Journal of Computational Physics 442 (2021) 110484

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A data-driven framework for the stochastic reconstruction of small-scale features with application to climate data sets

Zhong Yi Wan [a], Boyko Dodov [b], Christian Lessig [c], Henk Dijkstra [d], Themistoklis P. Sapsis [a,*]

[a] Department of Mechanical Engineering, Massachusetts Institute of Technology, USA
[b] AIR Worldwide, USA
[c] Otto-von-Guericke-Universität Magdeburg, Magdeburg, Germany
[d] Institute for Marine and Atmospheric Research, Utrecht University, the Netherlands

ARTICLE INFO

Article history:
Available online 2 June 2021

Keywords:
Stochastic downsampling of turbulence
Machine learning statistics
Conditionally Gaussian stochastic models
Climate data
Extreme events

ABSTRACT

Turbulent fluid flows in atmospheric and oceanic sciences are characterized by strongly transient features with spatial inhomogeneity, spanning a wide range of spatial and temporal scales. While large-scale dynamics are often well approximated by closure schemes there is still a need to efficiently represent the corresponding small-scale features, when it comes to the risk analysis for extreme events. We introduce a data-driven framework for the stochastic reconstruction of the small spatial scales in terms of the large ones. The framework employs a spherical wavelet decomposition to partition field quantities, obtained from reanalysis data into non-overlapping spectral components. Using these time-series we formulate, for each spatial location, a machine-learning scheme that naturally 'splits' the small-scales into a predictable part, which can be effectively parametrized in terms of the large-scales time-series, and a stochastic residual, which cannot be uniquely determined using the large-scale information. The later is represented using a conditionally Gaussian process, a choice that allows us to overcome the need for a vast amount of training data, which for climate applications, is naturally limited to a single realization for each spatial location. Using a second round of machine-learning we parametrize, for each location, the covariance of the stochastic component in terms of the large scales. We employ the machine-learned statistics to parsimoniously reconstruct random realizations of the small scales. We demonstrate the approach on reanalysis data involving vorticity over Western Europe and we show that the reconstructed random samples for the small scales result in excellent agreement to the spatial spectrum, single-point probability density functions, and temporal spectral content.

1. Introduction

Large-scale environmental flows are characterized by a wide range of spatial scales with strong dynamical coupling and important spatial inhomogeneities. While in several important problems in geophysical fluid dynamics one can observe or estimate accurately the large-scale behavior, the reconstruction of high-resolution features given the large-scale information

* Corresponding author.
E-mail address: sapsis@mit.edu (T.P. Sapsis).

https://doi.org/10.1016/j.jcp.2021.110484
0021-9991/

remains a challenge. Classical examples in ocean models are the Gent-McWilliams parameterization of the effects of ocean mesoscale features on heat and salt transport, [14], and the parameterizations of the effects of buoyancy induced convection on deepwater formation, [45]. Several other efforts have focused on the description of small-scale characteristics in terms of large scales in the context of geophysical turbulence. In [17,16,18] a stochastic superparametrization framework was developed where the effect of the small scales was parameterized in terms of large-scale quantities using linear stochastic differential equations excited by white noise. The coefficients of the linear stochastic differential equations were tuned based on high-resolution simulations. To account for spatial inhomogeneities a decomposition of the flow field into a large scale component and a time-uncorrelated component is adopted in [32–34], leading to a random version of the Reynolds transport theorem. In this series of papers the emphasis was primarily given on deriving a closed set of equations governing the large scales, rather than a detailed representation of the small scales.

In engineering turbulence several efforts have also been focused on representing the dependence of the small scale features on the larger scales, [10,28,48,11]. In particular, many works have considered the problem of reconstructing the flow field using sparse measurements or large-scale features of the flow. While these approaches have shown promise for low-Re laminar flows, they are less effective for high-Re complex flows, [9]. This is not surprising given the strongly turbulent, non-stationary, multi-scale character of these flows, which inevitably induces fundamental barriers on how much energy present in the small scales can be parameterized from the large scales in a deterministic way. Specifically, the nonlinear interactions between small and large scales are typically characterized by strong instabilities which, in turn, lead to loss of predictability of the small scales, [46,4,31]. Therefore, independently of what parametrization method is employed, one has to compromise with the fact that any deterministic parametrization of the small scales will be limited by the loss of predictability.

The scope of this work is the formulation of a stochastic representation framework that will be able to generate realizations of the small scale features conditioned on a given realization for the large scale dynamics. This is particularly important for the quantification of risk for certain events that can take place in small spatial domains, i.e. smaller than the large scale dynamics. This topic, known as statistical downscaling, has been the focus of numerous studies over the last decades. These include ideas based on linear regression and analog techniques [19,26]. For example in the context of statistical downscaling for precipitation, an estimator is obtained using an optimized linear combination of the local circulation features [20,23]. Beyond these linear schemes, there is a number of machine learning techniques that have been utilized for statistical downscaling. These efforts began with artificial neural networks [47,37] and subsequently relied on alternative machine learning approaches, such as support vector machines [41], random forests [22,30], nearest neighbor [12] or genetic programming [35]. More recently, deep learning ideas have also been employed for statistical downsampling. These include cases aiming for recovering high-resolution fields from low-resolution inputs with a generalized stacked super resolution convolutional neural network [44], autoencoder architectures [43], long short-term memory networks [29] and ideas based on blended architectures [27]. A comprehensive assessment of different deep learning techniques for statistical downscaling for precipitation and temperature is presented in [2].

While deep learning ideas have proven to be successful in several cases, they are not always able to adequately parametrize all the energy of the small scale features. In particular, when the downscaling goes to sufficiently fine scales, there is typically an energy component that behaves stochastically, i.e. it cannot be parametrized in a deterministic manner from the low-resolution input. This requires the adoption of a stochastic representation for the downscaling. Recent efforts have focused on this issue using adversarial deep learning [39], a promising approach for representing stochastic phenomena by machine learning the full probability density function. However, adversarial deep learning requires a vast amount of training data, which inevitably leads to the assumption of spatial homogeneity, i.e. machine learn a single neural network using as training data the information over different spatial regions. The obstacle in this case is the spatially-inhomogeneous character of climate dynamics which does not allow us to utilize data from one region in order to train a data-driven parametrization scheme for another region. Therefore, we have to rely for training data on a single realization (reanalysis data) for each spatial location.

To overcome these limitations, we introduce a new framework, the Stochastic Machine-Learning (SMaL) parametrization. The SMaL parameterization framework consists of a spatial wavelet decomposition at different scales, a machine-learning parameterization of the predictable part of the small scales, as well as a machine-learning parametrization of the stochastic part of the small scales. The latter is represented using a conditionally Gaussian statistical structure in order to minimize the need for training data. The assumption of Gaussian statistics for the small scale fluctuations conditioned on the large scales does not imply Gaussian statistics for the fluctuations. Such conditional Gaussian models have been developed and analyzed previously in the context of filtering and data-assimilation for special models, [6,7,5,24]. Here we focus on machine-learning the mean and covariance of the conditionally Gaussian models for the small scales in terms of the large scale features. This is a non-trivial task as we have to rely on a single realization (reanalysis data) in order to machine-learn statistical quantities such as the conditional covariance, while we also have to respect the spatially inhomogeneous character of the climate dynamics, i.e. we cannot use data from one spatial location to train in another location.

First we employ a wavelet decomposition to partition the training flow fields into large and small-scale features. The choice of a localized representation is a critical aspect, as it allows us to exploit the local character of the interactions between scales. In this way, it leads to more efficient parametrization of the dynamics governing the small scales and better predictability skill. Next, we train a temporal convolution network (TCN), which has as input the large-scale information and provides as output the small scales. Because of the turbulent character of the dynamics, only a portion of the overall energy

contained in the small scales can be parametrized and reproduced correctly by this TCN. In this way, this first stage of machine-learning naturally "splits" the small scales into i) a predictable component, which can be effectively parametrized by the large-scale features through the TCN scheme, and ii) a stochastic remainder, which cannot be parametrized by the data-driven scheme. The challenge now is how to compute the statistical characteristics, e.g. variance, of this stochastic remainder since we have only one realization available. To overcome this obstacle we introduce a local-in-time-averaging operator, as well as a method to select the averaging window, which provides the local-in-time variance of the stochastic residual. Using a second round of data-driven modeling with a TCN we parametrize the local-in-time variance of the fluctuations in terms of the large-scale features. As a final step for the training phase, we estimate the temporal correlation matrix of the stochastic fluctuations using the correlation coefficient matrix for the small scales, estimated from the full time series of the training data, to obtain a complete estimate of the statistics of the small-scale behavior. This is done taking into account the spatial inhomogeneous character of the dynamics.

The last component of the framework involves the reconstruction of random realizations, consistent with the large scales. This is done by superimposing the predictable part of the small scales (obtained from the first TCN) with random realizations of the stochastic part. The random realizations are generated by utilizing the predicted temporal statistics (obtained from the second TCN) and the assumption of conditionally Gaussian statistics. We demonstrate the SMaL parameterization framework on reanalysis data for atmospheric quantities of interest such as vorticity over Western Europe.

2. Problem formulation

The goal of this work is the development of a data-driven downscaling framework where small-scale features will be machine-learned in terms of the large-scale characteristics. In this section we provide an overview of our approach, SMaL, through a reanalysis data set involving atmospheric flows.

2.1. Data set

We consider a data set (ERA-Interim) consisting of high-resolution reanalysis fields, denoted as $\mathbf{Q}(\mathbf{x}, t)$, where $\mathbf{x}$ is the spatial variable and t is time. Specifically, our data is based on a 6-hourly reanalysis of atmospheric field measurements ranging from January 1, 1979 to December 31, 2017 (39 years in total), [3]. For the purpose of modeling, we consider a two-dimensional spatial domain discretized as a 512×256 (longitude $\times$ latitude) polar grid, by fixing the vertical coordinate at the sigma level 0.95 iso-pressure surface. For demonstration, we consider as modeled variables the vertical component of the vorticity vector. However, there are no technical limitations on considering other quantities. Although the considered variables do not capture the full state of the atmosphere, our aim is to demonstrate and assess the potential of the proposed framework through this example.

2.2. Spatial wavelet analysis

The first step of is to split the reanalysis fields into large- and small-scale features. We adopt a Parseval tight wavelet frame, e.g. an orthogonal wavelet basis, $\mathbf{\Psi} = \{\Psi_i\}_{i\in\mathcal{I}}$, where $\mathcal{I}$ is the index set of the basis functions. Since, we are interested in climate applications, this frame is in the present work given by spherical wavelets [8]. By projecting the data set we have the approximation due to truncation error:

$$\mathbf{Q}(\mathbf{x}, t) \simeq \sum_{l\in\mathcal{I}} c_l(t)\Psi_l(\mathbf{x}), \tag{1}$$

where $\mathbf{c} = [c_l]$ represents the corresponding coefficients. The wavelet basis $\{\Psi_l\}$ captures features of different spatial scales, and locations. We use the indices $\mathcal{L}$, $\mathcal{S}$ to denote a partition of the index set $\mathcal{I}$ (with coefficients $\mathbf{c}_\mathcal{L}, \mathbf{c}_\mathcal{S}$ and basis partition $\mathbf{\Psi}_\mathcal{L}, \mathbf{\Psi}_\mathcal{S}$) representing large- and small-scaled basis elements, respectively.

The use of wavelets, parameterized by their center and a scale parameter, allows us to achieve a good balance between spatial and (spatial) frequency localization in the description of climate events. For the considered reanalysis data set the wavelet decomposition consists of 5 scale levels, accounting for different bands of spatial wavenumbers in the original fields. These are denoted as $\mathbf{\Psi}_i, i = 1, ..., 5$ and the corresponding coefficients are given as $\mathbf{c}_i, i = 1, ..., 5$. From large to small scales (henceforth referred to as levels 1 to 5), there are 32, 128, 512, 2048 and 8192 wavelets that span the Earth's surface respectively. Without loss of generality, the proposed framework is applied over an area centered around a single level-1 wavelet located at (46.98°N, 10.04°E) in Western Europe (near Zuggenwald, Austria), with a radius of approximately 2557 kilometers.[1] The number of centers or nodes in the local region we consider are 1, 5, 19, 79, and 322 for the five levels we are considering, respectively. The geometry of the area is shown in Fig. 1 along with the location of the wavelet centers.

[1] This radius is selected such that 32 patches of the same area around each level 1 wavelet would cover up the Earth's entire surface.

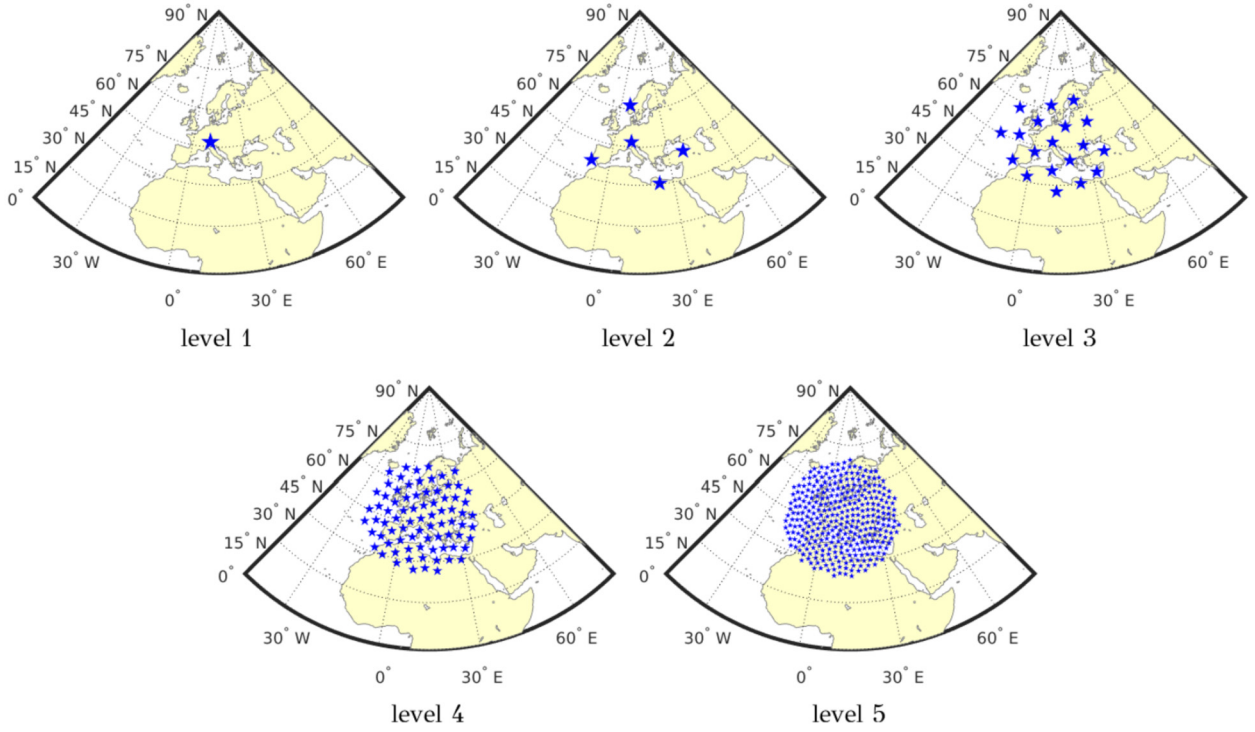


Fig. 1. Locations of the wavelet centers. The isotropic spherical wavelets divide all fields into 5 different levels, $\Psi_i, i = 1, ..., 5$. The area investigated in this application consists of 1, 5, 19, 79, and 322 wavelets, whose centers are shown here.

