# Koopman Ensembles for Probabilistic Time Series Forecasting

Anthony Frion, Lucas Drumetz, Guillaume Tochon, Mauro Dalla Mura, Abdeldjalil Aissa El Bey

# Koopman Ensembles for Probabilistic Time Series Forecasting

Anthony Frion
*Lab-STICC, IMT Atlantique*
Brest, France
anthony.frion@imt-atlantique.fr

Lucas Drumetz
*Lab-STICC, IMT Atlantique*
Brest, France
lucas.drumetz@imt-atlantique.fr

Guillaume Tochon
*LRE EPITA*
Le Kremlin-Bicêtre, France
guillaume.tochon@lrde.epita.fr

Mauro Dalla Mura
*Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab*
*Institut Universitaire de France*
Grenoble, France
mauro.dalla-mura@gipsa-lab.grenoble-inp.fr

Abdeldjalil Aissa El Bey
*Lab-STICC*
*IMT Atlantique*
Brest, France
abdeldjalil.aissaelbey@imt-atlantique.fr

***Abstract*—In the context of an increasing popularity of data-driven models to represent dynamical systems, many machine learning-based implementations of the Koopman operator have recently been proposed. However, the vast majority of those works are limited to deterministic predictions, while the knowledge of uncertainty is critical in fields like meteorology and climatology. In this work, we investigate the training of ensembles of models to produce stochastic outputs. We show through experiments on real remote sensing image time series that ensembles of independently trained models are highly overconfident and that using a training criterion that explicitly encourages the members to produce predictions with high inter-model variances greatly improves the uncertainty quantification of the ensembles.**

***Index Terms*—Dynamical systems, Koopman operator, Uncertainty quantification, Remote sensing, Sentinel-2**

## I. Introduction

With the simultaneously growing availability of observed geophysical data and advancement in machine learning methods, recent data-driven models have shown impressive performance in accurately forecasting physical dynamical systems [1]. Despite their performance, these models are harder to interpret than traditional methods, which means that it is more difficult to trust their predictions. One way to partially circumvent this flaw is to design models that can produce probability distributions of predictions instead of outputting a single prediction. Such models are then able to quantify the uncertainty, as the variance of a model's prediction can be seen as a measure of confidence. In this regard, one must be aware of the different sources of uncertainty that are to be identified. In the machine learning community, the uncertainty is usually decomposed into two categories: aleatoric uncertainty is the uncertainty that comes from the training data, e.g. due to noise and/or scarce sampling, while epistemic uncertainty denotes everything that comes from the model, e.g. the incapacity to fit the training data due to a lack of expressivity. We refer to [2] for an extensive discussion on the sources of uncertainty.

This work was supported by Agence Nationale de la Recherche under grant ANR-21-CE48-0005 LEMONADE.

Our work is based on the Koopman operator theory [3], which states that any nonlinear dynamical system can be described by a linear operator acting on the set of its measurement functions. While the Koopman operator is infinite dimensional in practice, many methods have been proposed to find approximate finite-dimensional representations, with applications to e.g. fluid dynamics and epidemiology [4].

We are most interested here in finite-dimensional representations based on neural auto-encoders [5]–[7]. These methods aim to learn a mapping from the input space of a dynamical system to a finite set of observation functions which are stable under the action of the Koopman operator, and vice versa. Using the machine learning vocabulary, the obtained Koopman invariant subspace is defined by the latent space of a learnt auto-encoder. Another learnt component (generally a matrix $\mathbf{K}$) then governs the evolution of the latent state through time. However, unlike for DMD-based methods, most of the existing works on Koopman autoencoders consider only deterministic models, which are unable to provide uncertainty estimates. Here, we are interested in finding simple ways to adapt these models to stochastic contexts. While there are many such ways in deep learning [8]–[10], we focus on deep ensembles, which are computationally intensive at training and inference but require no architectural change to a deterministic model and still outperform bayesian methods in some cases [11]. We show that in our case the usual way of training the members of an ensemble independently from each other leads to a highly overconfident ensemble. In order to alleviate this issue, we propose to train the members jointly with a loss function that encourages them to produce more diversified predictions.

The remainder of this paper is organised as follows: in section II, we review previous contributions on Koopman autoencoder models and on training neural networks for uncertainty quantification, especially with ensembles of models. We then introduce new methods for training ensembles of Koopman autoencoders in section III, and use them for forecasting time series of multispectral satellite images in section IV.

