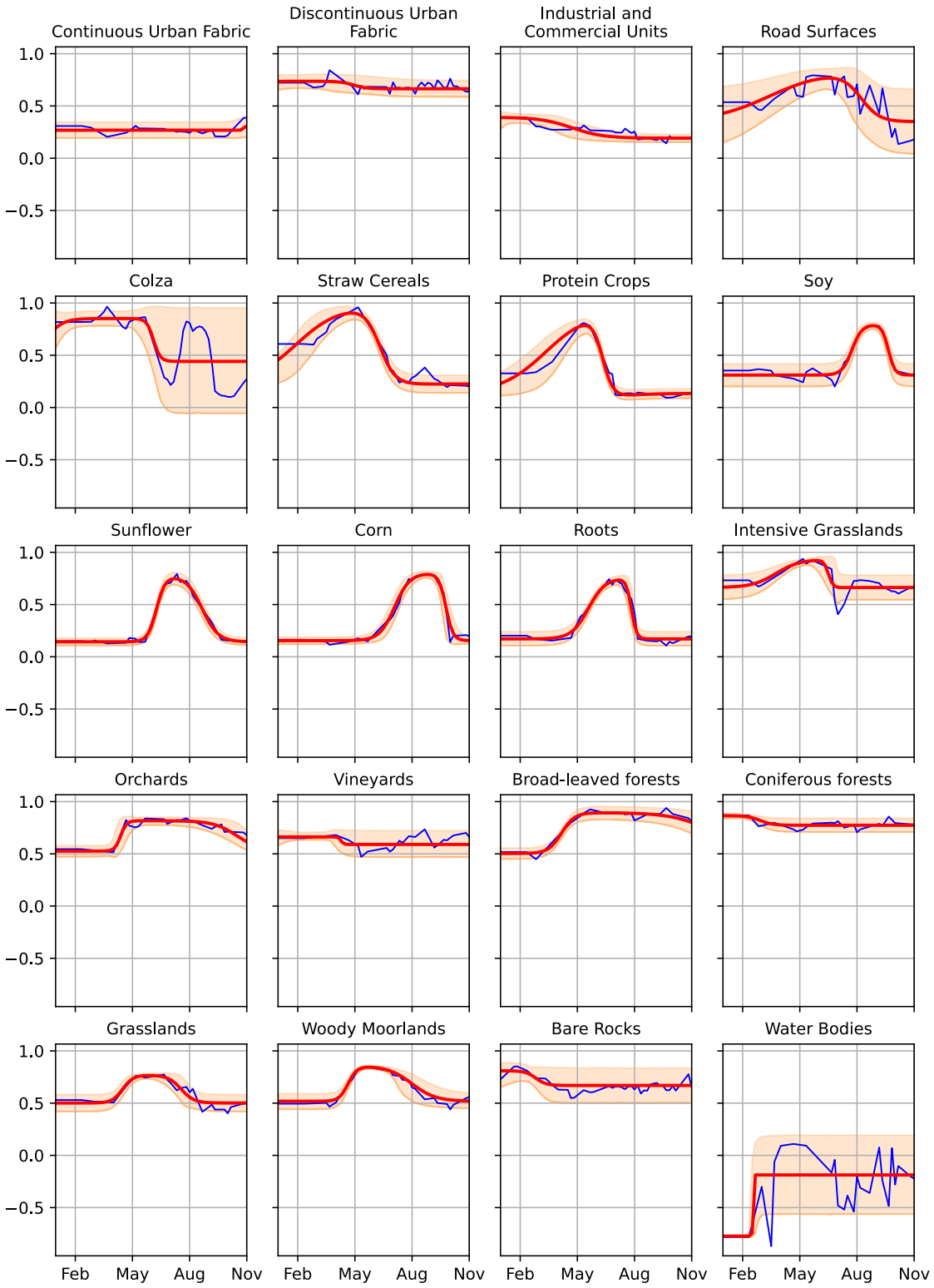


Fig. 13. Examples of reconstructions of Sentinel-2 NDVI time series with pheno-VAE trained on S2 data-set. Blue: 5-days interpolated S2 time series. Red: Reconstruction of the mode of phenological distribution. Orange: 5-95th centile interval.