2.3. Empirical-orthogonal-functions analysis of small-scale wavelet coefficients

An intermediate step before focusing on the description of the small scales in terms of the large features is an order-reduction of the small scales. This is meaningful since the small scales coefficient vector, $\mathbf{c}_{\mathcal{S}}$, has a much higher dimensionality than $\mathbf{c}_{\mathcal{L}}$, as the small scale wavelet basis elements are naturally defined over a much denser grid (Fig. 1). Therefore, it is often meaningful, although not essential, to obtain a reduced-order description of the small scales.

In this work we will apply empirical orthogonal functions (EOF) analysis or principal component analysis (PCA). Separate EOF modes are obtained for each of the small-scale levels, $\mathbf{c}_4, \mathbf{c}_5$. An interesting property to note is that the most energetic EOF modes (on each wavelet level) can involve dynamics with scales large enough to compare with those captured by the large-scale basis, $\Psi_{\mathcal{L}} = \{\Psi_1, \Psi_2, \Psi_3\}$. This is not contradictory with the orthogonality of $\Psi_{\mathcal{L}}$ and $\Psi_{\mathcal{S}}$. A simple example is shown in Fig. 2. A multi-scale signal is decomposed with a wavelet basis into a low frequency component, which 'lives' in $\Psi_{\mathcal{L}}$, and high frequency component, which 'lives' in $\Psi_{\mathcal{S}}$. Clearly there is a nonlinear dependence associated with high frequencies present only in large values of the low-frequency component. If one performs EOF analysis on the wavelet coefficients describing the small scales, $\mathbf{c}_{\mathcal{S}}$, the large-scale EOF modes will capture features associated with the nonlinear dependencies between small and large scales, i.e. features caused by the large-scales and which are being reflected in the small-scales. On the other hand, smaller-scale EOF modes will focus on capturing the exact shape of the small scale features.

For our data set, we apply a separate EOF analysis on the level-4 and level-5 (small-scale) wavelet coefficients:

$$\mathbf{c}_i(t) = \sum_{k=1}^{d_i} \Xi_{ik}\xi_{ik}(t), \quad i = 4, 5, \tag{2}$$

where for each i, ξ_{ik} are pairwise uncorrelated and Ξ_{ik} are deterministic vectors (or 'mode shapes') that have the same dimension as $\mathbf{c}_i$. The integer d_i represents the EOF truncation order for each wavelet level. This formulation allows us to represent the dominant part of the energy in each wavelet level of the small scales using a small fraction of the dimensions in $\{\xi_{ik}\}_{k=1}^{d_i}$.

The modes are computed from the training data and the resulting energy spectrum for the 79-dimensional level-4 coefficients is shown in Fig. 3. We choose $d_4 = 20$ modes, which cover more than 94% of the total energy. The shapes of the first four EOFs are also plotted in the same figure. As noted earlier, the high-energy EOFs are those associated with large spatial scales, i.e. large-distance dependencies of the small-scale wavelet coefficients.

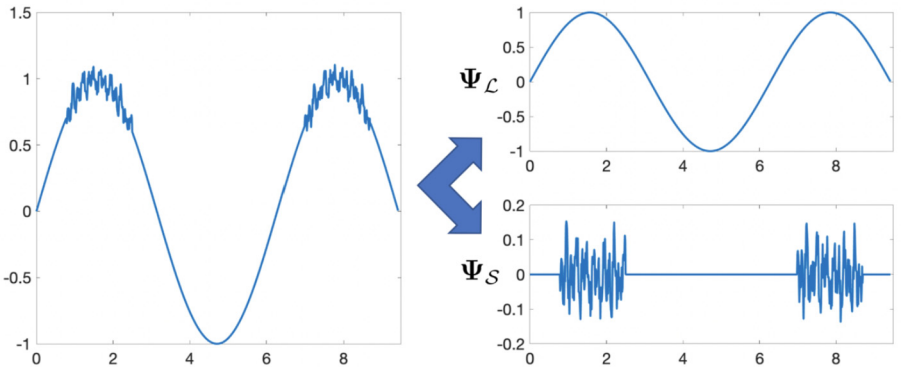


Fig. 2. Decomposition of a spatial field into large and small scales. Note that the small scales may still be characterized by large-scale dependencies.

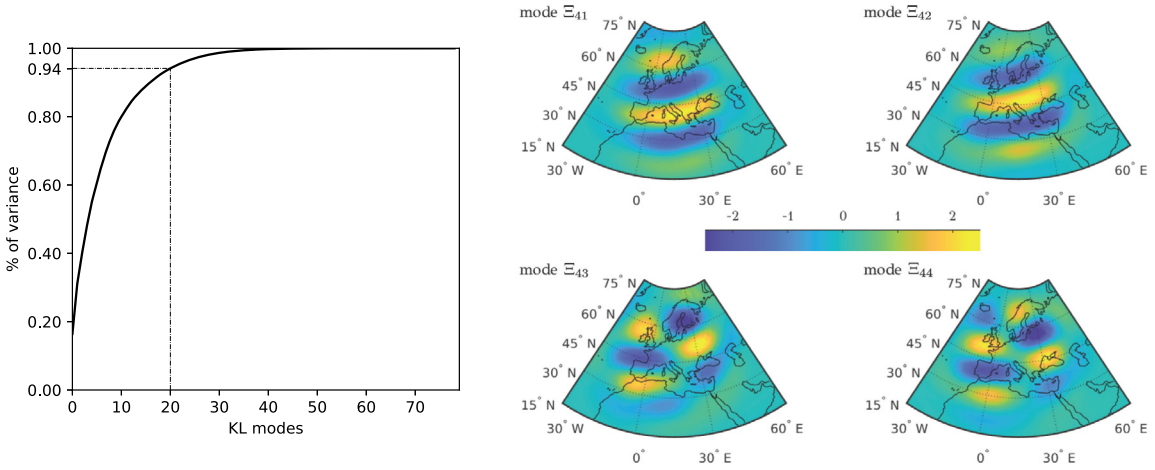


Fig. 3. EOF decomposition for level-4 wavelet coefficients. Left: energy spectrum - the top 20 EOF modes account for 94% of the total energy. Right: mode shapes (converted to physical domain) of the top 4 EOF modes. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

2.4. Deterministic parameterization of the small scales and limitations

Our goal is to formulate a framework which, for a given time series of the large-scale features, $\mathbf{c}_{\mathcal{L}}$, will provide realizations for the small-scale features, $\mathbf{c}_{\mathcal{S}}$, in a causal manner, i.e. $\mathbf{c}_{\mathcal{S}}(t)$ will be parameterized using only the time series segment, $\mathbf{c}_{\mathcal{L}}(\tau), \quad \tau \leq t$. Several studies, [10,28,48], have developed deterministic modeling methods based on data-driven techniques which explicitly parameterize the small-scales, $\mathbf{c}_{\mathcal{S}}$, in terms of the large scales, $\mathbf{c}_{\mathcal{L}}$. In many cases, delay coordinates and memory-incorporating neural networks are utilized to represent the effect of the large scales on the small scales. Delay coordinates in $\mathbf{c}_{\mathcal{L}}$ are able to provide an augmented state space, which, according to Takens' embedding theorem, [40], contains much richer state information than the corresponding Markovian models. Recently, it was shown that an architecture based on shallow neural networks outperforms all existing methods that aim to reconstruct laminar flows from sparse measurements, i.e. from large-scale features, [9]. However, for the case of turbulent flows the method had severe limitations.

Here we demonstrate these limitations in the considered data set. We apply a deterministic machine-learning scheme (see section 3 for details) to express the EOF coefficients ($\boldsymbol{\xi}_{4k}$) for level-4 wavelets as functions of the large-scale features, $\mathbf{c}_{\mathcal{L}}$. In this way we obtain the data-driven expression, G, that provides the best possible *approximation*, using the employed data-driven scheme, for the EOF coefficients, $\boldsymbol{\xi}_{4k}$:

$$\hat{\boldsymbol{\xi}}_{4k}(t_n) = G\Big(\mathbf{c}_{\mathcal{L}}(t_n), \mathbf{c}_{\mathcal{L}}(t_{n-1}), \ldots\Big), \quad k = 1, \ldots, d_4. \tag{3}$$

The reconstructed signal, $\hat{\boldsymbol{\xi}}_{4k}(t)$, is the predictable component of the level-4 EOF coefficients, $\boldsymbol{\xi}_{4k}(t)$, using large-scale information.

In Fig. 4 we present the energy of each level-4 EOF component, $\boldsymbol{\xi}_{4k}$, and of the remainder, $\boldsymbol{\xi}_{4k} - \hat{\boldsymbol{\xi}}_{4k}$, which is the stochastic component, i.e. the part of the signal that cannot be parametrized effectively in terms of the large scales and

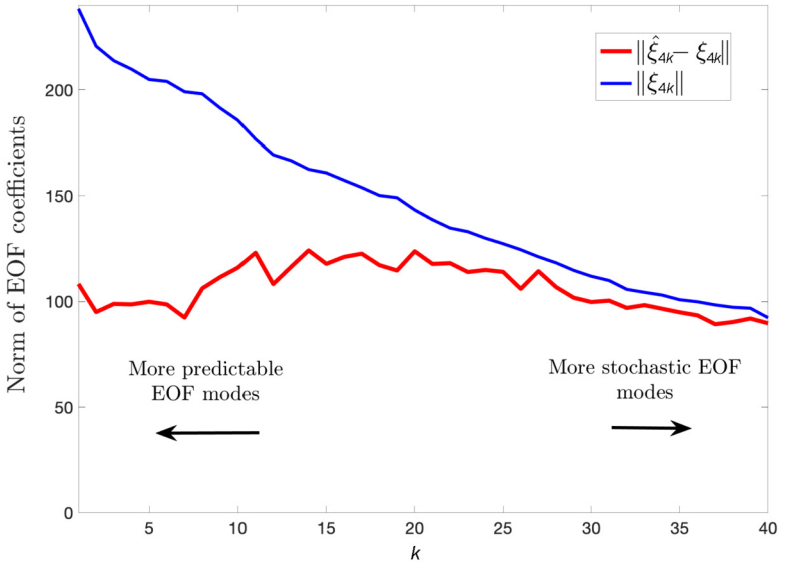


Fig. 4. Distribution of EOFs energy (blue curve) for level-4 wavelets shown together with the energy of the component that cannot be parametrized with a deterministic data-driven approach (red curve). Low-order EOF modes for the level-4 wavelet coefficients present more predictable behavior as a bigger component of their energy can be parameterized in terms of the large-scale wavelet coefficients, $\mathbf{c}_{\mathcal{L}}$.

their history using our data-driven scheme. As we saw, low-order EOFs are associated with large spatial scale dependencies (see Fig. 2). These are hence unsurprisingly the modes that can be parametrized most effectively by the deterministic data-driven scheme in terms of the large-scale wavelet coefficients, $\mathbf{c}_{\mathcal{L}}$. Note however that even for these modes there is significant error that cannot be captured by the deterministic data-driven scheme. For higher-order EOFs, associated with small-scale dependencies of the level-4 wavelet coefficients, a deterministic data-driven representation captures only a negligible amount of their energy. It is worth noticing that the level of energy that can be deterministically captured depends on the machine learning model/architecture employed as well as the training data size. However, there are intrinsic barriers on how much energy can be captured from a deterministic scheme that cannot be surpassed. These intrinsic limitations are related to the fact that the larger-scale coefficients cannot uniquely define how the small-scale wavelet coefficients will evolve. To this end, a stochastic parametrization is essential in order to represent effectively the properties of the small scales.

2.5. Stochastic parametrization scheme of the small scales

The predictability analysis based on a deterministic data-driven scheme in the last section revealed that the small-scale wavelet coefficients consist of i) large-scale EOFs which are possible to capture (to some extent) using the large-scale information in $\mathbf{c}_{\mathcal{L}}$, and ii) small-scale EOFs which have lower energy but they are not parameterizable in terms of the large scales (cf. Fig. 4) These findings are consistent with previous studies that analyze predictability of modes with different wavenumbers in the context of weather prediction, [4,31].

Motivated by these properties we formulate the Stochastic Machine-Learned (SMaL) Parameterization method where the EOF coefficients are represented as *non-stationary*, *Gaussian stochastic processes*, conditioned on the large-scale wavelet coefficients. The statistics of these non-stationary stochastic processes consist of i) a time-dependent mean that is parameterized (i.e. machine-learned) in terms of the large scales, $\mathbf{c}_{\mathcal{L}}(\tau), \quad \tau \leq t$, and which is essentially the predictable part, and ii) a time-dependent covariance matrix, also parameterized in terms of the large-scales, $\mathbf{c}_{\mathcal{L}}(\tau), \quad \tau \leq t$, and which defines the behavior of the stochastic part (or error of the deterministic prediction). More specifically, denoting as $\{X_0, X_1, ..., X_T\}$ the input stochastic processes (i.e. the large scales, $\mathbf{c}_{\mathcal{L}}(\tau), \quad \tau < T$) and as $\{Y_0, Y_1, ..., Y_T\}$ the output stochastic process (i.e., the EOF coefficients for the small scales, e.g., $\boldsymbol{\xi}_{4k}(t)$) a SMaL parameterization contains the following steps:

i) (**Model design**) establish a parametric causal model M for the joint conditional distribution

$$\{Y_0, Y_1, ..., Y_{T-1} | X_0 = \mathbf{x}_0, X_1 = \mathbf{x}_1, ..., X_{T-1} = \mathbf{x}_{T-1}\} \sim M(\boldsymbol{\Theta}; \mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_{T-1}), \tag{4}$$

where $\boldsymbol{\Theta}$ denotes the model parameters.

ii) (**Learning**) estimate $\boldsymbol{\Theta}$ systematically from a single realization of $\mathbf{X}$ and $\mathbf{Y}$.

iii) (**Predictive sampling**) given a large-scale realization $\mathbf{X}^* = \{\mathbf{x}(t_0^*), ..., \mathbf{x}(t_{P-1}^*)\}$ with a different time span $\{t_0^*, ..., t_{P-1}^*\}$ (typically non time-overlapping with training data, hence * is used to signify that the quantities are associated with prediction, *not* training), use M to provide characterizations of the conditional process

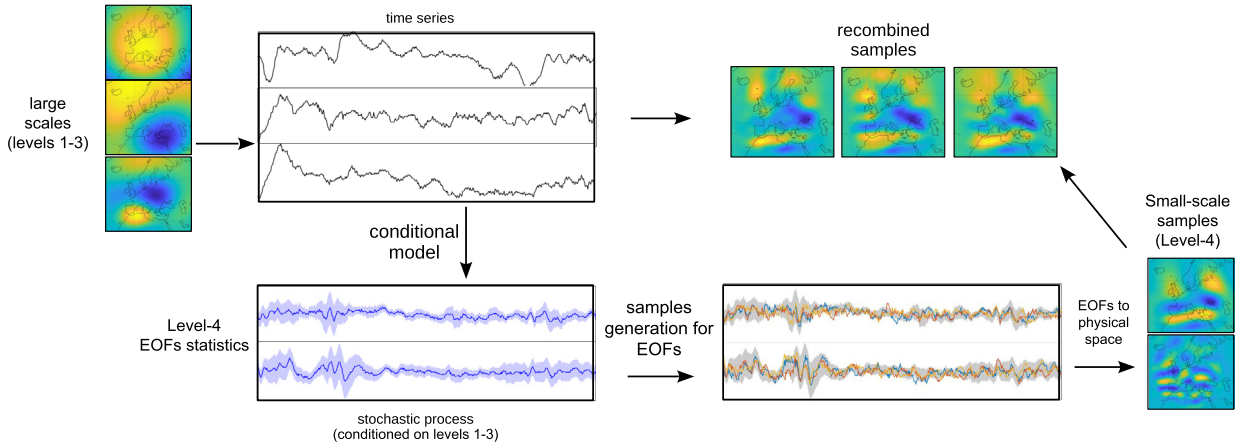


Fig. 5. SMaL parameterization framework: Decomposition of the fields into large- and small- scales. Small scales contained in level-4 wavelet coefficients are expressed in terms of an EOF basis. A machine-learned conditional Gaussian model for the EOF coefficients allows for the generation of a family of random samples. All samples have the same predictable part, given as a function of the large scales (blue curve) and a stochastic part with statistics given as a function of the large scales.

$$\{Y_0, ..., Y_{P-1} | X_0 = \mathbf{x}(t_0^*), ..., X_{P-1} = \mathbf{x}(t_{P-1}^*)\} \sim M\left(\mathbf{\Theta}; \mathbf{x}(t_0^*), ..., \mathbf{x}(t_{P-1}^*)\right), \tag{5}$$

and allow random sample paths $\mathbf{Y}^*(\omega)$ (ω represents an element drawn from an appropriate sample space) to be generated efficiently.