## II. Related works

### A. Data-driven implementations of the Koopman operator

Although the Koopman operator theory [3] dates back to the 1930s, it has known renewed interest when data-driven models related to this theory were introduced in the past two decades, notably DMD [12] and its extensions, on which the interested reader can find an extensive review in [4]. We will focus more specifically on recent methods [5]–[7] which consist in using a neural autoencoder to learn a finite set of observation functions on which the general evolution of a dynamical system can be described linearly. Most of the approaches in this line jointly learn an encoder $\phi : \mathbb{R}^n \to \mathbb{R}^d$, a decoder $\psi : \mathbb{R}^d \to \mathbb{R}^n$ and a matrix $\mathbf{K} \in \mathbb{R}^{d\times d}$ such that the advancement of an initial state $\mathbf{x}_0 \in \mathbb{R}^n$ by a time $\tau$ through the modelled dynamical system can be (approximately) written as

$$\mathbf{x}_\tau = \psi(\mathbf{K}^\tau \phi(\mathbf{x}_0)) \tag{1}$$

While this equation is often understood as a discrete model, where $\tau$ may only be a positive integer, a continuous formulation has been discussed in [6] and later implemented in [13].

### B. Uncertainty quantification for neural networks

Uncertainty quantification for neural networks is a field that has recently gained a lot of interest. We refer the interested reader to [14] for an in-depth survey of uncertainty quantification for neural networks, and to [2] for a very recent review that focuses on environmental-science applications. One of the most popular approaches to evaluate the quality of uncertainty estimates is the continuous ranked probability score (CRPS). First introduced in [15], the CRPS is defined as

$$\text{CRPS}(F, y_{true}) = \int_{-\infty}^{\infty} [F(y) - \mathbb{1}_{y\geq y_{true}}]^2 dy, \tag{2}$$

where $F$ is the cumulated distribution function of the output distribution, $y_{true}$ is the groundtruth and $\mathbb{1}_{y\geq y_{true}}$ is the Heaviside function, taking value 1 for $y \geq y_{true}$ and 0 otherwise. When the output distribution is a set of $M$ ensemble members $(y_j)_{j=1}^M$, the CRPS can be reframed [16] as

$$\text{CRPS} = \frac{1}{M}\sum_{j=1}^{M} |y_{true} - y_i| - \frac{1}{2}\frac{1}{M^2}\sum_{j=1}^{M}\sum_{k=1}^{M} |y_j - y_k|. \tag{3}$$

The first term is the mean absolute error (MAE) of the prediction and the second term is the halved mean absolute pairwise difference between ensemble members. Thus, the CRPS of a determinstic prediction is simply its MAE. The CRPS is being increasingly used as a loss function, e.g. in [17] with a parametric model for which the output is the mean and variance of a gaussian distribution. When the groundtruth is multivariate, the above expressions can simply be summed over all variables (hence ignoring correlations).

Let us now discuss more specifically the works on ensembles of neural networks. These ensembles generally consist in several instances of a same model, which can differ by various factors such as their initial parameterization or the set of data that they have been trained on. While the motivation of training ensembles of models in machine learning was at first to boost the performance by averaging the predictions of the ensemble members [18], it has been later noticed [11] that the variance of the predictions of an ensemble's members can also be used as a natural estimation of the uncertainty of the ensemble. However, in contrast to the methods that we propose in section III, [11] identified the independence of the trained members of the ensemble as a key element in training a deep ensemble, and the vast majority of subsequent works followed this principle, with the notable exception of [19].

## III. Proposed methods

### A. Introduction of a variance-promoting loss term

We train several instances of the model from [13], which is a classical Koopman autoencoder with 3 components, $\phi$, $\psi$ and $\mathbf{K}$ as described in subsection II-A.

