




# SUPERNOVAE: VAE BASED KERNEL-PCA FOR ANALYSIS OF SPATIO-TEMPORAL EARTH DATA

Xavier-Andoni Tibau<sup>1,2</sup>, Christian Requena-Mesa<sup>1,2,3</sup>, Christian Reimers<sup>1,2</sup>, Joachim Denzler<sup>2,4</sup>,  
 Veronika Eyring<sup>5,6</sup>, Markus Reichstein<sup>3,4</sup>, Jakob Runge<sup>1</sup>

**Abstract**—It is a constant challenge to better understand the underlying dynamics and forces driving the Earth system. Advances in the field of deep learning allow for unprecedented results, but use of these methods in Earth system science is still very limited. We present a framework that makes use of a convolutional variational autoencoder as a learnable kernel from which to extract spatio-temporal dynamics via PCA. The method promises the ability of deep learning to digest highly complex spatio-temporal datasets while allowing expert interpretability. Preliminary results over two artificial datasets, with chaotic and stochastic temporal dynamics, show that the method can recover a latent driver parameter while baseline approaches cannot. While further testing on the limitations of the method is needed and experiments on real Earth datasets are in order, the present approach may contribute to further the understanding of Earth datasets that are highly non-linear.

## I. INTRODUCTION AND MOTIVATION

In recent decades, the volume of Earth observations and Earth system model simulations has substantially increased. Yet, harvesting knowledge from such abundant and complex data is a difficult task. Finding and studying spatio-temporal patterns is one of the principal goals of the climate community.

First attempts in this direction came with the introduction of Empirical Orthogonal Functions (EOFs) [1]. While several limitations of EOFs [2], e.g., the modes are not orthogonal, are met by approaches such as Rotated EOF (REOF) [3][4], some limitations remain, e.g. only linear modes can be observed. Earth system dynamics are, however, often non-linear [5][6], hence, Kernel-PCA [7] can offer some explanatory power. The feature function maps the original data into a new feature space


where some non-linear dependencies of the original data become linear. However, choosing the best kernel function among the zoo of existing kernel functions is difficult and important [8].

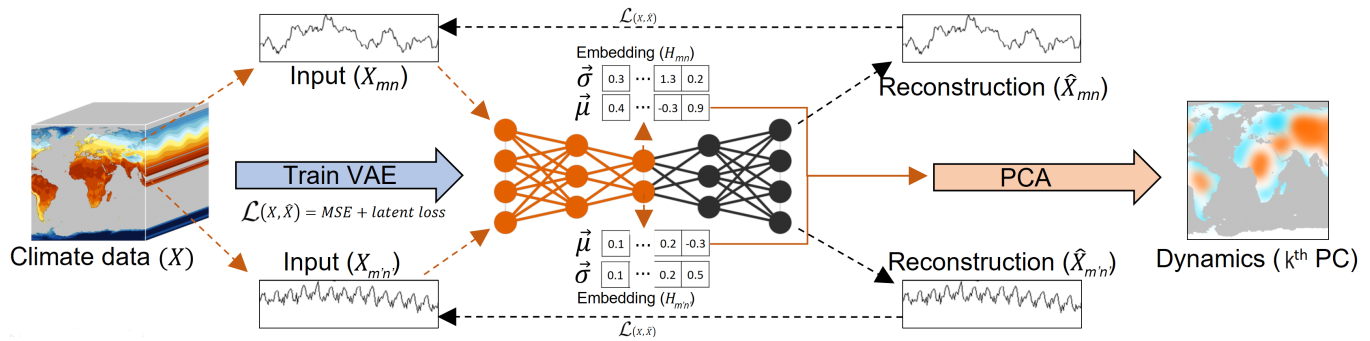
Deep learning is an extremely active research area that shows huge success in a broad area of applications [9]. We use a deep convolutional variational autoencoder (VAE) [10] to unsupervisedly approximate a useful feature function and therefore, indirectly a kernel function to further extract dynamical, non-linear patterns from Earth system data. It has been shown that VAEs are able to produce efficient high-level representations from complex inputs [10]. The original data is projected into a new abstract space where each dimension represents a higher order feature. Performing a PCA over the projected data results in a decomposition of the main dynamics driving the dataset. Kernel-PCA is one of the most polyvalent dimensionality reduction techniques [11], we paired it with the flexibility and power for encoding high order features of VAEs. Other authors have used autoencoders to approximate kernel functions [12]; however, the novelty of the present approach is to use the abstract representation of the VAE to represent latent dynamics through the PCs. Note that, despite the similarity, the resulting PCs of our method are no direct improvement of EOFs or REOFs since the PCA in our approach is performed over an abstract space of higher order features.