Each of the steps is associated with several technical challenges that we will discuss in detail in section 4.

A graphical summary of the framework for generating realizations for level-4 wavelets is shown in Fig. 5. The framework generates random realizations for the small scales conditioned on the history of the large scales. Details of the adopted machine-learning algorithm are given in the next section.

Our discussion so far involved the modeling of level-4 EOF coefficients which were conditioned on the large scale wavelet coefficients (levels 1-3). For higher level wavelets (e.g. level-5, etc.) the corresponding EOF coefficients are conditioned in terms of the wavelet coefficients of all previous levels, in a *nested structure*. For example, level-5 EOF coefficients will be represented as a non-stationary stochastic process conditioned on the large scale wavelet coefficients, $\mathbf{c}_{\mathcal{L}} = \{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3\}$, as well as random realizations of the level-4 wavelet coefficients. These random realizations are obtained from the stochastic process representing level-4 EOFs, which is conditioned on the large scale wavelet coefficients, $\mathbf{c}_{\mathcal{L}} = \{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3\}$.

In the following section we provide a detailed presentation of the machine-learning methods we will employ for the training and parametrization of the statistics (mean and variance) of the non-stationary stochastic processes that will represent the small-scale EOFs.

3. Temporal convolution networks

Before we describe details of the SMaL parametrization we introduce the machine-learning architecture at the heart of our work - the temporal convolution networks (TCN). This architecture has been recently shown to outperform its popular alternative, the recurrent neural networks (RNNs), in speech and language related tasks, [42,1]. For the modeling problems considered in this work, we have observed from numerical experiments that the performance is indistinguishable between a TCN and a similar-sized RNN in terms of prediction accuracy. A TCN is chosen because it is more easily parallelizable, faster to train, has more stable gradients and consumes lower memory during training (with the trade-off being that more memory is needed at test time), making it a more suitable architecture for even more sophisticated applications than illustrated here.

At a high level, a TCN learns a deterministic mapping between an input sequence $\mathbf{x}_0, ..., \mathbf{x}_{T-1}$ and an output sequence $\mathbf{y}_0, ..., \mathbf{y}_{T-1}$. By imposing appropriate constraints on the architecture, a TCN is designed to model a class of sequence mappings which are especially meaningful in a dynamical system context. Here we place the emphasis on the distinctive features of TCNs, compared with the commonly used neural networks. For a comprehensive overview of basic theories and concepts on deep networks in general, readers are referred to [15].

3.1. Temporal convolution

In contrast to regular convolution neural networks (CNNs), convolutions in TCNs are special in that they are *causal* and *dilated* (see Fig. 6a). The temporal convolution operation is defined for a sequence input (of arbitrary length)

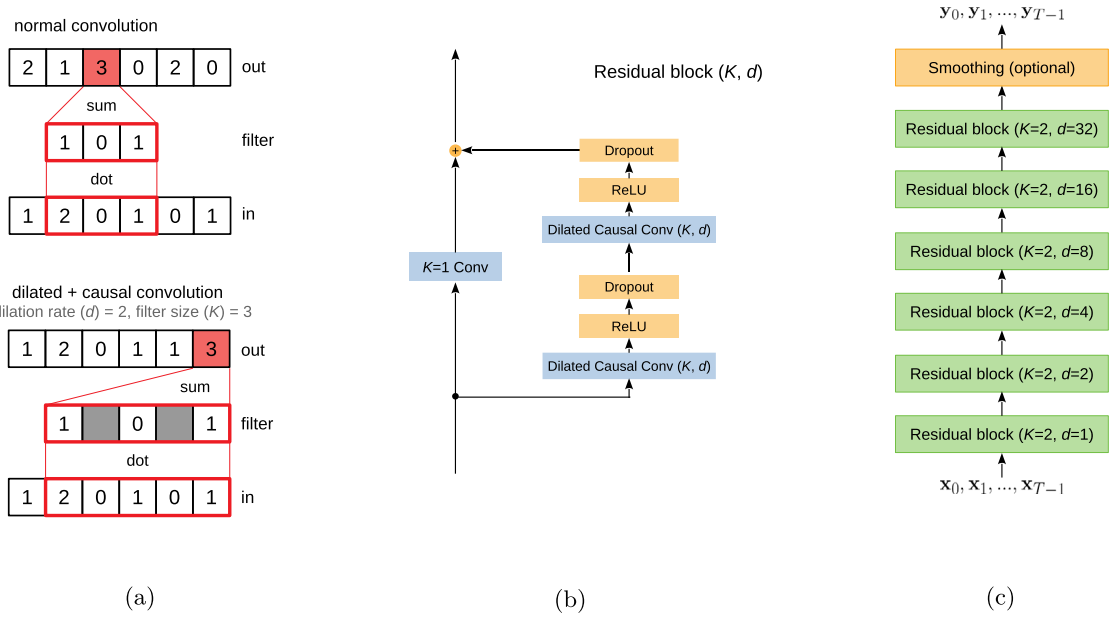


Fig. 6. Architectural design of temporal convolution network model. (a) Comparison between regular and causal + dilated convolution. (b) Residual block comprises two convolution layers. (c) The overall design with a series of stacked residual blocks whose dilation rate grows exponentially with layers.

$\mathbf{X} = \{\mathbf{x}_0, ..., \mathbf{x}_{T-1}\}, \mathbf{x}_i \in \mathbb{R}^n$ and a filter $\mathbf{F} = \{\mathbf{F}_0, ..., \mathbf{F}_{K-1}\}, \mathbf{F}_i \in \mathbb{R}^{m \times n}$. The outcome is a sequence $\mathbf{Y} = \{\mathbf{y}_0, ..., \mathbf{y}_{T-1}\}, \mathbf{y}_i \in \mathbb{R}^m$, obtained via:

$$\mathbf{y}_i = (\mathbf{X} *_d \mathbf{F})(i) = \sum_{k=0}^{K-1} \mathbf{F}_k \mathbf{x}_{i-d \cdot k}, \tag{6}$$

where d is the dilation factor and K is the filter size. In other words, the output at a particular step is calculated as the product of each filter matrix and a nearby input state all summed together. Here the convolution being causal refers to the fact that output $\mathbf{y}_i$ has dependence only on input states $\mathbf{x}_j$ with $j = i - d \cdot k \leq i$. This property implies that the output states are entirely determined by historical events with respect to them, which is consistent with the primary assumption of dynamical systems. We also assume $\mathbf{x}_j = \mathbf{0}$ whenever $j < 0$ is needed to ensure that the input and output sequences have the same length.

The convolution being dilated implies that adjacent filter taps are $d > 1$ steps apart. The case of $d = 1$ actually reduces the operation to a regular causal convolution. Utilizing a larger dilation rate, however, enables the output to gather information from a wider range of input states. Stacking such dilated convolution layers expands the receptive field of the model output exponentially. In a dynamical system context, this provides the infrastructure for the model to easily capture dynamics occurring at different time scales, which can be crucial for nonlinear systems.

Following [1], we employ a multi-layered architecture where the dilation rate is increased exponentially with the depth of the network (i.e., $d = \{1, 2, 4, ...\}$ for intermediate convolution layers). This setup ensures that output can extract information from every input step while maintaining a large effective history.

3.2. Architecture design

The full structure of the TCN architecture used in this work is illustrated in Fig. 6c. Apart from temporal convolutions, the following features of TCNs are worth highlighting:

- *Deep structure with residual blocks*: A TCN utilizes residual connections, [21], that learn modifications to the identity mapping rather than the entire transformation. Combined with a deep network structure, this type of connection has been shown to greatly improve the modeling capabilities. Each residual block used in TCN consists of two temporal convolution layers. In addition, since the input and output generally do not have the same number of dimensions, a 1×1 convolution is added to convert the input to a compatible shape. The overall block architecture is illustrated in Fig. 6b.
- *Output smoothing*: Optionally, our TCN architecture may employ a non-trainable layer that applies smoothing to the output prior to computing the cost. This operation serves to help the model match the smoothness of its predictions with that expected on the output. This smoothing layer is used for modeling the local-in-time variance in section 4.

- *Regularization*: TCN employs dropout layers, [38], as the primary regularization mechanism to prevent overfitting. The dropout layers are inserted after each temporal convolution operation in the residual blocks and randomly zeroes out any channel in the output with a fixed probability p. This operation forces the neurons to learn robust features that work well with many different random subsets of other neurons.
- *Weight normalization*: The convolution filters in TCNs are weight normalized, meaning that the length and direction of the weight vectors are decoupled [36]. This has been shown to improve the conditioning of the optimization problem and accelerate the training process noticeably.

3.3. Parameter optimization

Training the TCN is equivalent to finding the weights and biases (collectively denoted by $\mathbf{\Theta}$) that minimize a loss function, which evaluates the fit of the model to the data. Denoting the target output by $\{\mathbf{y}_0, ..., \mathbf{y}_{T-1}\}$ and the corresponding model prediction by $\{\hat{\mathbf{y}}_0, ..., \hat{\mathbf{y}}_{T-1}\}$, the cost function $\mathcal{J}$ is usually defined as the average of the errors committed for each training case and each time step:

$$\mathcal{J}(\mathbf{\Theta}) = \frac{1}{T}\sum_i \mathcal{L}\big(\hat{\mathbf{y}}_i(\mathbf{\Theta}), \mathbf{y}_i\big), \tag{7}$$

where $\mathcal{L}$ measures the penalty for a single point. For real-valued predictions, the mean absolute error (MAE) and mean squared error (MSE) are commonly used:

$$\mathcal{P}_{\text{MAE}}(\hat{\mathbf{y}}_i, \mathbf{y}_i) = \frac{1}{m}\sum_j |\hat{y}_{i,j} - y_{i,j}|, \tag{8a}$$

$$\mathcal{P}_{\text{MSE}}(\hat{\mathbf{y}}_i, \mathbf{y}_i) = \frac{1}{m}\sum_j (\hat{y}_{i,j} - y_{i,j})^2, \tag{8b}$$

where j indexes the vector components of the associated quantities. Having defined the loss function, one can solve the minimization problem using a gradient-based approach to find the weights.

Both MAE and MSE are known to produce relatively 'conservative' models because large-amplitude predictions are more likely to result in large absolute or squared errors. On the other hand, this is not necessarily the case for correlation-based loss functions (used to machine-learn the variance) which seek to maximize the covariance between the prediction and truth (see [25] for example). In this case, certain vector components may have relatively small variance compared with others. To have equally good training for all vector components of the correlation, independently of their magnitude, we introduce a novel cost function referred to as the mean negative anomaly correlation coefficient (MNACC), defined as

$$\begin{aligned}\mathcal{J}_{\text{MNACC}} &= -\frac{1}{m}\sum_j \frac{\sum_i \big(\hat{z}_{i,j} - [\hat{z}_{i,j}]\big)\big(z_{i,j} - [z_{i,j}]\big)}{\sqrt{\sum_i (\hat{z}_{i,j} - [\hat{z}_{i,j}])^2}\sqrt{\sum_i (z_{i,j} - [z_{i,j}])^2}} \\ &\equiv -\frac{1}{m}\sum_j \text{PCC}\big(\hat{z}_{\cdot,j}, z_{\cdot,j}\big),\end{aligned} \tag{9}$$

where $[\cdot]$ denotes the average value taken across all time steps (i.e. across index i with fixed index j) and

$$z_{i,j} = y_{i,j} - y^{\text{ref}}_{i,j}. \tag{10}$$

Here $y^{\text{ref}}_{i,j}$ is a reference prediction (depending on time step, i, and vector dimension, j) which represents a baseline that the model is expected to beat. Providing this additional information allows the model to directly focus on learning the *deviation from the reference* (a measure for the degree of anomaly), denoted by z, which in practice has led to improved performance. The MNACC effectively calculates the negative Pearson correlation coefficient (PCC) between the predicted and true deviation across time steps (negative sign ensures the optimization is a minimization) and punishes large-amplitude predictions much less severely as long as the truth is also deemed an anomaly with respect to its mean (see appendix A for more details).

Since the MNACC is both mean- and scale-invariant, an additional calibration operation is required. The final prediction is calculated as

$$\hat{\mathbf{y}}_i = \mathbf{y}^{\text{ref}}_i + \mathbf{W}\hat{\mathbf{z}}_i + \mathbf{b}, \tag{11}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the (diagonal) weight matrix and bias, and are obtained by optimizing the MSE between $\hat{\mathbf{y}}$ and $\mathbf{y}$. Although the use of MSE is eventually inevitable, incorporating MNACC as an intermediate step can potentially boost the model accuracy. We optimize for the hybrid loss:

$$\mathcal{J} = \mathcal{J}_{\text{MNACC}} + \gamma \mathcal{J}_{\text{MSE}}, \tag{12}$$

where γ is a weighting hyperparameter that needs to be tuned.

When a reference prediction is not easily found, one can use zero as the reference at all times and MNACC reduces to the negative Pearson correlation coefficient (NPCC). However, for variables exhibiting obvious periodic trends (e.g. annual), as is the case for many in weather and climate prediction, we can use as a reference prediction the cyclic mean [13].

