PHILOSOPHICAL TRANSACTIONS A

rsta.royalsocietypublishing.org

Research

Article submitted to journal

Subject Areas:
Applied Machine Learning

Keywords:
Deep Learning, Weather Uncertainty Quantification, Ensemble Post-Processing, Extreme Weather Events

Author for correspondence:
Peter Grönquist
e-mail: petergro@ethz.ch

Deep Learning for Post-Processing Ensemble Weather Forecasts

Peter Grönquist*, Chengyuan Yao*, Tal Ben-Nun*, Nikoli Dryden*, Peter Dueben†, Shigang Li*, Torsten Hoefler*

*ETH Zurich, 8092 Zürich, Switzerland.
†ECMWF, Reading RG2 9AX, United Kingdom.

Quantifying uncertainty in weather forecasts is critical, especially for predicting extreme weather events. This is typically accomplished with ensemble prediction systems, which consist of many perturbed numerical weather simulations, or trajectories, run in parallel. These systems are associated with a high computational cost and often involve statistical post-processing steps to inexpensively improve their raw prediction qualities. We propose a mixed model that uses only a subset of the original weather trajectories combined with a post-processing step using deep neural networks. These enable the model to account for non-linear relationships that are not captured by current numerical models or post-processing methods. Applied to global data, our mixed models achieve a relative improvement in ensemble forecast skill (CRPS) of over 14%. Furthermore, we demonstrate that the improvement is larger for extreme weather events on select case studies. We also show that our post-processing can use fewer trajectories to achieve comparable results to the full ensemble. By using fewer trajectories, the computational costs of an ensemble prediction system can be reduced, allowing it to run at higher resolution and produce more accurate forecasts.

© The Authors. Published by the Royal Society under the terms of the Creative Commons Attribution License http://creativecommons.org/licenses/by/4.0/, which permits unrestricted use, provided the original author and source are credited.

THE ROYAL SOCIETY PUBLISHING

1. Introduction

Operational weather predictions have a large impact on society. They influence individuals on a daily basis, and in more severe cases, save lives and property by predicting extreme events such as tropical cyclones. However, developing reliable weather prediction systems is a difficult task due to the complexity of the Earth System and the chaotic behaviour of its components. Small errors introduced by observations, their assimilation, and the forecast model configuration escalate chaotically, leading to a significant loss in forecast skill within a week. Numerical Weather Prediction (NWP) is based on computer models solving complex partial differential equations at limited resolution. To be useful, weather forecasts try to estimate the uncertainties in predictions using ensemble simulations, where a forecast model is run a number of times from slightly different initial conditions, parameter values, and stochastic forcing. The resulting spread of predictions among ensemble members provides an estimate for the prediction uncertainty. This enables us to estimate the probability of, for example, precipitation for a specific location and time of day as well as the probability of a tropical cyclone hitting a large city.

In this paper we will focus on post-processing ensemble predictions performed at the European Centre for Medium-Range Weather Forecasts (ECMWF) [1] using deep neural networks (DNNs). ECMWF runs an operational forecast that consists of one high resolution (9 km grid) deterministic forecast (HRES), and an ensemble (ENS) with 51 members at a lower resolution (18 km), of which one is the unperturbed control trajectory. Each ensemble member starts from slightly different initial conditions and uses a different stochastic forcing in the physical parameterisation schemes of subgrid-scale processes — so-called stochastic parameterisation schemes. While ensemble methods have become a standard tool for numerical weather predictions, there is an ongoing discussion on how many ensemble members should be used. Larger ensembles allow for a better sampling of the probability density function (PDF) of predictions. However, computing power is limited and forecasts are bound by strict operational time windows of a couple of hours. Smaller ensembles would therefore allow individual members to run at higher resolution, likely resulting in better forecasts by each ensemble member [2].

The demand for ever more precise and dependable forecasts has led NWP methods to rank amongst the scientific domains with the most significant demand for supercomputing time [3–6]. As such, the NWP field is constantly looking for new methods to improve accuracy and reduce the computational cost of its models. This is where the recent advances in DNNs [7] become relevant. The breadth of tasks and efficient inference DNNs enable has made them a very attractive option for improving weather forecasts [8–12]. Related studies have also shown their capabilities for predicting chaotic behavior [13,14]. However, the full potential of these methods remains unexplored in many areas of NWP.

We use convolutional neural networks (CNNs) [15] and locally connected networks (LCNs) [16] to both improve forecast skill and reduce the computational requirements for NWP. We approach these goals through three different tasks: Uncertainty Quantification, Bias Correction, and PDF Calibration. Each is a different task that is addressed with a different neural network. First, uncertainty quantification is usually performed by examining the spread (standard deviation) of the forecasting ensemble. Here, we train a neural network to produce a similar spread as an ensemble, using only a small fraction of the ensemble members as input. We achieve a relative RMSE improvement of over 16% in forecast ability compared to using five of ten ensemble members to predict temperature. Second, we train a neural network to predict a point-wise bias to account for local trends in weather patterns. This results in a relative RMSE improvement of 7.9% on temperature. Lastly, we calibrate the ensemble PDF given a bias-corrected input, using our uncertainty quantification network. This results in a forecast skill increase of over 14.5% using only half the trajectories of a full ensemble. The reduced number of input ensemble forecasts also allows NWP to be run at a fraction of the cost of additional trajectories. Prediction time is further reduced by making use of high throughput graphics processing units (GPUs) for DNN inference.

Our code and data are publicly available[1].

(a) Related Work

There have been many works leveraging the modelling capabilities of neural networks (NNs) for NWP. Early attempts at applying shallow NNs showed success in emulating physical processes and saving computational power [17]. Since then, building on recent DNN developments, much effort has gone into applying NNs to weather nowcasting [18–21]. Nowcasting focuses on the emulation of physical processes for short term (up to six hours), high-resolution forecasts. Other works have also shown the significant capabilities of DNNs to predict longer ranging forecasts and extreme weather patterns [12,22–26].

In contrast, we focus on the post-processing of operational medium-range ensemble forecasts and the prediction of extreme weather events. Post-processing ensemble outputs has been a long-standing effort in the weather forecasting community. Methods such as Ensemble Model Output Statistics (EMOS) [27] and Bayesian Model Averaging (BMA) [28] currently allow for improvements of the raw ensemble forecast skill. Hamill and Whitaker [29] show initial explorations of those techniques on re-forecast datasets, also used in this paper, for temperature at 850 hPa (T850) and geopotential at 500 hPa (Z500). Advances in neural networks have only recently reached the field of ensemble models in weather forecasting, focusing on its application to specific weather stations [9,30] or global interpolations [31]. We expand on this work by applying DNNs on the novel task of improving the forecast skill for global predictions, specifically extreme weather forecasts, while reducing their computational costs.

2. Data

The quality of ensemble forecasts has improved significantly over the last decades and ensemble predictions are using increased resolutions and numbers of trajectories. Learning from past ensemble predictions would therefore lead to inconsistencies as the correction for mean and spread would need to adjust for changes in the quality of predictions over time. To address this, *re-forecasts* [32] apply current state-of-the-art forecast models to past measurements.