APPENDIX C
PREDICTION INTERVAL PERFORMANCES

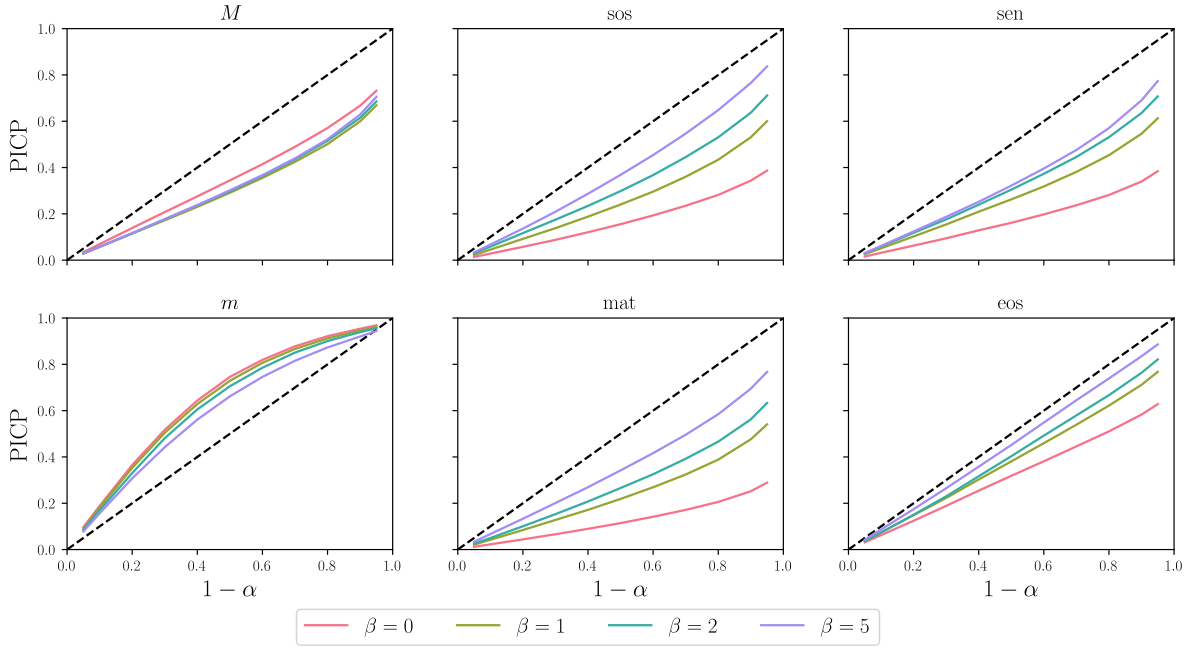


Fig. 14. PICP vs $1-\alpha$ for pheno-VAE trained on S2 Data-set, with various settings of the coefficient β of the KL loss term. The more β increases, the more the PICP increases at constant confidence level $1-\alpha$.