4. Training of SMaL parametrization

With the model variables and the scale-differentiating representation (i.e. the wavelet functions), we can proceed with the detailed description of the SMaL parametrization framework. We model the small-scale wavelet coefficients as a non-stationary *Gaussian* process conditioned on the large-scale wavelet coefficients. This process has as mean the predictable part obtained from a deterministic machine-learning scheme that takes as input the large-scale wavelet coefficients. The covariance matrix of the process describes the statistical characteristics of the remainder (unpredictable component) which are also conditioned on the large-scale wavelet coefficients. In order to obtain these statistical characteristics and train the corresponding machine-learning scheme, we can only utilize a single realization of the conditioning process, $\mathbf{c}_{\mathcal{L}}$, and a single realization of the target process, the EOF coefficients, $\boldsymbol{\xi}$. To achieve this goal we make the assumption that the statistical characteristics of the stochastic remainder of the target process (i.e. the component that cannot be captured by the deterministic data-drive scheme) change *slower* compared with the process itself. This is not a restrictive assumption as we are talking about small scales which evolve much faster than their statistics. With this setup we formulate the training procedure as follows:

4.1. Data preprocessing

As described in section 2, data is first transformed to wavelet space after the modeling variables (vorticity) and spatial domain (circular patch in Europe) are selected. The overall model input $\mathbf{c}_{\mathcal{L}}$ consists of the large scale wavelet coefficients within the selected spatial domain (see section 2.2). The target small scales undergo EOF analysis where the coefficients are retained up to a satisfying energy level (94% in our example). Data is partitioned into training (years 1979-2011) and testing (years 2011-2018) sets for the purpose of model verification.

4.2. Machine-Learning of the predictable part

The TCN is directly applied to learn a mapping from $\mathbf{c}_{\mathcal{L}}$ to the EOF coefficients, $\boldsymbol{\xi}_i = \{\xi_{ik}\}_{k=1}^{d_i}$, corresponding to level-i wavelets. Due to the TCN architecture, the mapping takes the form (for level-4 EOF coefficients)

$$\hat{\xi}_{4k}(t_n) = G_{\text{m},4k}\Big(\boldsymbol{\Theta}_{\text{m},4}; \mathbf{c}_{\mathcal{L}}(t_n), \mathbf{c}_{\mathcal{L}}(t_{n-1}), \ldots\Big), \text{ for } k = 1, \ldots, d_4. \tag{13}$$

$\mathbf{G}_{\text{m},4} = \{G_{\text{m},4k}\}_{k=1}^{d_4}$ attempts to capture all features that can be inferred from the past trajectory of the large-scale wavelet coefficients. Every prediction can be potentially affected by inputs that are maximum 64 steps (16 days) before. This parameter was chosen conservatively, by gradually increasing the length of history until performance gain is no longer significant. By using the MSE loss and carefully avoiding overfitting, $\mathbf{G}_{\text{m},4}$ typically has softened fluctuations in its predictions, which can therefore be viewed as the result of a local averaging operation performed on the realizations of $\boldsymbol{\xi}_4(t)$. Overfitting is avoided by tuning up the dropout percentage on the network architecture and making sure the errors on training and test sets are similar. It is important to emphasize that overfitting in the machine-learning scheme for the predictable part will lead to very bad performance for the machine-learning of the variance of the stochastic part. This is because the second machine-learning algorithm will be trying to learn overfitted residuals. Based on this training procedure, we consider $\mathbf{G}_{\text{m},4}$ as an approximate *mean model* for $\boldsymbol{\xi}_4(t)$.

For our data set we learn directly the mapping from $\mathbf{c}_{\mathcal{L}}$ (levels 1 to 3 of vorticity coefficients) to $\boldsymbol{\xi}_4(t) = \{\xi_{4k}(t)\}_{k=1}^{d_4}$ (EOF coefficients for level-4 vorticity) using the MSE loss. We observe from the validation results (Fig. 7) that the TCN learns to deterministically predict the high-energy EOF modes with high accuracy. However, the performance drops off for modes with lower energy (illustrated in Fig. 7). This is not surprising because we expect the high-variance modes to represent the structures formed as a direct consequence of the high-energy, large-scaled dynamics and should therefore be more predictable with respect to the input. On the other hand, the low-energy modes represent the dynamics largely contributed by the low-energy interactions between small-scaled features, which is less predictable and better represented stochastically. It is important to emphasize that we avoid that the TCN focuses only on the leading energy modes (as these are weighted more when MSE is computed) by training a separate TCN on each mode separately.

4.3. Time-averaging of the stochastic part

The next step is the statistical characterization of the error made by the first round of machine learning. To obtain statistics from a single realization, we make the assumption that the statistical characteristics of the stochastic remainder of

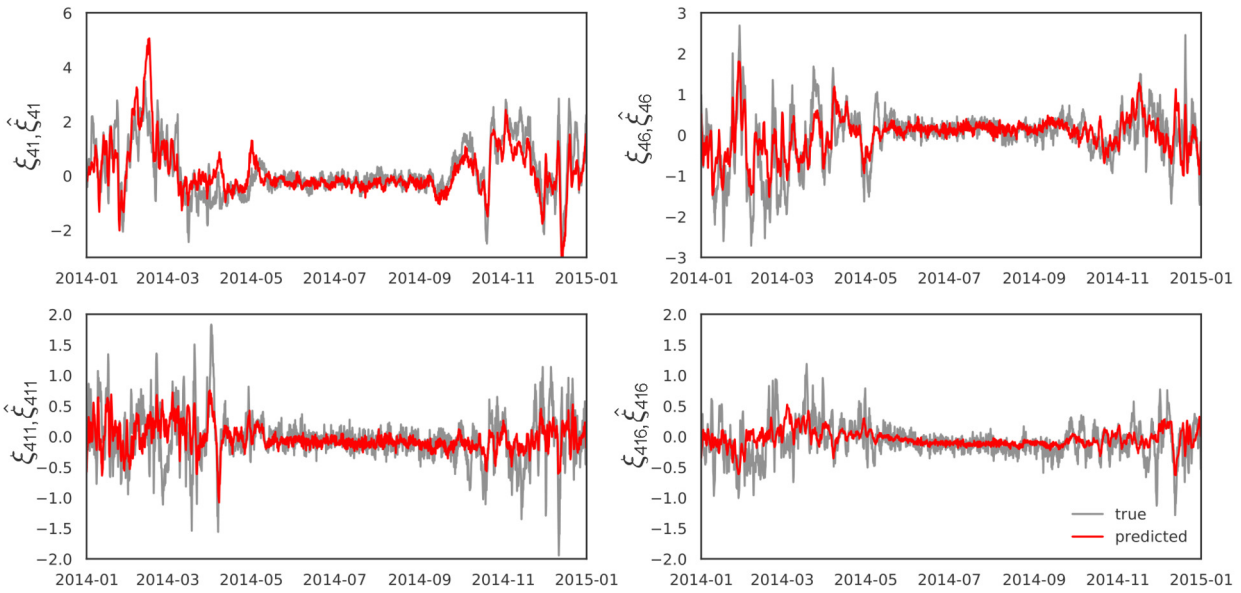


Fig. 7. Modeling of the mean, i.e. predictable part, for four EOF modes (1, 6, 11 and 16) of the level-4 wavelet coefficients for a segment not used for training (validation results). The high-energy modes are more predictable (in terms of percentage error) than the low-energy modes.

the target process, such as variance, change *slower* compared with the process itself. This assumption allows the estimation of the local-in-time statistics using a local-in-time averaging operator. This hypothesis relies on the fact that the small-scale processes, in general, vary much faster than the large scales which are the ones that primarily define the statistics of the small scales.

An important challenge, however, is how to choose the averaging window. In this work we use the averaging window which results in an averaged signal that has the same energy as the energy of the predictable part obtained by the machine-learning approximation. Note that with this each EOF mode will be characterized by a different averaging window, a time interval, w_{ik} (i is the wavelet level and k is the EOF index). This averaging window will depend directly on the skill of the machine-learning scheme to predict the EOF mode: EOF modes that are not approximated accurately by the machine-learning scheme will have larger error and therefore will require larger averaging window.

To select the window we first note the following properties for any EOF coefficient, ξ:

$$\lim_{w\to 0} \varphi_w * \xi = \xi,$$
$$\lim_{w\to \infty} \varphi_w * \xi = 0,$$
$$\|\varphi_{w_1} * \xi\| \geq \|\varphi_{w_2} * \xi\|, \quad \text{for } w_1 < w_2,$$

where $*$ denotes convolution, φ_w is a Gaussian window weight with center t and variance w^2. Therefore, for any value of the norm of the stochastic component, $\|\xi - \hat{\xi}\|$, within the allowable bounds, 0 and $\|\xi\|$, there is always an averaging window for which

$$\|\xi - \varphi_w * \xi\| = \|\xi - \hat{\xi}\|. \tag{14}$$

We use the last equality to implicitly define the averaging window for each EOF of the level-4 wavelet coefficients. Specifically, *the averaging window for each EOF, ξ_{4k}, is the one that results in a remainder of the original signal after averaging that has the same energy with the stochastic part, i.e. the remainder of the full signal after the predictable part is removed*. To enforce this condition we choose

$$w_{4k} = \text{argmin}_w \left| ||\xi_{4k} - \varphi_w * \xi_{4k}|| - ||\xi_{4k} - \hat{\xi}_{4k}|| \right|. \tag{15}$$

This selection ensures that the statistics estimated with an averaging window are energetically consistent with the error obtained by the machine-learning approximation. The results are displayed in Fig. 8 for EOF coefficients of level-4 wavelets. Because the low-energy modes have higher prediction errors (larger stochastic part), they also correspond to larger averaging windows. In the same figure we present for different EOF modes, the good agreement between the *mean-model* residual, $\xi_{4k} - \hat{\xi}_{4k}$ (orange curves), and the *smoothing* residual, $\xi_{4k} - \varphi_{w_{4k}} * \xi_{4k}$ (blue curves). This is a result of the window optimization with the criterion (15).

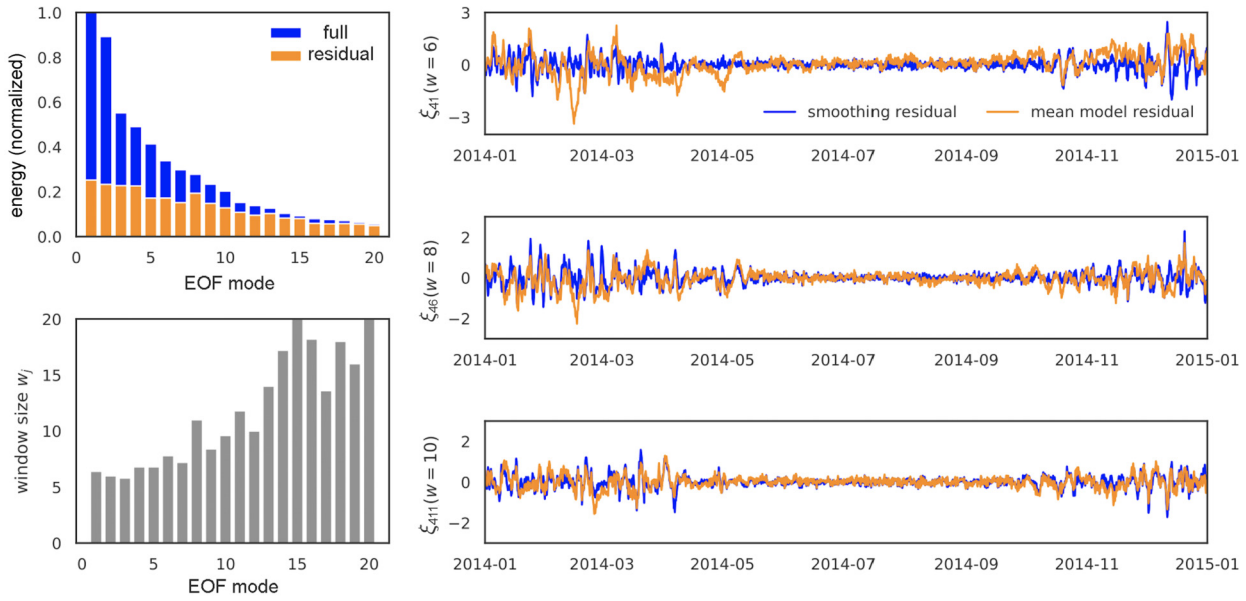


Fig. 8. Averaging windows for EOF modes of the level-4 wavelet coefficients. Top-left: Energy of the full EOF signal, $\|\xi_{4k}\|$, compared with norm of the mean-model residual, i.e. $\|\xi_{4k} - \hat{\xi}_{4k}\|$; Bottom-left: size of the averaging windows estimated with (15); Right: *mean-model* residual, $\xi_{4k} - \hat{\xi}_{4k}$, compared against *smoothing* the residual, $\xi_{4k} - \varphi_{w_{4k}} * \xi_{4k}$. Results are presented for a segment that has not been used for training (validation results).

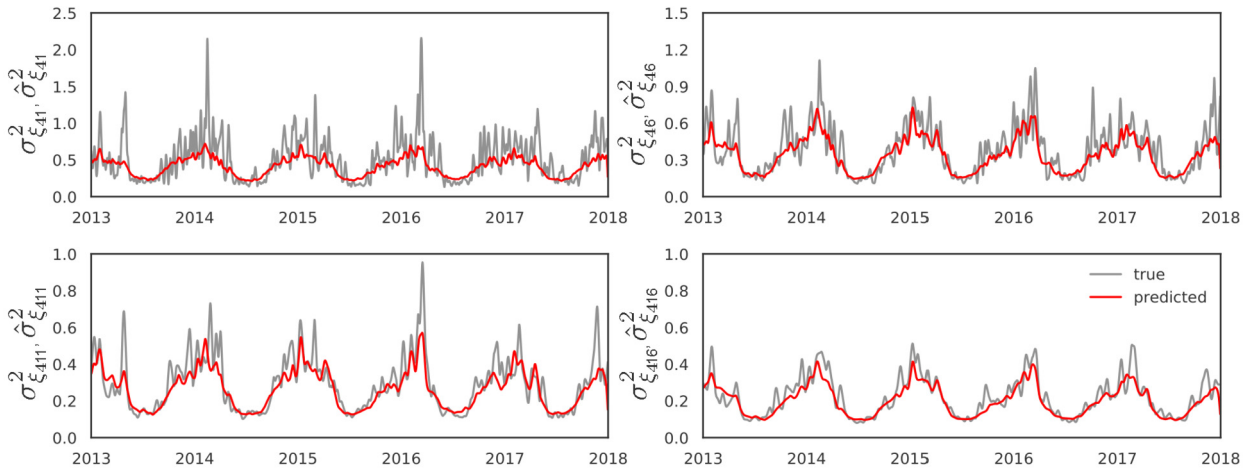


Fig. 9. Modeling of the non-stationary variance/standard deviation for four EOF modes (1, 6, 11 and 16) of the level-4 wavelet coefficients. The variance of the low-energy modes is captured more accurately. This is important as for the low-energy modes variance is the dominant factor for the overall mode energy.

