Exploring Bayesian deep learning for weather forecasting with the Lorenz 84 system

Liu Yang[1], Attema Jisk Jakob[2], and Hazeleger Wilco[3]

[1]Netherlands eScience Center / Wageningen University
[2]Netherlands eScience Center
[3]Utrecht University

November 16, 2022

Abstract

The need for uncertainty quantification placed by weather forecasting makes Bayesian deep learning (BDL) a suited candidate for data-driven weather forecasting. In this study, we use Bayesian Long-Short Term Memory neural networks (BayesLSTMs) to forecast output from the Lorenz 84 system with seasonal forcing. The latter represents the dynamics of large scale eddies (Rossby waves) on a westerly jet. We show that forecasts with the BayesLSTM can stay close to the attractor of the Lorenz model and conclude that they represent the nonlinear relations between each component in this simplified atmospheric circulation system. The forecasts are evaluated against persistence and a Vector Autoregressive Model (VAR). We demonstrate that the BayesLSTMs can produce reliable probabilistic forecasts and address uncertainties relevant to weather forecasting. Our study indicates that BDL is an easy and fast solution for probabilistic weather forecast and is promising to enhance weather forecasting capabilities at short to medium-range timescales.

Exploring Bayesian deep learning for weather forecasting with the Lorenz 84 system

Yang Liu[1,2], Jisk Attema[1], Wilco Hazeleger[3]

[1]Netherlands eScience Center, Amsterdam, the Netherlands
[2]Meteorology and Air Quality Group, Wageningen University, Wageningen, the Netherlands
[3]Faculty of Geosciences, Utrecht University, Utrecht, the Netherlands

Key Points:

- Bayesian deep neural networks are able to represent uncertainties relevant to weather forecasting.
- A trained Bayesian deep neural network can preserve the physical consistency of the Lorenz 84 system.
- The forecast quality of the trained Bayesian deep neural network deteriorates with forecast lead time and it is state-dependent.

Corresponding author: Yang Liu, `y.liu@esciencecenter.nl`

Abstract

[The need for uncertainty quantification placed by weather forecasting makes Bayesian deep learning (BDL) a suited candidate for data-driven weather forecasting. In this study, we use Bayesian Long-Short Term Memory neural networks (BayesLSTMs) to forecast output from the Lorenz 84 system with seasonal forcing. The latter represents the dynamics of large scale eddies (Rossby waves) on a westerly jet. We show that forecasts with the BayesLSTM can stay close to the attractor of the Lorenz model and conclude that they represent the nonlinear relations between each component in this simplified atmospheric circulation system. The forecasts are evaluated against persistence and a Vector Autoregressive Model (VAR). We demonstrate that the BayesLSTMs can produce reliable probabilistic forecasts and address uncertainties relevant to weather forecasting. Our study indicates that BDL is an easy and fast solution for probabilistic weather forecast and is promising to enhance weather forecasting capabilities at short to medium-range timescales.]

Plain Language Summary

[Recent developments in artificial intelligence (AI) have brought many techniques to climate science. Among these techniques, deep neural networks (DNN) serve as good candidates to improve and speed up weather forecasts. However, these DNN always have fixed structure and therefore can not satisfy the need of weather forecast for uncertainty estimation. To solve the problem, we introduce Bayesian deep learning (BDL), which is probabilistic and enables uncertainty quantification. In this study, we explore the BDL with a simplified chaotic system, the Lorenz 84 model with seasonal forcing. We test and use BDL to forecast the Lorenz 84 system and evaluate its probabilistic forecast skill against the persistence and a baseline statistical model. Our study indicates that the BDL is able to account for the uncertainty required by weather forecasting and it represents the nonlinear relations between each component in this simplified atmospheric circulation system. It is a promising tool for preliminary and quick probabilistic forecasts and therefore can enhance weather forecasting capabilities.]

1 Introduction

Deep neural networks (DNNs) are capable of representing intricate features of data and have been proven to be useful for many scientific disciplines (e.g., LeCun et al., 2015), including weather forecasting and climate science (Reichstein et al., 2019). It has been demonstrated by recent studies that typical DNN are able to mimic and predict the behavior of chaotic systems (e.g. Hochreiter & Schmidhuber, 1997; Chattopadhyay et al., 2019) and therefore they are potentially applicable to weather forecasting. However, mostly deterministic DNNs are considered and these are prone to overfitting and this can result in over-confident forecasts (Shridhar et al., 2019).

Due to the chaotic nature of the atmospheric dynamics and uncertainties in both initial conditions and models representing the atmosphere, weather forecasts are of probabilistic nature. In general, uncertainty estimation is achieved via an ensemble approach within trustworthy Numerical Weather Forecast systems (NWP) (Gneiting et al., 2007; Leutbecher & Palmer, 2008). However, this strategy is computationally expensive for NWP-based weather forecasts. Concerning the deep learning approaches, in order to meet the requirement for uncertainty quantification, many attempts have been made to adapt deterministic DNN to weather forecasting (e.g., Scher & Messori, 2018). These efforts mainly involve generating a DNN-based ensemble through perturbing either the training data or the structure of DNN (e.g., Zaier et al., 2010; H.-z. Wang et al., 2017). However, in practice, this technique is computationally expensive due to multiple training cycles that are needed and it is often difficult to manually select proper perturbations which can approximate the error growth of a real dynamical system. Fortunately, recent developments in deep learning have led to a branch of DNN to cope with overfitting and address uncertainties, which is known as Bayesian deep learning (BDL).

Unlike feed-forward DNN, BDL is constructed by replacing fixed weights with distributions and therefore are designed to represent uncertainties (Blundell et al., 2015). With a well-defined likelihood function, BDL is able to capture both the aleatoric and epistemic uncertainty (Kendall & Gal, 2017; Shridhar et al., 2018, 2019). They can avoid making over-confident decisions and incorporate regularization naturally by implementing the variational approaches (Shridhar et al., 2019). Together with the simplicity of implementing BDL on an already defined deep neural network, these make BDL an attractive approach for representing atmospheric dynamics and the practice of weather forecasting (Vandal et al., 2018).

An operational numerical weather forecast system is very complex. Here, we want to understand the characteristics of BDL within a simplified dynamical system that represents the essence of midlatitude atmospheric dynamics and explore the types of uncertainties addressed by BDL. In particular we examine how BDL can replicate the phase and amplitude of midlatitude Rossby waves on a jet as represented in a Lorenz 84 model (Lorenz, 1984; H. Wang et al., 2014). The predictive nature and time scale of propagation and development of Rossby waves form the basis of short to medium-range weather forecasting. We will assess whether BDL can represent the predictability of this simplified atmospheric circulation system. We notice that the concept of BDL in the perspective of weather forecasting is quite similar to the implementation of the Bayesian theorem in data assimilation (e.g., Ghil & Malanotte-Rizzoli, 1991; Navon, 2009; Bannister, 2017).

Long-Short Term Memory neural networks (LSTMs) have a network structure and characteristics that are found to be suitable to represent fluids in environmental studies (Liu et al., 2020). In this study, we explore BDL by turning LSTMs into Bayesian LSTMs (BayesLSTMs). We will use the BayesLSTMs to forecast the Lorenz 84 model and assess the forecast quality in the spatial and temporal space at different lead times. The probabilistic forecasts produced by the BayesLSTM will be evaluated against those with persistence of initial conditions and a baseline statistical model. An emphasis is placed on the uncertainties represented by the BayesLSTM and its capacity in preserving the physical consistency in a simplified atmospheric circulation system.

The paper is organized as follows: we elaborate on the concept of BDL and Lorenz 84 model with seasonal forcing in the section Methodology. An analysis of uncertainty estimation with BDL, and the procedure of sampling the BayesLSTM and generating ensemble forecasts are also provided in this section. The probabilistic forecasts of the Lorenz 84 system using the BayesLSTM are elucidated and analyzed in the section Results. This section also includes forecasts with persistence and a baseline statistical model for comparison and evaluation. Finally, in the section Conclusion and Discussion, we summarize this study and provide our perspective for future work.

2 Methodology

In this section, we briefly introduce the Lorenz 84 model with seasonal forcing and elaborate upon the concept of BDL as well as how an LSTM network is transformed into a BayesLSTM. Based on the characteristics of BDL, the procedure of producing ensemble forecasts and a description of uncertainty estimation with BayesLSTM is presented in this section.

