Improving Wind Forecasts in the Lower Stratosphere by Distilling an Analog Ensemble into a Deep Neural Network

Salvatore Candido[1], Aakanksha Singh[1], and Luca Delle Monache[2]

[1]Loon
[2]University of California San Diego

November 22, 2022

Abstract

We discuss improving forecasts of winds in the lower stratosphere using machine learning to post-process the output of the European Centre for Medium-Range Weather Forecasts (ECMWF) Integrated Forecast System. We post-process global three-dimensional predictions, and demonstrate distilling the analog ensemble (AnEn) method into a deep neural network which reduces post-processing latency to near zero maintaining increased forecast skill. This approach reduces the error with respect to ECMWF high-resolution deterministic prediction between 2-15% for wind speed and 15-25% for direction, and is on par with ECMWF ensemble (ENS) forecast skill to hour 60. Verifying with Loon data from stratospheric balloons, AnEn has 20% lower error than ENS for wind speed and 15% for wind direction, despite significantly lower real-time computational cost to ENS. Similar performance patterns are reported for probabilistic predictions, with larger improvements of AnEn with respect to ENS. We also demonstrate that AnEn generates a calibrated probabilistic forecast.

Improving Wind Forecasts in the Lower Stratosphere by Distilling an Analog Ensemble into a Deep Neural Network

Salvatore Candido[1], Aakanksha Singh[1], and Luca Delle Monache[1,2]

[1]Loon, Mountain View, California, USA
[2]Center for Western Weather and Water Extremes, Scripps Institution of Oceanography,University of California San Diego, La Jolla, California, USA

Key Points:

- An analog ensemble generates accurate predictions of lower-stratosphere winds and reliably quantifies the prediction uncertainty.
- A cloud-based distributed computing implementation builds global three-dimensional predictions in tens of minutes.
- Distilling the analog ensemble into a deep neural network allows scaling historical forecasts without slowing post-processing speed.

Corresponding author: Salvatore Candido, scandido@loon.com

Abstract

We discuss improving forecasts of winds in the lower stratosphere using machine learning to post-process the output of the European Centre for Medium-Range Weather Forecasts (ECMWF) Integrated Forecast System. We post-process global three-dimensional predictions, and demonstrate distilling the analog ensemble (AnEn) method into a deep neural network which reduces post-processing latency to near zero maintaining increased forecast skill. This approach reduces the error with respect to ECMWF high-resolution deterministic prediction between 2-15% for wind speed and 15-25% for direction, and is on par with ECMWF ensemble (ENS) forecast skill to hour 60. Verifying with Loon data from stratospheric balloons, AnEn has 20% lower error than ENS for wind speed and 15% for wind direction, despite significantly lower real-time computational cost to ENS. Similar performance patterns are reported for probabilistic predictions, with larger improvements of AnEn with respect to ENS. We also demonstrate that AnEn generates a calibrated probabilistic forecast.

Plain Language Summary

We demonstrate improvements in predicting winds in the stratosphere using machine learning. Our approach uses predictions and analyses from the European Centre for Medium-Range Weather Forecasts (ECMWF). By comparing how previous forecasts differed from what the winds ultimately were over many data points, we are able to modify the current forecast in a way that improves prediction of the winds observed by Loon high altitude balloons in the stratosphere.

A common barrier to using approaches like this to generate global predictions is processing a large amount of information quickly enough to be useful. We demonstrate that by using machine learning we are able to perform many of the slow calculations ahead of time, and that these forecast improvements can be deployed in real applications.

1 Introduction

This paper discusses forecasting stratospheric winds by post-processing numerical weather prediction models using machine learning techniques. Specifically, a new variant of the analog ensemble (AnEn; Delle Monache et al., 2013) algorithm that heavily leverages deep neural networks is proposed. The methodology is tested against analysis data and a dataset of observations of stratospheric winds from Loon (`http://www.loon.com`) high altitude balloons (Candido, 2020).

Our focus on winds in the lower stratosphere is driven by Loon's need to predict the trajectory of high altitude balloons drifting through the stratosphere. Loon is a company that provides connectivity to people in underserved (often remote and rural) locations by placing telecommunications on these balloons. These high altitude platforms can change altitude, and are navigated using a machine learning approach to synthesize in situ observations of winds (from the balloon movements) and wind forecasts. Improved forecast accuracy and reliable uncertainty quantification of the forecasts, which are both key results of the approach we present, determine the navigation efficiency of balloons. Because this navigation system is a real-time operational system (that has navigated balloons for over 1 million hours of flight through the stratosphere), the amount of data to be downloaded from operational forecast centers, the compute needed to utilize that data in real-time operations, and the length of time required to do the post-processing, are also important factors that drive the quality of the system's operation. These concerns led to the development of a less expensive model in post-processing and distilling the computational burden of the post-processing process into a neural network. It is expected that the approach proposed here to allow the real-time execution of postprocessing methods as the analog ensemble across millions of grid points and several lead times for global predictions, can be applied to several other atmospheric variables and parameters. (See below for the range of applications for which the analog ensemble method has already been implemented.)

Recently, with the availability of increased computation resources suitable for the execution of neural networks (e.g., on graphics processing units) and access to large training data sets, machine learning algorithms have been successfully explored to generate weather predictions and to postprocess numerical weather predictions (e.g., Tao et al., 2016; Gagne II et al., 2017; Rasp & Lerch, 2018; Scher, 2018; Chapman et al., 2019; Lagerquist et al., 2019; Burke et al., 2020). It has also been shown that machine learning can support the decision-making process associated with high-impact weather phenomena (McGovern et al., 2017) and it can be leveraged to enhance our physical understanding of atmospheric processes (Gagne II et al., 2019; McGovern et al., 2019).

Analog-based methods, which are a type of machine learning, have been explored for decades (Lorenz, 1969) to develop predictions for a range of weather parameters. The basic idea is to find situations from the past similar to the current one and use what unfolded in these situations to estimate the future evolution of a parameter (Klausner et al., 2009; Panziera et al., 2011) or to infer the errors of today's prediction from a dynamical model's past performance (Delle Monache et al., 2013), an ensemble of model runs (Hamill & Whitaker, 2006), or other methods (Mahoney et al., 2012; Cervone et al., 2017).

One of the challenges of finding these similar situations is the size of the historical dataset available to the algorithm. Van den Dool (1994) estimated that when matching fields over large spatial domains (e.g., the northern hemisphere) a training dataset 10^{30} years long would be needed to find matches with a degree of analogy below observational errors. However, Van den Dool (1994) also indicated that if the matching problem can be reduced to a few degrees of freedom, a much shorter historical dataset can be sufficient.

We apply one such approach, the AnEn (Delle Monache et al., 2011, 2013), to the prediction of lower-stratosphere winds. In our case, matching to analogous situations is performed independently at each grid location and lead time over two parameters: wind speed and direction. Forecast improvements are demonstrated with only two years of previous forecasts. Versions of the AnEn have been applied successfully for the prediction of weather parameters (Delle Monache et al., 2013; Nagarajan et al., 2015; Eckel & Delle Monache, 2016; Frediani et al., 2017; Keller et al., 2017; Sperati et al., 2017; Plenkovi et al., 2018; Yang et al., 2018), tropical cyclone intensity (Alessandrini et al., 2018), air quality (Djalalova et al., 2015; Huang et al., 2017; Delle Monache et al., 2020), and renewable energy (Mahoney et al., 2012; Alessandrini, Delle Monache, Sperati, & Nissen, 2015; Alessandrini, Delle Monache, Sperati, & Cervone, 2015; Vanvyve et al., 2015; Junk et al., 2015; Cervone et al., 2017; Davò et al., 2016; Ferruzzi et al., 2016; Shahriari et al., 2020), but this is the first application of the approach to stratospheric winds.

A common issue with real world use of an AnEn-based system is achieving the post-processing speed that is needed in an operational environment. We outline how a distributed computing system can apply the conventional AnEn globally using the past two years of forecasts in around 20 minutes. We demonstrate that this can be even more efficient by distilling the entire AnEn into a deep neural network (DNN). Distilling, in the machine learning community, refers to training a DNN to memorize and thus mimic another model. It has been used in reinforcement learning (Rusu et al., 2015), to compress an ensemble of predictions into a single model (Hinton et al., 2015; Bucilu et al., 2006), and to approximate a more complex neural network with a simpler one (Ba & Caruana, 2014). In all cases, the idea is to achieve a more computationally efficient version of a skillful, but perhaps inconvenient model.

Since the distilling process is performed offline (in advance), it does not impact real-time operations regardless of the size of the historical dataset. This is a key factor given that the skill of the AnEn tends to improve with a larger historical dataset.

2 Methods

In this section, we review the AnEn algorithm and discuss how it can be implemented at a global scale using distributed computing. We then discuss distilling the AnEn into a DNN. We use the former method to demonstrate that the much more efficient latter method achieves equivalent performance despite being significantly more desirable for use in a production system.

2.1 Conventional Analog Ensemble Algorithm

The AnEn estimates a probability distribution over a forecast parameter, such as wind speed or direction, given a forecast, previous forecasts made by the same model, and corresponding ground truth for those previous forecasts. A search for analogous situations, i.e., previous forecasts we consider to be similar to the current forecast, is performed and ground truth corresponding to these analogous forecasts is used to construct an ensemble (Delle Monache et al., 2013). We report (below) the skill of the analog ensemble and its mean, which we use to generate probabilistic and deterministic predictions, respectively.

Let $f(y|x^f)$ be the probability distribution of the observed value y of some predicted quantity given a model prediction x^f. The vector $x^f = (x^f_1, x^f_2, \cdots, x^f_k)$ contains k predictors from the model forecast, typically including a forecast value for y and other fields considered to be related or providing context on similarity. In the results reported below, x^f includes wind speed and direction.

AnEn is a nearest-neighbor algorithm using a learned distance function. The closest analogs to x^f from previous forecasts are selected, typically restricting to x^i at the same grid point, i.e., forecasts for the same latitude, longitude, and pressure and made for the same lead time. Each forecast has a corresponding ground truth referred to as y^i. We denote the set of forecast and observation tuples at a grid point as $\mathcal{P}$. We rank every $\mathbf{x}^i \in \mathcal{P}$ by a distance function

$$d(x^f, x^i) = \sum_{j=0}^{k} \frac{w^{\mathcal{P}}_j}{\sigma^{\mathcal{P}}_j} \left| x^f_j - x^i_j \right| \tag{1}$$

where $\sigma^{\mathcal{P}}_j$ is a normalization factor, e.g., the standard deviation, to bring all elements of x into a uniform numeric range and $w^{\mathcal{P}}_j$ is per-feature weight. The weight and normalization factors are chosen independently for every grid point to optimize the root-mean square-error (RMSE) of the ensemble mean on the training dataset using a leave one out cross-validation, with the removed $(\mathbf{x}^i, y^i)$ used as $(\mathbf{x}^f, y)$.

The N analogs with the smallest distance to x^f form an ensemble forecast. We use 25 analogs in the results below. The weighted ensemble mean can be used as a deterministic prediction (Delle Monache et al., 2011). We sort the candidate analogs by $d(x^f, x^i)$ and compute the weighted mean on the first N analogs

$$\hat{y}_{wm} = \alpha \sum_{j=0}^{N} \frac{y^i}{\max(d(x^f, x^i), \epsilon)} \tag{2}$$

where α is one over the sum of the weights and ϵ is a very small constant which guards against almost exact matches producing larger weights than can be represented numerically.

This procedure is designed for cases where there is a plurality of analogous situations, but in the case of a rare forecast that is, e.g., larger than most samples in the training, then the AnEn will predict a reversion to the mean and likely not produce a skillful forecast. Similar to Alessandrini et al. (2019) we apply a bias correction term to our forecast of wind speed.

$$\hat{y}_{bc} = \alpha \sum_{j=0}^{N} \frac{y^i}{\max(d(x^f, x^i), \epsilon)} + \left(y^f - \hat{y}_{wm}\right) m \tag{3}$$

where m is a learned parameter to correct for systematic forecast bias.

2.2 Global Scale with Distributed Computing

While the calculation described above at a particular grid point is tractable, a barrier to operationalizing a global AnEn system is processing the corpus of analogs, which can easily grow to 100's of terabytes of data for three-dimensional global predictions over several years. The AnEn algorithm provides a natural partitioning as execution is independent for each grid point and lead time. However, the data is not natively partitioned as both every historical forecast and the current prediction contain a piece of data needed to post-process every grid point. The challenge is to organize the data so that the calculations can be efficiently executed across many datacenter computers. We use the MapReduce paradigm (Dean & Ghemawat, 2004), which allows the computation to run on a distributed computing (cloud) infrastructure like Google's Flume (Chambers et al., 2010). We describe the mechanics of this technique and provide pseudo-code in the supporting information.

Using this technique at appropriate scale, one can post-process a stratospheric wind forecast in 10-20 minutes. In our case, we use 100's to 1000's of datacenter machines. We create a 3D forecast with 20 pressure levels and 0.5-degree resolution in latitude and longitude over 20 lead times. This adds up to the AnEn being applied at around 100 million grid points with analogs from around 3 years of prior forecasts, e.g., around 2196 candidate analogs per grid point. A rough estimate (ignoring inter-process overhead) of trying to do this work on a single machine by multiplying the number of workers by the 10-20 minute compute time highlights why an implementation on a single machine is likely easily too slow for an operational post-processing system.

Despite being able to achieve appropriate scale, this is an expensive computation that grows proportional to corpus size. Post-processing would take significantly longer in the case of a much larger historical corpus. In the next section we discuss distilling this computation into a DNN to address this issue.

2.3 Distilling the Analog Ensemble Into a Deep Neural Network

Every value of the analog ensemble mean, $\hat{y}_{bc}$, corresponds to an HRES prediction of wind speed and direction, x^f, which has been used to generate the analogs included in the set $\mathcal{P}$. A DNN can be used to learn, a.k.a., to memorize or distill, the function mapping the wind speed and direction of the HRES forecast to the resulting analog ensemble mean. An example function for a particular grid point is shown in Figure 1. The AnEn mean wind speed values ($\hat{y}_{bc}$; color shading on the isosurface), are shown for each HRES forecast x^f wind speed (distance from the origin) and direction forecast (rotation around z-axis). The plot is roughly conical, and would be exactly conical if AnEn post-processing had no effect. Some deformation from the perfect cone is introduced by the AnEn algorithm, which we denote by h, i.e., $\hat{y}_{bc} = h_{\mathcal{P}}(x^f)$.