We use data from re-forecast ensemble experiments at ECMWF. These are routinely generated to provide an estimate of the "climate" of the forecast model for each date of the year, which can be used to remove model drifts during post-processing and for measuring the generic skill of the forecast system [33,34]. The re-forecast experiments run a 10-member ensemble (ENS10) and an unperturbed control experiment. Simulations use 91 vertical levels, the spectral representation of the model fields is truncated at the global wavenumber 639, and a cubic octahedral reduced Gaussian grid is used for the representation of model fields in grid-point space that provides an approximately uniform distribution of grid-cells on the sphere with a 18 km grid-spacing (the "TCo639" grid [35]). Simulations are performed with the same system for 1999-2017, with two forecast simulations starting each week, providing a large dataset with consistent forecast quality.

To fully train our networks and evaluate their forecast skill we also need ground truth weather conditions at specific forecast lead times. For this, we use the fifth major ECMWF ReAnalysis (ERA5) [36], which includes data on weather from 1979 up to the present[2]. Compared to re-forecasts, reanalysis datasets are produced by applying a constant stream of observations through state-of-the-art data assimilation on state-of-the-art forecast models used for decade-long simulations. In ERA5, this process generates reanalysis fields that are available at an hourly frequency for over 300 parameters.

[1] https://github.com/spcl/deep-weather
[2] Available for download under https://cds.climate.copernicus.eu/

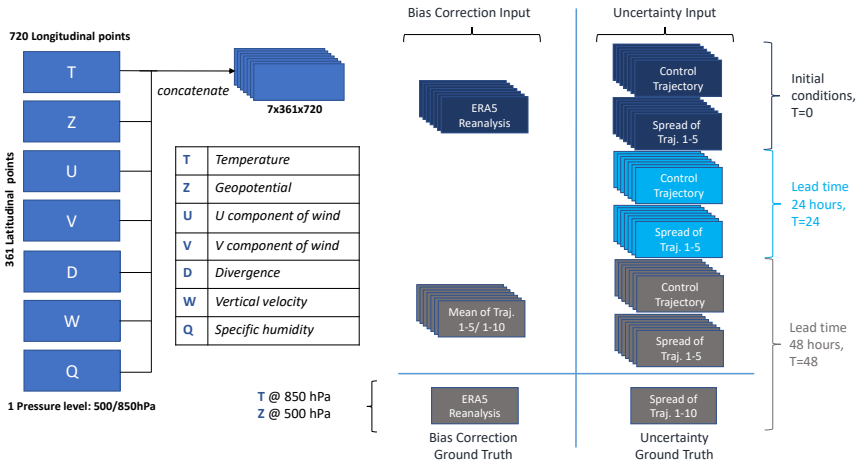


Figure 1: Inputs and ground truths to our neural networks.

(a) Data Selection

We use the ENS10 dataset for our forecasts and make use of ERA5's constant data assimilation product as ground truth, given the ENS10 forecast lead times. ENS10 and ERA5 both provide global data, which we interpolate to a latitude/longitude grid with a 0.5 degree resolution. We do this to avoid the native grid that was used within simulations, as it is unstructured in the longitudinal direction. While the use of latitude/longitude grids does lead to over-saturation of gridpoints in the poles, it simplifies our models; we leave the use of unstructured grids to future research. We also focus on a single pressure level for each model. When predicting temperature at 850 hPa (T850), we provide all input fields at 850 hPa. Similarly, when predicting geopotential at 500 hPa (Z500) all input fields are at 500 hPa. The years 1999-2013 are used for training, 2014-15 for validation, and 2016-17 for testing. Since the datasets are re-forecasts and a reanalysis, there is no difference in data assimilation and predictions between older and more recent dates, and therefore the selection of consecutive years should not have a major impact. We have verified that the selection of different training, validation and testing splits only has a minor impact on results (e.g., we see 7.6% improvement with our bias correction network, versus an average of 8.3% when performing cross-validation). Furthermore, we select these (most recent) years as it is our goal to model and evaluate our networks' capabilities to predict future weather given training-input from the past. The effects of climate change on the uncertainty of forecasts are currently being explored [37]. For our selected parameters and years, these effects are low. It is, however, important to use complete years, as different seasons demonstrate different weather patterns. We target forecasts with a lead time of 48 hours, and use the reduced ensemble forecasts for 0, 24, and 48 hour lead times as inputs.

(b) Data Preprocessing

As the datasets consist of several terabytes of data, we set up a data preprocessing pipeline to enable faster training. We first select the relevant inputs and labels to each of our respective models from the data provided in GRIB [38] format. We then convert the data from a 16-bit fixed point format to 32-bit floating point. This simplifies and speeds up training, while not impacting the results. Finally, we standardize (to zero mean, unit variance) our features and save them in the TFRecord format, which is the preferred dataset file extension in the TensorFlow deep learning framework. The resulting inputs and training targets can be seen in Figure 1. We base our model inputs on the first five ENS10 trajectories as we observed no significant differences in the average means or spreads when using different selections.

For DNNs to learn local weather patterns, it is important to keep local spatial differences in variability (coherence) when standardizing meteorological data [24]. However, if only one value per mean and standard deviation are applied to scale values on the whole globe, there will be massive differences for specific regions, e.g., different means and standard deviations closer to the poles compared to the equatorial region. This can lead to poor accuracy when applying CNNs that are translation-invariant. At the same time, just applying gridpoint-wise standardization will result in losing important information, otherwise represented through the coherence.

To remedy this problem, we apply a heuristic we refer to as Local Area-wise Standardization (LAS) (see Figures 2 and 3). First we apply a moving average and moving standard deviation filter on our training set. We use a step size of one and a filter size of 7×7 (the largest CNN filter size we apply in our DNNs). Then, as our mean and standard deviation maps now have reduced dimensions, we pad them using the edge values for latitudes and a wrap around for longitudes. Finally, we apply a Gaussian filter (truncated at four standard deviations) with a large standard deviation (of 10) to the padded result. This upscaling method, first padding and then blurring the upscaled feature map with a gaussian filter, allows for the coherence to be kept between singular grid points.

Using LAS we notice a relative improvement in our DNN results of around 15% for spread prediction on our validation sets, as well as faster convergence times compared to applying the same standardization for all grid points. However, as expected, we see no difference when using it with our LCNs, which are not translation-invariant (see Section 3(b)). The method is not fine-tuned and serves as an initial effort to reduce the workload on the neural networks by already accounting for reoccurring patterns, such as higher mean temperatures around the equator and reduced standard deviations.