2.1 Lorenz 84 Model with Seasonal Forcing

The Lorenz 84 system represents the general circulation of the atmosphere in a low dimensional space and therefore it is useful as a baseline model for exploring BayesLSTMs in weather forecasting (Lorenz, 1984). To incorporate more realistic features into the simple Rossby wave evolution system, we add a seasonal forcing to the classical Lorenz 84 model. The dynamical system is formulated as follows:

$$\begin{aligned}
\frac{dX}{dT} &= -Y^2 - Z^2 - aX + aF(1 + \epsilon cos(\omega T)) \\
\frac{dY}{dT} &= XY - bXZ - Y + G(1 + \epsilon sin(\omega T)) \\
\frac{dZ}{dT} &= bXY + XZ - Z
\end{aligned} \tag{1}$$

where X represents the intensity of the westerly wind circulating around the globe, Y and Z represent the cosine and sine phases of a chain of superimposed large-scale eddies, T is the time, a and b indicate mechanical and thermal damping, F and G the symmetric and asymmetric thermal forcing, ϵ the intensity of seasonal forcing, and ω the angular frequency of seasonality (Freire et al., 2008). In this study, we mainly focus on the sensitivity of the forecast quality to variations in the initial condition X and model parameter a.

To obtain a chaotic system that is suitable for the assessment of the BayesLSTM forecast, we chose the model parameters to be $a = 0.25$, $b = 4.0$, $F = 8.0$, $G = 1.0$, $\epsilon = 0.4$, and the initial conditions as $X, Y, Z = 1.0$. One unit of time in the Lorenz model corresponds to 5 days. The damping time of the wave is about 5 days (Lorenz, 1984). We sample the system with a temporal resolution equal to 1/30 unit time, which is 4 hours. The period of seasonal forcing is taken as 73 unit time steps and then the period of the entire system is equivalent to 356 days. With this configuration, the trajectories and the time series of each variable are shown in Figure 1. Unless specifically noted, the time step and lead time steps in this paper are based on the sampling interval, which is 4 hours.

2.2 BayesLSTM and Bayes by Backprop

Our aim is to investigate whether the BayesLSTM can represent the Lorenz 84 model described above. We can add Bayesian inference to an existing neural network by replacing fixed weights with distributions (e.g. see Figure 1 in Blundell et al., 2015). Given the structure of an LSTM network (Hochreiter & Schmidhuber, 1997), the Bayesian form of an LSTM network can be represented by equation 2:

$$\begin{aligned}
i_t &= \sigma(W^s_{xi} \circ x_t + W^s_{hi} \circ h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \\
f_t &= \sigma(W^s_{xf} \circ x_t + W^s_{hf} \circ h_{t-1} + W_{cf} \circ c_{t-1} + b_f) \\
c_t &= f_t \circ c_{t-1} + i_t \circ tanh(W^s_{xc} \circ x_t + W^s_{hc} \circ h_{t-1} + b_c) \\
o_t &= \sigma(W^s_{xo} \circ x_t + W^s_{ho} \circ h_{t-1} + W_{ct} \circ c_t + b_o) \\
h_t &= o_t \circ tanh(c_t)
\end{aligned} \tag{2}$$

with i_t the input gate, f_t the forget gate, c_t the cell state, o_t the output gate, h_t the hidden state, W^s the weight distribution, x_t the input, b the bias, $\circ$ the element-wise product, σ the sigmoid function and $tanh$ the hyperbolic tangent function. The subscripts describe the corresponding weight matrix to different gates and states. W^s_{xi} indicates the weight matrix of input values related to the input gate, while W^s_{hf} represents the weight matrix of hidden states corresponding to the forget gate. The subscript t indicates the time step. The structure of a BayesLSTM is illustrated in Figure 1c.

We need to search for the weight distribution W^s, thus the posterior $p(w|D)$ where w denotes the weight and $D = (x_j, y_j)_j$ indicates the training set. As the true posterior probability distribution is intractable (because of the marginal likelihood), we use a variational inference scheme, namely the Bayes by Backprop approach, to approximate it (Blundell et al., 2015; Shridhar et al., 2018, 2019). The reason for choosing this method is elaborated upon in detail in the supporting material. A simple variational distribution $q(w|\theta)$ (where θ is the variational posterior parameter), such as a Gaussian distribution, or a lognormal distribution is often chosen (Blundell et al., 2015; Shridhar et al., 2018; Vandal et al., 2018). Here we approximate the posterior $p(w|D)$ with a Gaussian distribution $q(w|\theta)$, which consists of two trainable parameters $\mu \in \mathbb{R}^d$ and $\sigma \in \mathbb{R}^d$. As a result, θ in the assumed variational distribution $q(w|\theta)$ can be denoted by $\mathcal{N}(\theta|\mu, \sigma^2)$.

The gap between the chosen variational distribution and the exact posterior distribution is reduced using the Kullback-Leibler (KL) divergence between $p(w|D)$ and $q(w|\theta)$ (Graves, 2011; Blundell et al., 2015). KL divergence measures the similarity between two distributions and in this we define the optimal parameters θ^* as:

$$\begin{aligned} \theta^* &= arg \min_\theta [q(w|\theta)||p(w|D)] \\ &= arg \min_\theta KL[q(w|\theta)||p(w)] - \mathbb{E}_{q(w|\theta)}[log\ p(D|w)] + log\ p(D) \end{aligned} \tag{3}$$

where KL indicates the full KL divergence operation and $\mathbb{E}$ represents the expectation. This equation includes a data dependent part $\mathbb{E}_{q(w|\theta)}[logp(D|w)]$ and a prior dependent part $KL[q(w|\theta)||p(w)]$ (Neal & Hinton, 1998; Blundell et al., 2015; Shridhar et al., 2019). We sample the weight w from $q(w|\theta)$ and the cost function that we optimize is:

$$\mathcal{F}(D, \theta) = \sum_{n=1}^{N} log\ q(w^{(n)}|\theta) - log\ p(w^{(n)}) - log\ p(D|w^{(n)}) \tag{4}$$

where $w^{(n)}$ denotes the nth Monte Carlo sampling from the variational posterior $q(w^{(n)}|\theta)$.

Together with the local reparameterization method (explained in the supplementary material), which translates the global uncertainty in the weights into a form of local uncertainty (Kingma et al., 2015; Shridhar et al., 2019), our BayesLSTMs are ready for training and back-propagation. We constructed the networks using the Pytorch library, and our code is published on Github (https://github.com/geek-yang/DLACs).

2.3 Ensemble Forecasting with BDL and Numerical Configurations

The ensemble method is generally used for uncertainty assessment in weather forecasting (Gneiting et al., 2005; Buizza et al., 2008; Leutbecher et al., 2017). In numerical weather prediction systems (NWP), uncertainties in the initial conditions and model parameters are projected by ensemble forecasts with perturbations in the initial conditions and model formulations (Palmer, 2002; Milinski et al., 2020). It has been explained in many previous studies that BDL is able to address the uncertainties in initial conditions and model parameters (e.g., Kendall & Gal, 2017). More details about the uncertainty estimation with BDL are provided in the supplementary material. This characteristic is fundamental for any probabilistic forecast and therefore makes BDL a candidate for weather forecasting. However, they are treated differently than in operational NWP approaches.

The ability of BayesLSTM to characterize uncertainty is reflected in its forecasting procedure. During a prediction process, the whole time series preceding the forecast date ($t < t_0$) will be fed to the model to initialize the memory and position the state of the network. Therefore the model itself is constrained by the past and this is similar to an NWP-based forecast. When producing a forecast that takes uncertainties into account for a next time step, the BayesLSTM will first sample the weight distributions multiple times to build an ensemble and then use the sampled weight matrix to generate the predictions for the target time step ($t = t_1$). The ensemble forecast can be extended to more time steps ahead ($t = t_n > t_1$) by continuing with each individual LSTM.

In order to evaluate ensemble forecasts with the BayesLSTM, several scores are calculated, including continuous ranked probability score (CRPS), root mean square error (RMSE) and Euclidean distance (EuD). The mathematical expressions of these scores can be found in the supplementary material.

For all the experiments in this paper, we generate sequences including 1500 time steps (250 days) with the Lorenz 84 model. The training set contains 1300 time steps (about 216 days) and the validation set consists of 200 time steps (about 33 days). The optimization is based on the minimization of training loss, which consists of likelihood cost (data-dependent) and complexity cost (prior dependent) (Shridhar et al., 2019). A scaling factor between these two sources of loss should be tuned, since it accounts for the trade-off between the width of ensemble spread in terms of uncertainty estimation and saturation of forecasts around the variance displayed in the observations. Note that the scaling factor is related to the normalization of the distributions and cannot be calculated exactly. The training time is about 20 hours on a single GPU (Nvidia Tesla K40m). The hyperparameters like the learning rate, number of epochs and number of layers, were tested and determined in terms of the EuD error. It shows that a combination of a learning rate equal to 0.01, a single BayesLSTM layer and 3000 epochs is sufficient to achieve satisfying results. The training loss is shown in Figure S1. More details about the numerical configurations are shown in the supplementary material.