Generating this figure does not require actual new, unseen HRES forecasts. We instead plot the response of the AnEn in anticipation of potential HRES forecasts. Much the same in the learning process, the response curve can be learned by the DNN in ad-

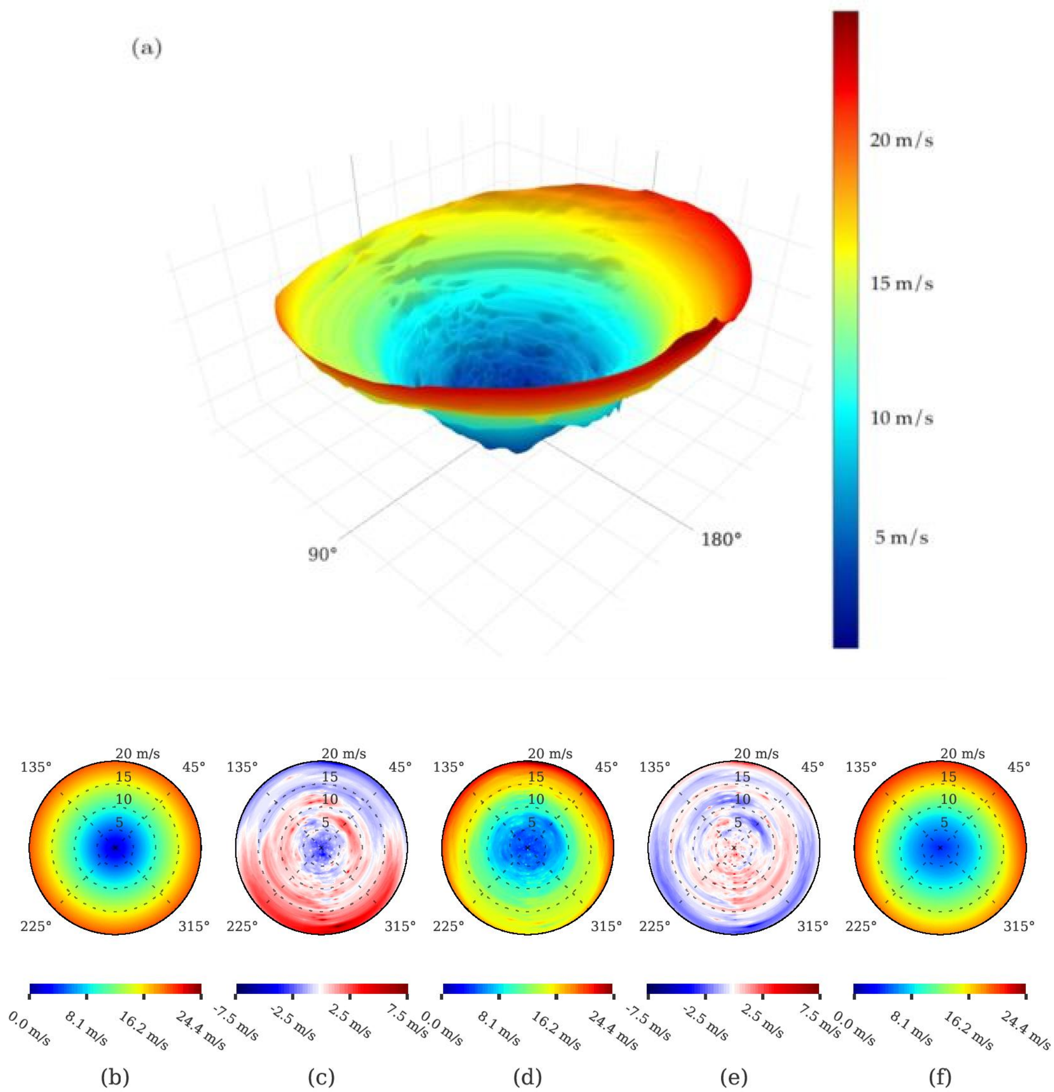


Figure 1. The output of the AnEn is a function mapping x^f to $\hat{y}_{bc}$. The plot in (a) shows the speed prediction for a particular $\mathcal{P}$ swept over speed (distance from origin) and direction (rotation around z-axis). For speed, this function will typically look like a cone. Taking a top-down view of this plot, we see in (b) the identity operator, i.e., no AnEn post-processing. In (d) we see the same cone as (a) from the top down. Finally (f) shows the output of the distilled AnEn. Note that it resembles the transformation applied by the conventional AnEn but is not expected to be identical as the function is generalized across multiple $\mathcal{P}$. The plots in (c) and (e) show the difference (m s^{-1}) between the adjacent plots.

vance of receiving a forecast and when the times comes to post-process the operational forecast we do not require access to the corpus of potential analogs. This makes the distilled AnEn significantly faster than AnEn and more efficient when handling a new forecast in real-time.

In this study we distill $h_{\mathcal{P}}$ by training over all grid points. We train the DNN to learn the function $\hat{y}_{distilled} = \hat{h}(k, x^f)$, where k is a specific grid point. Because $h_{\mathcal{P}}$ varies from grid point to grid point, we add the grid point parameters (latitude, longitude, pressure altitude, and lead time) as arguments to $\hat{h}$ so that the DNN can learn different post-processing transformations at different grid points.

While we discuss results on ensemble mean below, this procedure is not specific to the mean. For example, we have distilled both the ensemble mean of speed and direction (analyzed below), and ensemble forecasts for both quantities into a single DNN with multiple outputs (not shown). The results presented below use a DNN with 10 trainable fully-connected ReLu layers 50 units wide trained with stochastic gradient descent in TensorFlow (Abadi et al., 2015). Full details on the DNN architecture, training parameters, and non-standard data flow, which is conceptually similar to a replay buffer in deep reinforcement learning (Lin, 1992; Mnih et al., 2015), can be found in the supporting information.

We are able to demonstrate a good approximation (next section) for forecast speed and direction within a few billion training examples. Because the training procedure can be performed once (or perhaps periodically, but infrequently) prior to using the network in the operation pipeline, the training time is not particularly important to optimize. Our unoptimized implementation was able to train the network used in our results within a few days on a single CPU (being fed from a distributed data flow). The specific DNN architecture and the data flow to supply training with examples are outlined in greater detail in the supporting information.

Once a network is trained it can be applied point-based, i.e., at a particular place and time with the HRES forecast as input. It adds only a few milliseconds to the real-time computational cost needed to look up forecast data, because the computation performed is a forward (inference) pass through a deep network, i.e., a simple mathematical expression is executed. More study is required to optimize the balance between generalization across different grid points and fitting the particular nuance of a given dataset.

3 Results

We present the forecast skill of the AnEn and distilled AnEn aggregated over a half year of forecasts from July to December, 2019, compared against the ECMWF Integrated Forecast System's high-resolution forecast (HRES) and the ensemble forecast (ENS). We also provide a year long comparison at a different time period (October, 2017 to September, 2018) against the HRES and a persistence ensemble that provides an equivalent result in the supporting information (see Figure S8).

Comprehensive ground truth measurements of winds throughout the stratosphere are not currently available, so to evaluate the quality of the various forecasts we use two proxies for ground truth. The first proxy is the HRES analysis which provides an 'observation' comprehensively across all grid points. The second proxy is true observations from Loon high altitude balloons. This dataset of 10.5 million observations, largely concentrated in the lower latitudes, is significantly more sparse as it only allows us to compare forecasts at places and times where a Loon balloon was present. Taken together, these two comparisons characterize the quality of our method.

To summarize the detailed results that follow, the AnEn and distilled AnEn improve the ECMWF Integrated Forecast System's high-resolution forecast (HRES) of winds in the lower stratosphere. The AnEn methods also produce a skillful probabilistic forecast that is able to quantify the forecast uncertainty, which is an advantage over using the raw deterministic HRES forecast. The ENS ensemble mean outperforms the AnEn methods when evaluating using the HRES analysis as ground truth, but underperforms the AnEn methods on the sparser observations from real Loon flights. The AnEn method has a significantly reduced computational cost of creating or using a 51-member ensemble forecast. Overall the results that follow indicate the AnEn methods are very competitive when both considering practical implications, and on the merits of forecast skill alone.

Our region of interest is the lower stratosphere, from around 48 to 145 hPa. We apply the technique globally and consider the lead times forecast in the HRES which range from 12 hours to 10 days in the future. The results reported in this section are in latitudes below 70 degrees. Results at higher latitudes are similar, but not shown. Our training dataset is the HRES forecasts produced from July, 2016, to June, 2019. We use this to choose weights used in the analog matching process. The validation period is over the HRES forecasts produced from July, 2019, to December, 2019. The data available in the AnEn matching includes all the forecasts in the training dataset plus any additional forecasts between the beginning of the validation time period but prior to the current forecast. This simulates operational use of an AnEn system. To evaluate the distilled AnEn we only use a DNN distilled from the training dataset. In practice, one would distill the AnEn into a new DNN from time to time to incorporate additional forecasts into the training corpus, but that has not been attempted in this study.

Figure 2 shows a comparison of the aggregated add[ldm]deterministic forecast error of the HRES, ENS, AnEn, and distilled AnEn grouped by lead time. Note that 90% bootstrap confidence intervals are omitted because they are very small because for each metric computed and for each lead time we have almost 2 billions and more than 10.5 millions ground truth / prediction pairs when using HRES and Loon data, respectively. The reader can find a view of the these confidence intervals in Figures S4 and S5 of the supporting information. Figure 2(a) shows the evaluation performed using the HRES operational analysis as the ground truth field. The centered root-mean-square (CRMSE) is the portion of the RMSE measuring the random (or anomaly) differences between two fields (Taylor, 2001). The AnEn methods have a lower CRMSE than HRES across all lead times for wind direction, and after hour 84 for wind speed. The AnEn methods have the same skill as ENS up to hour 60 and are competitive for longer lead times, which is remarkable considering that AnEn realtime computation cost, given that it is based on HRES, is significantly lower than ENS. The correlation between the fields and the ground truth is either preserved or improved with the analog-based methods when compared to HRES. The remaining portion of RMSE is the bias, which in this study is significantly lower than CRMSE for all the prediction systems analyzed (not shown). The large reductions of CRMSE for both wind speed and direction obtained with AnEn confirm the ability to tackle conditional biases, which is a result of the algorithm being designed to learn the error of the current prediction from the errors of analogous past forecasts. The ability of the distilled approach to reproduce AnEn deterministic skill is remarkable, as shown by the minimal differences between the two AnEn versions across the different metrics and cases considered.

Figure 2(b) shows the results when the measurements from Loon stratospheric balloons are used as ground-truth. This is a much smaller dataset and lacks global coverage, but is real in situ observations from the stratosphere. (see Figure S2 of the supporting information for the geographical distribution of Loon's measurements). For the convenience of the reader, we provide basic statistical breakdowns and ranges of the observations in the dataset overlapping with our validation period in Figure S1 of the supporting information. The AnEn methods exhibit lower CRMSE than HRES, and significantly lower than ENS for both wind speed and direction. AnEn correlation is significantly higher than ENS for wind speed and better than HRES for wind direction. The better performance of AnEn compared to ENS when using Loon data can be explained by the fact that AnEn, by design, is an excellent downscaling method. This is more evident when making a comparison with data that has a high spatial and temporal resolution, like Loon in situ observations. On the other hand, that is a disadvantage for the coarser ENS.

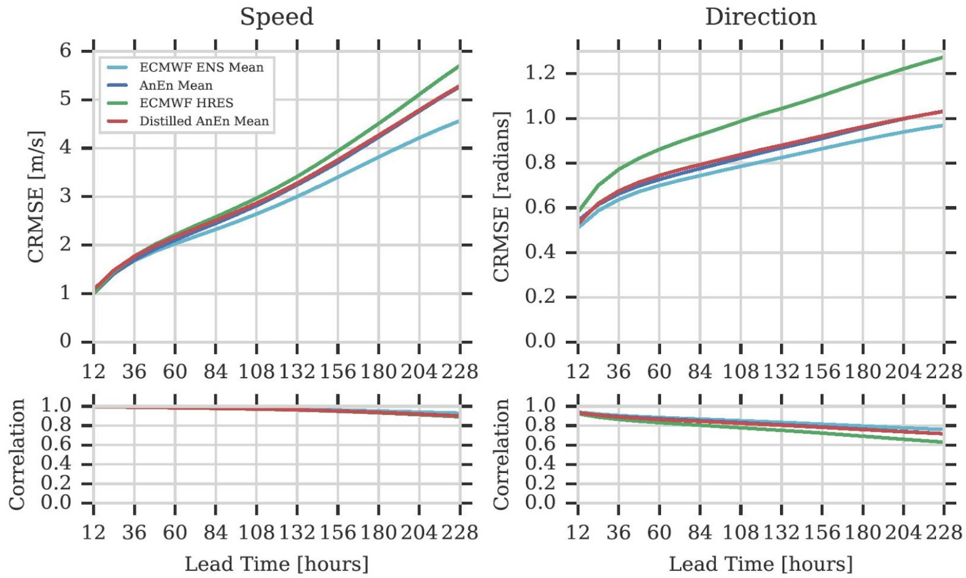
We turn our attention to probabilistic forecasts. We compare the ensemble forecast generated by the AnEn on stratospheric winds to the ENS. Figure 3 shows the continuous ranked probability score (CRPS), rank histogram, and binned-spread/skill plot across different lead times for the AnEn and ENS. We show these metrics for wind direction forecasts using the HRES analysis (left) and Loon data (right) as ground truth. Results for wind speed are qualitatively similar, and are shown in Figure S3 of the supporting information.

The CRPS provides an assessment of the quality of a probabilistic forecast that is not necessarily of a binary event (Hersbach, 2000). It is the probabilistic equivalent of the mean absolute error for deterministic predictions, and a zero indicates a perfect forecast. Similarly to the deterministic results with HRES analysis as the ground truth, AnEn is competitive with ENS up to hour 60 and better then HRES at all lead times. However, when this performance metric is calculated against the Loon data, AnEn is significantly better even of ENS, reducing the latter CRPS between 7 and 70%.