The estimated averaging windows are used to approximate the local-in-time statistics of the deviation of $\xi_{4k}(t)$ from its local-in-time mean (predictable part: $\hat{\xi}_{4k}$) over time. Specifically, the variance of $\xi_{4k}(t)$ is computed as

$$\sigma^2_{\xi_{4k}}(t) = \varphi_{w_{4k}} * (\xi_{4k} - \hat{\xi}_{4k})^2 \equiv \int \varphi_{w_{4k}}(t' - t)\big(\xi_{4k}(t') - \hat{\xi}_{4k}(t')\big)^2 dt'. \tag{16}$$

4.4. Machine-Learning the variance of the stochastic part

The next step is the application of the second TCN to learn a mapping from $\mathbf{c}_{\mathcal{L}}$ to $\sigma^2_{\xi_{4k}}$, i.e. the variance sequence corresponding to ξ_{4j} and $\boldsymbol{G}_{\mathrm{m},4}$ calculated using Eq. (16) for $k = 1, \ldots, d_4$:

$$\hat{\sigma}^2_{\xi_{4k}}(t_n) = G_{\mathrm{var},4k}\Big(\boldsymbol{\Theta}_{\mathrm{var},4}; \mathbf{c}_{\mathcal{L}}(t_n), \mathbf{c}_{\mathcal{L}}(t_{n-1}), \ldots\Big). \tag{17}$$

The MNACC cost is used at this step so that the model makes a more aggressive attempt to capture the anomalies in the target. The resulted model $\boldsymbol{G}_{\text{var},4k}$ then constitutes an approximate *variance model* for each of the EOF coefficients, ξ_{4j}. We note that for this case a smoothing (non-trainable) layer is used to the output, in order to help the predicted local-in-time variance matches the expected output. Results and comparison between training data and machine-learning approximation are presented in Fig. 9. It is important to emphasize that for low-order EOF modes, where the prediction error is smaller, the averaging-window is smaller, and the resulted time series for the local-in-time variance is harder to approximate with the machine-learned map (17). On the other hand, for higher-order EOFs, where the prediction error is significant, the machine-learned map for the variance (Eq. (17)) is very accurate.

4.5. Estimation of the cross-covariance matrix

The local-in-time variance of each EOF is an essential ingredient for its stochastic description. However, it is not sufficient to provide a complete description from which we can obtain statistically-consistent realizations. To achieve this task we need an estimation of the local-in-time cross-covariance matrix:

$$R_{4jk}(\tau,t)=\int\sqrt{\varphi_{w_{4j}}(t'-t)\varphi_{w_{4k}}(t'-\tau-t)}\left(\xi_{4j}(t')-\hat{\xi}_{4j}(t')\right)\left(\xi_{4k}(t'-\tau)-\hat{\xi}_{4k}(t'-\tau)\right)dt', \tag{18}$$

where $j,k=1,\ldots,d_4$. This matrix encodes the covariance of EOF j at time t with EOF k at time $t-\tau$. Direct machine learning the full functional dependence is practically impossible due to the vast computational cost but also the lack of enough data to achieve this. We therefore approximate this matrix function by scaling the *global* cross-correlation coefficient, i.e. the cross-correlation coefficient computed using the full training time series, with the local-in-time standard deviation which has been estimated with the TCN of the previous step:

$$\hat{R}_{4jk}(\tau,t)=\rho_{4jk}(\tau)\hat{\sigma}_{\xi_{4j}}(t)\hat{\sigma}_{\xi_{4k}}(t-\tau), \tag{19}$$

with the global cross-correlation coefficient given by,

$$\rho_{4jk}(\tau)=R^{\text{global}}_{4jk}(\tau)\Big/\sqrt{R^{\text{global}}_{4jj}(0)R^{\text{global}}_{4kk}(0)}, \tag{20}$$

$$R^{\text{global}}_{4jk}(\tau)=\int\left(\xi_{4j}(t')-\hat{\xi}_{4j}(t')\right).\left(\xi_{4k}(t'-\tau)-\hat{\xi}_{4k}(t'-\tau)\right)dt'. \tag{21}$$

The underlying assumption for this approximation is that the *shape* of the cross-covariance function remains relatively invariant over time. One can improve this approximation by using seasonal estimations of the cross-correlation coefficient.

In Fig. 10, we plot the comparison for a few combinations of EOFs at a fixed combination of times. We present the empirical local truth for the cross-covariance function Eq. (18) and compare it against two estimates: 'estimate 1' is based on Eq. (19) with exact local-in-time variances, while 'estimate 2' is also based on Eq. (19) but with the approximated variances (obtained from the machine-learning scheme). For reference we also show the global cross-covariance function, $R^{\text{global}}_{4jk}(\tau)$. We observe reasonable agreement, especially for the covariance, $R_{4jj}(\tau)$. The cross-covariance typically has lower values and is therefore less important. It also appears to be less stationary in general but our model is able to offer a good baseline.

Collectively, the equations for the mean, (13), the averaging windows, (15), the local-in-time variance, (17), and the local-in-time cross-covariance, (19), provide a complete characterization of the underlying non-stationary, Gaussian stochastic process.

5. Samples-generation and validation

The next step involves drawing samples according to the modeled mean and cross-covariance matrix function. In particular, based on the assumption of Gaussian statistics for the stochastic part, we will obtain realizations for the small-scale coefficients that are conditioned on their past, as well as the obtained statistics from the large scales.

5.1. Sample generation algorithm

The sampling procedure proceeds in a sequential manner: let $\mathbf{m}_i$ and $\boldsymbol{\Sigma}_i$ be the mean and covariance of the sampled process up to step i, where (denoting as $\boldsymbol{\xi}$ the vector of EOF coefficients for a given wavelet level and $\hat{\mathbf{R}}=\{\hat{R}_{jk}\}_{j,k=1}^{d}$),

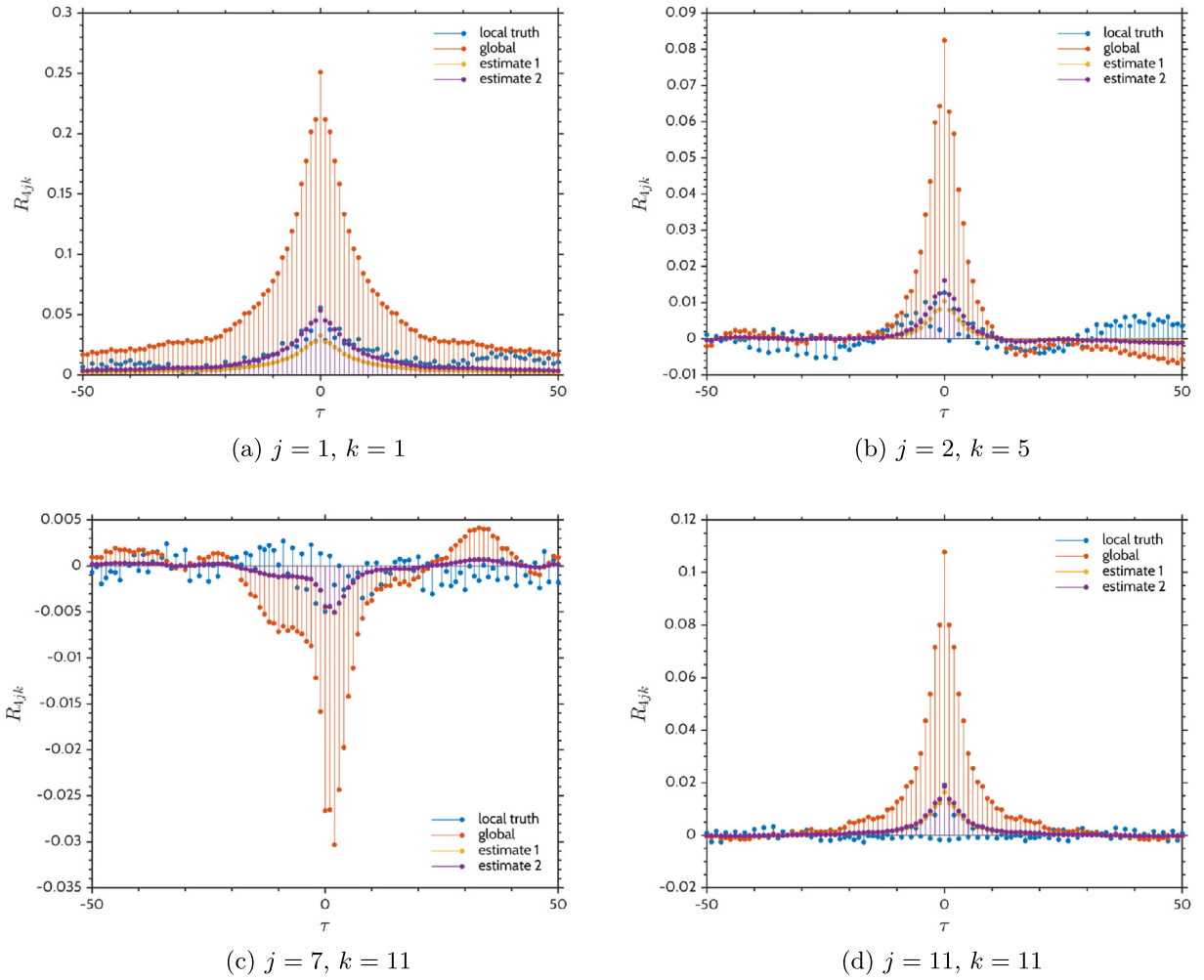


Fig. 10. Approximation of the non-stationary cross-covariance, $R_{4jk}(\tau, t)$. Indices j, k indicate the EOF modes involved in each cross-covariance shown. Legend: 'local truth' (blue) corresponds to (18); 'global' represents R_4^{global}, i.e. the cross-covariance computed from the entire signal (21); 'Estimate 1' represents approximation (19) normalized with *true* local standard deviation (i.e. square root of (16)); 'Estimate 2' is the same approximation normalized with the *predicted* standard deviation - (19) exactly.

$$\mathbf{m}_i = \left[\hat{\boldsymbol{\xi}}(t_0^*) \quad \hat{\boldsymbol{\xi}}(t_1^*) \quad \dots \quad \hat{\boldsymbol{\xi}}(t_i^*)\right],$$

$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \hat{\mathbf{R}}(0,t_0^*) & \hat{\mathbf{R}}(\Delta,t_1^*)^T & \hat{\mathbf{R}}(2\Delta,t_2^*)^T & \cdots & \hat{\mathbf{R}}(i\Delta,t_i^*)^T \\ \hat{\mathbf{R}}(\Delta,t_1^*) & \hat{\mathbf{R}}(0,t_1^*) & \hat{\mathbf{R}}(\Delta,t_2^*)^T & \vdots & \vdots \\ \hat{\mathbf{R}}(2\Delta,t_2^*) & \hat{\mathbf{R}}(\Delta,t_2^*) & \hat{\mathbf{R}}(0,t_2^*) & \vdots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \vdots \\ \hat{\mathbf{R}}(i\Delta,t_i^*) & \cdots & \cdots & \cdots & \hat{\mathbf{R}}(0,t_i^*) \end{bmatrix}. \tag{22}$$

Here Δ is the sampling interval (same as training data). The structure of $\boldsymbol{\Sigma}_i^*$ is such that the jk^{th} block represents the covariance between $\boldsymbol{\xi}_j^*$ and $\boldsymbol{\xi}_k^*$. The conditional mean and covariance for $\boldsymbol{\xi}_{i+1}^*$ is given by

$$\begin{aligned} \mathbf{m}(i+1|0,\dots,i) &= \boldsymbol{\xi}(t_{i+1}^*) + \boldsymbol{\Sigma}_{i+1,i}\boldsymbol{\Sigma}_i^{-1}(\mathbf{s}_i - \mathbf{m}_i), \\ \boldsymbol{\Sigma}(i+1|0,\dots,i) &= \hat{\mathbf{R}}(0,t_{i+1}^*) - \boldsymbol{\Sigma}_{i+1,i}\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\Sigma}_{i+1,i}^T, \end{aligned} \tag{23}$$

where

$$\boldsymbol{\Sigma}_{i+1,i} = \left[\hat{\mathbf{R}}((i+1)\Delta,t_{i+1}^*) \quad \hat{\mathbf{R}}(i\Delta,t_{i+1}^*) \quad \cdots \quad \hat{\mathbf{R}}(\Delta,t_{i+1}^*)\right], \tag{24}$$

and $\mathbf{s}_i$ denotes the vector of samples that has been drawn up to step i. The conditioning is kept for the most recent 20 samples only (entries prior to t^*_{i-20} are truncated) to prevent the size of $\mathbf{m}$ and $\mathbf{\Sigma}$ from constantly growing with i. This truncation is sensible as for small scales the decorrelation time is typically short and the samples very far back have negligible covariance with the current time. The full sampling procedure is described in Algorithm 1.

Algorithm 1: Sampling of non-stationary Gaussian process (for a fixed wavelet level).

Input: large-scale wavelet coefficients $\mathbf{c}^*_{\mathcal{L}} = \{\mathbf{c}_{\mathcal{L}}(t^*_0), ..., \mathbf{c}_{\mathcal{L}}(t^*_{P-1})\}$,
Data: mean model $\mathbf{G}_{\mathrm{m}}$, variance model $\mathbf{G}_{\mathrm{var}}$, global cross-correlation function $\boldsymbol{\rho}(\cdot)$
Result: sampled EOF coefficients for small-scale wavelets of a given level $\boldsymbol{\xi}^* = \{\boldsymbol{\xi}(t^*_0), ..., \boldsymbol{\xi}(t^*_{P-1})\}$

Compute $\hat{\boldsymbol{\xi}}^* = \mathbf{G}_{\mathrm{m}}(\mathbf{c}^*_{\mathcal{L}}) = \{\hat{\boldsymbol{\xi}}(t^*_0), ..., \hat{\boldsymbol{\xi}}(t^*_{P-1})\}$, $\hat{\boldsymbol{\sigma}}^* = \mathbf{G}_{\mathrm{var}}(\mathbf{c}^*_{\mathcal{L}})$;
Construct $\hat{\boldsymbol{\sigma}}^*$ using (19) and then $\hat{\mathbf{R}}(\cdot,\cdot)$ from $\boldsymbol{\rho}(\cdot)$ and $\hat{\boldsymbol{\sigma}}^*$;
Draw sample $\boldsymbol{\xi}(t^*_0) \sim \mathcal{N}\big(\hat{\boldsymbol{\xi}}(t^*_0), \hat{\mathbf{R}}(0, t^*_0)\big)$ and record $\boldsymbol{\xi}^* = \{\boldsymbol{\xi}(t^*_0)\}$;
Set $\mathbf{m}_i = [\hat{\boldsymbol{\xi}}(t^*_0)]$, $\mathbf{\Sigma}_i = [\hat{\mathbf{R}}(0, t^*_0)]$, $\mathbf{s}_i = [\boldsymbol{\xi}(t^*_0)]$;
for $i = 0, ..., P-2$ **do**
  Assemble $\mathbf{\Sigma}_{i+1,i}$ as in (24);
  Compute $\mathbf{m}$ and $\mathbf{\Sigma}$ as in (23);
  Draw sample $\boldsymbol{\xi}(t^*_{i+1}) \sim \mathcal{N}(\mathbf{m}, \mathbf{\Sigma})$ and append to $\boldsymbol{\xi}^*$;
  Update $\mathbf{m}_i = [\mathbf{m}_i, \hat{\boldsymbol{\xi}}(t^*_{i+1})]$, $\mathbf{s}_i = [\mathbf{s}_i, \boldsymbol{\xi}(t^*_{i+1})]$;
  Update $\mathbf{\Sigma}_i = [\mathbf{\Sigma}_i, \mathbf{\Sigma}^T_{i+1,i}; \mathbf{\Sigma}_{i+1,i}, \hat{\mathbf{R}}(0, t^*_{i+1})]$; `// keep 20 blocks at most after updates`
end

5.2. Application to the atmospheric data set

A number of samples for the level-4 EOF coefficients are drawn using the described procedure (shown in Fig. 11). Subsequently, the generated samples are transformed (from the EOF space) to the wavelet space and then to the physical domain. Fig. 12 shows a number of snapshots from the drawn samples corresponding to different times in 2013, along with the corresponding conditioning vorticity field and the actual reanalysis data. We can observe that the samples reflect a decent number of small-scale features that differ from the actual reanalysis data.