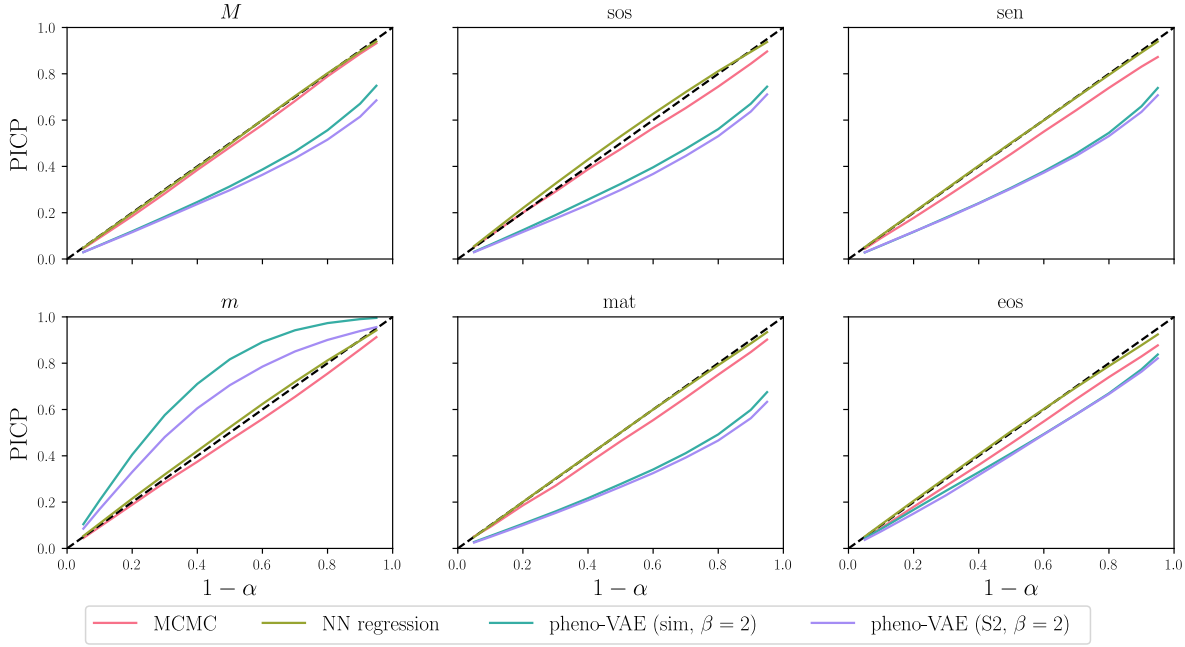


Fig. 15. PICP vs $1-\alpha$ for MCMC, Neural Network regression and pheno-VAE (with $\beta = 2$, trained on the S2 or simulated data-set.) The PICP curves of Neural Network regression and MCMC are very close to PICP=α for all α, while pheno-VAE underestimates uncertainty for all confidence levels, for all phenological variables, except for m where uncertainty is overestimated.