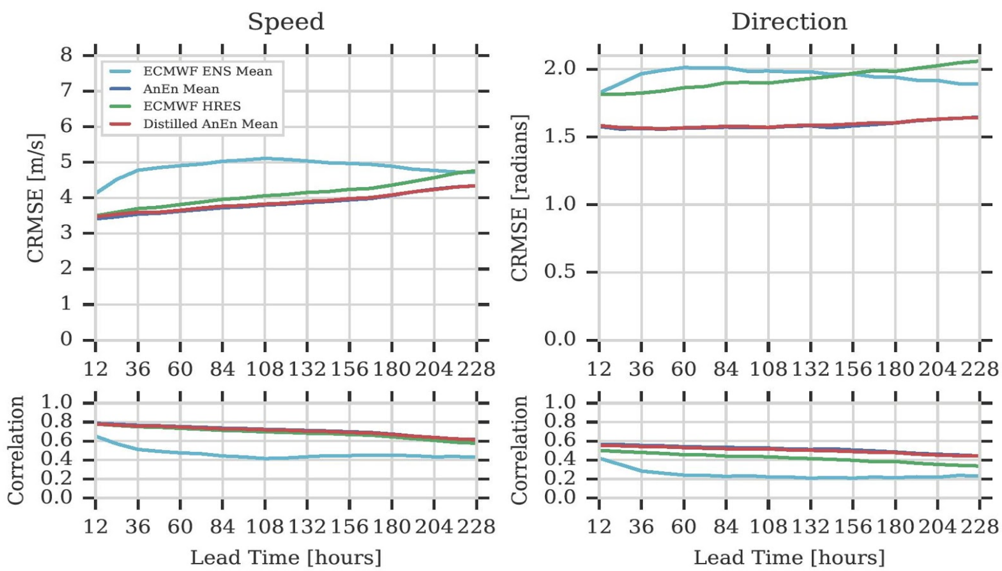
The rank histogram estimates the statistical consistency of an ensemble (Anderson, 1996). For a perfect ensemble, the observation will appear to be drawn from the same distribution as the ensemble members. The rank histogram is flat in that case. The ENS has a U-shaped rank histogram with both ground-truth data sets, which indicates a lack of spread. With HRES as the ground-truth, the AnEn rank histogram instead is closer to the ideal flat shape, though it exhibits for the first few lead times a dome shape indicating an excess spread. This may reflect that the AnEn is including a few analogs that have a larger match distance at early lead times. Against Loon data, AnEn has a rank histogram significantly closer to the ideal shape, being U-shaped but less so than ENS.

The binned-spread/skill plot (van den Dool, 1989; Wang & Bishop, 2003) (which is only applicable to probabilistic predictions) characterizes, perhaps, the most important attribute of an ensemble system: the ability to quantify uncertainty while accounting for the flow-dependent error characteristics. This is approximated by analyzing the spread-skill relationship across different spread bins. A perfect ensemble results in a diagonal line. Against the HRES analysis, AnEn is closer to the diagonal than ENS, although both system exhibit a good spread-skill relationship. However, when Loon measurements are used as ground-truth, AnEn exhibits a significantly better ability to characterize the prediction uncertainty. The ENS diagram is horizontal for most bins and lead times, which reflects a lack of a spread-skill relationship for the ECMWF ensemble system when predicting wind direction.

Figure 4(a) shows an example of the difference in forecast wind speed between the (distilled) AnEn-based forecast and the HRES across a constant-pressure slice of the stratosphere. Figure 4(b) shows the percent change. In this particular example, which was arbitrarily chosen at random, the largest percent changes are made in the tropics. This tends to be a common pattern. Most regions we have analyzed see forecast improvements with the AnEn when compared to HRES and the largest improvements are at latitudes below 23 degrees. The arrows in Figure 4(b) indicate the flow of the wind direction vector field at this pressure level.



(a)



(b)

Figure 2. A deterministic wind speed and direction forecast skill comparison between HRES and the means of ECMWF ENS, AnEn, and Distilled AnEn over all lead times is shown using as ground truth (a) HRES analysis and (b) Loon observations of stratospheric winds. The metrics are computed for each lead time across the available observation-prediction pairs from all the grid points in latitudes below 70 degrees.

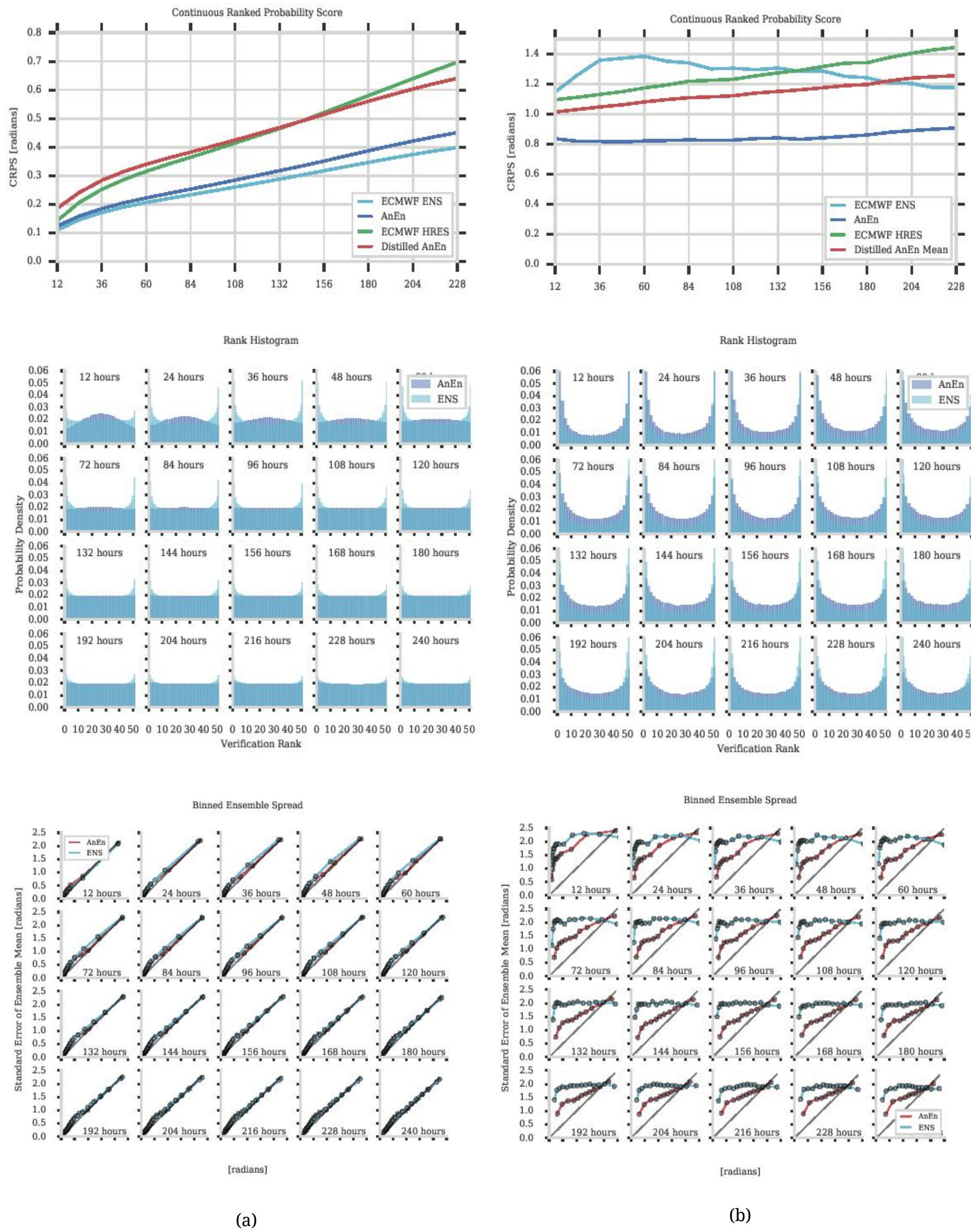


Figure 3. Probabilistic forecast evaluation metrics comparing the AnEn forecast of wind direction to forecasts produced by a ENS. Results with HRES analysis as ground truth are shown on the left (a), while results against Loon's measurements are on the right (b). From top to bottom, the metrics shown are CRPS, rank histogram, and binned-spread skill.

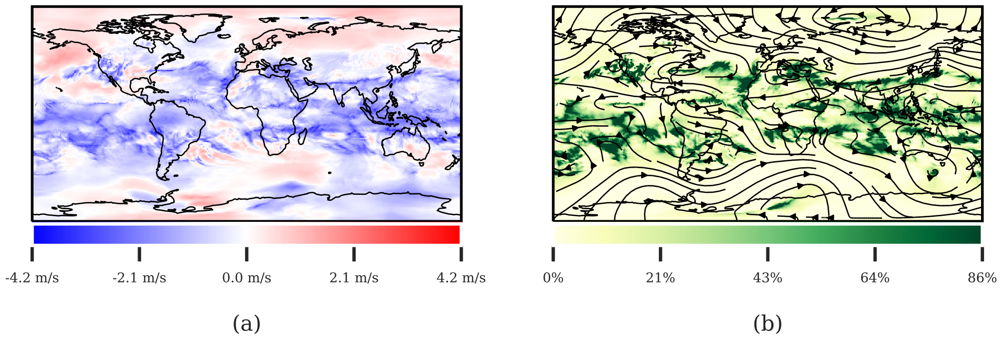


Figure 4. Differences between the HRES and distilled AnEn forecast of wind speed worldwide at 50 hPa for 2019-10-20 18:00 UTC with a 5 day lead time. (a) shows the difference between the two forecasts and (b) shows the absolute relative change (absolute value of change between the two forecasts as a percentage of the HRES forecast) between the two forecasts with the direction field overlaid.

4 Discussion

The analog ensemble (AnEn) and distilled AnEn improve the European Centre for Medium-Range Weather Forecasts (ECMWF) high-resolution (HRES) deterministic forecast of winds in the lower stratosphere in our evaluation over half a year of forecasts using both global ECMWF analyses and a smaller set of observations from Loon high altitude balloons as ground truth. The AnEn is also competitive with ECMWF ensemble (ENS) system up to hour 60 for deterministic and probabilistic forecasts when HRES analysis is used as ground truth and significantly better when the performance metrics are computed against Loon's dataset of true ground truth observations. In particular, AnEn is able to quantify the prediction uncertainty, as evident from the analysis of the probabilistic systems spread-skill relationship, while ENS lacks such attribute, particularly for wind direction predictions. This is true, despite AnEn being computationally cheaper in real-time.

Physics-based numerical weather models, such as the ECMWF's HRES, are marvels of engineering and science and produce high quality forecasts of many meteorological fields in a coupled and principled manner. However, improvements can sometimes come at great cost, both in research time and in computation and power. Pure machine learning techniques, i.e., end to end learned model-free forecasting, hold promise but are limited due to training on a small number of observations and a limited ability to extrapolate beyond that training data.

For example, a weakness of an analogs-based approach is new situations. If not handled properly, post-processing can reduce forecast skill. We found a specific example of this in our experiments which covered a period of vortex breakdown over North America during February, 2018. Because there was only a single Northern hemisphere winter in our training corpus and it did not exhibit a large vortex breakdown over North America, the algorithm was not able to find analogs with sufficiently high wind speed. When testing the method without the bias correction term of Equation (3), the method decreased forecast skill. While bias correction acts as a stop-gap in this scenario, the desired approach would be to extend the historical corpus to be long enough to find analogous vortex breakdown scenarios.

Recently there have been several contributions exploring the potential of machine learning for weather and climate predictions (e.g., Tao et al., 2016; Gagne II et al., 2017; McGovern et al., 2017; Rasp & Lerch, 2018; Scher, 2018; Chapman et al., 2019; Gagne II et al., 2019; Lagerquist et al., 2019; McGovern et al., 2019; Burke et al., 2020). However, although there have been encouraging attempts to develop pure machine learning weather forecasting methods (e.g., Weyn et al., 2019), those may still be out of reach given the relatively low number of available learning examples compared to the number of degrees of freedom in the atmosphere. Currently, successful attempts have been reported only in replacing individual physical processes (e.g., O'Gorman & Dwyer, 2018).

The AnEn distilling procedure can be seen through two lenses. One can consider the distilled AnEn as an approximation of the conventional AnEn, i.e., a highly efficient implementation of the conventional technique. A second lens is that the DNN is the learning technique and the process of distilling the AnEn is a data augmentation method to increase the number of examples used to train the network. One may prefer to distill an AnEn over directly training a DNN to improve forecasts because DNNs have a high capacity (the complexity of the function the model can encode) and, unfortunately, there are limited numbers of forecast-ground truth pairs that are available for training. The lack of training data is exacerbated by growing the number of outputs we want the DNN to produce, e.g., a probability distribution over our forecast field. The AnEn has been shown to generalize well as a machine learning algorithm, i.e., to provide an improved forecast when deployed on long validation periods on unseen meteorological forecasts. The distilled AnEn bootstraps training a DNN off the AnEn, effectively combining the AnEn's strength of being able to generate forecasts with a relatively small corpus of training examples with the DNNs ability to memorize this complex correction function with a significantly smaller amount of data.

This may be a pragmatic compromise. It seems there is a large opportunity for machine learning by relying on the extremely high quality numerical weather models and making improvements in post-processing. The authors believe there is potential in this fused approach. This paper provides an example of how machine learning can contribute to increasing forecast skill and uncertainty quantification. As forecasts are asked to be simultaneously faster, more granular, and more accurate, the physics-based models can continue to do the heavy lifting and machine learning post-processing can improve forecast quality to alleviate some issues of scale.

Acknowledgments

For more information on Loon stratospheric wind observations contact Loon at stratospheric-data@loon.com. The data used in this this paper is portion of the publicly available Loon observations data set (Candido, 2020).