We present an overview of our SupernoVAE approach and preliminary results over two simulated datasets with chaotic and stochastic temporal dynamics driven by a latent space dependent parameter. SupernoVAE can recover the underlying parameter from the datasets while EOF, REOF, and kernel-PCA using some standard kernel functions cannot.

## II. METHODS AND NOTATION

**SupernoVAE.** Figure 1 summarizes the SupernoVAE workflow. Let  $X$  be the input-data in the input space  $\mathcal{X}$ . First, the data  $X$  is used to train a VAE that learns two functions. One function, the encoder, maps every  $X_{mn} \in X$  onto a distribution over the feature space  $\mathcal{H}$ ,

Corresponding author: X-A. Tibau, xavier.tibau@dlr.de <sup>1</sup>Climate Informatics Group, Institute of Data Science, German Aerospace Center (DLR), Jena, Germany. <sup>2</sup>Computer Vision Group, Friedrich-Schiller-Universität Jena, Germany <sup>3</sup>Max-Planck-Institute for Biogeochemistry, Jena, Germany. <sup>4</sup>Michael Stifel Center Jena for data-driven and simulation science. <sup>5</sup>German Aerospace Center (DLR), Institute for Atmospheric Physics, Oberpfaffenhofen, Germany <sup>6</sup>University of Bremen, Institute of Environmental Physics, Bremen, Germany.  These authors contributed equally to this work.



**Fig. 1:** SupernoVAE workflow: As explained in Sec. II, in a first step, a VAE is trained with the climate data. In the second step (orange lines), the encoder part of the VAE is used to compute  $H$  and the PCs of it.

the other one, the decoder, maps a sample drawn from this latent distribution back to  $\mathcal{X}$ . Both functions are trained to combine to the identity. The ratio between the temporal dimensions in  $\mathcal{X}$  and the dimensions of the feature space  $\mathcal{H}$  is given by the hyper-parameter  $\theta$ . If  $\theta = 1.00$  the entire temporal series could be passed into the embedding and the VAE would not learn any temporal dynamics. To prevent this, variational autoencoders are trained with a latent loss that penalizes them for choosing the covariance of the latent distribution far of the identity matrix. Therefore, the encoder and decoder can not learn the identity since the sample, drawn for reconstruction, might be far off the mean of the latent distribution.

For our tests, the encoder network consists of 6 one-dimensional convolutional layers, performing convolutions in the time dimension, alternating with batch normalization layers. The last convolutional layer is followed by a fully-connected layer mapping to a mean and a covariance of a distribution over  $\mathcal{H}$ . The architecture of the decoder mirrors the encoder. In a second step, PCA is performed over the means  $H$  of the latent distributions corresponding to  $X$ . Each  $H_{mn} \in H$  is an abstract representation of higher order features of  $X_{mn} \in X$ , thus, the first PCs of  $H$  depict the main underlying dynamics that drive  $X$  and are laid out for expert interpretation.

**Kernel PCA** SupernoVAE is a Kernel-PCA using a learned kernel function. To apply PCA to inputs  $x_i$ , we first calculate the covariance matrix  $(c_{ij}) = \langle x_i, x_j \rangle$ , do an eigenvalue decomposition and then project the inputs onto the space spanned by the first  $k$  eigenvectors. In a Kernel-PCA, we substitute the scalar product by a kernel function  $k : X \times X \rightarrow \mathbb{R}$ . The resulting Gram matrix is used instead of the covariance matrix. From the representer theorem [7] we know that every kernel function can be expressed as the concatenation of a scalar product and a feature function

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle.$$

Therefore, applying kernel PCA is the same as applying the feature function to every example and applying regular PCA. In the SupernoVAE framework the encoder-part of the autoencoder works as the feature function  $\phi : X \rightarrow H$ . Hence, one can think of SupernoVAE as an unsupervised method to learn a kernel function for a kernel PCA.