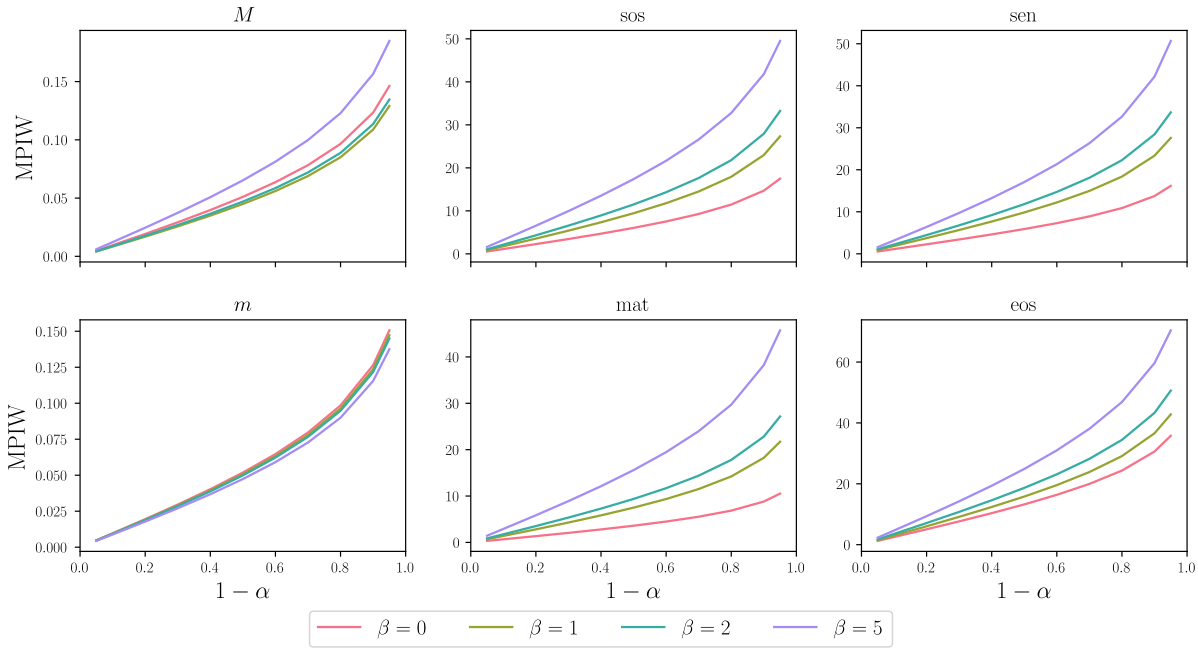


Fig. 16. MPIW vs $1-\alpha$ for pheno-VAE trained on S2 Data-set, with various settings of the coefficient β of the KL loss term. The more β increases, the more the MPIW increases at constant confidence level $1-\alpha$.

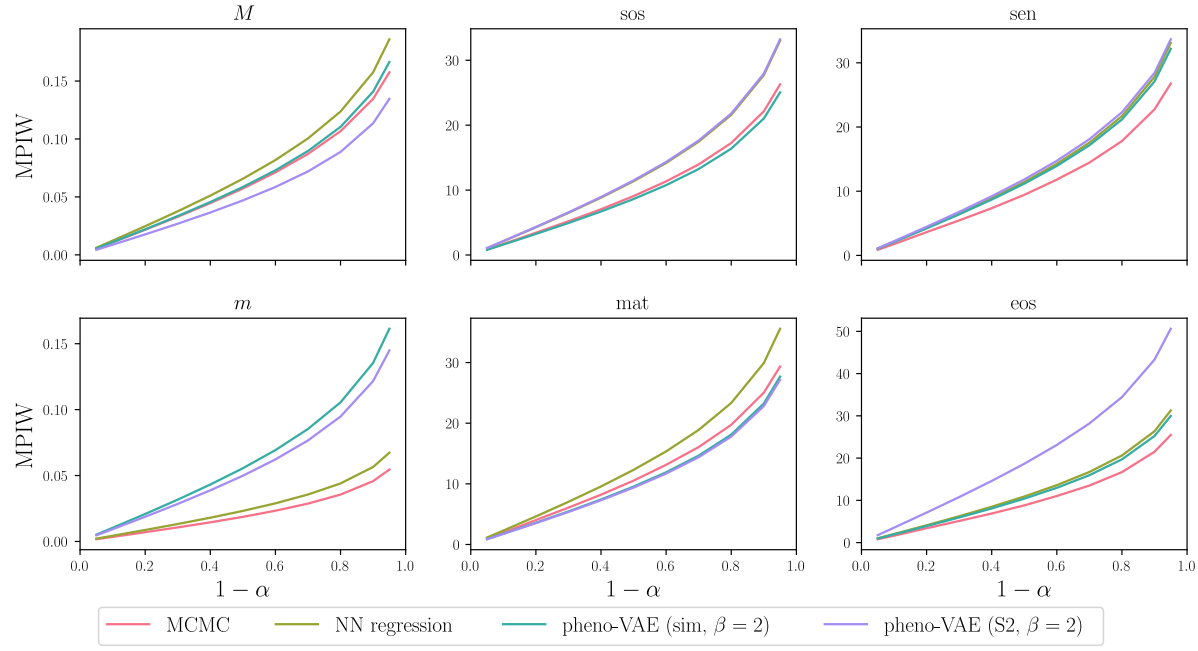


Fig. 17. MPIW vs $1-\alpha$ for MCMC, Neural Network regression and pheno-VAE (with $\beta=2$, trained on the S2 or simulated data-set.) prediction interval sizes are similar for all methods, except for m, where prediction intervals are larger for pheno-VAE, and for the eos of $pheno-VAE$ trained on the S2 data-set, that also has larger prediction intervals.