We are grateful to James Antifaev, Marc Bellemare, Rob Carver, Justin Garofoli, Max Kamenetsky, Sameera Ponda, and Aneesh Subramanian for providing useful comments and edits to an earlier version of this paper.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.* Retrieved from `https://www.tensorflow.org/` (Software available from tensorflow.org)

Alessandrini, S., Delle Monache, L., Rozoff, C. M., & Lewis, W. E. (2018). Probabilistic prediction of tropical cyclone intensity with an analog ensemble. *Monthly Weather Review*, *146*(6), 1723-1744. Retrieved from `https://doi.org/10.1175/MWR-D-17-0314.1` doi: 10.1175/MWR-D-17-0314.1

Alessandrini, S., Delle Monache, L., Sperati, S., & Cervone, G. (2015). An analog ensemble for short-term probabilistic solar power forecast. *Applied Energy*, *157*, 95–110. Retrieved 2016-11-21, from http://www.sciencedirect.com/science/article/pii/S0306261915009368

Alessandrini, S., Delle Monache, L., Sperati, S., & Nissen, J. N. (2015). A novel application of an analog ensemble for short-term wind power forecasting. *Renewable Energy*, *76*, 768-781. Retrieved from http://www.sciencedirect.com/science/article/pii/S0960148114007915" doi: https://doi.org/10.1016/j.renene.2014.11.061

Alessandrini, S., Sperati, S., & Delle Monache, L. (2019). Improving the analog ensemble wind speed forecasts for rare events. *Monthly Weather Review*, *147*(7), 2677-2692. Retrieved from https://doi.org/10.1175/MWR-D-19-0006.1 doi: 10.1175/MWR-D-19-0006.1

Anderson, J. L. (1996). A method for producing and evaluating probabilistic forecasts from ensemble model integrations. *Journal of Climate*, *9*(7), 1518-1530. Retrieved from https://doi.org/10.1175/1520-0442(1996)009<1518:AMFPAE>2.0.CO;2 doi: 10.1175/1520-0442(1996)009⟨1518:AMFPAE⟩2.0.CO;2

Ba, J., & Caruana, R. (2014). Do deep nets really need to be deep? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 27* (pp. 2654–2662). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/5484-do-deep-nets-really-need-to-be-deep.pdf

Bucilu, C., Caruana, R., & Niculescu-Mizil, A. (2006). Model compression. In *Proceedings of the 12th acm sigkdd international conference on knowledge discovery and data mining* (pp. 535–541).

Burke, A., Snook, N., Gagne II, D. J., McCorkle, S., & McGovern, A. (2020). Calibration of machine learningbased probabilistic hail predictions for operational forecasting. *Weather and Forecasting*, *35*(1), 149-168. Retrieved from https://doi.org/10.1175/WAF-D-19-0105.1 doi: 10.1175/WAF-D-19-0105.1

Candido, S. (2020, April). *Loon stratospheric sensor data.* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.3763022 doi: 10.5281/zenodo.3763022

Cervone, G., Clemente-Harding, L., Alessandrini, S., & Monache, L. D. (2017). Short-term photovoltaic power forecasting using artificial neural networks and an analog ensemble. *Renewable Energy*, *108*, 274 - 286. Retrieved from http://www.sciencedirect.com/science/article/pii/S0960148117301386 doi: https://doi.org/10.1016/j.renene.2017.02.052

Chambers, C., Raniwala, A., Perry, F., Adams, S., Henry, R., Bradshaw, R., & Nathan. (2010). Flumejava: Easy, efficient data-parallel pipelines. In *Acm sigplan conference on programming language design and implementation (pldi)* (p. 363-375). 2 Penn Plaza, Suite 701 New York, NY 10121-0701. Retrieved from http://dl.acm.org/citation.cfm?id=1806638

Chapman, W. E., Subramanian, A. C., Delle Monache, L., Xie, S. P., & Ralph, F. M. (2019). Improving atmospheric river forecasts with machine learning. *Geophysical Research Letters*. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019GL083662 doi: 10.1029/2019GL083662

Davò, F., Alessandrini, S., Sperati, S., Delle Monache, L., Airoldi, D., & Vespucci, M. T. (2016). Post-processing techniques and principal component analysis for regional wind power and solar irradiance forecasting. *Solar Energy*, *134*, 327 - 338. Retrieved from http://www.sciencedirect.com/science/article/pii/S0038092X16300962 doi: https://doi.org/10.1016/j.solener.2016.04.049

Dean, J., & Ghemawat, S. (2004). Mapreduce: Simplified data processing on large clusters. In *Osdi'04: Sixth symposium on operating system design and imple-*

mentation (pp. 137–150). San Francisco, CA.

Delle Monache, L., Alessandrini, D. I., S., Wilczak, K. J. C., James, & Kumar, R. (2020). Improving air quality predictions over the united states with an analog ensemble. *Weather and Forecasting*.

Delle Monache, L., Eckel, F. A., Rife, D. L., Nagarajan, B., & Searight, K. (2013). Probabilistic weather prediction with an analog ensemble. *Monthly Weather Review*, *141*(10), 3498-3516. Retrieved from `https://doi.org/10.1175/MWR-D-12-00281.1` doi: 10.1175/MWR-D-12-00281.1

Delle Monache, L., Nipen, T., Liu, Y., Roux, G., & Stull, R. (2011). Kalman filter and analog schemes to postprocess numerical weather predictions. *Monthly Weather Review*, *139*(11), 3554–3570.

Djalalova, I., Delle Monache, L., & Wilczak, J. (2015). Pm 2.5 analog forecast and kalman filter post-processing for the community multiscale air quality (cmaq) model. *Atmospheric Environment*, *108*, 76–87.

Eckel, F. A., & Delle Monache, L. (2016, February). A hybrid nwp-analog ensemble. *Monthly Weather Review*, *144*, 897–911. Retrieved 2016-02-16, from `https://journals.ametsoc.org/doi/full/10.1175/MWR-D-15-0096.1` doi: 10.1175/MWR-D-15-0096.1

Ferruzzi, G., Cervone, G., Delle Monache, L., Graditi, G., & Jacobone, F. (2016). Optimal bidding in a day-ahead energy market for micro grid under uncertainty in renewable energy production. *Energy*, *106*, 194 - 202. Retrieved from `://www.sciencedirect.com/science/article/pii/S0360544216302432` doi: https://doi.org/10.1016/j.energy.2016.02.166

Frediani, M. E. B., Hopson, T. M., Hacker, J. P., Anagnostou, E. N., Delle Monache, L., & Vandenberghe, F. (2017, December). Object-based analog forecasts for surface wind speed. *Monthly Weather Review*, *145*(12), 5083–5102. Retrieved 2017-12-20, from `http://journals.ametsoc.org/doi/10.1175/MWR-D-17-0012.1` doi: 10.1175/MWR-D-17-0012.1

Gagne II, D. J., Haupt, S. E., Nychka, D. W., & Thompson, G. (2019). Interpretable deep learning for spatial analysis of severe hailstorms. *Monthly Weather Review*, *147*(8), 2827-2845. Retrieved from `https://doi.org/10.1175/MWR-D-18-0316.1` doi: 10.1175/MWR-D-18-0316.1

Gagne II, D. J., McGovern, H. S. H., Amy, Sobash, R. A., Williams, J. K., & Xue, M. (2017). Stormbased probabilistic hail forecasting with machine learning applied to convectionallowing ensembles. *Weather and Forecasting*, *32*(5), 1819-1840. Retrieved from `https://doi.org/10.1175/wafd170010.1`

Hamill, T. M., & Whitaker, J. S. (2006). Probabilistic quantitative precipitation forecasts based on reforecast analogs: Theory and application. *Monthly Weather Review*, *134*(11), 3209-3229. Retrieved from `https://doi.org/10.1175/MWR3237.1` doi: 10.1175/MWR3237.1

Hersbach, H. (2000). Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather and Forecasting*, *15*(5), 559-570. Retrieved from `https://doi.org/10.1175/1520-0434(2000)015<0559:DOTCRP>2.0.CO;2` doi: 10.1175/1520-0434(2000)015⟨0559:DOTCRP⟩2.0.CO;2

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. In *Nips deep learning and representation learning workshop.* Retrieved from `http://arxiv.org/abs/1503.02531`

Huang, J., McQueen, J., Wilczak, J., Djalalova, I., Stajner, I., Shafran, P., ... Delle Monache, L. (2017). Improving noaa naqfc pm2.5 predictions with a bias correction approach. *Weather and Forecasting*, *32*(2), 407-421. Retrieved from `https://doi.org/10.1175/WAF-D-16-0118.1` doi: 10.1175/WAF-D-16-0118.1

Junk, C., Delle Monache, L., Alessandrini, S., Cervone, G., & von Bremen, L. (2015). Predictor-weighting strategies for probabilistic wind power forecasting with an analog ensemble. *Meteorologische Zeitschrift*, *24*(4), 361–79.

Keller, J. D., Delle Monache, L., & Alessandrini, S. (2017, July). Statistical downscaling of a high-resolution precipitation reanalysis using the analog ensemble method. *Monthly Weather Review*, *56*, 2081–2095. Retrieved 2017-07-12, from `https://journals.ametsoc.org/doi/full/10.1175/JAMC-D-16-0380.1` doi: 10.1175/JAMC-D-16-0380.1

Klausner, Z., Kaplan, H., & Fattal, E. (2009). The similar days method for predicting near surface wind vectors. *Meteorological Applications*, *16*(4), 569-579. Retrieved from `https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/met.158` doi: 10.1002/met.158

Lagerquist, R., McGovern, A., & Gagne II, D. J. (2019). Deep learning for spatially explicit prediction of synoptic-scale fronts. *Weather and Forecasting*, *34*(4), 1137-1160. Retrieved from `https://doi.org/10.1175/WAF-D-18-0183.1` doi: 10.1175/WAF-D-18-0183.1

Lin, L.-J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, *8*(3-4), 293–321.

Lorenz, E. N. (1969). The predictability of a flow which possesses many scales of motion. *Tellus*, *21*(3), 289-307. Retrieved from `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.2153-3490.1969.tb00444.x` doi: 10.1111/j.2153-3490.1969.tb00444.x

Mahoney, W. P., Parks, K., Wiener, G., Liu, Y., Myers, W. L., Sun, J., … Haupt, S. E. (2012, Oct). A wind power forecasting system to optimize grid integration. *IEEE Transactions on Sustainable Energy*, *3*(4), 670-682. doi: 10.1109/TSTE.2012.2201758

McGovern, A., Elmore, K. L., Gagne, D. J., Haupt, S. E., Karstens, C. D., Lagerquist, R., … Williams, J. K. (2017). Using artificial intelligence to improve real-time decision-making for high-impact weather. *Bulletin of the American Meteorological Society*, *98*(10), 2073-2090. Retrieved from `https://doi.org/10.1175/BAMS-D-16-0123.1` doi: 10.1175/BAMS-D-16-0123.1

McGovern, A., Lagerquist, R., John Gagne, D., Jergensen, G. E., Elmore, K. L., Homeyer, C. R., & Smith, T. (2019). Making the black box more transparent: Understanding the physical implications of machine learning. *Bulletin of the American Meteorological Society*, *100*(11), 2175-2199. Retrieved from `https://doi.org/10.1175/BAMS-D-18-0195.1` doi: 10.1175/BAMS-D-18-0195.1

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., … others (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529.

Nagarajan, B., Delle Monache, L., Hacker, J. P., Rife, D. L., Searight, K., Knievel, J. C., & Nipen, T. N. (2015, September). An evaluation of analog-based post-processing methods across several variables and forecast models. *Weather and Forecasting*, *30*, 1623–1643. Retrieved 2015-12-01, from `http://journals.ametsoc.org/doi/abs/10.1175/WAF-D-14-00081.1` doi: 10.1175/WAF-D-14-00081.1

O'Gorman, P. A., & Dwyer, J. G. (2018). Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change, and extreme events. *Journal of Advances in Modeling Earth Systems*, *10*(10), 2548-2563. Retrieved from `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018MS001351` doi: 10.1029/2018MS001351

Panziera, L., Germann, U., Gabella, M., & Mandapaka, P. V. (2011). Noranowcasting of orographic rainfall by means of analogues. *Quarterly Journal of the Royal Meteorological Society*, *137*(661), 2106-2123. Retrieved from `https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.878` doi: 10.1002/qj.878

Plenkovi, I. O., Delle Monache, L., Horvath, K., & Hrastinski, M. (2018). Deterministic wind speed predictions with analog-based methods over complex topography. *Journal of Applied Meteorology and Climatology*, *57*(9), 2047-2070. Retrieved from https://doi.org/10.1175/JAMC-D-17-0151.1 doi: 10.1175/JAMC-D-17-0151.1

Rasp, S., & Lerch, S. (2018). Neural networks for postprocessing ensemble weather forecasts. *Monthly Weather Review*, *146*(11), 3885-3900. Retrieved from https://doi.org/10.1175/MWR-D-18-0187.1 doi: 10.1175/MWR-D-18-0187.1

Rusu, A. A., Colmenarejo, S. G., Gulcehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., … Hadsell, R. (2015). Policy distillation. *arXiv preprint arXiv:1511.06295*.

Scher, S. (2018). Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning. *Geophysical Research Letters*, *45*(22), 12,616-12,622. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018GL080704 doi: 10.1029/2018GL080704

Shahriari, M., Cervone, G., Clemente-Harding, L., & Delle Monache, L. (2020). Using the analog ensemble method as a proxy measurement for wind power predictability", journal = "renewable energy. , *146*, 789 - 801. Retrieved from http://www.sciencedirect.com/science/article/pii/S0960148119309668 doi: https://doi.org/10.1016/j.renene.2019.06.132

Sperati, S., Alessandrini, S., & Delle Monache, L. (2017). Gridded probabilistic weather forecasts with an analog ensemble. *Quarterly Journal of the Royal Meteorological Society*, *143*(708), 2874–2885.