2.4 Vector Autoregressive Model

The VAR model is used as a baseline method to assess the probabilistic forecast skill of BayesLSTM. As a variant of the autoregressive model (AR), the VAR model generalizes univariate AR by allowing for multivariate time series and therefore can capture the relation between multiple variables. The VAR model and many variants belonging to the VAR family haven shown skill in many weather forecast applications (e.g., Gneiting et al., 2006; L. Wang et al., 2016, and many others). To expand its forecast capacity from the deterministic domain to the probabilistic domain, we replaced the Gaussian noise term (ϵ_t) with Gaussian distributed variations based on the variance of input time series from the chosen lag step to the current step. The optimal number of the lag to be included in the model is determined based on the auto-correlation of each variable of the Lorenz 84 model output (shown in Figure S2 in the supplementary material), and tests of forecast quality in terms of the CRPS score. In our case, the VAR model with a lag equal to 3 provides the best probabilistic forecast. Mathematically, our modified VAR model can be expressed as:

$$
\begin{aligned}
X_t &= \alpha_1 + \sum_{l=1}^{Lag} (\beta_{11,l} X_{t-l} + \beta_{12,l} Y_{t-l} + \beta_{13,l} Z_{t-l}) + \epsilon_{1,t} \\
Y_t &= \alpha_2 + \sum_{l=1}^{Lag} (\beta_{21,l} X_{t-l} + \beta_{22,l} Y_{t-l} + \beta_{23,l} Z_{t-l}) + \epsilon_{2,t} \\
Z_t &= \alpha_3 + \sum_{l=1}^{Lag} (\beta_{31,l} X_{t-l} + \beta_{32,l} Y_{t-l} + \beta_{33,l} Z_{t-l}) + \epsilon_{3,t} \qquad (5) \\
&with \\
\epsilon_{1,t} &= \mathcal{N}[0, \sigma(X_{t-1}, X_{t-2}, ..., X_{t-l})^2] \\
\epsilon_{2,t} &= \mathcal{N}[0, \sigma(Y_{t-1}, Y_{t-2}, ..., Y_{t-l})^2] \\
\epsilon_{3,t} &= \mathcal{N}[0, \sigma(Z_{t-1}, Z_{t-2}, ..., Z_{t-l})^2]
\end{aligned}
$$

Where α and β are trainable parameters in the model, ϵ_t the Gaussian distributed variations, and X_{t-l}, Y_{t-l} and Z_{t-l} the Lorenz model output at time lag l. The parameters were updated by fitting the model to the time series of Lorenz model output using maximum likelihood.

3 Results

We evaluate the capacity of BDL in representing the dynamics of Rossby wave propagation on a westerly jet by investigating the forecasts in the spatial and temporal domains. Based on the selected scoring metrics, we further assess the forecast quality of BayesLSTM against the forecasts with persistence and a VAR model.

3.1 Representing the Evolution of Lorenz 84 Model

A retrospective forecast of the Lorenz 84 system with the BayesLSTM is shown in Figure 2. The forecasts start every time step (4 hours) and each has been extended to a lead time of 3 days. Given the time series of the BayesLSTM forecasts in Figure 2a, it can be observed that in general the forecasts are close to the time series of the Lorenz 84 model output, which is considered to be the "truth". Although the forecast quality drops down with the increase of lead time as expected, the BayesLSTM shows good skill in replicating the variations of the Lorenz 84 model, especially for the state-transitions of the Lorenz 84 system and the sinusoidal patterns of the eddy components, like the forecast of X around valid date 14 and the forecast of Y around valid date 16. This indicates that the BayesLSTM learns to predict the state of the Lorenz system. Considering the typical predicting process of an LSTM network, in which the whole time series of the Lorenz 84 system preceding the forecast time should be fed to the system, it implies that our BayesLSTM is well constrained by the Lorenz 84 model output. Given the fact that the learning process of a deep neural network is characterized by the relationship between input fields, it further indicates that the non-linear relations between the variables in this Lorenz 84 system, the westerly X and the large scale eddies Y and Z, were addressed by the BayesLSTM.

In addition, we plot the forecast trajectory in Figure 2d and compare it with the Lorenz model output to further evaluate the performance of BayesLSTM. It can be noticed that the forecast trajectory is close to the attractor and the "behavior" of the forecast trajectory as a function of lead time resembles the evolution of the Lorenz 84 model. The result is consistent with the assessment based on the time series of each component as shown in Figure 2a. As a follow-up check, we investigate the physical consistency of BayesLSTM forecasts via the log power spectrum density of forecast time series, which is shown in Figure 2c. Only the high frequency components of X (with the frequency between 0.9 and 1.5) differ from the Lorenz model output. In general, the power spectrum density of the BayesLSTM forecasts is similar to that of the Lorenz 84 model. This indicates that the phases of the waves simulated by BayesLSTM do not differ much from the Rossby waves in the Lorenz 84 model. Considering the time step (4 hours) and the damping time of the Lorenz system (5 days), such similarity over the whole frequency space reflects that the BayesLSTM can account for the dynamics of this Rossby wave system across different time scales, which potentially benefits from its ability of multiple-level information abstraction. Together with the similar amplitudes of waves displayed in Figure 2a, it implies that the BayesLSTM manages to learn the Rossby wave propagation. The interaction between the jet and eddy components in this simplified atmospheric circulation system and the forecasts are physically realistic.

In order to evaluate the probabilistic forecast skill of the BayesLSTM, we generated a 20-member ensemble by sampling the BayesLSTM network and the time series of these retrospective forecasts up to 3 lead days are shown in Figure 2b. The blue shades serve to approximate the error growth of the Lorenz 84 system, which are selected as the range between the current Lorenz model series persisting for 3 lead and lag days. Note that this selection is made based on the auto-correlation in Figure S2 and it aims to assist the evaluation of the probabilistic forecasts, specifically for the uncertainty estimation. It is observed that the forecast members are located around the Lorenz model output and the spread is comparable to the error growth of this Rossby wave system. This indicates that the spread of the BayesLSTM ensemble is neither over-dispersive nor under-dispersive. The probabilistic forecasts therefore address uncertainties in a reasonable way. Collectively, the development of these forecasts as a function of lead time in 2b are similar to the single forecast in 2a. This means almost all the ensemble members capture the properties of the propagating waves and the jet strength while allowing for the occurring of uncertainty. Consequently, the probabilistic forecasts generated by sampling the BayesLSTM are physically plausible.

Nevertheless, the BayesLSTM forecasts may lose skill at certain valid time. For instance, in Figure 2a and b between valid date 0 to 6, forecasts of X drift away unrealistically. This might result from the state-dependency of the BayesLSTM, or in general the state-dependency of any deep learning approaches. For a numerical model, it is common to have state-dependency, for example, the prediction of NAO/blocking events in medium-range forecasts (e.g., Parker et al., 2018). This may also apply to the deep learning approaches if the training data fails to provide adequate information for forecasting at some points.

3.2 Evaluate the BayesLSTM ensemble forecasts

A reliability assessment of probabilistic forecasts with the BayesLSTM ensemble was performed using the chosen metrics. The BayesLSTM ensemble consists of 20 members and they are evaluated against a deterministic forecast with persistence and a probabilistic forecast with the VAR model, which is also a 20-member ensemble. The results are shown in Figure 3. Regarding the CRPS score, in general the BayesLSTM ensemble forecast is better than the VAR ensemble forecast considering all the variables for almost all lead days. Only around day 1 for predictand X, the VAR ensemble forecast shows slightly better skill. The error growth of the BayesLSTM ensemble forecast is much slower than that of the VAR ensemble forecast. Given the definition of CRPS score, which provides a quadratic measure of discrepancy between the forecast cumulative density function (CDF) and the empirical CDF of the scalar observation (Gneiting et al., 2005), this indicates that the forecast CDF with the BayesLSTM centered around the Lorenz model output, while the forecast CDF with the VAR is relatively over-dispersive.