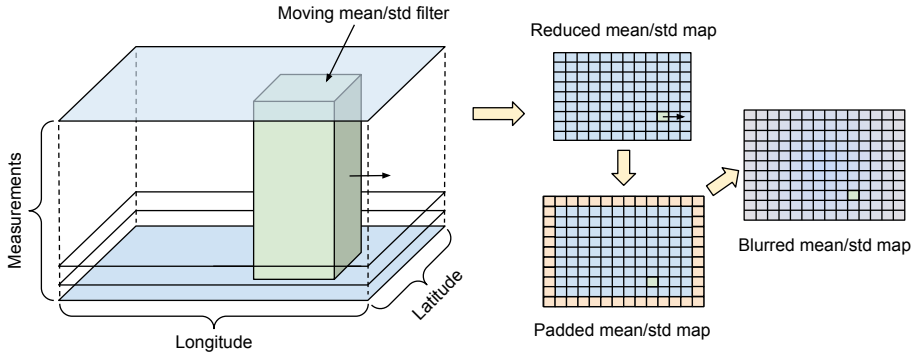
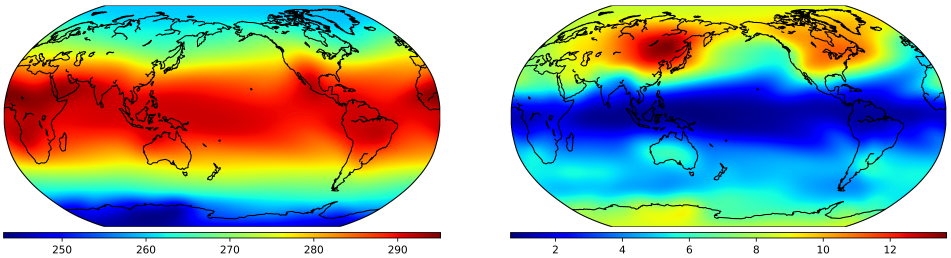


Figure 2: Local Area Standardization. The process is done twice, once by taking the mean of the moving filter, and once by taking its standard deviation, thereby obtaining a mean and standard deviation (std) map respectively.



(a) Mean values.

(b) Standard deviation values.

Figure 3: LAS values for Temperature at 850 hPa (T850), years 1999-2013 of ENS10.

3. Neural Networks

We develop separate neural networks for our uncertainty quantification and output bias correction tasks, which are described throughout this section: a residual neural network composed of Inception-style modules [39] and a U-Net [40] architecture with an additional locally-connected layer, respectively. While prior work in weather uncertainty prediction [31] used a 3D U-Net [41] architecture, the DNNs we develop perform better for our tasks.

Residual connections [42] (also called "skip" connections) pass unmodified features between layers that are not directly connected to each other, allowing them to be directly used by later layers. Such connections were found to be crucial for our results. Indeed, Chen et al. [43] demonstrated that applying successive residual connections has many similarities to ordinary differential equations. We also considered recurrent neural networks, but do not apply them here due to the short input sequences and lack of improvement in prior work [31].

(a) Uncertainty Quantification Model

There are many uncertainties present in NWP models and data. Data, or aleatoric, uncertainty stems from observational measurement noise, while model uncertainty comes from structural (e.g., forecast model) and parametric uncertainties. In addition to these inherent uncertainties, we also introduce a structural uncertainty by applying a common assumption of NWP that the distribution of errors and uncertainties for meteorological fields, which are represented by the distribution of ensemble members in ensemble predictions, follow Gaussian distributions. Our DNN is able to address data and structural uncertainties stemming from NWP; however, we cannot address parametric uncertainties, as the data assimilation pipelines and forecast models we use are fixed and used as prediction labels. More specifically, to reduce computational requirements, our DNN initially aims to predict the full ensemble spread using only a subset of NWP ensemble trajectories. The architecture is summarised in Figure 4.

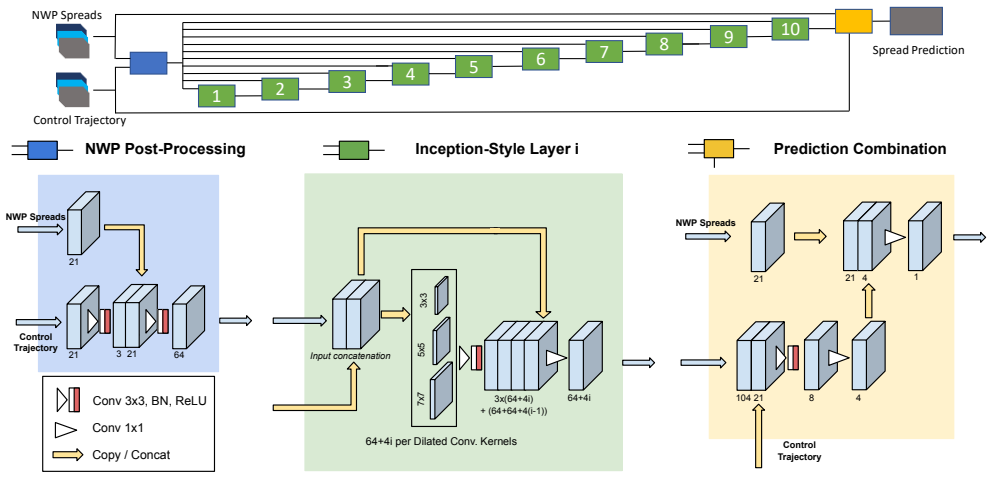


Figure 4: Spread Prediction Network for Uncertainty Quantification. All layers marked *Conv* have a kernel dimension of 1×1 and are meant to reduce the number of filters. For other convolutional layers we use Batch Normalization (BN) and ReLU activations on the outputs.

The non-linear nature of our DNN model, which introduces a previously unused structure in NWP and statistical post-processing, combined with the forecasts of the reduced NWP ensemble, helps address the original structural uncertainty. Such a design also allows our model to take into account deterministic forecasts, which it would otherwise struggle to learn. Additionally,

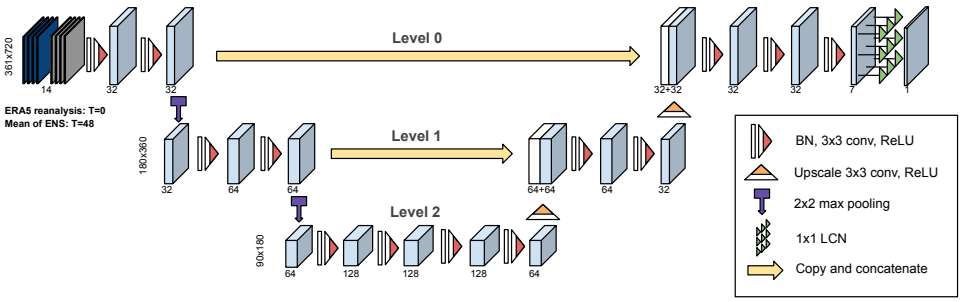


Figure 5: Output Bias Correction model, based on a three-level U-Net and added LCN structure.

as CNNs are relatively robust to noise, they are naturally able to account for data uncertainty. We also perform a minimal post-processing on NWP output, which reduces the number of parameters by encoding all input features aside from spreads into reduced dimensions. This output is then used for all subsequent steps.

The core of the DNN is based on the ResNet architecture [42]. We use ten *Inception-style modules* [39] with residual connections; we did not see any improvement with more layers. Each Inception-style module is composed of three parallel dilated convolutions (where dilation refers to an increased stride between the convolution kernel elements), allowing the network to learn differently-sized receptive fields (local regions). We also perform a channel-wise concatenation of the post-processed NWP output to the input of each Inception-style layer (Figure 4, top). This allows the network to prioritise between different lead times from NWP forecast spreads and its own outputs. Using auxiliary losses for different lead times and depths performed worse than pure NWP predictions.

Finally, the output of the last Inception-style module is combined with the NWP spread through a weighted mean. This guarantees the network performs at least as well as the NWP spread used as input during training. By combining this model with our bias correction model and training with ERA5 data as the ground truth (see Section 3(c)), it is possible to also account for parametric uncertainty. This allows our combined networks to cover all types of uncertainty.