Tao, Y., Gao, X., Hsu, K., Sorooshian, S., & Ihler, A. (2016). A deep neural network modeling framework to reduce bias in satellite precipitation products. *Journal of Hydrometeorology*, *17*(3), 931-945. Retrieved from https://doi.org/10.1175/JHM-D-15-0075.1 doi: 10.1175/JHM-D-15-0075.1

Taylor, K. E. (2001). Summarizing multiple aspects of model performance in a single diagram. *Journal of Geophysical Research: Atmospheres*, *106*(D7), 7183–7192.

van den Dool, H. M. (1989). A new look at weather forecasting through analogues. *Monthly Weather Review*, *117*(10), 2230-2247. Retrieved from https://doi.org/10.1175/1520-0493(1989)117<2230:ANLAWF>2.0.CO;2 doi: 10.1175/1520-0493(1989)117⟨2230:ANLAWF⟩2.0.CO;2

Van den Dool, H. M. (1994). Searching for analogues, how long must we wait? *Tellus A*, *46*(3), 314-324. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1034/j.1600-0870.1994.t01-2-00006.x doi: 10.1034/j.1600-0870.1994.t01-2-00006.x

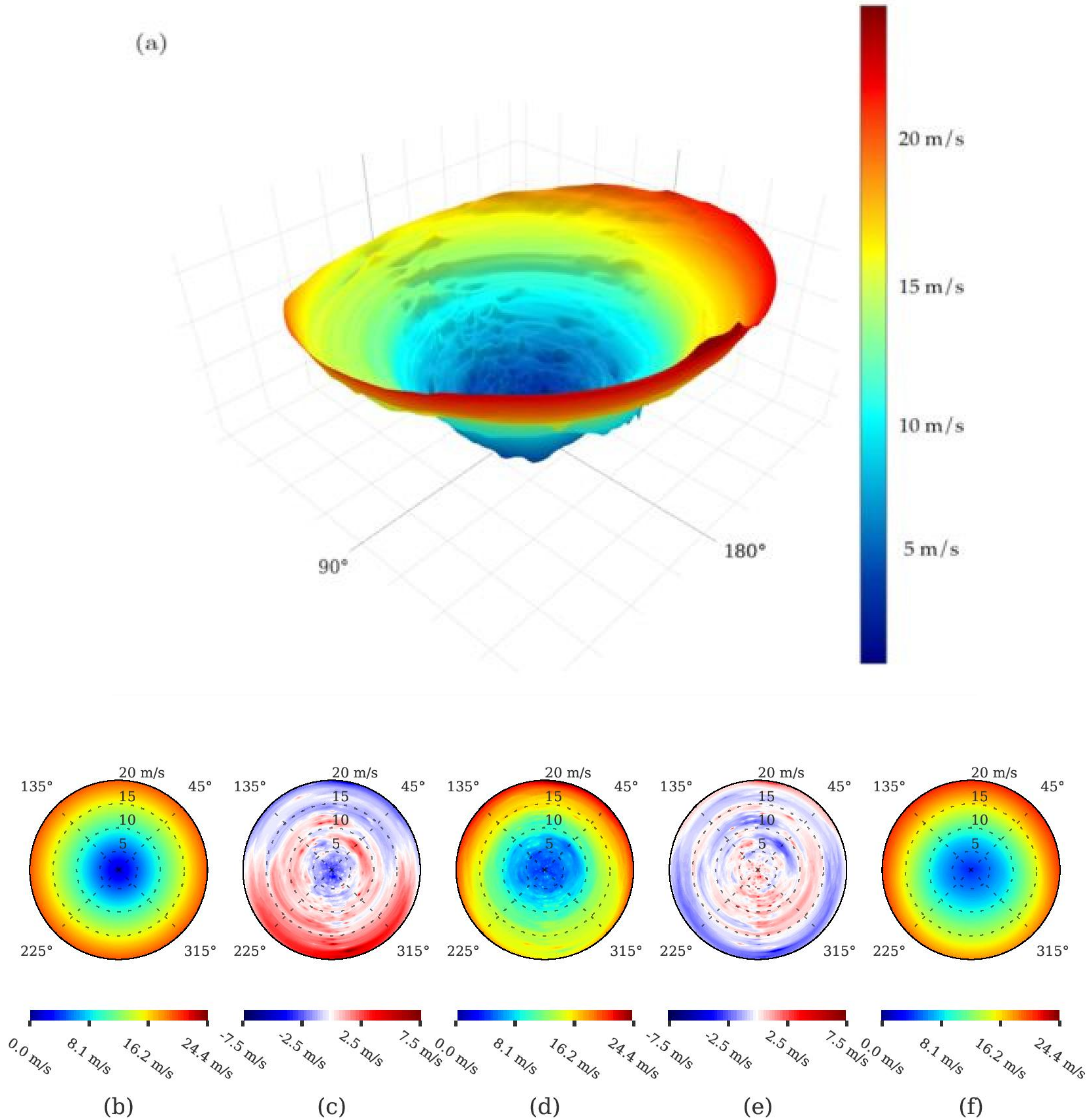
Vanvyve, E., Delle Monache, L., Monaghan, A. J., & Pinto, J. O. (2015). Wind resource estimates with an analog ensemble approach. *Renewable Energy*, *74*, 761 - 773. Retrieved from http://www.sciencedirect.com/science/article/pii/S0960148114005308 doi: https://doi.org/10.1016/j.renene.2014.08.060

Wang, X., & Bishop, C. H. (2003). A comparison of breeding and ensemble transform kalman filter ensemble forecast schemes. *Journal of the Atmospheric Sciences*, *60*(9), 1140-1158. Retrieved from https://doi.org/10.1175/1520-0469(2003)060<1140:ACOBAE>2.0.CO;2 doi: 10.1175/1520-0469(2003)060⟨1140:ACOBAE⟩2.0.CO;2

Weyn, J. A., Durran, D. R., & Caruana, R. (2019). 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*, *11*(8), 2680-2693. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019MS001705 doi: 10.1029/2019MS001705

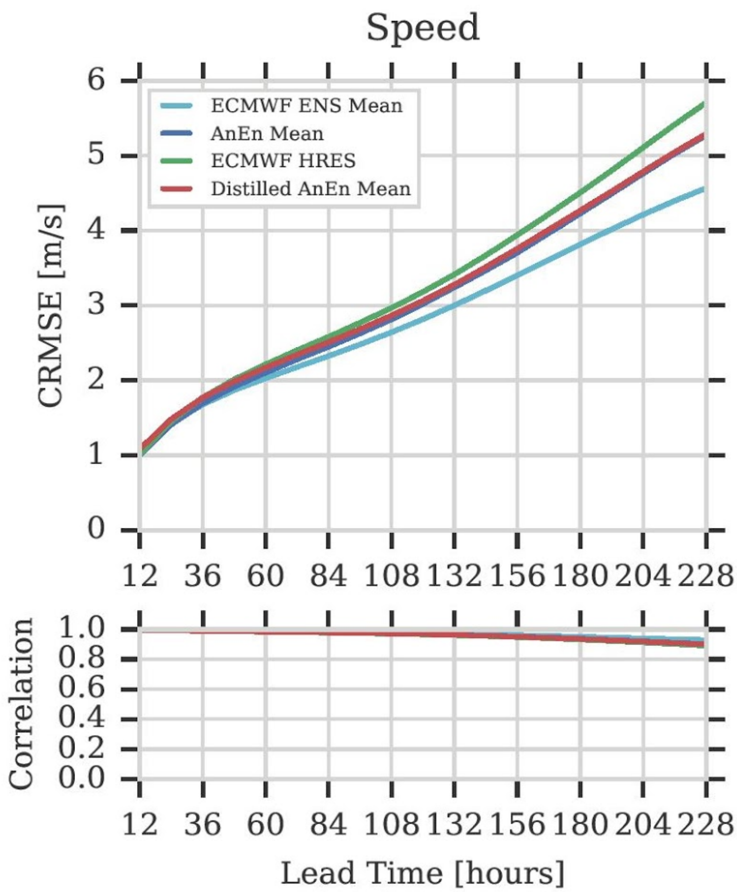
Yang, J., Astitha, M., Delle Monache, L., & Alessandrini, S. (2018). An analog technique to improve storm wind speed prediction using a dual nwp model approach. *Monthly Weather Review*, *146*(12), 4057-4077. Retrieved from https://doi.org/10.1175/MWR-D-17-0198.1 doi: 10.1175/MWR-D-17-0198.1

Figure 1.



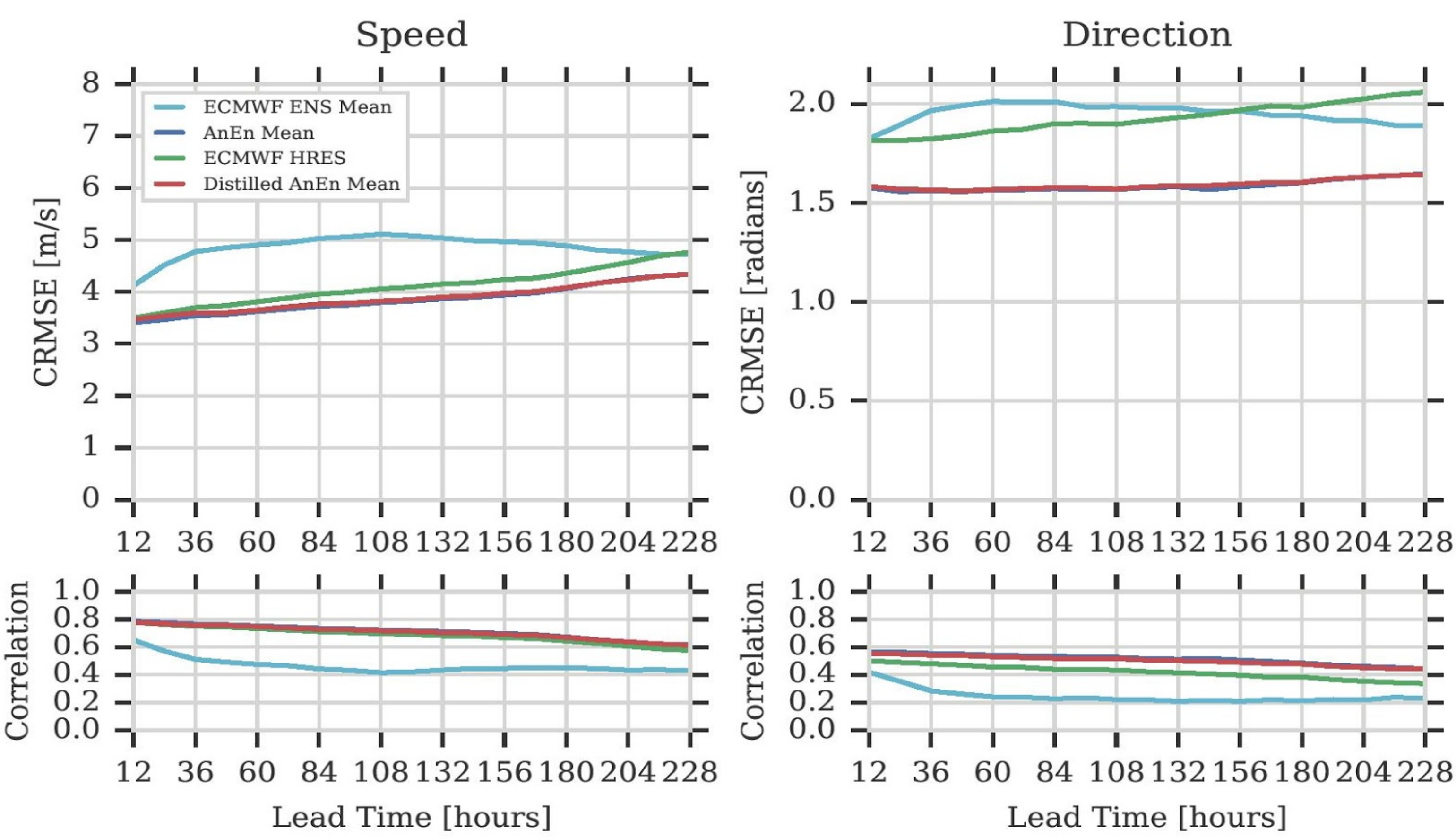
(a)
20 m/s
15 m/s
10 m/s
5 m/s
90°
180°
135°
45°
225°
315°
20 m/s
15
10
5
0.0 m/s
8.1 m/s
16.2 m/s
24.4 m/s
-7.5 m/s
-2.5 m/s
2.5 m/s
7.5 m/s
(b)
(c)
(d)
(e)
(f)

Figure 2.



Speed
Direction
ECMWF ENS Mean
AnEn Mean
ECMWF HRES
Distilled AnEn Mean
CRMSE [m/s]
CRMSE [radians]
Correlation
Lead Time [hours]

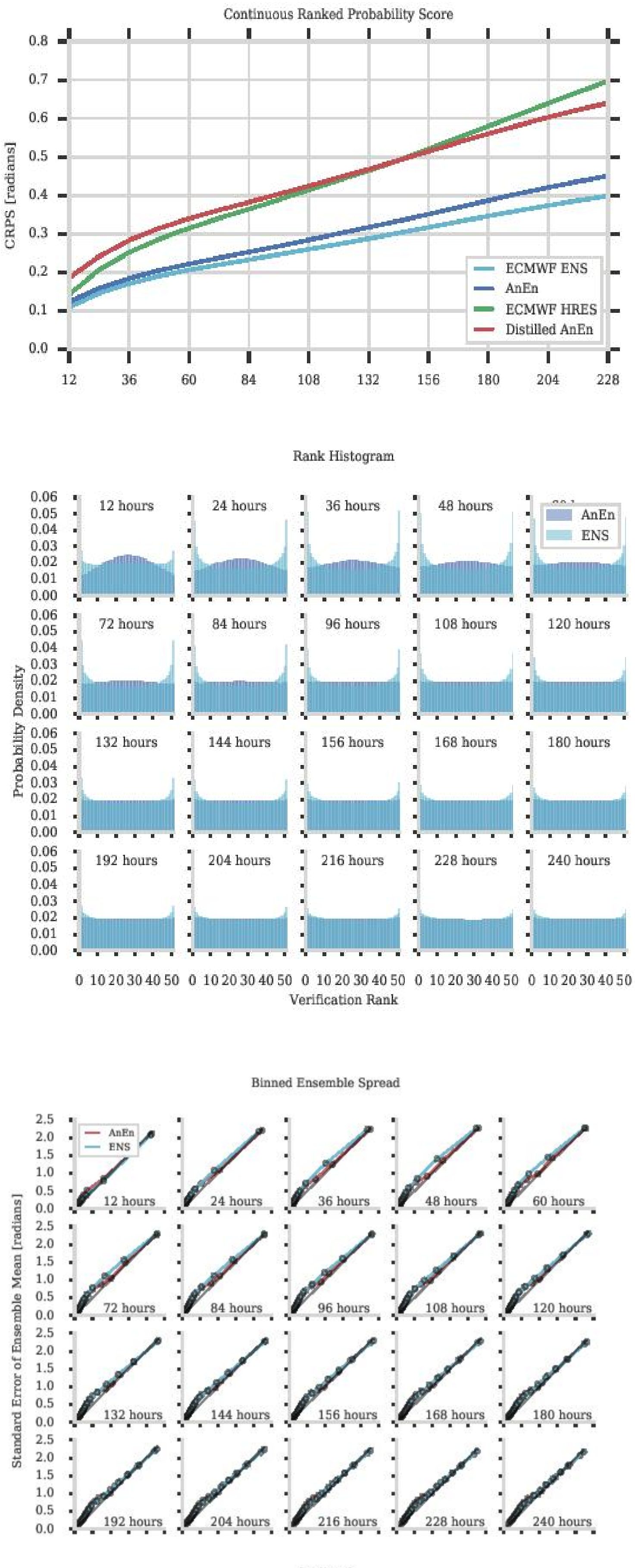
(a)