**Datasets.** Two toy datasets were generated to test this approach. Both have two spatial dimensions  $m \in \{1, \dots, 360\}$ ,  $n \in \{1, \dots, 180\}$  and one temporal dimension  $t \in \{1, \dots, 696\}$ . Let  $\mathcal{F}$  be a real valued matrix (Fig. 2.a) which will represent the latent driver parameter in the datasets. The first dataset, in the following referred to as *Lorenz '96*, is created by computing a ten-dimensional Lorenz dynamical system [13] with dynamics

$$\frac{dx_i^{mn}}{dt} = (x_{i+1}^{mn} - x_{i-2}^{mn})x_{i-1}^{mn} - x_i^{mn} + \mathcal{F}_{mn}$$

and assuming that only the first variable  $x_1$  is measured in analogy to the real climate system where not all physically relevant variables can be measured. The Lorenz '96 can be considered as a simple model for chaotic spatio-temporal weather dynamics. The second dataset, here referred to as *Cellular automaton*, is a variant of the model proposed by von Neumann [14]. The value at each position in the grid rises at every timestep until a threshold is exceeded and the value is set to 0. The probability to start rising again afterwards depends on the values of its *Moore neighbors* in the grid. There is an oscillatory hyper-parameter that increases the chance of starting to rise and introduces the possibility of becoming 0 prior to reaching the threshold. The effect of this parameter is smoothed out by  $\mathcal{F}$ . That is,  $\mathcal{F}_{mn}$  regulates the inherent stochasticity of  $X_{mn}$ . This can be understood as a very simple model of a forest area with  $\mathcal{F}$  representing different soil properties and  $X$  representing the biomass of each tree.

As a result, dynamics in both datasets vary across locations  $(m, n)$  and lead to chaotic in one and chaotic

**TABLE I** Summary of results. The column *Reconstruction* shows the coefficient of determination of the VAE reconstructions and the input time series, the columns  $1^{st}$ ,  $2^{nd}$  and  $3^{rd}$  PC show the coefficient of determination between the  $k^{th}$  principal component and the forcing pattern  $\mathcal{F}$ . The coefficient of determination for the baseline methods were smaller than 0.00114, see Table II. The PC marked \* is represented in Fig. 2 (c).

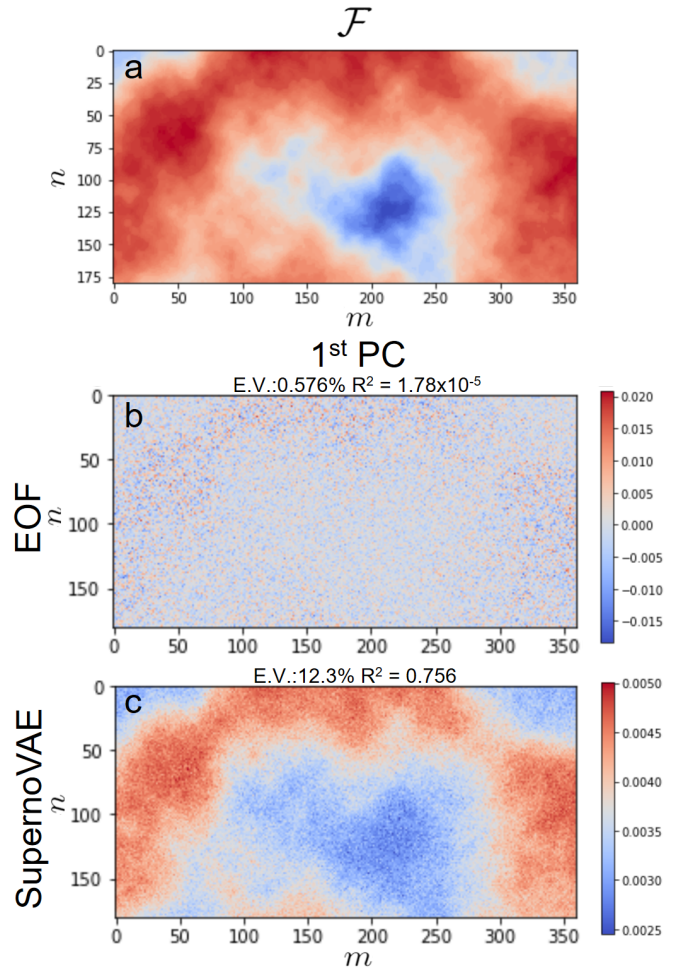
$\theta$	Lorenz '96				Cellular automata			
	Reconstruction	$1^{st}$ PC	$2^{nd}$ PC	$3^{rd}$ PC	Reconstruction	$1^{st}$ PC	$2^{nd}$ PC	$3^{rd}$ PC
0.01	0.129	0.000	0.001	0.001	0.513	0.009	0.002	0.005
0.10	0.643	0.476	0.000	0.022	0.968	0.627	0.003	0.005
1.00	0.865	0.756*	0.851	0.001	0.981	0.287	0.005	0.767
	<i>time-permuted</i>				<i>time-permuted</i>			
0.10	0.446	0.001	0.000	0.000	0.939	0.598	0.027	0.007
1.00	0.988	0.006	0.000	0.000	0.997	0.397	0.328	0.074