In the single-instance version of this model, we denote by $\theta$ the set of all trainable parameters (including the coefficients of $\mathbf{K}$ and the trainable parameters of $(\phi, \psi)$). Suppose that we are working with a $n$-dimensional dynamical system and that our training dataset $(\mathbf{x}_{i,t})_{1\leq i\leq N, 0\leq t\leq T}$ is composed of $N$ time series of length $T+1$ resulting from the dynamical system of interest. Note that $\mathbf{x}_{i,t}$ is a $n$-dimensional vector. In what follows, we may drop the index $i$ to designate any of these time series. The loss function is composed of the terms:

$$L_{pred}(\theta) = \sum_{1\leq i\leq N}\sum_{1\leq\tau\leq T} ||\mathbf{x}_{i,\tau} - \psi(\mathbf{K}^\tau\phi(\mathbf{x}_{i,0}))||^2 \tag{4}$$

$$L_{ae}(\theta) = \sum_{1\leq i\leq N}\sum_{0\leq t\leq T} ||\mathbf{x}_{i,t} - \psi(\phi(\mathbf{x}_{i,t}))||^2 \tag{5}$$

$$L_{lin}(\theta) = \sum_{1\leq i\leq N}\sum_{1\leq\tau\leq T} ||\phi(\mathbf{x}_{i,\tau}) - \mathbf{K}^\tau\phi(\mathbf{x}_{i,0})||^2 \tag{6}$$

$$L_{orth}(\theta) = L_{orth}(\mathbf{K}) = ||\mathbf{K}\mathbf{K}^T - \mathbf{I}||_F^2 \tag{7}$$

where $||.||_F$ denotes the Frobenius norm. The terms $L_{pred}$, $L_{ae}$, $L_{lin}$ and $L_{orth}$ are respectively the prediction loss, autoencoding loss, linearity loss and orthogonality loss. We refer to [13] for further interpretation. They can all be weighted equally except for the orthogonality loss for which a suitable weight $\alpha$ has to be found, resulting in the global loss function

$$L(\theta) = L_{pred}(\theta) + L_{ae}(\theta) + L_{lin}(\theta) + \alpha L_{orth}(\theta). \tag{8}$$

Let us now suppose that we are training an ensemble of $M$ instances of this model. We then denote the parameters of these instances as $\Theta = (\theta_1, ..., \theta_M)$. The instances can be trained in parallel by defining a global loss function which is simply the sum of the loss functions for each of the $M$ instances:

$$\mathcal{L}_{independent}(\Theta) = \frac{1}{M}\sum_j L(\theta_j). \tag{9}$$

In this equation and in the following ones, all sums are defined on index $j$ from 1 to $M$. Using this loss function is equivalent to training the $M$ members of the ensemble sequentially and independently. As we will show experimentally, this may lead to a low diversity of the members, since the instances tend to all capture similar features in the data, which is undesirable in

ensemble learning. Given an input state $\mathbf{x}_0 \in \mathbb{R}^n$, the outputs of the members, obtained by equation (1), are denoted as $\hat{\mathbf{x}}_{t,j}$ with $0 \leq t \leq T$ denoting time and $1 \leq j \leq M$ denoting the member. The mean prediction of the members is defined as

$$\hat{\mathbf{x}}_t = \frac{1}{M} \sum_j \hat{\mathbf{x}}_{t,j}. \tag{10}$$

For the uncertainty quantification to be accurate, we would like the empirical variance of these predictions to be close to the squared error between the mean prediction and the groundtruth $\mathbf{x}_t$: see [2] for reference. Thus, we seek

$$\frac{1}{M-1} \sum_j ||\hat{\mathbf{x}}_{t,j} - \hat{\mathbf{x}}_t||^2 \approx ||\hat{\mathbf{x}}_t - \mathbf{x}_t||^2. \tag{11}$$

In practice, since the empirical variance is too low when training an ensemble with equation (9), we introduce a variance-promoting loss term, which takes into account all members:

$$L_{var}(\Theta) = -\frac{1}{M} \sum_j ||\hat{\mathbf{x}}_{t,j} - \hat{\mathbf{x}}_t||^2. \tag{12}$$

We write this term with simplified notations, yet the full loss term includes sums over $i$ and $t$ like in equations (4) to (6). Using the new loss term from equation (12), one can now introduce a new loss function for training ensembles:

$$\mathcal{L}_{var,\lambda}(\Theta) = \mathcal{L}_{independent}(\Theta) + \lambda L_{var}(\Theta), \tag{13}$$

where the case $\lambda = 0$ corresponds to equation (9). This loss promotes the sum of variances over each of the $N$ variables of the state individually, i.e. the total variance of the predictions.