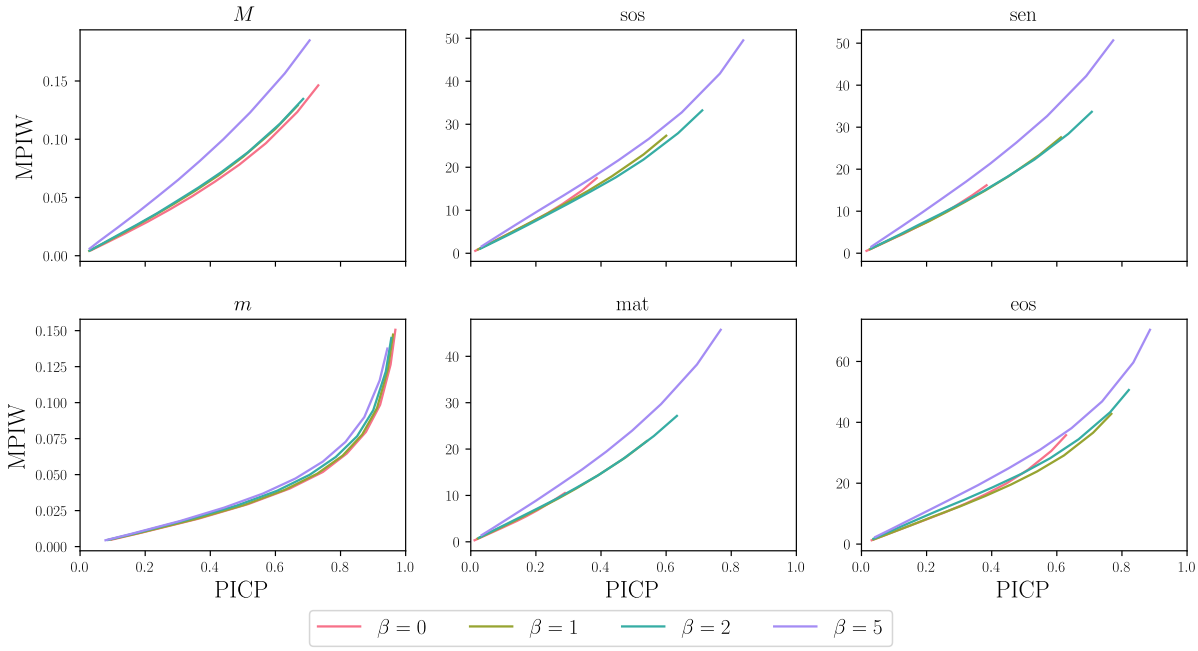


Fig. 18. MCWI vs PICP for pheno-VAE trained on S2 Data-set, with various settings of the coefficient β of the KL loss term. The larger β is, the closer to 1 the PICP is able to get, but also the larger the MCWI is getting at constant PICP.