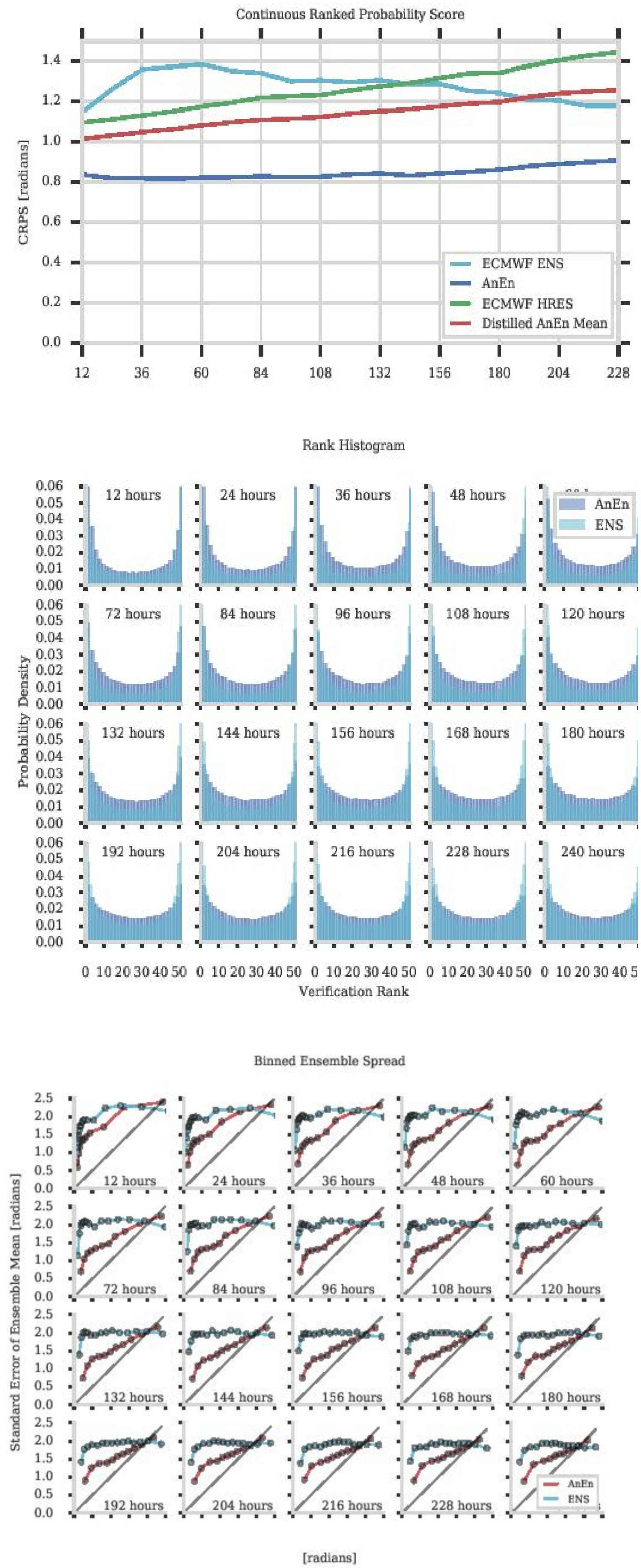
Speed
Direction
ECMWF ENS Mean
AnEn Mean
ECMWF HRES
Distilled AnEn Mean
CRMSE [m/s]
CRMSE [radians]
Correlation
Lead Time [hours]

(b)

Figure 3.

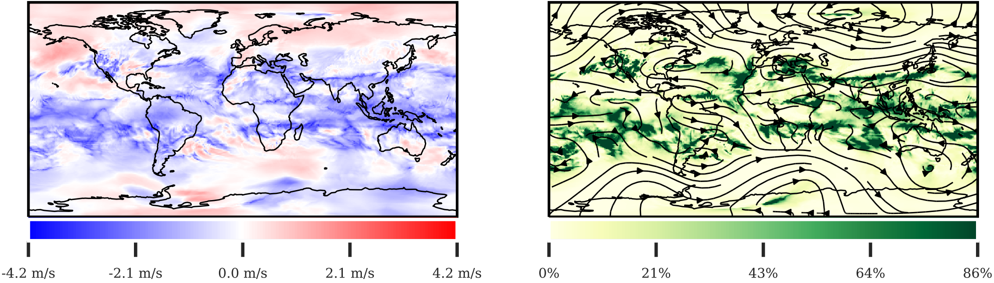

Continuous Ranked Probability Score
CRPS [radians]
ECMWF ENS
AnEn
ECMWF HRES
Distilled AnEn
Rank Histogram
AnEn
ENS
12 hours
24 hours
36 hours
48 hours
72 hours
84 hours
96 hours
108 hours
120 hours
132 hours
144 hours
156 hours
168 hours
180 hours
192 hours
204 hours
216 hours
228 hours
240 hours
Probability Density
Verification Rank
Binned Ensemble Spread
AnEn
ENS
12 hours
24 hours
36 hours
48 hours
60 hours
72 hours
84 hours
96 hours
108 hours
120 hours
132 hours
144 hours
156 hours
168 hours
180 hours
192 hours
204 hours
216 hours
228 hours
240 hours
Standard Error of Ensemble Mean [radians]
[radians]

(a)


Continuous Ranked Probability Score
CRPS [radians]
ECMWF ENS
AnEn
ECMWF HRES
Distilled AnEn Mean
Rank Histogram
AnEn
ENS
12 hours
24 hours
36 hours
48 hours
72 hours
84 hours
96 hours
108 hours
120 hours
132 hours
144 hours
156 hours
168 hours
180 hours
192 hours
204 hours
216 hours
228 hours
240 hours
Probability Density
Verification Rank
Binned Ensemble Spread
12 hours
24 hours
36 hours
48 hours
60 hours
72 hours
84 hours
96 hours
108 hours
120 hours
132 hours
144 hours
156 hours
168 hours
180 hours
192 hours
204 hours
216 hours
228 hours
AnEn
ENS
Standard Error of Ensemble Mean [radians]
[radians]

(b)

Figure 4.



(a)

(b)

Supporting Information for Improving Wind Forecasts in the Lower Stratosphere by Distilling an Analog Ensemble into a Deep Neural Network

Salvatore Candido[1], Aakanksha Singh[1], and Luca Delle Monache[1,2]

[1]Loon, Mountain View, California, USA

[2]Center for Western Weather and Water Extremes, La Jolla, California, USA

Contents of this file

1. Computing the Conventional Analog Ensemble with MapReduce
2. Details of Training a Distilled Analog Ensemble Model
3. Statistics of the Loon Data Set
4. Probabilistic Evaluation Metrics for Wind Speed
5. Confidence Intervals on Deterministic Evaluations
6. Algorithm Skill Comparison By Geography
7. Results for an Earlier Validation Period

Introduction

The supplemental data here covers a few disparate sets of material. The first two sections provide additional technical detail on the methods of the paper that are not necessary to understand the approach, but are useful when replicating the results. We then share statistical breakdowns of the Loon observations which can also be derived by

---

processing the data set at (Candido, 2020), but we include for convenience here. The following three sections contain views of our results that we do not include in the main text, but are of interest to some readers of early drafts of this paper. Finally, we present an additional validation set that leads to similar conclusions as the results in the main text. We include this for completeness as some of our early discussions of this work used this validation set, rather than the newer validation set that allows us to compare against the European Centre for Medium-Range Weather Forecasts (ECMWF) ensemble system (ENS).

Computing the Conventional Analog Ensemble with MapReduce

A barrier to operationalizing a global analog ensemble (AnEn) system is processing the corpus of analogs, which can easily grow to 100's of terabytes of data for three-dimensional global predictions over several years. The AnEn algorithm provides a natural partitioning as execution is independent for each $\mathcal{P}$ (grid point and lead time). Every historical forecast and the current prediction contain a piece of data for every grid point. The challenge is to organize the data so that the calculations can be efficiently executed across many datacenter computers.

Our approach is to use the MapReduce paradigm (Dean & Ghemawat, 2004), which allows the computation to run on a distributed computing (cloud) infrastructure like Google's Flume (Chambers et al., 2010). The idea is to break the computation into two subsequent `Map` and `Reduce` phases, each of which operate many times in parallel on different portions of the data and output key-value pairs. Once written this way, the framework can handle scheduling the program's execution across many machines and moving the various subsets of data to the appropriate machines.

The above procedure is accomplished as follows. Let latitude, longitude, pressure, and forecast lead time be the tuple k, which is unique for each grid point. The first `Map` phase scans all historical forecast files and generates a key-value pair $(k, t_f) \to x^i$ for each grid point and $(k, t_f) \to y^i$ for each analysis point. The k corresponds to the location and forecast lead time for the particular x^i (past forecast) and t_f (calendar time being forecast). For every y^i we generate multiple key-value pairs corresponding to a k for *every* lead time the system will forecast.

Notice that the above rule gives each forecast-observation pair a unique key. Prior to the reduce phase all identical keys are grouped into one `Reduce` call by the MapReduce framework. The `Reduce` phase joins these x^i and y^i pairs into a single record and saves them as new key-value pairs $k \to (x^i, y^i)$. This first MapReduce gives us a historical corpus. This corpus could be built in advance of receiving a new forecast to post-process.

The second MapReduce groups the data by grid point and runs the AnEn algorithm. A `Map` phase on the forecast data file from the ECMWF generates key-value pairs $k \to x^f$ for each grid point. The historical corpus key-value pairs are used directly. Note that x^f and every set of candidate analogs $\{(x^i, y^i)\}$ for a grid point have the same key. The data is grouped by key and fed to the `Reduce` phase that has all the data needed to apply equations (1) and (3) to generate the forecast.

Details of Training a Distilled Analog Ensemble Model

This section describes the low-level technical details of the distillation process.

Our training corpus is prepared by using a MapReduce similar to what is described in the previous section to process the set of forecast data files (both the 00Z and 12Z epochs) archived during the training period. Rather than running the AnEn algorithm logic, i.e., equations (1) and (3), at each grid point given a new forecast, we instead save the candidate analogs (forecast-observation pairs) for a given grid point in a single record. These records are stored together on disk for retrieval by the training system, i.e., we have a set of records where each record corresponds to a unique latitude, longitude, pressure, and lead time grid point and contains all the viable forecast-observation pairs at this location at the appropriate lead time.

This data set is used to feed the training process of our deep neural network (DNN). We use a distributed architecture. We train our DNN based on the output of the AnEn, not directly from these forecast-observation pairs. Thus, we need to sample a hypothetical forecast to generate a training example of an input-output pair for the AnEn system. We use 10 datacenter worker processes that sample uniformly among grid points in records on disk, sample a hypothetical wind speed and wind heading forecast, and construct the input to the DNN (corresponding to this grid point and forecast) and the output (of the AnEn algorithm). In the results presented in this paper, we sample heading uniformly and wind speed from a beta distribution with $\alpha = 1.2$, $\beta = 3$, and a coefficient of 100. Effectively this creates a weighted distribution of wind speeds which seems generally applicable to the pressure altitudes ranges of interest in the stratosphere.

Unlike many applications, we do not simply loop over the dataset on disk a fixed number of times or until training error stabilizes. This is because every time we touch a record we sample a new forecast and generate a new input-output pair. Saving these pairs on disk versus generating them online during training is an engineering trade-off, and we have chosen the latter approach.

These examples from the 10 worker processes are injected into a reservoir datacenter process, whose job is essentially to receive new examples, store them in a limited size buffer, and respond to requests (from the learning process) for samples. Rather than choosing a circular buffer or some other first-in, first-out structure, we use a flat data array of 1 million examples and, for each new example, sample an index in the array at random to replace. This means some examples will persist in the buffer longer, and some for a shorter period of time. The typical dwell time of an example in the buffer can be characterized probabilistically. The learning process repeatedly queries the reservoir for batches of training examples, which are selected uniformly at random from examples in the data array. A slowly changing flow of examples where each batch (on average) tends to be drawn from disparate parts of the function mapping being learned is conceptually similar to the replay buffer in deep reinforcement learning (Lin, 1992; Mnih et al., 2015).

We use the Tensorflow (Abadi et al., 2015) library to create and train our DNN. Our network has inputs of latitude, longitude, pressure altitude, forecast lead time, forecast direction, and forecast speed. We transform these into a graph layer that is normalized using the following code snippet where the array 'domain' represents the inputs described above.

```
nlat = tf.multiply(domain[:, 0], 1. / 90.0)
coslng = tf.cos(tf.multiply(domain[:, 1], np.pi / 180.0))
sinlng = tf.sin(tf.multiply(domain[:, 1], np.pi / 180.0))
npre = tf.multiply(tf.subtract(domain[:, 2], 4799.), 1. / (14432. - 4799.))
nlea = tf.multiply(tf.subtract(domain[:, 3], 43200.), 1. / (864000. - 43200.))
coshead = tf.cos(domain[:, 4])
```

```
sinhead = tf.sin(domain[:, 4])
nspeed = tf.multiply(domain[:, 5], 1. / 100.)
normalized_domain = tf.transpose(
    tf.stack([nlat, coslng, sinlng, npre, nlea, coshead, sinhead, nspeeed]))
```

We do this to avoid the discontinuity in longitude being present in our DNN, and to make our inputs have roughly the same order of magnitude (which is a domain trick to decrease training time).

At this point the network consists of 10 fully-connected hidden layers with ReLu activation functions. Each layer has a width of 50 elements. These plus an ultimate layer containing the ultimate post-processed speed and heading forecast (width 2, fully-connected, no activation function) comprise the trained layers of the DNN. We can also include additional network outputs such as forecast uncertainty (standard deviation of the forecast) or ensemble members. This is not discussed in this paper.