### B. Analysis of the choice of $\lambda$

It is a standard and easy to prove result in statistics that

$$\frac{1}{M} \sum_j ||\hat{\mathbf{x}}_{t,j} - \mathbf{x}_t||^2 = \frac{1}{M} \sum_j ||\hat{\mathbf{x}}_{t,j} - \hat{\mathbf{x}}_t||^2 + ||\hat{\mathbf{x}}_t - \mathbf{x}_t||^2 \tag{14}$$

for any value $\mathbf{x}_t$ and predictions $\hat{\mathbf{x}}_{t,j}$. From this, we obtain

$$\frac{1}{M} \sum_j ||\hat{\mathbf{x}}_{t,j} - \mathbf{x}_t||^2 - \frac{\lambda}{M} \sum_j ||\hat{\mathbf{x}}_{t,j} - \hat{\mathbf{x}}_t||^2 = \frac{1-\lambda}{M} \sum_j ||\hat{\mathbf{x}}_{t,j} - \hat{\mathbf{x}}_t||^2 + ||\hat{\mathbf{x}}_t - \mathbf{x}_t||^2 \tag{15}$$

for a given $\lambda$. If $\lambda \leq 1$, then this expression is trivially positive. However, if $\lambda > 1$, then it is not negatively bounded. Indeed, since $\mathbf{x}_t$ is a constant vector, one can simply choose arbitrarily large member predictions $\hat{\mathbf{x}}_{t,j}$ satisfying $\hat{\mathbf{x}}_t = 0$ through equation (10) (e.g. the members can be arranged in opposite pairs) in order for the expression (15) to become arbitrarily low. As this analysis remains true for any $\mathbf{x}_0$ and $t$, the prediction loss term (4) can counterbalance the variance loss term (12) as long as $0 \leq \lambda \leq 1$ in equation (15). If $\lambda > 1$ then the training procedure will diverge with an inter-model variance growing to infinity. Therefore, the hyperparameter $\lambda$ for training the ensemble should be chosen between 0 and 1, and the variance of the predictions grows with the value of $\lambda$, as we will show in our experiments. This simple interpretation of $\lambda$ motivates the use of a biased estimator of the variance in equation (12): with an unbiased estimator, the acceptable range for $\lambda$ would depend on the number $M$ of members.

To the best of our knowledge, the introduction of this loss term for training an ensemble of neural networks is a novelty. The closest contribution that we identified was in [19], which introduced loss terms similar to (12) with a notable difference being that the authors work with models that have bounded outputs, so that the parameter $\lambda$ can be set arbitrarily without fear of obtaining variances that diverge to infinity. The choice of unbounded models trades more flexibility in the individual members with the constraints on $\lambda$ that we described above.

### C. Using a loss term inspired by the CRPS

An alternative to the loss function (13) is to use the CRPS for training. While easy to compute for small models, the formulation of equation (3) gets very costly to compute as the number $M$ of members gets higher because of the pairwise differences in the second term. Therefore, we propose to replace this term by the mean absolute error between the individual predictions and the mean prediction. Using the same notations as in equation (12), we introduce

$$L_{abs}(\Theta) = -\frac{1}{2} \frac{1}{M} \sum_j |\hat{\mathbf{x}}_{t,j} - \hat{\mathbf{x}}_t|. \tag{16}$$

One can easily prove that

$$\begin{aligned} \frac{1}{M} \sum_j |\hat{\mathbf{x}}_{t,j} - \hat{\mathbf{x}}_t| &\leq \frac{1}{M^2} \sum_{j=1}^{M} \sum_{k=1}^{M} |\hat{\mathbf{x}}_{t,j} - \hat{\mathbf{x}}_{t,k}| \\ &\leq \frac{2}{M} \sum_j |\hat{\mathbf{x}}_{t,j} - \hat{\mathbf{x}}_t|, \end{aligned} \tag{17}$$

so that (16) can be seen as a proxy to the second term of the CRPS formulation in (3), while the first term is analogous to the sum of the prediction losses (4) using the $\mathcal{L}_1$ distance instead of the squared $\mathcal{L}_2$ distance. This motivates the introduction of a new diversity-promoting loss function:

$$\mathcal{L}_{CRPS}(\Theta) = \sum_j L_1(\theta_j) + \lambda L_{abs}(\Theta), \tag{18}$$

where $\lambda$ is usually set to 1, $L_1(\theta)$ is a variant of equation (8) where all squared $\mathcal{L}_2$ distances are replaced by $\mathcal{L}_1$ distances for the purpose of consistency/homogeneity between all loss terms. Note that this loss is not the true CRPS but an approximation of it, with auxiliary terms.