Regarding the RMSE shown in Figure 3b, forecasts with persistence are better than that with BayesLSTM and VAR ensemble concerning only X. This is consistent with the high auto-correlation of the zonal wind X shown in Figure S2. While for the eddy components Y and Z, the BayesLSTM provides much better forecasts within 3 lead days, with the averaged RMSE error smaller than the standard deviation of the full time series of the Lorenz 84 model output. Considering the nonlinear relation between the westerly X and large scales eddies Y and Z, this means that the BayesLSTM is able to preserve the physical consistency between the zonal wind and the propagation of large scale eddies in this atmospheric circulation system, and therefore produces better probabilistic forecasts. It is evident by analyzing the time series in Figure 2a, that the variations of Y and Z are well represented by the BayesLSTM forecasts up to a lead time of 3 days.

More information about forecast quality in terms of the trajectories, which intrinsically embody the properties of Rossby waves and jet strength, is reflected by the EuD in Figure 3c. Starting from the first forecast time step (4 hours), the BayesLSTM shows better forecast skill concerning the EuD. Although the EuD error grows with the increase of lead time for all the forecast methods, the BayeLSTM forecasts are better than the others for the whole inspected lead time range. Note that within 2 lead days, the EuD error of BayesLSTM is smaller than the standard deviation of the Lorenz 84 model output, which is about 0.6. Since the EuD of BayesLSTM ensemble forecast shown in Figure 3c is the average of 20 members with forecasts starting every time step, this implies that these ensemble members are able to replicate the patterns of the attractor and the spread of the ensemble is properly distributed around the target Lorenz 84 model trajectory. It further demonstrates that probabilistic forecast with BayesLSTM can address uncertainties adequately.

4 Discussion

We demonstrate the capability of BayesLSTM in probabilistic weather forecasting. Intuitively, by perturbing the Lorenz 84 model, it seems possible to compare the BayesLSTM forecasts to the perturbed Lorenz model output and check if the BayesLSTMs are able to address uncertainties in the initial conditions and model formulation, respectively. However, there is no objective way to determine the amplitude of the perturbation which can appropriately approximate the error growth in the Lorenz system that is analogue to a realistic dynamical system of Rossby waves on a jet. So this experiment is not feasible at the moment.

In addition, we extended the ensemble forecasts to more than 60 lead days and noticed that after 20 days, the forecast errors increase dramatically with the increase of lead time (not shown). From this point, it seems that the BayesLSTM is useful for medium-range forecasts and it is not suitable for seasonal forecast and climate change predictions. The outcome of this study is insufficient to prove that, either the Bayesian deep neural networks can mathematically represent the differential equations which depict the Lorenz 84 system with seasonal forcing (note that due to the features of deep learning and the nature of deep neural networks, there is no direct mapping between weight matrix in a trained BayesLSTM and Lorenz model parameters), or BayesLSTMs only abstract and store the physical linkages in a latent space and use them to produce memory-based forecasts at relatively short time scales. This can be explored in the future.

Although not the main topic of this paper, we note that the formulations of BDL are very similar to data assimilation, specifically the Bayesian data assimilation, which is extensively used in weather forecasting to combine the knowledge from observations and models, and deal with the uncertainty in the initial conditions (Evensen, 1994; P. L. Houtekamer & Mitchell, 1998; P. Houtekamer & Zhang, 2016). Based on the Bayes' theorem, it incorporates model knowledge into the prior and corresponding observations as likelihood, and treats the observation involving uncertainty estimation as posterior. Given the large dimensional systems, in reality approximate solutions are always made based on different methods, like variational methods, Kalman-based methods and particle filters (Navon, 2009).

Given the fact that forecasts with BayesLSTM stay close to the Lorenz 84 attractor, the BayesLSTM may be also chaotic. This question can be answered by the chaotic system diagnostics, for instance, with the Lyapunov spectrum (Broer et al., 2002; Freire et al., 2008). However, this is beyond our scope now but worth the effort in the future. So far, it can be concluded that BayesLSTM is a useful candidate for weather forecasts, at relatively small lead times up to several days. For a long term climate forecast, the BayesLSTM may not be a good choice in terms of the error accumulation and the lack of skill in physical model representation. Also, for simulating and forecasting changes in the climate system boundary condition uncertainty will need to be taken into account. This can be further tested by studies using observational data and climate model ensembles in the future.

5 Conclusion

In this study, we explored the potential of BDL for weather forecasting using the modified Lorenz 84 model as a model for the atmosphere. The probabilistic character of the BDL is addressed and assessed using the chaotic nature of the Lorenz 84 system with seasonal forcing as 'truth'. Specifically, we chose BayesLSTM as an example of BDL to forecast the Lorenz 84 model and evaluate its forecast skill. It was observed that the retrospective forecasts are similar to those of the Lorenz model output in the spatial and temporal domain. The forecast trajectories are close to the attractor. This indicates that

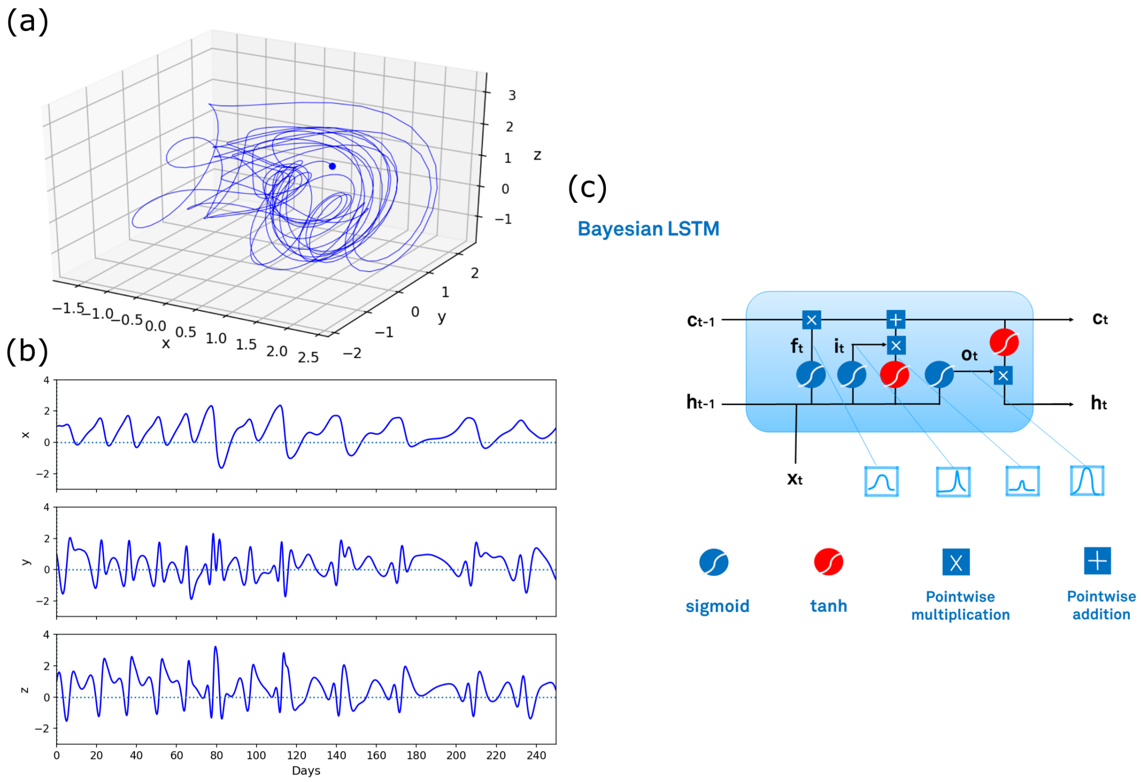

Figure 1. (a) Trajectory and (b) time series of each variable of the Lorenz 84 model with seasonal forcing. The sequences contain 250 days (1500 time steps) and the starting point is marked with a blue dot. (c) Structure of the Bayesian Long-Short Term Memory neural networks (Fortunato et al., 2017).

BayesLSTM is able to learn the propagation of Rossby waves in this atmospheric system, in terms of both the amplitude and phase. It further demonstrates that the BayesLSTM is able to replicate the interaction between the jet stream and large-scale eddies and thus the evolution of Rossby waves on a midlatitude jet. The forecasts get worse with increasing lead times due to the accumulation of errors, as expected.