To train the network we use stochastic gradient descent with a learning rate of 0.0001 and batch size 100. We train until the root mean square error between the DNN forecasts and the AnEn mean forecasts (from the training examples) stabilizes. In the distilled AnEn used to generate the results in this paper, we trained the DNN with about 6 billion examples.

This network architecture was not tuned for efficiency, but instead chosen to demonstrate how a fairly standard and basic deep learning approach could be used to implement this algorithm.

Statistics of the Loon Data Set

The following plots show the distribution of Loon's approximately 10.5 million observations used for one of the comparisons between algorithms shown in the main text of the paper. This is the intersection of Loon's dataset of observations of stratospheric winds from Loon (`http://www.loon.com`) high altitude balloons (Candido, 2020) and the region and time period for the validation paper used in our study.

Figure S1 shows the distribution of the data over pressure altitude and latitude.

Figure S2 shows the geographical distribution of the data.

Probabilistic Evaluation Metrics for Wind Speed

In the main text of the paper we presented the CRPS, Spread Skill, and Rank Histogram plots for comparing the ensemble systems predictions on wind direction, an omitted plots for wind speed given a similar pattern on skill between the approaches. We include the figures for wind speed in Figure S3.

Confidence Intervals on Deterministic Evaluations

Figures S4 and S5 show the same data as in Figure 2(a) in the main text, but include box plot views of the 90% bootstrap confidence intervals.

Algorithm Skill Comparison By Geography

Figure S6 show the CRMSE averaged across all lead times grouped by geography. One can observe that the Distilled AnEn has higher skill (lower CRMSE) than the baseline ECMWF HRES generally across the stratosphere globally.

Results for an Earlier Validation Period

Our original analysis of the methods included a comparison of AnEn mean against the ECMWF high-resolution deteriministic forecast (HRES) and, for the probabilistic predictions, against a persistence ensemble (PeEn) over a year long validation period from October, 2017 to September, 2018. However, to add a comparison to the ECMWF ENS in a revised version of the manuscript, we changed our validation period in the main text due to data availability.

The PeEn is a simple way to generate an ensemble that consists of selecting the last N available ground-truth values to generate an N-member ensemble. It has been used in, e.g., Alessandrini, Delle Monache, Sperati, and Cervone (2015) and Cervone, Clemente-Harding, Alessandrini, and Monache (2017), as a probabilistic baseline forecast and can be interpreted as the probabilistic extension of a deterministic persistence forecast.

For the results shown in in Figures S7 and S8 the training dataset is the HRES forecasts produced from July, 2016, to September, 2017. We use this to choose weights used in the analog matching process. The validation period is over the HRES forecasts produced from October, 2017, to September, 2018. The data available in the AnEn matching includes all the forecasts in the training dataset plus any additional forecasts between the beginning of the validation time period but prior to the current forecast. This simulates operational use of an AnEn system. To evaluate the distilled AnEn we only use a DNN distilled from the training dataset.

Please refer to the main text of the paper where the relevance of the metrics shown in the below figures are explained in greater detail, albeit for a different validation period.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., … Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.* Retrieved from `https://www.tensorflow.org/` (Software available from tensorflow.org)

Alessandrini, S., Delle Monache, L., Sperati, S., & Cervone, G. (2015). An analog ensemble for short-term probabilistic solar power forecast. *Applied Energy*, *157*, 95–110. Retrieved 2016-11-21, from http://www.sciencedirect.com/science/article/pii/S0306261915009368

Candido, S. (2020, April). *Loon stratospheric sensor data.* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.3763022 doi: 10.5281/zenodo.3763022

Cervone, G., Clemente-Harding, L., Alessandrini, S., & Monache, L. D. (2017). Short-term photovoltaic power forecasting using artificial neural networks and an analog ensemble. *Renewable Energy*, *108*, 274 - 286. Retrieved from http://www.sciencedirect.com/science/article/pii/S0960148117301386 doi: https://doi.org/10.1016/j.renene.2017.02.052

Chambers, C., Raniwala, A., Perry, F., Adams, S., Henry, R., Bradshaw, R., & Nathan. (2010). Flumejava: Easy, efficient data-parallel pipelines. In *Acm sigplan conference on programming language design and implementation (pldi)* (p. 363-375). 2 Penn Plaza, Suite 701 New York, NY 10121-0701. Retrieved from http://dl.acm.org/citation.cfm?id=1806638

Dean, J., & Ghemawat, S. (2004). Mapreduce: Simplified data processing on large clusters. In *Osdi'04: Sixth symposium on operating system design and implementation* (pp. 137–150). San Francisco, CA.

Lin, L.-J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, *8*(3-4), 293–321.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... others (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529.

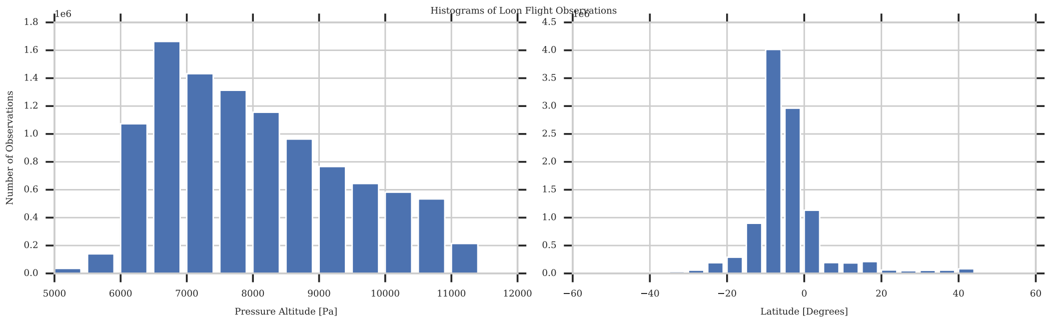


Figure S1. Distribution of Loon's measurements as a function of pressure altitude and latitude.

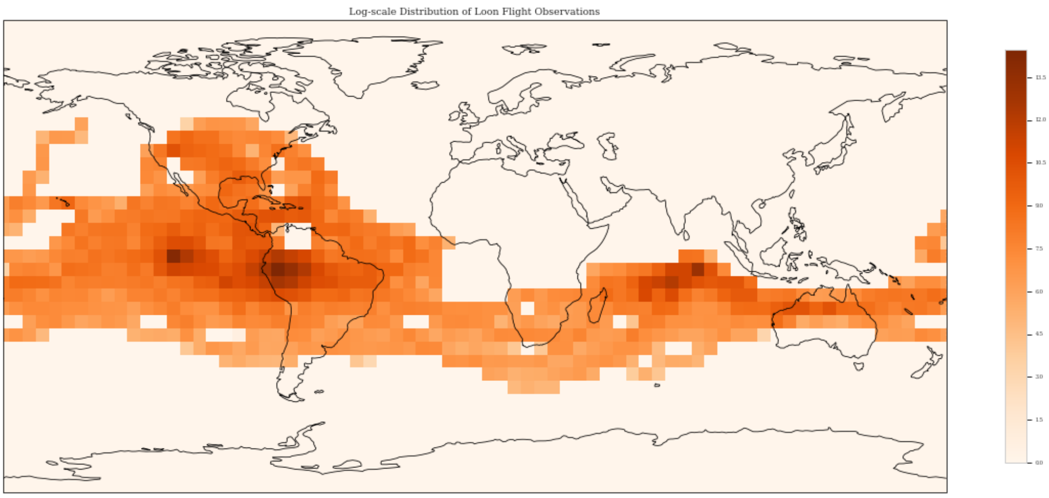


Figure S2. Geographical distribution of Loon's measurements.

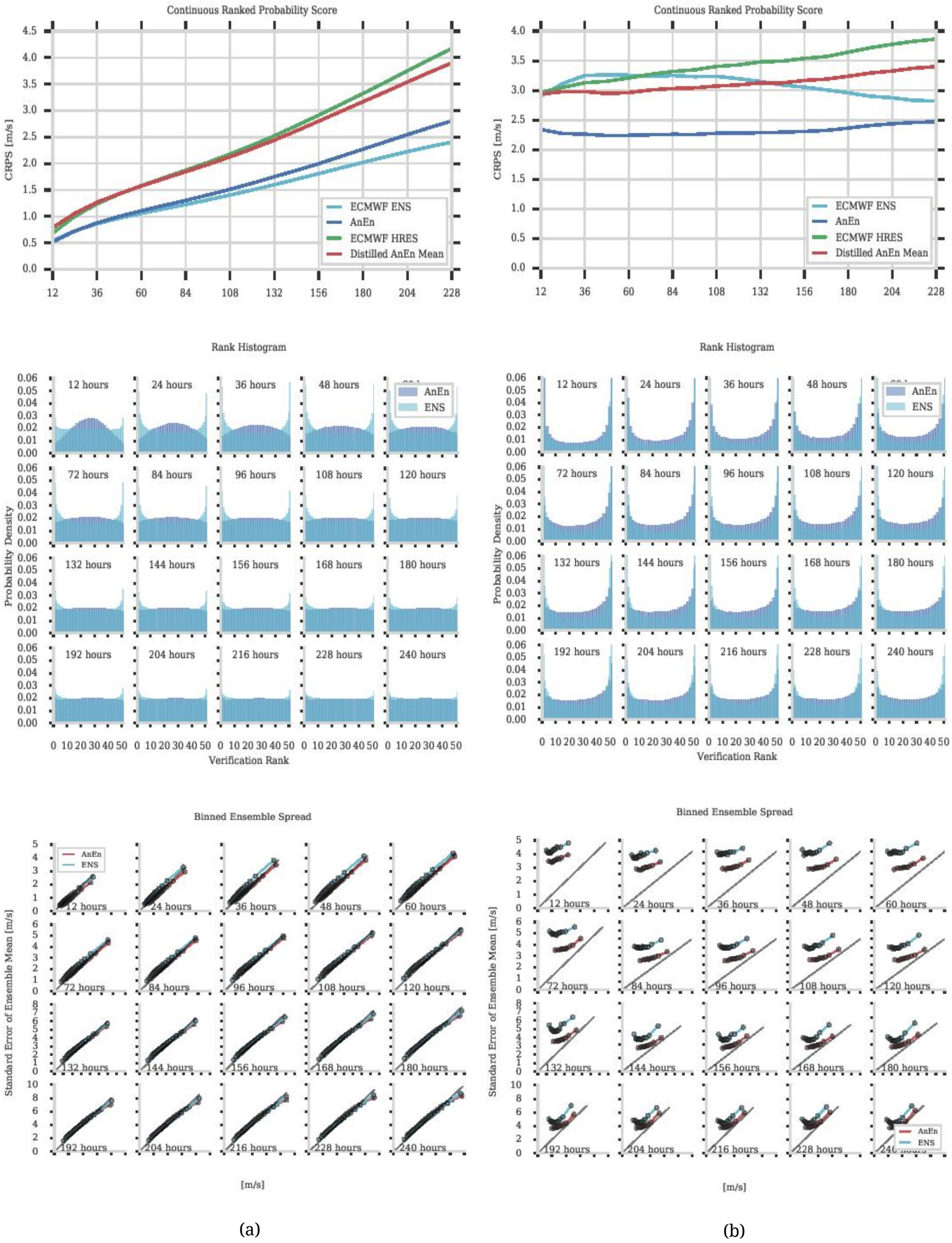


Figure S3. Probabilistic forecast evaluation metrics comparing the AnEn forecast of wind speed to forecasts produced by a ENS. Results with HRES analysis as ground truth are shown on the left (a), while results against Loon's measurements are on the right (b). From top to bottom, the metrics shown are CRPS, rank histogram, and binned-spread skill.

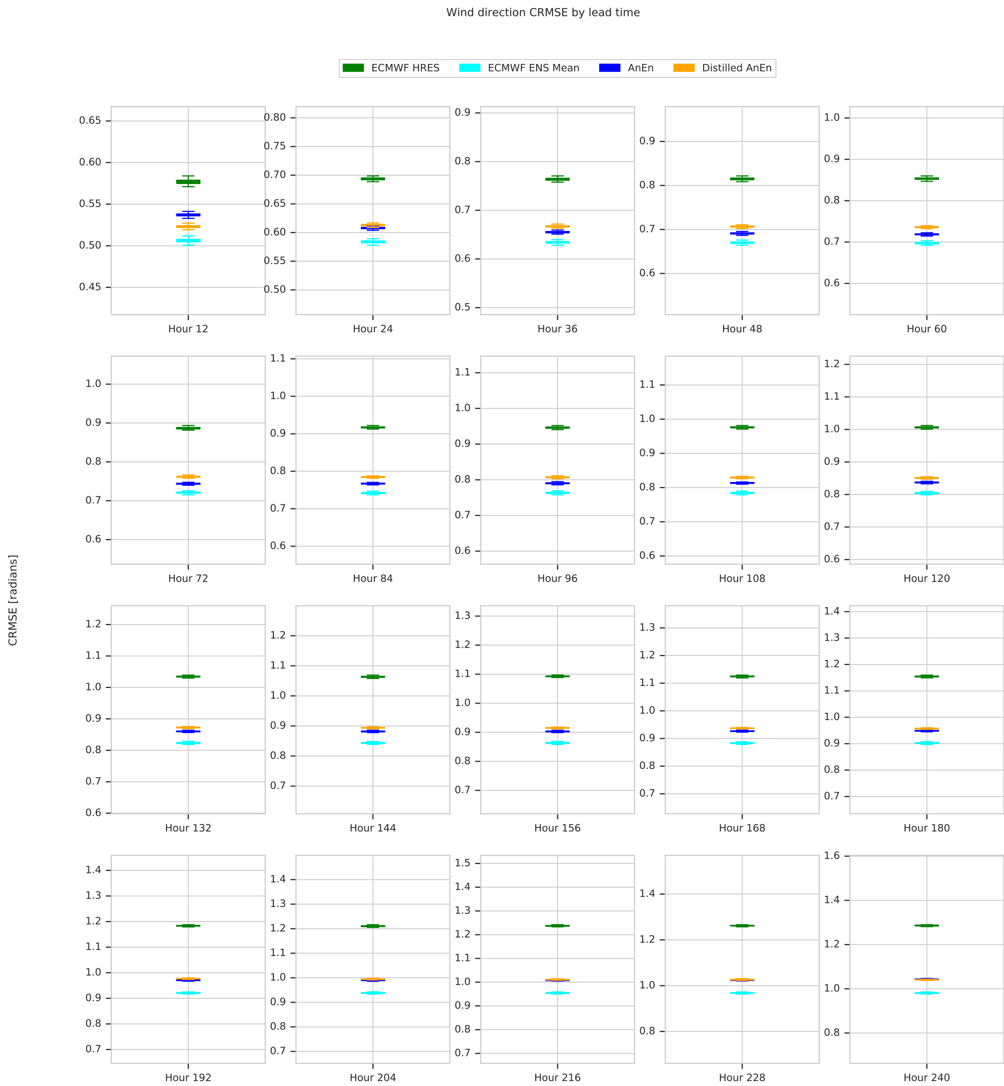


Figure S4. CRMSE for wind direction predictions including boxplots showing the bootstrap 90% confidence intervals.