## IV. Experiments

In this section, we present our experiments on datasets originally introduced in [20], and consisting of time series of Sentinel-2 multispectral satellite images over two spatial areas: the forest of Fontainebleau and the forest of Orléans in France. The datasets and codes are available at https://github.com/anthony-frion/Sentinel2TS. The time series for the two areas are available in two versions. The original versions contain raw images with no pre-processing other than the classical (level-2A) atmospheric correction, yet the time series are incomplete

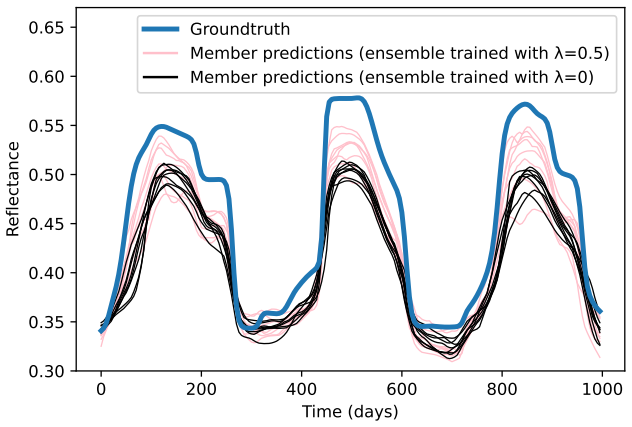


Fig. 1. Forecasting from time 0 by two ensembles for the reflectance of the B7 band (in near infrared) for a Fontainebleau pixel. Here, both ensembles are biased, but the ensemble trained with a variance-promoting loss term ($\lambda = 0.5$) yields a higher inter-member variance, and hence a better uncertainty estimate, than the ensemble of independently trained models ($\lambda = 0$).

as only the images that are not corrupted by the presence of clouds are retained. Since training from irregularly-sampled time series is very challenging, the second versions interpolate these available images in time through Cressman interpolation, resulting in partly synthetic but regularly-sampled time series. Although Koopman autoencoders are able to handle irregularly-sampled time series [13], we choose to train on the regular version of the Fontainebleau data in order to keep the training procedure simple. We test the trained ensembles on two tasks: 1) extrapolating on the training Fontainebleau area to times unseen during training 2) predicting from an initial time on the test Orléans area. Thus, task 1) is used to test temporal extrapolation while task 2) tests the ability to transfer the knowledge to a new area with a distribution shift.

We first motivated the introduction of our customized ensemble training loss (15) by making the simple observation that independently trained members from an ensemble of Koopman autoencoders tend to learn very similar dynamics. We show on figure 1 a typical example for an ensemble trained with loss (9) (i.e. $\lambda = 0$ in loss (15)), where all 8 instances make similar predictions from an initial observation belonging to the training area. Here, the member predictions are all much closer to each other than any of them is to the groundtruth. Although this is an illustrative exemple with a relatively high forecasting error, it is symptomatic of a very overconfident ensemble. We also show the predictions of an ensemble trained in the same conditions but with $\lambda = 0.5$ in loss (15): although this ensemble is biased too, its higher variance makes it less overconfident, and thus better in this case.

In order to promote the diversity of the members, we now train ensembles with the loss function (15) with different values of $\lambda$, remembering that $\lambda = 0$ corresponds to training the models independently from one another, as classically done in the literature. As expected, the variance of the predictions increases with $\lambda$. We also train an ensemble with the loss function (18). We evaluate the quality of the uncertainty