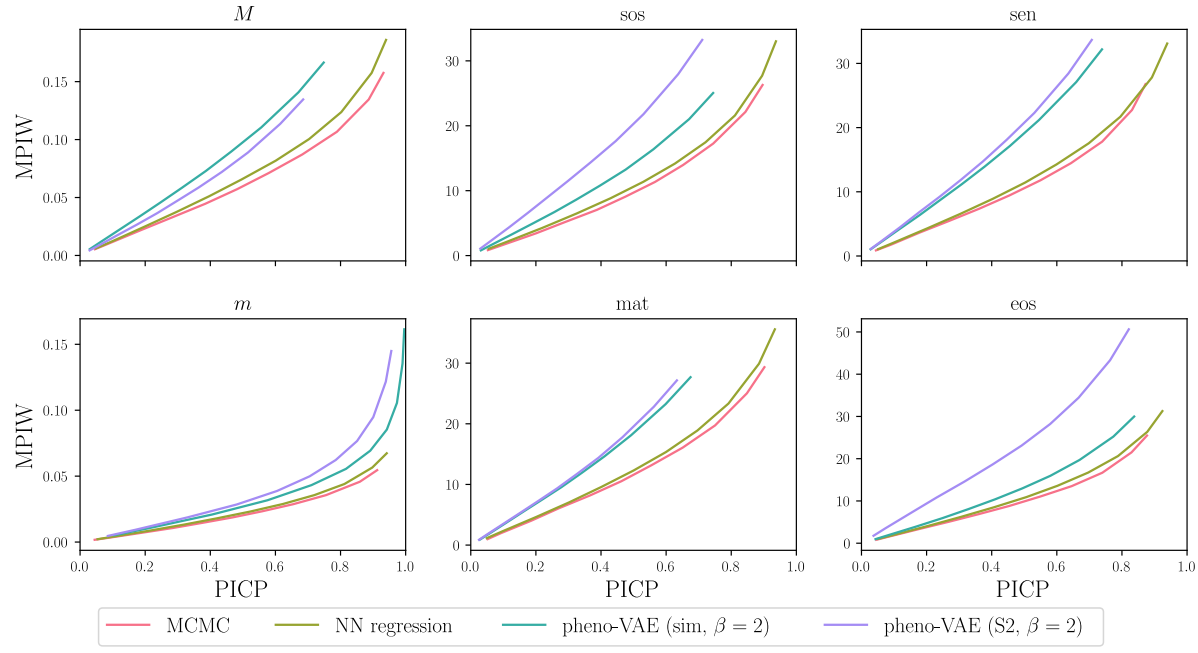


Fig. 19. MCWI vs PICP for MCMC, Neural Network regression and pheno-VAE (with $\beta = 2$, trained on the S2 or simulated data-set.). MCMC and Neural Network regression have a PICP that can be almost all possible values possible, between 0 and 1. Pheno-VAE for both data-sets cannot have high PICP for most phenological variables, and have higher MCWI than MCMC and Neural Network regression at similar PICP.

APPENDIX D
DENSITY OF MAXIMUM OF CONTINUOUS DISTRIBUTIONS

Let Y be the maximum of n independent continuous random variables X_i. The CDF of Y is:

$$\begin{aligned} F_Y(y) &= P(Y < y) \\ &= P\left(\max_{i\in[[1,n]]} X_i < y\right) \\ &= P\left(\bigcap_{i=1}^{n} (X_i < y)\right) \\ &= \prod_{i=1}^{n} P\left(X_i < y\right) \\ &= \prod_{i=1}^{n} F_{X_i}(y) \end{aligned} \tag{28}$$

The log-derivative of the CDF of Y yields:

$$\begin{aligned} \frac{d\ln F_Y}{dy}(y) &= \frac{d}{dy}\ln\left(\prod_{i=1}^{n} F_{X_i}(y)\right) \\ &= \frac{d}{dy}\sum_{i=1}^{n}\ln\left(F_{X_i}(y)\right) \\ &= \sum_{i=1}^{n}\frac{d}{dy}\ln\left(F_{X_i}(y)\right) \\ &= \sum_{i=1}^{n}\frac{dF_{X_i}(y)}{dy}\frac{1}{F_{X_i}(y)} \\ &= \sum_{i=1}^{n} f_{X_i}(y)\frac{1}{F_{X_i}(y)} \end{aligned} \tag{29}$$

Finally, using the log-derivative of the CDF of Y enables deriving its PDF as a function of the PDFs and CDFs of X_i:

$$\begin{aligned} f_Y(y) &= \frac{dF_Y}{dy}(y) \\ &= F_Y(y)\frac{d\ln F_Y}{dy}(y) \\ &= \prod_{i=1}^{n} F_{X_i}(y)\sum_{i=1}^{n} f_{X_i}(y)\frac{1}{F_{X_i}(y)} \end{aligned} \tag{30}$$