and stochastic behavior in the other dataset. To ensure that the results are caused by the different dynamics and not by statistical properties we created for each dataset a *time-permuted* version where the values of time series at each location have been independently randomly permuted. Finally, each time series was normalized by subtracting its mean and dividing it by its standard deviation.

**Experiments.** The VAE was trained for  $\theta \in \{0.01, 0.10, 1.00\}$  on both datasets, and for  $\theta \in \{0.10, 1.00\}$  on the time-permuted datasets. We expect to recover  $\mathcal{F}$  in one of the first principal components since it is the main source of variability in the datasets. The autoencoder is able to reconstruct the time series from the latent distribution corresponding to this time series. Hence the information on the driving parameter  $\mathcal{F}$  must be contained in the latent distribution. The correlation between  $\mathcal{F}$  and the first three components of each model were recorded. Analogously, the correlation of  $\mathcal{F}$  to the first components of EOF over the original data and off-the-shelf Kernel PCA using the suggested kernels of the Scipy-library [15] (Linear, Polynomial, RBF, Sigmoid and Cosine) and there standard hyper-parameters in Scipy-library were compared and used as a baseline. The Lorenz model is chaotic and  $\mathcal{F}$  corresponds to the amount of chaos in each time series. The cellular automaton is stochastic and  $\mathcal{F}$  corresponds to the amount of stochasticity in each time series. Both of these properties are hard to identify using off-the-shelf Kernels. Hence, all baselines failed to identify  $\mathcal{F}$ .

### III. RESULTS AND DISCUSSION

Table I summarizes the experimental results of SupernoVAE. It can be observed that there is a correlation between the quality of the reconstruction and the coefficient of determination between  $\mathcal{F}$  and the PCs.



**Fig. 2:** Plot of (a)  $\mathcal{F}$ , (b) the  $1^{st}$  PC for the Lorenz '96 dataset and (c) the  $1^{st}$  PC for Lorenz '96 after applying SupernoVAE. E.V.: Explained Variation by the first principal component.

This shows that the different temporal dynamics are learned by the autoencoder and mapped into the latent distributions if the dimensionality of the feature space allows for such information to be stored. The coefficient of determination for  $\mathcal{F}$  and the results of the baseline methods (EOF, Kernel-PCA) applied directly to  $X$  was smaller than 0.001, consequently, none of these approaches were able to find  $\mathcal{F}$ , ref. Table II.

**TABLE II** Summary of results. Coefficient of determination for the Baseline Kernel PCA and EOF methods between the  $k^{\text{th}}$  principal component and the forcing pattern  $\mathcal{F}$ .

Method	Lorenz '96			Cellular automata		
	1 <sup>st</sup> PC	2 <sup>nd</sup> PC	3 <sup>rd</sup> PC	1 <sup>st</sup> PC	2 <sup>nd</sup> PC	3 <sup>rd</sup> PC
EOF	$5.94 \cdot 10^{-6}$	$5.22 \cdot 10^{-8}$	$1.36 \cdot 10^{-7}$	$7.13 \cdot 10^{-5}$	$4.81 \cdot 10^{-6}$	$7.42 \cdot 10^{-5}$
Linear Kernel	$8.15 \cdot 10^{-4}$	$3.24 \cdot 10^{-4}$	$3.84 \cdot 10^{-5}$	$7.80 \cdot 10^{-5}$	$1.63 \cdot 10^{-6}$	$5.16 \cdot 10^{-5}$
Poly Kernel	$2.12 \cdot 10^{-4}$	$2.54 \cdot 10^{-4}$	$1.87 \cdot 10^{-6}$	$4.72 \cdot 10^{-4}$	$1.36 \cdot 10^{-4}$	$1.31 \cdot 10^{-5}$
RBF Kernel	$5.12 \cdot 10^{-4}$	$1.17 \cdot 10^{-5}$	$9.53 \cdot 10^{-5}$	$4.06 \cdot 10^{-4}$	$9.65 \cdot 10^{-5}$	$1.63 \cdot 10^{-4}$
Sigmoid Kernel	$9.42 \cdot 10^{-4}$	$2.48 \cdot 10^{-7}$	$3.83 \cdot 10^{-4}$	$6.65 \cdot 10^{-4}$	$1.13 \cdot 10^{-3}$	$1.13 \cdot 10^{-5}$
Cosine Kernel	$1.06 \cdot 10^{-3}$	$6.35 \cdot 10^{-5}$	$2.76 \cdot 10^{-5}$	$1.17 \cdot 10^{-4}$	$4.50 \cdot 10^{-5}$	$2.60 \cdot 10^{-5}$

