Predicting Weather Uncertainty with Deep Convnets

Peter Grönquist
ETH Zürich
petergro@student.ethz.ch

Tal Ben-Nun
ETH Zürich
tal.bennun@inf.ethz.ch

Nikoli Dryden
ETH Zürich
nikoli.dryden@inf.ethz.ch

Peter Dueben
ECMWF
peter.dueben@ecmwf.int

Luca Lavarini
ETH Zürich
lucalav@student.ethz.ch

Shigang Li
ETH Zürich
shigang.li@inf.ethz.ch

Torsten Hoefler
ETH Zürich
htor@inf.ethz.ch

Abstract

Modern weather forecast models perform uncertainty quantification using ensemble prediction systems, which collect nonparametric statistics based on multiple perturbed simulations. To provide accurate estimation, dozens of such computationally intensive simulations must be run. We show that deep neural networks can be used on a small set of numerical weather simulations to estimate the spread of a weather forecast, significantly reducing computational cost. To train the system, we both modify the 3D U-Net architecture and explore models that incorporate temporal data. Our models serve as a starting point to improve uncertainty quantification in current real-time weather forecasting systems, which is vital for predicting extreme events.

1 Introduction

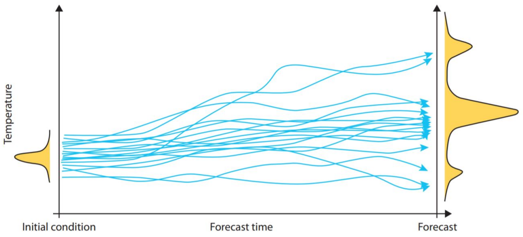


Figure 1: ECMWF Ensemble Prediction System.

Weather forecasting is of critical importance for many areas of life, from civil safety to food security. Uncertainty quantification of weather forecasts is essential due to the chaotic dynamics of weather and the exponential growth of forecast errors in time. When a tropical cyclone approaches a coastline, it is important to know both the most likely position of landfall and the probability that it will hit other locations. Thus, numerical weather predictions (NWP) need to provide reliable probability distributions for their predictions. To this end, weather centers, such as the European Centre for Medium-Range Weather Forecasts (ECMWF), typically run an ensemble of unique weather forecasts (*trajectories*) in parallel, to estimate a probability distribution for several parameters, e.g. temperature [1, 2] (Figure 1). Each ensemble member is perturbed, and the difference in the resulting predictions, measured by their standard deviation (*spread*), can be used to identify the uncertainty of a high-resolution forecast. Modeling such distributions is important for predictions of extreme weather events, and requires as many as fifty ensemble members [3]. Running these ensemble simulations with many members at a global scale dramatically increases the computational demands of weather forecast models.

Second Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019), Vancouver, Canada.

In an effort to speed up parts of NWP, recent works apply machine learning techniques to weather and climate models [4]. Most works revolve around augmenting physical models for improved accuracy [5–10], or for short-term predictions such as precipitation nowcasting [11–13]. However, there has been relatively little work on using machine learning for uncertainty quantification of multi-parameter forecasts and global data. Previous works focus instead on producing a single scalar value [14] or station-specific information [15] on local regions.

In this paper, we present preliminary work exploring 3D CNN architectures for predicting the spread of ensemble forecasts using as few as one simulation with our initial parameters. Our forecasts come from the ECMWF ensemble forecasting system, which is based on the Integrated Forecasting System (IFS) [16]. We consider several novel architectures for exploiting spatial and temporal effects, such as affine convolutions, as well as study existing architectures, such as CNN-LSTMs [11]. Once trained, our models are able to approximate the mean and spread of a large ensemble of models accurately and with low computational cost.

2 Data

For our preliminary exploration, we use the ERA5 dataset [17], as it is similar to the data used by production NWP and is publicly available. The dataset is produced by the ECMWF, and currently consists of weather data reanalysis from 1979 to the present. It includes an ensemble of nine perturbed trajectories and a single unperturbed (control) trajectory, of which we explore different subsets. The available data was mapped to a latitude and longitude grid with a 0.5 degree resolution and contains 37 pressure levels. We use temperature prediction as an initial target. Based on previous exploratory work [18], we select a subset of Initial Parameters (IP) that have an influence on temperature: zonal and meridional wind, geopotential, temperature, relative humidity, and the fraction of cloud cover.