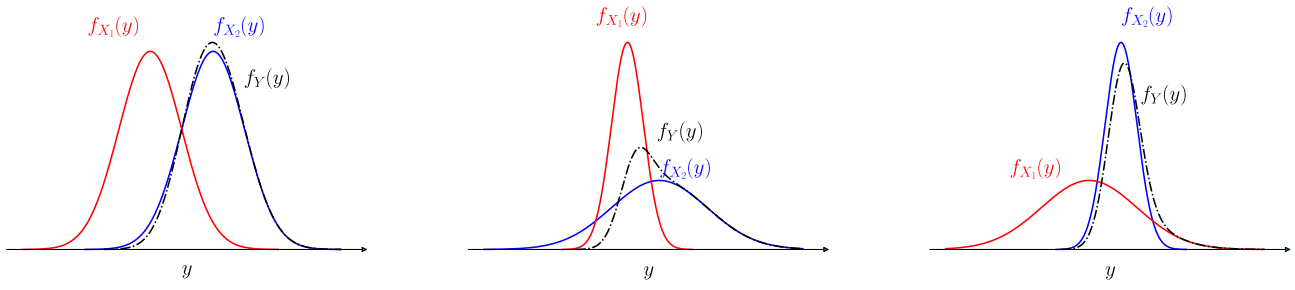


Fig. 20. Examples of distribution of the maximum Y of two Gaussian variables X_1 and X_2.

APPENDIX E
KL-DIVERGENCE OF TRUNCATED GAUSSIANS AND UNIFORM DISTRIBUTIONS

Let:

$$p \sim \mathcal{TN}(\mu, \sigma, a, b), \qquad q \sim \mathcal{U}(a, b) \tag{31}$$

with the truncated Gaussian density:

$$p(x) = \frac{\psi(\frac{x-\mu}{\sigma})}{\sigma\eta}, \qquad \psi(x) = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}$$

with

$$\eta = \Psi(\tilde{b}) - \Psi(\tilde{a}), \qquad \tilde{a} = \frac{a-\mu}{\sigma}, \qquad \tilde{b} = \frac{b-\mu}{\sigma}$$

and standard Gaussian CDF:

$$\Psi(x) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$$

KL divergence is then:

$$\begin{aligned} \mathbb{KL}(p(x)\|q(x)) &= \int_a^b p(x) \ln \frac{p(x)}{q(x)} dx \\ &= \int_a^b p(x) \ln p(x) dx - \int_a^b p(x) \ln q(x) dx \end{aligned} \tag{32}$$

Its second term is:

$$\begin{aligned} \int_a^b p(x) \ln q(x) dx &= \int_a^b p(x) \ln \frac{1_{[a,b]}}{b-a} dx \\ &= -\ln(b-a) \int_a^b p(x) dx \\ &= -\ln(b-a) \end{aligned} \tag{33}$$

The first term is:

$$\begin{aligned} \int_a^b p(x) \ln p(x) dx &= \int_a^b p(x) \ln \frac{\psi(\frac{x-\mu}{\sigma})}{\sigma\eta} dx \\ &= -\ln(\sigma\eta) \int_a^b p(x) dx + \int_a^b p(x) \ln \psi\left(\frac{x-\mu}{\sigma}\right) dx \\ &= -\ln(\sigma\eta) + \int_a^b p(x) \ln \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}} dx \\ &= -\ln(\sigma\eta) - \frac{1}{2}\ln(2\pi) - \int_a^b p(x) \frac{(x-\mu)^2}{2\sigma^2} dx \\ &= -\ln(\sigma\eta) - \frac{1}{2}\ln(2\pi) - \frac{1}{2\sigma^2} \int_a^b p(x) \left(x^2 - 2\mu x + \mu^2\right) dx \\ &= -\ln(\sigma\eta) - \frac{1}{2}\ln(2\pi) - \frac{\mu^2}{2\sigma^2} - \frac{1}{2\sigma^2} \int_a^b x^2 p(x) dx + \frac{\mu}{\sigma^2} \int_a^b x p(x) dx \\ &= -\ln(\sigma\eta) - \frac{1}{2}\ln(2\pi) - \frac{\mu^2}{2\sigma^2} - \frac{1}{2\sigma^2}\left\langle p^2 \right\rangle + \frac{\mu}{\sigma^2}\langle p \rangle \end{aligned} \tag{34}$$