The probabilistic forecast skill of BayesLSTM was analyzed and evaluated against persistence and a VAR model. We found that the BayesLSTM forecasts saturate around the model output considering both the sequences of each variable and the trajectory. In terms of the scores in the chosen metrics, the BayesLSTM shows better probabilistic forecast skill than persistence and the VAR model in the inspected lead days. It shows that the BayesLSTM is able to account for uncertainties relevant to the evolution of this simplified atmospheric circulation system, though the procedure differs from well-known NWP based approaches. Given the relatively low cost of ensemble forecasts compared to deterministic DNN and NWP systems, and the capacity in probabilistic forecasting, BayesLSTM, or in general BDL, is useful to produce fast and reliable probabilistic weather forecast and therefore is promising to enhance weather forecasting capabilities at short to medium-range timescales.

Acknowledgments

The authors gratefully acknowledge the support by the Netherlands eScience Center and

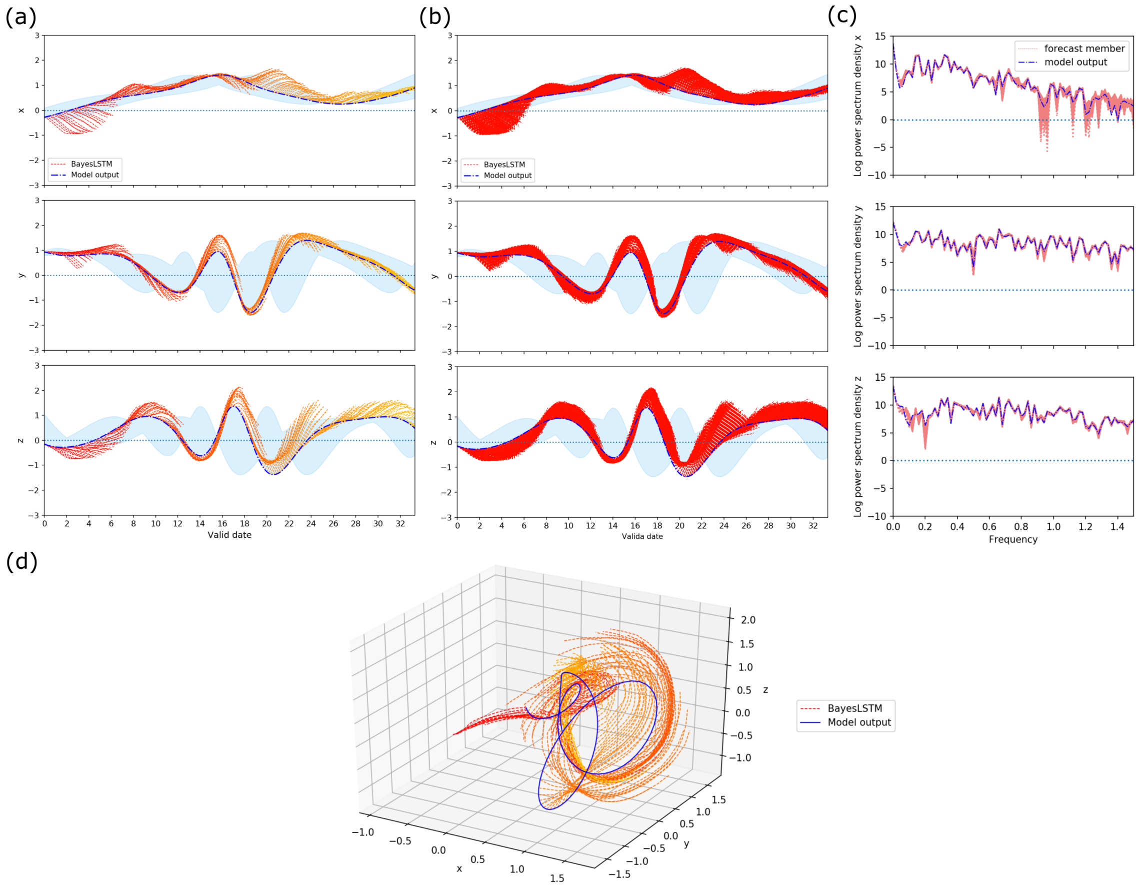

Figure 2. BayesLSTM retrospective forecasts up to a lead time of 3 days (18 time steps), with forecasts starting every time step (every 4 hours). (a) Time series of each variable (b) time series of a 20-member ensemble (c) logarithmic power spectrum and (d) trajectory in phase space. Except for (b) all the figures contain the results from a single BayesLSTM retrospective forecast. The Lorenz model output is included as reference (blue, labelled as "model output") and the blue shades indicate the range between the Lorenz model output persisting for 3 days, both lead (3 days forward) and lag (3 days backward).

Wageningen University. This study is supported by Blue Action project (European Union's Horizon 2020 research and innovation programme, grant number: 727852). We would like to thank SURFsara (Netherlands) for providing us their super computing infrastructure for our project. All data analyzed here are publicly available. The forecasts from BayesLSTM network can be accessed at a public github repository (https://doi.org/10.5281/zenodo.4494116, in the data folder), which also contains the forecasts from VAR model. The Lorenz 84 model output is available through Lorenz (1984), with modifications described in the section Methodology.

References

Bannister, R. (2017). A review of operational methods of variational and ensemble-variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, *143*(703), 607–633.

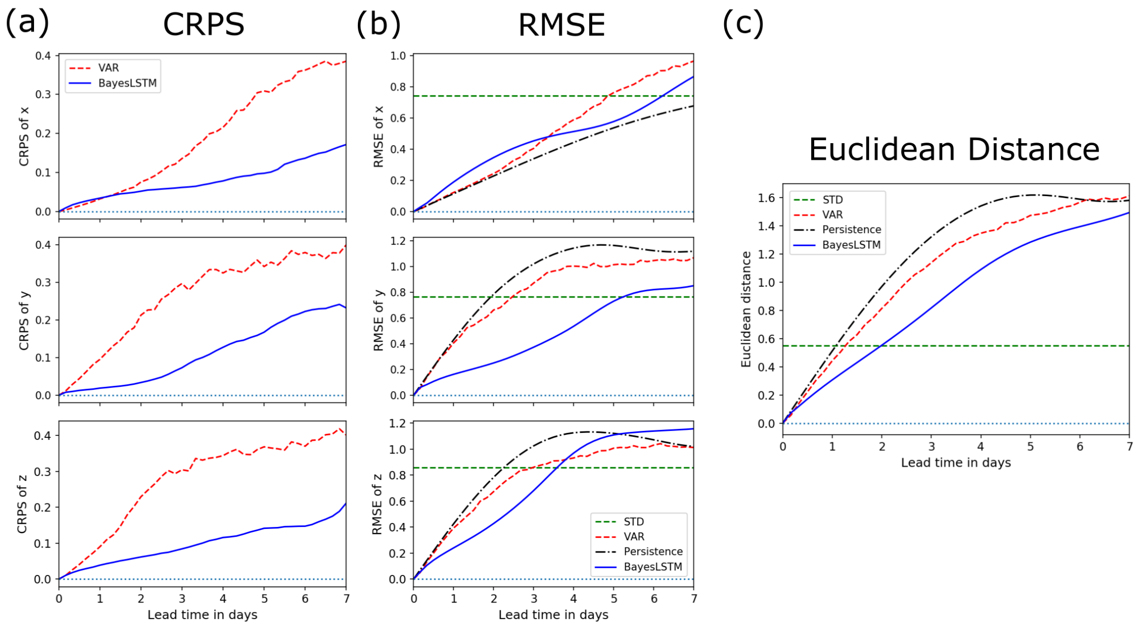

Figure 3. Skill evaluation of the BayesLSTM ensemble forecasts against VAR and persistence with (a) CRPS and (b) RMSE and (c) EuD, which are averaged over 200 forecasts starting every time step (4 hours). The standard deviation of the full time series of Lorenz model output (based on 250 days data) is included in (b) and (c) (green, labelled as "STD").

Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*.

Broer, H., Simó, C., & Vitolo, R. (2002). Bifurcations and strange attractors in the lorenz-84 climate model with seasonal forcing. *Nonlinearity*, *15*(4), 1205.

Buizza, R., Leutbecher, M., & Isaksen, L. (2008). Potential use of an ensemble of analyses in the ecmwf ensemble prediction system. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, *134*(637), 2051–2066.

Chattopadhyay, A., Hassanzadeh, P., Palem, K., & Subramanian, D. (2019). Data-driven prediction of a multi-scale lorenz 96 chaotic system using a hierarchy of deep learning methods: Reservoir computing, ann, and rnn-lstm. *arXiv preprint arXiv:1906.08829*.

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, *99*(C5), 10143–10162.