(b) Output Bias Correction Model

Our output bias correction model, summarised in Figure 5, corrects for weather-dependent, local biases in NWP forecasts. It is trained using the mean ensemble predictions with a 48-hour lead time and ERA5 data as ground-truth. Since the forecast can resemble the ground-truth, a straightforward predictor will closely resemble the identity function. Prior research [42] suggests that approximating an identity mapping with several non-linear layers is difficult. We therefore train our model to predict the difference between the NWP prediction and the ground-truth.

The network is based on a U-Net structure, which repeatedly convolves and downscales inputs, followed by similarly upscaling the features (i.e., forming a "U" shape, as seen in Figure 5). Specifically, each scaling operation is applied after several layers of convolution. Residual connections are also used between the down- and upscaling sides. We make three key changes to adapt the standard U-Net to our task. First, instead of up-convolution, we use bilinear interpolation to upscale, followed by a 3×3 convolution with stride 1. This is due to checkerboard artifacts that are known to appear when only using a simple deconvolution operation [44]. Second, we reduce the number of levels in the U-Net, from five levels to three, as we found using additional levels resulted in overfitting on our data. Finally, we reduced the number of filters in each convolution by half, as we observed no additional performance improvements by using more.

As we aim to predict the bias emerging from specific regional patterns, the translational invariance of regular convolution hinders performance. We therefore use a locally-connected network (LCN) as the last layer. LCNs perform a similar operation to regular convolution, but

instead of sharing filters across all spatial points, independent filters are used for each output. When training, we apply ℓ_1 regularization on the difference between all adjacent filters in an LCN, to encourage adjacent filters to learn similar weights; for an infinite regularization parameter, the LCN converges to a convolutional layer. This helps avoid overfitting.

In order to remain computationally efficient, our best models first use a U-Net to perform feature extraction and then apply an LCN to obtain our final output bias correction. As the U-Net is able to learn long-range dependencies, a single LCN with 1×1 kernels is sufficient to learn the gridpoint-wise dependencies, and we observed no improvements by using larger filter sizes.

(c) Metrics

As our models solve different tasks, we need to use different metrics to evaluate them. When training, we treat both uncertainty quantification and output bias correction as regression problems, and aim to predict extreme cases (outliers). The ENS10 spread is used as ground truth for the uncertainty network, while the ERA5 values are used for the bias correction. Initially, we train both networks on the mean-squared error (MSE) and evaluate them with root mean-squared error (RMSE). However, when predicting the spread, the results lack the sharp edges that exist in the original forecasts. In computer vision, this problem is mitigated using the Structural SIMilarity (SSIM) metric [45]. There can be infinitely many solutions to the task of minimising RMSE. While remaining within the realm of these solutions, the SSIM measures the structural similarity between two images, with 1.0 being a perfect match. Therefore, we switch to using the negative mean SSIM of our prediction compared to the full ENS10 ensemble as our training loss for the uncertainty quantification model.

To then gain an understanding of the forecast skill of our combined predictions and of the ENS10 forecasts, we use the Continuous Ranked Probability Score (CRPS) [46]. CRPS, generally used to measure whether ensemble methods represent uncertainty correctly, is the integral of the square of the difference between the Cumulative Distribution Function (CDF) of the probabilistic predictions F and the ground truth y (see Figure 6):

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{\infty} [F(x) - \mathbf{1}_{x>y}]^2 dx$$

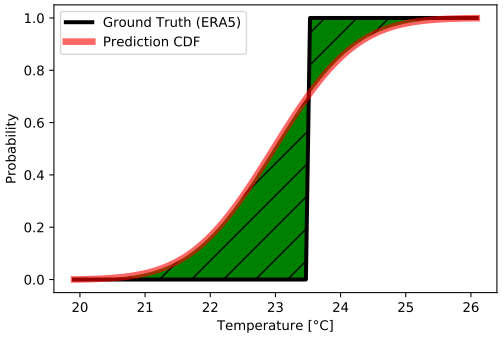


Figure 6: Visualisation of CRPS for a temperature prediction. CRPS is calculated as the integral of the square of the green area.

Here, $\mathbf{1}_{x>y}$ is the indicator function (equivalent to the CDF of a deterministic value). In the following, we do not only combine the uncertainty and bias correction networks to calculate CRPS, but also perform PDF calibration by training a combination of both in a network that is optimised to minimise the CRPS. We achieve this by replacing the labels of our uncertainty quantification network, which were previously the spread values of the full ENS10 trajectories, by the difference between the ground truth and the output bias corrected forecast ΔP. With our assumption of the forecasts being of a Gaussian distribution, ΔP, the error function Φ and standard deviation σ we then set the CRPS loss as follows (see Appendix A for full derivation):

$$\Delta P = \text{Ground Truth} - \text{Prediction}$$

$$\Phi(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

$$\mathrm{CRPS}(\sigma, \Delta P) = \Delta P \Phi\left(\frac{\Delta P}{\sqrt{2}\sigma}\right) + \frac{\sigma}{\sqrt{\pi}}(-1 + \sqrt{2} e^{-\frac{\Delta P^2}{2\sigma^2}}).$$

Finally, the relative CRPS improvement of a prediction over the original raw ensemble is defined as the Continuous Ranked Probability Skill Score (CRPSS):

$$\mathrm{CRPSS}(\mathrm{CRPS}_{pred}, \mathrm{CRPS}_{orig}) = 1 - \frac{\mathrm{CRPS}_{pred}}{\mathrm{CRPS}_{orig}}.$$

(d) Implementation

Our networks are implemented with the TensorFlow deep learning framework [47]. Layers are initialised with a truncated normal distribution. We train with the Adam optimiser [48] with a learning rate of 0.001 and ℓ_2 regularization. Our models have not been fine-tuned extensively, and there is further potential for improvement. More implementation details can be found in our GitHub repository.

The uncertainty networks are trained for 4,725 update steps with a batch size of 2, requiring about four hours on one Nvidia V100 32 GB GPU. The bias correction networks are trained for the same wall-clock time, taking about 25,000 update steps with a batch size of 2. We use early stopping, i.e., ending the training process once the validation loss stops decreasing, to identify the best parameters. Training can be done once and the resulting networks used until the ensemble prediction system is upgraded. Using the same GPU, inference for one parameter and forecast on a global grid takes approximately 0.31 seconds per network.

4. Results

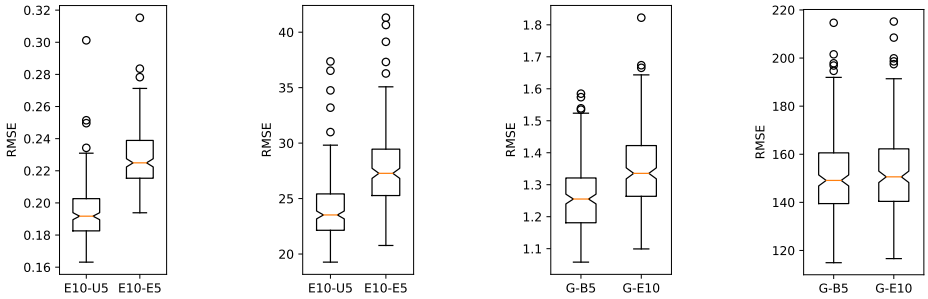
Notation	Description
B{n}	Output bias correction NN trained with n trajectories
U{n}	Spread prediction NN trained with n trajectories
E{n}	Ensemble with n trajectories
Lin{n}	Gridpoint-wise linear regression from n trajectories
C	Uncertainty NN trained on CRPS
G	Ground truth data from ERA5

Table 1: Notation for our model configurations and ground-truth data.