We use a subset of the data that includes Europe and parts of the Atlantic Ocean (Figure 2), with a window of 40 latitude by 136 longitude points. We choose seven pressure levels, including 500 hPa (which is in the middle of the atmosphere) and 850 hPa (close to the surface), which can be used to identify warm and cold fronts due to limited daily temperature variations. For the temporal dimension, we use measurements from 0600 and 1800 UTC, with forecasts made for three ($t = 3$h) and six ($t = 6$h) hours into the future.

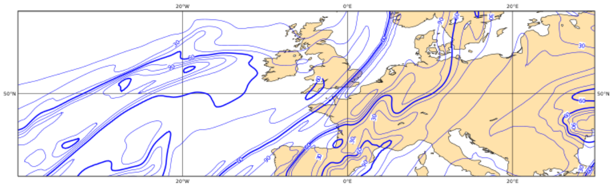


Figure 2: Example of relative humidity in our selected area, at 150 hPa from longitudes -40° to 30° and latitudes 40° to 60°, plotted using ECMWF's Magics library [19].

Our subset of the ERA5 data is available in the GRIB format [20] for the years 2000-2011, from which we extract our region of interest and standardize the data for each pressure level and parameter. We save the data as single-precision floats (due to TensorFlow limitations) in chronological order and in the correct spatial distribution in NumPy arrays [21]. We then convert the data to the TFRecord [22] format with one year per file. We use the most recent years (2010, 2011) as a test set; of the remaining data, we randomly select 80% for training and 20% for validation. We shuffle data to ensure that training and validation datasets are drawn from a variety of times and that seasons are not observed in sequence.

We also consider a second dataset that we call ENS10. ENS10 is based on ECMWF re-forecasts that perform 10-member ensemble predictions two times per week for the last twenty years [23]. All trajectories are perturbed. Re-forecasts are very similar to the operational 51-member ensemble used by the ECMWF but run at coarser resolution of 0.5 degrees. This allows us to gain additional insight into what our model's expected performance on the 51-member ensemble would be, as this data is not available to us.

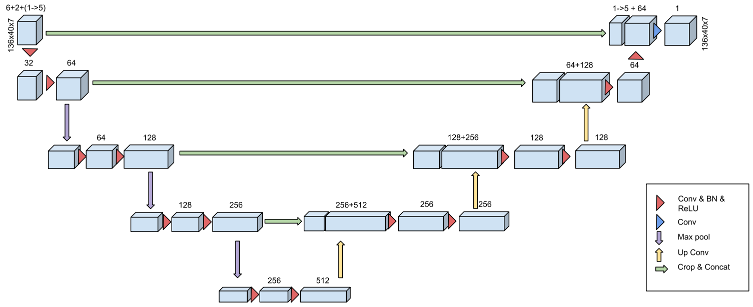


Figure 3: Our baseline model, adapted from 3D U-Net [24].

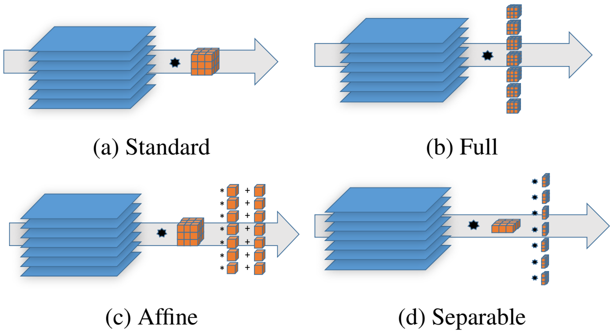
3 Model and Methods

To comprehensively explore this problem, we consider three aspects: integrating spatial effects from multiple pressure levels and resolutions, predicting temperature for all pressure levels at once; learning from temporal trends by including previous forecasting timesteps, predicting temperature one pressure level at a time; and efficient implementations.

Our baseline model is adapted from the 3D U-Net [24] DNN, which contains residual connections to preserve spatial information (Figure 3). We concatenate all parameters and trajectories channel-wise to form each input sample. We also considered a DeepLab v3+ model [25] with a ResNet-50 backbone [26], but observed that it performed 9.6% worse than the results we report here for our baseline model on temporal data.