Fortunato, M., Blundell, C., & Vinyals, O. (2017). Bayesian recurrent neural networks. *arXiv preprint arXiv:1704.02798*.

Freire, J. G., Bonatto, C., DaCamara, C. C., & Gallas, J. A. (2008). Multistability, phase diagrams, and intransitivity in the lorenz-84 low-order atmospheric circulation model. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *18*(3), 033121.

Ghil, M., & Malanotte-Rizzoli, P. (1991). Data assimilation in meteorology and oceanography. In *Advances in geophysics* (Vol. 33, pp. 141–266). Elsevier.

Gneiting, T., Balabdaoui, F., & Raftery, A. E. (2007). Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *69*(2), 243–268.

Gneiting, T., Larson, K., Westrick, K., Genton, M. G., & Aldrich, E. (2006). Calibrated probabilistic forecasting at the stateline wind energy center: The regime-switching space–time method. *Journal of the American Statistical Association*, *101*(475), 968–979.

Gneiting, T., Raftery, A. E., Westveld III, A. H., & Goldman, T. (2005). Calibrated probabilistic forecasting using ensemble model output statistics and minimum crps estimation. *Monthly Weather Review*, *133*(5), 1098–1118.

Graves, A. (2011). Practical variational inference for neural networks. In *Advances in neural information processing systems* (pp. 2348–2356).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Houtekamer, P., & Zhang, F. (2016). Review of the ensemble kalman filter for atmospheric data assimilation. *Monthly Weather Review*, *144*(12), 4489–4532.

Houtekamer, P. L., & Mitchell, H. L. (1998). Data assimilation using an ensemble kalman filter technique. *Monthly Weather Review*, *126*(3), 796–811.

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems* (pp. 5574–5584).

Kingma, D. P., Salimans, T., & Welling, M. (2015). Variational dropout and the local reparameterization trick. In *Advances in neural information processing systems* (pp. 2575–2583).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444.

Leutbecher, M., Lock, S.-J., Ollinaho, P., Lang, S. T., Balsamo, G., Bechtold, P., ... others (2017). Stochastic representations of model uncertainties at ecmwf: State of the art and future vision. *Quarterly Journal of the Royal Meteorological Society*, *143*(707), 2315–2339.

Leutbecher, M., & Palmer, T. N. (2008). Ensemble forecasting. *Journal of computational physics*, *227*(7), 3515–3539.

Liu, Y., Bogaardt, L., Attema, J., & Hazeleger, W. (2020). Extended range arctic sea ice forecast with convolutional long-short term memory networks. *Monthly Weather Review*. (under review)

Lorenz, E. N. (1984). Irregularity: A fundamental property of the atmosphere. *Tellus A*, *36*(2), 98–110.

Milinski, S., Maher, N., & Olonscheck, D. (2020). How large does a large ensemble need to be. *Earth System Dynamics*, *11*, 885–901.

Navon, I. M. (2009). Data assimilation for numerical weather prediction: a review. In *Data assimilation for atmospheric, oceanic and hydrologic applications* (pp. 21–65). Springer.

Neal, R. M., & Hinton, G. E. (1998). A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models* (pp. 355–368). Springer.

Palmer, T. (2002). Predicting uncertainty in numerical weather forecasts. In *International geophysics* (Vol. 83, pp. 3–13). Elsevier.

Parker, T., Woollings, T., & Weisheimer, A. (2018). Ensemble sensitivity analysis of greenland blocking in medium-range forecasts. *Quarterly Journal of the Royal Meteorological Society*, *144*(716), 2358–2379.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., et al. (2019). Deep learning and process understanding for data-driven earth system science. *Nature*, *566*(7743), 195–204.

Scher, S., & Messori, G. (2018). Predicting weather forecast uncertainty with machine learning. *Quarterly Journal of the Royal Meteorological Society*, *144*(717), 2830–2841.

Shridhar, K., Laumann, F., & Liwicki, M. (2018). Uncertainty estimations by softplus normalization in bayesian convolutional neural networks with variational inference. *arXiv preprint arXiv:1806.05978*.

Shridhar, K., Laumann, F., & Liwicki, M. (2019). A comprehensive guide to bayesian convolutional neural network with variational inference. *arXiv preprint arXiv:1901.02731*.

Vandal, T., Kodra, E., Dy, J., Ganguly, S., Nemani, R., & Ganguly, A. R. (2018). Quantifying uncertainty in discrete-continuous and skewed data with bayesian deep learning.

In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining* (pp. 2377–2386).
Wang, H., Yu, Y., & Wen, G. (2014). Dynamical analysis of the lorenz-84 atmospheric circulation model. *Journal of Applied Mathematics*, *2014*.
Wang, H.-z., Li, G.-q., Wang, G.-b., Peng, J.-c., Jiang, H., & Liu, Y.-t. (2017). Deep learning based ensemble approach for probabilistic wind power forecasting. *Applied energy*, *188*, 56–70.
Wang, L., Yuan, X., Ting, M., & Li, C. (2016). Predicting summer arctic sea ice concentration intraseasonal variability using a vector autoregressive model. *Journal of Climate*, *29*(4), 1529–1543.
Zaier, I., Shu, C., Ouarda, T., Seidou, O., & Chebana, F. (2010). Estimation of ice thickness on lakes using artificial neural network ensembles. *Journal of Hydrology*, *383*(3-4), 330–340.

Supporting Information for "Exploring Bayesian deep learning for weather forecasting with the Lorenz 84 system"

Yang Liu[1,2], Jisk Attema[1], Wilco Hazeleger[3]

[1]Netherlands eScience Center, Amsterdam, the Netherlands

[2]Meteorology and Air Quality Group, Wageningen University, Wageningen, the Netherlands

[3]Faculty of Geosciences, Utrecht University, Utrecht, the Netherlands

Contents of this file

Introduction This supplementary material includes additional information regarding BDL and the variational inference method for training the Bayesian neural networks, namely the Bayes by Backprop. It also includes a brief introduction of the local re-parameterization trick and the evaluation metrics. A detailed explanation of the lead time dependent forecasts and numerical configurations of training and forecasting with BayesLSTM is provided.

S1. BDL and Bayes by Backprop In this section, we elaborate on BDL and Bayes by Backprop in detail. Deep neural networks are equipped with multiple layers of fixed weights to extract the multiple levels of abstraction in the training data (LeCun et al., 2015). These neural networks are deterministic and prone to overfitting and forecasts with over-confidence (Blundell et al., 2015). In order to incorporate uncertainty estimation during training and forecasting, we need to transform the deterministic neural network into a probabilistic structure and therefore we seek help from the Bayes' theorem.

To begin with, we have a training set $D = (x_i, y_i)_i$ which consists of input data $x \in \mathbb{R}^d$ and output data $y \in \mathbb{R}^d$. We aim for a probabilistic neural network $p(y|x, w)$ with a distributed weight w. To obtain the weight distribution, we apply the Bayesian inference and we will get the expression:

$$p(w|x, y) = \frac{p(y|x, w)p(w)}{p(x, y)} \tag{1}$$

By substituting $p(x, y)$ according to the law of total probability, we can obtain:

$$p(w|x, y) = \frac{p(y|x, w)p(w)}{\int p(y|x, w)p(w)dw} \tag{2}$$

The integral in the denominator requires coverage of all possible values of w, which is computationally prohibitive. This makes the true posterior probability distribution intractable (Blundell et al., 2015; Shridhar et al., 2018, 2019). To solve this, various approximation methods (e.g. maximum-a-posteriori scheme, variational inference schemes) were studied in the past (Graves, 2011; Shridhar et al., 2018). Among all of them, variational inference schemes are widely embraced and they work well for a wide range of BDL applications.

So far, three popular variational inference schemes are always used to approximate the intractable posterior: Monte Carlo Markov Chain (MCMC), Monte Carlo dropout (MCD) and Bayes by Backprop (BBB).

MCMC is a family of methods that combines stochastic variational inference and Monte Carlo approaches (Salimans et al., 2015). Unlike variational inference in general which takes a random draw from a simple variational distribution and keeps optimizing the distribution, MCMC methods subsequently apply a stochastic transition operator to the random draw, so as to cover the exact posterior distribution. As long as the iterations are sufficient, MCMC can approximate the exact posterior well. However, we do not know if the iterations are sufficient and this procedure is always very costly.