We primarily train our models to predict T850 but also evaluate their prediction capacity on Z500. First, our uncertainty quantification and bias correction networks are evaluated separately on the global RMSE for the spread of ENS10 forecasts or the ERA5 ground-truth respectively. All results are for a forecast lead time of 48 hours. In addition to our DNNs, we train linear regression models on ensemble trajectories as another baseline (see Table 1).

Figures 7 (a, b) show the improvement in spread prediction of our uncertainty network using five ensemble trajectories, compared to simply using the five trajectories. There are significant improvements for both temperature and geopotential. Figures 7 (c, d) show our output bias correction for predicting deviating weather patterns given a forecast mean and no measure of uncertainty. We see improvements for T850, but our network does not provide a strong global improvement for Z500, which we analyze more thoroughly through case studies (see Section 4(a)).

Table 2 shows our analysis of the final results for the uncertainty quantification network using different numbers of trajectories on T850. With a low number of trajectories we can see that our model is already able to predict uncertainty to some degree. We observe diminishing returns in the relative performance as the number of trajectories increases, due to the rapidly growing proportion of trajectories out of the full ensemble that is used. However, the networks still consistently outperform the reduced ensemble baselines. This is even the case for a relatively high number of trajectories (half). We also present the results of an ablation study — exploring the impact of different structures and features — on the output bias correction networks in Table 3. The results show that T850 correction benefits from convolution operations, whereas Z500



(a) T850 (b) Z500 (c) T850 (d) Z500

Figure 7: Notched boxplots for the global RMSE of the Uncertainty Quantification and Output Bias Correction Networks each day of our test set (2016-17). For the x-axis "A − B" $\widehat{=}$ RMSE(A, B).

T850	E3	E4	E5	E6	E7	E8	E9	Lin5	U3	U4	U5
Abs.	0.35	0.28	0.23	0.19	0.15	0.11	0.07	0.21	0.26	0.23	0.19
Rel.	-	-	-	-	-	-	-	8.9%	26.6%	18.7%	16.4%

Table 2: T850 average RMSE towards E10 spread for different ensemble sizes and models for the test set (2016-17). Abs.: Absolute rounded values. Rel.: Relative improvement over the original forecast.

Parameter	Lin10	UN0	UN1	UN2	UN0-LCN	UN1-LCN	UN2-LCN	UN1-LCN-reg
T850	4.8%	6.3%	7.1%	6.7%	7.6%	7.7%	7.6%	**7.9%**
Z500	2.1%	1.2%	1.6%	1.0%	**2.6**%	2.5%	2.4%	2.3%

Table 3: Ablation study of output bias correction, measured by the relative RMSE improvement over the ensemble mean forecast on the test set. UNi: U-Net with i-levels. UNi-LCN: U-Net with i-levels followed by an LCN. *-reg: with ℓ_1 regularization ($\lambda = 10$) on the LCN.

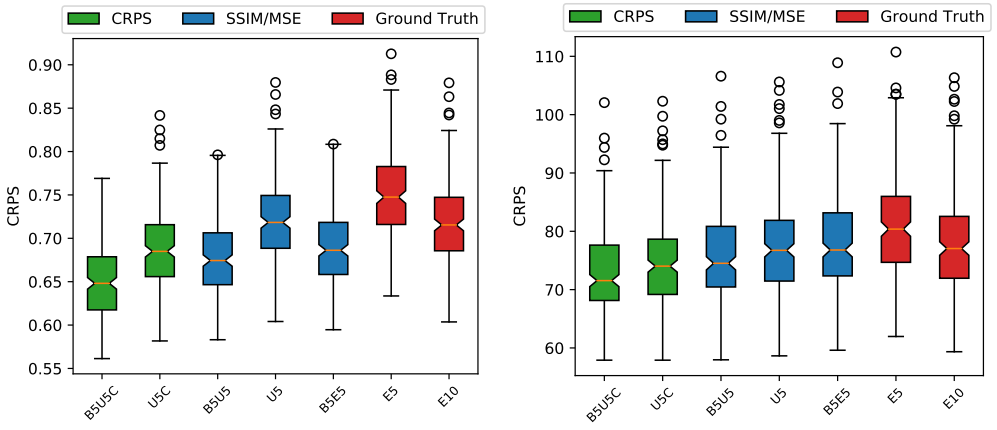
correction benefits most from using a locally connected structure. Further supporting this, we observe that ℓ_1 regularization does not have as significant an impact for Z500, suggesting that local, independent filters are important for prediction.

To understand the contribution of the input fields to the result, we perform another ablation study, training networks with the predicted field as the only input. The results are listed in Table 4, confirming the importance of using multiple input fields to provide more accurate predictions.

Network	Predicted Parameter	Input Fields: Predicted Only	Input Fields: All Fields
Uncertainty Quantification	T850	12.3%	16.4%
	Z500	12.8%	12.9%
Bias Correction	T850	6.90%	7.59%
	Z500	2.06%	2.37%

Table 4: Ablation study of input fields and the resulting relative improvement for the uncertainty quantification and bias correction (UN1-LCN) networks.

RMSE is insufficient to measure probabilistic forecast skill, as it does not encompass both mean and spread. We therefore also consider CRPS (lower scores are better). We use E5 and E10, the raw five- and ten-member ensembles, as our baselines. The results are presented in Figure 8.



(a) T850 (b) Z500

Figure 8: Notched boxplots for the global average CRPS values for each day in our test set (2016-17), for all our networks and raw ensemble combinations.

CRPSS	Z500	T850
B5U5C towards E10	0.0756	0.1098
B5U5C towards E5	0.1074	0.1458

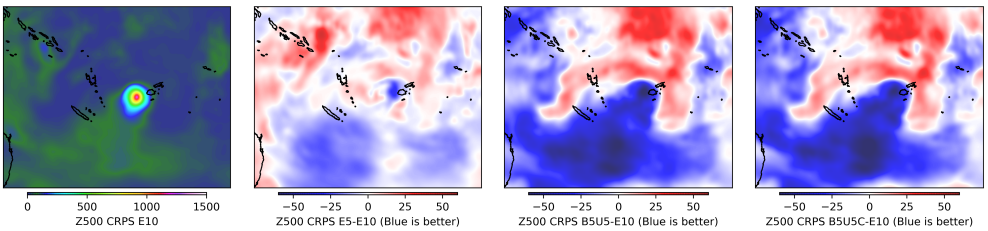
Table 5: Continuous Ranked Probability Skill Scores over our test set (2016-17).

To measure our improvements, we specifically focus on the CRPSS of our PDF calibration network trained on CRPS, as compared to the raw ensemble outputs in Table 5. However, both of the models, trained on CRPS and SSIM, outperform the full ensemble forecast CRPS for both T850 and Z500, despite the smaller number of ensemble members that are used. The major source of improvement for our DNNs, especially for T850, is a reduction in extreme values (outliers), which are indicative of forecast busts. We are probably able to achieve better improvements for T850 when compared to Z500 as the temperature field is likely to show more significant biases.