Due to the chaotic nature of weather, small perturbations have a large influence on weather patterns, especially for the long-term forecasts we are trying to predict. We speculate that it is harder to extract information from these smaller spatial patterns in our data when they are downsampled through convolutions and max pooling compared to natural images. This limits the benefit of deeper networks.

Spatial effects. It is challenging to apply 3D convolution directly to our data, because grid points are not uniformly spaced. This is most evident in the pressure levels, where distances increase exponentially with respect to geopotential height. For large maps, Gaussian grids also hamper the use of simple convolution windows. We consider several approaches to address this, as depicted in Figure 4. Weight sharing on each pressure level separately ("Full") enables more flexibility to learn different representations, but significantly increases the number of parameters. More conservatively, we can introduce learned point-wise affine transforms per pressure level after each convolution ("Affine"). As a compromise between parameters and representational power, we also consider horizontal 2D convolutions followed by vertical 1D convolutions ("Separable").


(a) Standard

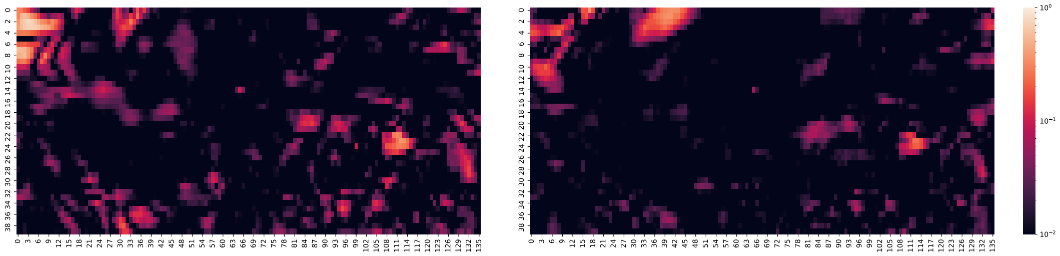
(b) Full

(c) Affine

(d) Separable

Figure 4: Different convolutions for exploring spatial effects.

Temporal effects. To explore temporal trends, we look at the data of the spread of all ten trajectories at times $t = 0$h, $t = 3$h and $t = 6$h. Our goal is to see if our models can learn the temporal progression of the spread. As a first step, we only use the spread at $t = 0$h and $t = 3$h as input. Subsequently we also explore the additional concatenation of Initial Parameters (IP) from the unperturbed trajectory. Our basic method is inspired by the work on precipitation nowcasting [12]. We also evaluate CNN-LSTMs [11] as an efficient method that also retains spatial information. However, we find that with our limited time slices, simply treating time sequences as additional channels in U-Net and



(a) Linear regression from spread at $t = 0$h.
RMSE = 0.1346

(b) Baseline model prediction.
RMSE = 0.1060

Figure 5: Logarithmic heatmap of squared difference towards full ensemble spread at $t = 6$h (850 hPa, temperature in K). Axes are our selected longitude and latitude.

ResNet structures offers the best performance. Fully understanding the impact of temporal data on our predictions requires additional study with more timesteps.

Distributed training. To efficiently train on the global high-resolution data, as necessary in production NWP, it is crucial to employ distributed training. In particular, the memory requirements of training on such data can easily exceed GPU memory. We currently leverage distributed data-parallelism and propose to further mitigate this problem with a combination of pipeline parallelism for network depth [27, 28] and model parallelism for high-resolution data [29, 30].

4 Evaluation

We compare the initial results of our models using the root mean squared error (RMSE) as the optimization target, showing the final test values in Figure 7. As an algorithmic comparison, we use a linear regression on the full ensemble spread at time $t = 0$h, which has about the same error as the spread of three trajectories. Our baseline model, using only one unperturbed trajectory, provides a better spread estimation than using four perturbed trajectories. Additional changes to our model that include spatial effects put us on equal footing with five trajectories (half of what is available in ERA5). We further improve upon this by incorporating temporal data and predicting $t = 6$h.

We also evaluate the impact of concatenating our Initial Parameters (IP) to the input data in Figure 7. While it does show an improvement, it is minor compared to the impact of model architecture.