An alternative is MCD, which uses dropout to perform variational inference where the variational distribution comes from a Bernoulli distribution. Hence it is also known as Variational inference with Bernoulli Distribution (Gal & Ghahramani, 2015, 2016). This method makes use of dropout in each layer of neural network during training as well as testing and thus the process is equivalent to sampling from a Bernoulli distribution and provides a measure of uncertainty. Nevertheless, it provides a rough approximation of the target posterior and the control of uncertainty is limited by the complexity of the neural network.

Given the drawback of MCMC and MCD, in this study, we choose a more functional and affordable approach, BBB, to approximate the posterior. This method is a back-propagation (BP) compatible variational inference approach that estimates a density function with a known distribution and progressively update it with BP (Blundell et al., 2015; Shridhar et al., 2018). More details about BBB can be found in the section Methodology in the main body of the paper.

S2. Local re-parameterization method With the implementation of BBB and optimization function, it seems our BayesLSTM is ready for training. However, the whole procedure is not ready for back-propagation (BP) due to the stochastic node including $\mathcal{N}(\theta|\mu, \sigma^2)$ in our chosen variational distribution. In order to enable the optimization of the parameters of Gaussian posterior with BP, we introduce the local reparameterization method, which translates the global uncertainty in the weights into a form of local uncertainty (Kingma et al., 2015; Shridhar et al., 2019). In this case, we replace $\mathcal{N}(\theta|\mu, \sigma^2)$ with $w = \mu + \sigma * \epsilon$ where $\epsilon \sim \mathcal{N}(0, 1)$. More information about the local reparameterization method can be found in the related literature (Kingma & Welling, 2013; Kingma et al., 2015; Shridhar et al., 2018, 2019).

S3. Evaluation metrics In order to evaluate these ensemble forecasts, we include continuous ranked probability score (CRPS), root mean square error (RMSE) and Euclidean distance.

CRPS is a popular verification tool for the assessment of probabilistic forecast systems. It is used to evaluate an ensemble forecast against a single deterministic observation and it has the following form (Gneiting et al., 2005):

$$CRPS(F, v) = \int_{-\inf}^{\inf} [F(r) - H(r - v)]^2 dr \tag{3}$$

with F the predictive cumulative density function (CDF), v the verifying observation, r the threshold value, and $H(r-v)$ the Heaviside function which takes the value 0 when $r < v$ and otherwise 1.

RMSE score is also included in this study and it is defined in the following way so as to work with the probabilistic forecast results:

$$RMSE = \frac{1}{N}\Sigma_{n=1}^{N}\frac{1}{T}\Sigma_{t=1}^{T}\sqrt{(x_{pred}-x_{obs})} \tag{4}$$

with N the number of ensemble members, T the number of time steps, x_{pred} and x_{obs} the predicted and observed value.

Euclidean distance (EuD) is used to evaluate the similarity between trajectories, which is expressed as:

$$EuD = \frac{1}{N}\Sigma_{N=1}^{N}\frac{1}{T}\Sigma_{T=1}^{T}\sqrt{(x_{pred}-x_{obs})^2+(y_{pred}-y_{obs})^2+(z_{pred}-z_{obs})^2} \tag{5}$$

S4. Estimate uncertainty with BDL Typically, three types of uncertainties are considered: (1) Uncertainties in the initial conditions (2) Necessary approximations and corresponding uncertainties in the construction of a numerical model of the real atmosphere (3) Uncertainties posed by external forcing and boundaries. The last one is often ignored in an operational weather forecast system, but relevant for climate studies and local forecasts. Kendall and Gal (2017) explained how the model uncertainty and initial condition uncertainty are addressed by BDL respectively. Given the nature of the forward process of BDL, model uncertainty is addressed by the BDL via placing a prior distribution over a model's weight, which is already explained by the prior dependent part $KL[q(w|\theta)||p(w)]$ during the training of BDL when minimizing the KL divergence (Kendall & Gal, 2017).

The representation of initial condition uncertainty with BDL is less straightforward. During training, the BayesLSTMs approach output $y_j \in \mathbb{R}^d$ with an input $x_j \in \mathbb{R}^d$ and weight distribution w_j, as $P(y_j|x_j, w_j)$. However, there is always noise ϵ_j in the input fields (initial conditions) and this propagates through the network and affects the output, thus $P(y_j|x_j, w_j, \epsilon_j)$. With the help of Bayes rule, we bring in the pre-knowledge of model weight distribution as prior and search for the best weight distribution as posterior that is able to achieve the testing set $\hat{x_j}$ and $\hat{y_j}$ following $P(\hat{y_j}|\hat{x_j}) = \mathbb{E}_{P(w_j|D)}[P(\hat{y_j}|\hat{x_j}, w_j)]$. To simplify the problem, we assume a Gaussian distribution for the prior and the weight distribution then can be solved via maximum a posteriori (MAP). Due to the use of the BBB approach, we need to solve the KL divergence and the knowledge of observation is brought in via the data-dependent term $\mathbb{E}_{q(w|\theta)}[log\ p(D|w)]$ in equation 3 in the paper, and finally the $-log\ p(D|w^{(j)})$ in equation 4. It then can be reformulated as:

$$\mathbb{L}(\theta, D) = -\frac{1}{N}\sum_{j=1}^{N} log\ p(y_j|f^{\hat{w_j}}(x_j, \epsilon_j)) \tag{6}$$

If we assume the error in the output is Gaussian, then $-logp(y_j|f^{\hat{w_j}}(x_j, \epsilon_j)) \propto \frac{1}{2\sigma^2}||y_j - f^{\hat{w_j}}(x_j)||^2 + \frac{1}{2}log\sigma^2$. The shape of this Gaussian probability is determined by the variance σ, which is in this case the deviation between the ground truth in reality and the uncertainty corresponding to the noise in the input fields ϵ_j. As a result, the data dependent loss evolves as:

$$\mathbb{L}_{NN}(\theta, D) = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{2\sigma(\epsilon_j)^2}||y_j - f^{\hat{w_j}}(x_j)||^2 + \frac{1}{2}log\ \sigma(\epsilon_j)^2 \tag{7}$$

with $\sigma(\epsilon_i)$ indicating the variance as a function of noise in the input fields, which represents the uncertainty in the initial conditions. In other words, uncertainty in the initial condition is modeled through the neural networks by generating a distribution over the output of the model. Note that this is recognized as heteroscedastic and aleatoric uncertainty in machine learning and it does not exist in a non-Bayesian (deterministic) neural network since this observation noise parameter ϵ is fixed as part of the model's weight decay and therefore neglected after training (Kendall & Gal, 2017). So far, our analysis is based on the hypothesis of linear propagation of expectation with variational inference in the LSTM (Fortunato et al., 2017). However, it should be noted that this becomes different when forecasting more than one step ahead since there is also uncertainty coming from the forecasts of previous time steps.

S5. Lead time dependent forecast and numerical configurations The whole numerical processes of training the BayesLSTM and making lead time dependent forecasts with BayesLSTM are described in details in this section. We generate time sequences of variable X, Y and Z with modified Lorenz 84 model, which including 1500 time steps in total. We perform sequence-to-one forecast with BayesLSTM and the first 1300 time steps are used for training. During training, in each epoch we pass 3 x 1300 points to the BayesLSTM model. The model samples its weight distribution at each time step (with 3 points X, Y and Z passed to the model) and after 1300 iterations we perform back-propagation to update the BayesLSTM model. The procedure will be repeated until the maximum epochs are reached (or the early stop module is called, depending on the setup).

The validation set consists of 200 time steps. The forecasting procedure is the same as the training process, except that there is no back-propagation and we perform lead time dependent forecasts (Liu et al., 2020). At each forecast time step t_n, we use the Lorenz model output (observation) to initialize the model and make forecasts, which can be described by the equation below:

$$(X, Y, Z)_{pred[t_n]} = BayesLSTM((X, Y, Z)_{obs[t_0,t_1,t_2,...,t_{n-1}]}) \tag{8}$$

with $(X, Y, Z)_{pred[t_n]}$ denotes the forecast X, Y and Z at time step t_n, and $(X, Y, Z)_{obs[t_m]}$ represents the Lorenz model output of X, Y and Z at time step t_m with $m < n$. It is still sequence-to-one prediction, which means data at time step t_0 to t_{n-1} will be passed to the BayesLSTM model to make the forecast for the time step t_n.

For one more lead time step t_{n+1}, the forecast is based on both the model output from t_0 to t_{n-1} and the forecast of time step t_n, and therefore it can be expressed as:

$$(X, Y, Z)_{pred[t_{n+1}]} = BayesLSTM((X, Y, Z)_{obs[t_0,t_1,t_2,...,t_{n-1}]} + (X, Y, Z)_{pred[t_n]}) \tag{9}$$

It can be noticed that starting from the time step t_{n+1} (the second lead time step), the forecast quality also depends on the forecasts of previous time steps. This helps us explore how far the BayesLSTM can predict the Lorenz system, which corresponds to the actual predictability of a weather and climate system.