As another baseline comparison, we run EMOS [27] with the first 5 ensemble members of ENS10 as input. There are 8 parameters in this formulation, and they are trained to minimise CRPS using the ERA5 dataset as the ground truth. We use the same training/validation/test set split and training algorithm (Adam with early stopping) as our DNNs. For T850, we observe a resulting CRPS improvement of 5.5% with EMOS over the test set, compared with 14.5% for B5U5C.

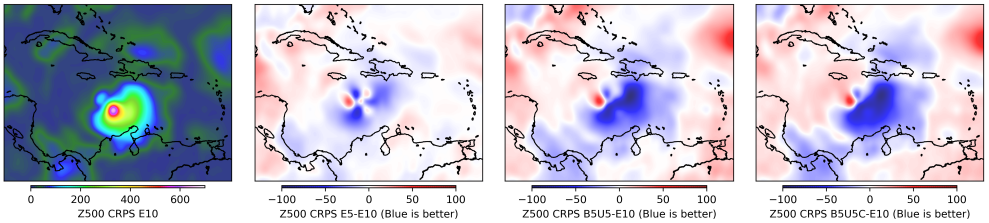
(a) Extreme Weather Forecasts

Thus far, our results have only demonstrated improvements on average values. They do not cover the performance for uncertainty quantification of extreme events, where forecast reliability is essential. In the following, we present three cases of extreme weather phenomena within our test set, selected from across the world, to demonstrate the networks' improvements for specific predictions. However, we would like to warn the reader that the interpretation of the probabilistic scores for specific events is difficult, as improvement or degradation for a single event are not sufficient to conclude that a method is better or worse in general. This limitation is visible in the following examples, as the E5 ensemble is able to outperform the E10 ensemble for specific events and locations. This can be explained by the limited number of ensemble members that have been used, and insufficient sampling of the probability distribution. Furthermore, the interpretation of predictions of extreme events by the end-user needs to be considered as well (see for example Lerch et al. [49]). Figures 9 to 11 show our CRPS improvements for tropical cyclone Winston,



(a) E10 (b) E5-E10 (c) B5U5-E10 (d) B5U5C-E10

Figure 9: Tropical cyclone Winston, raging from February until March 2016 over Fiji, has been classified as the most intense cyclone in the southern hemisphere ever recorded, according to the Southwest Pacific Enhanced Archive for Tropical Cyclones. It reached category five on February 20. We present the prediction for Z500 as forecast on February 19 for February 21, and differences in CRPS. (a) The CRPS for the ten-member ensemble. The centre of the cyclone is clearly visible. (b)-(d) The difference in CRPS between the ten-member ensemble and five-member ensembles with and without post-processing. Our CRPS-trained network shows improvement over E10. It also demonstrates similar confidence in the southern area where the cyclone is moving, while being worse where the cyclone has already passed. This results in a large forecast skill improvement in the selected area, with a CRPSS of 0.261 (26.1% improvement over E10).



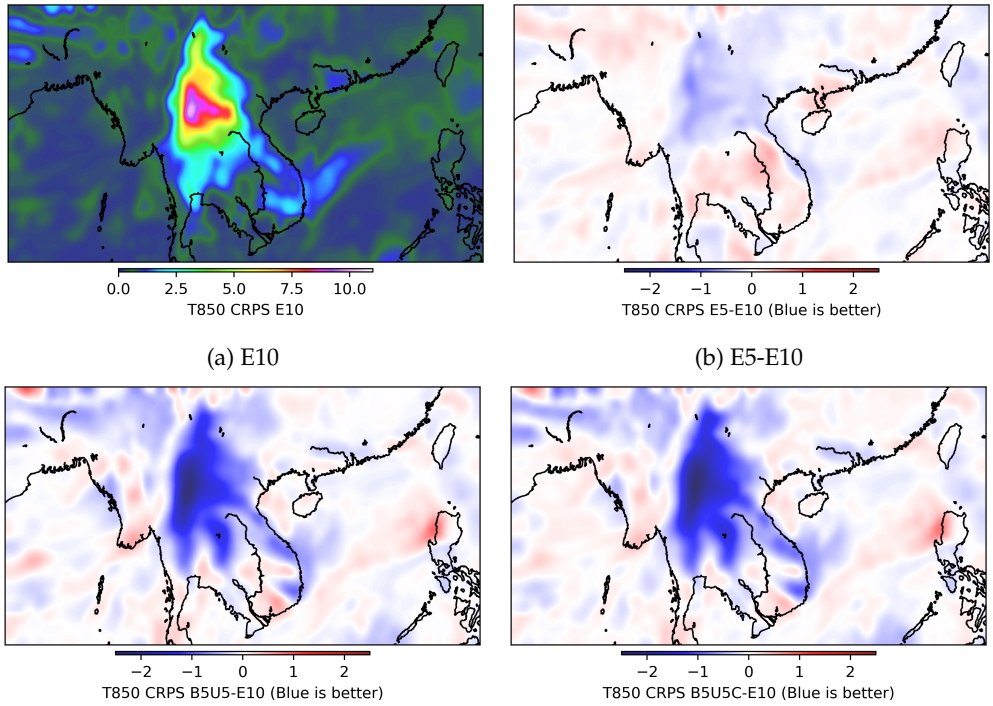
(a) E10 (b) E5-E10 (c) B5U5-E10 (d) B5U5C-E10

Figure 10: Hurricane Matthew, a category five hurricane brought severe destruction to the Carribean and southeastern United States during September and October of 2016. We look at the Z500 forecast for the third of October. Again we see large improvements over the centre of the hurricane, as well as in the northern regions the hurricane will later progress over, with slightly reduced skill for outer regions.

hurricane Matthew, and a cold wave over Asia, respectively. Blue colour indicates that the (post-processed) five member ensemble has more skill when compared to the ten member ensemble. All times are 00:00 UTC. We also present global CRPS plots in Figure 12.

5. Conclusion

We show that using informed model construction, deep learning can indeed improve the skill of global ensemble weather predictions. In particular, since we do not only predict an ensemble spread, but also perform a locally-adaptive output bias correction, we improve the results of a five-member ensemble to even surpass the forecast skill of a ten-member ensemble in terms of CRPS. When tasked with hard-to-predict extreme weather cases, such as tropical cyclone Winston, the combined models exhibit especially pronounced forecast skill improvements. Through the use of heterogeneous hardware, they are able to run these global post-processing steps within tenths of a second. In the future, such deep learning tools could allow for reduced ensembles to be run at higher resolutions, providing cheaper and more informed predictions.