We are also able to easily extend the framework to level-5 EOF coefficients using a similar pipeline - constructing the mean and cross-covariance model for level-5 coefficients based on the level 1-4 wavelet coefficients (obtained from the reanalysis data, i.e. training data). The generated stochastic model is then utilized conditioned on the level 1-4 wavelet coefficients. Note that the levels 1-3 are fixed (large scales) while level-4 wavelet coefficients are those generated from the previous model (which has also been conditioned on levels 1-3). In this way, we effectively generate complete level-5 vorticity fields using only the first 3 levels for conditioning. The resulting snapshots, consisting of all levels (1-5) superimposed, are illustrated in Fig. 13. The generated levels (4-5) are shown without the levels 1-3 in Fig. 14. For comparison, we also include the samples obtained by using the *true* (i.e. obtained from reanalysis data) level-4 data for conditioning (in column labeled 'sample 1'). As expected, the vorticity fields appear closer to the reanalysis data than those generated with sampled level-4 coefficients.

5.3. Validation

To systematically assess the model results, we revert the sampled time series back to the physical space and first evaluate the results at fixed locations using a number of metrics. In Fig. 15, we show three sampled time series in comparison with both the corresponding conditioning (levels 1-3) and full (levels 1-5) reanalysis data for three major cities of Europe: Berlin, London and Rome during 2013. The sampled time series exhibit important fluctuations away from the large scales in a way that is consistent with the full reanalysis. In Fig. 16, we show the probability density function (PDF) of the 5-year (2013-2017) vorticity signal at the same cities. The difference in the probability density that resulted from having the additional small-scaled levels (4 and 5) is reflected almost perfectly in the samples. We can also observe the very good agreement between the tails regions. We note that since it is relatively inexpensive to sample multiple trajectories using the SMaL framework, the PDF can be estimated very effectively allowing for the computation of extreme event catalogs.

The (temporal) power spectral density (PSD) also agrees well between the sample and reanalysis data (Fig. 17 shows the PSD for Berlin; patterns for other cities are similar). Finally, we also examine the spatial distribution in terms of the (spatial) radially-averaged power density spectrum of vorticity (i.e. energy against the magnitude of the 2D wavenumber $|k| = \sqrt{k^2_{\mathrm{lat}} + k^2_{\mathrm{lon}}}$). As shown in Fig. 18, there is very good agreement between the samples and the reanalysis as well. Note that there is a systematic underestimation of the spatial energy density by the SMaL framework. This agrees as an order of magnitude with energy truncated by the EOF expansion (Fig. 3), which is close to 6%. This is an inaccuracy that can be improved by including more EOF modes in our analysis. For these higher-order EOF modes one may adopt simpler parameterizations focusing on the stochastic part as the deterministic part has very low energy (Fig. 4).

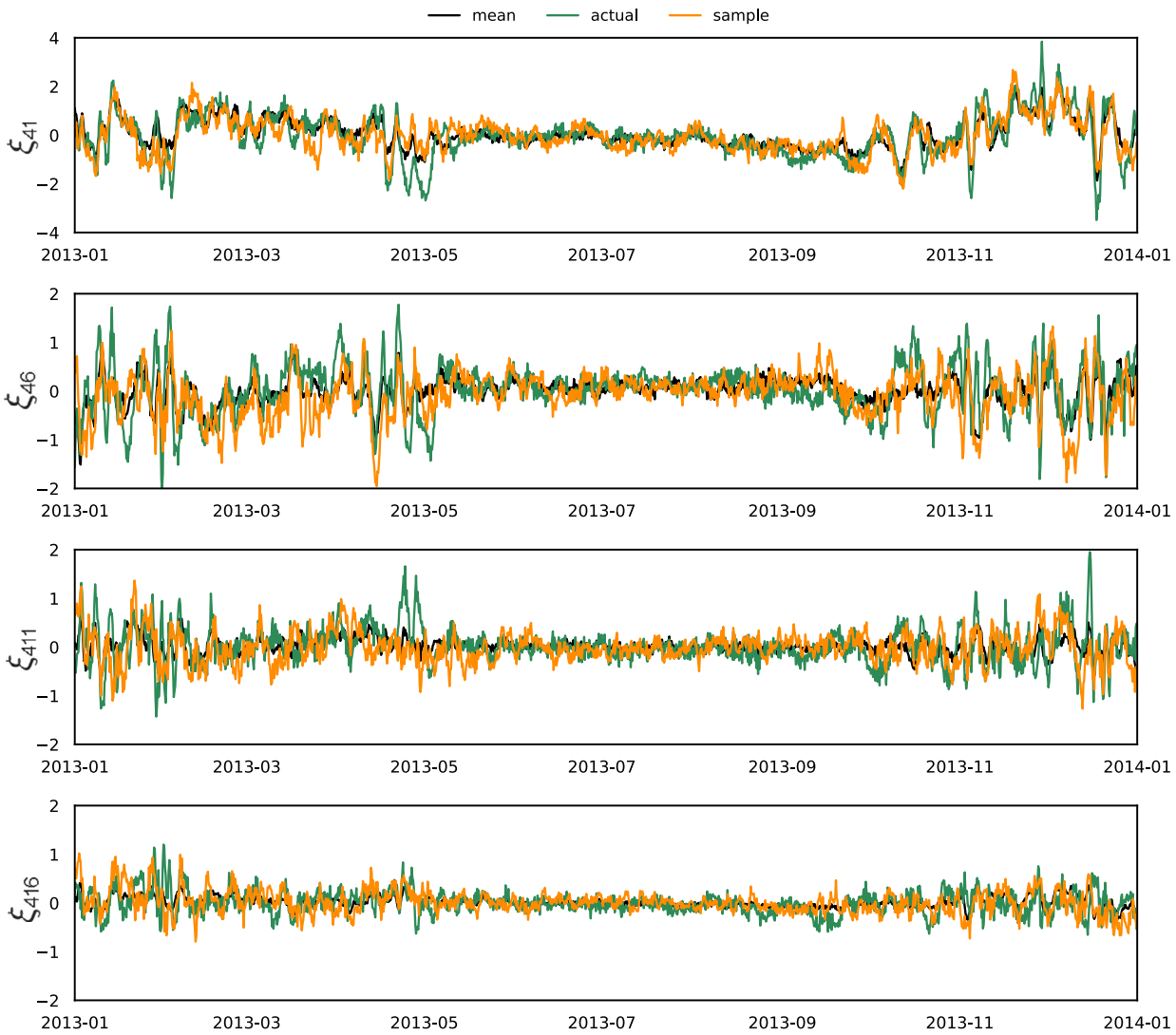


Fig. 11. Sample time series in EOF space, compared with the data and mean prediction, for modes 1, 6, 11, 16 in 2013. Black curve is the machine learning prediction of the mean; green is the actual realization and orange is the generated realization by superimposing to the mean prediction a random sample generated from the machine-learned variance.

6. Conclusions

We have introduced a data-driven strategy, the Stochastic Machine-Learning (SMaL) framework, for explicitly parameterizing small-scale atmospheric features in terms of the corresponding large-scale observations, taking into account the spatially inhomogeneous character of the problem and the fact that there is only a single realization available for each spatial location that can be used for training. Our framework represents the large- and small-scale features using a versatile wavelet basis and relies on temporal convolutional networks (TCN), a history-incorporating deep learning architecture, to machine learn the parameterization. More specifically, the target small-scale time series is systematically divided into a predictable and an unpredictable component, through empirical orthogonal function (EOF) decomposition and model learning. First, TCN is applied directly to capture the portion of the small scales which are immediately predictable from the large scales. The residuals are then modeled stochastically by using additional TCN to approximate the non-stationary statistics, quantified with carefully selected, local-in-time-averaging operators. This procedure allows one to generate an ensemble of small-scale time series realizations which are consistent with the given large scales.

We illustrate and validate the SMaL parameterization framework through application to reanalysis data, where we parameterize the small-scaled vorticity features in Western Europe, in terms of the large-scaled vorticity in the same area. The results demonstrate that physical-space patterns reflected in the samples agree very well with the ground truth small-scaled features from reanalysis, in terms of temporal and spatial spectra, as well as the density distribution at fixed locations.

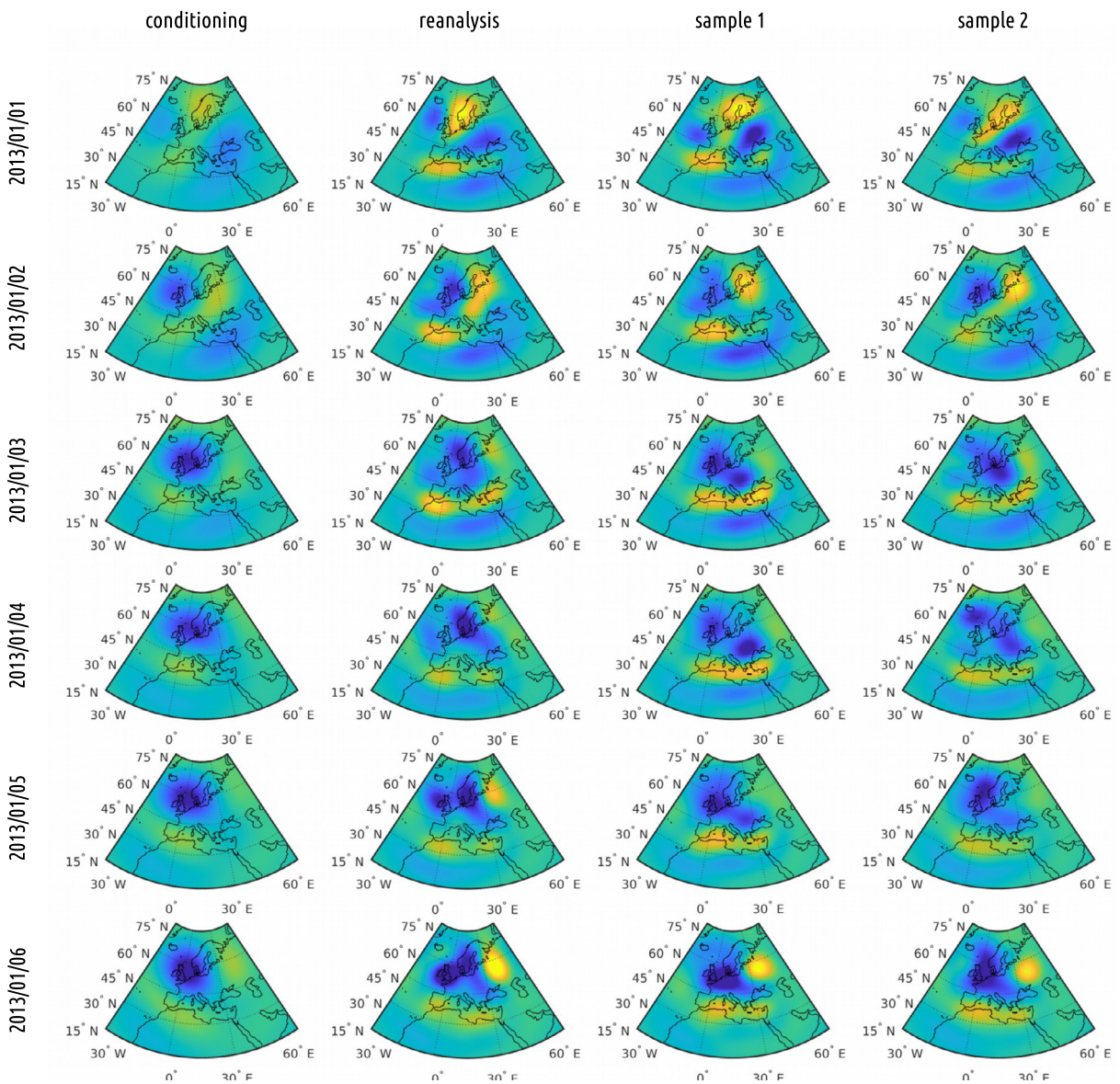


Fig. 12. Vorticity snapshots for the conditioning data (levels 1-3 only), reanalysis data (levels 1-4), and two drawn samples (levels 1-4).

Combined with coarse-scale climate models, we believe that the introduced framework paves the way for the probabilistic quantification of extreme events and the development of extreme event catalogs with much smaller computational cost. In addition, the developed ideas can be used for the development of parsimonious filtering schemes for short-term weather prediction and this is a direction that we plan to pursue in the future.

CRediT authorship contribution statement

Zhong Yi Wan: Investigation, Methodology, Software, Writing – original draft. **Boyko Dodov:** Conceptualization, Data curation, Investigation, Methodology, Writing – review & editing. **Christian Lessig:** Data curation, Investigation, Methodology, Software, Writing – review & editing. **Henk Dijkstra:** Investigation, Methodology, Writing – review & editing. **Themistoklis P. Sapsis:** Conceptualization, Funding acquisition, Investigation, Methodology, Software, Supervision, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

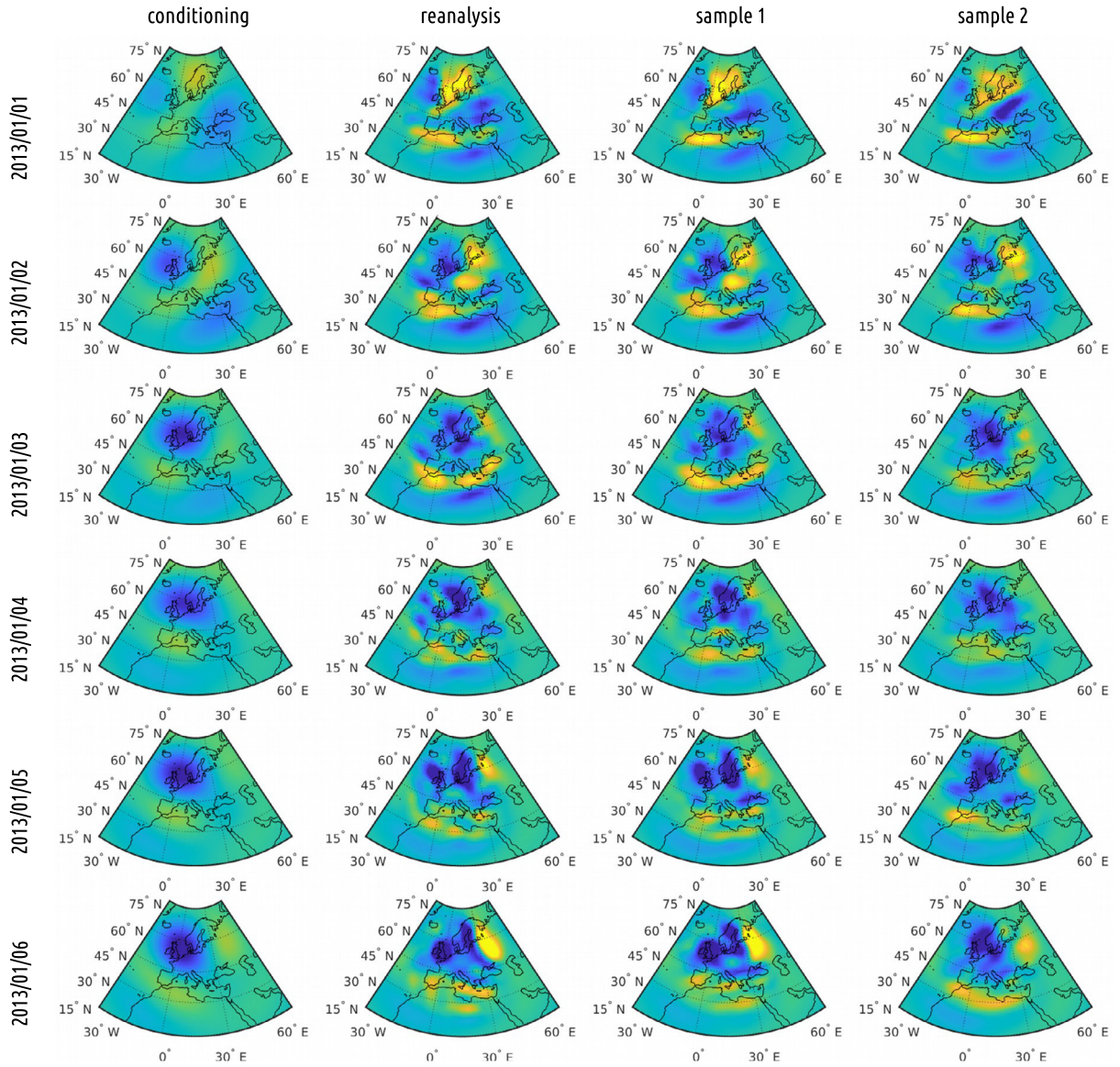


Fig. 13. Vorticity snapshots for the conditioning data (levels 1-3 only), reanalysis data (levels 1-5), and two drawn samples (levels 1-5). For 'sample 1', the level 5 sample is conditioned on the *reanalysis* level-4 coefficients, i.e. the true level-4. The level 5 of 'sample 2' is conditioned on *sampled* level-4 coefficients.