References

Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*.

Fortunato, M., Blundell, C., & Vinyals, O. (2017). Bayesian recurrent neural networks. *arXiv preprint arXiv:1704.02798*.

Gal, Y., & Ghahramani, Z. (2015). Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*.

Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning* (pp. 1050–1059).

Gneiting, T., Raftery, A. E., Westveld III, A. H., & Goldman, T. (2005). Calibrated probabilistic forecasting using ensemble model output statistics and minimum crps estimation. *Monthly Weather Review*, *133*(5), 1098–1118.

Graves, A. (2011). Practical variational inference for neural networks. In *Advances in neural information processing systems* (pp. 2348–2356).

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems* (pp. 5574–5584).

Kingma, D. P., Salimans, T., & Welling, M. (2015). Variational dropout and the local reparameterization trick. In *Advances in neural information processing systems* (pp. 2575–2583).

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint* *arXiv:1312.6114*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444.

Liu, Y., Bogaardt, L., Attema, J., & Hazeleger, W. (2020). Extended range arctic sea ice forecast with convolutional long-short term memory networks. *Monthly Weather Review*. (under review)

Salimans, T., Kingma, D., & Welling, M. (2015). Markov chain monte carlo and variational inference: Bridging the gap. In *International conference on machine learning* (pp. 1218–1226).

Shridhar, K., Laumann, F., & Liwicki, M. (2018). Uncertainty estimations by softplus normalization in bayesian convolutional neural networks with variational inference. *arXiv preprint arXiv:1806.05978*.

Shridhar, K., Laumann, F., & Liwicki, M. (2019). A comprehensive guide to bayesian convolutional neural network with variational inference. *arXiv preprint arXiv:1901.02731*.

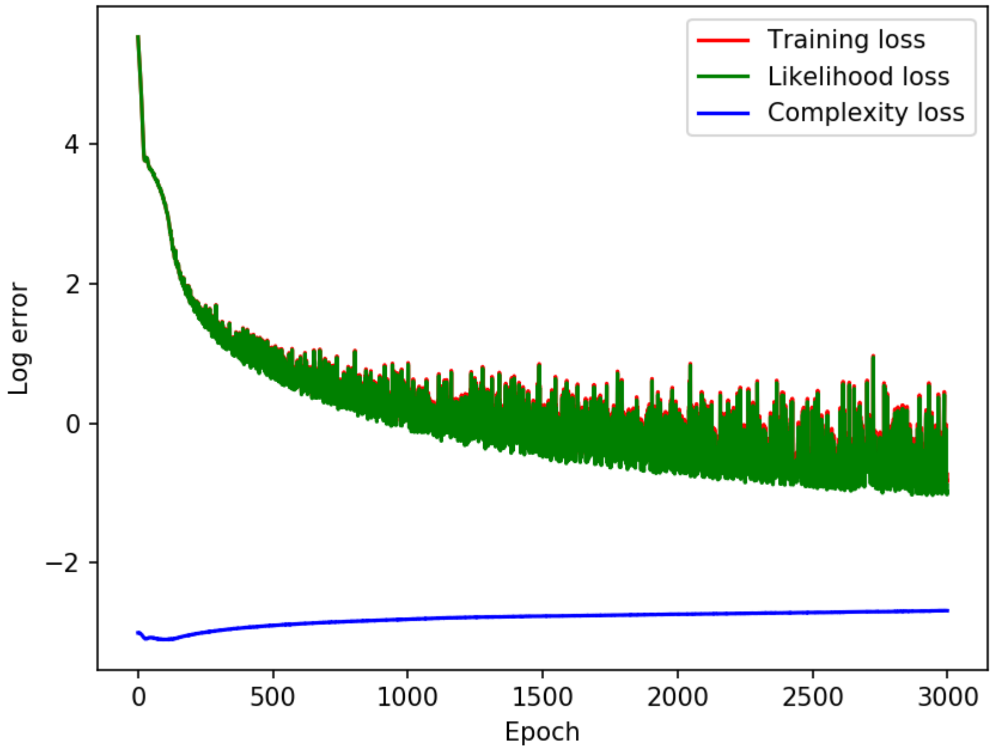

Figure S1. Training loss in the logarithmic form. The likelihood loss and complexity loss are included separately.

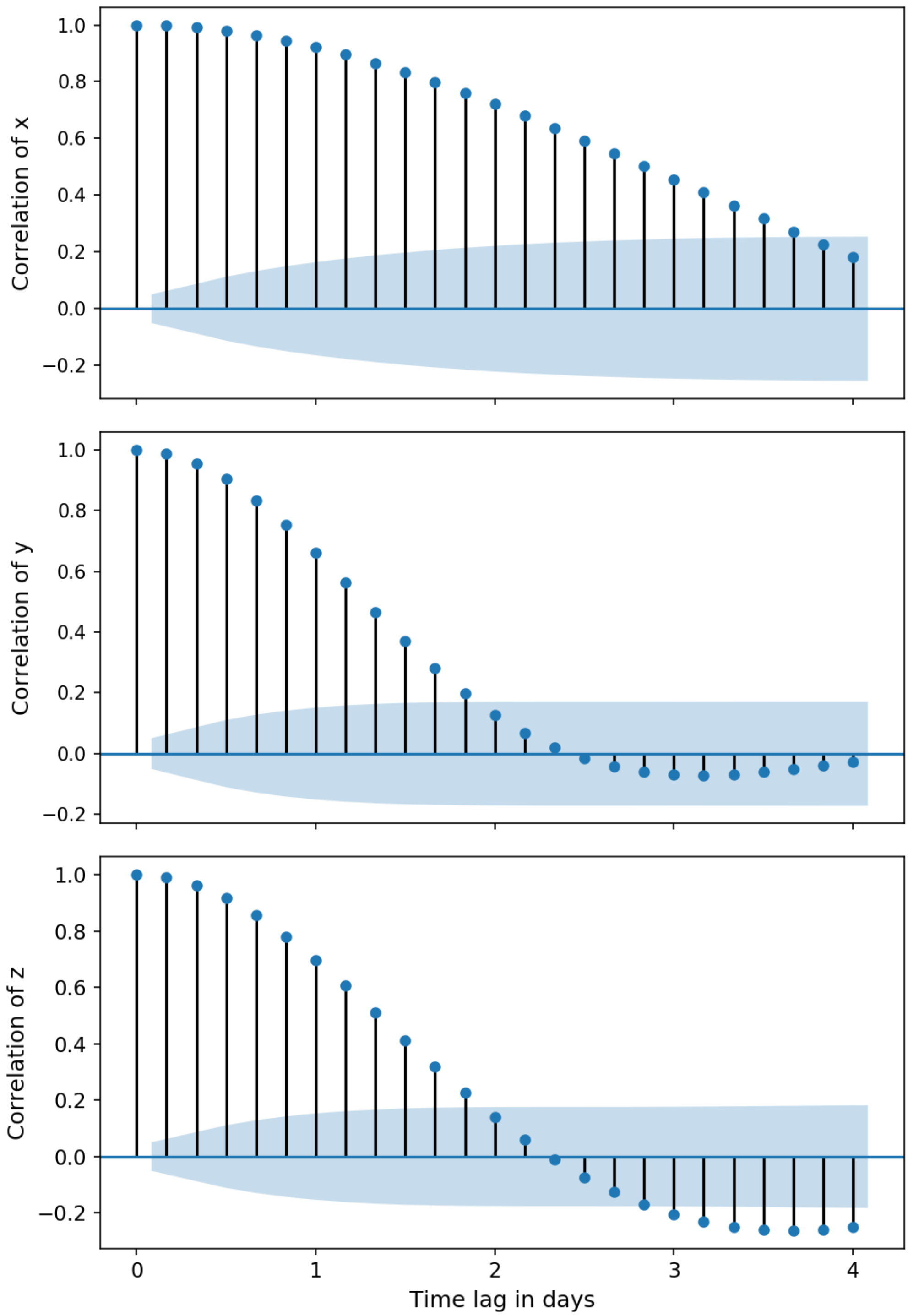

Figure S2. Autocorrelation of each variable of the Lorenz 84 model output. The blue shades indicate the p-value in null hypothesis significance testing.