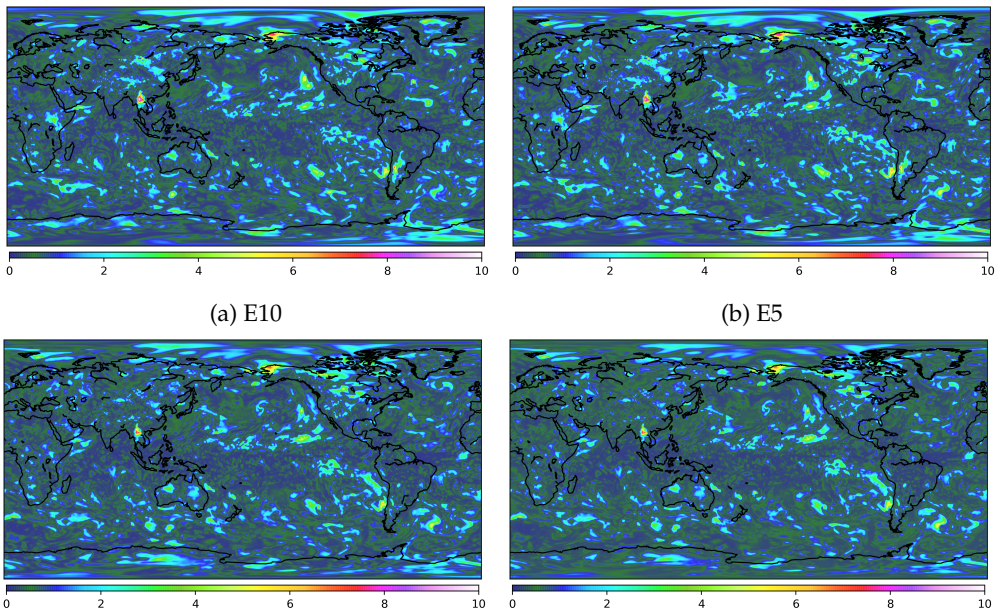
(a) E10 (b) E5-E10

(c) B5U5-E10 (d) B5U5C-E10

Figure 11: Cold Wave over Asia: During January 2016 an unprecedented cold wave rushed over East and South Asia, leading to record lows. We focus on a forecast for January 24 where T850 forecast CRPS has its worst spike. In this case our CRPS trained model brings a large improvement of more than 25.5% (CRPSS), compared to the 10 member ensemble, over the most affected region, while keeping regions of low CRPS fairly close to their original values, resulting in a total forecast improvement of around 19.5% for the selected region.

The network structures used in this paper should also be tested for other applications of deep learning in NWP, such as the learning of model error in data-assimilation systems or the learning of the global equations of motion. Future research could be conducted into whether the networks need to be re-trained to process other physical fields and forecast lead times or whether the normalisation of spread values could allow the same network to also be applied to these tasks through transfer learning. Recurrent networks that encompass more time steps, as well as deep learning models that are capable of working on the native unstructured grid of the prediction model (e.g., graph neural networks [50]), can also be investigated in this context. Furthermore, the presented improvements would need to be studied when applied to ensembles with more members, such as the operational 50-member ensemble system of the ECMWF. Lastly, while the fields that are investigated in this paper (Z500 and T850) are important to explore the potential capabilities of deep learning for this study, other fields that have more local dependence and output three-dimensional representations of the atmosphere, should be investigated in future work.

We encourage researchers to make use of the ERA5 and ENS10 datasets as well as our code, to apply new deep learning methods and expand on our initial architectures, helping weather forecast centres worldwide.



(a) E10 (b) E5

(c) B5U5 (d) B5U5C

Figure 12: Global T850 CRPS plots for our models and the ENS10 forecast for January 24 2016 during the cold wave over East and South Asia (lower values are better).

Acknowledgements

We thank the Swiss National Supercomputing Centre for providing compute resources and technical support. Tal Ben-Nun is supported by the Swiss National Science Foundation (Ambizione Project No. 185778). Nikoli Dryden is supported by the ETH Postdoctoral Fellowship. This project also received funding from the European Research Council (ERC) under the European Union's Horizon 2020 programme (grant agreement DAPP, No. 678880 and EPiGRAM-HS, No. 801039). Peter Dueben gratefully acknowledges funding from the Royal Society for his University Research Fellowship and the ESIWACE2 project. The ESIWACE2 has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 823988.

References

1. European Centre for Medium-Range Weather Forecasts, "The ECMWF ensemble prediction system," https://www.ecmwf.int/sites/default/files/the_ECMWF_Ensemble_prediction_system.pdf, 2012.
2. M. Leutbecher and Z. B. Bouallègue, "On the probabilistic skill of dual-resolution ensemble forecasts," 2019.
3. T. Schulthess *et al.*, "Reflecting on the goal and baseline for exascale computing: a roadmap based on weather and climate simulations," *Computing in Science and Engineering (CiSE)*, vol. 21, no. 1, Jan. 2019.
4. P. Neumann *et al.*, "Assessing the scales in numerical weather and climate predictions: will exascale be the rescue?" *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 377, no. 2142, 2019. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2018.0148
5. C. Schär *et al.*, "Kilometer-scale climate models: Prospects and challenges," *Bulletin of the American Meteorological Society*, vol. 100, no. 12, Dec. 2019, early Online Release.

6. P. Dueben, N. Wedi, S. Saarinen, and C. Zeman, "Global simulations of the atmosphere at 1.45 km grid-spacing with the integrated forecasting system," *Journal of the Meteorological Society of Japan. Ser. II*, vol. advpub, 2020.
7. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, 2015.
8. N. D. Brenowitz and C. S. Bretherton, "Prognostic validation of a neural network unified physics parameterization," *Geophysical Research Letters*, vol. 45, no. 12, 2018. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018GL078510
9. S. Rasp and S. Lerch, "Neural Networks for Postprocessing Ensemble Weather Forecasts," *Monthly Weather Review*, vol. 146, no. 11, Nov 2018.
10. S. Rasp, M. S. Pritchard, and P. Gentine, "Deep learning to represent sub-grid processes in climate models," 2018.
11. P. D. Dueben and P. Bauer, "Challenges and design choices for global weather and climate models based on machine learning," *Geoscientific Model Development*, vol. 11, no. 10, 2018. [Online]. Available: https://www.geosci-model-dev.net/11/3999/2018/
12. S. Rasp *et al.*, "WeatherBench: A benchmark dataset for data-driven weather forecasting," *arXiv preprint arXiv:2002.00469*, 2020.
13. J. Pathak *et al.*, "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach," 2018.
14. P. R. Vlachas *et al.*, "Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics," 2019.
15. Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," in *Shape, Contour and Grouping in Computer Vision*, 1999. [Online]. Available: https://doi.org/10.1007/3-540-46805-6_19
16. A. Coates *et al.*, "Deep learning with COTS HPC systems," in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ser. ICML'13, 2013, p. III–1337–III–1345.
17. F. Chevallier, F. Chéruy, N. A. Scott, and A. Chédin, "A neural network approach for a fast and accurate computation of a longwave radiative budget," *Journal of Applied Meteorology*, vol. 37, no. 11, 1998.
18. S. Xingjian *et al.*, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015.
19. X. Shi *et al.*, "Deep learning for precipitation nowcasting: A benchmark and a new model," in *Advances in neural information processing systems*, 2017.
20. A. Heye, K. Venkatesan, and J. Cain, "Precipitation nowcasting: Leveraging deep recurrent convolutional neural networks," *Proceedings of the Cray User Group (CUG)*, 2017.
21. S. Agrawal *et al.*, "Machine learning for precipitation nowcasting from radar images," 2019.
22. T. Kurth *et al.*, "Exascale deep learning for climate analytics," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis*, 2018.
23. P. R. Larraondo, L. J. Renzullo, I. Inza, and J. A. Lozano, "A data-driven approach to precipitation parameterizations using convolutional encoder-decoder neural networks," 2019.
24. J. A. Weyn, D. R. Durran, and R. Caruana, "Can machines learn to predict weather? using deep learning to predict gridded 500-hpa geopotential height from historical weather data," *Journal of Advances in Modeling Earth Systems*, vol. 11, no. 8, 2019. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019MS001705
25. R. Lagerquist, A. McGovern, and D. J. Gagne II, "Deep learning for spatially explicit prediction of synoptic-scale fronts," *Weather and Forecasting*, vol. 34, no. 4, 2019. [Online]. Available: https://doi.org/10.1175/WAF-D-18-0183.1
26. Y. Liu *et al.*, "Application of deep convolutional neural networks for detecting extreme weather in climate datasets," 2016.
27. T. Gneiting, A. E. Raftery, A. H. Westveld, and T. Goldman, "Calibrated probabilistic forecasting using ensemble model output statistics and minimum crps estimation," *Monthly Weather Review*, vol. 133, no. 5, 2005. [Online]. Available: https://doi.org/10.1175/MWR2904.1
28. A. E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski, "Using bayesian model averaging to calibrate forecast ensembles," *Monthly Weather Review*, vol. 133, no. 5, 2005. [Online]. Available: https://doi.org/10.1175/MWR2906.1
29. T. M. Hamill and J. S. Whitaker, "Ensemble calibration of 500-hpa geopotential height and 850-hpa and 2-m temperatures using reforecasts," *Monthly Weather Review*, vol. 135, no. 9, 2007. [Online]. Available: https://doi.org/10.1175/MWR3468.1