Acknowledgements

The authors appreciate several stimulating discussions with Dr. Zoltan Toth, NOAA. Funding by AIR Worldwide and Verisk Analytics is gratefully acknowledged.

Appendix A. Optimizing model architecture

In this section, we provide additional supporting evidence for choosing the particular model architecture described in section 3. Specifically, we compare the performance of models with different architectures or trained with different loss functions, measured by the same evaluation metrics: mean squared error (MSE), mean absolute error (MAE) and the mean negative anomaly correlation coefficient (MNACC) on the validation data set (using wavelet levels 1-3 to predict the level 4 EOF modes for 2009-2017).

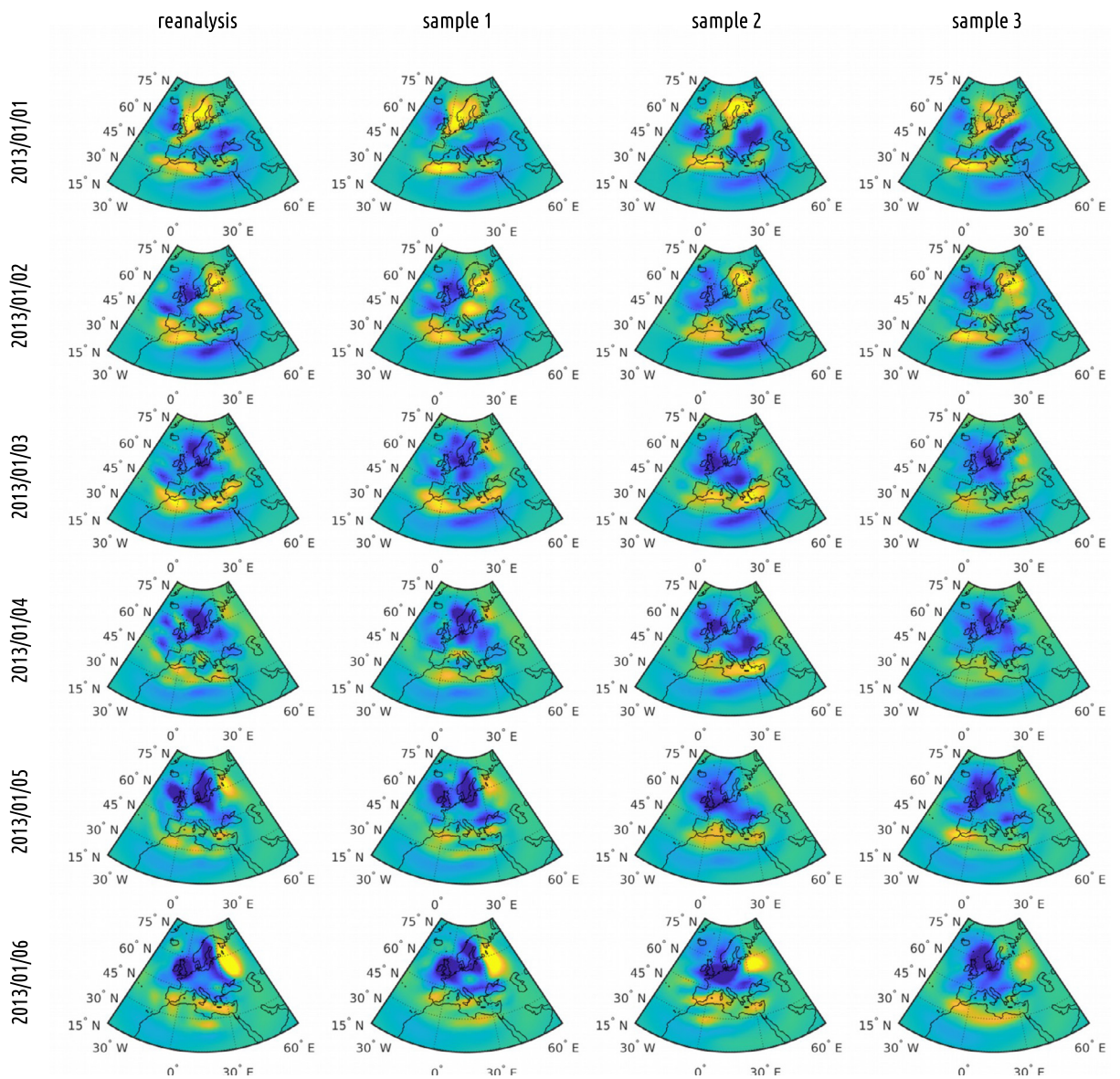


Fig. 14. Vorticity snapshots showing levels 4 and 5 only. For 'sample 1', the level-5 sample is conditioned on the *reanalysis* level-4 coefficients. The level-5 of 'sample 2' and 'sample 3' is conditioned on *sampled* level-4 coefficients.

A.1. Reference signal

Table 1 compares the performance for using different loss functions to learn the deterministic component of the small-scaled vorticity time series. Unsurprisingly, the model always does the best on the metric for which it is explicitly optimized. The use of reference with the MNACC loss (recall that without the reference it becomes mean negative Pearson correlation coefficient) does not introduce any noticeable advantage. This is likely due to the fact that the reference signal for optimizing MNACC, obtained by taking the same-day-of-the-year historical average, does not differ significant from zero (see Fig. 19a), offering little additional information to improve training.

The situation is different for learning the non-stationary standard deviation of the stochastic component, where the corresponding time series exhibits strong annual periodicity. In this case, the reference signal in itself provides a decent baseline estimate for the target (see 'ref' signal in Fig. 19b). Including this information allows the model to directly focus on predicting the deviation from the reference, which noticeably improves the performance (compare first and second block in Table 2).

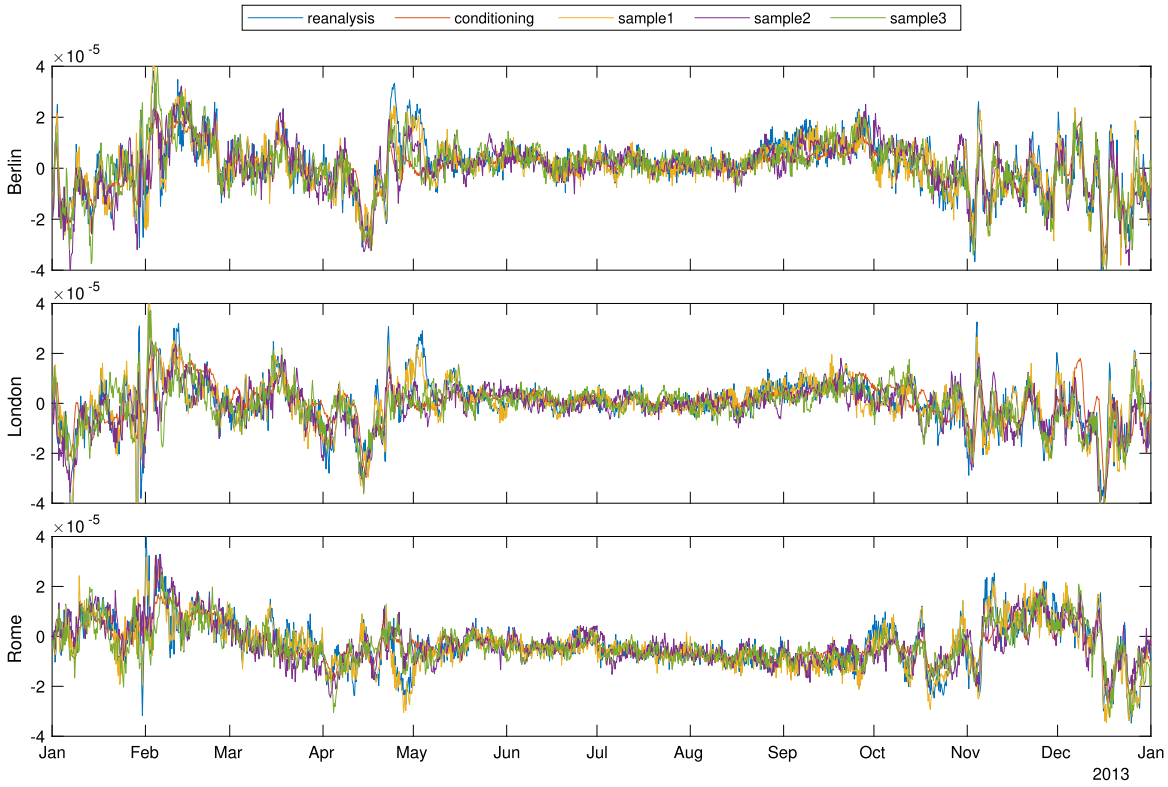


Fig. 15. Time series of vorticity for Berlin, London and Rome showing reanalysis data, conditioning and 3 samples drawn (all 5 levels are included) for year 2013.

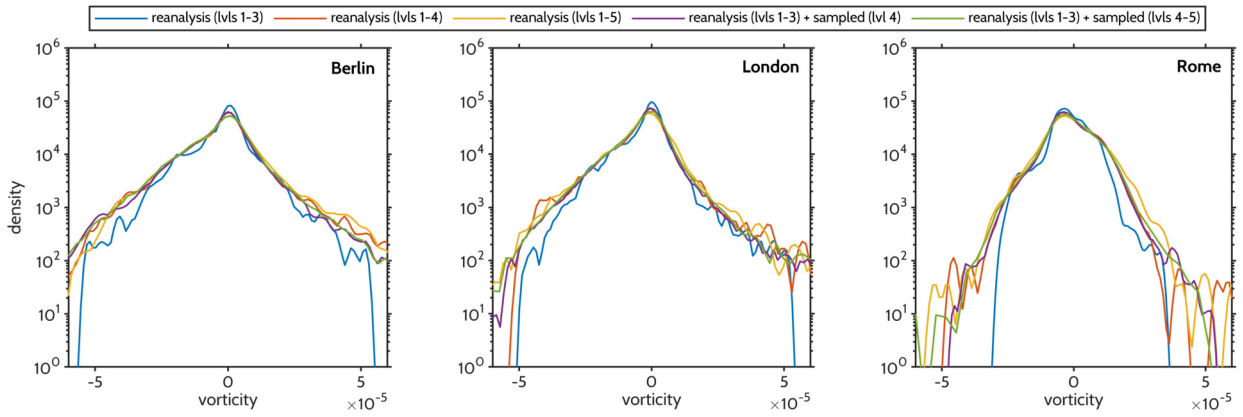


Fig. 16. Probability density function (pdf) of the vorticity at three major European cities (Berlin, London and Rome). The densities are computed for the 5-year period from 2013 to 2017. 9 separate sample trajectories are used for those that involve sampling.

Table 1
Model performance for learning the deterministic component with different loss functions.

Optimized loss	Use reference	Evaluation metric		
		MSE	MAE	MNACC
MSE	No	0.149	0.264	−0.539
MAE	No	0.153	0.259	−0.509
MNACC	No	0.157	0.269	−0.552
MNACC	Yes	0.156	0.270	−0.550

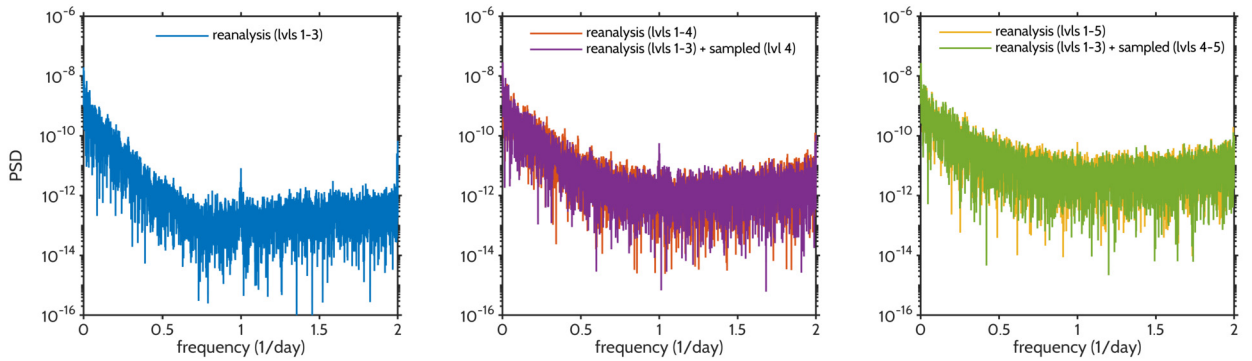


Fig. 17. Power spectral density for the vorticity signal at Berlin. Results are computed for the 5-year period from 2013 to 2017. A single sample trajectory is used for those that involve sampling. Patterns are similar for other cities.