As the RMSE does not incorporate spatial coherency, we visualize the predictions of our baseline model to better understand them (Figure 5). We observe that the model places greater importance on larger shifting spread regions (upper left corner) while neglecting some of the lower spread regions. This shows promise for detecting extreme weather events.

Finally, we evaluate our baseline model on the ENS10 dataset on a global scale and observe similar promising results (Figure 6). Our model successfully approximates larger ensembles using only a few input trajectories. We observe diminishing returns (MSE towards the 10-member ensemble spread) as we increase the number of trajectories used. We theorize that this is due to the small number of ensemble members in the dataset: Five trajectories is half of the ensemble. Additionally, we do not observe any benefit from our temporal models. We hypothesize this is because the temporal grid in ENS10 (24 hour intervals) is coarser than in ERA5 (3 hour intervals), generating weather pattern changes that are significantly harder to predict.

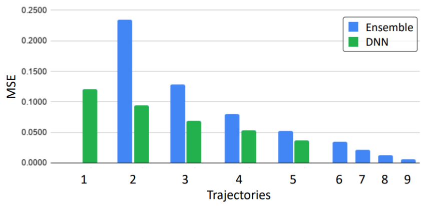


Figure 6: ENS10 final validation loss (baseline model, global).

It is too early to provide a comprehensive cost/benefit comparison between conventional ensemble predictions and those post-processed with deep learning. However, current NWP models use significant time on state-of-the-art supercomputers. ECMWF's 51-member ensembles run twice per day for one hour on a Cray XC40 supercomputer with more than 4 PetaFLOPs peak performance. These

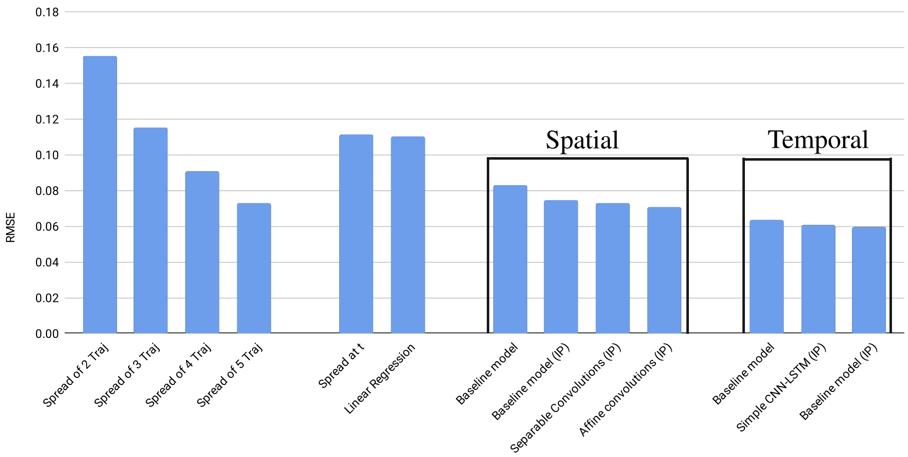


Figure 7: ERA5 test RMSE compared to the full ensemble spread.

calculations use and predict roughly ten times more pressure levels and diverse parameters for 15-day forecasts. In comparison, for our models to predict seven pressure levels for one parameter and 2-day forecasts takes about 15 ms on one Nvidia V100 GPU, which would be performed after the trajectory calculations. Thus, even when scaling up to full-resolution production predictions, our models will be considerably faster than running an ensemble trajectory. Further, there remain many optimization opportunities in the training and inference pipelines.

5 Discussion

We have demonstrated promising preliminary results on using CNNs to approximate ensembles for NWP with significantly reduced computational requirements. This was possible due to the use of techniques such as affine convolution to incorporate spatial information and CNN-LSTMs to incorporate temporal information. This serves as an important first step toward integrating deep learning into production NWP pipelines and improving real-time weather forecasting. By using our method to reduce the number of ensemble members, we can both reduce the compute requirements and improve the forecast product for simulations that are constrained by the time required to run current NWP models. In the future, we plan to evaluate these models on larger and higher-resolution datasets containing global information. This introduces additional challenges, such as nonuniform grid points and higher computational and memory costs for training.

References

[1] T. N. Palmer, "Predicting uncertainty in forecasts of weather and climate," *Reports on progress in Physics*, vol. 63, no. 2, p. 71, 2000.