For the two datasets, the VAE finds meaningful features and temporal dynamics such that the time series can be reconstructed despite the noise introduced by sampling from the latent distribution. However, when the time series is permuted, there are no temporal patterns and the network is not able to learn higher-order features. Therefore the only way to decrease the reconstruction loss is to learn the identity even if that implies a high latent loss created by pushing the covariance towards zero. The correlation found between the principal components on the embeddings of the time-permuted Lorenz '96 is less than 0.001, while the reconstruction in both datasets is better for the time-permuted model corroborating this explanation.

In the cellular automaton dataset, the latent forcing  $\mathcal{F}$  affects both the temporal dynamics and the data distribution. Permuting in time does erase the temporal dynamics, but the data distribution is still hinting at the latent forcing  $\mathcal{F}$ . Hence the forcing pattern is correlated to the PCs in this model. These preliminary results show that SupernoVAE is capable of recovering latent forcing from the temporal dynamics that otherwise might go unnoticed by the baseline methods.

#### IV. CONCLUSIONS

We introduced SupernoVAE and demonstrated in two toy examples that it was capable of finding the driving parameter of dynamics in contrast to the tested baseline methods. We showed that SupernoVAE captured time dynamics and not just distribution information. The proposed method allows the clustering of temporal dynamics in non-linear Earth datasets. Since the core of the method is based on a convolutional neural network, it scales very well in the number of input variables. This allows for using multivariate inputs. Besides, it can be extended to account for data in three spatial dimensions by using 3D convolutions and can adopt other neural network architectures like RNN-LSTM

mechanics. Even though the work presented here is only a proof of concept and further research is needed to reach substantial results in climate science, there are reasons to believe that the method is capable of finding unknown non-linear latent dynamics underlying climate data.

#### REFERENCES

- [1] E. N. Lorenz, "Empirical orthogonal functions and statistical weather prediction," *Science Report 1*, 1956.
- [2] D. Dommenges and M. Latif, "A cautionary note on the interpretation of eofs," *Journal of Climate*, vol. 15, no. 2, pp. 216–225, 2002.
- [3] H. F. Kaiser, "The varimax criterion for analytic rotation in factor analysis," *Psychometrika*, vol. 23, no. 3, pp. 187–200, 1958.
- [4] M. B. Richman, "Rotation of principal components," *International Journal of Climatology*, vol. 6, no. 3, pp. 293–335, 1986.
- [5] H. A. Dijkstra, *Nonlinear climate dynamics*. Cambridge University Press, 2013.
- [6] S. H. Schneider, "Abrupt non-linear climate change, irreversibility and surprise," *Global Environmental Change*, vol. 14, no. 3, pp. 245–258, 2004.
- [7] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *International Conference on Artificial Neural Networks*, pp. 583–588, Springer, 1997.
- [8] C. E. Rasmussen, *Evaluation of Gaussian processes and other methods for non-linear regression*. University of Toronto, 1999.
- [9] X.-W. Chen and X. Lin, "Big data deep learning: challenges and perspectives," *IEEE access*, vol. 2, pp. 514–525, 2014.
- [10] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
- [11] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf, "A kernel view of the dimensionality reduction of manifolds," in *Proceedings of the twenty-first international conference on Machine learning*, p. 47, ACM, 2004.
- [12] M. Kampffmeyer, S. Løkse, F. M. Bianchi, R. Jenssen, and L. Livi, "Deep kernelized autoencoders," in *Scandinavian Conference on Image Analysis*, pp. 419–430, Springer, 2017.
- [13] E. N. Lorenz, "Predictability: A problem partly solved," in *Proc. Seminar on predictability*, vol. 1, 1996.
- [14] J. Von Neumann, "The general and logical theory of automata," *Cerebral mechanisms in behavior*, vol. 1, no. 41, pp. 1–2, 1951.
- [15] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001.