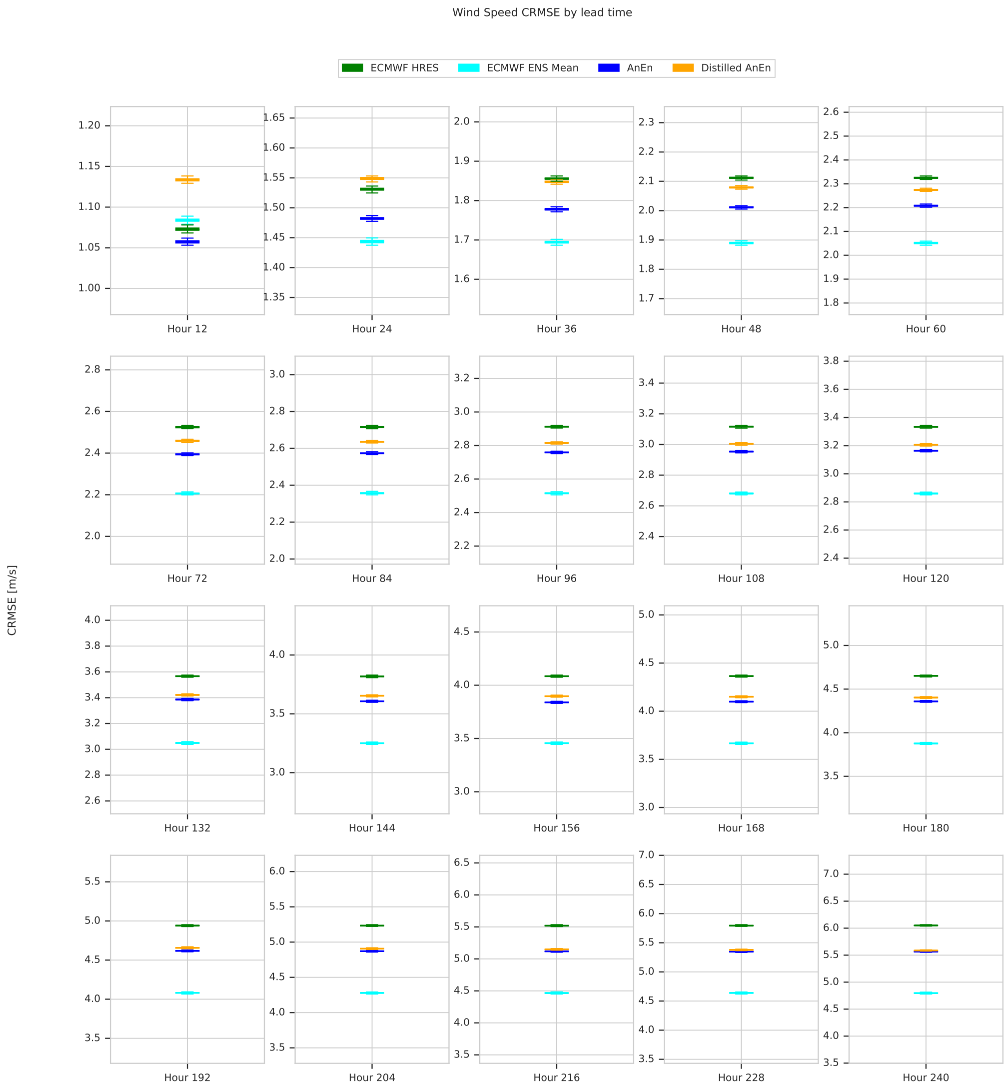


Figure S5. CRMSE for wind speed predictions including boxplots showing the bootstrap 90% confidence intervals.

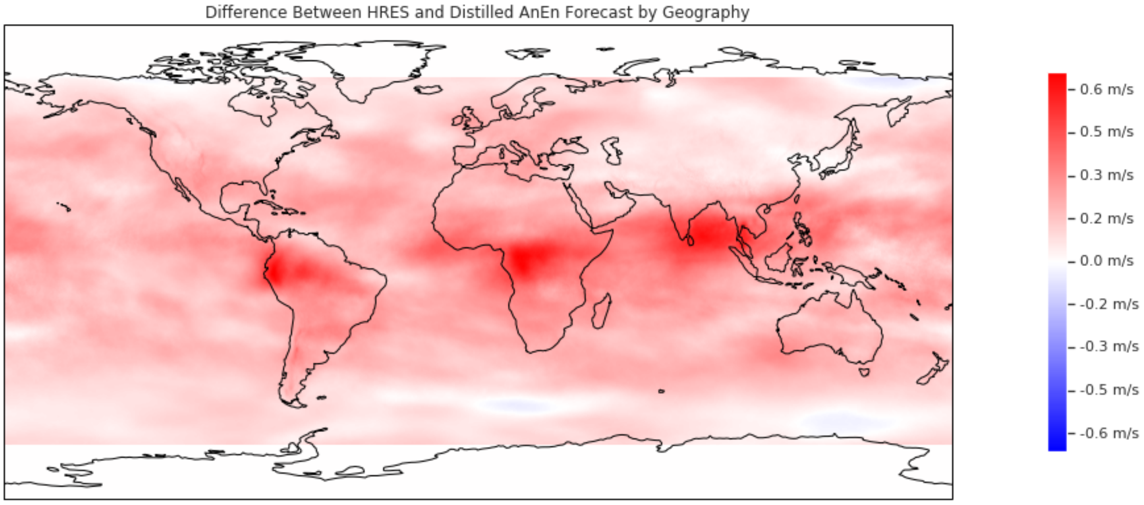
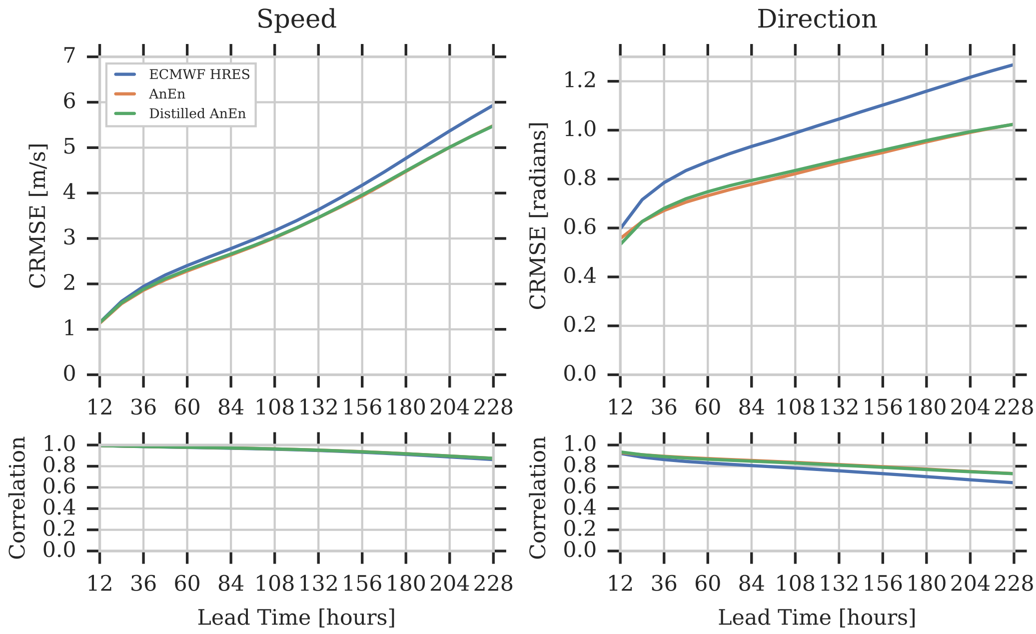
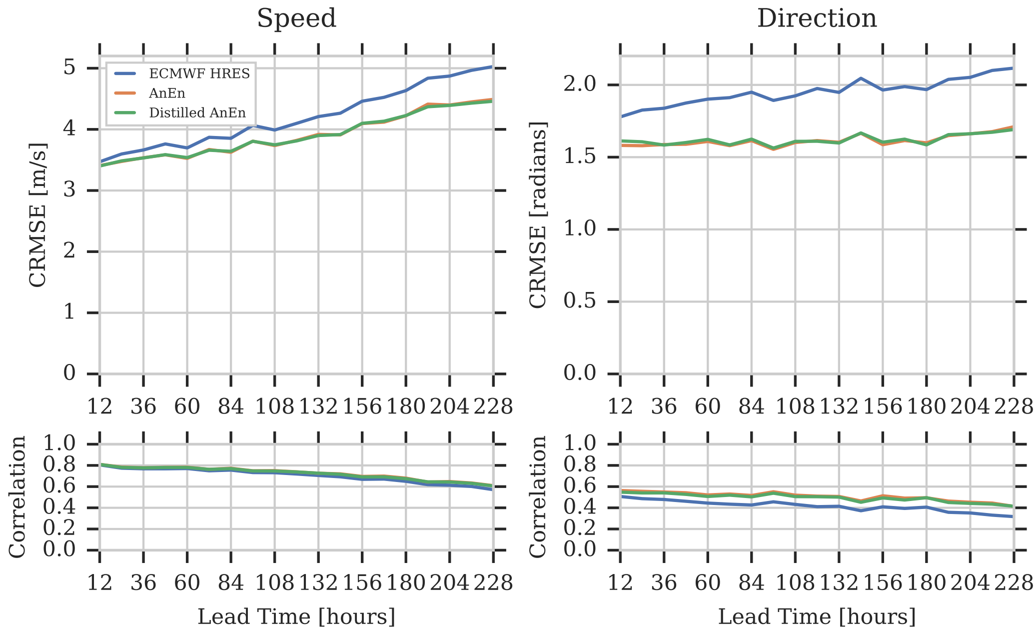


Figure S6. Geographical distribution of CRMSE for the distilled AnEn prediction of wind speed with HRES analysis as ground truth.



(a)



(b)

Figure S7. A deterministic wind speed and direction forecast skill comparison between the HRES, AnEn, and Distilled AnEn over all lead times is shown using as ground truth (a) HRES analysis and (b) Loon observations of stratospheric winds.

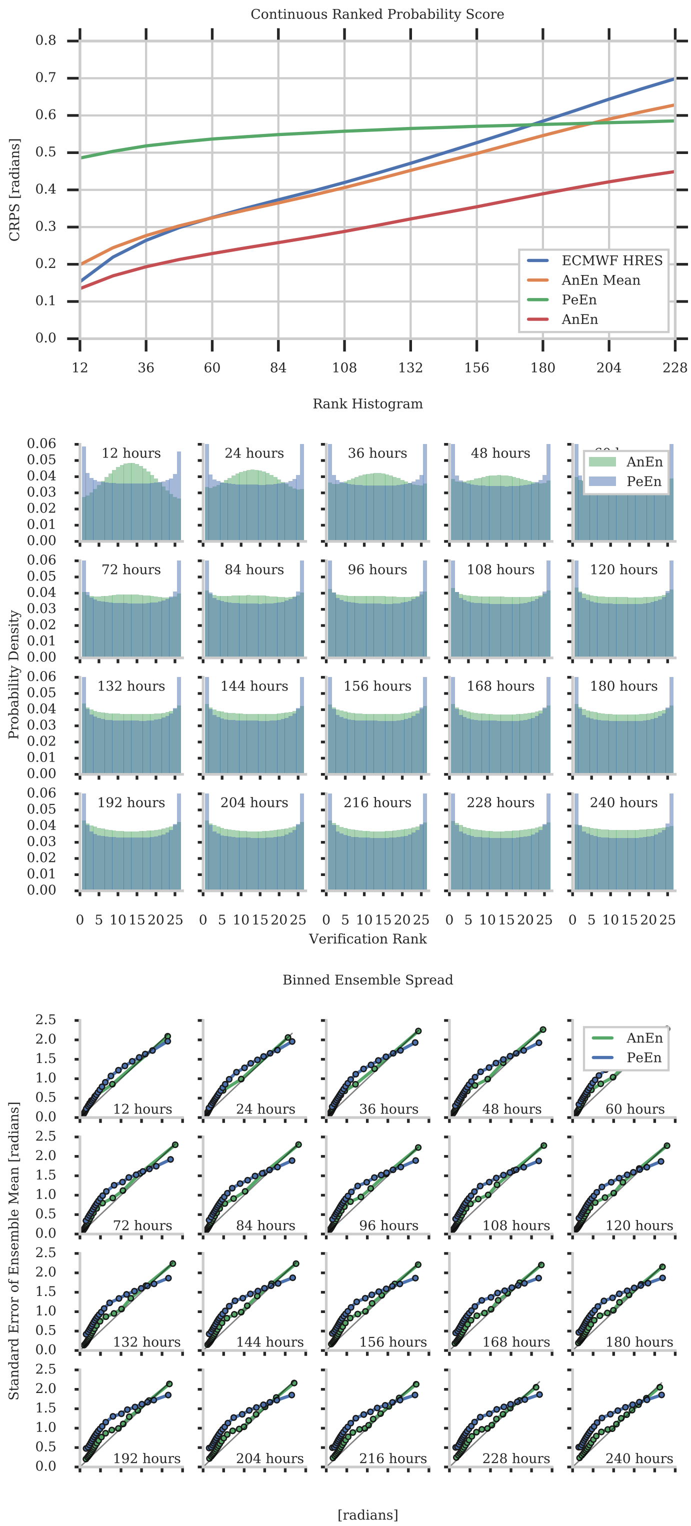


Figure S8. Probabilistic forecast evaluation metrics comparing the AnEn forecast of wind direction to forecasts produced by HRES, AnEn mean, and PeEn using HRES analysis as the ground truth. From top to bottom, the metrics shown are CRPS, rank histogram, and binned-spread skill.