[2] 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, accessed: 2019-09-14.

[3] M. Leutbecher, "Ensemble size: How suboptimal is less than infinity?" *Quarterly Journal of the Royal Meteorological Society*, 2018.

[4] D. Rolnick *et al.*, "Tackling climate change with machine learning," *arXiv preprint arXiv:1906.05433*, 2019.

[5] A. McGovern *et al.*, "Using artificial intelligence to improve real-time decision-making for high-impact weather," *Bulletin of the American Meteorological Society*, vol. 98, no. 10, pp. 2073–2090, 2017.

[6] 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.

[7] S. Scher, "Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning," *Geophysical Research Letters*, vol. 45, no. 22, 2018.

[8] M. Mudigonda *et al.*, "Segmenting and tracking extreme climate events using neural networks," in *Deep Learning for Physical Sciences Workshop@NeurIPS*, 2017.

[9] E. Racah, C. Beckham, T. Maharaj, S. E. Kahou, M. Prabhat, and C. Pal, "ExtremeWeather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events," in *Advances in Neural Information Processing Systems*, 2017.

[10] 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.

[11] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015.

[12] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Deep learning for precipitation nowcasting: A benchmark and a new model," in *Advances in neural information processing systems*, 2017.

[13] A. Heye, K. Venkatesan, and J. Cain, "Precipitation nowcasting: Leveraging deep recurrent convolutional neural networks," *Proceedings of the Cray User Group (CUG)*, 2017.

[14] S. Scher and G. Messori, "Predicting weather forecast uncertainty with machine learning," *Quarterly Journal of the Royal Meteorological Society*, vol. 144, no. 717, 2018.

[15] S. Rasp and S. Lerch, "Neural Networks for Postprocessing Ensemble Weather Forecasts," *Monthly Weather Review*, vol. 146, no. 11, pp. 3885–3900, Nov 2018.

[16] European Centre for Medium-Range Weather Forecasts, "Modeling and prediction," https://www.ecmwf.int/en/research/modelling-and-prediction, 2019, accessed: 2019-09-14.

[17] ——, "ERA5," https://www.ecmwf.int/en/forecasts/datasets/reanalysis-datasets/era5, 2019, accessed: 2019-09-14.

[18] H.-L. Merenti-Välimäki and P. Laininen, "Analysing effects of meteorological variables on weather codes by logistic regression," *Meteorological Applications*, vol. 9, no. 2, 2002.

[19] European Centre for Medium-Range Weather Forecasts, "Magics," https://confluence.ecmwf.int/display/MAGP/Magics, 2019, accessed: 2019-09-14.

[20] World Meteorological Organization, "FM 92 GRIB," https://www.wmo.int/pages/prog/www/DPS/FM92-GRIB2-11-2003.pdf, 2003, accessed: 2019-09-14.

[21] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, 2011.

[22] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation*, 2016.

[23] European Centre for Medium-Range Weather Forecasts, "ENS10," https://www.ecmwf.int/en/forecasts/documentation-and-support/extended-range/re-forecast-medium-and-extended-forecast-range, 2019, accessed: 2019-09-14.

[24] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*, 2016.

[25] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision*, 2018.

[26] 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.

[27] Y. Huang, Y. Cheng, D. Chen, H. Lee, J. Ngiam, Q. V. Le, and Z. Chen, “GPipe: Efficient training of giant neural networks using pipeline parallelism,” *arXiv preprint arXiv:1811.06965*, 2018.

[28] Y. Li, M. Yu, S. Li, S. Avestimehr, N. S. Kim, and A. Schwing, “Pipe-SGD: A decentralized pipelined SGD framework for distributed deep net training,” in *Advances in Neural Information Processing Systems*, 2018.

[29] N. Dryden, N. Maruyama, T. Benson, T. Moon, M. Snir, and B. V. Essen, “Improving strong-scaling of CNN training by exploiting finer-grained parallelism,” in *International Parallel and Distributed Processing Symposium*, 2019.

[30] N. Dryden, N. Maruyama, T. Moon, T. Benson, M. Snir, and B. V. Essen, “Channel and filter parallelism for large-scale CNN training,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis*, 2019, to appear.