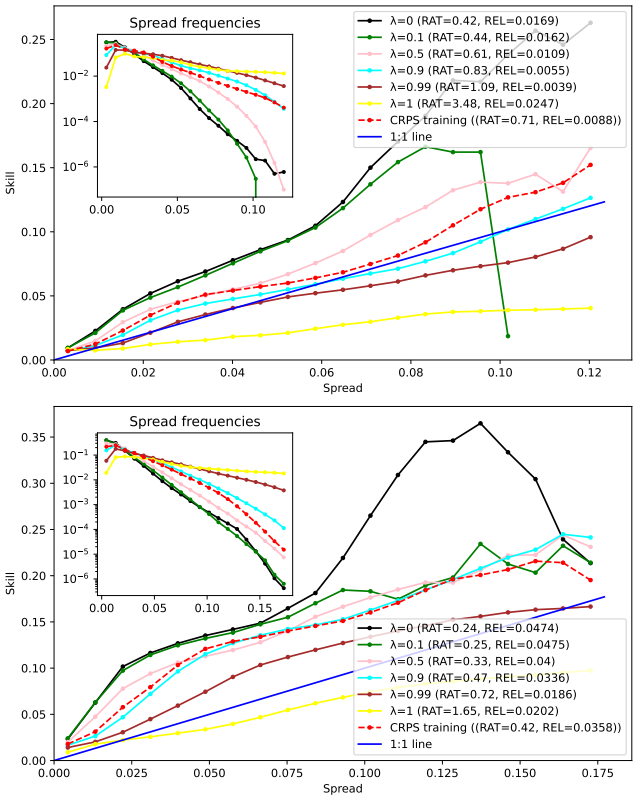


Fig. 2. Spread-skill plots for two different datasets. Top: spread-skill plot of extrapolation on the training Fontainebleau area. Bottom: spread-skill plot of predictions from time 0 on the test Orléans area.

estimates with 2 methods: the CRPS and the spread-skill plot.

We show in figure 2 the spread-skill plots for the two identified tasks. The idea of these plots is to represent the skill as a function of the spread. The skill is defined as the root mean squared error between the average prediction of the ensemble (obtained by equation (10)) and the groundtruth. The spread is defined as the standard deviation of the predictions of the members. The spread and skill are the square roots of the left and right members of equation (11): one would like them to be approximately equal. In practice, we consider a set of ensemble predictions and groundtruth, where all spectral bands and prediction time spans are separated, resulting in univariate values. We compute a 20-bin histogram of this set according to the spread. Then, for each bin, we compute the mean skill of the predictions and the number of points inside the bin. On the main plot, each point corresponds to one of the bins, and we would like the points to stay close to the 1:1 line, which corresponds to equation (11). If the plot is above the 1:1 line, it means that the ensemble tends to underestimate its errors, hence it is overconfident. On the contrary, a plot below the 1:1 line means that the ensemble is underconfident. The inner plots show the frequencies associated to each bin.

A spread-skill plot can be summarized by two metrics: the SSREL is the sum of the absolute distances to the 1:1 line over the bins, weighted by the bin frequencies. It is positive and its ideal value is zero. The SSRAT measures the spread-skill ratio

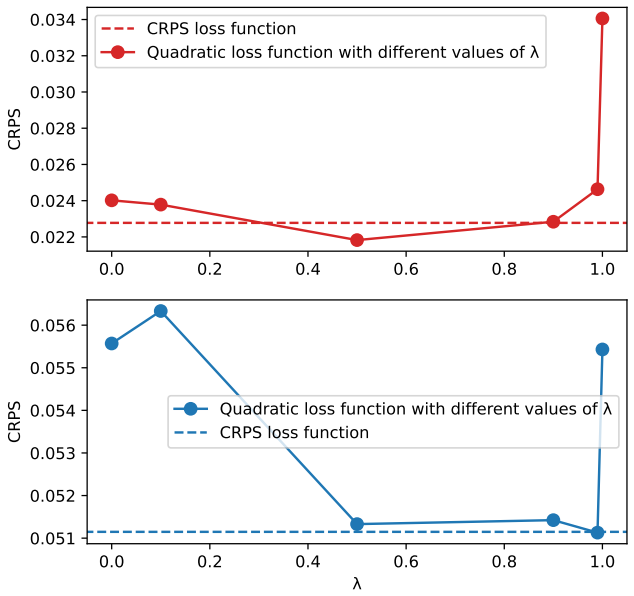


Fig. 3. CRPS of ensembles of Koopman autoencoders according to the weight $\lambda$ of their variance-promoting loss term during training. Top: extrapolation on training Fontainebleau area. Bottom: transfer to test Orléans area. The represented values of $\lambda$ are $0, 0.1, 0.5, 0.9, 0.99, 1$.

globally, and is unaffected by the binning process. It is positive and unitless, and a value smaller or greater than 1 respectively characterize an overconfident or an underconfident model. We refer to [2] for more extensive discussions on these notions.

Several conclusions can be drawn from figure 2. First, the ensemble with independently trained models (corresponding to $\lambda = 0$) is highly overconfident, which quantitatively confirms the intuition gained from figure 1. Then, one can clearly see that the ensembles get less confident as the value of $\lambda$ increases. The case of $\lambda = 1$ is a limit case, and results in the only model that is severely underconfident on both training and test areas. The value $\lambda = 0.99$ yields the best spread-skill ratios, and the model trained with a proxy to the CRPS lies in between $\lambda = 0.5$ and $\lambda = 0.9$.