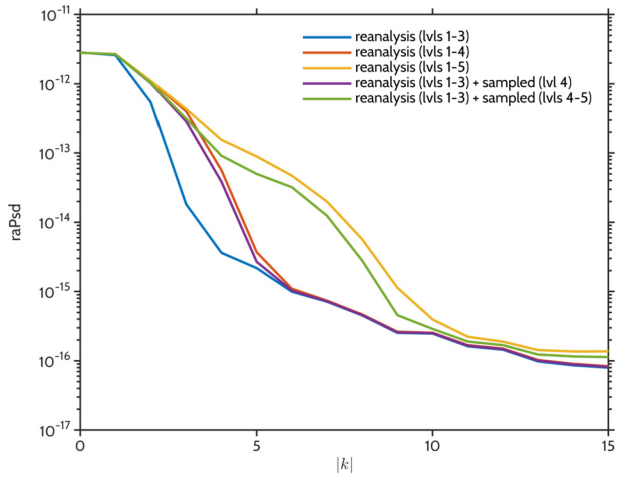


Fig. 18. Radially averaged spatial spectrum of vorticity. Results are computed for the 5-year period from 2013 to 2017. 9 separate sample trajectories are used for those that involve sampling.

A.2. Norm-based vs. correlation-based loss

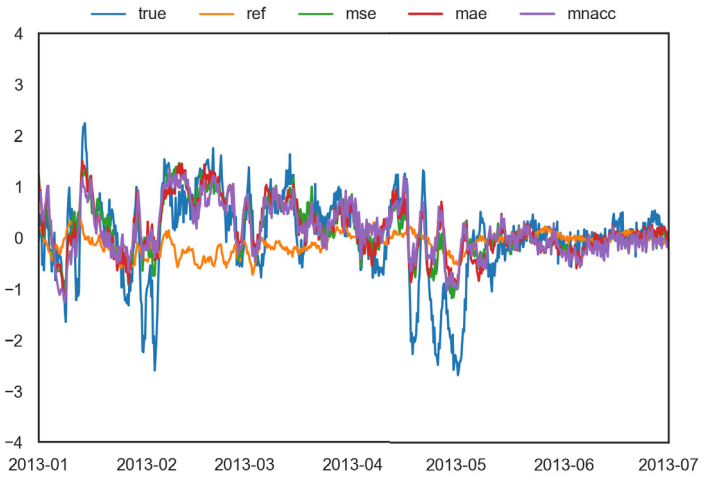
In Fig. 19b, it is also easy to visualize the difference between norm-based (MSE and MAE) and correlation-based (PCC, MNACC) loss functions. Specifically, the latter prefers predictions that have high covariance with truth, paying much less attention to penalizing predictions far away from the truth in terms of the norm distance. In consequence, models learned using a correlation-based loss tend to make more aggressive attempts to follow peaks in the output because it would not contribute greatly to the overall cost as long as the truth and prediction are on the same side of their respective means.

A.3. Output smoothing

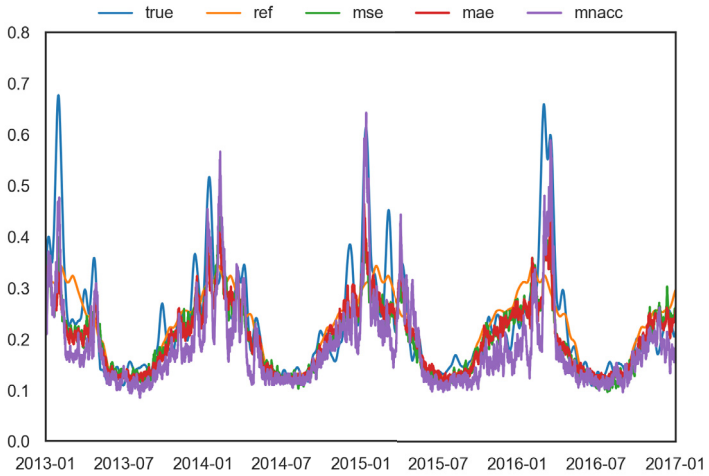
We observe also that the combination of MNACC loss and the use of output smoothing is able to provide significant boost to the model performance on all evaluation metrics (last block in Table 2). The smoothing operation applies the same Gaussian kernel (with width obtained from (15)) that is used to compute the ground truth standard deviation, effectively forcing the final model output to the correct smoothness (as opposed to making the model learn it). In addition, the model is tolerated to make short, bursty predictions as they can be mitigated by smoothing when loss is computed. Interestingly, MAE and MSE loss do not benefit from this smoothing operation.

A.4. Balancing correlation and calibration

Table 3 presents the effect of the γ hyperparameter in (12). γ balances the correlation and calibration in the MNACC loss. Large values of γ make the learning process prioritize the calibration MSE while small values lead to heavy emphasis on the correlation. Based on the results of numerical experiments, it is clear that $\gamma = 10$ is an ideal value to use. However, the choice of γ may not apply to other problem settings since the range of calibration MSE (arbitrary) relative to the correlation (always between -1 and 1) is problem-dependent.



(a) Deterministic component of the vorticity.



(b) Non-stationary standard deviation of the stochastic component of vorticity.

Fig. 19. Example prediction of vorticity by models trained with different loss functions.

Table 2
Model performance for learning the non-stationary standard deviation of the stochastic component with different loss functions, use of reference signal and output smoothing.

Optimized loss	Use reference	Output smoothing	Evaluation metric		
			MSE	MAE	MNACC
MSE	No	No	1.95e-2	8.43e-2	−0.331
MAE	No	No	1.99e-2	8.93e-2	−0.344
MNACC	No	No	2.13e-2	9.30e-2	−0.363
MSE	Yes	No	1.46e-2	7.23e-2	−0.352
MAE	Yes	No	1.51e-2	7.13e-2	−0.349
MNACC	Yes	No	1.61e-2	7.67e-2	−0.374
MSE	Yes	Yes	1.74e-2	7.97e-2	−0.356
MAE	Yes	Yes	1.79e-2	7.67e-2	−0.371
MNACC	Yes	Yes	1.48e-2	7.20e-2	−0.390

Table 3
Model performance for learning the non-stationary standard deviation of the stochastic component with different γ parameter in (12). Output smoothing is always applied.

Optimized loss	γ	Evaluation metric		
		MSE	MAE	MNACC
MNACC	0.01	4.77e-2	1.43e-1	−0.407
MNACC	0.1	1.92e-2	8.83e-2	−0.385
MNACC	1	1.56e-2	7.47e-2	−0.376
MNACC	**10**	**1.48e-2**	**7.20e-2**	**−0.390**
MNACC	100	1.85e-2	8.43e-2	−0.363

References

[1] S. Bai, J.Z. Kolter, V. Koltun, An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling, 2018.
[2] J. Baño-Medina, R. Manzanas, J.M. Gutierrez, Configuration and intercomparison of deep learning neural models for statistical downscaling, Geosci. Model Dev. (2020).
[3] P. Berrisford, D.P. Dee, P. Poli, R. Brugge, M. Fielding, M. Fuentes, P.W. Kallberg, S. Kobayashi, S. Uppala, A. Simmons, The ERA-Interim archive Version 2.0, ECMWF Report, 2011.
[4] G.J. Boer, Predictability as a function of scale, Atmos.-Ocean 41 (3) (2003) 203–215.
[5] N. Chen, Y. Li, Efficient nonlinear filtering, smoothing, forward and backward sampling algorithms for partially observed complex turbulent nonlinear dynamical systems with intermittency and extreme events, 2021, submitted for publication.
[6] N. Chen, A.J. Majda, Filtering nonlinear turbulent dynamical systems through conditional Gaussian statistics, Mon. Weather Rev. 144 (12) (2016) 4885–4917.
[7] N. Chen, A.J. Majda, Conditional Gaussian systems for multiscale nonlinear stochastic systems: prediction, state estimation and uncertainty quantification, Entropy 20 (7) (2018) 1–80.
[8] C.C. Da Silva, C. Lessig, B. Dodov, H. Dijkstra, T. Sapsis, A local spectral exterior calculus for the sphere and application to the shallow water equation, 2020, submitted for publication, arXiv:2005.03598.
[9] N.B. Erichson, L. Mathelin, Z. Yao, S.L. Brunton, M.W. Mahoney, J.N. Kutz, Shallow learning for fluid flow reconstruction with limited sensors and limited data, Feb. 2020, arXiv.
[10] R. Everson, L. Sirovich, Karhunen–Loève procedure for gappy data, J. Opt. Soc. Am. A 12 (8) (Aug. 1995) 1657.
[11] K. Fukami, K. Fukagata, K. Taira, Super-resolution reconstruction of turbulent flows with machine learning, J. Fluid Mech. 870 (Jul. 2019) 106–120.
[12] S. Gangopadhyay, M. Clark, B. Rajagopalan, Statistical downscaling using K-nearest neighbors, Water Resour. Res. (2005).
[13] W.A. Gardner, A. Napolitano, L. Paura, Cyclostationarity: half a century of research, 2006.
[14] P.R. Gent, J.C. McWilliams, Isopycnal mixing in ocean circulation models, J. Phys. Oceanogr. 20 (1) (1990) 150–155.
[15] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[16] I. Grooms, A.J. Majda, Efficient stochastic superparameterization for geophysical turbulence, Proc. Natl. Acad. Sci. USA 110 (12) (2013) 4464–4469.
[17] I. Grooms, A.J. Majda, Stochastic superparameterization in a one-dimensional model for wave turbulence, Commun. Math. Sci. 12 (3) (2014) 509–525.
[18] I. Grooms, A.J. Majda, K.S. Smith, Stochastic superparameterization in a quasigeostrophic model of the Antarctic circumpolar current, Ocean Model. 85 (2015) 1–15.
[19] J.M. Gutiérrez, D. Maraun, M. Widmann, R. Huth, E. Hertig, R. Benestad, O. Roessler, J. Wibig, R. Wilcke, S. Kotlarski, D. San Martín, S. Herrera, J. Bedia, A. Casanueva, R. Manzanas, M. Iturbide, M. Vrac, M. Dubrovsky, J. Ribalaygua, J. Pórtoles, O. Räty, J. Räisänen, B. Hingray, D. Raynaud, M.J. Casado, P. Ramos, T. Zerenner, M. Turco, T. Bosshard, P. Štěpánek, J. Bartholy, R. Pongracz, D.E. Keller, A.M. Fischer, R.M. Cardoso, P.M. Soares, B. Czernecki, C. Pagé, An intercomparison of a large ensemble of statistical downscaling methods over Europe: results from the VALUE perfect predictor cross-validation experiment, Int. J. Climatol. (2019).
[20] A. Hannachi, I.T. Jolliffe, D.B. Stephenson, Empirical Orthogonal Functions and Related Techniques in Atmospheric Science: A Review, 2007.
[21] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: The IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Jun. 2016.
[22] C. Hutengs, M. Vohland, Downscaling land surface temperatures at regional scales with random forest regression, Remote Sens. Environ. (2016).
[23] D.I. Jeong, A. St-Hilaire, T.B. Ouarda, P. Gachon, Multisite statistical downscaling model for daily precipitation combined by multivariate multiple linear regression and stochastic weather generator, Clim. Change (2012).
[24] Y. Lee, A.J. Majda, D. Qi, Stochastic superparameterization and multiscale filtering of turbulent tracers, Multiscale Model. Simul. 15 (2017) 215–234.
[25] S. Li, M. Deng, J. Lee, A. Sinha, G. Barbastathis, Imaging through glass diffusers using densely connected convolutional networks, Optica 5 (7) (Jul. 2018) 803–813.
[26] D. Maraun, M. Widmann, J.M. Gutiérrez, S. Kotlarski, R.E. Chandler, E. Hertig, J. Wibig, R. Huth, R.A. Wilcke, VALUE: a framework to validate downscaling approaches for climate change studies, Earth's Future 3 (1) (2015) 1–14.
[27] Q. Miao, B. Pan, H. Wang, K. Hsu, S. Sorooshian, Improving Monsoon Precipitation Prediction Using Combined Convolutional and Long Short Term Memory Neural Network, Water, Switzerland, 2019.
[28] M. Milano, P. Koumoutsakos, Neural network modeling for near wall turbulent flow, J. Comput. Phys. 182 (1) (2002) 1–26.
[29] S. Misra, S. Sarkar, P. Mitra, Statistical downscaling of precipitation using long short-term memory recurrent neural networks, Theor. Appl. Climatol. (2018).
[30] S.H. Pour, S. Shahid, E.S. Chung, A hybrid model for statistical downscaling of daily rainfall, in: Procedia Engineering, 2016.
[31] N.C. Privé, R.M. Errico, Spectral analysis of forecast error investigated with an observing system simulation experiment, Tellus, Ser. A Dyn. Meteorol. Oceanogr. 67 (1) (2015) 25977.
[32] V. Resseguier, E. Mémin, B. Chapron, Geophysical flows under location uncertainty, Part I. Random transport and general models, Geophys. Astrophys. Fluid Dyn. 111 (3) (Apr. 2017) 149–176.
[33] V. Resseguier, E. Mémin, B. Chapron, Geophysical flows under location uncertainty, Part II. Quasi-geostrophy and efficient ensemble spreading, Geophys. Astrophys. Fluid Dyn. 111 (3) (Apr. 2017) 177–208.
[34] V. Resseguier, E. Mémin, B. Chapron, Geophysical flows under location uncertainty, Part III. SQG and frontal dynamics under strong turbulence conditions, Geophys. Astrophys. Fluid Dyn. 111 (3) (Apr. 2017) 209–227.
[35] D.A. Sachindra, S. Kanae, Machine learning for downscaling: the use of parallel multiple populations in genetic programming, Stoch. Environ. Res. Risk Assess. (2019).

[36] T. Salimans, D.P. Kingma, Weight normalization: a simple reparameterization to accelerate training of deep neural networks, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, Red Hook, NY, USA, Curran Associates Inc., 2016, pp. 901–909.
[37] J.T. Schoof, S.C. Pryor, Downscaling temperature and precipitation: a comparison of regression-based methods and artificial neural networks, Int. J. Climatol. (2001).
[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (56) (2014) 1929–1958.
[39] K. Stengel, A. Glaws, D. Hettinger, R.N. King, Adversarial super-resolution of climatological wind and solar data, Proc. Natl. Acad. Sci. USA 117 (29) (2020) 16805–16815.
[40] F. Takens, Detecting strange attractors in turbulence, in: Lecture Notes in Mathematics, vol. 898, 1981, pp. 366–381.
[41] S. Tripathi, V.V. Srinivas, R.S. Nanjundiah, Downscaling of precipitation for climate change scenarios: a support vector machine approach, J. Hydrol. (2006).
[42] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, WaveNet: A Generative Model for Raw Audio, 2016.
[43] T. Vandal, E. Kodra, A.R. Ganguly, Intercomparison of machine learning methods for statistical downscaling: the case of daily and extreme precipitation, Theor. Appl. Climatol. (2019).
[44] T. Vandal, E. Kodra, S. Ganguly, A. Michaelis, R. Nemani, A.R. Ganguly, DeepSD: generating high resolution climate change projections through single image super-resolution, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017.
[45] M. Visbeck, J. Marshall, T. Haine, M. Spall, Specification of eddy transfer coefficients in coarse-resolution ocean circulation models, J. Phys. Oceanogr. 27 (3) (1997) 381–402.
[46] F. Waleffe, The nature of triad interactions in homogeneous turbulence, Phys. Fluids A 4 (1992) 350–363.
[47] R.L. Wilby, T.M. Wigley, D. Conway, P.D. Jones, B.C. Hewitson, J. Main, D.S. Wilks, Statistical downscaling of general circulation model output: a comparison of methods, Water Resour. Res. (1998).
[48] J. Yu, J.S. Hesthaven, Flowfield reconstruction method using artificial neural network, AIAA J. 57 (2) (Nov. 2019) 482–498.