30. A. Baran, S. Lerch, M. E. Ayari, and S. Baran, "Machine learning for total cloud cover prediction," *arXiv:2001.05948*, 2020.
31. P. Grönquist *et al.*, "Predicting weather uncertainty with deep convnets," *arXiv:1911.00630*, 2019.
32. T. M. Hamill, J. S. Whitaker, and S. L. Mullen, "Reforecasts: An important dataset for improving weather predictions," *Bulletin of the American Meteorological Society*, vol. 87, no. 1, 2006. [Online]. Available: https://doi.org/10.1175/BAMS-87-1-33
33. F. Vitart, "Evolution of ECMWF sub-seasonal forecast skill scores," *Q.J.R. Meteorol. Soc., 140: 1889-1899*, 2014.
34. F. Vitart *et al.*, "Extended-range prediction," *ECMWF Technical Memorandum 854, 58 pp*, 2019.
35. N. P. Wedi, "Increasing horizontal resolution in numerical weather prediction and climate simulations: illusion or panacea?" *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 372, no. 2018, p. 20130289, 2014. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2013.0289
36. European Centre for Medium-Range Weather Forecasts, "ERA5," https://www.ecmwf.int/en/forecasts/datasets/reanalysis-datasets/era5, 2019.
37. S. Scher and G. Messori, "How global warming changes the difficulty of synoptic weather forecasting," *Geophysical Research Letters*, vol. 46, no. 5, 2019. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018GL081856
38. World Meteorological Organization, "FM 92 GRIB," https://www.wmo.int/pages/prog/www/DPS/FM92-GRIB2-11-2003.pdf, 2003.
39. C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
40. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597
41. Ö. Çiçek *et al.*, "3D U-Net: learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*, 2016.
42. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
43. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Advances in neural information processing systems*, 2018.
44. A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, 2016.
45. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, 2004.
46. H. Hersbach, "Decomposition of the continuous ranked probability score for ensemble prediction systems," *Weather and Forecasting*, vol. 15, no. 5, 2000. [Online]. Available: https://doi.org/10.1175/1520-0434(2000)015<0559:DOTCRP>2.0.CO;2
47. M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/
48. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
49. S. Lerch, T. L. Thorarinsdottir, F. Ravazzolo, and T. Gneiting, "Forecaster's dilemma: Extreme events and forecast evaluation," *Statist. Sci.*, vol. 32, no. 1, pp. 106–127, 02 2017. [Online]. Available: https://doi.org/10.1214/16-STS588
50. F. Scarselli *et al.*, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, 2009.

A. CRPS derivation

The Cumulative Distribution Function (CDF) of a Normal Distribution can be written as:

$$F(x) = \frac{1}{2}\left(1 + \Phi\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right)$$

$$\Phi(x) := \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,dt\,,$$

with Φ being the error function. The CRPS is then defined as:

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{\infty} [F(x) - \mathbf{1}_{x>y}]^2 dx\,, \tag{A 1}$$

which can be written as:

$$\int_{-\infty}^{y} \left(\frac{1}{2}\left(1 + \Phi\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right)\right)^2 dx \tag{A 2}$$

$$+ \int_{y}^{\infty} \left(\frac{1}{2}\left(1 + \Phi\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right) - 1\right)^2 dx\,. \tag{A 3}$$

Continuing, we get:

$$\left[\frac{1}{2}\left(\left(\sqrt{\frac{2}{\pi}}\sigma e^{-\frac{(\mu-x)^2}{2\sigma^2}} + x - \mu\right)\Phi\left(\frac{x-\mu}{\sqrt{2}\sigma}\right) + \frac{1}{2}(x-\mu)\,\Phi\left(\frac{x-\mu}{\sqrt{2}\sigma}\right)^2 + \frac{\sigma\Phi\left(\frac{\mu-x}{\sigma}\right)}{\sqrt{\pi}} + \sqrt{\frac{2}{\pi}}\sigma e^{-\frac{(\mu-x)^2}{2\sigma^2}} + \frac{x}{2}\right)\right]_{-\infty}^{y} \tag{A 4}$$

from (A 2) and

$$\left[\frac{1}{2}\left(\left(\sqrt{\frac{2}{\pi}}\sigma e^{-\frac{(\mu-x)^2}{2\sigma^2}} + \mu - x\right)\Phi\left(\frac{x-\mu}{\sqrt{2}\sigma}\right) + \frac{1}{2}(x-\mu)\,\Phi\left(\frac{x-\mu}{\sqrt{2}\sigma}\right)^2 + \frac{\sigma\Phi\left(\frac{\mu-x}{\sigma}\right)}{\sqrt{\pi}} - \sqrt{\frac{2}{\pi}}\sigma e^{-\frac{(\mu-x)^2}{2\sigma^2}} + \frac{x}{2}\right)\right]_{y}^{\infty} \tag{A 5}$$

from (A 3). If we now place in the bounds and sum (A 4) and (A 5) up we arrive at:

$$(y-\mu)\,\Phi\left(\frac{y-\mu}{\sqrt{2}\sigma}\right) + \sqrt{\frac{2}{\pi}}\sigma e^{-\frac{(\mu-x)^2}{2\sigma^2}} \tag{A 6}$$

from inserting the y bounds, and

$$-\frac{\sigma}{\sqrt{\pi}} \tag{A 7}$$

from inserting the ∞ bounds, resulting in:

$$\Delta P = y - \mu = \mathrm{Ground\_Truth} - \mathrm{Prediction} \tag{A 8}$$

$$\mathrm{CRPS}(\sigma, \Delta P) = \Delta P \cdot \Phi\left(\frac{\Delta P}{\sqrt{2}\sigma}\right) + \frac{\sigma}{\sqrt{\pi}}(-1 + \sqrt{2}e^{-\frac{\Delta P^2}{2\sigma^2}})\,. \tag{A 9}$$