Finally, we show in figure 3 the CRPS of the ensembles as a function of the value of $\lambda$ used in their training function (13). Note that lower values are better for the CRPS metric. Again, one can see that a well-chosen value of $\lambda$ can significantly improve the performance compared to an ensemble of independently trained members ($\lambda = 0$). The values $\lambda = 0.5$ and $\lambda = 0.9$ seem to be good compromises between the CRPS on the two tasks, while the model trained with a loss function similar to the CRPS also performs well on both.

## V. Conclusion

In this work, after noticing that ensembles of Koopman autoencoders tend to be very overconfident when their members are trained independently, we introduced a variance-promoting loss term which encourages the members of an ensemble to produce more diverse forecasts. We studied, both analytically and empirically, the influence of this term on the trained ensembles according to its weight relatively to the other loss terms. We found that, according to several metrics, the quality of the uncertainties produced by the ensembles improves as the weight of the variance-promoting loss term gets closer to its theoretical limit of 1. In future works, we will try using this loss term in conjunction with other uncertainty quantification methods, e.g. Monte Carlo dropout and ensemble predictions with a single model. We also intend to further study the specificities of uncertainty quantification for long-term forecasting tasks, and for Koopman autoencoders in particular.

## References


[1] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds, Z. Eaton-Rosen, W. Hu *et al.*, "Graphcast: Learning skillful medium-range global weather forecasting," *arXiv preprint arXiv:2212.12794*, 2022.

[2] K. Haynes, R. Lagerquist, M. McGraw, K. Musgrave, and I. Ebert-Uphoff, "Creating and evaluating uncertainty estimates with neural networks for environmental-science applications," *Artificial Intelligence for the Earth Systems*, vol. 2, no. 2, p. 220061, 2023.

[3] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.

[4] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, "Modern Koopman theory for dynamical systems," *arXiv preprint arXiv:2102.12086*, 2021.

[5] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature communications*, vol. 9, no. 1, p. 4950, 2018.

[6] S. E. Otto and C. W. Rowley, "Linearly recurrent autoencoder networks for learning dynamics," *SIAM Journal on Applied Dynamical Systems*, vol. 18, no. 1, pp. 558–593, 2019.

[7] A. Frion, L. Drumetz, M. Dalla Mura, G. Tochon, and A. Aïssa-El-Bey, "Leveraging neural Koopman operators to learn continuous representations of dynamical systems from scarce data," in *ICASSP*. IEEE, 2023, pp. 1–5.

[8] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *ICML*. PMLR, 2016, pp. 1050–1059.

[9] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," *NeurIPS*, vol. 31, 2018.

[10] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *ICML*. PMLR, 2015, pp. 1613–1622.

[11] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *NeurIPS*, vol. 30, 2017.

[12] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of fluid mechanics*, vol. 656, pp. 5–28, 2010.

[13] A. Frion, L. Drumetz, M. Dalla Mura, G. Tochon, and A. A. E. Bey, "Neural Koopman prior for data assimilation," *arXiv preprint arXiv:2309.05317*, 2023.

[14] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher *et al.*, "A survey of uncertainty in deep neural networks," *arXiv preprint arXiv:2107.03342*, 2021.

[15] J. E. Matheson and R. L. Winkler, "Scoring rules for continuous probability distributions," *Management science*, vol. 22, no. 10, pp. 1087–1096, 1976.

[16] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *Journal of the American statistical Association*, vol. 102, no. 477, pp. 359–378, 2007.

[17] S. Baran and A. Baran, "Calibration of wind speed ensemble forecasts for power generation," *Weather*, vol. 125, pp. 609–624, 2021.

[18] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[19] S. Jain, G. Liu, J. Mueller, and D. Gifford, "Maximizing overall diversity for improved uncertainty estimates in deep ensembles," in *Proceedings of the AAAI conference*, vol. 34, no. 04, 2020, pp. 4264–4271.

[20] A. Frion, L. Drumetz, G. Tochon, M. Dalla Mura, and A. A. El Bey, "Learning sentinel-2 reflectance dynamics for data-driven assimilation and forecasting," in *EUSIPCO*. IEEE, 2023, pp. 1390–1394.