with truncated Gaussian moments:

$$\left\langle p^2 \right\rangle = \sigma^2 + \frac{\sigma^2}{\eta}\left(\tilde{a}\psi(\tilde{a}) - \tilde{b}\psi(\tilde{b})\right) + \mu^2 + \frac{2\mu\sigma}{\eta}\left(\psi(\tilde{a}) - \psi(\tilde{b})\right)$$

$$\langle p \rangle = \mu + \frac{\sigma}{\eta}\left(\psi(\tilde{a}) - \psi(\tilde{b})\right)$$

Finally:

$$\mathbb{KL}(p(x)\|q(x)) = -\frac{1}{2} - \frac{1}{2}\ln(2\pi) - \ln(\sigma\eta) - \frac{\tilde{a}\psi(\tilde{a}) - \tilde{b}\psi(\beta)}{2\eta} + \ln(b-a)$$

APPENDIX F
NOTATIONS

A. Variables notations

Bold font denotes a vector or a matrix variable. Variables with a hat denote an estimated quantity. Underlined variables are rectified variables. Indexing with i denotes a dimension of latent variables, and indexing with j denotes an element of a data-set.

Notation	Definition
$\mathbf{x}$	Observation, input data
$\hat{\mathbf{x}}$	Reconstruction of input data
$\mathbf{z}$	Latent variable
$\underline{z}$	Rectified latent variable
$\boldsymbol{\lambda}$	Parameter of variational distribution
ϕ	Parameters of encoder's neural network
θ	Parameters of decoder's neural network
$1-\alpha$	Confidence level
β	Coefficient on the KL term in the ELBO used in β-VAE
$\boldsymbol{\mu}_{\mathbf{z}}$	Mean parameter of Gaussian latent space
$\underline{\mu}_{z_i}$	Rectified mean of Gaussian latent distribution
$\boldsymbol{\Sigma}_{\mathbf{z}}$	Covariance matrix of Gaussian latent space
$\boldsymbol{\mu}_{\hat{\mathbf{x}}}$	Mean parameter of Gaussian decoder distribution
$\boldsymbol{\Sigma}_{\hat{\mathbf{x}}}$	Covariance matrix of Gaussian decoder distribution
K	Number of latent samples drawn to estimate the decoder's output distribution parameters
Δ_z	difference between two consecutive latent variables
ρ	Reflectance
$\mathcal{F}$	General notation for user defined decoder
$\Omega_{\mathbf{z}}$	Double-sigmoid function parametrized by $\mathbf{z}$
a, b	Bounds of a variational distribution support
l, u	prediction interval bounds
N	Number of samples in test data-set
S	Sigmoid function
ψ	Gaussian PDF
Ψ	Gaussian CDF
$\mathcal{U}$	Uniform distribution
$\mathcal{N}$	Gaussian distribution
$\mathcal{TN}$	Truncated Gaussian distribution
$\mathbb{KL}$	Kullback-Leibler divergence
$\mathbb{E}$	Expectation
$\#$	Cardinality
$\emptyset$	Empty set

B. Acronyms

Acronym	Definition
ELBO	Evidence Lower BOund
KL	Kullback-Leibler (Divergence)
VAE	Variational Autoencoder
PDF	Probability Density Function
CDF	Cumulative Density Function
NDVI	Normalized Difference Vegetation Index
MAE	Mean Average Error
MPIW	Mean Prediction Interval Width
PICP	Prediction Interval Coverage Probability